LLM-Based Intent Processing and Network Optimization Using Attention-Based Hierarchical Reinforcement Learning

Md Arafat Habib[1], Pedro Enrique Iturria Rivera[1], Yigit Ozcan[2], Medhat Elsayed[2],
Majid Bavand[2], Raimundus Gaigalas[3] and Melike Erol-Kantarci[1], *Senior Member, IEEE*
[1]*School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada*
[2]*Ericsson Inc., Ottawa, Canada* [3]*Ericsson AB, Stockholm, Sweden*
Emails:{mhabi050, pitur008, melike.erolkantarci}@uottawa.ca,
{yigit.ozcan, medhat.elsayed, majid.bavand, raimundas.gaigalas }@ericsson.com

Abstract—**Intent-based network automation is a promising tool that enables easier network management; however, certain challenges must be addressed effectively. These are: 1) processing intents, i.e., identification of logic and necessary parameters to fulfill an intent, 2) validating an intent to align it with current network status, and 3) satisfying intents via network optimizing applications. This paper addresses these points via a three-fold strategy to introduce intent-based automation for modern 5G architectures. First, intents are processed via a lightweight Large Language Model (LLM). Secondly, once an intent is processed, it is validated against future incoming traffic volume profiles (high or low). Finally, a series of network optimization applications has been developed. With their machine learning-based functionalities, they can improve certain key performance indicators such as throughput, delay, and energy efficiency. In the final stage, using an attention-based hierarchical reinforcement learning algorithm, these applications are optimally initiated to satisfy the intent of an operator. Our simulations show that the proposed method can achieve at least a 12% increase in throughput, a 17.1% increase in energy efficiency, and a 26.5% decrease in network delay compared to the baseline algorithms.**

Index Terms—**Attention-based hierarchical reinforcement learning, intent-based network automation, network optimization**

I. INTRODUCTION

An intent can be defined as the expectations, including requirements, goals, and constraints for a specific service or network management workflow [1]. It is typically understandable by humans and must be interpreted by machines without ambiguity. Furthermore, it emphasizes "What" needs to be achieved, rather than focusing on "How" to achieve the outcomes, specifying the desired metrics without detailing the process to attain them [1]. It is anticipated that Mobile Network Operators (MNOs) will leverage advancements in natural language processing to express intents in plain language for automated network management. In modern 5G networks, achieving such intent-based automation can be challenging especially because intents can vary widely based on network conditions, customer demands, and industry-specific terminologies [1]. Therefore, a contextual understanding of the intents is vital for accurate intent processing. Large Language Models (LLMs) with its transformer architecture excel at capturing such context and understanding the underlying meaning of sentences (intents) within a broader spectrum [2].

Processing intents expressed in natural language by the human network operator is important to derive policies for network performance optimization. However, allowing any kind of intent to put an impact on the network configurations can lead to serious degradation in performance. For example, an intent: "Increase energy efficiency by 30%" during peak hours can degrade performance due to high incoming traffic volume. This kind of scenario requires a sophisticated method of intent validation using contemporary network status to ensure a seamless user experience without compromising any QoS.

The concept of an Open Radio Access Network (Open RAN) facilitates the idea of disaggregating the traditional RAN and promoting interoperability between components from different vendors. This concept enables the deployment of network-optimizing applications (apps) in intelligent RAN controllers, which can be triggered by processed intents. In [3], intents are transformed into goals for a Hierarchical Reinforcement Learning (HRL) algorithm to initiate such applications for intent fulfillment. However, the computational intensity of selecting optimal combinations from numerous applications becomes significant for an AI agent.

To this end, we propose a three-fold strategy to perform intent processing, validation, and application initiation based on intents. We begin by processing intents using a pre-trained LLM named Bidirectional Encoder Representations from Transformers (BERT) [4]. In the next step, intent validation is performed using a Transformer-based time series predictor for upcoming traffic volume [5]. This ensures that the intended adjustments to the network are not only theoretically sound but also practical and feasible given expected future traffic patterns. Lastly, to overcome the computational burden of selecting the appropriate combination of applications from a high number of possible choices, we utilize an attention-based HRL framework that takes the processed intents (magnitude of the intended performance metric by an operator) as goals. By using an attention mechanism to filter only the feasible options [6], the system can significantly reduce the computational

burden.

The simulation results demonstrate that the proposed scheme yields significant improvements compared to two baseline approaches: HRL without attention and intent validation, and a single-application scenario developed using Deep RL (DRL). The proposed approach surpasses the HRL baseline in terms of throughput, delay, and energy efficiency by 12.02%, 26.5%, and 17.1% respectively. Moreover, compared to the DRL baseline, the proposed method achieves superior results across the same performance metrics, with improvements of 17.2%, 48.6%, and 39.3% respectively.

II. RELATED WORK

Even though using LLMs to process intents for network optimization is a relatively new concept, a few works have been found in the literature. Wang et al.'s development of a transformer model based on an LLM is one such example, where multi-modal representation learning is employed to integrate various forms of network data into a unified feature space [7]. Similarly, Kristina et al. have introduced a pipeline that utilizes an LLM to process intents into structured, policy-based abstractions, linking them to APIs for execution [2].

To manage network applications based on operator intents, an approach based on HRL is proposed in [3] where intents are processed into goals for the control algorithm to optimize performance. Polese et al. propose a framework to collect control requests (intents) and select the optimal Machine Learning (ML) models to achieve the operator's goals to avoid conflicts [8].

Compared with existing literature, the main contribution of this work lies in deploying an LLM to interpret a human network operator's intent and extract specific parameters for further network performance optimization. Processed intents are validated against future wireless traffic conditions. Intents lead HRL agents to initiate and orchestrate multiple network applications with an efficient exploration of action space via an attention mechanism.

III. SYSTEM MODEL

A cellular system employing downlink Orthogonal Frequency Division Multiplexing (OFDM) is considered in this work. There are B base stations that concurrently serve U users. Within the macro cell's coverage area, multiple small cells are deployed. The system supports K classes of traffic, and users are connected through dual connectivity across multiple Radio Access Technologies (RATs). These RAT classes can facilitate various technologies like LTE, and 5G.

The system model, shown in Fig. 1, proposes two distinct controllers responsible for managing RAN functionalities: the strategic controller and the tactical controller. The strategic controller on top is responsible for long-term goals, such as network-wide performance optimizations, and policy formulation. It can define the network's broader objectives, including service quality guarantees, energy efficiency, or overall traffic management for a timespan. The tactical controller on the
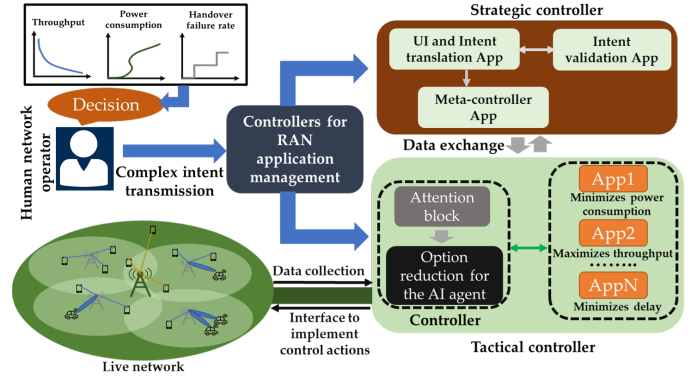

Fig. 1. Three-step methodology for intent processing, validation, and performance optimization.

bottom focuses on real-time decisions, such as handling immediate network changes, performing handovers, or dynamically adjusting resources based on the current network state. It operates within the framework set by the strategic controller but is more focused on near-real-time adjustments to ensure smooth network operation from moment to moment.

The controllers host applications focusing on control and optimization tasks across two different time scales. We deploy BSs capable of switching bands from 5G mid band to high band frequencies. This deployment enables us to cater to high-throughput traffic by employing intelligent beamforming techniques. An application can be developed to regulate power based on UE location, utilizing minimal transmission power for enhanced energy efficiency. The system facilitates analog beamforming, with each BS employing a uniform linear array of μ antennas [9]. Beamforming weights for each beamforming vector are implemented using constant modulus phase shifters. Additionally, we assume the presence of a beam steering-based codebook F, from which beamforming vectors are selected [9]. Each BS b operates with a transmit power $P_{TX,b} \in P$, where P represents the set of candidate transmit powers. The energy consumption model for the BS is obtained from [10].

A. Problem Definition

Based on the system and network model discussed so far, we introduce multiple applications. There are two sets of them. The first set consists of intent translation, validation, and meta-controller (associated with our RL-based control algorithm). The second set consists of traffic steering, beamforming, cell sleeping, power allocation, and handover management applications that can be directly controlled by the first set. All applications from the first set, along with cell sleeping, operate on a time frame longer than 1 second within the strategic controller, while the others preferably operate on a time scale between 10 ms and 1s within the tactical controller.

Let us define Ψ as the collection of applications, and their various combinations. Let σ represent the subset of Ψ which contains at least one element that is capable of enhancing network performance in response to operator intents. We also define ρ as the set of potential KPIs that an application might

improve, and Q_s as the set of QoS requirements that must be met by the system. Given these definitions, the challenge of orchestrating applications based on operator intent can be formulated as follows:

$$P_1 : \max \sum_{\rho \in \varrho} \sum_{q_s \in Q_s} (C_\rho - \varpi \gamma_{q_s}), \tag{1}$$
$$\text{s.t.} \quad \forall (\Psi) \exists (\sigma) : \zeta(O) = 1,$$

where C is the magnitude of the intended performance metric the operator intends to improve. ϖ is the penalty parameter for QoS requirement violation, and γ_{q_s} represents the number of UEs whose QoS requirements have been violated. The evaluation is binary: a UE is counted only if its QoS standards are not met. Lastly, $\zeta(O)$ is the proposition that "An application can improve a performance metric", which is either '0' or '1'.

The second problem we want to address in this work is associated with intent validation. QoS parameters deviate from the originally defined QoS metrics over time due to performance degradation in the wireless networks, causing QoS drifts. For a specific traffic class k ($K = \{k_1, k_2, \ldots, k_\eta\}$), with its set of QoS requirements Q_s ($Q_s = \{Q_{s_1}, Q_{s_2}, \ldots, Q_{s_j}\}$) and corresponding performance metrics C, the QoS drift can be formulated as follows:

$$P_2 : \min \sum_{q_s \in Q_s} \sum_{c \in C} \{D_{QoS} - A_{QoS}\}, \tag{2}$$
$$\text{s.t.} \quad A_{QoS} \leq D_{QoS} \quad \text{and} \quad A_{QoS} > 0.$$

In (2), A_{QoS} represents the achieved value of a metric (e.g. delay, throughput) for a certain traffic type at time t, and D_{QoS} represents the pre-defined value of the same parameter which has to be maintained for optimal performance. The objective here is to validate intents that will not result in QoS drifts. For cases where $A_{QoS} > D_{QoS}$, we consider there is no QoS drift and the intent is valid. Otherwise, we calculate the QoS drift, $I_Q = D_{QoS} - A_{QoS}$ based on (2).

Another objective of this work is to reduce the large action space, which consists of numerous potential combinations of applications. To solve this problem, we define an attention function $\alpha(s_t, i, a)$ that scores each action $a \in A$ based on the current state s_t and the operator's intent i.

IV. PROPOSED METHODOLOGY

In this section, first, we discuss processing intents using an LLM and then predictive intent validation. The last part of this section consists of an HRL-based attention mechanism for application initiation and orchestration.

A. Intent Processing using LLM

Intent-based networking simplifies network and administrative tasks by automating system operations through the comprehension and execution of operator-defined intents, such as "Increase overall energy efficiency by 10%", "Boost system throughput by 15%", or "Reduce network delay by 13%". These intents focus on improving metrics like throughput, energy efficiency, and network delay. Our approach uses few-shot learning to classify intents and extract relevant keywords.

The methodology involves constructing a prompt that includes a few examples of intents with their corresponding types and keywords. This prompt is then used to query the LLM, which classifies new intents and extracts the required information based on the provided examples. Before processing new intents, a set of examples is prepared. Each example consists of:

- **Intent:** A natural language statement describing the intent.
- **Type:** The category or classification of the intent.
- **Keywords:** Key phrases or terms extracted from the intent.

A prompt is constructed by sequentially adding the prepared examples. Each example is formatted to include the intent ($I = \{i_1, i_2, \ldots, i_n\}$), type ($I_{t_y} = \{t_{y_1}, t_{y_2}, \ldots, t_{y_m}\}$), and keywords ($\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_w\}$). After adding all the examples, the new intent is appended to the prompt (p), followed by a request for classification and keyword extraction. The constructed prompt is sent to the pre-trained LLM (a lightweight version of BERT) [11]. The LLM uses the context provided by the examples to understand the task and generate a response that includes the classification and keywords for the new intent. The response from the LLM is parsed to extract the classified type of the new intent and the extracted keywords. Each new intent's classified type and keywords are compiled into a result set.

This intent processing functionality is presented as a User Interface (UI) and intent translation application in Fig. 1. The job of this application is to work as a UI for a human operator and process the intents with crucial information to feed the intent validation and the meta-controller application. We further elaborate our intent processing methodology via Algorithm 1. The approach can be categorized as prompt engineering, where the prompt serves as a template for the LLM to infer the correct response based on minimal data examples.

B. Predictive Intent Validation Using Transformer

To perform intent validation, we utilize one of our previous works [5] where traffic data are collected so that it can be fed to the transformer-based time series predictor for future traffic volume prediction. The traffic predictor resides in the strategic controller (named as intent validation app, as in Fig 1.). The forecast for traffic volume is performed for successive time intervals, with two distinct thresholds derived from historical data. The higher threshold, denoted as Th_p, is decided based on previous instances of heavy traffic. If the forecast traffic surpasses Th_p, applications like traffic steering and power allocation can be activated to enhance or sustain high throughput, depending on the intent of a human operator. The lower threshold, Th_t, is established based on low traffic volume. In the event that the anticipated traffic decreases to less than this threshold, an intent to increase energy efficiency can be validated. It will then lead to the initiation of energy-saving applications like cell sleeping due to the expected lower traffic levels. The algorithmic process of predictive traffic validation is presented in Algorithm 2.

C. Network Optimization Using Attention-based HRL

After intent validation, we optimize the network performance based on the operator intent using an HRL algorithm. We

Algorithm 1 Intent Processing Using LLM

Require: $E = \{(i_1, t_{y_1}, \lambda_1), (i_2, t_{y_2}, \lambda_2), ...\}, \Xi = \{(i, t_y, \lambda)\}$

1: **procedure** CLASSIFYANDEXTRACT(I, E, LLM)
2:     $\Xi \leftarrow \emptyset$
3:     **for all** $i \in I$ **do**
4:         $p \leftarrow$ CREATEPROMPT(i, E)
5:         $response \leftarrow$ QUERYLLM(p, LLM)
6:         $(t_y, \lambda) \leftarrow$ PARSERESPONSE($response$)
7:         $\Xi \leftarrow \Xi \cup \{(i, t_y, \lambda)\}$
8:     **end for**
9:     **return** Ξ
10: **end procedure**
11: **procedure** CREATEPROMPT(i, E)
12:     $p \leftarrow$ " "
13:     **for all** $e \in E$ **do**
14:         $p \leftarrow p +$ "**Example:**"
15:         $p \leftarrow p +$ "**Intent:** " $+ e[0]$
16:         $p \leftarrow p +$ "**Type:** " $+ e[1]$
17:         $p \leftarrow p +$ "**Keywords:** " $+ e[2]$
18:     **end for**
19:     $p \leftarrow p +$ "**New Intent:** " $+ i$
20:     $p \leftarrow p +$ "**Type, Keywords**"
21:     **return** p
22: **end procedure**
23: **procedure** QUERYLLM(p, LLM)
24:     **return** LLM.QUERY(p)
25: **end procedure**
26: **procedure** PARSERESPONSE($response$)
27:     $type \leftarrow$ EXTRACTTYPE($response$)
28:     $keywords \leftarrow$ EXTRACTKEYWORDS($response$)
29:     **return** $(type, keywords)$
30: **end procedure**

Algorithm 2 Intent Validation Based on Traffic Prediction

1: Predict traffic T_p for $t+1$ and extract i using Algorithm 1
2: **if** $(T_p > Th_p \,||\, T_p < Th_t)$ **then**
3:     Check QoS profiles for the relevant traffic class k
4:     **if** i impacts Q_s **then**
5:         **for** each QoS parameter Q_{s_j} in k_η's profile **do**
6:             Calculate QoS drift $I_Q(Q_{s_j})$ using eq. 2
7:             **if** $I_Q(Q_{s_j}) \neq 0$ **then**
8:                 Intent invalid
9:                 **break**
10:             **end if**
11:         **end for**
12:         **if** I_Q for elements in Q_s is 0 $||\, A_{QoS} > D_{QoS}$ **then**
13:             Validate intent
14:         **end if**
15:     **end if**
16: **else**
17:     Re-calculate Th_p and Th_t
18: **end if**
19: **return** Validation results

---

opt for a hierarchical Deep-Q-Network (h-DQN) having a two-level hierarchy with a meta-controller on top [12] and a controller on the bottom. These two controllers are solely related to the h-DQN algorithm that we are proposing to use to manage network-optimizing applications. Meta-controller and the controller are hosted in the strategic and tactical controller, respectively. We employ an attention mechanism [6] to reduce the action space for our h-DQN agent, enhancing efficiency and decision-making while expediting convergence. We can define the attention function as follows:

$$\alpha(s_t, i, a) = f_\theta(s_t, i, a), \qquad (3)$$

where f_θ computes the attention score trained using supervised learning, to predict which actions are most likely to fulfill the operator's intent. Once the attention scores $\alpha(s_t, i, a)$ are computed, the action space is reduced to a subset, A_s^{att} containing only the most relevant actions.

$$A_s^{att} = \{a \in A \mid \alpha(s_t, i, a) > \epsilon\}, \qquad (4)$$

where ϵ is a threshold that filters out actions with low relevance. This subset A_s^{att} is much smaller than the full action space A, allowing the agent to make decisions more efficiently. For each Markovian option ω [6], the attention mechanism selects only the relevant options based on the operator's intent. Each option is defined by a policy π_ω, a termination condition β_ω, and an initiation set I_s. After reducing the action space, the agent applies its policy π to select the best action from A_s^{att}:

$$\pi(s_t) = \arg \max_{a \in A_s^{att}} Q(s_t, a), \qquad (5)$$

where $Q(s_t, a)$ is the action-value function that represents the expected return from taking action a in state s_t.

Meta-controller in h-DQN can take in a network state (e.g., traffic class) and a goal (desired change of a performance metric extracted from an intent) to achieve. The controller in the lower level takes the action of choosing an application or combinations of them based on the state and the goal. The meta-controller is designed as a top-level application in the strategic controller (see Fig. 1). Applications are controlled by the meta-controller that can directly optimize system performance. At this stage of the work, we define the following MDP for the h-DQN to solve the problem formulated in (1).

- **State and Action:** The set of states consists of traffic flow types of different users in the network. $S = \{T_1, .., T_2, ..., T_3, ..., T_4, ..\}$. Both meta-controller and controller share the same states. Selecting network optimizing applications, or combinations of them is considered as actions to be performed by the controller which is defined as $\{A = A_{App1}, A_{App1,2}, ..., A_{AppN}\}$.
- **Intrinsic reward:** The intrinsic reward function (r_{in}) for the controller is: $r_{in} = C_\rho - \varpi\gamma_{q_s}$ (see (1) for more details.).
- **Goal for the controller:** Increased level of a performance metric that can satisfy operator intent is passed to the controller as goals. For example, $G = \{TP_1, TP_2, ..., TP_n\}$ for throughput increasing intents.

- **Extrinsic reward:** Summation of the intrinsic reward over τ steps.

We have used five applications in total that are controlled by the meta-controller. They are traffic steering [13], cell sleeping [3], beamforming [3], power allocation [14], and energy efficient handover management [15]. All of these applications are based on DRL. The traffic steering application is engineered to concurrently uphold QoS across various traffic types by employing a steering mechanism to ensure optimal performance in terms of delay and throughput. The cell sleeping app (works on a longer time scale in the strategic controller) aims to reduce network power consumption by deactivating idle or less busy BSs based on traffic load and queue length. The beamforming application uses UE coordinates to sort out the steering angle, and the array steering vector corresponding to the respective element in the codebook. The power allocation application tries to maximize the total throughput and lastly, the handover management app is specifically tailored with DRL to achieve energy efficiency via optimally tailored handover policies. The process of selecting and orchestrating apps for network optimization using HRL can be summarized as follows:

- **Step 1:** An input is provided by the human network operator associated with a performance metric.
- **Step 2:** Intent is processed and crucial information like intent type: "energy-efficiency/ throughput/ Delay", and magnitude of change (in percentage) are extracted using pre-trained LLM.
- **Step 3:** Extracted information is passed to the intent validation application.
- **Step 4:** With a valid intent, a target associated with the performance metrics (goals) and feasible action set for the intent are provided to the controller by the meta-controller.
- **Step 5:** The controller selects an application or a combination of them to reach the target performance as close as possible.
- **Step 6:** Selected applications optimize the performance of the network as a response to the intent of the operator.

V. PERFORMANCE EVALUATION

In this section, first, we introduce the simulation settings followed by the simulation results showing the effectiveness and superiority of the proposed method.

A. Simulation setup

The simulation setup for this study features a macro cell surrounded by densely deployed small cells in a multi-RAT (multi-radio access technology) environment, accommodating a total of 60 users. For the 5G NR configuration, a bandwidth of 60 MHz is used, with carrier frequencies of 3.5 GHz and 30 GHz, a subcarrier spacing of 15 KHz, and a maximum transmission power of 43 dBm, based on the 3GPP urban micro channel model [16]. The LTE setup, on the other hand, employs a 40 MHz bandwidth, an 800 MHz carrier frequency, a 15 KHz subcarrier spacing, and a maximum transmission power of 38 dBm, following the 3GPP urban macro channel model [16].
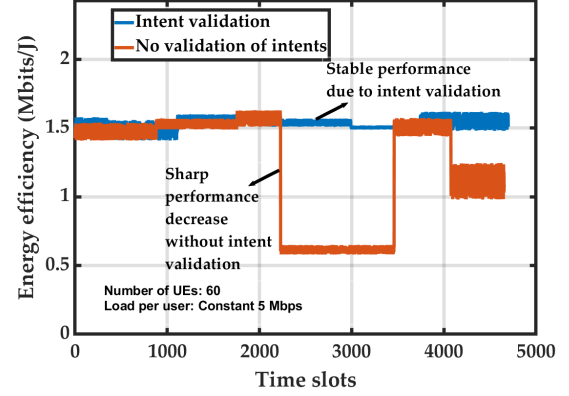

Fig. 2. Impact of intent validation on energy efficiency.
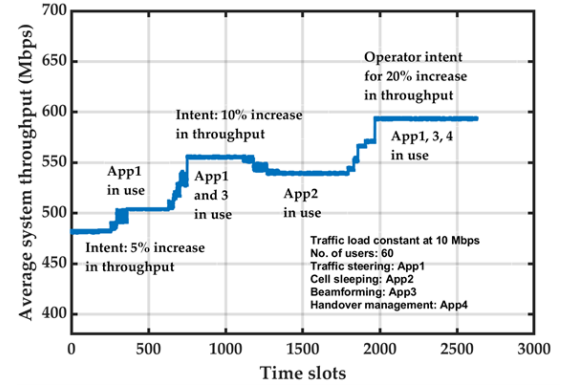

Fig. 3. Impacts of operator intents on throughput.

There are four traffic types: video, gaming, voice, and Ultra Reliable Low Latency Communications (URLLC), each characterized by specific inter-arrival times for packets: 12.5 ms, 40 ms, 20 ms, and 0.5 ms, respectively. These packet inter-arrival times follow different distributions—Pareto for video, Uniform for gaming, and Poisson for both voice and URLLC. In the following section, we present results using shorthand identifiers for applications: App1 (traffic steering), App2 (cell sleeping), App3 (beamforming), App4 (power allocation), and App5 (handover management).

Table I provides a summary of the tunable parameters for both the Autoformer (used for traffic prediction) and the RL methods (including the HRL control algorithm and DRL-based network applications).

TABLE I
PARAMETER SETTINGS FOR AUTOFORMER AND RL

Autoformer	
Learning rate (α), Batch size	10^{-4}, 32 [5]
Model architecture	2 encoder layers, 1 decoder layer
HRL parameters (h-DQN)	
Exploration Probability (ϵ_2)	1, annealed to 0.1 [12]
Learning rate (α), discount factor γ	0.00025, 0.99 [12]
RL paramters (applications)	
Batch size, Initial exploring steps	32, 3000
Learning rate (α), discount factor γ	0.5, 0.9

B. Simulation Results

The first simulation result is associated with intent validation. When there is no intent validation, conflicting intents from an operator can severely degrade performance. Fig. 2 presents such an example where we have a traffic load of 5Mbps. At 2200th time slot, there is a sharp decrease in energy efficiency. This is because of the unwanted intent of increasing throughput even though the cell sleeping application is active due to low traffic demand. An intent of increasing throughput would result in activating more cells unnecessarily and decrease overall energy efficiency. Similar impacts can be observed for other KPIs like delay, and throughput also.

As shown in Fig. 3, the human operator's goal to "increase throughput" leads to multiple application selection. For instance, aiming for a 5% throughput increase results in a noticeable surge after several time slots due to App1. Aiming for a 10% increase involves both App1 and App3. Conversely, reducing power consumption significantly decreases throughput around the 1250th time slot, reflecting the deactivation of App1 and App3 and the activation of App2. When the operator aims for a 20% increase, App1, App3, and App4 together produce a sharp throughput rise.

The proposed method's effectiveness is compared with two baselines: an HRL-based application initiation without an attention mechanism or intent validation, and a single DRL-based application scenario. The proposed method employs a hard attention mechanism, focusing on the subset of applications most relevant for the current traffic type and operator's intent. This focus reduces conflicting actions and shows superior performance over the HRL baseline (improvements of 12.02% in throughput, 26.5% in delay, and 17.1% in energy efficiency) and the DRL baseline (improvements of 17.25%, 48.6%, and 39.3% in the same metrics), as observed in Fig. 4.

VI. Conclusions

In this paper, first, we process intents using an LLM and extract crucial information regarding which performance metric to optimize and by how much. Next, a Transformer-based intent validation technique has been used to rule out intents that conflict with the contemporary network state to avoid performance degradation. Lastly, we use an attention-based HRL to initiate and orchestrate applications for performance optimization. The proposed method outperforms the HRL baseline in throughput, delay, and energy efficiency by 12.02%, 26.5%, and 17.1% respectively. Additionally, compared to the DRL baseline, it achieves superior results across the same performance metrics, with enhancements of 17.25%, 48.6%, and 39.3% respectively. In the future, we plan to perform highly complex intent translations from the human network operator.
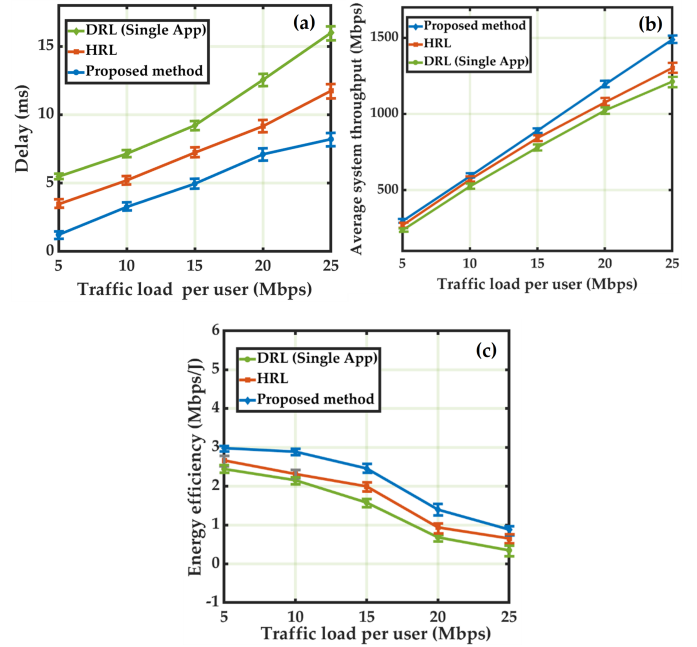
Acknowledgement

Fig. 4. Performance analysis of the proposed method: (a) network delay, (b) throughput, and (c) energy efficiency.

References

[1] 3GPP, "Management and Orchestration; intent Driven Management Services for Mobile Networks," ETSI, Technical Specification TS 28.312, March 2022, release 17. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/128300_128399/128312/17.00.01_60/ts_128312v170001p.pdf

[2] K. Dzeparoska, J. Lin, A. Tizghadam, and A. Leon-Garcia, "LLM-Based Policy Generation for Intent-Based Management of Applications," in *2023 19th International Conference on Network and Service Management (CNSM)*, 2023, pp. 1–7.

[3] M. A. Habib, H. Zhou, P. E. Iturria-Rivera, M. Elsayed, M. Bavand, R. Gaigalas, Y. Ozcan, and M. Erol-Kantarci, "Intent-driven Intelligent Control and Orchestration in O-RAN Via Hierarchical Reinforcement Learning," in *2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems*, 2023, pp. 55–61.

[4] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *CoRR*, vol. abs/1810.04805, 2018.

[5] M. A. Habib, P. E. Iturria-Rivera, Y. Ozcan, M. Elsayed, M. Bavand, R. Gaigalas, and M. Erol-Kantarci, "Transformer-Based Wireless Traffic Prediction and Network Optimization in O-RAN," *arXiv preprint arXiv:2403.10808*, Mar. 2024.

[6] A. C. Nica, K. Khetarpal, and D. Precup, "The Paradox of Choice: Using Attention in Hierarchical Reinforcement Learning," *CoRR*, vol. abs/2201.09653, 2022.

[7] J. Wang, L. Zhang, Y. Yang, Z. Zhuang, Q. Qi, H. Sun, L. Lu, J. Feng, and J. Liao, "Network Meets ChatGPT: Intent Autonomous Management, Control and Operation," *Journal of Communications and Information Networks*, vol. 8, no. 3, pp. 239–255, 2023.

[8] S. D'Oro, L. Bonati, M. Polese, and T. Melodia, "OrchestRAN: Network Automation Through Orchestrated Intelligence in the Open RAN," in *IEEE INFOCOM 2022*, 2022, pp. 270–279.

[9] F. B. Mismar, B. L. Evans, and A. Alkhateeb, "Deep Reinforcement Learning for 5G Networks: Joint Beamforming, Power Control, and Interference Coordination," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1581–1592, 2020.

[10] P. Ren and M. Tao, "A Decentralized Sleep Mechanism in Heterogeneous Cellular Networks with QoS Constraints," *IEEE Wireless Communications Letters*, vol. 3, no. 5, pp. 509–512, Oct. 2014.

[11] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," *CoRR*, vol. abs/1909.11942, 2019.

[12] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. B. Tenenbaum, "Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation," *CoRR*, vol. abs/1604.06057, 2016.

[13] M. A. Habib, H. Zhou, P. E. Iturria-Rivera, M. Elsayed, M. Bavand, R. Gaigalas, S. Furr, and M. Erol-Kantarci, "Traffic Steering for 5G Multi-RAT Deployments using Deep Reinforcement Learning," in *IEEE 20th CCNC*, 2023, pp. 164–169.

[14] H. Zhang, H. Zhou, and M. Erol-Kantarci, "Team Learning-Based Resource Allocation for Open Radio Access Network (O-RAN)," in *ICC 2022*, 2022, pp. 4938–4943.

[15] Y. Song, S. H. Lim, and S.-W. Jeon, "Handover Decision Making for Dense HetNets: A Reinforcement Learning Approach," *IEEE Access*, vol. 11, pp. 24 737–24 751, 2023.

[16] E. Dahlman, S. Parkvall, and J. Skold, *4G, LTE-Advanced Pro and The Road to 5G, Third Edition*, 3rd ed. USA: Academic Press, Inc., 2016.