Can ML Empower Efficient Wireless Network Self-Configuration and Optimization?

Prof. Dr.-Ing. Marina Petrova

BOWW 2025
Berlin, Germany
Sept. 09-10, 2025

Gefördert durch:
Bundesministerium für Forschung, Technologie und Raumfahrt
Open6GHub
MCC Mobile Communications and Computing
RWTH AACHEN UNIVERSITY

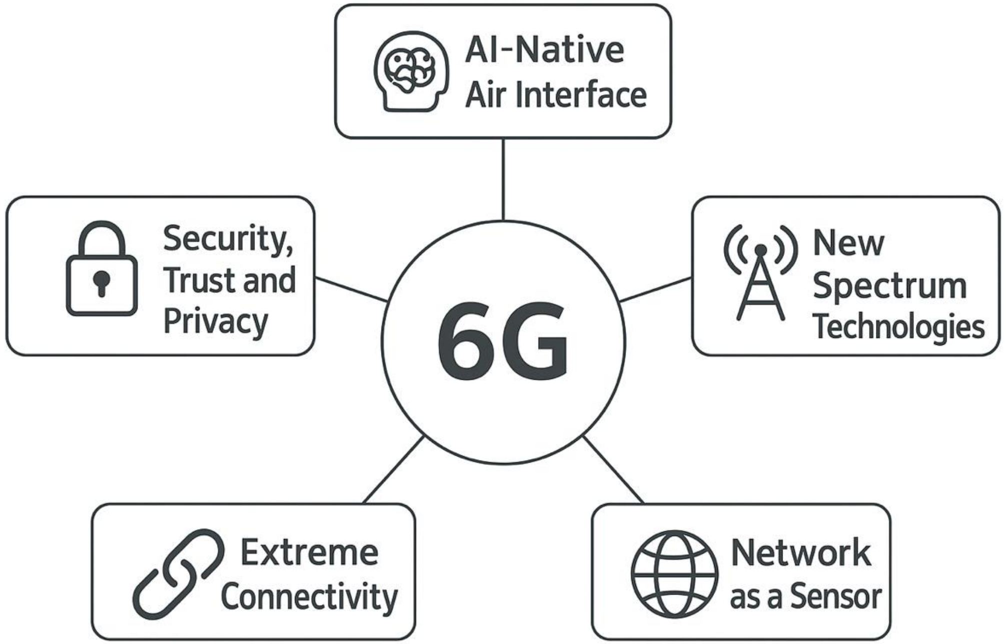
Wireless Traffic of the Future

Human-centric devices

Machine-centric devices

The Promise of 6G…

- 6G are expected to revolutionize human and machine communications.
- Should deliver unprecedented capacity, low latency, energy efficiency, and cognitive capabilities to manage vast radio resources.



AI for Wireless Networks

input
data, measurements

algorithm

output
decision

- signals theory
- optimization theory
- Fourier analyses
- signal processing
- …
- **AI**

AI for Wireless Networks

PHY	MAC	Network/Transport	APPs
• channel estimation • digital pre-distortion • channel resource optimisation • Autoencoder • …	• resource allocation • scheduling • link adaptation • …	• congestion control • mobility management • …	• AI as a service • digital twins • predictive maintenance • …

Protocols design and engineering?

Challenges

- explainability (technical depth and dependencies)
- unstable decisions in unseen situations
- efficient data collection and learning
- energy and computational efficiency
- Cost $

This Talk…

will introduce

- Multi-Agent DRL for MAC Protocol Synthesis and Optimization
- LLM based Resource Block Allocation in Multi-Cell Networks

… and discuss the trade-offs of automation, flexibility and efficiency.

Background

- 6G networks will offer a variety of services beyond connectivity
  - in licensed and unlicensed bands.
  - through coexistence of different access technologies.
  - addressing a wide spectrum of service requirements.
  - This calls for flexibility and adaptivity in the radio access protocols
  - Can ML assist the design of reconfigurable protocols?
  - Here we study a distributed MARL-based Medium Access Control (MAC)

.

Advancement Beyond State-of-the-Art

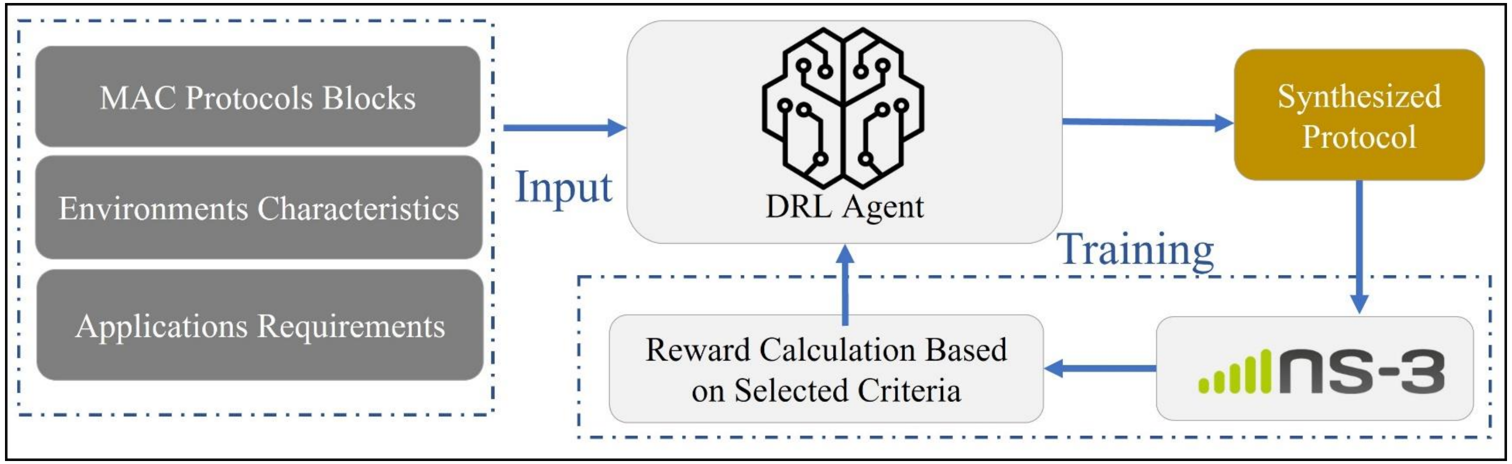
- In heterogeneous networks, it’s desirable to
  - adapt the algorithm and protocols parameters on-the-fly according to the radio environment, network loads, and application requirements.
  - compose/select the right algorithm and parameter depending on the use case.

Advancement Beyond State-of-the-Art

- We design a MARL-driven MAC Protocol framework:
  - adopts a fully distributed protocol design approach
  - optimizes several MAC parameters and functions simultaneously and generates new policies.
  - deploys intelligent agents directly on network devices, rather than embedding fixed protocols
  - agents autonomously synthesize, optimize, and dynamically adapt MAC protocols based on local observations, and radio and traffic conditions.

Multi-Agent Deep Reinforcement Learning (MADRL) framework

- enables fully distributed learning and decision-making by network nodes.
- Modular MAC protocol synthesis using ML-driven policies.

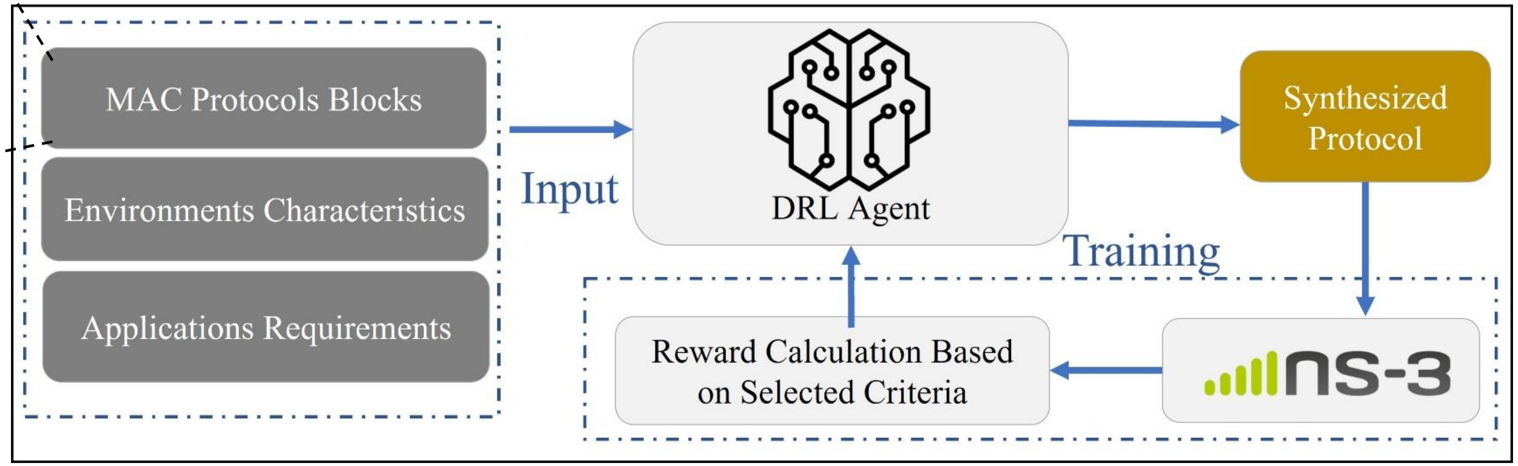


LBT: Listen Before Talk | RS: Reservation signal | MCOT: Maximum Channel Occupancy Time | CS: Carrier Sensing | mCW: minimum Contention Windows | BEB: Binary Exponentially Backoff
EIED: Exponential Increased Exponential Decreased | ED: Energy Detection

Multi-Agent Deep Reinforcement Learning (MADRL) framework

- enables fully distributed learning and decision-making by network nodes.
- Modular MAC protocol synthesis using ML-driven policies.

	Action parameter	Values Range	Standard value
a_1	Sensing Slot Size	{0, 1, 2, …, 20}	9 μs
a_2	Backoff type	Off, EDID, BEB, Constant	BEB
a_3	Minimum CW	{0, 1, 2, …, 63}	15
a_4	MCOT [ms]	{0, 1, 2, …, 10}	2, 3, 5, 8
a_5	MCS	{0, 1, 2, …, 28}	Auto. Rate Control
a_6	T_{df} [μs]	{0, 1, 2, …, 20}	16
a_7	ED_{Th} [dBm]	{-90, -89, …, -60}	-62 dBm
a_8	T_x [dBm]	{10, 11, …, 30}	23 dBm



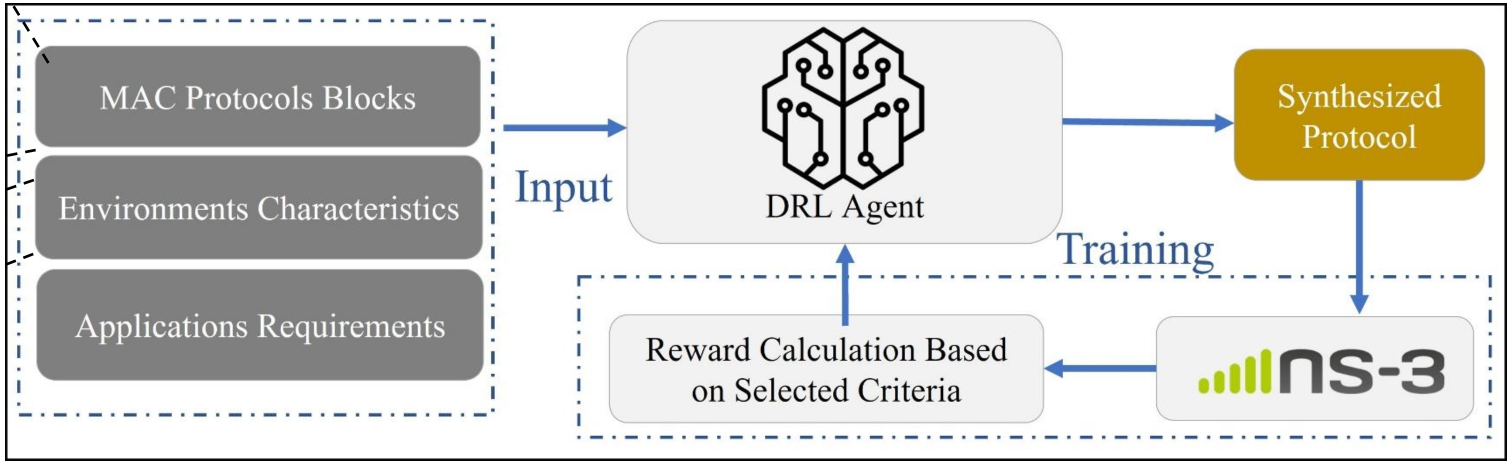
LBT: Listen Before Talk
RS: Reservation signal
MCOT: Maximum Channel Occupancy Time
CS: Carrier Sensing
mCW: minimum Contention Windows
BEB: Binary Exponentially Backoff
EIED: Exponential Increased Exponential Decreased
ED: Energy Detection

Multi-Agent Deep Reinforcement Learning (MADRL) framework

- enables fully distributed learning and decision-making by network nodes.
- Modular MAC protocol synthesis using ML-driven policies.

	Action parameter	Values Range	Standard value
a_1	Sensing Slot Size	{0, 1, 2, …, 20}	9 μs
a_2	Backoff type	Off, EDID, BEB, Constant	BEB
a_3	Minimum CW	{0, 1, 2, …, 63}	15
a_4	MCOT [ms]	{0, 1, 2, …, 10}	2, 3, 5, 8
a_5	MCS	{0, 1, 2, …, 28}	Auto. Rate Control
a_6	T_{df} [μs]	{0, 1, 2, …, 20}	16
a_7	ED_{Th} [dBm]	{-90, -89, …, -60}	-62 dBm
a_8	T_x [dBm]	{10, 11, …, 30}	23 dBm

Number of Network , Traffic rate and type , $RSSI_C$, $RSSI_I$, Throughput, delay, Airtime



LBT: Listen Before Talk
RS: Reservation signal
MCOT: Maximum Channel Occupancy Time
CS: Carrier Sensing
mCW: minimum Contention Windows
BEB: Binary Exponentially Backoff
EIED: Exponential Increased Exponential Decreased
ED: Energy Detection

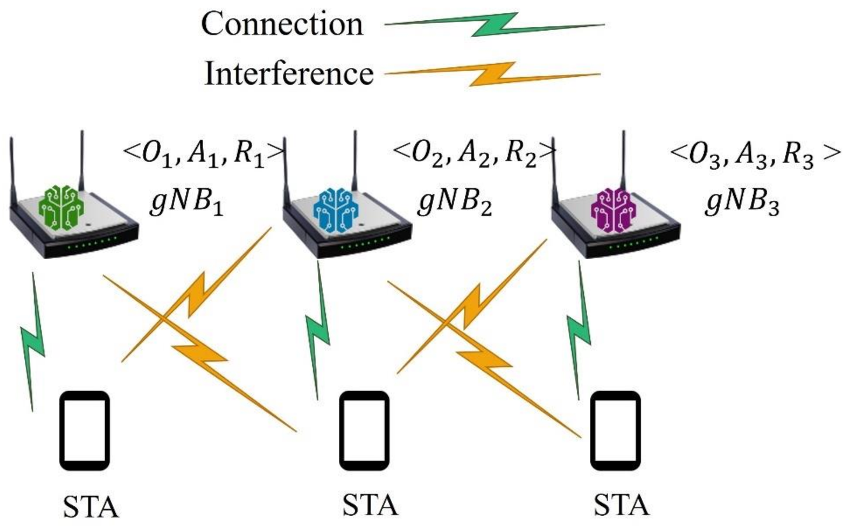
Learning approach

Distributed Training and Distributed Execution (DTDE)

Partial Observation Markov decision process

$o_x = \ll \text{Current Action}_x, \text{NN}_x, \text{TR}_x, \text{RSSI}_C, \text{RSSI}_I,$
$throughput\_x \backslash delay\_x \backslash Airtime\_x >| \forall\, gnb_x, x \in gNBs\ in\ the\ sensing\ range\}>$

$A_x = < MCOT_X, Power_x, MCS_x, ED_{THR_X}, defer\ time_X$
$Backoff_{type_x}, CW_{min_x}, Sensing\ slot\ duration_x >$



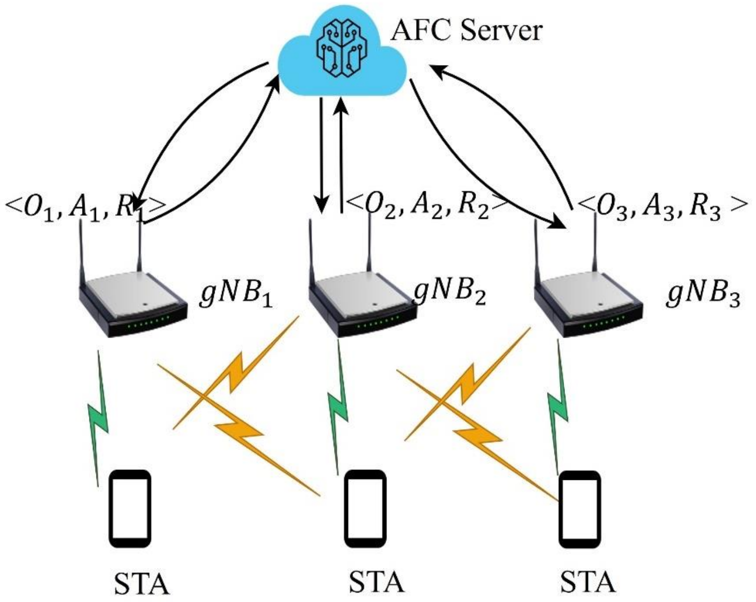
- In DTDE, each agent broadcasts its throughput, traffic rate, and airtime to the nodes within its range.

Centralized Training Centralized Execution (CTCE)

Markov Decision Process

$o_x = \ll \text{Current Action}, \text{NN}, \text{TR}, \text{RSSI}_C, \text{RSSI}_I,$
$throughput \backslash delay \backslash Airtime >| \forall\, gnb_x, x \in \{1, \ldots, NN\}>$

$A_x = < MCOT, Power, MCS, ED_{THR}, defer\ time,$
$Backoff_{type}, CW_{min}, Sensing\ slot\ duration> >| \forall\, gnb_x, x \in \{1, \ldots, NN\}>$



Learning approach

- Reward for each agent:

$$R = \sum \frac{\overline{Th_i}}{\bar{\lambda}} - \alpha \, \overline{t_{air,i}}$$

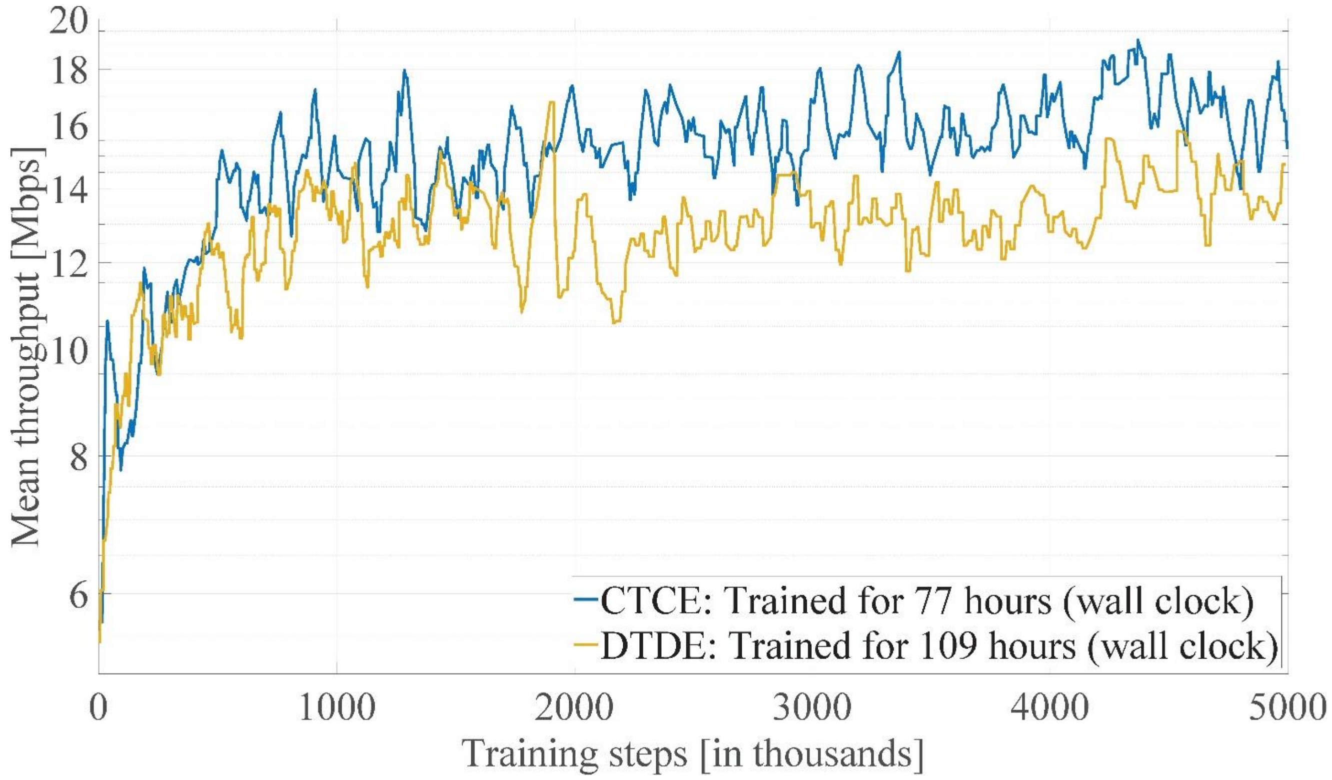
- Proximal Policy Optimization (PPO)

Table 1.Taining and Environment Parameters

Number of networks (NN)	1-6
Operating Frequency, Bandwidth	6 GHz, 20 MHz
Traffic characteristic (TR): Poisson and AR/VR with arrival rates λ	λ =[0 - 3000]
Packet size	1500
Learning Rate, Optimizer	0.001, Adam
Policy	RNN (2 layers of 256)
batch size, M	1000
Step size, Episode duration	0.1 s, 50 s
α	0.3

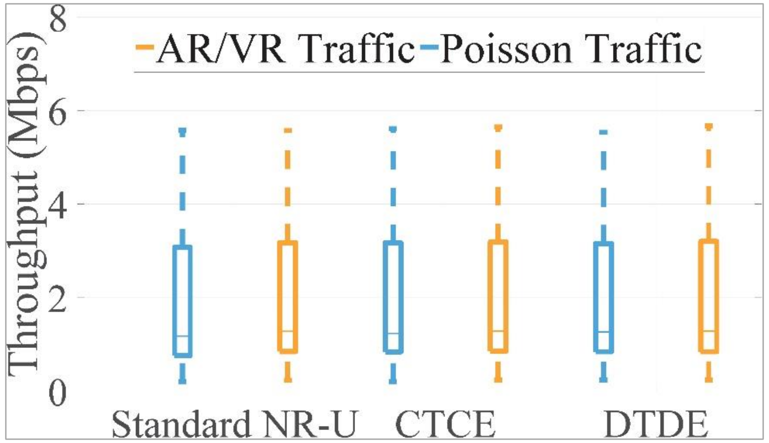
$\overline{Th_i}$: Mean normalized aggregated downlink throughput of i_{th} network
$\bar{\lambda}_i$: Normalized traffic arrival rate
$\overline{t_{air,i}}$: normalized airtime of i_{th} gNB

Learning Convergence

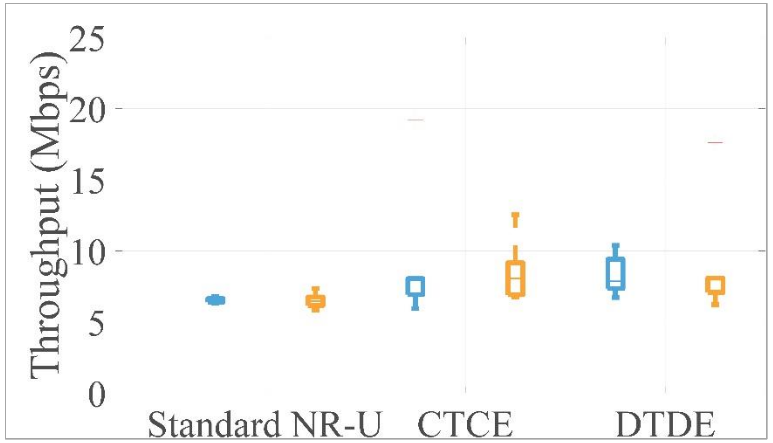


DTDE achieves slightly lower mean reward compared to centralized learning, due to lack of full control and knowledge
The simulation and training processes were conducted on a server with 2 NVIDIA A30 GPU units and 64 CPU cores.

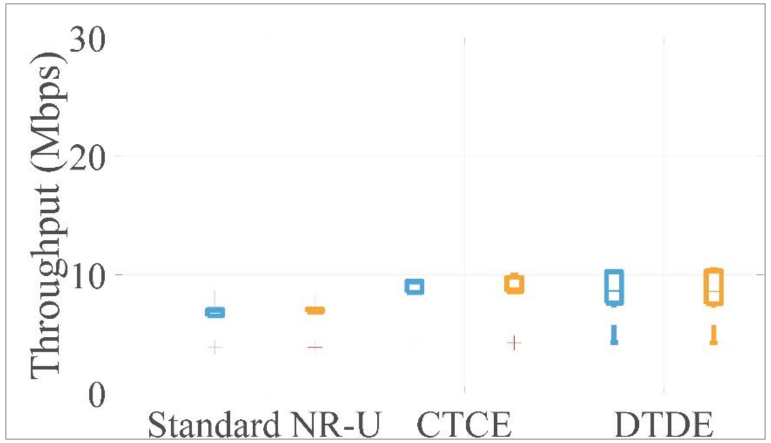
Performance Analyses



Low Traffic Scenario
(10 to 500 packets/sec)



High Traffic Scenario
(1000 to 3000 packets/sec)



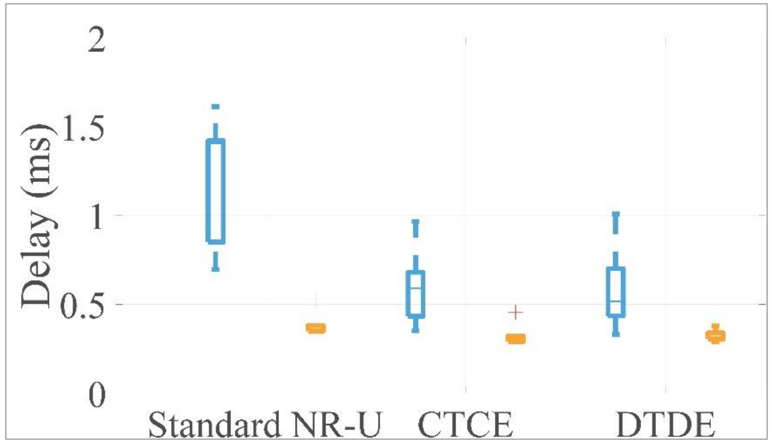
Random Traffic Scenario
(1000 to 3000 packets/sec)

- The results obtained for six networks within the environment.
- Performance under diverse traffic scenarios (Poisson, AR/VR).
- MADRL improves throughput by at least **10%** compared to standard 5G NR-U.
- Performance closely matches centralized learning approaches despite decentralized, partial observability.

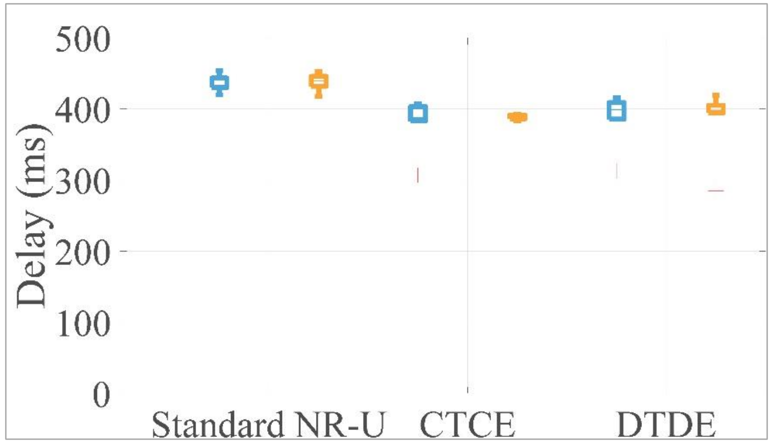
Mobile Communications and Computing

RWTH AACHEN UNIVERSITY

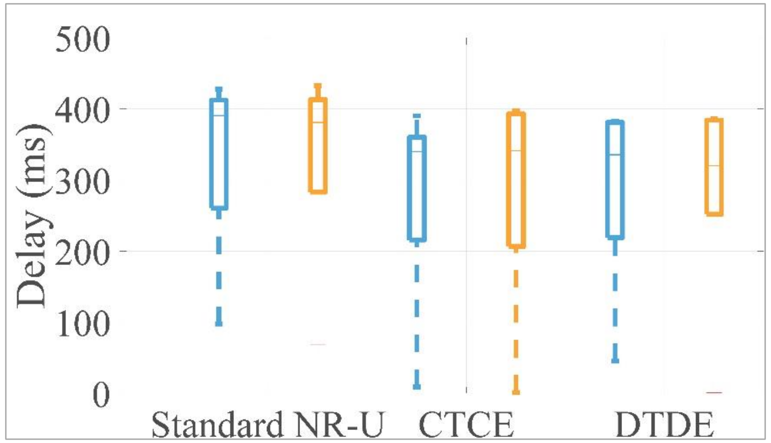
Performance Analyses



Low Traffic Scenario
(10 to 500 packets/sec)



High Traffic Scenario
(1000 to 3000 packets/sec)



Random Traffic Scenario
(1000 to 3000 packets/sec)

- Substantial reduction in end-to-end packet delay.
- Reduced carrier-sensing overhead contributes to lower latency.
- Power control and energy detection thresholds dynamically adjusted by each node minimize interference.

Concluding Remarks 1

- DLR agents autonomously synthesize, optimize, and dynamically adapt MAC protocols based on local observations and conditions.
- The synthesis protocols demonstrate notable enhancements in throughput and latency reduction.

- Future work:
  - Analyzing distributed learning approaches for enhanced adaptability in heterogeneous environments 5G NR/Wi-Fi.
  - Implementing on the real hardware.
  - Explore accelration

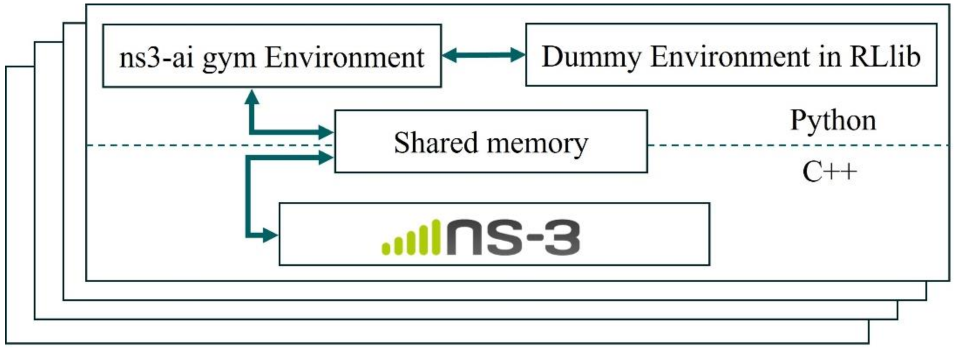
Dissemination and Open-Source Availability

- Paper Reference:

  N. Keshtiarast, O. Renaldi and M. Petrova, "Wireless MAC Protocol Synthesis and Optimization With Multi-Agent Distributed Reinforcement Learning," in IEEE Networking Letters, vol. 6, no. 4, pp. 242-246, Dec. 2024, doi: 10.1109/LNET.2024.3503289.

- Open-Source Implementation:

  - Applicable for multi agent optimizing for single or multiple MAC/PHY layer parameters.
  - Supports diverse technologies: 5G NR, 5G NR-U, Wi-Fi (IEEE 802.11 protocols)
  - Highly adaptable to various application scenarios and network environments.



Vectorized Environment

https://github.com/navid-keshtiarast/ML-Framework-for-NR-U-MAC-Protocol-Design-Multi-agent

LLMs for Resource Block Assignment with QoS Constraints in OFDMA Multi-Cell (Open) RAN

Can Large Language Models (LLMs) help?

task → LLM → output

expressed in
human language

in text, code, script,
and configuration

what is it? huge neural networks
trained on large corpus of text

task
expressed in
human language

→

LLM

→

output
in text, code, script,
and configuration

how does it work? given text input,
predict next sequence of words

Can Large Language Models (LLMs) help?

task → LLM → output

task
expressed in human language

LLM

output
in text, code, script, and configuration

OpenAI ChatGPT4.0

Claude 3

LLaMA by Meta

Gemini

Example of LLMs

Can Large Language Models (LLMs) help?

task → LLM → output

expressed in human language

in text, code, script, and configuration

knowledgeable

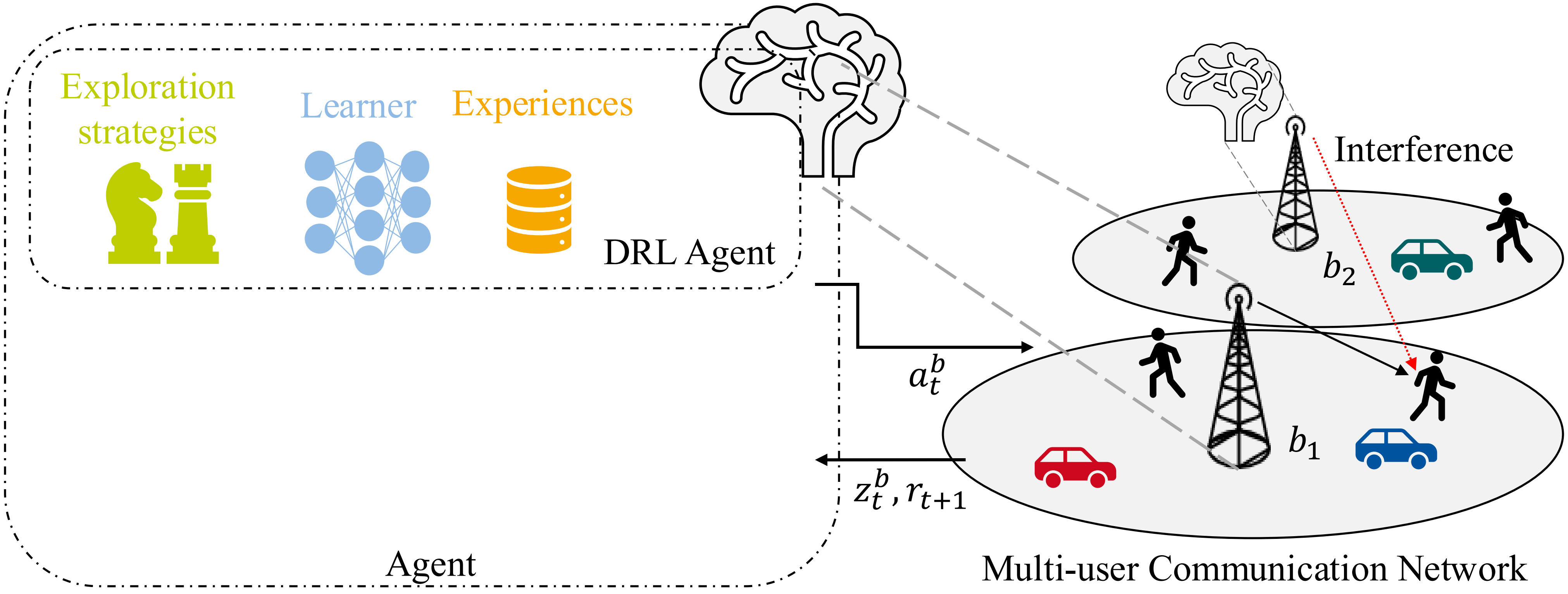
"LLMs can generate coherent, contextually relevant text based on prompts." →

fast learner

creative

Can LLMs **help** in wireless **network configuration**?

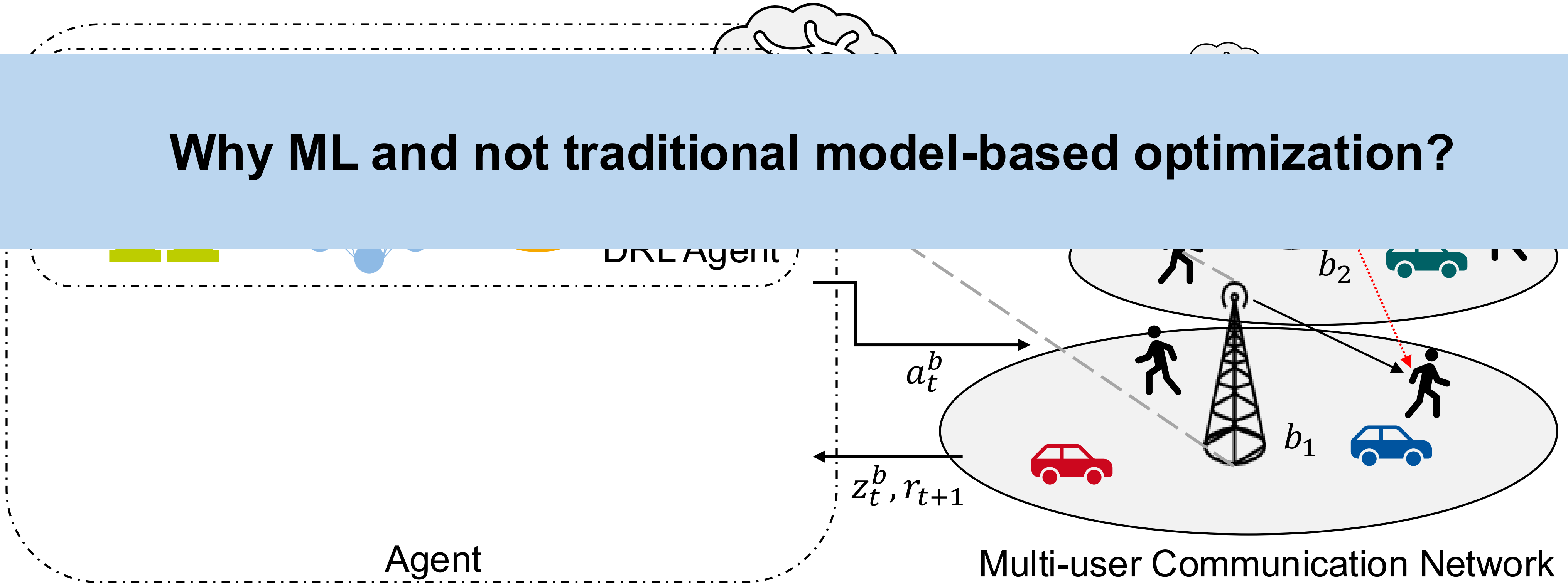
Resource Block Assignment in OFDMA Wireless Systems



z_t^b → input observation (channel gains, resource block assignments, user requirements)
a_t^b → action (resource block assignments)
r_{t+1} → reward (data rate of the base station)

Resource Block Assignment in OFDMA Wireless Systems

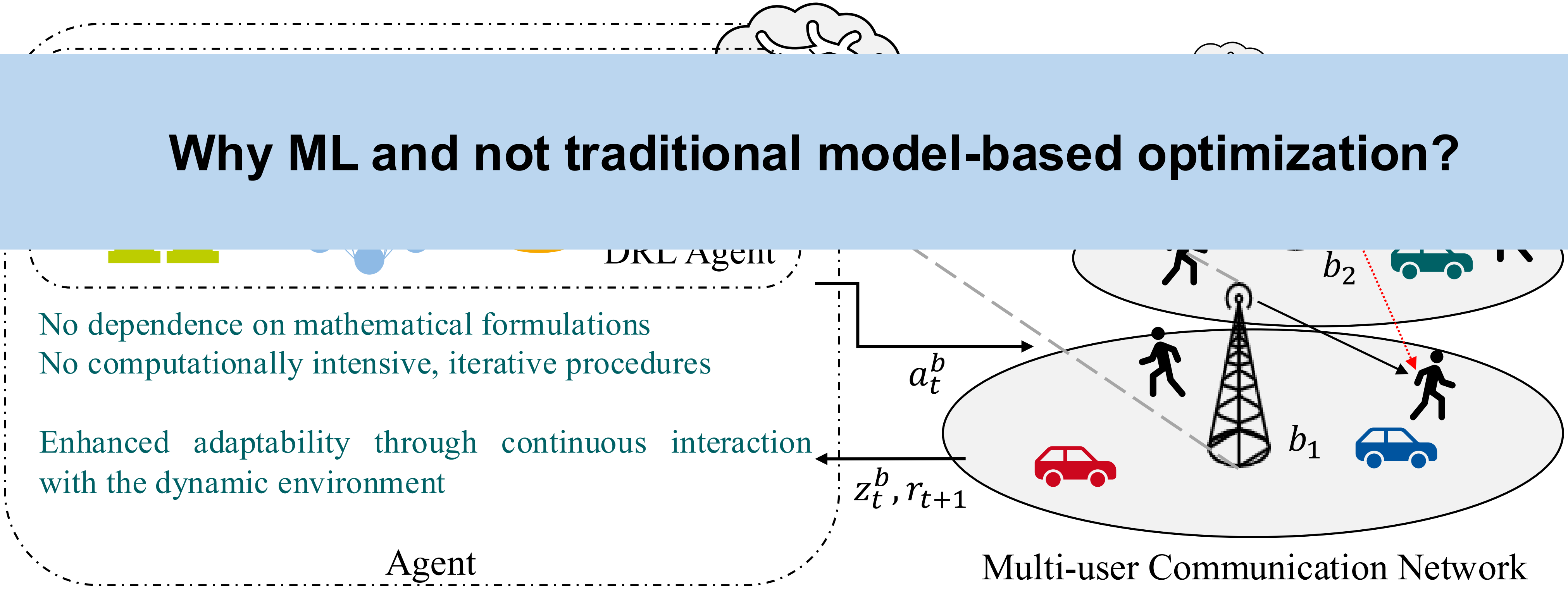
Why ML and not traditional model-based optimization?



z_t^b → input observation (channel gains, resource block assignments, user requirements)
a_t^b → action (resource block assignments)
r_{t+1} → reward (data rate of the base station)

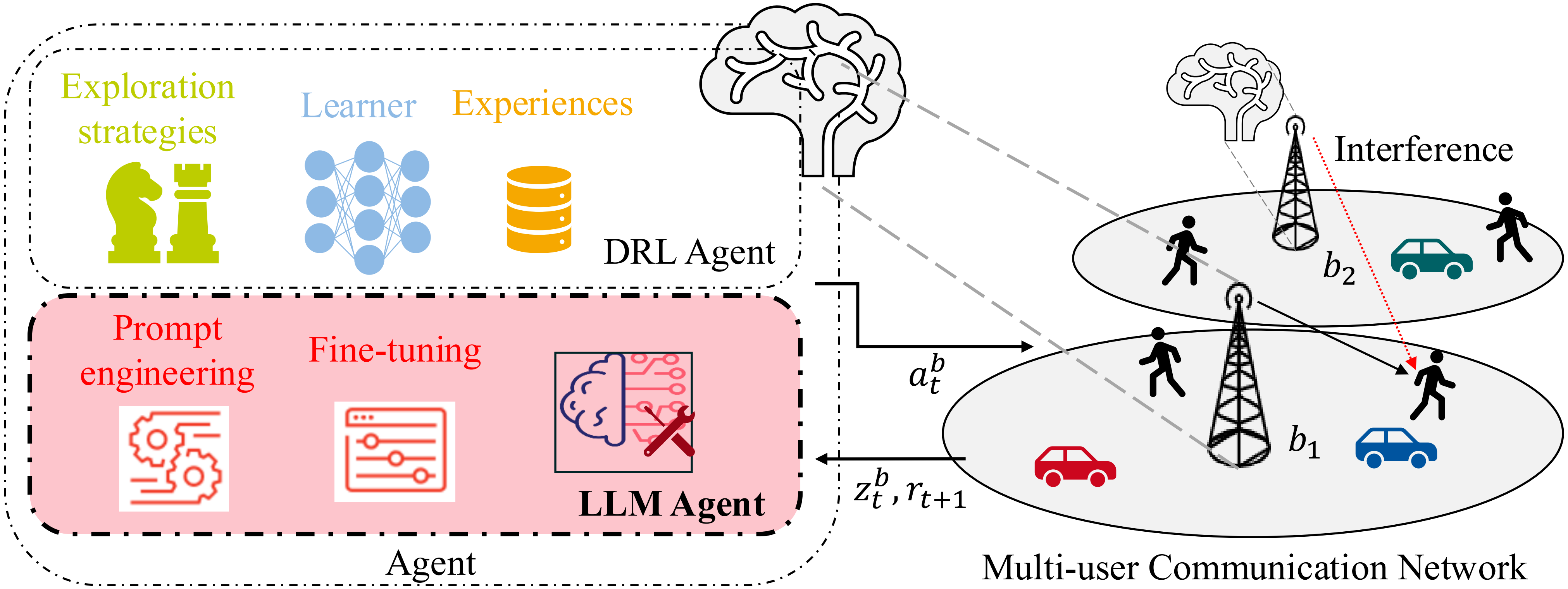
Resource Block Assignment in OFDMA Wireless Systems

Why ML and not traditional model-based optimization?



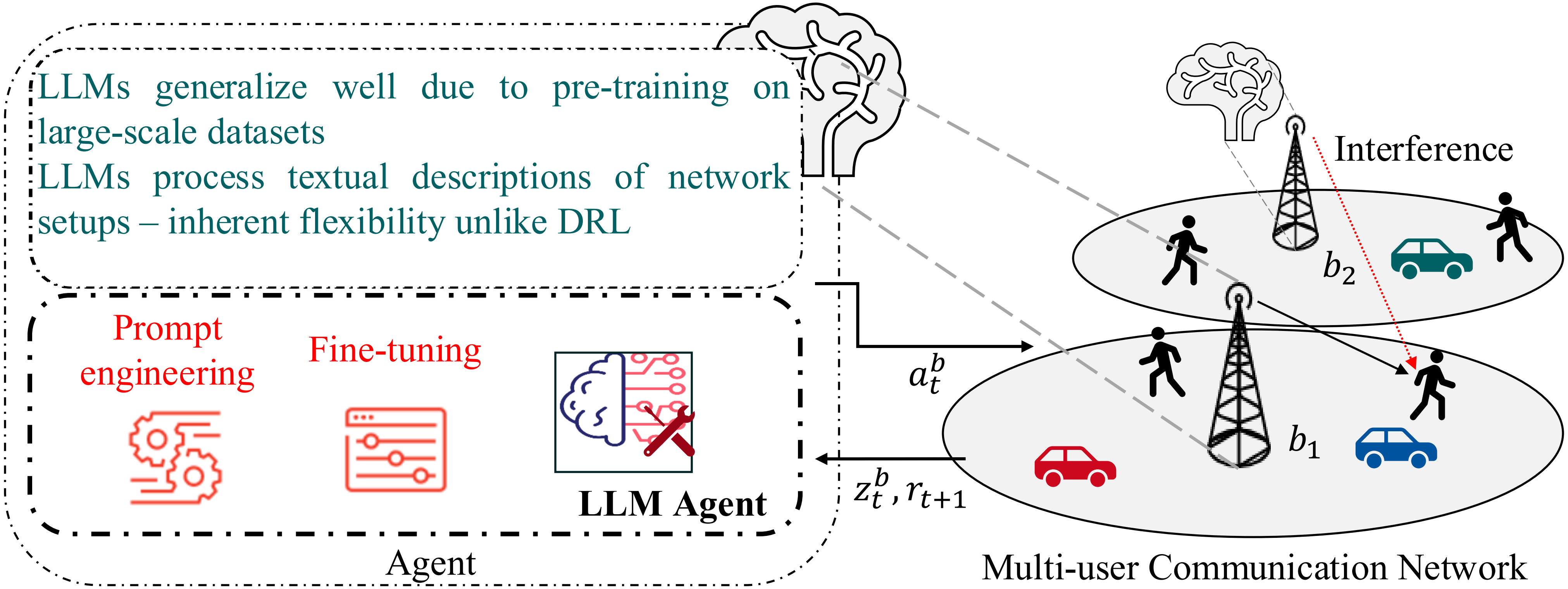
z_t^b → input observation (channel gains, resource block assignments, user requirements)
a_t^b → action (resource block assignments)
r_{t+1} → reward (data rate of the base station)

Resource Block Assignment in OFDMA Wireless Systems



z_t^b → input observation (channel gains, resource block assignments, user requirements)
a_t^b → action (resource block assignments)
r_{t+1} → reward (data rate of the base station)

Resource Block Assignment in OFDMA Wireless Systems

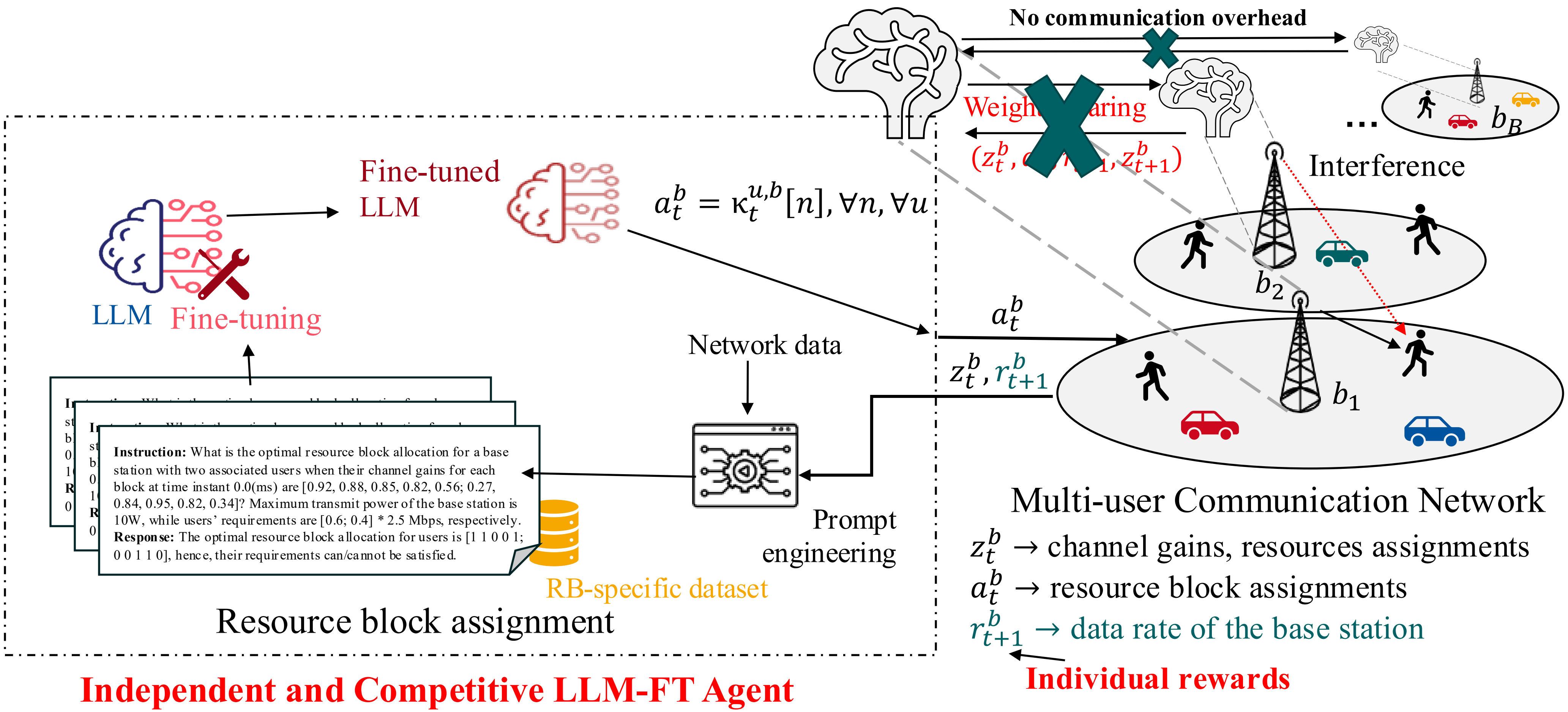


z_t^b → input observation (channel gains, resource block assignments, user requirements)
a_t^b → action (resource block assignments)
r_{t+1} → reward (data rate of the base station)

In the following

- we address the resource block assignment problem in a multi-user, multi-cell OFDM Open RAN
  - Constraints: minimum user rate requirements and maximum transmit power constraints for each base station.
  - This design ensures vendor-agnostic deployment of xAPPs and seamless integration into heterogeneous Open RAN ecosystems.
- we propose a competitive agent interaction model with independent learning
  - LLM-FT performs resource block assignment—ensuring adaptability across varying network configurations
  - This approach eliminates the communication overhead of exchanging weight parameters and experiences
- the LLM-FT-based framework enables simultaneous resource block assignment across multiple resource blocks.

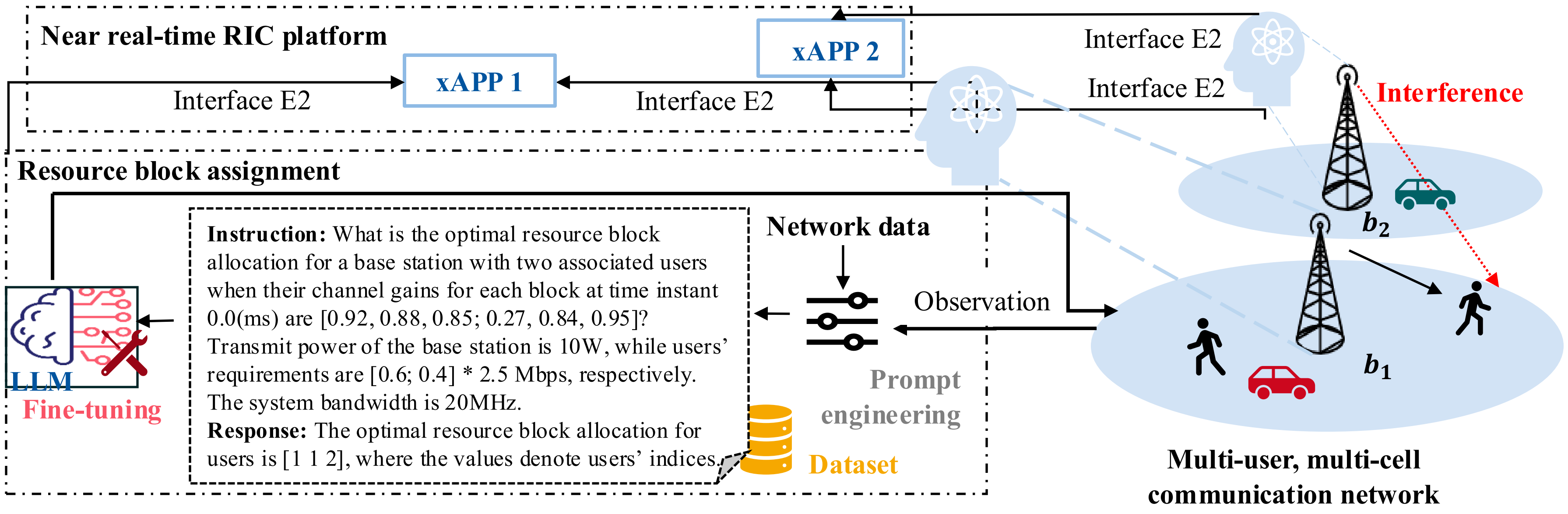
MALLM-FT-based Implementation Framework [1]



Independent and Competitive LLM-FT Agent

[1] A. Kopić, L. Simić, M. Petrova, "Large Language Models for Resource Block Assignment with QoS Constraints in OFDMA Multi-Cell Open RAN," under review.

Open RAN: MALLM-FT-based Implementation Framework



Baseline Algorithms

- Resource block allocation
  - Hungarian algorithm [2] – suboptimal solution
  - Max-rate (MR) scheduler (greedy)
  - Proportional fair (PF) scheduler (time/spectrum round robin scheduler)
  - DRL-based solution: Deep Q-Network (DQN)
    - Same components' design as LLM agent

[2] H. Yin and H. Liu, "An efficient multiuser loading algorithm for ofdm-based broadband wireless systems," in Proc. Globecom '00 - IEEE, vol. 1, 2000, pp. 103–107.

DRL Model Parameters

Parameter	Value
Number of test episodes (N_{test})	10 000
Number of warm-up episodes (N_{wrm})	1000
Number of training episodes (N_{tr})	**20 000**
Batch size (S)	64
Size of replay memory (M)	31 000
Discount factor (γ)	0.95

Parameter	Value
DQN-specific – RB assignment	
Frequency of the target network update (T)	10
Epsilon (training values)	$\varepsilon_I = 1.0 \rightarrow \varepsilon_F = 0.001$
Learning rate (α)	0.001

Network	DNN Architecture
DQN	$[N_{RB}+U^b+U^b N_{RB}$, 128, 32,$U^b N_{RB}]$ *(activation = elu)*

Reward function: $$r^b_{t+1}+ = \begin{cases} X^{u,b}_{t+1}, & \text{if rate demand holds;} \\ -0.1(R^{u,b}_{min} - X^{u,b}_{t+1})^{0.5}, & \text{otherwise;} \end{cases}$$
$\forall u, \ \forall b.$

Mobile Communications and Computing

RWTH AACHEN UNIVERSITY

LLM Parameters – Resource Block Allocation

Parameter	Value
Phi-3 Mini-specific	
Number of parameters (N_{prm})	3.82 billion
Context length (N_{con})	128 000
Fine-tuning method (FT)	LoRA
Learning rate (α)	0.00005
Number of epochs to perform (N_e)	3.0
Number of samples for LLM-FT (N_{ft})	**21 000**
Custom dataset format *(alpaca, sharegpt)*	*sharegpt*
Compute type	fp16
Cutoff length – max number of input tokens (N_{cut})	1024
Total train batch size (S_{LLM})	32
Percentage of trained parameters	0.33%

System and Communication Channel Model Parameters

Symbol + value	
B = 4	N_{RB} = 50 ***
U = 40	W = 20 MHz
$P_{max} = 40dBm$	
Channel model parameters	
$\overline{\sigma}^2 = 1$	$f_c = 1.8$ GHz
$v = [0, 50)$ km/h *	$\sigma_{SF} = 7.82\ dB$
L = 8	$\mu_\tau = 1200$ ns
$T_{s,OFDM} = 33.3\ \mu s$ – OFDM symbol duration	
$\Delta f = 30\ kHz - subcarrier\ spacing$	
$W_{min,guard} = 845\ kHz$ – minimal guard bandwidth	

- $N_{subc}^{RB} = 12 \rightarrow$ 360 kHz per RB
- $N_{RB} = \left\lfloor \frac{W - 2\,W_{min,guard}}{\Delta f\ N_{subc}^{RB}} \right\rfloor = 50 \rightarrow$ 600 subcarriers
- User traffic model:

Application	User Percentage	Rate requirement
Web browsing / HTTP	20%	0.5 (Mbps)
FTP	10%	1 (Mbps)
Video (SD)	20%	1.5 (Mbps)
VoIP	30%	0.1 (Mbps)
Online gaming	20%	0.3 (Mbps)

*Evenly spaced values within an interval $v = [0, 50)$ for all users

***Subcarrier spacing configuration 1 for 30 kHz subc. spacing

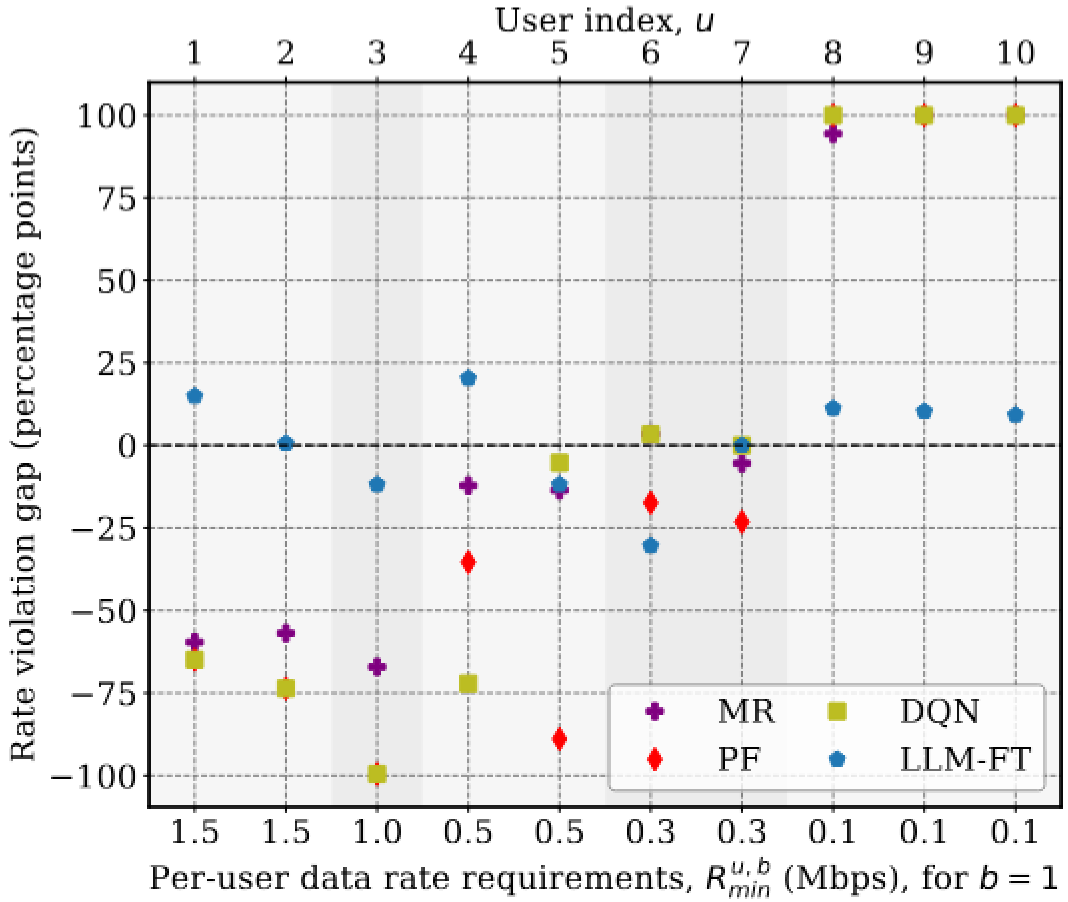
3GPP, "5g NR; Physical Channels and Modulations (3gpp ts 38.211 version 16.6.0 release 16)."

Performance Evaluation

- Users' rate requirement violations
  - Performance with high-rate users
  - Performance with low-demand users
- Generalizability of DQN and LLM-FT across user configurations
- Training, fine-tuning, and inference times

Users' Rate Requirement Violations

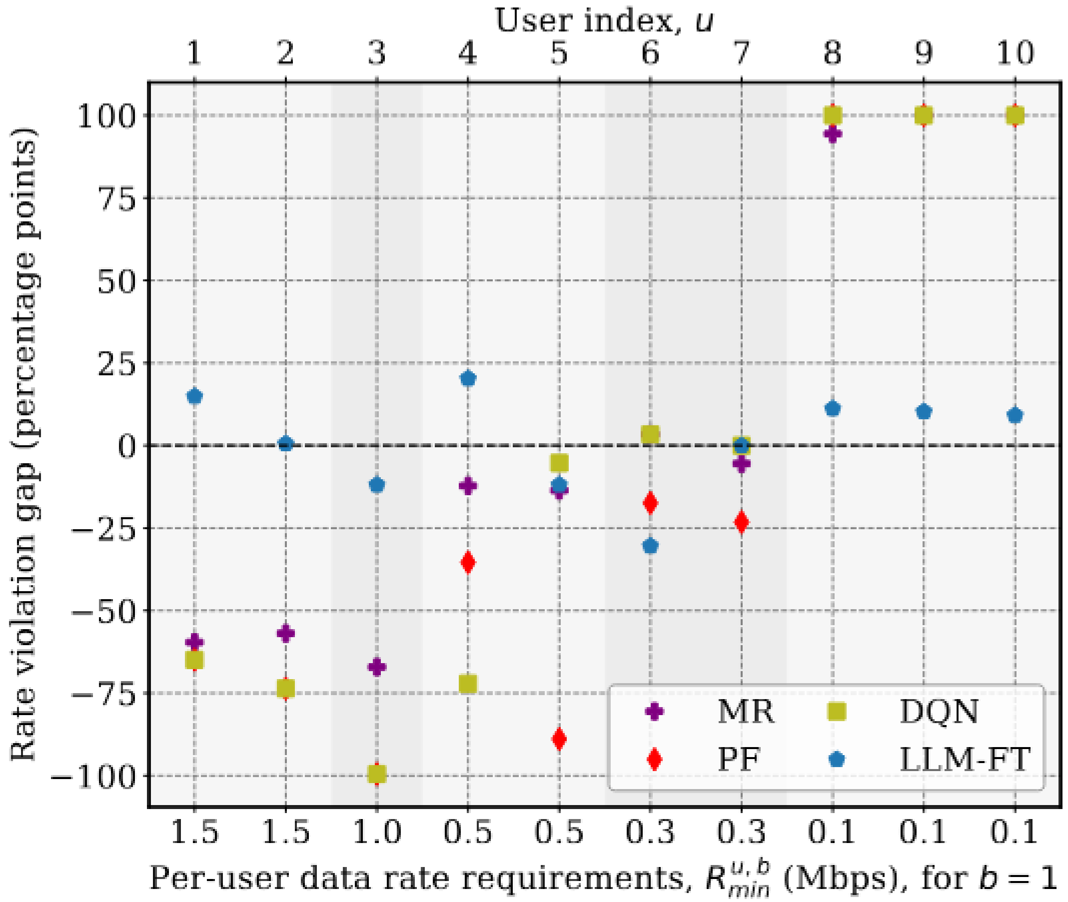
- Likelihood that a user's data rate requirements is not satisfied



- Note: A positive rate violation gap indicates that the method outperforms the benchmark Hungarian method, and vice versa.

Users' Rate Requirement Violations

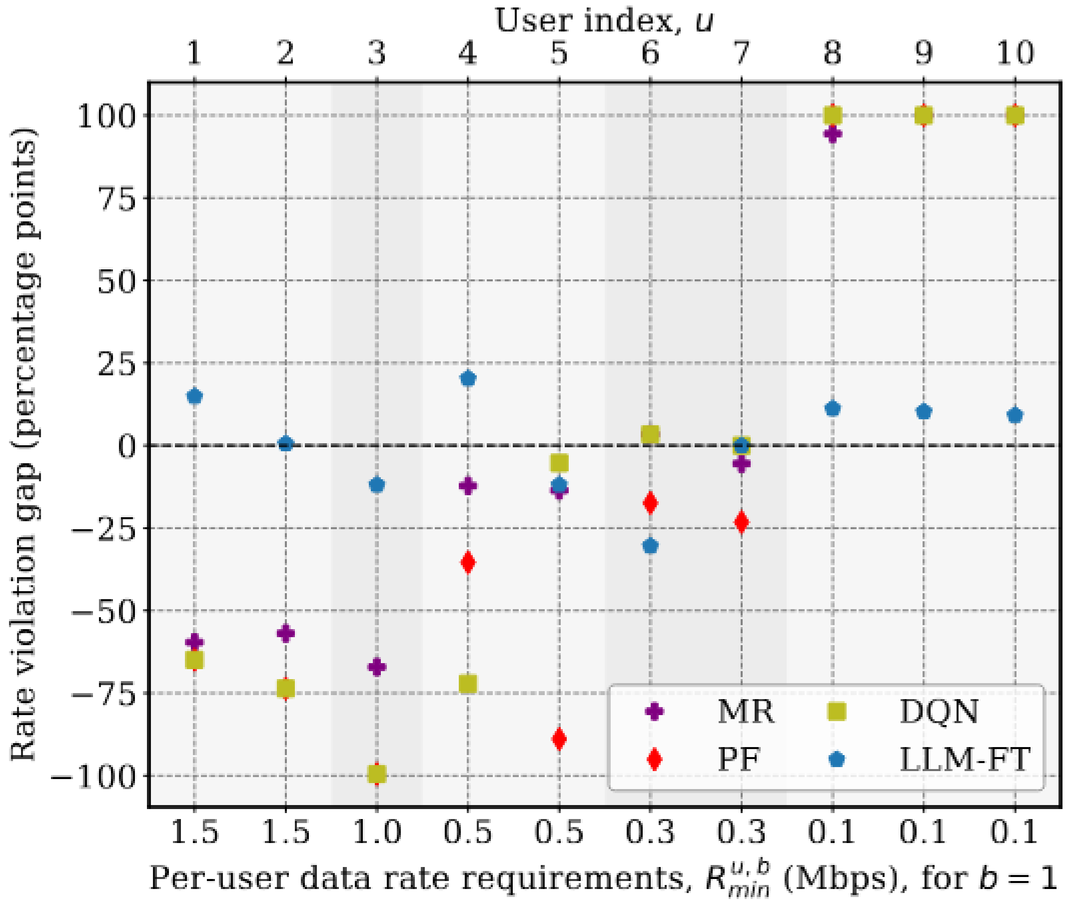
- Likelihood that a user's data rate requirements is not satisfied



- Note: A positive rate violation gap indicates that the method outperforms the benchmark Hungarian method, and vice versa.
- Across almost all configurations, LLM-FT component outperforms Hungarian benchmark
- Why?
  - LLM is more adaptable to heterogeneous rate demands
    - Inherently QoS-aware and incorporates historical performance or tracks unmet user demands
    - Leads to long-term satisfaction optimization
    - Especially apparent for users 8-10, who rely on less bandwidth-heavy VoIP services

Users' Rate Requirement Violations

- Likelihood that a user's data rate requirements is not satisfied



- Performance with high-rate users
  - For users 1-5, associated with web browsing, FTP, and video services, LLM-FT and Hungarian methods outperform MR, PF scheduling, and DQN.
  - User 1: LLM-FT reduces the likelihood of rate violations by 80 percentage points compared to both PF and DQN methods.
  - MR and PF limitations stem from their lack of sensitivity to individual user requirements, leading to suboptimal decisions in environments with heterogeneous per-user QoS demands.
  - The DQN model exhibits similar behavior

Generalizability of LLM across User Configurations

AVERAGE PER-CELL b SUM RATE $\overline{X}_t^b$ (MBPS) ACROSS USER DENSITIES U FOR DIFFERENT ALGORITHMS

Number of cellular users (U)	**Hungarian**				**DQN**				**LLM-FT**			
	b=1	b=2	b=3	b=4	b=1	b=2	b=3	b=4	b=1	b=2	b=3	b=4
24	6.23	6.17	6.38	6.34	5.94	5.99	6.28	6.29	7.03	6.56	6.62	6.53
40	7.70	7.41	7.26	7.57	7.33	6.86	6.56	6.86	8.62	7.51	7.02	7.63
56	7.96	8.14	8.17	7.95	–	–	–	–	7.91	6.71	6.97	6.59

- DQN is trained and LLM-FT framework is fine-tuned on a 40-user OFDM system
  - Both are tested on various user configurations
  - DQN achieves a slightly lower average per-base station sum rate in the default 40-user scenario
    - Similar observed in with fewer users, where model operates with zero-padded inputs – reasonable level of adaptability
    - However, it fails to support settings with 56 users due to fixed neural network architecture
  - LLM-FT exhibits strong generalization and adaptability across user setups
    - Yet, slight decline in high-density scenarios
    - Why?
    - Primarily due to hallucinations within the LLM output.

Training, Fine-tuning, and Inference Times

Algorithm/Time	Training	Fine-tuning	Inference (ms)
Traditional model-based methods			
MR	-	-	2.54
PF	-	-	0.69
Hungarian	-	-	34.81
DRL-based model			
DQN	~ 0.08 hour	-	1.05
LLM-based solution			
LLM-FT	~ 10 years	~ 13.25 hour	1745 (1.75 s)

- LLM has longer inference time compared to both TMBO and DRL solutions
  - Requires 50 times more time for inference than Hungarian benchmark
  - Nearly 1600 times larger inference time than DQN framework
  - Why?

Training, Fine-tuning, and Inference Times

Algorithm/Time	Training	Fine-tuning	Inference (ms)
Traditional model-based methods			
MR	-	-	2.54
PF	-	-	0.69
Hungarian	-	-	34.81
DRL-based model			
DQN	$\sim$ 0.08 hour	-	1.05
LLM-based solution			
LLM-FT	$\sim$ 10 years	$\sim$ 13.25 hour	1745 (1.75 s)

- LLM has longer inference time compared to both TMBO and DRL solutions
  - Requires 50 times more time for inference than Hungarian benchmark
  - Nearly 1600 times larger inference time than DQN framework
  - Why?
- LLM consists of 3.82 billion parameters
  - Significantly more than DRL
  - DQN: 92,436

Phi-3 Mini: One of the smallest LLMs recently introduced

Training, Fine-tuning, and Inference Times

Algorithm/Time	Training	Fine-tuning	Inference (ms)
Traditional model-based methods			
MR	-	-	2.54
PF	-	-	0.69
Hungarian	-	-	34.81
DRL-based model			
DQN	∼ 0.08 hour	-	1.05
LLM-based solution			
LLM-FT	∼ 10 years	∼ 13.25 hour	1745 (1.75 s)

- Unlike DRL, LLM-FT leverages a pre-trained LLM
  - Eliminates the need for training from scratch or full retraining
  - Pre-training would take up to 10 years on a single GPU – bypassed by a light-weight fine-tuning of existing models

Training, Fine-tuning, and Inference Times

Algorithm/Time	Training	Fine-tuning	Inference (ms)
Traditional model-based methods			
MR	-	-	2.54
PF	-	-	0.69
Hungarian	-	-	34.81
DRL-based model			
DQN	∼ 0.08 hour	-	1.05
LLM-based solution			
LLM-FT	∼ 10 years	∼ 13.25 hour	1745 (1.75 s)

- Unlike DRL, LLM-FT leverages a pre-trained LLM
  - Eliminates the need for training from scratch or full retraining
  - Pre-training would take up to 10 years on a single GPU – bypassed by a light-weight fine-tuning of existing models
  - LLM has significantly shorter inference time compared to the training or retraining duration of DRL
- Hardware acceleration advancements
  - (GPUs and TPUs), tensor and pipeline parallelism expected to reduce LLM inference latency
  - Moreover, inference libraries like FlashAttention or multi-token decoding techniques can also accelerate LLM performance

Remarks

- Proposed LLM-FT framework outperforms both model-based and DRL-based solutions
  - Achieves up to 21 percentage points lower probabilities of users' rate requirement violations
  - LLM-FT is a QoS-aware solution that incorporates historical performance and tracks unmet user demands
- LLM-FT exhibits strong adaptability across various user densities post fine-tuning, unlike DRL approaches with fixed neural network architecture
- Challenges:
  - The computational complexity of LLMs remains a major challenge
  - Yet, ongoing advancements in hardware acceleration and process parallelism are expected to substantially reduce LLM inference latency
  - This would enhance their practicality in future wireless networks

Vielen Dank
für Ihre Aufmerksamkeit