Article

6G Self-Evolution Based on Digital Twin Network

Yuhong Huang , Mancong Kang * , Yanhong Zhu , Na Li , Guangyi Liu , Qixing Wang

Posted Date: 24 March 2025

doi: 10.20944/preprints202503.1730.v1

Keywords: 6G; digital twin network; pre-validation environment; graph neural network

Article

6G Self-Evolution Based on Digital Twin Network

Yuhong Huang [1], Mancong Kang [1,*], Yanhong Zhu [1,2], Na Li [1,3], Guangyi Liu [1] and Qixing Wang [1]

[1] China Mobile Research Institute, Beijing 100053, China

[2] School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing 100091, China

[3] ZGC Institute of Ubiquitous-X Innovation and Application, Beijing 100191, China

Abstract: Digital twin (DT) will revolute network autonomy. Current studies have promoted DT-native 6G network by deeply integrate DT into mobile network architectures to improve the timeless of physical-digital synchronization and network optimizations. However, DT has mainly acted as just a tool for network autonomy, leading a gap towards the ultimate goal of network self-evolution. This paper analyzes the future direction of the DT-native network. Specifically, the proposed architecture introduces a key concept called "future shots", which gives accurate network predictions under different time scales of self-evolution strategies for various network elements. To realize the future shots, we propose a long-term hierarchical convolutional graph attention model for cost-effectively network predictions, a conditional hierarchical graph neural network for strategy generations, and methods for efficient small-large scale interactions. The architecture is expected to facilitate high-level network autonomy for 6G networks.

Keywords: 6G; digital twin network; pre-validation environment; graph neural network

1. Introduction

In future mobile communication networks, emerging technologies and more flexible, open network architectures will be integrated to efficiently meet the demands of rapidly evolving applications such as low-altitude economy, immersive communications, and mobile large-scale AI models, etc [1]. Network parameters will become increasingly complex, network environments will be more heterogeneous in terms of vendors and resources, and service traffic will be denser and more diverse. These trends will significantly increase the complexity and cost of network operation optimization and maintainance, necessitating efficient network autonomy methods. Against this backdrop, Network Digital Twin (NDT) become particularly critical [2]. By intelligently mapping the physical network into a digital space, NDT enables autonomous network inference and intervention [3], facilitating self-configuration, seelf-healing, self-optimization, and eventually — self-evolution.

Due to the established architecture of 5G networks, current industry explorations of NDT primarily adopt an external NDT approach to realize fragmented network autonomy [4], characterized by the following aspects. Firstly, NDT is often employed to address specific issues, such as wireless coverage pre-verification [5], network traffic prediction [6], and user behavior trajectory forecasting. Different autonomy use cases typically require siloed, independent designs, leading to high expert costs and a lack of comprehensive consideration of multiple factors. Secondly, existing NDT solutions primarily rely on large-scale historical physical network data to train inference models offline [7]. This results in high data collection costs, long model update cycles, and the inability to optimize the physical network in real time. Consequently, this external approach can only serve network operation and maintenance use cases with low real-time requirements, failing to support network optimizations in running state that demand high real-time performance.

To address the limitations of external NDT, researchers have proposed the integration of NDT into 6G network design from the outset, forming a DT-native network architecture [8]. This approach embeds NDT as an internal function within different network layers or network elements, such as OAM, RAN, and gNB, enabling real-time data synchronization, dynamic modeling, and closed-loop control to serve both operational and running states. The NDT model is primarily constructed using Software Development Kits (SDKs) and runtime/performance models provided by equipment vendors. Data synchronization can be performed locally or flexibly across different DT locations through a newly designed 6G data plane. Ultimately, by orchestrating and integrating models and data from various sources, diverse digital twin tasks can be efficiently completed, enabling network awareness (e.g., user QoS analysis) and network self-optimization (e.g., intelligent resource management) for different network use cases.

However, current DT-native networks primarily focus on using embedded DT to support fragmented use cases, falling short of achieving self-evolution network. Attaining self-evolution network is a highly challenging and complex endeavor, requiring consideration of several key factors. Firstly, it is essential to accurately and cost-effectively predict the overall network status and the status of individual network elements over various time periods. This enables the timely detection of potential issues and large-small time scale performance degradation trends. However, the presence of numerous network elements—including network nodes, terminals, services, and environmental factors—renders short-time-scale and long-time-span predictions computationally prohibitive. Additionally, predicting each network element necessitates considering multiple complex network factors, further increasing prediction costs. Secondly, network evolution strategies range from minor parameter optimizations to major technological changes, such as the introduction of cell-free or terahertz technologies. Precisely selecting appropriate network strategies for different scenarios to evolve the network remains a key challenge. Finally, network interactions should occur across different architectural layers in non-real-time, near-real-time, and real-time modes, posing further implementation challenges.

To address these challenges, this paper proposes a self-evolution enbled DT-native network framework. Specifically, the framework introduces a key concept called "future shots," which gives accurate network state predictions under future evolution strategies across different time scales for various network elements. To enable future shots, we propose a comprehensive network prediction method, and an approach for generate evolution strategies, which operates across different time scales. In detail, for network predictions, we first introduce a cross-time hierarchical graph attention model (CTHGAM) to generates adaptive prediction strategies—including prediction period and prediction time-length —for different network elements. Then, we present a long-term hierarchical convolutional graph attention model (LTHCGAM) to cost-effectively predict network states and strategy performance under the predicting strategy. For the approach of generating evolution strategy across different scales, we first employ a Large Language Model (LLM) to generate a potential network evolution strategy. Next, we propose a conditional hierarchical Graph Neural Network (GNN) model to select the specific network elements requiring the evolution. Finally, an LLM is used to generate the final evolution strategies for these network elements. Furthermore, we analyze hierarchical network interaction strategies required for DT-native networks across different time and scale dimensions. The main contributions of this paper are summarized as follows:

- First, an architecture is proposed for DT-native network to realize 6G self-evolution, which include a new concept of “future shot” for predicting future states and evolution strategy performance.
- Second, we propose a full-scale network prediction method for requirement predictions and strategy validations, which include a CTHGAM model for generating predicting strategy and a LTHCGAM for generating predicting results.
- Third, method for generating evolution strategy is proposed, which incoporates a conditional hierarchical GNN for selecting evolution elements and LLM for giving evolution strategy models. In addition, we design efficient hierarchical virtual-physical interaction strategies.
- Finally, we analyze four potential applications of the proposed DT-native network.

The remainder of our work is organized as follows. Section 2 gives the proposed architecture of DT-native network. Section 3 gives the detail of three key technologies of DT-native network. Section 4 gives several potential use cases of the proposed DT-native network. Finally, we conclude this paper in Section 5.

2. Architecture of DT-Native Network

This section gives the architecture of DT-Native Network, which gives a detailed understanding of what reflects "DT-Native", the key features of it, and the running paradim.

2.1. Overview

A DT-native network comprises a hiarachical physical network and DTs of different network levels, as depicted in Figure 1. Concretely, the DTs are deployed in a hybrid centralized-distributed form. There is a global DT for monitoring and evoluting the whole physical network, and several local DTs (LDT) of different physical entities (gNBs, terminals, AMFs, etc.) for managing local areas.

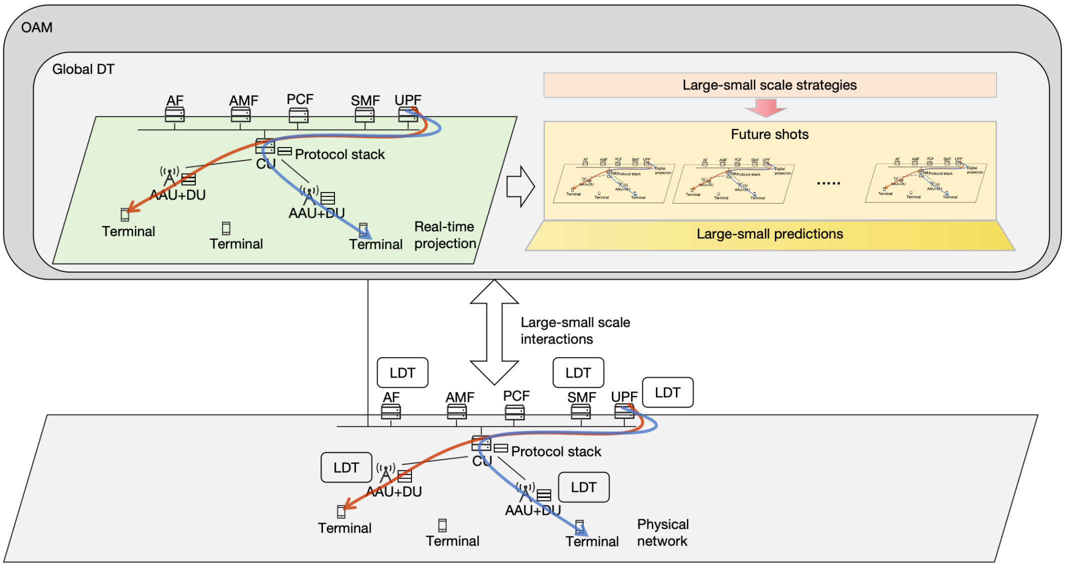

Figure 1. Architecture of DT-native network.

In each DT, there are modules for real-time projection, future shots, autonomous large-small scale strategy generation (ALSSSG), differentiating long-short term prediction (DLSTP) and large-small time scale controls (LSTSC). In detail, the real-time projection module is a sythchronization of physical network in virtual space, including its states and strategies, which is the basis for evoluting reference. Future shots module shows the evoluting path of future physical network in different future time based on DLSTP, under the strategies generated by ALSSSG. Finally, the DT interacts with physical counterpart through LSTSC to realize network evolution.

To be mention, a DT has a wider virtual space than the physical space. The virtual space is immensive, has soft and dynamic edges and nodes, where only a part of them have counterparts in physical network. The DT dynamically adds or deletes nodes and edges from physical network according to the reflections from virtual apace. The adding action can be implemented by robots or humans to put the physical equipments on the right place, or implemented by DT itself to deploy softwares on cloud-native general service platform. The deleting action can be made by setting the equipment into sleep mode, or completely removed out of the network by humans.

2.2. Future Shot

The concept of "future shot" is a vital feature of DT-native network. In a DT, its future shots part gives numerous future shots of its physical counterpart, which is an adding results of the dynamic network self-changes and the network strategies. The future shots are a collaborative production of several technologies, such as network prediction, strategy generation, pre-validation, etc. In this way, DT-native network can realize autonomous network planning, optimization and evolving. There are two key questions to answer:

What time period in future shot: Nurmous future shots would lead enormous cost because of the large network scale and computing workload for predictions. To reduce cost, the predicting time periods of different elements are different, which may change in varying-time. On the one hand, part of the elements may be the most problemable part, whose number would increase with the length of time goes large, e.g., according to the ages of different types of equipments, the critical or challenging traffics, movements of people, etc. Accordingly, their predicting time period would get smaller with time going by. On the other hand, near future shots are proper to fine-grain, while long future shots are proper to groase-grain. For example, near future shots can predict the states of network elements or types of resources, and give resource optimization results for them for real-time optimization. By comparison, long future shots can predict the network capacity and overall service qualities, and give large-scale network evolution strategies like adding new technologies, e.g., Teherz, RIS, etc., and new AAU station planning. In addition, the predicting time period of elements in near and long future shots may dynamically changing according to the real condition of physical network.

What needs to be done in generating future shot: It is an adding results of the predictions of dynamic network self-changes and the performance under autonomous network strategies. First, the basic work is to predict future states of self-changing elements, e.g., users trajectories, traffic distributions, etc. Second, it needs to predict the performance of small-scale optimization strategies and large-scale evolution strategies. If the performance does not achieve the performance requirement, the DT automatically analyze the bad reason, generate and train a proper strategies whose performance updates the network condition in future shots. If the ultimous performance still cannot reach the requirement, the DT would inform operators to seek for solutions in advance.

In summary, in order to modify the physical network in advance, DT-native network needs to generate a series of future shots in near and long future time for short-long scale network evolution.

2.3. Hybrid Centralized-distributed Autonomy

A physical mobile network with hiarachical feature is proper to use hybrid centralized-distributed DTs for autonomous network management with high efficiency. While OAM deploys the centric global DT, RAN deploys sub-centralized RAN LDT and network elements deploys distributed LDT.

In this architecture, the future shots are hiarachically generated and managed. Parts of elements in different layers of future shots may repeat, while others are different. The repeated part can be upper-transfer to the higher-layer to reduce repeatedly building cost. On central equipment, elements in future shots are more global view, include two types: first, which part may play a vital part on future global performance or behavior trend, which is inignoreable when predict the acting influence of other part; second, which part may have potential global problems, e.g., traffic jam, network overload, etc. Similar, on distributed equipment, the elements in future shots also need to include the critical elements that can vitally influence future local performance, and the potentially problematic ones, e.g., the behavior and performance of high-priority user device need to predict in future shots.

It needs to be mention, to realize hiarachical controls, DTs of different layers may use different technologies, which may be intertwined with each other performance. For example, cell-free technology are RAN-level technology, which may influence RIS (for single gNB) performance.

3. Key Technologies of DT-Native Network

In DT-native network, to realize network self-evolution, efficient DLSTP and ALSSSG need to design to support future shots, and proper LSTSC needs to proposed to accomplish physical evolution. Therefore, this section sequentially gives general methods for DLSTP, ALSSSG and LSTSC to facilitate self-evolution in 6G networks.

3.1. *Differentiating Long-short Term Prediction*

In DT-native network, DLSTP needs to predict the future states and strategy performance (also called pre-validation) for different network elements. The states and performance can be taken as properties of network elements, e.g., properties of a gNB can include the number of access terminals (states) and cell capacity (preformance). The predicting process can be split into two steps, first give predicting strategy (including predicting time period and predicting time length) for different elements, then give predicting results. The detail is as follows.

3.1.1. Determining Predicting Strategies

Issue: DLSTP needs to determine what are the potentially problematic network elements in the line of future time going by, which needs to balance the system performance and computing & memory cost. These may include self-changing ones, like mobile device trajectory, services traffic, equipement default probability, and the performance metrics like network capacity and workload distribution on different network elements. Considering the properties of different elements are twisted with each other, DLSTP needs to jointly considerate different elements to give predicting strategies. The strategy needs to include predicting time period and predicting time-length.

Solution: To generate proper predicting strategy, we propose a cross-time hierarchical graph attention model (CTHGAM), denoted as $\mathbb{S}_{\text{O,CTHGAM}}$. The input is the predicting results for future, with different predicting time period and predicting time number (their product result is the predicting time-length). Based on this, the model needs decide whether any element performance is decreasing, which need to give smaller predicting time period and larger predicting time number. Meanwhile, it needs to decide whether an element performance influence other critical performance, which can be trained to learn via a predicting loss function. The output gives the predicting time period and predicting time number, suggested for the next predicting decision period, for different elements. The detail is as follows:

Firstly, entities are grouped based on hierarchical relationships. Globally connected components (e.g., AMF, UPF, CU) form one group, while DUs under different CUs and user devices connected to different AAUs are assigned separate groups. Each group is denoted as Ψ_g ($g \in \mathbb{G}$), where $\mathbb{G} = \{1, 2, \ldots, G\}$.

Secondly, the model inputs the states, performances, strategies, and previous predicting time periods of different network elements from different groups, as well as the hierarchical network topology for generating predicting strategy. The architecture of the CTHGAM can be expressed as

$$\begin{gathered}\mathbb{S}_{\text{O,CTHGAM}}(\mathbb{S}_{n\in\Psi}, \mathbb{P}_{n\in\Psi}, \mathbb{U}, \zeta_{n\in\Psi}, \tau^{\text{now}}_{n\in\Psi}) = \\ \text{FCN}\Big(\text{GphAttention}\Big(\Big\{\text{MultiHead}_g\Big(\text{TA}\Big((\mathbb{S}_{n,i-1}, \mathbb{P}_{n,i-1}, \zeta_{n,i-1}), \tau^{\text{now}}_{n,i-1}\Big)|_{i=1,\ldots,I;n\in\Psi_g}\Big)\Big\}|_{g\in\mathbb{G}}, \mathbb{U}\Big)\Big)\end{gathered} \tag{1}$$

where $(i-1)$ represents the lastest prediction iteration, $\tau^{\text{now}}_{n,i-1}$ gives the time strap of latest predicting result of the n-th physical element, $\mathbb{S}_{n,i-1}$ is the state of the n-th physical element, $\mathbb{P}_{n,i-1}$ is its network performance, $\zeta_{n,i-1}$ is its network strategy which directly influences network performance. The model comprises four layers, where the first layer comprises a time alignment part $\text{TA}(\cdot)$; the second layer comprises a few identical model parts $\text{MultiHead}_g(\cdot)$ ($g \in \mathbb{G}$) for intra-group feature extractions; the third layer comprises the model part $\text{GphAttention}(\cdot)$ for extracting the inter-group features; the fourth layer comprises a fully connected network (FCN) for feature integration to generate inferring results. The detail of the whole architecture is depicted in Figure 2. The detail of each layer is designed as follows.

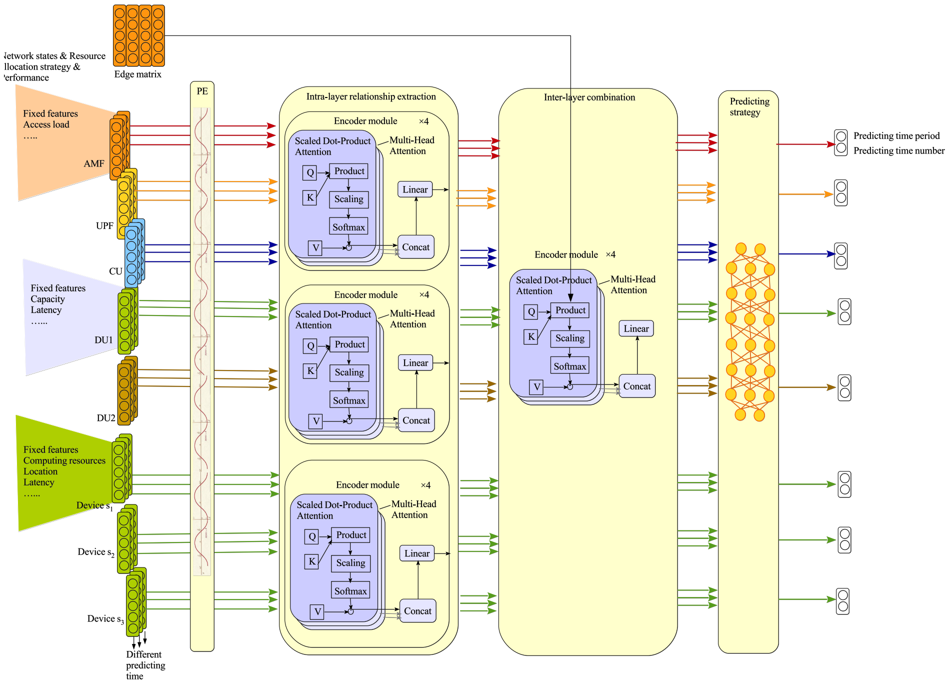

Figure 2. Proposed cross-time hierarchical graph attention model (CTHGAM).

In the first layer, time alignment $\mathrm{TA}(\cdot)$ aims at adding different each time straps $\tau_{n,i-1}^{\mathrm{now}}$ onto each element input vector $(\mathbb{S}_{n,i-1}, \mathbb{P}_{n,i-1}, \zeta_{n,i-1})$. It is designed using the expression of positional encoding based on sine and cosine functions

$$\begin{aligned}
\mathrm{TA}\Big((\mathbb{S}_{n,i-1}, \mathbb{P}_{n,i-1}, \zeta_{n,i-1}), \tau_{n,i-1}^{\mathrm{now}}\Big) &= \mathrm{FCN}(\mathbb{S}_{n,i-1}, \mathbb{P}_{n,i-1}, \zeta_{n,i-1}) + \Big\{\sin(\tau_{n,i-1}^{\mathrm{now}}/10,000^{2j/d})\Big\}|_{j=1,\ldots,d} \\
\mathrm{TA}\Big((\mathbb{S}_{n,i-1}, \mathbb{P}_{n,i-1}, \zeta_{n,i-1}), \tau_{n,i-1}^{\mathrm{now}}\Big) &= \mathrm{FCN}(\mathbb{S}_{n,i-1}, \mathbb{P}_{n,i-1}, \zeta_{n,i-1}) + \Big\{\cos(\tau_{n,i-1}^{\mathrm{now}}/10,000^{2j/d})\Big\}|_{j=1,\ldots,d}
\end{aligned} \tag{2}$$

where d is the dimension of the property vector, and i denotes different elements in input vector.

In the second layer, network elements from different groups transmit their states, strategies, and performance to $\mathrm{MultiHead}_g(\cdot)$, which captures entity features and inter-group relationships using an attention-based encoder. This process is expressed as follows

$$a_1^{(3)}, a_2^{(3)}, \ldots, a_{d\times n_{\mathrm{G}_g}}^{(3)} = \mathrm{MultiHead}_g\Big(\mathrm{TA}\Big((\mathbb{S}_{n,i-1}, \mathbb{P}_{n,i-1}, \zeta_{n,i-1}), \tau_{n,i-1}^{\mathrm{now}}\Big)|_{i=1,\ldots,I; n\in\Psi_g}\Big) \tag{3}$$

where n_{G_g} denotes the number of entities in the g-th group, $a^{(3)}$ is the second-layer output. Thus, each entity's property vector in $a^{(3)}$ integrates its original features with both known and unknown inter-group relationships. For instance, CU-related vectors capture positions and connections with AMF and UPF, DU-related vectors reflect resource and user access dynamics, and user device-related vectors learn data distribution and wireless environment interactions within the same AAU.

In the third layer, outputs from different groups pass through a graph attention module, which computes Q and K matrices and applies hierarchical topology. The module's architecture is as follows

$$\mathrm{GphAttention}(\cdot) = \mathrm{softmax}\left(\frac{QK^T\mathbb{U}}{\sqrt{d}}\right)V \tag{4}$$

where V is the value matrix. The third layer processes vectors from different element groups as follows

$$a_1^{(4)}, \ldots, a_{d\times n_{\mathrm{G}_1}+d\times n_{\mathrm{G}_2}+d\times n_{\mathrm{G}_3}}^{(4)} = \mathrm{GphAttention}(\{a_1^{(3)}, a_2^{(3)}, \ldots, a_{d\times n_{\mathrm{G}_g}}^{(3)}\}|_{g\in\mathrm{G}}, \mathbb{U}) \tag{5}$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2025 doi:10.20944/preprints202503.1730.v1

By incorporating GphAttention$(\cdot)$, the third layer integrates features of groups with direct relationships, such as user devices and their associated AAU. Additionally, multiple repetitions of GphAttention$(\cdot)$ enable integration of weakly related groups, offering supplementary references for the final learning objective.

The fourth layer is a feature integration layer that merges various element features. Its output represents the global intelligent model's inference goals, such as latency and capacity estimations. Aligned with the FCN architecture, it is expressed as follows

$$[\text{PD}_{\text{len},n}, \text{PD}_{\text{num},n}]_{n\in\Psi} = \text{FCN}(a_1^{(4)}, ..., a^{(4)}_{d\times n_{\text{G}_1}+d\times n_{\text{G}_2}+d\times n_{\text{G}_3}}) \tag{6}$$

where $\text{PD}_{\text{len},n}, \text{PD}_{\text{num},n}$ are the predicting period and predicting time number of the n-th element.

3.1.2. Giving Predicting Results

Issue: After obtaining a proper predicting strategy, the next step for DLSTP is to generate predicting results for different network element based on the predicting strategy. The predicting result for each element needs to jointly consider its own previous states and the co-influence of different elements, under an acceptable cost.

Solution: To efficiently generate accurate predicting result, we propose a long-term hierarchical convolutional graph attention model (LTHCGAM). In detail, future time are divided into different time eras which indicates the network gradually evolves into different eras, e.g., by introducing new technologies or accomplish new network planning solutions. In each time era, predictions for different elements are iterated between accurate prediction and light predictions (i.e., a hybrid predicting process), where the predicted time period and predicted time number are determind by the proposed predicting strategy and are different between elements. The predicting results would be integrated into different future shots, where the groasely predicted results remain stable in small-time scale, so as to align the time periods between different elements. When a time era is over, the last predicting results of it are sending to the next hybrid prediction module designed for the next era. Different hybrid prediction modules between time eras are similar but with different sizes to adapt to different number of features in different eras (e.g., new technology introduced). The process is depicted in Figure 3.

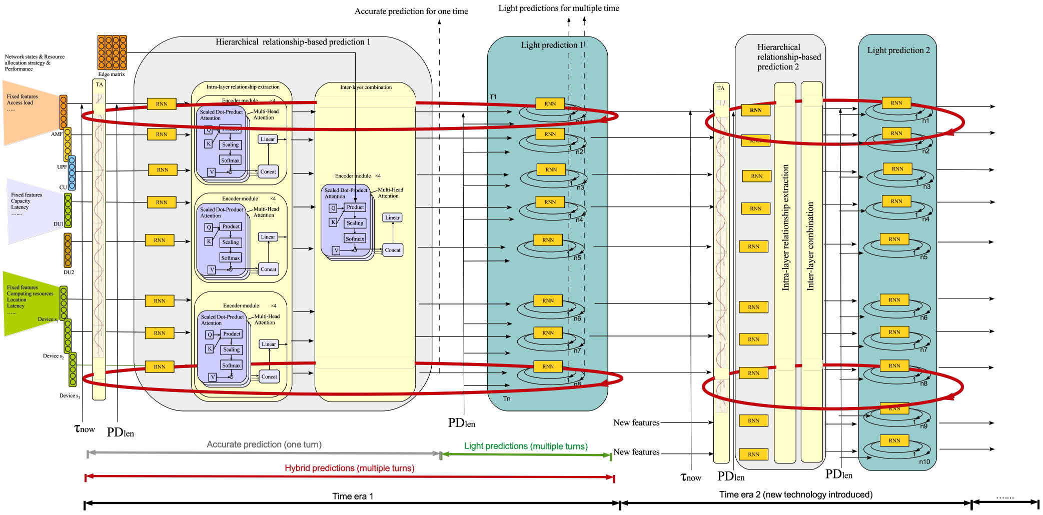
Accurate prediction for one time
Light predictions for multiple time
Network states & Resource allocation strategy & Performance
Edge matrix
TA
Hierarchical relationship-based prediction 1
Intra-layer relationship extraction
Inter-layer combination
Encoder module ×4
Scaled Dot-Product Attention
Multi-Head Attention
Q
K
V
Product
Scaling
Softmax
Linear
Concat
RNN
Fixed features
Access load
……
AMF
UPF
CU
Fixed features
Capacity
Latency
……
DU1
DU2
Fixed features
Computing resources
Location
Latency
……
Device s1
Device s2
Device s3
Light prediction 1
T1
Tn
n1
n2
n3
n4
n5
n6
n7
n8
1
τnow
PDlen
Accurate prediction (one turn)
Light predictions (multiple turns)
Hybrid predictions (multiple turns)
Time era 1
New features
Hierarchical relationship-based prediction 2
Intra-layer relationship extraction
Inter-layer combination
Light prediction 2
n9
n10
Time era 2 (new technology introduced)
…….

Figure 3. The process of proposed long-term hierarchical convolutional graph attention model (LTHCGAM).

The overall algorithm can be expressed as follows

$$\mathbb{S}_{n,i-1}, \mathbb{P}_{n,i-1}, \zeta_{n,i-1} = \mathbb{S}_{\text{O,CTHGAM}}(\mathbb{S}_{n\in\Psi}, \mathbb{P}_{n\in\Psi}, \mathbb{U}, \zeta_{n\in\Psi}, \tau^{\text{now}}_{n\in\Psi}) = \begin{cases} \text{GphAttention}\Big(\Big\{\text{MultiHead}_g\Big(\text{RNN}(\text{TA}((\mathbb{S}_{n,i-1},\mathbb{P}_{n,i-1},\zeta_{n,i-1}),\tau_n^{\text{now}}),\text{PD}_{\text{len},n})|_{n\in\Psi_g}\Big)\Big\}|_{g\in\mathbb{G}},\mathbb{U}\Big), \ \text{if } i = I_\gamma^{\text{start}}+1+\zeta_{n,\gamma}k\,, \\ \text{RNN}(\mathbb{S}_{n,i-1},\mathbb{P}_{n,i-1},\zeta_{n,i-1},\text{PD}_{\text{len},n}), \ \text{else} \end{cases} \tag{7}$$

where time era γ is determined based on $i \cdot L_n$, $I_{\gamma}^{\text{start}}$ is the iteration number of time era γ. In a time era, the predicting precedure iteratively go through one accurate predicting process (corresponding to the first condition) and multiple light but less-accurate predicting process (corresponding to the second and third condition). Next, the detail of the two processes are sequentially given as follows.

As for the accurate predicting process, first, predictions are input to time alignment $\text{TA}(\cdot)$ to align predicting time period for different network elements as follows

$$\begin{aligned} \text{TA}((\mathbb{S}_{n,i-1}, \mathbb{P}_{n,i-1}, \zeta_{n,i-1}), \tau_n^{\text{now}}) &= \text{FCN}(\mathbb{S}_{n,i-1}, \mathbb{P}_{n,i-1}, \zeta_{n,i-1}) + \left\{ \sin(\tau_n^{\text{now}}/10,000^{2j/d}) \right\}|_{j=1,\dots,d} \\ \text{TA}((\mathbb{S}_{n,i-1}, \mathbb{P}_{n,i-1}, \zeta_{n,i-1}), \tau_n^{\text{now}}) &= \text{FCN}(\mathbb{S}_{n,i-1}, \mathbb{P}_{n,i-1}, \zeta_{n,i-1}) + \left\{ \cos(\tau_n^{\text{now}}/10,000^{2j/d}) \right\}|_{j=1,\dots,d} \end{aligned} \tag{8}$$

which aims at adding different each time straps τ_n^{now} onto each element input vector $(\mathbb{S}_{n,i-1}, \mathbb{P}_{n,i-1}, \zeta_{n,i-1})$. Then, it input to hierachical relationship-based prediction module for accurate prediction.

$$\mathbb{S}_{n,i}, \mathbb{P}_{n,i}, \zeta_{n,i} = \text{GphAttention}\Big(\Big\{ \text{MultiHead}_g \Big(\text{RNN}(\text{TA}((\mathbb{S}_{n,i-1}, \mathbb{P}_{n,i-1}, \zeta_{n,i-1}), \tau_n^{\text{now}}), \text{PD}_{\text{len},n})|_{n \in \Psi_g} \Big) \Big\}|_{g \in \mathbb{G}}, \mathbb{U} \Big) \tag{9}$$

where $\text{RNN}(\cdot)$ dictate the historical state of each element, $\text{MultiHead}(\cdot)$ and $\text{GphAttention}(\cdot)$ aggregate the states from different elements to give the predicting results for each element on their next predicting time in τ_n^{now}.

As for light predicting process, the predicting result of the next predicting turn for each element is generated only based on its own historical data (more accurately the latest predicting data), expressed by

$$\mathbb{S}_{n,i}, \mathbb{P}_{n,i}, \zeta_{n,i} = \text{RNN}(\mathbb{S}_{n,i-1}, \mathbb{P}_{n,i-1}, \zeta_{n,i-1}, \text{PD}_{\text{len},n}) \tag{10}$$

3.2. Autonomous Large-Small Scale Strategy Generation

The “future shot” comprehensively demonstrates the network state, network strategies, and network performance across different time in future network. If network performance is suboptimal during a certain period, it becomes necessary to automatically generate improved network strategies to enhance performance. Here, network strategies encompass strategies of“large and small scales”. The word “scale” has two meanings, including the "scope" of its application and "magnitude" of the improvement approach. The former indicates whether it pertains to the entire network range managed by OAM, a specific local area within a RAN, or an individual network element node. The latter indicates whether it involves optimizing configuration parameters based on existing technology or making significant adjustments by adopting or introducing alternative technologies. The following provides a detailed description of the strategy generation process.

First, demand self-awareness is required based on the results of "future shots" to identify the time point when performance begins to degrade, determine which technology is needed to address it, and pinpoint the critical network element node for resolving the degradation. This is categorized into two scenarios based on the scope of application:

- Large-scope self-awareness: For example, it may be demand self-awareness on OAM, which can be implemented in three steps. First, it can identify a time point after which performance falls below a certain threshold. Then, it can use an LLM model to provide recommendations on which technology should be employed for optimization or evolution. Finally, it inputs the heterogeneous graph, formed by the heterogeneous performance of various network elements with their connecting topology, along with the optimization method suggested by the LLM, into a conditional-heterogenous graph neural network (Conditional-HGNN) to output a prioritized ranking of network nodes recommended for optimization, as depicted in Figure 4.
- Small-scope self-awareness: For example, it may be demand self-awareness on a single network element, implemented in two steps. First, it can identify a point in time after which performance falls below a certain threshold. Then, it uses an LLM to suggest which technology should be used for optimization or evolution.

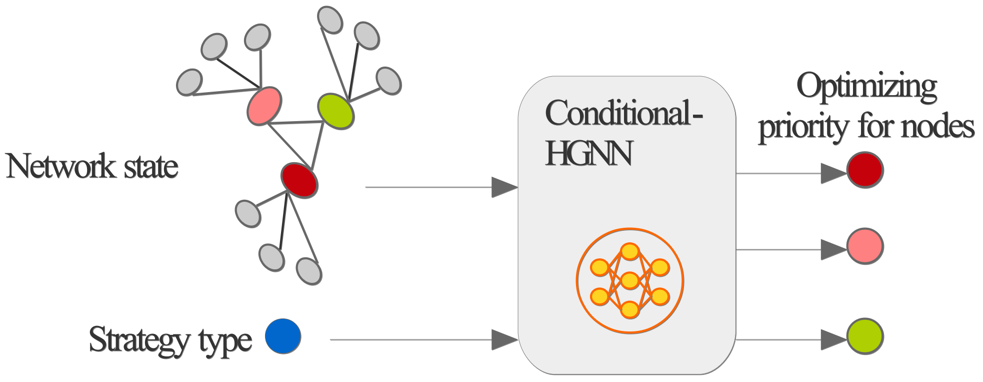

Figure 4. Generating optimizing priority for nodes using Conditional-HGNN.

Second, generate the initial state of the specific strategy, train the final configuration parameters, and deploy them to "future shots." This is divided into two scenarios based on the magnitude of the improvement approach:

- Large magnitude: If the improvement approach is significant and requires introducing new technology, it involves two steps. First, it initializes configuration parameters for the new technology. If a flexible, intelligent, and dynamic configuration scheme based on AI is needed, an LLM can be used to define the AI model's structure and initial parameters. If AI-based configuration is not required, a pre-set template can be used for initialization. Second, it performs adaptive optimization of the initial configuration. Concretly, it can builds a performance pre-validation environment based on the APIs of different network elements to train AI-based configuration strategies or optimize non-AI configuration schemes based on templates.
- Small magnitude: If the improvement approach is minor and only requires optimizing configuration parameters of existing technology, proceed directly with the second step above.

Note that if a "future shot" undergoes strategy optimization, subsequent "future shots" predictions must be adjusted accordingly. If the ultimous performance still cannot reach the requirement, the DT would inform operators to seek for solutions in advance, which would be dictated into database for suggesting optimizing strategy in future.

3.3. Large-small time Scale Controls

DTN manages the physical network across large and small time scales, leveraging the network strategies and performance data from "future-shots" to optimize and evolve the physical network in a layered manner. The concept of "scale" encompasses two key aspects.

Optimization and Evolution: NDT's autonomous control over the physical network often involves simultaneous small-scale optimization and large-scale evolution.

- On a small time scale, NDT continuously optimizes the physical network. By regularly updating the intelligent network AI algorithms used in the physical network, NDT enables them to timely adapt to the changing physical environments, ensuring precise and efficient strategies for resource allocation, mobility management, and other network operational states.
- On a large time scale, NDT facilitates the gradual evolution of the physical network. Many network evolution strategies require long-term, incremental progression and directional transitions, as they cannot be implemented all at once. For instance, a cell-free network might begin with pilot deployments in areas with fewer users, then progressively expand to densely populated regions until the entire network is fully upgraded.

Local and Global: NDT employs a centralized-distributed deployment architecture, where local control is suited for smaller time scale operations, and global control is tailored to larger time scale management. This requires cost-effective, multi-time-scale top-down control.

- At the smallest time scale, an network element DT is deployed alongside its physical counterpart, performing real-time status monitoring and closed-loop control of the equipment.
- At the largest time scale, a DT is deployed within an operation and maintenance administration system, enabling large time scale optimization and control across the entire network.

It needs to mention, effective "large-small" time scale control is only achievable by simultaneously addressing both "local and global" and "optimization and evolution." For example, within the DT of a single network element under "small time scale control", there may be two distinct time-scale controls: optimization and evolution. Take the DT of a gNB as an example—it not only needs to handle network optimization controls like resource allocation but also evaluate whether to introduce new technologies such as terahertz (THz) or reconfigurable intelligent surfaces (RIS) for iterative network evolution.

4. Use Cases

This section presents various application scenarios of DT-native network, encompassing large language models and AGI, transportation and driving, industrial internet, and low altitude airspace economy.

4.1. Large Language Models and AGI

With the continuous advancement of network intelligence, LLMs and Artificial General Intelligence (AGI) are being introduced into network operation and maintenance to address challenges such as the complexity of network management, achieving notable positive outcomes in some cases [9]. However, due to difficulties in obtaining network data and inconsistent data quality, LLMs and AGI may exhibit hallucination phenomena and produce inaccurate inferences and perception, thereby adversely affecting network performance or even causing systemic failures. Therefore, the pre-validation of intelligent decision feasibility is a critical step to ensure stable network operations.

In a DT-native network, the NDT, as a high-fidelity digital replica of the physical network, plays a vital role in addressing these challenges. Firstly, NDT can simulate arbitrary network states to generate large volumes of high-quality and training data for LLMs,effectively solving the generalization limitations. LLMs often struggle to adapt to diverse scenarios, especially unexpected or rare network events, due to a lack of diverse training datasets. By leveraging NDT, these models can train on simulated data that includes complex and abnormal network states, thereby significantly enhancing their adaptability and generalization performance. At the same time, NDT provides interactive and dynamic data specifically for AGI, enabling it to refine its understanding of complex network behaviors and improve its decision-making processes in a safe, controlled environment. This interactive capability also contributes to enhancing cognitive level of AGI, making it more robust and effective in handling real-world network operations. Secondly, NDT serves as an interactive training environment for LLM and AGI learning. It allows these models to interact with simulated network environments, replacing risky direct interactions with physical networks. During training, the feedback provided by NDT, such as performance metrics or simulated network responses, acts as reinforcement signals, helping LLMs continuously optimize their strategies and achieve faster convergence and improving AGI cognitive level. Lastly, NDT provides a critical validation platform for intelligent decisions generated by LLMs or AGI. Before deploying these decisions in physical networks, NDT allows for pre-configuration and simulation of their execution. If a decision proves to be infeasible during simulation, the model iterates further until an optimal and feasible strategy is achieved. Only then is the decision deployed to the physical network, ensuring minimal risk of performance degradation or network failure caused by erroneous decisions. This closed- loop validation process not only enhances the reliability of intelligent decision-making but also builds trust in the deploy- ment of LLMs and AGI across various network scenarios.

In summary, NDT provides interactive training environments and validates intelligent decisions, making it an indispensable tool for applying LLMs and AGI into network intelligence. By addressing key challenges such as data quality, model general- ization, and decision feasibility, NDT significantly accelerates the development and deployment of network intelligence, paving the way for more robust and trustworthy AI-driven network systems.

4.2. Transportation and Driving

Currently, the intelligent transportation and driving fields face numerous challenges, primarily manifested in the com- plexity of traffic scenarios. A traffic scenario is a multi-element system comprising interconnected components such as people, vehicles, roads, and environment. The construction of current hybrid traffic scenarios presents two main issues: insufficient consideration of road dynamic evolution's impact on mixed traffic and a lack of multi-person hybrid traffic perception methods.

Traditional on-site testing has significant limitations, struggling to reproduce extreme scenarios and comprehen- sively simulate complex factors like obstacle volume, angles, vehicle speed, lighting, weather, and sensor conditions, with certain hazardous scenarios (such as collisions or rollovers) being nearly impossible to directly collect data from. Actual data collection faces challenges including sample imbalance, scarcity of long-tail and sensitive high-safety domain data, and increasing difficulty in acquiring effective data as algo- rithm maturity increases.

DT-native network provides innovative solutions for autonomous driving testing, enabling high-precision reconstruction of traffic scenarios, simulating real natural environments (rain, snow, lighting), offering more diverse and balanced test scenarios, generating synthetic data to fill gaps in real data, and enhancing data diversity, complete- ness, and balance. Compared to traditional mileage testing, simulation testing is more economically efficient, capable of covering the most scenarios with the least mileage and shortest time, significantly improving testing efficiency. By integrating maps and digital twin technology, more realistic and accurate simulation environments can be created, mimicking real-world roads, traffic flows, signals, and signs, comprehensively con- ducting functional, performance, and safety tests, and enhanc- ing user experience.

As technological capabilities continue to be released and implemented, DT-native network will accelerate the arrival of Level 3 autonomous driving, improve human- machine collaboration capabilities, and realize better human- machine interaction and co-driving.

4.3. Industrial Internet

DT-native network, as an emerging technology, demon- strate enormous potential in the industrial Internet domain. By creating virtual replicas of physical entities, they enable real- time monitoring, analysis, and optimization of physical sys- tems, thereby improving production efficiency, reducing costs, and enhancing product quality. In the industrial Internet, DT-native network primarily cover five core application scenarios: equipment management and maintenance, production process optimization, product design and development, supply chain management, and safety production.

In equipment management and maintenance, NDT collect real-time equipment operational data through sensors, proactively identifying potential failures and predict- ing equipment failure times based on historical data and ma- chine learning models, thus achieving predictive maintenance. Simultaneously, by simulating equipment operation in virtual environments, they can effectively validate new maintenance proposals and reduce actual operational risks.

In production process optimization, NDT can simulate the impact of different production parameters on product quality and efficiency, precisely locate production bottlenecks, and rapidly respond to market demand changes, enabling flexible production. In product design and develop- ment, they support creating DT models for virtual testing and verification, significantly shortening product development cycles and promoting collaborative design across multiple teams. Within supply chain management, digital twin networks provide end-to-end visualization tracking, improve supply chain transparency, and optimize inventory manage- ment by predicting future demands based on data. In terms of safety production, they can identify potential risks, simulate accident scenarios, develop targeted emergency plans, and enhance emergency response capabilities.

The core advantages of DT-native network in the indus- trial Internet include significantly improving production effi- ciency, reducing operational costs, enhancing product quality consistency, strengthening data-driven decision-making capa- bilities, and creating favorable conditions for the research, de- velopment, and application of new technologies and products. This demonstrates their immense potential and strategic value as a key enabling technology in the industrial Internet.

4.4. Low Altitude Airspace Economy

One of the main challenges faced by the low-altitude economy is how to rationally allocate network resources to ensure effective scheduling [10]. The low-altitude economic scenario is complex and dynamic, involving various types of UAVs, different task requirements, and constantly changing environmental factors such as weather and terrain. Especially when multiple UAVs are working together, the dynamic changes in communication and computing resources make resource allocation and flight scheduling even more difficult. Traditional experience-based decision-making methods can no longer meet the rapidly changing demands, particularly in key tasks such as collision avoidance and airspace sharing, which may lead to resource wastage or flight safety risks.

To address these challenges, NDT provides significant sup- port for the low-altitude economy by advanced simulation and intelligent resource scheduling. Firstly, NDT can offer high-precision geographic information for the low-altitude economy through accurate low-altitude planning digital maps and sensing data. This information, through the grid space constructed by the NDT, transforms airspace management from continuous trajectory solving to discrete digital grid space probability prediction and control calculation, which plays a crucial role in flight path planning and collision avoidance. Secondly, NDT can reduce the dimensionality of large-scale traffic flow management optimization problems by employing grid-based traffic flow models and digital grid space relationship decoupling methods, thereby improving decision-making efficiency. Thirdly, NDT provides an interactive training environment, allowing different types of UAVs to interact in real-time in a simulated environment. Based on feedback, UAVs can optimize their task execution and resource scheduling strategies. Through this dynamic optimization pro- cess, NDT can provide real-time feedback on changes in the simulation environment, helping Intelligence Engines optimize decision strategies based on new information, while ensuring personalized communication, computation, and AI service support. This effectively reduces collision probabilities, energy consumption, and improves overall system service quality.

In conclusion, NDT offers significant advantages for the low-altitude economy by improving resource utilization ef- ficiency and ensuring the safety and sustainable operation of UAV systems. Additionally, NDT enables the efficient organization of low-altitude resources, driving the intelligent development of the low-altitude economy.

5. Conclusions

This paper designs a new architecture for DT-native network to realize 6G self-evolution. Specifically, a key concept of "future shots" is introduced into the architecture for predicting future states and validating evolution performance. Moreover, we propose a strategy generation method for giving network evolution strategy, a full-scale network prediction method with CTHGAM model and LTHCGAM model for predicting network states and strategy performance, and a hierarchical network interaction strategies for efficient virtual-physical interactions. Finally, we give four potential applications for DT-native network.

Funding: This work was funded by the National Key R&D Program of China (2022YFB2902100).

Informed Consent Statement: The study did not involve humans.

Data Availability Statement: Data can be found in the article.

References

1. Cui, Q.; You, X.; Wei, N.; Nan, G.; Zhang, X.; Zhang, J.; Lyu, X.; Ai, M.; Tao, X.; Feng, Z.; et al. Overview of AI and Communication for 6G Network: Fundamentals, Challenges, and Future Research Opportunities. *arXiv preprint arXiv:2412.14538* **2024**.

2. Cui, Y.; Liu, J.; Yu, M.; Jiang, J.; Zhang, L.; Lu, L. Network Digital Twin. *IEEE Network* **2024**, *38*, 5–6. https://doi.org/10.1109/MNET.2024.3352356.
3. Hakiri, A.; Gokhale, A.; Yahia, S.B.; Mellouli, N. A comprehensive survey on digital twin for future networks and emerging Internet of Things industry. *Computer Networks* **2024**, *244*, 110350. https://doi.org/https://doi.org/10.1016/j.comnet.2024.110350.
4. Jang, S.; Jeong, J.; Lee, J.; Choi, S. Digital twin for intelligent network: Data lifecycle, digital replication, and AI-based optimizations. *IEEE Communications Magazine* **2023**, *61*, 96–102.
5. Masaracchia, A.; Van Huynh, D.; Alexandropoulos, G.C.; Canberk, B.; Dobre, O.A.; Duong, T.Q. Toward the metaverse realization in 6G: Orchestration of RIS-enabled smart wireless environments via digital twins. *IEEE Internet of Things Magazine* **2024**, *7*, 22–28.
6. Lai, J.; Chen, Z.; Zhu, J.; Ma, W.; Gan, L.; Xie, S.; Li, G. Deep learning based traffic prediction method for digital twin network. *Cognitive Computation* **2023**, *15*, 1748–1766.
7. Cheng, N.; Wang, X.; Li, Z.; Yin, Z.; Luan, T.; Shen, X. Toward Enhanced Reinforcement Learning-Based Resource Management via Digital Twin: Opportunities, Applications, and Challenges, 2024, [arXiv:eess.SY/2406.07857].
8. Li, Q.; Zhang, Y.; Wang, L.; Duan, X.; Lu, L.; Hu, Y.; Zhan, Y.; Lian, Y.; Sun, T. Native Network Digital Twin Architecture for 6G: From Design to Practice. *IEEE Communications Magazine* **2024**, pp. 1–8. https://doi.org/10.1109/MCOM.001.2400216.
9. Xu, M.; Niyato, D.; Kang, J.; Xiong, Z.; Mao, S.; Han, Z.; Kim, D.I.; Letaief, K.B. When Large Language Model Agents Meet 6G Networks: Perception, Grounding, and Alignment. *IEEE Wireless Communications* **2024**, *31*, 63–71. https://doi.org/10.1109/MWC.005.2400019.
10. Zhang, H.; Wu, S.; Feng, O.; Tian, T.; Huang, Y.; Zhong, G. Research on Demand-Based Scheduling Scheme of Urban Low-Altitude Logistics UAVs. *Applied Sciences* **2023**, *13*. https://doi.org/10.3390/app13095370.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.