HERO: Holistic Envisioned Reinforcement Learning Multi-Domain Orchestration with Latent ODE

Parsa Rajabzadeh, Abdelkader Outtagarts, Yassine Hadjadj-Aoul, Soraya Ait-Chellouche

HERO: Holistic Envisioned Reinforcement Learning Multi-Domain Orchestration with Latent ODE

Parsa Rajabzadeh, Abdelkader Outtagarts
Bell Labs
Nokia Networks France,
Paris, France
parsa.rajabzadeh@nokia.com, Abdelkader.Outtagarts@nokia.com

Yassine Hadjadj-Aoul, Soraya Ait-Chellouche
INRIA
Univ Rennes, CNRS, IRISA,
Rennes, France
yassine.hadjadj-aoul@irisa.fr, soraya.ait-chellouche@irisa.fr

***Abstract*—6G promises E2E cross domains continuous intelligence to optimize resource management and orchestration. However, state-of-the-art methods fall short in providing promised reliable and optimal resource management due to their inefficient proactive decision-making and planning capabilities. This paper proposes a novel Holistic Predictive Framework designed to enhance decision-making and achieve proactive multi-domain resource management. Our framework comprises of predictive, focus, and decision making elements, enabling exceptional proactive planning, and decisions-making based on a holistic vision of network's future. To select the best predictive and decision-making elements, Various combinations of predictive Machine Learning and Reinforcement Learning algorithms were examined in our testbed. To demonstrate the superiority of our framework, we have conducted another test where our framework was compared with state-of-the-art solutions. The test results indicate that coupling the predictive element and attention-augmented decision making unit significantly improves the orchestrator's performance. Based on the result of both tests, our multi-domain orchestration solution, which exploits Latent ODE, outperforms all Cutting-Edge frameworks and is the best combination of the algorithms for our framework.**

***Index Terms*—6G, Multi-Domain Orchestration, Actor-Critic, Latent ODE, Federated Learning, Reinforcement Learning**

I. Introduction

Given the rapid deployment of 5G networks, research and development on 6G are already underway, paving the way for new era of connectivity and technological advancements [1], [2]. Indeed, 6G, expected to be launched around 2030, envision various and ambitious use cases that comes with stringent requirements on latency, reliability, density, security, energy efficiency, etc. To meet 6G promises, AI and ML are identified to play a significant role. Indeed, envisioned 6G aims to deeply incorporate Artificial Intelligence(AI) into the Mobile Networks, transforming them into general purpose platforms capable of offering Smart Connectivity to a wide variety of extremely diverse and heterogeneous domains. This smart connectivity ,or in another term connected intelligence, intends to revolutionize the network resource management by enabling self-optimize and autonomous AI-enabled orchestrators that are interconnected together to provide E2E Network Intelligence (NI) across multiple administrative domains [3], [4].

Achieving efficient cross domains E2E automation of network orchestration necessitates proactive AI-enabled orchestrators to coordinate with each other to provide heterogeneous cloud environments that will support various services and applications on the same physical infrastructure [5]. Multi-domain coordination would enable massive number of slices with diverse requirements to operate simultaneously across domains. This will bring major challenges on the traditional centralized approaches that are Incapable of offering a global view of multi-domain network for orchestrators. Figure 1 displays an E2E multi-domain mobile network. These challenges demand a novel decentralized architecture that would allow harmonized and smart orchestrators ,acting in heterogeneous network domains, to anticipate the next state of their domain with respect to their holistic vision of the entire network and to act proactively to their environments. Predicting ahead and planning would be impossible for these smart orchestrators if they are unable to exchange coded information related to their domain while preserving their privacy.

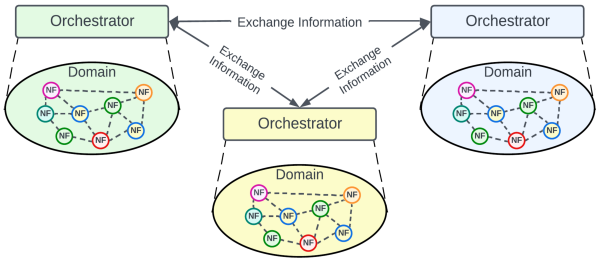

Fig. 1. E2E Mutli-domain Mobile Network resource management.

Many works such as [6], [11] suggested using Reinforcement Learning (RL) methods for resource management within a domain. Although these RL methods could adapt to the conditions of the environment and dynamically learn to make optimal decisions, they often adopt sub-optimal strategies due to their inability to fully anticipate future states, particularly in unobservable or partially observable domains. Moreover, usually these solutions are proposed in the context of managing only one domain and they often neglect to consider effects of domains on each other where consecutive traffic loads with changing magnitudes cross multiple domains. In that case, a decision made by one orchestrator in a domain can impact the subsequent state of other domains. As long as these inter-domain influences are not considered and the

lack of prediction ability is not addressed, E2E Multi-Domain orchestration methods can not reach the optimal cross domain resource management strategy.

In this paper, we propose an innovative framework that grants distributed RL agents to take initiative based on their own holistic vision of the network's future. The list of this paper's contributions are as follows:

1) Designing a Novel framework for distributed orchestrators in multi-domain networks. This framework comprised of predictive element, multi-domain domain focus element, and decision making unit.
2) Defining what information to exchange between the orchestrators and how to proceed with this exchange without exposing private information from their environments.
3) Introducing Variational Autoencoders (VAEs) to encode the history of states in the RL agents' environment by encoder model and generate estimated states of the multi-domain network using the decoder model.
4) Introducing attention mechanism to provide a unique weighted holistic vision of the network for each orchestrator.

The rest of the paper is organized as follow: section II discusses the state-of-the-art methods for addressing research management and orchestration for Mobile Network. The proposed framework architecture is described in section III. The performance evaluation of the proposed framework is extensively examined by implementing various test scenarios in section IV. The section V conclude by summarizing the target problem, proposed framework for multi-domain orchestration, and the results of theframework performance evaluation.

II. Related Work

Physical resources are virtualized as VNFs in 5G, which can be dynamically deployed in the cloud, based on service demands. To manage Network Function (NF) scaling for network slices, Radio Resource Management (RRM) framework has been proposed by [7], that utilizes static slicing ratios for resource management in a Sliced RAN. Other approaches such as Seasonal Auto-Regressive Integrated Moving Average (SARIMA) [8] and Markov Chains (MC) [9] have been proposed for predicting Network Slice resources. However, these methods are not efficient and passive due to their inherent limitations as SARIMA struggles with adapting to sudden changes in data, while MC lack the memory required to efficiently handle complex scenarios, and static models lack adaptability and are not optimal for managing NF.

The complexity of 6G network orchestration involves AI-driven approaches for the management of distributed resource and the placement of Virtual Network Functions (VNF). The stringent latency requirements and high Service Level Agreement (SLA) envisioned in 6G necessitate resource management to be distributed in order to guarantee seamless functioning of parallel network slices across multiple domains [10].

Reinforcement Learning (RL) algorithms are well-suited for decision-making in dynamic environments due to their closed-loop feedback mechanism and incentive-driven nature, allowing them to continuously adapt and optimize strategies in real-time. Given the complexity of E2E multi-domain orchestration in 6G, Deep RL based agents are proposed as powerful tools for real-time decision-making in network resource management [11].

However, relying solely on current intra-domain states is sub-optimal due to dynamic traffic changes and inter-domain influences. It might be suggested that providing predictions about future traffic pattern to RL algorithms could advantage the orchestrator, enabling it to take proactive actions rather than relying on the reactive RL algorithms implemented in the mentioned works. To provide a future vision for the RL agent orchestrator, the authors in [12] incorporated predictive elements that learn the traffic patterns coming to slices into the decision-making unit, allowing for advance planning based on traffic load predictions.

Although RL orchestrator, benefiting from predictive ML model, could potentially offer more sustainable and trustworthy decision making in network function placement and resource management within the network domain, there still remain major obstacles and drawbacks of using such methods in multi-domain orchestration, where the network slices could be shared among domains. Recent research efforts try to introduce multi-agent RL methods to manage multi-domain scenarios, where each RL agent manages its own environment and they coordinate to achieve a common goal [13], [14].

Nevertheless, these works have not considered the impact that agents' decisions and domain traffic loads could have on each other domains. Additionally, they have not coupled predictive algorithms with these distributed RL agents to enable advanced planning and future predictions.

Moreover, distributed RL agents in these multi-Agent solutions make their actions in their own respective domain only based on their domain state condition which could not be optimal considering the influence that other domains states could have on them. Therefore, having holistic vision of the multi-domain network could be highly beneficial for the RL agents to make optimal decision to fulfill their expected Application Programming Interfaces (API) requirements. The current approaches for resource management and orchestration may fall short to achieve optimal strategies to manage these dynamic environments, as they have not addressed these vital issues for resource management and orchestration.

III. Proposed Method

Generally, multiple slices could operate simultaneously within a domain. These slices are handled by Network Functions (NFs) providing a variety of services. In order to allocate optimal resources needed to satisfy service demands, orchestrator should scale up or scale down the number of NFsbelonging to each slice, which makes this scenario extremely challenging considering the dynamic changes of the traffic load for each slice in real-time and the inter-domain effect of orchestrators decisions on other domains. This section outlines the methodology of our proposed multi-domain orchestration

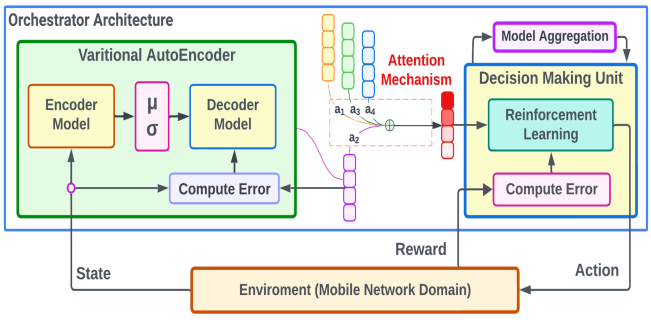

Fig. 2. Proposed structure for Multi-domain orchestrator.

framework, which consists of three main components: a Predictive Element, a Multi-Domain Focus component, and the Decision Making unit, as illustrate in figure 2.

Considering that the dynamic changes within each network domain demands a proactive orchestrator that not only adjust its actions regarding the current state of its environment but it should be able to plan ahead and anticipate the possible changes in near future, each orchestrator has a predictive element that estimates the future states using the compressed knowledge about the past states of the environment. In general, decisions made by the orchestrators within each domain will directly or indirectly affect other domains due to the interconnections in Mobile Networks.

Recognizing that orchestrator decisions might influence each other's environments or that traffic load might migrate between domains, orchestrators should be able to inform each other about their environmental changes in some form. These exchanged information should be compressed and encoded such that they do not reveal the exact historical state data of the domain as they are private domain's data.

For other orchestrators to be able to use these encoded data, they have to reconstruct an approximation of the encoded data by capturing the key features. Our proposed framework contains Latent ODE-based auto-encoder models introduced by [16] as predictive elements, where the encoder model, $\text{ODE}_{\text{enc}}(x_t, \phi)$, compresses all previous historical states into a latent space, providing a valuable abstract representation of them. The Latent ODE is auto-encoder model where both encoder and decoder are Neural ODEs. These encoded latent spaces, denoted as $z_t = q_\phi(z_t \mid x_t) = \text{ODE}_{\text{enc}}(x_t, \phi) = \mathcal{N}(\mu_t, \sigma_t^2))$, where μ and σ are the mean and variance vectors learned by the encoder, are exchanged between the orchestrators in different domains. The latent space of each domain evolves over each step within episodes according to $\frac{dz}{dt} = f_\phi(z_t, t)$ differential equation parameterized by the encoder model Neural Network. To preserve privacy during the exchange of latent spaces, noise is added to the encoded latent space, resulting in $z' = z + \epsilon$, where ϵ denotes the noise. This addition protects the actual historical representation of the domains' states. With the protected encoded states from all domains, each orchestrator can use its decoder model, $p_\theta(\hat{x} \mid z') = \text{ODE}_{\text{dec}}(z_t, \theta)$, to predict an approximation of all domains' future states. Considering the stochastic nature of the encoding process, decoder models could provide similar but not exact predictions of the domains' future states. Importantly, due to the added noise and the fact that each Latent ODE is trained only on its operating domain states, private information of the domains cannot be extracted. Instead, the Latent ODE's decoder model provides a prediction of the future states of other domains. The Latent ODE's encoder and decoder models are regularly updated after each change in the domain state considering the following formula:

$$\mathcal{L} = \mathbb{E}_{q_\phi(z_t|x_t)}\left[\frac{1}{2}\|x_{\text{nxt}} - \hat{x}\|^2\right] + \text{KL}\left(q_\phi(z_t \mid x_t) \parallel p(z)\right) \quad (1)$$

where $\hat{x} = \text{ODE}_{\text{dec}}\left(\text{ODE}_{\text{enc}}(x_t, \phi), \theta\right)$ is the predicted next state by the Variational Auto Encoder (VAE) (Latent ODE) and x_{nxt} is the actual next state. The Kullback–Leibler (KL) divergence denoted by $\text{KL}\left(q_\phi(z_t \mid x_t) \parallel p(z)\right)$ is used to regularize the latent space. Fortunately, this method provides orchestrators with the advantage of estimating the most probable future states of other domains generating valuable information for each orchestrator about the whole multi-domain future.

However, these valuable predictions about the future states of entire domains would not benefit the decision-making unit unless they are processed in a way that a holistic vision of the network's future is created out of these domains state predictions, with special emphasis on the orchestrator's domain. This allows decision-making units to act based on an abstract representation of the network's future, heavily focused on the orchestrator's domain while mildly considering other important domains.

To construct the holistic vision for each domain, our proposed orchestrator structure has a multi-domain focus element which use Gated Attention mechanism proposed by [17] to learn the abstracted provided anticipations of all domain future states, $\hat{x}_{n=1}, \ldots, \hat{x}_i, \ldots, \hat{x}_N$, into a single holistic representation of the entire network with heavy attention to the orchestrator domain and soft focus to other domains. The gate weight assigned to each domain states' future prediction should be computed based on its effect on the orchestrator domain. Acknowledging that it is obvious that each orchestrator should give majority of its attention to its domain, a minimum base value is assigned to the orchestrator domain that could increase if the attention layer learns to give higher weights to the orchestrator domain otherwise the weight assigned to its domain can not be lower than the defined minimum base attention. considering that there are N domains and orchestrator i operates in domain i, the gate weight assigned to the domain i's prediction array, g_i, which is equal to $\gamma + \text{sigmoid}(\beta)$, where $\gamma = 0.75$ in our implementation and β the is a learnable parameter. The rest of gate weights are computed as below:

$$g_n = \frac{(1 - g_i) \cdot \exp(W_g \cdot \hat{x_n} + b_g)}{\sum_{j=1, j\neq i}^{N-1} \exp(W_g \cdot \hat{x_j} + b_g)} \quad (2)$$

Here, g_n, represents the normalized importance score assigned to the domain n's state, $\hat{x}_n$. The parameters W_g and b_g refer to the learnable weight matrix and bias vector, respectively, which are employed in the computation of g_n. As a result the normalized attention score for each domain state would be formulated as follow:

$$\text{AttentionScore}(\hat{x_n}) = g_n \cdot \text{softmax}(Q \cdot K_n^\top) \tag{3}$$

Where Q is the query vector, and K_n is the key vector associated with $\hat{x_n}$. The sum of the attention scores, as expressed by $\sum_{n=1}^{N} \text{AttentionScore}(\hat{x_n}) \cdot V_n$ should be equal to 1, where V_n represents value vector corresponding to $\hat{x_n}$. Considering domains' attention scores, a Holistic Attention Weight is computed according to Equation 4 for each domain and is used for model aggregation at the end of each episode.

$$\text{HAW}_d = \frac{\text{AttentionScore}(\hat{x}_d)}{\sum_{k=1}^{D} \text{AttentionScore}(x_k)} \tag{4}$$

In which HAW_d and $\text{AttentionScore}(\hat{x}_d)$ represent the Holistic Attention Weight and the attention score for domain d. This gated Attention layer helps the orchestrator to learn its domain state condition and how much it is related to the other domains; and in case of change of traffic route this mechanism helps the orchestrator to dynamically change its focus from some domains to the other in real-time, which is an extremely powerful ability for the orchestrator. As a result of this attention layer, each orchestrator can individually determine its focus and strategy independently from any central point. This attention layer is attached to the RL agent's receptive layer and trained alongside the RL agent as a single unit, enabling the RL agent to dynamically learn how to adjust its focus on the domains of the mobile network. As a result, our proposed structure for the distributed orchestrators empowers them to give majority of their attention to the estimated traffic load pattern within their own environment, meanwhile being aware of possible situation of other domains. Empowered by a holistic vision, decision-making units benefit from utilizing this predicted information, making optimal decisions.

The decision-making unit of each orchestrator is a Reinforcement Learning (RL) agent that uses a gated attention layer to focus on the most relevant predictions from other domains. These RL agents manage network domains containing multiple slices with variable service loads over time. In some cases, slices may be shared among domains, presenting a significant challenge for orchestrators in managing their respective domains. The agent aims to optimally increase the number of VNFs handling a service's traffic load when a significant portion of those network function resources are engaged. Conversely, it decreases the number of VNFs when the majority of those resources are abundant. Due to superiority of the Actor-Critic (AC) method explained by [18] in comparison with Deep Q-learning (DQN) proposed in [19], in this section AC considered to be attached to attention mechanism. Therefore, the Temporal Difference (TD) error [18] would be calculated as:

$$\delta_t = r_t + \gamma V(s_{t+1}, c_{t+1}; \theta_c) - V(s_t, c_t; \theta_c) \tag{5}$$

with δ_t denotes the TD error, r_t is the reward at time t, and γ is the discount factor, $V(s_{t+1}, c_{t+1}; \theta_c)$ represents Value of Next State, and $V(s_t, c_t; \theta_c)$ Value of current State. Having the TD error, parameters of the critic model is updated using the subsequent equation:

$$\theta_c \leftarrow \theta_c + \alpha_c \cdot \delta_t \cdot \nabla_{\theta_c} V(s_t, c_t; \theta_c) \tag{6}$$

θ_c stands for the parameters of the critic's value function, α_c is the learning rate for the critic, δ_t refers to TD error, and $\nabla_{\theta_c} V(s_t, c_t; \theta_c)$ signifies the gradient of the value function with respect to the critic's parameters. Thereafter, the policy gradient for actor models is computed as below:

$$\nabla_{\theta_a} J(\theta_a) = \delta_t \cdot \nabla_{\theta_a} \log \pi(a_t \mid s_t, c_t; \theta_a) \tag{7}$$

In which $\nabla_{\theta_a} J(\theta_a)$ refers to the gradient of the objective function with respect to the actor's parameters, and $\nabla_{\theta_a} \log \pi(a_t \mid s_t, c_t; \theta_a)$ refers to the gradient of the log-probability of taking action a_t given state s_t and context c_t, with respect to the actor's parameters. Then , the actor model parameters are updated according to the following formula:

$$\theta_a \leftarrow \theta_a + \alpha_a \cdot \delta_t \cdot \nabla_{\theta_a} \log \pi(a_t \mid s_t, c_t; \theta_a) \tag{8}$$

where θ_a denotes to the parameters of the actor model, α_a is the learning rate for the actor. After updating the Value and policy function, The attention weights including the weight matrix W_g, bias vector b_g, and domain's gate parameters β, are updated at each step of the episode using gradient descent based on the temporal difference (TD) error. The weight matrix W_g and the bias vector b_g are updated based on formulas below considering α_g as a learning rate:

$$W_g \leftarrow W_g + \alpha_g \cdot \delta_t \cdot \nabla_{W_g} \log \pi(a_t \mid s_t, c_t; \theta_a) \tag{9}$$

$$b_g \leftarrow b_g + \alpha_g \cdot \delta_t \cdot \nabla_{b_g} \log \pi(a_t \mid s_t, c_t; \theta_a) \tag{10}$$

And the domain's learnable domain parameters, β, are adjusted according to the following:

$$\beta \leftarrow \beta + \alpha_\beta \cdot \delta_t \cdot \nabla_\beta \log \pi(a_t \mid s_t, c_t; \theta_a) \tag{11}$$

In to accelerate the convergence of the AC agents across domains, the AC model parameters are exchanged between orchestrators to conduct distributed Federated Training, once the training episode is finished. The shared local model parameters among orchestrators are then aggregated by taking into account the computed Holistic Attention Weights of the domains as expressed in formula 12 for the Actor model:

$$\theta_a^{\text{agg, t}} = \sum_{d=1}^{D} \text{HAW}_{d,t} \cdot \theta_a^{d,t} \tag{12}$$

With $\theta_a^{\text{agg,t}}$, and $\theta_a^{(d),t}$ are the generated aggregated actor model and actor parameters for domain d at episode t, respectively. Similarly, aggregated Critic model is generated as below, considering $\theta_c^{\text{agg},t}$ and $\theta_c^{(d),t}$ as aggregated Critic model and its model parameter from domain d at episode t:

$$\theta_c^{\text{agg},t} = \sum_{d=1}^{D} \text{HAW}_{d,t} \cdot \theta_c^{d,t} \tag{13}$$

Algorithm 1 Multi-Domain Orchestrator with Gated Attention

VAE models: $\phi, \theta \leftarrow$ InitializeParameters()
AC parameters: $\theta_a, \theta_c \leftarrow$ InitializeParameter()
Gate Attention: $W_g, b_g, \beta \leftarrow$ Initialize(), and $\gamma = 0.75$
for $t = 1$ to T **do**
  while $(r_t \in [-100, 100])$ **and** $(t < 201)$ **do**
    Encode latent space: $z = q_\phi(z \mid x) = \mathcal{N}(\mu, \sigma^2)$
    Add noise & exchange with other agents: $z' = z + \epsilon$
    for $n = 1$ to N **do**
      if $i = n$ **then**
        $g_i = \gamma + \text{sigmoid}(\beta)$
        Predict next state: $p_\theta(\hat{x_{i+1}} \mid z'_i)$
      else
        Predict next state: $p_\theta(\hat{x_{i+1}} \mid z'_i)$
        $g_n = \frac{(1-g_i)\cdot\exp(W_g \cdot \hat{X}_n + b_g)}{\sum_{j \neq i} \exp(W_g \cdot \hat{X}_j + b_g)}$
      end if
      $\text{AttentionScore}(\hat{X_n}) = g_n \cdot \text{softmax}(QK_n^\top)$
    end for
    for $d = 1$ to D **do**
      $\text{HAW}_d = \frac{\text{AttentionScore}(\hat{X}_d)}{\sum_{k=1}^{D} \text{AttentionScore}(X_k)}$
    end for
    Construct Holistic Vision: $\hat{\mathcal{X}}_\mathcal{H} = \sum_d^D \text{HAW}_d \hat{x_d}$
    Take action, observe change: $x_{t+1} \leftarrow \hat{\mathcal{X}}_\mathcal{H}$
    Compute reward(r_t) and TD loss:

$$\delta_t = r_t + \gamma V(s_{t+1}, c_{t+1}; \theta_c) - V(s_t, c_t; \theta_c)$$

    Update Critic: $\theta_c \leftarrow \theta_c + \alpha_c \delta_t \nabla_{\theta_c} V(s_t, c_t; \theta_c)$
    Update Actor:

$$\theta_a \leftarrow \theta_a + \alpha_a \delta_t \nabla_{\theta_a} \log \pi(a_t \mid s_t, c_t; \theta_a)$$

    Update Gate Attention parameters:

$$W_g \leftarrow W_g + \alpha_g \delta_t \nabla_{W_g} \log \pi(a_t \mid s_t, c_t; \theta_a)$$

$$b_g \leftarrow b_g + \alpha_g \delta_t \nabla_{b_g} \log \pi(a_t \mid s_t, c_t; \theta_a)$$

$$\beta \leftarrow \beta + \alpha_\beta \delta_t \nabla_\beta \log \pi(a_t \mid s_t, c_t; \theta_a)$$

    Compute VAE loss and update VAE models:

$$\mathcal{L} = \mathbb{E}_{q_\phi(z_t|x_t)} \left[\frac{1}{2} \|x_{\text{nxt}} - \hat{x}\|^2 \right] + \text{KL}\left(q_\phi(z_t \mid x_t) \parallel p(z)\right)$$

  end while
  Exchange and generate aggregate AC model:

$$\theta_c^{\text{agg, t}} = \sum_{d=1}^{D} \text{HAW}_d \cdot \theta_c^{d,t}, \quad \theta_a^{\text{agg, t}} = \sum_{d=1}^{D} \text{HAW}_d \cdot \theta_a^{d,t}$$

end for

The learned attention vector enables the orchestrator to create its aggregated local RL model, where the models send by RL agents operating in the most important domains have a greater influence than those in less relevant domains. The Algorithm 1 describes the pseudocode of our proposed HERO framework.

TABLE I
TESTBED CONFIGURATION

Clusters	***Containers***
2 Clusters	Min: 10, Max: 50
3 Clusters	Min: 15, Max: 75
5 Clusters	Min: 25, Max: 125

TABLE II
TRAINING PARAMETERS

Parameter	***Value***
Episodes	400
Episode Duration	200 Steps
Eval Metric	Avg Reward

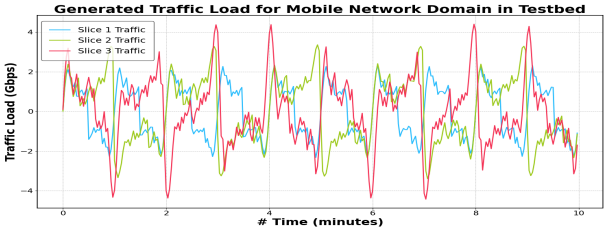

Fig. 3. Slice Traffic Patterns in domain handling 3 network slices.

IV. PERFORMANCE EVALUATION

For purpose of performance evaluation of our proposed framework, we have developed an emulation testbed based on Containernet [20] and Knetsim [21] opensource projects. Our developed testbed provides a multi-domain network where a variety of slices are shared between domains. These slices have limited resources and unique traffic patterns that are handled by worker nodes which are implemented as Containerized network functions (CNFs). Figure 3 displays the total traffic load coming to each domain. Each domain structure imitates the Mobile Network Domain having SDN Controllers, Load Balancers, and CNFs.

The goal of our testbed is to provide a Multi-Domain environment where orchestrators can exchange protected encoded information about their historical states and their updated RL model parameters. This exchange aims to enrich orchestrators' perspectives on the global network state and enhance their decision-making models. In this testbed, the virtual links between network elements are considered to be at their fullest resource potential. Similar to the Kubernetes structure, in our testbed, each domain contains multiple master nodes deployed as real containers responsible for data configuration, status information about the worker nodes, and assigning workloads by routing traffic to the worker nodes (i.e., CNFs). Essentially, the orchestrator oversees multiple slices and makes decisions to deploy or remove worker nodes (CNFs) within a domain.

These decisions are executed by a scheduler that deploys worker nodes (CNFs) under the supervision of a master node to handle the workload of a particular network slice. The aim of the orchestrator is to deploy the minimum number of worker nodes necessary to ensure that domain resources are optimally used for service provisioning. To achieve this, the orchestrator should manage minimum number of CNFs, whose CPU and memory usage remain within acceptable thresholds, defined by lower and upper limits. Tables I and II describe the testbed container scale configurations and the training parameters for

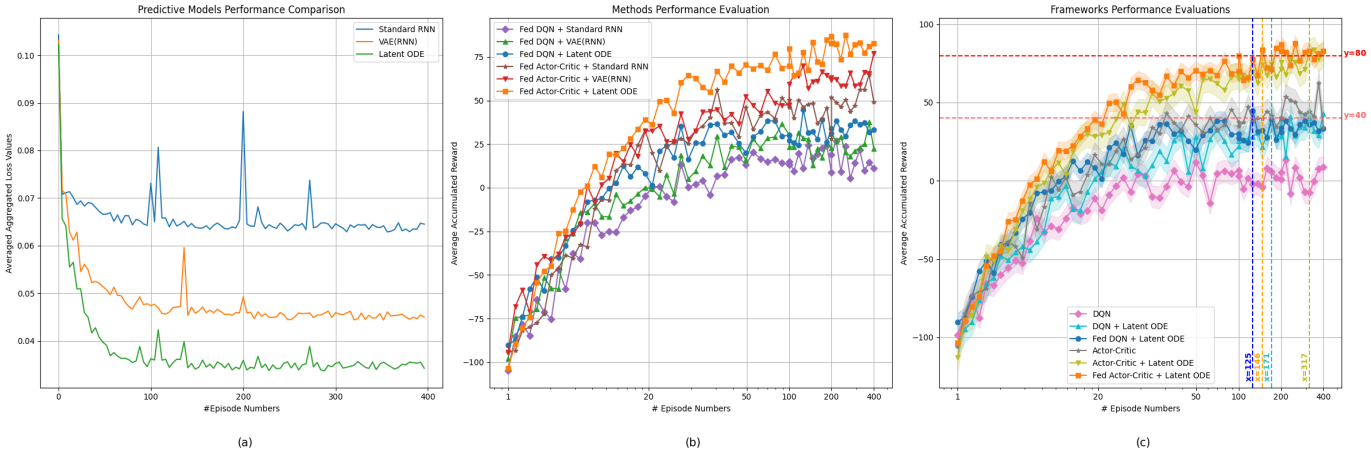

Fig. 4. (a) illustrates performance of various predictive ML models coupled with Attention-Augmented AC model, where Latent ODE achieves lower MSE loss and has more stable predictions compared to other methods. (b) demonstrates different combinations of predictive algorithms and decision-making units in our Holistic Predictive framework, where RL models are updated via Federated Training. (c) evaluates performance across three scenarios: only RL agents, Attention-Augmented RL agents with prediction model, and Federated Training for Attention-Augmented RL agents coupled with a predictive algorithm.

this framework and algorithms comparison, respectively.

Three algorithms were developed for predictive elements: a standard Reccurent Neural Network (RNN), a VAE (RNN), and a Latent ODE, along with implementing AC and DQN for decision-making units. To evaluate the performance of the proposed framework against current state-of-the-art solutions for managing resources in networks, our testbed provides a variety of scenarios. These scenarios allow the orchestrator to either operate with only a decision-making unit using state-of-the-art reactive methods, integrate a predictive element alongside the decision-making unit to enable proactive action selection, or employ an auto-encoder predictive element, a multi-domain focus component, and a decision-making unit. This configuration enables the orchestrator to proactively make decisions based on its environment and other important related domains. Moreover, this platform allows the orchestrators to conduct federated learning to enhance the training of their decision-making units. The first scenario is conducted to find the best combination of the implemented predictive and decision-making algorithms within our proposed framework, in which all the orchestrators participate in federated training for generating their private aggregated RL model. Among all combinations of implemented predictive models and decision-making units, the Fed Actor-Critic augmented with the Latent ODE came out as the most reliable and powerful combination. As expected, the scenarios where the Actor-Critic algorithm is used outperform those where DQN is used as RL models. The accuracy, the performance, and the reliability of implemented predictive models are shown in Figure 4 (a) and Table III, where these models have been coupled with the Actor-Critic RL model within our framework. As illustrated in Figure 4 (a), the Latent ODE provides more reliable, trustworthy, and accurate predictions of CPU and memory usage of the worker nodes (CNFs) within the domain compared to the VAE (RNN) and Standard RNN.

The results in Fig 4 (b) demonstrates that the RL agents

TABLE III
PERFORMANCE COMPARISON OF PREDICTIVE ALGORITHMS

Methods	***Accuracy***
Standard RNN	0.936
VAE(RNN)	0.954
Latent ODE	**0.965**

coupled with the more powerful predictive algorithm, specifically the Latent ODE, could reach a higher average reward per episode due to their more reliable and accurate predictive element. A detailed comparison of different combinations of predictive algorithms and RL methods across various domain scales is described in Table IV. For sake of evaluating our proposed orchestration framework with other traditional methods we have examined solution with only decision making units, coupling predictive element with decision making unit, and our proposed framework that contain all auto-encoder predictive model that can encode the historical states and decode the exchanged encoded historical state from other domains and focus element using Gated attention layer and the decision making unit that regularly updated its models with aggregated model generated through federated training.

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT SCENARIOS

Methods	***End Avg Reward***	***Avg Max Reward***	***Avg Dev***
DQN	8.79	13.24	± 6.13
DQN + Latent ODE	42.63	42.63	± 6.89
Fed DQN + Latent ODE	33.33	44.80	± 5.14
AC	34.03	71.32	± 6.67
AC + Latent ODE	81.154	83.59	± 6.73
Fed AC + Latent ODE	**82.66**	**87.61**	± **5.37**

TABLE V
HIGHEST AVG SCORE COMPARISON OF DIFFERENT METHODS

Methods	*2 Clusters*	*3 Clusters*	*5 Clusters*
Fed DQN + Standard RNN	26.36 ±2.7	26.74 ±3.4	26.94 ±5.2
Fed DQN + VAE(RNN)	36.98 ±2.5	37.11 ±3.2	37.52 ±5.1
Fed DQN + Latent ODE	44.23 ±2.6	44.49 ±3.3	44.80 ±5.1
Fed AC + Standard RNN	64.44 ±3.1	64.58 ±3.5	64.72 ±5.5
Fed AC + VAE(RNN)	76.59 ±3	76.86 ±3.5	77.01 ±5.3
Fed AC + Latent ODE	**87.61 ±2.9**	**87.61 ±3.4**	**87.61 ±5.3**

For the methods that have both predictive element and decision making we have consider the latent ODE as predictive model. The result of this comparison is depicted in fig 4(c), showing proactive methods, which exchange their encoded historical states and use auto-encoder predictive model and attention layer, can achieve higher average reward than the methods only using RL models to make decision based on the current state. Based on this result depicted in fig (c) and Table V, these methods enhanced by holistic predictive vision can reach slightly higher average rewards in fewer training episodes in case if they update their RL models within a federated training.

As final point to mention, Achieving High accuracy, fast convergence are not the only advantages of our proposed Hero method and its non federated form have Proactive decision making, holistic vision capabilities for mobile network for multi-domain orchestration. Nevertheless these advantages comes with cost of having higher communication overhead due to exchange of encoded information and model parameters where bigger RL model would lead to higher communication overhead during training and sustained communication overhead during inference.

CONCLUSION

This paper proposed a framework for End-to-End Multi-Domain resource management and orchestration for 6G, promising continuous intelligence across multiple domains. Current solutions often make sub-optimal decisions caused by their limited foresight to anticipate future states, and they neglect to consider that actions of orchestrators and the conditions of their respective domains affect each other's future states. Our framework addresses these problems by allowing orchestrators to exchange their encoded historical states as well as their RL model weights among themselves in privacy preserving manner. These exchanged encoded pieces of information are then utilized by our novel HERO to make decisions based on a holistic estimation of the entire network's future state, constructed by factoring the significance of each domain's future estimation for the respective orchestrator. Finally, our proposed HERO method was evaluated against other state-of-the-art solutions and various algorithm combinations within our framework for multi-domain orchestration across different scenarios in our testbed. The results demonstrate that HERO delivers remarkably more reliable and accurate resource management compared to other combinations and achieves higher average rewards in fewer episodes when trained using the Federated Training procedure.

REFERENCES

[1] Chergui, H., Ksentini, A., Blanco, L. & Verikoukis, C. Toward zero-touch management and orchestration of massive deployment of network slices in 6G. *IEEE Wireless Communications*. **29**, 86-93 (2022)

[2] Velasquez, K., Abreu, D., Curado, M. & Monteiro, E. Resource orchestration in 5G and beyond: Challenges and opportunities. *Computer Communications*. **192** pp. 311-315 (2022)

[3] Letaief, K., Chen, W., Shi, Y., Zhang, J. & Zhang, Y. The roadmap to 6G: AI empowered wireless networks. *IEEE Communications Magazine*. **57**, 84-90 (2019)

[4] Camelo, M., Cominardi, L., Gramaglia, M., Fiore, M., Garcia-Saavedra, A., Fuentes, L., De Vleeschauwer, D., Soto-Arenas, P., Slamnik-Krijestorac, N., Ballesteros, J. & Others. Requirements and Specifications for the Orchestration of Network Intelligence in 6G. *2022 IEEE CCNC*. pp. 1-9 (2022).

[5] Moubayed, A., Shami, A. & Al-Dulaimi, A. On end-to-end intelligent automation of 6G networks. *Future Internet*. **14**, 165 (2022)

[6] Sami, H., Otrok, H., Bentahar, J. & Mourad, A. AI-based resource provisioning of IoE services in 6G: A deep reinforcement learning approach. *IEEE TNSM*. **18**, 3527-3540 (2021)

[7] Pérez-Romero, J., Sallent, O., Ferrús, R., & Agustí, R. (2018). On the configuration of radio resource management in a sliced RAN. *NOMS 2018-2018 IEEE/IFIP Network Operations And Management Symposium, 1-6.

[8] Yu, Y., Wang, J., Song, M. & Song, J. Network traffic prediction and result analysis based on seasonal ARIMA and correlation coefficient. *2010 International Conference On Intelligent System Design And Engineering Application*. **1** pp. 980-983 (2010)

[9] Xie, Y., Hu, J., Xiang, Y., Yu, S., Tang, S. & Wang, Y. Modeling oscillation behavior of network traffic by nested hidden Markov model with variable state-duration. *IEEE TPDS*. **24**, 1807-1817 (2012)

[10] Guan, W., Zhang, H. & Leung, V. Customized slicing for 6G: Enforcing artificial intelligence on resource management. *IEEE Network*. **35**, 264-271 (2021)

[11] Chen, J., Chen, J. & Zhang, H. DRL-QOR: Deep reinforcement learning-based QoS/QoE-aware adaptive online orchestration in NFV-enabled networks. *IEEE TNSM*. **18**, 1758-1774 (2021)

[12] Bouroudi, A., Outtagarts, A. & Hadjadj-Aoul, Y. A dynamic AI-based algorithm selection for Virtual Network Embedding. *Annals Of Telecommunications*. pp. 1-17 (2024)

[13] Sulaiman, M., Moayyedi, A., Salahuddin, M., Boutaba, R. & Saleh, A. Multi-agent deep reinforcement learning for slicing and admission control in 5G C-RAN. *NOMS 2022-2022 IEEE/IFIP Network Operations And Management Symposium*. pp. 1-9 (2022)

[14] Mason, F., Nencioni, G. & Zanella, A. A multi-agent reinforcement learning architecture for network slicing orchestration. *2021 MedComNet*. pp. 1-8 (2021)

[15] Rezazadeh, F., Zanzi, L., Devoti, F., Chergui, H., Costa-Perez, X. & Verikoukis, C. On the specialization of fdrl agents for scalable and distributed 6g ran slicing orchestration. *IEEE TVT*. **72**, 3473-3487 (2022)

[16] Rubanova, Y., Chen, R. & Duvenaud, D. Latent ordinary differential equations for irregularly-sampled time series. *Advances In Neural Information Processing Systems*. **32** (2019)

[17] Dhingra, B., Liu, H., Yang, Z., Cohen, W. & Salakhutdinov, R. Gated-attention readers for text comprehension. *ArXiv Preprint ArXiv:1606.01549*. (2016)

[18] Sutton, R. & Barto, A. Reinforcement learning: An introduction. (MIT press,2018)

[19] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J., Bellemare, M., Graves, A., Riedmiller, M., Fidjeland, A., Ostrovski, G. & Others Human-level control through deep reinforcement learning. *Nature*. **518**, 529-533 (2015)

[20] Peuster, M. Containernet. (https://github.com/mpeuster/containernet), Accessed: 2023-08-17

[21] IBM, "k8s-netSim: Kubernetes Network Simulator," https://github.com/IBM/k8s-netsim.