

Semantic-Aware Task Offloading for QoE-Driven Resource Allocation in Mobile Edge Computing

Luciano de S. Fraga, Leizer de L. Pinto, Kleber V. Cardoso

Abstract—Semantic communication (SemCom) reduces communication overhead by transmitting task-relevant information instead of raw data, making it suitable for remote inference with large neural networks. We propose a semantic-aware offloading framework that combines a Vision Transformer (ViT) for semantic feature extraction with joint source-channel coding (JSCC) for noisy wireless transmission. We derive an accuracy function capturing the effects of semantic compression and channel quality on image classification tasks. We define a quality-of-experience (QoE) model considering latency, energy consumption, and accuracy. The resulting optimization problem is solved heuristically. Simulations demonstrate improved QoE, lower overhead, and better scalability than baseline approaches.

Keywords—Semantic communications, Mobile edge computing, Resource allocation, Task offloading.

I. INTRODUCTION

The evolution toward sixth-generation (6G) wireless networks is driving an unprecedented surge in data generation, connectivity density, and computational demands across heterogeneous intelligent applications [1]. Emerging services such as autonomous driving, immersive extended reality (XR), high-resolution video analytics, and multi-modal machine intelligence impose strict constraints on latency, energy efficiency, and reliability, thereby challenging the limited processing and battery capabilities of mobile devices [2], [1]. Mobile edge computing (MEC) has emerged as a promising solution, enabling user equipment (UE) to offload computation-intensive workloads to nearby edge servers and thus reduce local processing delays while mitigating energy consumption [3], [4]. However, the effectiveness of conventional MEC systems is fundamentally restricted by the communication bottleneck inherent in wireless uplinks, particularly in scenarios involving large data volumes, multi-task environments, and volatile channel conditions [5], [6], [7].

Traditional communication systems operate under the Shannon paradigm, which aims to reliably transmit bit sequences irrespective of the underlying task semantics. While this approach is adequate for data-centric delivery, it becomes inefficient for task-oriented applications in which only a subset of the transmitted information is relevant to the task outcome [2], [1]. Transmitting full-resolution data, such as raw images, text sequences, or multi-modal sensory streams, results in high

bandwidth usage, offloading latency, and degraded quality-of-service (QoS) under constrained resources.

To address these challenges, semantic communication (SemCom) has recently emerged as a paradigm-shifting technology that aims to transmit meaning rather than bits [1]. By extracting high-level task-relevant representations, semantic encoders substantially reduce the communication load while maintaining task performance [8], [9], [10]. SemCom has demonstrated remarkable efficiency gains in text transmission, image classification, video analytics, and multi-modal tasks [8], [9], [10]. When integrated with MEC, semantic communication holds the potential to reshape task offloading strategies by coupling communication efficiency with task-level performance [11], [12], [5].

Existing MEC works predominantly focus on optimizing latency, energy consumption, or offloading decisions under conventional bit-level communications [6], [3], [4]. More recently, semantic-aware MEC frameworks have incorporated semantic extraction into the offloading process [5], [7], [11]. However, most existing studies do not explicitly characterize the relationship between wireless channel conditions, semantic extraction factors, and inference accuracy, despite this tradeoff being fundamental for reliable semantic communications [5], [7], [11]. In addition, several prior works assume simplified architectures with a single edge server and neglect wired network latency, limiting their applicability to practical multi-tier edge infrastructures with multiple candidate offloading points. Furthermore, many existing approaches rely on computationally intensive deep reinforcement learning (DRL)-based optimization methods [11], [12], which may present high complexity and limited suitability for real-time deployment.

These limitations motivate the need for semantic-aware offloading frameworks capable of jointly considering semantic compression, communication conditions, and distributed computing resources. Therefore, this paper makes the following contributions: (1) We investigate the relationship between semantic extraction, wireless channel conditions, and inference accuracy; (2) We propose a semantic-aware multi-offloading framework that jointly accounts for wireless transmission and wired network latency; (3) We develop a low-complexity heuristic to efficiently solve the mixed integer nonlinear programming (MINLP) optimization problem.

The remainder of this paper is organized as follows. Section II presents the system model and problem formulation. Section III describes the heuristic approach used to solve the MINLP problem. Section IV presents the implementation details and experimental results. Finally, Section V concludes the paper.

Luciano de S. Fraga, Leizer de L. Pinto and Kleber V. Cardoso are with the Institute of Informatics, Universidade Federal de Goiás, 74691-300 Goiânia, GO, Brazil (e-mail: lucianosouza@discente.ufg.br; kleber@ufg.br; leizer@ufg.br). This work was partially supported by CAPES, by MCTIC/CGI.br/FAPESP under grant no. 2020/05127-2 (SAMURAI project), by CNPq under grant no. 306283/2025-5, and by RNP/MCTIC under grant no. 01245.020548/2021-07 (Brasil 6G project).

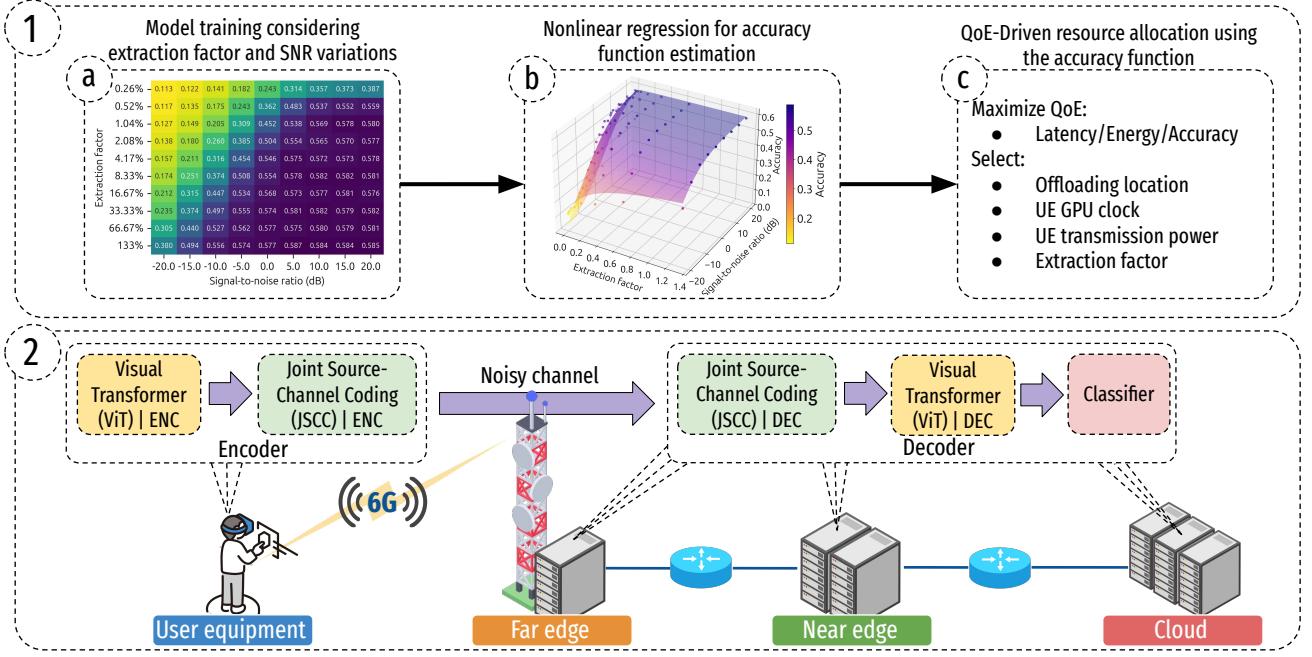


Fig. 1: SemCom framework in which the accuracy function is derived for use in resource allocation systems (top), considering an architecture with multi-tier offloading points (bottom).

II. SYSTEM MODEL

We consider a semantic-aware task offloading system composed of a set of users $\mathcal{N} = \{1, \dots, |\mathcal{N}|\}$ connected to a next-generation NodeB (gNB) in a 5G/6G network. The gNB allocates resource blocks (RBs) to users through an orthogonal frequency-division multiple access (OFDMA) scheme. For simplicity, each UE is assigned a single RB, and the number of available RBs is assumed sufficient to serve all users.

Each UE generates image-classification tasks that can be executed locally or offloaded to a remote server in the set $\mathcal{S} = \{1, \dots, |\mathcal{S}|\}$. These servers may operate as edge servers (ESs), with heterogeneous computational capabilities and geographical locations, or as cloud servers (CSs), which provide higher computational resources at the cost of larger network latency.

Each UE $n \in \mathcal{N}$ may apply a semantic encoder with extraction factor $\mu_n \in [0, 1]$, which controls the semantic compression level over a noisy channel. When offloading is selected, the UE transmission power p_n also becomes an optimization variable that directly affects inference accuracy together with the extraction factor. To capture the impact of semantic extraction and transmission quality over accuracy, we apply non-linear regression to several trained models under different signal-to-noise ratio (SNR) conditions, obtaining an accuracy function suitable for QoE-driven resource allocation.

Fig. 1 illustrates the proposed framework. First, several models are trained under different extraction factors and SNR conditions to collect inference accuracy data (1.a). Next, non-linear regression is applied to derive an accuracy function representing its relationship with semantic compression and transmission quality (1.b). The obtained function is then incorporated into a QoE-driven resource allocation framework that jointly optimizes energy consumption, inference accuracy,

and latency (1.c). The semantic-aware image classification system consists of semantic and channel encoders located at the UE, while the semantic/channel decoders and classifier may be executed locally or offloaded to remote servers (2). The semantic encoder/decoder is implemented using a Visual Transformer (ViT) [13], while the channel encoder/decoder adopts a joint source-channel coding (JSCC) architecture [14].

A. Semantic extraction

The amount of data transmitted by user n after applying a semantic encoder with a compression ratio μ_n is

$$\tilde{D}_n = \mu_n D_n, \quad (1)$$

where D_n represents the original data size. The local semantic extraction latency can be defined as

$$t_n^{\text{ext}} = \begin{cases} 0, & \text{if } \mu_n = 1, \\ \frac{l_n^{\text{SE}}}{\mu_n c_n}, & \mu_n^{\min} \leq \mu_n < 1, \end{cases} \quad (2)$$

where l_n^{SE} is the graphics processing unit (GPU) consumption at the UE for semantic extraction. $c_n = \alpha NU f_n$ is the GPU computational capability of UE n . $\alpha = 2$ corresponds to the floating-point operations (FLOPs) per cycle per core (FLOPs/cycle/core). NU represents the number of compute unified device architecture (CUDA) cores available on UE n . f_n denotes the GPU clock frequency of UE n .

B. Network model

The achievable uplink rate for user n is

$$R_n = B \log_2 (1 + \text{SNR}_n), \quad (3)$$

where the SNR is defined as $\text{SNR}_n = g^2 p_n / \sigma^2$, B is the bandwidth, p_n is the transmit power decision variable, g is

the channel gain, and σ^2 is the additive white Gaussian noise (AWGN) power. For simplicity g and σ^2 are constant. The network latency consists of the wireless transmission latency and the wired network latency between the gNB and the remote server s , denoted by t_s^{wired} , and can be defined as

$$t_n^{\text{tx}} = \frac{\tilde{D}_n}{R_n} + t_s^{\text{wired}}. \quad (4)$$

C. Computation models

If offloaded, the remote server s execution time is

$$t_n^{\text{rem}} = \frac{\sum_{n=1}^N l_n^s}{c_s}, \quad (5)$$

where $c_s = \alpha N S f_s$ is the GPU computation capability of the server s . l_n^s is the GPU consumption of the task offloaded by the UE n to the server s . $N S$ denotes the number of CUDA cores available on the server. f_s denotes the GPU clock frequency of the server. Local execution latency can be denoted as $t_n^{\text{loc}} = l_n^U / c_n$, where l_n^U represents the total task workload allocated to UE n . Finally, the total processing and network latency can be formulated as

$$t_n = \rho_n (t_n^{\text{ext}} + t_n^{\text{tx}} + t_n^{\text{rem}}) + (1 - \rho_n) t_n^{\text{loc}}, \quad (6)$$

where $\rho_n = \sum_{s=1}^S x_n^s$ represents the offloading indicator (1 = offload after semantic extraction, 0 = no offloading). The decision variable $x_n^s = 1$ indicates that the user n decoder workload is offloaded to the server s , otherwise $x_n^s = 0$.

D. Energy consumption model

The total energy expenditure for processing a task is determined by the energy consumed at the user device and the remote server. The user-side energy consumption E_n^U accounts for both local computation and semantic extraction, as well as wireless transmission:

$$E_n^U = \kappa_n (\rho_n t_n^{\text{ext}} + (1 - \rho_n) t_n^{\text{loc}}) f_n^3 + p_n (t_n^{\text{tx}} - t_s^{\text{wired}}). \quad (7)$$

The first term represents user n GPU energy consumption, considering semantic extraction and local processing. The coefficient κ_n is the energy coefficient of UE n . The second term accounts for the radio-frequency energy consumed during the wireless transmission of the semantically encoded data. The remote-side computation energy for processing the offloaded portion of the task is given by:

$$E_s = \kappa_s t_n^{\text{rem}} f_s^3, \quad (8)$$

where κ_s is the energy coefficient of the remote server s . The total energy consumed by UE n tasks demands is:

$$E_n = E_n^U + E_s. \quad (9)$$

E. Accuracy function

To capture the relationship between semantic extraction, SNR, and task accuracy, we apply non-linear regression to the data collected during the test process of our implemented semantic-aware image classification model (Fig. 2). The general function used as a baseline for the regression can be defined as

$$f(\mu_n, \text{SNR}_n) = \frac{L}{1 + e^{-k(a(\ln(\mu_n) + b\text{SNR}_n + c))}}, \quad (10)$$

where the parameters $L = 0.6$, $k = 0.93$, $a = 0.6$, $b = 0.14$ and $c = 3.52$ are learned by the end of the regression process.

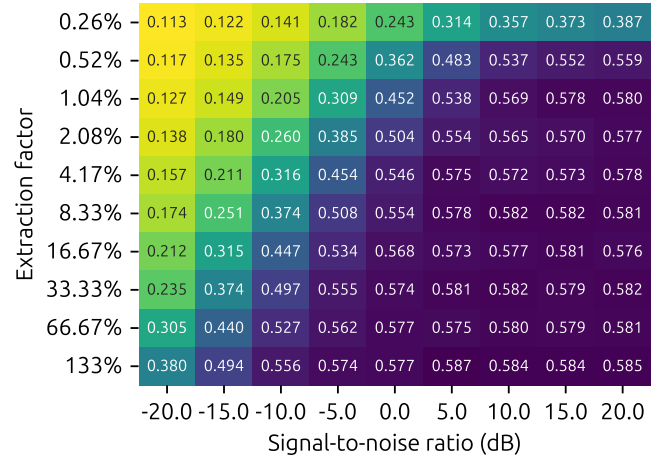


Fig. 2: Accuracy for different extraction factors and SNR.

F. Quality of experience

The performance of the resource allocation model is related to the total latency, energy consumption, and accuracy of the model for the UE n . We can define a normalized QoE function that fuses these three metrics as

$$\text{QoE}_n = \omega_t f(\lambda(t_n^l - t_n)) + \omega_e f(\beta(E_n^l - E_n)) + \omega_a f(\eta(\varepsilon_n - \varepsilon_n^l)), \quad (11)$$

where $f(x) = 1/(1+e^{-x})$ is the logistic function, $\{\omega_t, \omega_e, \omega_a\}$ are weights such as $\omega_t + \omega_e + \omega_a = 1$, t_n^l, E_n^l and ε_n^l are offset values. Finally, as mentioned before, the accuracy of the semantic-aware image classification task in function of the SNR, and extraction factor is defined as $\varepsilon_n = f(\mu_n, \text{SNR}_n)$.

G. Problem formulation

The scope of the problem is to maximize aggregate QoE by choosing $\{x, \mu, p, f\}$ as defined in equations (12)-(19).

$$\begin{aligned} & \underset{\{x, \mu, p, f\}}{\text{maximize}} && \sum_{n=1}^N \text{QoE}_n \end{aligned} \quad (12)$$

subject to:

$$\sum_{s=1}^S x_n^s \leq 1, \forall n \in \mathcal{N}, \quad (13)$$

$$\mu^{\min} \leq \mu_n \leq 1, \forall n \in \mathcal{N}, \quad (14)$$

$$p_n \leq p^{\max}, \forall n \in \mathcal{N}, \quad (15)$$

$$f_n \leq f^{\max}, \forall n \in \mathcal{N}, \quad (16)$$

$$t_n \leq t^{\max}, \forall n \in \mathcal{N}, \quad (17)$$

$$E_n \leq E_n^{\max}, \forall n \in \mathcal{N}, \quad (18)$$

$$\varepsilon_n \geq \varepsilon^{\min}. \quad (19)$$

The problem can be classified as a MINLP problem with integer and continuous decision variables and several nonlinear functions, making it challenging to solve it optimally.

III. HEURISTIC

We propose a greedy heuristic formalized in **Algorithm 1**, that maximizes the QoE through two sequential stages: offloading optimization and continuous variable optimization.

Algorithm 1 Greedy heuristic for QoE maximization

Require: Number of users $\mathcal{N}$, iteration limit $I_{\max}$
Ensure: Best QoE value and corresponding solution

- 1: $(x_n^s, p_n, f_n, \mu_n) \leftarrow (0, p^{\min}, f^{\min}, \mu^{\min}) \forall n \in \mathcal{N}, \forall s \in \mathcal{S}$
- 2: $best_qoe \leftarrow QoE(x, \mu, p, f)$
- 3: $best_sol \leftarrow (x, \mu, p, f)$
- # Offloading decision optimization**
- 4: **for all** $s \in \mathcal{S}$ **do**
- 5: **for** $n = 1$ to $\mathcal{N}$ **do**
- 6: $x_n^s \leftarrow 1$ and remove all other UE n associations
- 7: $qoe \leftarrow QoE(x, \mu, p, f)$
- 8: **if** $qoe \neq \text{None}$ and $qoe > best_qoe$ **then**
- 9: Update $best_qoe$ and $best_sol$
- 10: **else**
- 11: **break**
- # Continuous variables optimization**
- 12: **for all** $v \in \{\mu, p, f\}$ **do**
- 13: **for** $i = 1$ to $I_{\max}$ **do**
- 14: **for** $n = 1$ to $\mathcal{N}$ **do**
- 15: Randomly sample $v_n \sim U(v^{\min}, v^{\max})$
- 16: $qoe \leftarrow QoE(x, \mu, p, f)$
- 17: **if** $qoe \neq \text{None}$ and $qoe > best_qoe$ **then**
- 18: Update $best_qoe$ and $best_sol$
- 19: **else**
- 20: **break**
- 21: **return** $best_qoe, best_sol$

Initially, we initialize the offloading variables assuming local inference, while the transmission power, computational frequency, and semantic factor variables are initialized at their minimum admissible values. The corresponding QoE is then evaluated and stored as the current best solution.

In the first stage, the heuristic greedily optimizes the offloading decisions. For each server and user pair, the user is associated with a server and the QoE is re-evaluated. If the new solution is feasible and improves the current best QoE, the solution is accepted, otherwise the search branch is terminated early. In the second stage, the heuristic optimizes the continuous variables by randomly sampling candidate values within their feasible ranges for a maximum of $I_{\max}$ iterations. After each update, the QoE is evaluated and the new solution is accepted only if it improves the best QoE found so far. Otherwise, the current search process is interrupted.

The QoE function in lines 7 and 16 applies the definition given in equation (11). It is important to note that the remote servers in $\mathcal{S}$ are ordered in decreasing order of computational capability. This prioritizes computational latency over network latency. However, this can be easily modified by reordering $\mathcal{S}$ according to its inherent network latency.

IV. EVALUATION

In this section, we evaluate, through simulation-based experiments, the performance of the proposed semantic-aware task offloading framework under dynamic wireless conditions. We compare our approach with two alternative solutions: (i) semantic-unaware offloading, where users transmit raw data

without semantic compression; and (ii) local execution, where all tasks are processed on the device without offloading.

A. Implementation details

The system was implemented using a combination of PyTorch with the CIFAR-10 dataset for semantic encoding/decoding and scikit-learn for logistic curve fitting. The MINLP problem was modeled using IBM CPLEX 22.1.0.0. The channel model follows a quasi-static fading profile with AWGN. In order to calculate the computational demand of the encoder and decoder, we use the tool *ptflops* [15]. The list of parameter values considered in the experimentation is detailed in TABLE I. The values of NS and f_s represent the number of CUDA cores and the GPU clock frequency of the far edge, near edge, and cloud, respectively. Similarly, the value of t_s^{wired} represents the latency between the gNB and each of these remote servers. We consider an GTX 1630, an RTX 3090 Ti, an RTX 4090, and an RTX 5090 as the UE, the far edge, the near edge, and the cloud server, respectively. The complete code of this work is publicly available in a GitHub repository ¹.

TABLE I: Simulation Parameters

Parameter	Value	Parameter	Value
$ \mathcal{N} / \mathcal{S} $	4 / 3	B	100 MHz
$t^{\max}$	20 ms	$\varepsilon^{\min}$	50%
p_n	(10, 90) mW	μ_n	(0.1, 1)
NU	512	f_n	(1.740, 1.785) GHz
NS	10752/16384/21760	f_s	1.86/2.52/2.41 GHz
κ_n	1×10^{-26}	κ_s	1.2×10^{-26}
σ^2	2 mW	t_s^{wired}	0/501/791 μs

B. Experimental results

TABLE II shows the performance under different user preferences (w_t , w_e , and w_a). The no-preference mode provides a balanced trade-off among delay, energy, and accuracy. Delay and energy preferences minimize latency (1.02 ms) and energy consumption (5.39×10^{-29} J), respectively, yielding the same solution. The accuracy-preference mode achieves the highest accuracy (58.34%) with higher delay and energy consumption.

TABLE II: Characterization of the evaluated applications

Preference	Delay (ms)	Energy consumption (J)	Accuracy (%)
Equal	1.14	4.54×10^{-5}	56.84
Delay	1.02	5.39×10^{-29}	50
Energy	1.02	5.39×10^{-29}	50
Accuracy	2.65	2.36×10^{-4}	58.34

Fig. 3 shows the QoE performance under different bandwidth and noise conditions. As bandwidth increases, all methods improve, but semantic-aware offloading achieves higher QoE due to lower transmission delay and preserved inference accuracy. Under increasing noise power, its performance degrades more slowly than semantic-unaware offloading, since compressed semantic features require fewer transmitted bits and are more robust to channel impairments. Overall, semantic-aware offloading provides superior performance in both bandwidth-limited and noisy-channel scenarios.

¹<https://github.com/LABORA-INF-UFG/paper-LLK-2026>

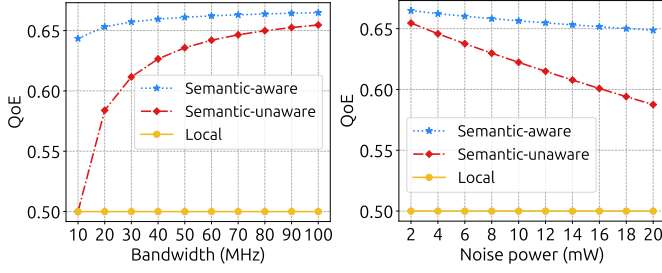


Fig. 3: QoE evaluation under bandwidth (left) and noise power (right) variations.

Fig. 4 evaluates the optimization performance and computational cost of the proposed framework. The left side compares the QoE achieved by the heuristic and MINLP approaches. The Optimality GAP measures the difference between the current and optimal MINLP solutions, while the Heuristic-MINLP represents the difference between the heuristic and MINLP results. The proposed heuristic achieves near-optimal QoE with significantly lower complexity. The right side shows the heuristic execution time for different numbers of users, with all instances limited to 2 hours.

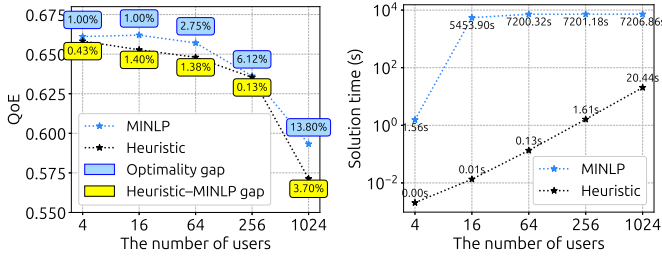


Fig. 4: Comparison of solution quality (left) and execution time (right) between the MINLP and heuristic approaches.

Fig. 5 shows the distribution of UEs among computing nodes for different user densities. As the number of users increases, the framework redistributes tasks across heterogeneous resources to balance the system load. Low-performance remote nodes initially process most tasks, while higher user densities require a more distributed allocation. Local execution is rarely selected due to its higher energy consumption and processing latency, being used mainly when remote servers become heavily loaded.

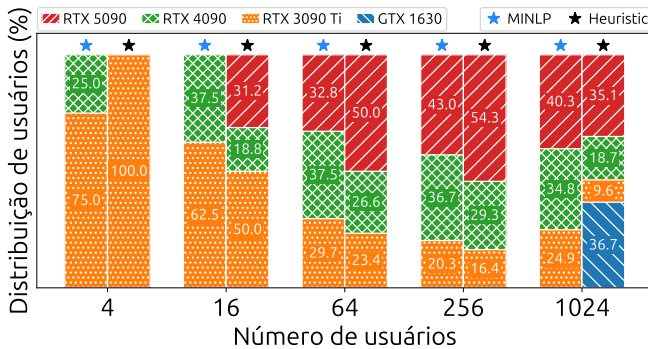


Fig. 5: UE distribution for different numbers of users.

V. CONCLUSION AND FUTURE WORK

In this work, we propose a semantic-aware task offloading framework that jointly integrates semantic feature extraction, wireless transmission, and computation scheduling into a QoE-driven model. We characterize the relationship between semantic extraction, wireless channel conditions, and inference accuracy, and incorporate this relationship into a multi-offloading framework that accounts for wireless transmission and wired network latency. We also develop a low-complexity heuristic to solve the resulting MINLP problem and evaluate its performance through simulation-based experiments. The results show that semantic-aware offloading consistently outperforms semantic-unaware and local execution approaches, especially under limited bandwidth and noisy channel conditions, while the heuristic reduces execution time. Future work may investigate multi-layer offloading and multi-modal inference scenarios.

REFERENCES

- [1] Y. Li, F. Zhou, L. Yuan, Q. Wu, and N. Al-Dhahir, "Cognitive semantic communication: A new communication paradigm for 6g," *IEEE Communications Magazine*, vol. 63, no. 6, pp. 122–129, 2025.
- [2] C. Zhang, L. Huang, and Q. Ning, "Resource allocation in wireless semantic communications: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2025.
- [3] Y. Zheng, T. Zhang, X. Mu, Y. Liu, and R. Huang, "Joint semantic transmission and resource allocation for intelligent computation task offloading in mec systems," *IEEE Transactions on Wireless Communications*, vol. 24, no. 10, pp. 8756–8770, 2025.
- [4] Y. Zheng, T. Zhang, R. Huang, and Y. Wang, "Computing offloading and semantic compression for intelligent computing tasks in mec systems," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, 2023, pp. 1–6.
- [5] Y. Cang, M. Chen, Z. Yang, Y. Hu, Y. Wang, C. Huang, and Z. Zhang, "Online resource allocation for semantic-aware edge computing systems," *IEEE Internet of Things Journal*, vol. 11, no. 17, pp. 28 094–28 110, 2024.
- [6] Z. Ji and Z. Qin, "Computational offloading in semantic-aware cloud-edge-end collaborative networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 18, no. 7, pp. 1235–1248, 2024.
- [7] Z. Ji, Z. Qin, X. Tao, and Z. Han, "Resource optimization for semantic-aware networks with task offloading," *IEEE Transactions on Wireless Communications*, vol. 23, no. 9, pp. 12 284–12 296, 2024.
- [8] G. Zhang, Q. Hu, Z. Qin, Y. Cai, G. Yu, and X. Tao, "A unified multi-task semantic communication system for multimodal data," *IEEE Transactions on Communications*, vol. 72, no. 7, pp. 4101–4116, 2024.
- [9] C. Wang, R. Olsson, S. Forsström, and Q. He, "Deep semantic inference over the air: An efficient task-oriented communication system," 2025. [Online]. Available: <https://arxiv.org/abs/2508.12748>
- [10] H. Zhang, S. Shao, M. Tao, X. Bi, and K. B. Letaief, "Deep learning-enabled semantic communication systems with task-unaware transmitter and dynamic data," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 170–185, 2023.
- [11] X. Chen, D. Feng, W. Jiang, Q. Luo, G. Chen, and Y. Sun, "Qoe-driven multi-task offloading for semantic-aware edge computing systems," *IEEE Transactions on Network Science and Engineering*, vol. 13, pp. 7100–7117, 2026.
- [12] 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.
- [13] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, M. Tomizuka, J. Gonzalez, K. Keutzer, and P. Vajda, "Visual transformers: Token-based image representation and processing for computer vision," 2020. [Online]. Available: <https://arxiv.org/abs/2006.03677>
- [14] E. Boursoulatz, D. Burth Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 567–579, 2019.
- [15] V. Sovrasov, "Ptflops: A flops counting tool for neural networks in pytorch framework," *GitHub repository*, 2018.