Towards the Development of 6G System Level Simulators: Addressing the Computational Complexity Challenge

Marvin Manalastas, Muhammad Umar bin Farooq, Syed Muhammad Asad Zaidi, and Ali Imran
AI4Networks Research Center, University of Oklahoma, Tulsa, OK, USA
Email: {marvin, umar.farooq, asad, ali.imran} @ou.edu

***Abstract*- The advancement and standardization of the cellular network system require thorough investigation, analysis, and experimentation of novel protocols, architectures, and functionalities. In this regard, computer-aided tools or simulators allow the execution of these requirements with the much-needed controllability, reproducibility, cost efficiency, and convenience. Simulators have been proven beneficial since the dawn of the cellular network era and are likely to be critical for the upcoming 6G development. However, mimicking such a complex network requires developing intricate and at the same time, practical simulators. One of the major concerns of the existing simulators is the computational complexity of modeling a realistic and complete network. This challenge is anticipated to exacerbate with the advent of 6G considering its scope and peculiarities. In this article, we analyze the computational complexity of incoming 6G simulators and provide solutions to mitigate this issue. The presented novel framework for future simulators aims to transform the traditional way of building network simulators to serve the unprecedented demand of 6G. The presented use case highlights the efficacy of the proposed framework where we show a 100-fold improvement in the run-time performance of the innovative architecture compared to traditional simulators.**

I. INTRODUCTION

The role of simulators has become increasingly important as cellular technology evolves to support more novel use cases. These simulators accelerate innovation and reduce the cost of research and development towards next generation networks for researchers, network vendors, and operators alike. This is particularly true with the upcoming 6G network. For instance, up until 5G, one side of the network has always remained static, i.e., base station, while serving mobile users. However, in 6G, the network nodes are anticipated to become mobile as well, i.e., satellites and drones, bringing a totally different level of complexity in modeling the network. With this novel network deployment, analytical modeling, i.e., point processes and stochastic geometry, which work in 5G and other legacy network deployments with static elements, might not remain insightful anymore. Thus, it can be concluded that simulators will play even more crucial roles in modeling and simulating key technologies and components of the upcoming 6G network.

Another major distinction of 6G vis-à-vis 5G is the ubiquitous use of Artificial Intelligence (AI). Unlike in 5G where AI is still an option, AI is envisioned to become a fundamental part of 6G playing pivotal roles in the entire 6G ecosystem. However, the utility of the AI models is inherently reliant on rich training data. However, real network data for training AI models is currently sparse and scarce due to several reasons including privacy concerns, high cost and potential degrading impact on live networks of any data gathering campaigns, among others. For a detailed review of sparsity challenge in cellular networks, reader is referred to [1]. The challenge of AI training data sparsity in cellular networks further highlights the crucial function of simulators in future cellular networks. Using simulators, synthetic data can be generated to enrich the sparse data from the real network that can be used to effectively train the AI models. However, large data generation through simulators can become time-consuming if not accompanied with computationally efficient methods. In this article, we address one of the most notorious challenges that debilitate the utility of simulators for the next generation networks; the inherent computational complexity brought by the goal to make simulators realistic and complete. The computational and time efficiency becomes increasingly crucial when huge data is required to enable AI in 6G networks.

The contributions and organization of this paper can be summarized as follows: We first present a short look ahead on what 6G is anticipated to look like and explicate the challenge of computational efficiency, analyze how it currently affects 5G, and the future implications in developing 6G simulators (Section II). We then present potential solutions and recommendations to address these computationally demanding aspects of network simulation and combine these solutions in an architecture to enhance the computational efficiency of future simulators (Section III). This proposed architecture is computationally efficient without compromising realism and completeness of the functionalities and features. Then, we evaluate the potential of the proposed simulator architecture against typical simulator designs in generating large amounts of data to effectively train AI-models (Section IV). In Section V, we present the key conclusions and insights of this work.

II. THE CHALLENGE OF COMPUTATIONAL EFFICIENCY IN FUTURE MOBILE NETWORK SIMULATORS

A. *6G Anticipations: A Look Ahead to Next Generation Networks*

Although no standard has yet been crafted for 6G, based on the current trend, it is not difficult to anticipate where 6G is heading and what will it look like. At this early point in time, several studies are available in the literature ranging from the anticipated architecture, potential applications and use cases, and the expected enabler technologies to support 6G [2-4]. In terms of the architecture, one of the most evident distinctions of 6G from 5G is the utilization of diverse network types in the form of 3D networking. In addition, 6G is likely to sustain a much broader range of use cases and applications compared to 5G. Some of the most notable novel applications are flying cars, the internet of everything, multisensory extended reality (XR), and wireless brain computer interaction. To realize these use cases, the next generation networks will leverage novel sets of enablers such as ultra-massive MIMO, Thz band, quantum communication, blockchain, optical wireless technologies, and reflective surfaces.

In the wake of the anticipations, it is evident that 6G requires trial, evaluation, and validation procedures, that are more flexible and extensive than they have ever been. As a result, the simulators to support the timely development of the incoming 6G network deserve more attention. Particularly, modeling the above-mentioned architecture, use cases, and enablers of 6G realistically and completely brings unprecedented impediments to the computational efficiency of next generation simulators. In the next subsection, we discuss the foreseen challenge of computational complexity with reference to 5G network simulators.

B. *Computational Efficiency as a Bottleneck in Simulator Development*

The computational efficiency requirement for 6G simulator conflicts with the goal of creating the simulator as realistic and complete as possible. All three of these are not only desirable features for research and development of 6G systems but are key necessities for generating synthetic data to train AI for enabling zero-touch operation and optimization in 6G networks. Fig. 1 shows the relationship between realism, completeness, and computational efficiency of mobile network system level simulators. Ideally, a simulator should incorporate a high degree of realism, completeness and at the same time, utilize a small amount of computational resources in terms of time, memory, and processing power. Realism in this context is measured by the degree of realistic implementation of features such as propagation model, mobility model, PHY layer model, radio access network procedures, and core network model to name a few. Meanwhile, completeness refers to aspects such as the complete implementation of technology enablers, comprehensive incorporation of configuration and optimization parameters (COPs) and key performance indicators (KPIs), and integration of a wide range of use cases and applications. However, with the current approach of simulator development, computational efficiency deteriorates rapidly with the increase in realism and completeness. To avoid unbearable computational costs, most of the existing simulators tend to employ more abstraction and oversimplification of the computationally intensive tasks and incorporate fewer features and functionalities, thereby compromising realism or completeness or both. Some well-known academic simulators, such as ns-3 [5] and Vienna [6], have managed to render a moderate amount of realism and completeness, but at the cost of high computational costs. On the other hand, commercial simulators such as Atoll [7], are usually utilized in network planning and thus, give particular attention to realistically modeling the propagation model. But to improve the computational efficiency, these commercial simulators usually eliminate the support for mobility or handover and usually implement only a handful of COPs and KPIs thereby offering little completeness. Despite the good intention, the simplification and downscaling of the nuances during simulator development to induce efficiency in the run time may lead to undesirable results.

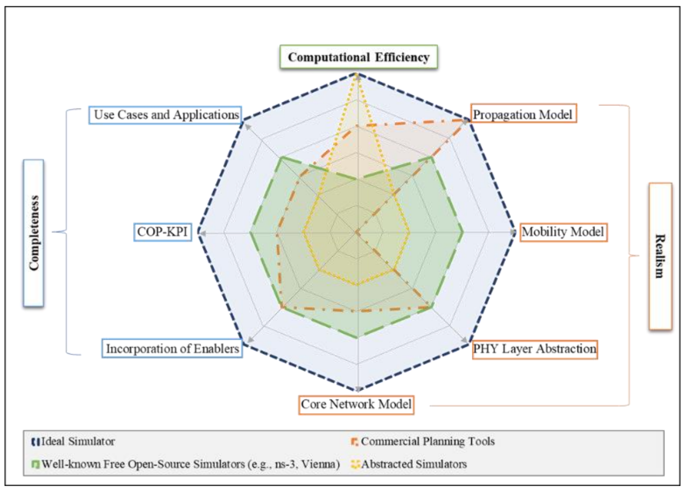

Figure 1. Relationship between realism, completeness, and computational efficiency in mobile network simulators

C. *Computational Demanding Aspects of Running Simulations*

Table 1 shows the summary of some of the most computationally demanding tasks in simulating a cellular network. Among these components are the propagation model,

Computationally Challenging Components	Details	5G network	6G network
Propagation Model	As operating frequency increases, more parameters in propagation modeling must be considered to improve accuracy (i.e., type of clutter, vegetation, buildings, people, and cars), making it computationally expensive.	mmWave	THz band
MIMO Beam Management	The increased computational complexity introduced by the increased number of antennas used to achieve MIMO is related to the increased number of precoding matrices required to generate during the simulation.	Massive MIMO (16x16 to 64x64 antenna arrays)	Ultra-massive MIMO (1024x1024 antenna arrays)
Mobility	The constant variation in the location and environment as the users move leads to continuous updating of the signal strength during simulation.	2D mobility	3D mobility
Number of network elements and connected devices	The size of network elements such as base stations, connected devices, configuration parameters, measured KPIs, and supported features directly affects the simulation run time.	BS Parameters: 2500+ User density: 10^6 devices/km^2	BS Parameters: 3000+ User density: 10^7 devices/km^2
AI-model training	Training AI-based model is computationally expensive depending on scale of the model (i.e., model depth), and size of the data.	Arbitrary	Compulsory
Reconfigurable Intelligent Surfaces	Calculation and delivery of all active beamforming weights and phase shifts to the associated nodes.	N/A	Among the enablers
Network Deployment	The computing requirements for the aforementioned components exacerbate with the introduction of non-traditional network deployment.	Terrestrial (2D Networking)	Terrestrial + Aerial (3D Networking)

Table 1. Summary of some of the most computationally demanding aspects when running a network simulator, how it affects simulation run time in simulating 5G network and the future implications on the upcoming 6G networks.

MIMO precoding, mobility, network element size, compulsory use of AI, and reconfigurable intelligent surfaces. Moreover, this table also shows how these components are affecting the current 5G simulators and more importantly, their implications to the upcoming 6G technology.

Selected computationally challenging components for running simulations are explored in further detail below:

1. *Propagation Model*

   The increasing factors that must be considered in the propagation model to increase the accuracy (i.e., clutter type, vegetation, buildings, etc.) make it computationally expensive. In mmWave used for 5G, factors such as building materials absorption, vegetation, vehicles and even the effect of humans as blockage should be considered which further increases the complexity of the process. The incoming 6G will require an unprecedented amount of sophistication when it comes to propagation modeling. This is due to the anticipated utilization of the THz band which, compared to the GHz band used in 5G, is more demanding in terms of Line of Sight (LOS) requirements and is highly susceptible even to the slightest obstruction (i.e., a sheet of paper) [4]. To realistically model these peculiarities of the THz band requires considering more factors and thus, would require immense computing resources.

2. *MIMO Precoding/Beam Management*

   The multiplicity of the utilized antennas to realize MIMO brings additional complexity due to an increase in the precoding matrices needed to be generated during the simulation. The available antenna configurations for 5G massive MIMO range from 16x16 to 64x64. With this much antenna, precoding matrices generation consumes more time when running simulations. The concept of ultra-massive MIMO is proposed for 6G, wherein a plasmonic nano antenna array of size 1024x1024 is envisioned [2]. With this huge antenna configuration, the generation of precoding matrices will be daunting for the simulators.

3. *Mobility*

   The constant variation in the location and environment as the users move leads to continuous updating of the signal strength during simulation. In addition, mobile users need to perform handover which is also computationally demanding. 5G is designed to support mobile users with a maximum speed of up to 500 kph. Calculation of the changes in the signal strength especially for high-speed users is an expensive task. Moreover, due to speed as well as the dense base station deployment, number of handovers increases. The maximum speed at which 6G will be molded is 1000 kph, making the calculation of the changes in the received signal more daunting.

4. *Huge Number of Network Elements and Connected Devices*

The number of network elements such as base stations, users, types of services, connected devices, configuration parameters, measured KPIs, and supported features directly affect the simulation run time. A recent article [8] highlights this dramatic increase in the simulation execution time with an increase in the network size. Based on this article, the simulation time showed a non-linear surge in time complexity increasing from less than 100 seconds for 1000 nodes to more than 2000 seconds for 3000 nodes scenario. Meanwhile, it is estimated that the connectivity density in 6G will reach far beyond 10^6 km^2 which is the density threshold for designing 5G [2]. With dense BS deployment supporting diverse types of users, use cases, and billions of connected devices, computationally inefficient simulators will fail to mimic the vastness of the 6G ecosystem.

5. *Network Deployment*

Meanwhile, one of the major game-changers that set 6G apart from its predecessors is the introduction of 3D networking. The conglomeration of aerial and terrestrial base stations is expected to shift the way we analyze and simulate cellular networks. 3D networking is anticipated to bring a whole new level of complexity in simulating the network and making the aforementioned challenges even more daunting. For instance, the wide adaptation of drones will give birth to 3D handover. This, in turn, results in more sophisticated user mobility and handover model implementations.

III. PROPOSED ARCHITECTURE TO ADDRESS THE COMPUTATIONAL COMPLEXITY OF NEXT GENERATION NETWORK SIMULATORS

The usability and performance of a simulator are severely undermined if it fails to immediately test the use cases and generate results rapidly. Therefore, the new design, protocol, or algorithm evaluation, as well as data generation through such a simulator can become time-consuming if not accompanied by computationally efficient methods. Currently, several methods to speed up the simulation process for 5G networks are being leveraged. For example, several simulators such as Atoll, OMNET++ [9], MATLAB-Simulink [10], and SiMoNe [11], exploit parallel processing to improve the run-time of simulations. Similarly, other simulators attempted to explore simulation speedup options to make up for high computational complexity either due to a large number of network elements, realistic modeling, or completeness. For instance, Vienna 5G has pre-generated channel traces while ns-3 tries to perform link-to-link computation in parallel. Meanwhile, WiSE incorporates MIMO precoding matrices pre-generation and smart beam sweeping link selection, in addition to parallel processing, to improve time efficiency [12].

Although some efforts are made to improve the computational efficiency of simulators, our extensive analysis of existing simulators shows that the bulk of the computational load is caused by their object-oriented architecture, where extensive iterative functions (e.g., for-loops) are used in each transmission time interval (TTI) to calculate KPIs. Taking this into consideration, there is a call for a major shift in designing a simulator that minimizes the use of iterative functions. In addition to the above-mentioned approaches, we highlight some key techniques for addressing computational efficiency while maintaining a high level of realism and completeness below.

1) *Pre-generation and Preloading of COP dependent KPIs:* The performance of cellular networks, measured in terms of Key Performance Indicators (KPIs) may vary depending on configuration and optimization parameters (COPs) settings. COPs are the backbone of any cellular network system. These tunable COPs, depending on the set values, affect how the network performs. For instance, COPs such as tilt, azimuth, tower heights and azimuth determine several KPIs like coverage and reference signal quality that do not change with time-variant factors such as user mobility or channel variations but only with respective COP values. Therefore, these types of KPIs can be pre-calculated even before the start of the simulation. Meanwhile, other KPIs vary not only with COPs but also with time variant factors. Examples of KPIs include throughput, handover success rate and quality of service and quality of user experience. Although these KPIs are also affected by COPs, their actual values cannot be predetermined unless the simulation is started, and users start to move around the network and request resources. Having this knowledge, instead of calculating all KPIs in each TTI, simulator design can divide KPI modeling into two categories: time-dependent and COP-dependent. All COP dependent KPIs such as reference signal received power (RSRP), reference signal received quality (RSRQ), geometric signal-to-noise-plus-interference ratio (SINR) also known as G-Factor, and slow shadowing can be pre-calculated only once at the beginning of the simulation thus incurring only a small computation cost despite using extremely realistic models. These pre-generated KPIs can then be pre-loaded at the start of the simulation process. This leaves only time dependent KPIs such as instantaneous user SINR, physical resource block (PRB) usage, handover evaluation metrics, and throughput to be calculated in each TTI.

Nonetheless, generating such voluminous amounts of pre-generated COP-KPI data would need huge storage space, particularly if flexibility in the simulated scenario, e.g., in user mobility, is desired. To overcome this issue, cloud storage may be a viable alternative. Instead of keeping the pre-generated data on a local system, they can be saved in a data lake on the cloud. This data lake will include a rich collection of network scenarios and will be expanded further as new scenarios are executed. When needed, the pre-generated network data can be readily retrieved from the data lake. Given that storage is usually cheaper than processing power, and not as constrained resource in simulators as the simulation run time, this solution can offer desired degree of trade-off between simulator's use case flexibility, storage capacity and run time.

2) *Modeling Mobility Innovatively Through Binning:* In current simulators, especially those which are developed to provide a very high level of coverage prediction fidelity such as Atoll, mobility modeling is avoided as it is computationally very expensive, i.e., the next location of each UE must be calculated in each TTI and all the KPIs must be recalculated with respect to the new location of the UE. However, being the raison d'etre of mobile cellular networks, mobility is an essential feature of mobile networks and must be considered to achieve holistic performance evaluation and optimal design that consider mobility related KPIs and not only static coverage predictions. Particularly optimal mobility management's contribution to system performance will increase drastically in 6G wherein ultra-high user speeds of up to 1000 km/h are anticipated to be supported. To model user mobility while eliminating the associated computational cost, the binning approach can be leveraged. The key idea here is to divide the network area into bins (cubes in the case of a 3D network) and to model COP-dependent KPIs with respect to spatial bins in the network instead of the UE locations. Thus, pre-calculated COP-dependent KPIs, as explained above, can be used in each TTI for the bins to which the user is associated. This approach eliminates the need for recalculation of all KPIs in each TTI even with a large number of mobile users. Secondly, the binning approach will require recalculation of UE-specific KPIs only when the UE changes bin location and not at every TTI.

3) *Leverage AI to Model Computationally Intensive Tasks:* Pathloss calculation is another computationally expensive task if modeled realistically, e.g., using raytracing. For instance, Atoll, by utilizing the ray-tracing model, yields far superior fidelity in coverage prediction than Vienna and ns-3, which use simple empirical and hence unrealistic propagation models. However, to offset the huge computational cost of raytracing's, Atoll omits all dynamics in the simulation, such as dynamic PRB allocation and scheduling, as well as user mobility and detailed handover procedures and signaling. Therefore, Atoll, while being very powerful for cell planning, cannot be used for research and development of realistic mobile networks or optimization of any of the dozens of mobility related KPIs and COPs. In other words, the Atoll architecture, like many other simulators, trades realism in certain aspects for incompleteness to keep the computational cost low. To address this tradeoff in a more optimal fashion, the potential of AI is leveraged to model computationally demanding tasks such as radio propagation. This idea is recently demonstrated in a study [13] that shows it is possible to leverage machine learning to achieve comparable accuracy to raytracing in propagation modeling. Results from this recent study show that Light Gradient Boosting Machine (LightGBM)-based propagation model outperforms all empirical models (e.g., used in Vienna and ns-3) in terms of accuracy whilst being 12x faster than raytracing used in Atoll. Similarly, authors in [14] propose a practical and accurate channel estimation for cell-free mmWave Massive MIMO framework based on the fast and flexible denoising convolutional neural network.

4) *Enabling Parallelization and Distributed Processing:* Although several existing simulators already support parallel and distributed processing, the majority of the existing simulators remain for-loops-based operations that do not allow parallelization. To enable parallelization, the conventional for-loop approach should be avoided by replacing them with paralleled matrix manipulations for most calculations. For example, interference calculation per PRB in each TTI can be modeled as a parallel-able matrix operation. In addition, the pre-loading will enable parallel processing of different COPs for COP-KPI combinatorial exploration and hence, will reduce the time complexity of COP-KPI data generation with higher computational resources.

5) *Build on a computationally efficient platform (i.e., Python):* Most of the existing simulators are built in either C++ or MATLAB. For example, ns-3 is a C++ based simulator, but it requires a highly C++ specific skill set. The need for an extremely experienced and skilled workforce hinders the use of ns-3 to some extent despite its ability to utilize the high-performance computing power of C++. On the other hand, MATLAB-based simulators are relatively easier to learn, but open-source MATLAB lacks the high-performance computing power to fully utilize the available resources. This limitation makes the MATLAB-based simulators such as Vienna, Simulink, and C-RAN, relatively slow. However, Python possesses both the advantages of C++ and MATLAB; it is open source and utilizes high-performance computing like C++ and it is

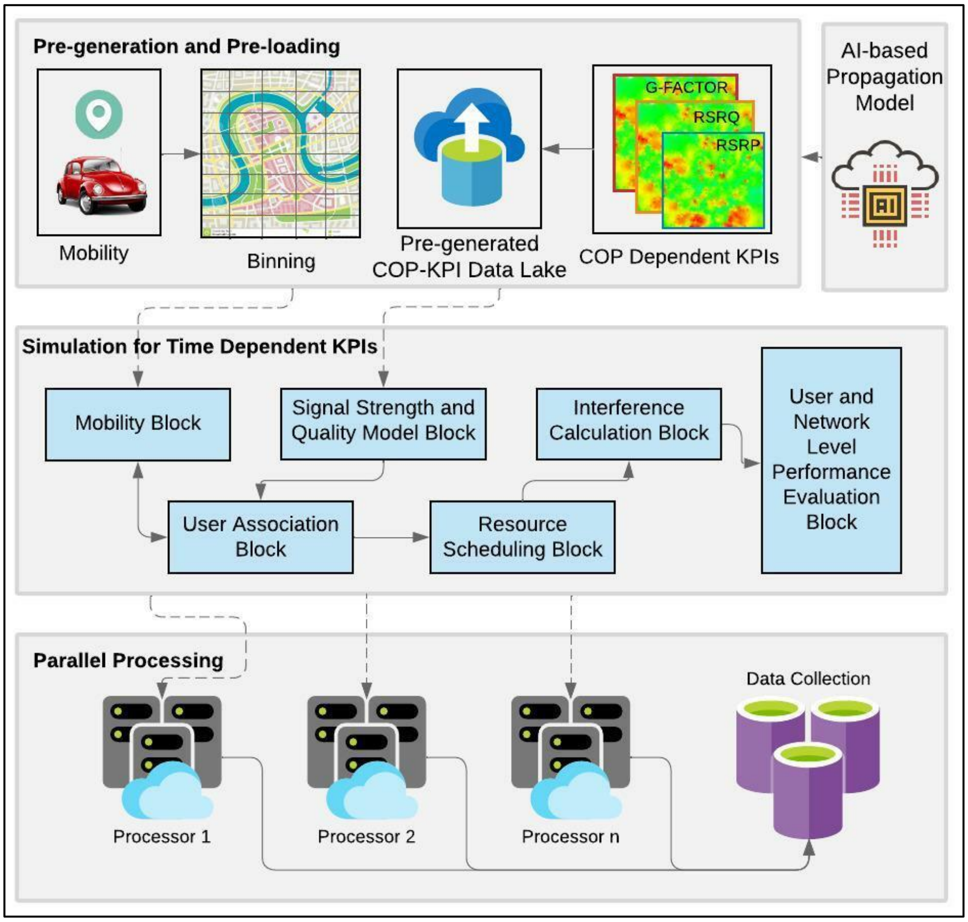

Figure 2: Proposed Python-based simulator architecture to enhance the computational efficiency of future simulators.

easier to learn like MATLAB. In addition, Python has a plethora of AI and DL libraries, which are far richer than those of C++ and MATLAB. This capability makes Python a better choice for the development of a system level 6G simulator as it will enable easier integration and testing of AI solutions on emerging networks.

To accelerate the simulation time regardless of the size of the network, with high degree of realism and completeness, and to generate a large amount of dataset in a short amount of time, the approach towards developing future simulators demands an overhaul. In this regard, next generation simulators will require leveraging innovative simulator architecture, parallel, and distributed processing capabilities as well as time and computationally efficient modeling of time complex functions like mobility and propagation. Fig. 2 illustrates the proposed architecture and techniques to improve the computational efficiency of future simulators. The goal of this paper is not to list all the possible novel techniques to implement all the idiosyncrasies of 6G in a simulator or to address all the challenges that may arise therein. Instead, the emphasis of this paper is on presenting and addressing only the core challenge in building a 6G simulator, i.e., the computational complexity problem, which if not addressed can become the bottleneck. More specifically, we propose and present a simulator architecture that can leverage innovative techniques to address the computational complexity in developing system level simulators for future cellular networks. Implementation of other 6G-specific features and use cases will become feasible only if this core challenge is addressed first (e.g., by adapting the computationally efficient architecture proposed in the paper) at the very early stages of 6G simulator design. In summary, this framework aims to investigate and develop novel techniques to expedite simulation time, which include: 1) a novel architecture leveraging the fast and efficient matrix implementation instead of for loop-based structure; 2) reducing the number of computational and time-hungry calculations of user mobility traces, RSRP, RSRQ, and SINR in each TTI; 3) utilizing ML-based pathloss models with significantly less time complexity than ray-tracing based pathloss model; 4) exploring innovative ways to model user mobility in a large cellular network; 5) fully utilizing the powerful parallel processing capabilities on one client without compromising the quality of the data; 6) proposing a decentralized simulator architecture to integrate multi-client data generation.

IV. CASE STUDY: EFFICIENT GENERATION OF SYNTHETIC NETWORK DATA FOR TRAINING AI MODELS

To quantify the advantage of the proposed simulator architecture over the state-of-the-art architectures, this paper examines the data generation time in three types of simulators: (1) a legacy simulator, (2) a simulator with parallel processing, and (3) our proposed architecture-based simulator i.e., a simulator that leverages binning and pre-generation, paired with parallel processing to reduce run time/online computational complexity. Specifically, the use case demonstrates the increase in the simulation runtime as the number of nodes (i.e., connected devices) increases and quantifies the potential of the proposed architecture to address this issue. As 6G is anticipated to support a greater number of connected devices, the case study analyzes the capabilities of various simulation techniques/architectures to accommodate a high user density in terms of simulation completion time. In addition, the use case highlights the effectiveness of the proposed solution in generating data for more effective machine learning model training. This is accomplished by evaluating the performance variation of AI solutions trained on data generated using different simulation strategies.

A. *Simulation Setup*

We exploit a 3GPP-compliant state-of-the-art system-level simulator named SyntheticNet [15] to validate our proposed simulator architecture. SyntheticNet is a modular, flexible, and versatile simulator supporting advanced features like adaptive numerology, handover, and futuristic database-aided edge computing to name a few. In this article, we have created three versions of SyntheticNet. The first version is the legacy simulator, which has none of the aforementioned innovative approaches implemented in it to improve the computational efficiency. In the second version, we equip off-the-shelf SyntheticNet with the ability to perform parallel processing similar to the current state-of-the-art approach of other simulators such as OMNET++, Simone, MATLAB-Simulink, and Atoll. In this version of the simulator, we allocate 20 cores capable of running the simulation in parallel. Finally, in the third simulator version, we implement the proposed innovative approaches to reduce computational complexity such as pre-generation and preloading of COP-dependent KPIs and mobility modeling through binning. In addition, this version is also capable of performing parallel simulations.

For the first part of the use case, we run the three versions of the simulator using similar settings shown in Table 2. To see the impact of varying the number of network elements in the simulation runtime, we vary the number of users from 100 to 2000. Each simulator version is run to generate data equivalent to 15s of real network data. The generated data is composed of several combinations of COPs: A3-Offset, A3-Time to trigger (TTT), A2-Threshold, A2-Time to trigger (TTT) with the corresponding KPI (Throughput) for each of the combinations. In total, each simulator version generated data of around 3,575 combinations of COP-KPI. Meanwhile, for the second part of the use case, each of the simulator versions is run for 1hr with 100 users. After this period, simulations are stopped, and the data generated are gathered to train AI models.

Table 2. Network simulation settings.

Parameters	**Values**
Number of Base Stations	100
Number of Users	100, 200, 500, 1000, 2000
Number of COP-KPI combinations	3,575
COPs	A3-Offset, A3-TTT, A2-Threshold, A2-TTT
KPI	Throughput
A3-Offset	[0, 1, 2, 4, 5, 6, 7, 8, 9, 10] dB
A3-TTT	[64, 128, 256, 512, 640] ms
A2-Threshold	[-95,-97,-99,-101,-103,-105,-107, -109,-111,-113,-115,-117,-119] dB
A2-TTT	[32, 64, 128, 256, 512] ms
Simulation Time (Data Generation)	15 s

B. *Run-time Comparison and Analysis*

The first set of results is the comparison of the total run time of the three simulator versions shown in Fig. 3. As expected, for all versions of the simulator, the run time increases as the number of users increases. More importantly, results reveal that as the user density increases, traditional simulator requires exponentially higher time to complete the simulation. From almost 100 hours for 100 users, the run time grows to more than 2166 hours. This means the legacy simulator architecture cannot be scaled to simulate 6G networks. Meanwhile, a simulator with parallel processing demonstrates it can better deal with the higher user density compared to the legacy simulator. Since the simulations are run in parallel using 20 cores, the simulation time is reduced by 20x the original. Although parallel processing showcased the ability to cut the runtime, the simulator with proposed innovations surpasses its performance. With a maximum of 2000 UEs, it takes only 21.7 hours for the simulator to generate the data. In summary, the performance of the proposed simulator architecture is 5x better than the state-of-the-art parallel processing approach and 100x better than most legacy simulators that do not allow parallel processing. These results demonstrate the potential of the proposed architecture to simulate networks with high user density scenarios that are hallmark of 6G.

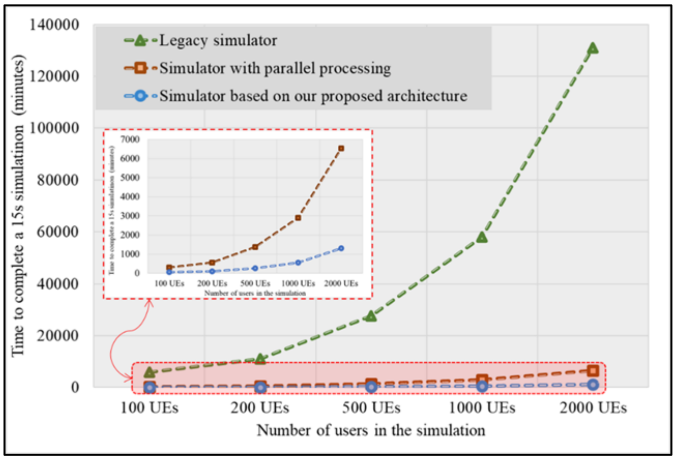

Figure 3. Run-time comparison between simulators with and without innovation to improve computational efficiency.

C. AI Model Training Analysis

One of the potential applications of AI in 6G is the prediction of network behavior. AI models have the capability to model and map out functions that cannot be directly or mathematically interpreted in the data. With available data, these models give insights into the traffic patterns and network behavior. However, to effectively train AI models, a representative amount of data is needed. In this use case, we analyze the effect of data sparsity caused by the inefficient data generation capability of traditional simulators. We build models that can map out the relations between mobility-related parameters, i.e., A3-Offset, A3-TTT, A2-Threshold, A2-TTT against certain KPI, i.e., throughput. The performance of the model is measured by the root mean square error (RMSE) metric. Lower values of RMSE correspond to a well-trained model, while large values indicate insufficient training. We train and evaluate several AI models, namely Linear Regression, Polynomial Regression, Support Vector Regression (SVR), Decision Tree, Random Forest, and XGBoost using the data generated by the three simulator versions.

The second set of results shows the effect of the simulators' capacity to generate the required amount of data needed to train machine learning models. As legacy simulators are inherently slow, they generate the least data points in 1 hour runtime (50 combinations of COP-KPI data), followed by the simulator with only parallel processing (700 data points). Lastly, due to its computational efficiency, the proposed simulator generates more than 3500 data points in 1 hour. We use the datasets generated by the three simulation techniques separately to train AI models for throughput prediction and compare the performance in Fig. 4. It can be observed that the models trained on the data generated by the proposed simulator architecture have the lowest RMSE. The average RMSE of the models trained with a large amount of data using the proposed architecture is 8.87 kbps, better than the 9.38 kbps and 12.90 kbps average RMSE of legacy and only parallel processing-capable simulators, respectively. This implies that the proposed architecture-based simulator is more capable of enabling the training of AI models for existing (i.e., 5G) and emerging complex cellular networks such as 6G. This further highlights the utility of the proposed architecture for various R&D use cases for 6G, particularly for aiding data driven modeling and AI-based network optimization.

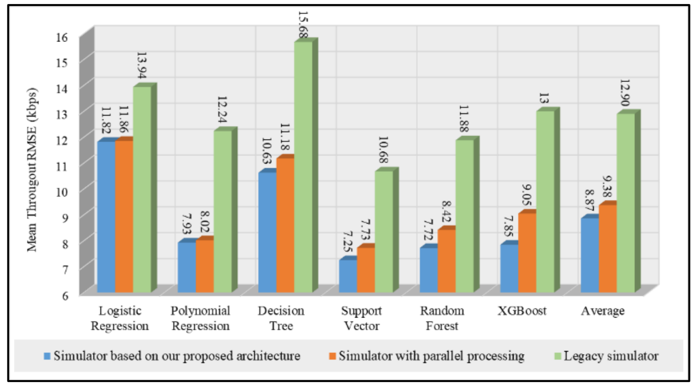

Figure 4. Comparison of AI models trained with different amounts of data generated from the legacy simulator, parallel processing-capable simulator, and simulator based on our proposed architecture.

V. CONCLUSION

In this article, we analyze the computational complexity challenge in simulator development for future 6G networks and discuss the root cause of this challenge. We analyze some of the most computationally expensive tasks in running simulations in the current 5G network and map them out to anticipate how they will affect the future 6G network. We then explicate innovative potential solutions to address this issue. These solutions include pre-generation and preloading of COP dependent KPIs, modeling mobility innovatively through binning, leveraging AI to model computationally intensive tasks (i.e., propagation model), enabling parallelization, and building on a computationally efficient platform (i.e., Python). Utilizing these solutions, we propose a novel Python-based simulator architecture to transform the simulator's computational efficiency. We evaluate the efficacy of the proposed architecture by presenting a use case. The results show not only the superiority of the proposed architecture against the state-of-the-art approach not only in terms of higher computational efficiency but also as a better enabler for AI-based model training using synthetic data.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation under Grant Number 1730650 and 1923669. The statements made herein are solely the responsibility of the authors. For more details about these projects please visit: http://www.ai4networks.com.

REFERENCES

[1] H.N. Qureshi, et al., “Towards Addressing Training Data Scarcity Challenge in Emerging Radio Access Networks: A Survey and Framework” submitted to *IEEE Communications Surveys and Tutorials*

[2] M. Z. Chowdhury, M. Shahjalal, S. Ahmed and Y. M. Jang, "6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions," in *IEEE Open Journal of the Communications Society*, vol. 1, pp. 957-975, 2020, doi: 10.1109/OJCOMS.2020.3010270.

[3] T. Huang, W. Yang, J. Wu, J. Ma, X. Zhang and D. Zhang, "A Survey on Green 6G Network: Architecture and Technologies," in *IEEE Access*, vol. 7, pp. 175758-175768, 2019, doi: 10.1109/ACCESS.2019.2957648.

[4] F. Tariq, M. R. A. Khandaker, K. -K. Wong, M. A. Imran, M. Bennis and M. Debbah, "A Speculative Study on 6G," in *IEEE Wireless Communications*, vol. 27, no. 4, pp. 118-125, August 2020, doi: 10.1109/MWC.001.1900488.

[5] Riley, G.F., Henderson, T.R. (2010). The ns-3 Network Simulator. In: *Wehrle, K., Güneş, M., Gross, J. (eds) Modeling and Tools for Network Simulation. Springer, Berlin, Heidelberg*. https://doi.org/10.1007/978-3-642-12331-3_2

[6] Müller, M., Ademaj, F., Dittrich, T. et al. Flexible multi-node simulation of cellular mobile communications: the Vienna 5G System Level Simulator. J *Wireless Com Network* 2018, 227 (2018). https://doi.org/10.1186/s13638-018-1238-7

[7] “Atoll.” Accessed: Apr. 24, 2022. [Online]. Available: https://www.forsk.com/atoll-overview.

[8] “Possibilities, Advantages Benefits of Real Time Network Simulation.” Accessed: Apr. 24, 2022. [Online]. Available: https://www.scalable-networks.com/possibilitiesadvantages-benefits-real-time-network-simulation/.

[9] “OMNeT++: Discrete Event Simulator.” Accessed: Apr. 24, 2022. [Online]. Available: https://apps.nsnam.org/app/mmwave

[10] “5G Toolbox: Simulate, analyze, and test 5G communications systems.” Accessed: Apr. 28, 2022. [Online]. Available: https://www.mathworks.com/products/5g.html.

[11] D. M. Rose, J. Baumgarten, S. Hahn, and T. Kurner, “Simone-simulator for mobile networks: System-level simulations in the context of realistic scenarios,” in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pp. 1–7, IEEE, 2015.

[12] C. -K. Jao et al., "WiSE: A System-Level Simulator for 5G Mobile Networks," in *IEEE Wireless Communications*, vol. 25, no. 2, pp. 4-7, April 2018, doi: 10.1109/MWC.2018.8352614.

[13] U. Masood, H. Farooq and A. Imran, "A Machine Learning Based 3D Propagation Model for Intelligent Future Cellular Networks," in *2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 2019, pp. 1-6, doi: 10.1109/GLOBECOM38437.2019.9014187.*

[14] Y. Jin, J. Zhang, S. Jin and B. Ai, "Channel Estimation for Cell-Free mmWave Massive MIMO Through Deep Learning," in *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10325-10329, Oct. 2019, doi: 10.1109/TVT.2019.2937543.

[15] S. M. A. Zaidi, M. Manalastas, H. Farooq and A. Imran, "SyntheticNET: A 3GPP Compliant Simulator for AI Enabled 5G and Beyond," in *IEEE Access*, vol. 8, pp. 82938-82950, 2020, doi: 10.1109/ACCESS.2020.2991959.