

Dynamic Relative Representations for Goal-Oriented Semantic Communications

Simone Fiorellino^{1,2}, Claudio Battiloro³, Emilio Calvanese Strinati⁴, Paolo Di Lorenzo^{1,2}

¹DIET department, Sapienza University, Rome, Italy. ²CNIT, Parma, Italy.

³Harvard T.H. Chan School of Public Health, Harvard University, Boston, U.S.A.

⁴CEA Leti, University Grenoble Alpes, 38000, Grenoble, France

Emails: {simone.fiorellino,paolo.dilorenzo}@uniroma1.it, cbattiloro@hsph.harvard.edu, emilio.calvanese-strinati@cea.fr

Abstract—In future 6G wireless networks, semantic and effectiveness aspects of communications will play a fundamental role, incorporating meaning and relevance into transmissions. However, obstacles arise when devices employ diverse languages, logic, or internal representations, leading to semantic mismatches that might jeopardize understanding. In latent space communication, this challenge manifests as misalignment within high-dimensional representations where deep neural networks encode data. This paper presents a novel framework for goal-oriented semantic communication, leveraging relative representations to mitigate semantic mismatches via latent space alignment. We propose a dynamic optimization strategy that adapts relative representations, communication parameters, and computation resources for energy-efficient, low-latency, goal-oriented semantic communications. Numerical results demonstrate our methodology’s effectiveness in mitigating mismatches among devices, while optimizing energy consumption, delay, and effectiveness.

Index Terms—Goal-oriented Semantic Communications, Relative Representation, Deep Learning, Stochastic Optimization.

I. INTRODUCTION

Classic communication paradigms prioritize the accurate transmission of individual symbols or bits without explicit consideration of their underlying meaning. As Shannon identified, communication can be understood through three levels: the syntactic level (focusing on the reliable transmission of signals), the semantic level (concerned with the meaning conveyed by those signals), and the effectiveness level (ensuring the message achieves its intended purpose) [1]. However, the exponential growth of data-intensive wireless applications demands a fundamental shift in this approach. The sheer volume of information required by applications like autonomous vehicles and augmented reality overwhelms the capacity of traditional bit-centric communication systems. In contrast, modern networks increasingly demand the exchange of information with shared meaning [2]. Semantic communication (SemCom) addresses this need by embedding meaning directly into transmissions, moving beyond the mere transmission of bits [3]–[6]. To achieve this, SemCom relies heavily on artificial intelligence (AI) and deep neural networks (DNNs) to extract and transmit key semantic features from data rather than the raw data itself. Instead of focusing on the perfect

reconstruction of raw symbols, SemCom prioritizes transmitting the underlying meaning, offering inherent robustness to noise and imperfect transmission conditions. At the receiver side, a DNN decoder reconstructs and interprets the intended message. The flexibility and resilience of SemCom make it ideal for emerging applications where devices operate in dynamic and resource-constrained environments [7]. Building on this idea, novel approaches to semantic joint source and channel coding [5], semantic extraction and compression [8], [9], goal-oriented system design and optimization [7], [10], semantic reasoning [11], and semantic communications based on generative AI [12] have been proposed in the literature. Despite these advancements, disruptions caused by misinterpretations of semantic information, known as *semantic noise*, pose a significant challenge. Such disturbances often arise from discrepancies in how different devices represent and interpret data, leading to errors in understanding, even when the physical transmission is error-free. This issue is particularly acute in dynamic edge environments, where various devices are involved, and complex system-wide retraining is not feasible. Misalignments in high-dimensional data representations can severely disrupt semantic understanding under these conditions [13]. This underscores the need for mechanisms that mitigate semantic mismatches without necessitating system updates. To address these challenges, this paper introduces a novel framework for goal-oriented semantic communication at the wireless network edge based on the recent concept of relative representations (RelReps) [14], [15]. Our main contribution is two-fold: (i) We introduce relative representations in SemComs to mitigate the impact of semantic mismatches in dynamic scenarios with devices having heterogeneous logics; (ii) We introduce a dynamic optimization strategy based on Lyapunov stochastic optimization [16], which dynamically optimizes communication (i.e., rates), computation (i.e., CPU frequency cycles), and learning resources (i.e., anchor sets, encoders) to enable energy-efficient, low-latency, effective semantic communications. The proposed dynamic resource allocation enjoys simple closed-form solutions, and does not require prior knowledge of the statistics of the random variables affecting the system (e.g., wireless channels). Numerical results assess the performance of our methodology in achieving semantic alignment while striking an optimal trade-off between energy consumption, delay, and goal effectiveness.

This work was supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” (PE00000001 - program “RESTART”), and by the SNS JU project 6G-GOALS under the EU’s Horizon program Grant Agreement No 101139232.

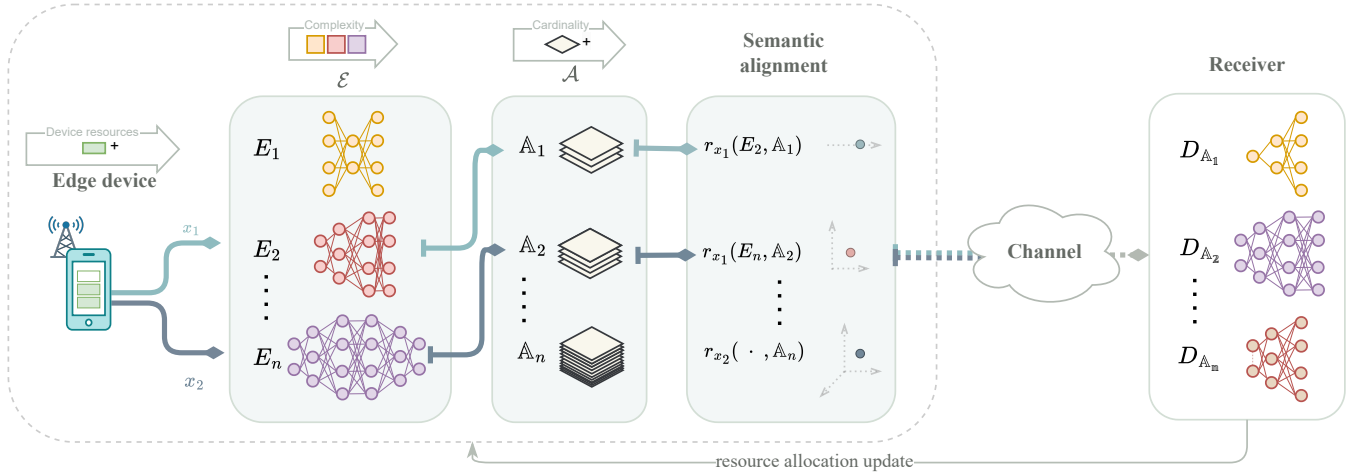


Fig. 1: The edge device (TX) encodes data according to the optimal encoder-anchor set pair (E^*, \mathbb{A}^*) . The RX is equipped with relative decoders trained on shared anchor sets to interpret the transmitted relative representations.

II. RELATIVE REPRESENTATION FOR SEMANTIC COMMUNICATION

SemCom aims to exchange the meaning conveyed by data rather than just the raw data itself. To achieve this, transmitters (TXs) and receivers (RXs) typically employ DNNs to encode and decode semantic information, transmitting absolute representations (raw DNN encoder outputs), as illustrated in Figure 1. In this scenario, semantic noise can arise from discrepancies in how different encoders semantically represent the same information due to stochastic factors or design choices. These misalignments introduce misinterpretations at the decoder, disrupting the overall communication objective.

To address this, we introduce RelRep into the SemCom framework, enabling zero-shot stitching – the ability to communicate effectively across diverse encoders without the need for retraining. With RelRep, a TX can switch encoders on the fly, and the RX can still interpret the message without needing to retrain itself for each new encoder. The rationale of RelRep for SemCom is explained in the sequel.

Relative representation. Given a training dataset $\mathbb{X} = \{\mathbf{x}_i, y_i\}_i$, the encoding function $E(\cdot)$, e.g., a DNN, maps each sample $\mathbf{x}_i \in \mathbb{X}$ to its *absolute representation* $\mathbf{e}_{\mathbf{x}_i} := E(\mathbf{x}_i) \in \mathbb{R}^d$. A subset of \mathbb{X} is defined as the anchors set, $\mathbb{A} = \{\mathbf{a}_1, \dots, \mathbf{a}_{|\mathbb{A}|}\}$. A generic similarity function $sim : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is used to compute the similarity between any two absolute representations. The similarity score r between two such representations is given by $r = sim(\mathbf{e}_{\mathbf{x}_i}, \mathbf{e}_{\mathbf{x}_j})$. Then, the *relative representation* of a sample $\mathbf{x}_i \in \mathbb{X}$ is defined as:

$$\mathbf{r}_{\mathbf{x}_i}(E, \mathbb{A}) := (sim(\mathbf{e}_{\mathbf{x}_i}, \mathbf{e}_{\mathbf{a}_1}), \dots, sim(\mathbf{e}_{\mathbf{x}_i}, \mathbf{e}_{\mathbf{a}_{|\mathbb{A}|}})) \quad (1)$$

We define $D_{\mathbb{A}} : \mathbb{R}^{|\mathbb{A}|} \rightarrow \mathbb{R}^{\mathcal{T}}$ as a relative decoder function. This is specialized in a downstream task, enabling it to interpret the relationships represented in the relative representation and accurately extract meaning.

Semantic Alignment. RelRep effectively mitigates the effects of semantic mismatches by focusing on data relationships (1)

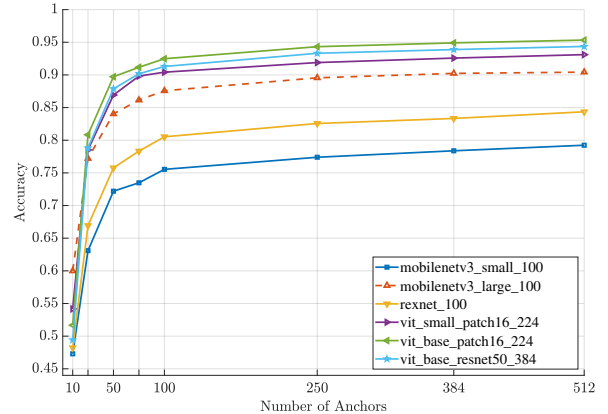


Fig. 2: Performance of zero-shot stitching using different pre-trained encoders and varying anchor set sizes on the CIFAR-10 classification task. The dashed line is end-to-end performance.

Timm model name	# Parameters	FLOPs
mobilenetv3_small_100	1.5M	111.98 M
mobilenetv3_large_100	4.2M	435.36 M
rexnet_100	3.5M	799.28M
vit_small_patch16_224	21.6M	8.5 G
vit_base_patch16_224	85.7M	33.72 G
vit_base_resnet50_384	98.1M	99.04 G

TABLE I: Timm models used in Fig. 2.

rather than absolute representations. This approach facilitates semantically equivalent latent spaces and allows dynamic semantic compression based on the number of anchors used. The relative decoders interpret these relationships, enabling zero-shot stitching without retraining, even when encoders are dynamically switched at the edge. This flexibility is demonstrated in Fig. 2, which shows the performance of model stitching across varying encoding functions and anchor set cardinalities. Notably, zero-shot stitching in absolute cases often fails due to dimensionality mismatches or misalignment, leading to unpredictable behavior across encoders. In Fig. 2, we used pre-trained encoders (obtained by removing the final classifier from models trained on ImageNet – see Table I for

a full list). The relative decoders were trained using a frozen *mobilenetv3_large_100* encoder on the CIFAR-10 dataset. Performance for *mobilenetv3_large_100* is therefore end-to-end, while results for other encoders demonstrate zero-shot stitching capabilities. In the development of the RelRep design, we followed the approach proposed in [14]. We employed cosine similarity as the similarity function, based on the assumption that the angles between data representations remain consistent across different encoding functions. The anchor sets are generated through a uniform random selection of samples from the training dataset. This methodology ensures a balance between simplicity in implementation and robustness in performance outcomes. Future work may explore optimizing the RelRep design to enhance performance in zero-shot stitching scenarios. The flexibility enabled by RelRep, through its dynamic semantic compression and encoder agnosticism, is a major advantage in resource-constrained settings, reducing complexity and deployment overhead for edge devices. Consequently, the semantic understanding workload shifts to the receiver. These advantages motivate the resource optimization strategy we present in the next section, which aims to achieve energy-efficient, goal-oriented semantic communication.

III. DYNAMIC RESOURCE ALLOCATION

In this section, referring to the general communication scheme in Fig. 1, we devise a dynamic allocation of computation (e.g., CPU clock frequency), communication (e.g., data rate), and learning (e.g., number of anchors, encoders) resources for goal-oriented communication involving inference tasks, while promoting a semantic understanding of the transmitted information. We assume to have sets \mathcal{E} and \mathcal{A} of pre-trained encoders and anchor sets, respectively, available at the TX side. The RX is assumed to know \mathcal{A} , and to have a set of relative decoders (one for each anchor set) to guarantee the semantic alignment between the different pre-trained encoders. The relative decoders are assumed to be trained using the RelReps generated from an arbitrary encoder, which could belong or not to \mathcal{E} . In this setting, context parameters like channel quality, data characteristics, and variations in how devices model information (i.e., the semantic language) can fluctuate. Consequently, the proposed dynamic methodology aims to strike an optimal trade-off between power consumption, latency, and inference accuracy by working on the semantic integrity of the communication. The choice of the encoder and the anchor set directly impacts the achievable compression and the computational demands on the TX: Complex models may extract more meaningful information but require higher computational frequencies, consuming more power; similarly, a larger anchor set allows for more nuanced RelReps but increases the volume of data that needs to be transmitted.

A. System Model

We assume a simplified RX that incurs negligible decoding delay. Perfect channel state information (CSI) is assumed to be available at the TX. We work with a time-slot-based model,

where the system adaptively and dynamically optimizes resource allocation within each time-slot. In the following, we will describe our system model in detail.

Power consumption. We consider two sources of power consumption for the TX: local computation and transmission. At time t , the power spent by the TX for local computation is $p_t^c = \kappa(f_t)^3$, where κ is the effective switched capacitance of the processor and f_t is the CPU clock frequency (Hz) [17]. The power spent for transmission at time t is computed from the Shannon formula as

$$p_t^u = \frac{N_0}{|h_t|^2} \left[\exp\left(\frac{R_t}{B} \ln 2\right) - 1 \right],$$

where B is the bandwidth, $|h_t|^2$ the uplink channel power gain, N_0 is the noise power spectral density, and R_t is the data rate. Therefore, the total power consumption is given by

$$p_t = p_t^u + p_t^c. \quad (2)$$

Latency. The overall latency of each time slot has two main sources of delay: the local processing time and the uplink communication time. The choice of the encoder and the anchor set directly impacts both the local processing time due to the encoder complexity and the uplink communication time due to the size of the relative representation to be transmitted induced by a certain anchor set. The *local processing time* can be computed as $L_t^c = \frac{N_{E_t, \mathbb{A}_t}}{f_t}$, where N_{E_t, \mathbb{A}_t} is the number of CPU cycles necessary to encode the message, and depends on the encoder $E_t \in \mathcal{E}$ and the anchor set $\mathbb{A}_t \in \mathcal{A}$ chosen at time t . The *uplink communication time* can be computed as $L_t^u = \frac{n_{\mathbb{A}_t} \cdot q}{R_t}$ where $n_{\mathbb{A}_t} \cdot q$ is the number of bits to be transmitted, and it depends on the size of the relative representation, i.e., the cardinality of the anchor set \mathbb{A}_t . Therefore, the total latency is given by

$$L_t = L_t^u + L_t^c. \quad (3)$$

Learning Accuracy. We assume that the RX is provided with a validation set \mathbb{X}^{VAL} . A task-dependent function $G_{E, \mathbb{A}}$ is introduced to measure the learning performance, given a pair of encoder and anchor set E, \mathbb{A} . As an example, let us consider a classification task, whose validation accuracy can be used to estimate learning performance as

$$G_{E, \mathbb{A}} = \frac{1}{|\mathbb{X}^{\text{VAL}}|} \sum_{\mathbf{x}, y \in \mathbb{X}^{\text{VAL}}} \mathbb{I}(\hat{y}_{E, \mathbb{A}}(\mathbf{x}) = y), \quad (4)$$

where $\hat{y}_{E, \mathbb{A}}(\mathbf{x})$ is the prediction for the sample \mathbf{x} if $E \in \mathcal{E}$ and $\mathbb{A} \in \mathcal{A}$ are chosen as encoder and anchor set, respectively. Examples of how the performance metric in (4) is affected by different encoders and anchor sets are illustrated in Fig. 2.

B. Problem Formulation

We can now formulate the problem of dynamic resource allocation for goal-oriented communication. The aim is to find the optimal policy to allocate, at each t , the uplink data rate R_t , the CPU cycles at devices f_t , the encoder E_t and the anchor set \mathbb{A}_t to minimize the long-term average system power consumption in (2), with constraints on the average

inference accuracy in (4) and the average latency in (3). Then, the dynamic resource allocation problem can be cast as:

$$\begin{aligned}
& \min_{R_t, f_t, E_t, \mathbb{A}_t} \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{p_\tau\} & (5) \\
& \text{subject to} \quad (a) \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{\mathbf{1}_{[L_\tau \geq L^{\text{IST}}]}\} \leq p^{\text{IST}}; \\
& (b) \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{L_\tau\} \leq \bar{L}; \quad (c) \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{G_\tau\} \geq \bar{G}; \\
& (d) R^{\text{MIN}} \leq R_t \leq R^{\text{MAX}}; \quad (e) f_t^{\text{MIN}} < f_t \leq f_t^{\text{MAX}}; \\
& (f) E_t \in \mathcal{E}; \quad (g) \mathbb{A}_t \in \mathcal{A}.
\end{aligned}$$

Expectations are taken over the relevant random variables in the system, i.e., the channel condition and the available computation frequency at the TX side. G_τ is a short-hand notation for G_{E_t, \mathbb{A}_t} in (4). Therefore, we are implicitly assuming that the accuracy evaluated on a validation set is a good proxy for the inference (test) accuracy. This is a standard and reasonable assumption if no shifts in the data distribution happen. The constraints of the problem (5) have the following meanings: (a) ensures that the probability for the instantaneous latency to exceed a certain value L^{IST} does not exceed a threshold p^{IST} ; (b) ensures that the mean latency does not exceed a threshold \bar{L} ; (c) ensures that the mean accuracy of the system does not decrease under a certain threshold \bar{G} . Imposing both constraints (a) and (b) is instrumental to better control the statistical distribution of latency. Finally, the constraints in (d)-(g) impose instantaneous bounds (e.g., minimum rates and maximum CPU frequencies) on the resource variables R_t, f_t , and impose that E_t and \mathbb{A}_t can take values only from finite sets \mathcal{E} and \mathcal{A} encoders and anchor sets, respectively.

C. Algorithmic Solution via Stochastic Optimization

We now introduce a dynamic algorithmic framework to solve the long-term optimization problem (5), by recasting it as a stability problem via the tools of stochastic Lyapunov optimization [16]. First, we define the virtual queues Z_t, Q_t and Y_t to enforce the constraints (a)-(b)-(c), respectively, as

$$Y_{t+1} = \max\{0, Y_t + \epsilon_y(\mathbf{1}_{[L_\tau \geq L^{\text{IST}}]} - p^{\text{IST}})\} \quad (6)$$

$$Z_{t+1} = \max\{0, Z_t + \epsilon_z(L_t - \bar{L})\} \quad (7)$$

$$Q_{t+1} = \max\{0, Q_t + \epsilon_q(\bar{G} - G_t)\} \quad (8)$$

where ϵ_z, ϵ_q , and ϵ_y are positive step sizes used to control the convergence speed of the algorithm. At this point, we define the Lyapunov function as $\mathcal{U}_t = \mathcal{U}(\Phi_t) = \frac{1}{2}(Z_t^2 + Q_t^2 + Y_t^2)$, with $\Phi_t = [Z_t^2, Q_t^2, Y_t^2]$. Then, the drift-plus-penalty function is $\Delta_t^p = \mathbb{E}\{\mathcal{U}_{t+1} - \mathcal{U}_t + V \cdot p_t^{\text{TOT}} | \Phi_t\}$. Now, we want to minimize an upper bound of the drift-plus-penalty [16]:

$$\begin{aligned}
\Delta_t^p \leq \xi + \mathbb{E}\{ & Z_t(L_t - \bar{L}) + Q_t(G_t - \bar{G}) + \\
& + Y_t(L_t + 1 - p^{\text{IST}}) + V \cdot p_t^{\text{TOT}} | \Phi_t \} & (9)
\end{aligned}$$

with ξ a positive finite constant, obtained following a similar procedure to [18]. Finally, using stochastic approximation

Algorithm 1 Resource Optimization for SemCom

Inputs: Sample \mathbf{x}_t

Output: $E_t^*, \mathbb{A}_t^*, R_t^*, f_t^*$

for $t \geq 0$ **do**

(RX) Update the virtual queues Z_t, Q_t, Y_t ;

(RX) Compute f_t^* and R_t^* as in (12);

(RX) Set $E_t^*, \mathbb{A}_t^* = \arg \min_{E_t \in \mathcal{E}, \mathbb{A}_t \in \mathcal{A}} \Psi(E_t, \mathbb{A}_t, R_t^*, f_t^*)$;

(TX) Compute and transmit $r_{\mathbf{x}_t}(E_t^*, \mathbb{A}_t^*)$;

end for

arguments [16], we optimize the upper bound of the drift-plus-penalty, removing the expectation, thus getting the following per-slot problem (omitting all constant terms):

$$\begin{aligned}
& \min_{R_t, f_t, E_t, \mathbb{A}_t} (Z_t + Y_t)L_t - Q_t G_t + V \cdot p_t^{\text{TOT}} \\
& \text{subject to} \quad (f) E_t \in \mathcal{E}; \quad (g) \mathbb{A}_t \in \mathcal{A}; \\
& (d) R^{\text{MIN}} \leq R_t \leq R^{\text{MAX}}; \quad (e) f_t^{\text{MIN}} < f_t \leq f_t^{\text{MAX}}. & (10)
\end{aligned}$$

Now, because of (d)-(e)-(f)-(g), (10) is a mixed-integer nonlinear optimization problem, which might be very complicated to solve. However, for any given \mathbb{A}_t and E_t , problem (10) becomes a convex problem in R_t and f_t . Therefore, we can derive the optimal uplink rate R_t^* and CPU frequency f_t^* by imposing the KKTs conditions [19], obtaining

$$R_t^* = \left[\frac{2B}{\log 2} W \left(\frac{1}{2} \sqrt{\frac{(Z_t + Y_t)n_{\mathbb{A}_t} q |h_t|^2 \ln 2}{BV N_0}} \right) \right]_{R^{\text{MIN}}}^{R^{\text{MAX}}} \quad (11)$$

$$f_t^* = \left[\sqrt[4]{\frac{(Z_t + Y_t)N_t}{3V\kappa}} \right]_{f_t^{\text{MIN}}}^{f_t^{\text{MAX}}} \quad (12)$$

At this point, to determine the optimal encoder and anchor set, the algorithm needs to evaluate the objective function for each possible combination, setting the optimal communication and computation parameters as in (11)-(12). This means that the system performs $|\mathcal{E}| \cdot |\mathcal{A}|$ independent function evaluations, where $|\mathcal{E}|$ and $|\mathcal{A}|$ are the cardinalities of the encoder and anchor sets, respectively. In this context, the computational cost associated with a single-function evaluation is generally negligible, but greedy approaches could be employed to alleviate the burden further. An overview of the methodology is described in Alg.1, where we denote as $\Psi(R_t, f_t, E_t, \mathbb{A}_t)$ the objective in problem (10).

D. Numerical Results

To assess the performance of our dynamic resource allocation framework, we run Algorithm 1 while varying the constraints on average latency (\bar{L}) and average accuracy (\bar{G}) in (5). This allows us to explore the inherent trade-offs between power consumption, latency, and inference accuracy, highlighting our approach's flexibility. As discussed in Sec. II, we focus only on RelReps due to the unpredictable behavior of the absolute case in the zero-shot stitching setting. Our

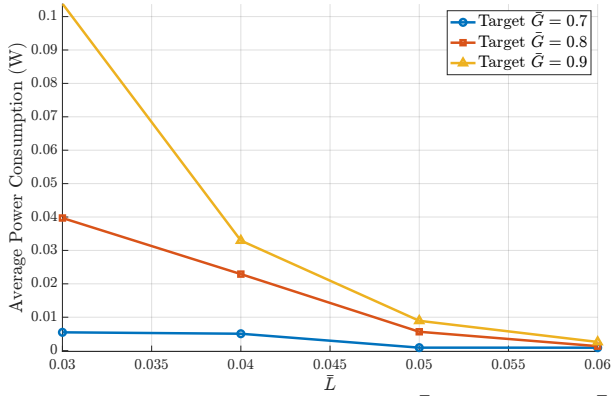


Fig. 3: Long-term average power vs \bar{L} , for different \bar{G} .

simulation setup aligns with the preliminary study in Fig. 2, employing a set of pre-trained encoders \mathcal{E} (see Table I) and anchor sets \mathcal{A} . We optimize, across 5 random seeds, according to the following \bar{L} values [0.03,0.04,0.05,0.06] (s) and \bar{G} values [0.7,0.8,0.9]. The channel takes into account path loss propagation, the message quantization (q) is set to 32 bits, and the channel bandwidth is 1 MHz. To ensure timely responses, we define a maximum tolerable latency of $L^{\text{IST}} = \bar{L} + 7.5 \times 10^{-3}$ s with a corresponding probability threshold of $p^{\text{IST}} = 0.3$. Then, in Fig. 3, we illustrate the behavior of the average power versus the average latency constraint \bar{L} , for different accuracy constraint \bar{G} . As we can notice from Fig. 3, RelReps enable the effective exchange of semantic information for all system configurations; as expected, more stringent accuracy constraints necessitate more computationally expensive choices for a given latency constraint. Finally, Fig. 4 illustrates the instantaneous behavior of latency (top) and accuracy (bottom) for a specific system configuration, i.e., $\bar{L} = 0.04$ s and $\bar{G} = 0.8$, showing the capability of Algorithm 1 to guarantee the long-term constraints in (5).

IV. CONCLUSIONS

This paper introduced a novel framework for goal-oriented and SemCom, addressing the challenges posed by semantic mismatches in dynamic environments. Our key contributions lie in using relative representations for robust communication across diverse devices and a dynamic resource optimization strategy for goal-oriented tasks. This strategy adapts communication, computation, and learning resources to enable energy-efficient, low-latency inference crucial for edge applications. Numerical results validate the effectiveness of our framework in achieving a favorable trade-off between energy consumption, delay, and inference reliability. Future developments include the optimal selection of anchors and similarity functions used in RelReps, with the aim of improving the performance of goal-oriented semantic communications.

REFERENCES

- [1] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.
- [2] E. C. Strinati and S. Barbarossa, "6G networks: Beyond shannon towards semantic and goal-oriented communications," *Computer Networks*, vol. 190, p. 107930, 2021.

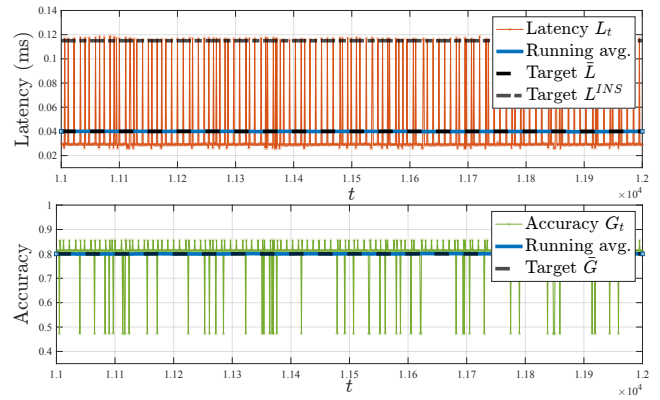


Fig. 4: Latency and accuracy vs t , $\bar{L} = 0.04$ s and $\bar{G} = 0.8$.

- [3] B. Güler, A. Yener, and A. Swami, "The semantic communication game," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 787–802, 2018.
- [4] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, "Towards a theory of semantic communication," in *2011 IEEE Network Science Workshop*. IEEE, 2011, pp. 110–117.
- [5] 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, 2022.
- [6] E. C. Strinati, P. Di Lorenzo *et al.*, "Goal-oriented and semantic communication in 6G AI-native networks: The 6G-goals approach," *arXiv preprint arXiv:2402.07573*, 2024.
- [7] P. Di Lorenzo, M. Merluzzi, F. Binucci, C. Battiloro, P. Banelli, E. C. Strinati, and S. Barbarossa, "Goal-oriented communications for the IoT: System design and adaptive resource optimization," *IEEE Internet of Things Magazine*, vol. 6, no. 4, pp. 26–32, 2023.
- [8] M. Kountouris and N. Pappas, "Semantics-empowered communication for networked intelligent systems," *IEEE Communications Magazine*, vol. 59, no. 6, pp. 96–102, 2021.
- [9] P. A. Stavrou and M. Kountouris, "The role of fidelity in goal-oriented semantic communication: A rate distortion approach," *IEEE Transactions on Communications*, 2023.
- [10] F. Binucci, P. Banelli, P. Di Lorenzo, and S. Barbarossa, "Adaptive resource optimization for edge inference with goal-oriented communications," *EURASIP J. on Advances in Signal Proc.*, no. 1, p. 123, 2022.
- [11] C. K. Thomas, E. C. Strinati, and W. Saad, "Reasoning with the theory of mind for pragmatic semantic communication," *arXiv preprint arXiv:2311.18224*, 2023.
- [12] S. Barbarossa, D. Comminiello, E. Grassucci, F. Pezone, S. Sardellitti, and P. Di Lorenzo, "Semantic communications based on adaptive generative models and information bottleneck," *IEEE Communications Magazine*, vol. 61, no. 11, pp. 36–41, 2023.
- [13] M. Sana and E. C. Strinati, "Semantic channel equalizer: Modelling language mismatch in multi-user semantic communications," *arXiv preprint arXiv:2308.03789*, 2023.
- [14] L. Moschella, V. Maiorca, M. Fumero, A. Norelli, F. Locatello, and E. Rodola, "Relative representations enable zero-shot latent space communication," *arXiv preprint arXiv:2209.15430*, 2022.
- [15] A. Norelli, M. Fumero, V. Maiorca, L. Moschella, E. Rodola, and F. Locatello, "Asif: Coupled data turns unimodal models to multimodal without training," *Proc. of NeurIPS*, vol. 36, 2024.
- [16] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool Publ., 2010.
- [17] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 13, no. 2-3, pp. 203–221, 1996.
- [18] M. Merluzzi, P. Di Lorenzo, S. Barbarossa, and V. Frascolla, "Dynamic computation offloading in multi-access edge computing via ultra-reliable and low-latency communications," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 342–356, 2020.
- [19] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.