Graph-augmented PPO

Towards scalable O-RAN resource management

Duc-Thinh Ngo
duc-thinh.ngo@imt-atlantique.fr

Orange Innovation, Rennes, France
IMT Atlantique, Nantes, France

November 5th, 2025

Outline

1. Motivation
2. System model
3. Problem formulation
4. Proposed Method
5. Experimental Results
6. Conclusion

1 Motivation

Why O-RAN?

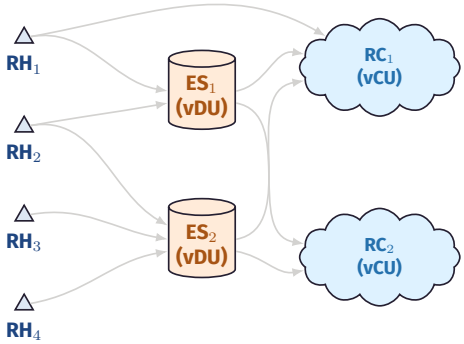
- Virtualized baseband functions (vDU, vCU)
- Support diverse services: **eMBB**, **URLLC**, **mMTC**
- Flexible deployment $\rightarrow$ Cost reduction

The Problem

- **Decision 1**: Which functional split to use?
- **Decision 2**: Where to place vDU and vCU?
- Subject to constraints
- Traffic demands change over time

→ This is an **NP-hard optimization problem**

2.1 O-RAN Architecture



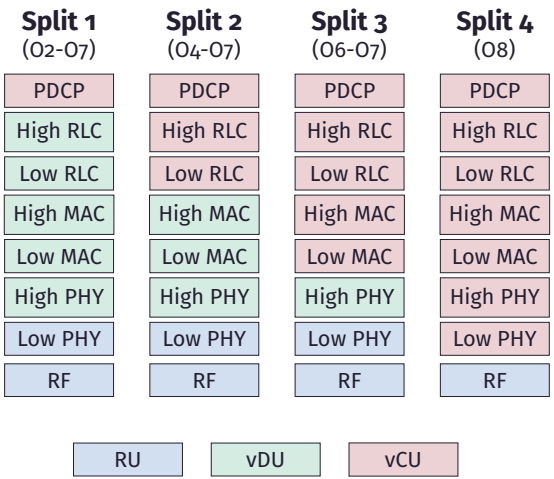
Network Components

- **RH**: Radio Heads
  Generate traffic
- **ES**: Edge Servers
  Host vDU functions
- **RC**: Regional Clouds
  Host vCU functions

Key Features

- Hierarchical topology
- Multiple routing paths
- Flexible connectivity

2.2 Functional Split Options



Key Observation

Split 1 → Split 4:
More centralization
Each split = 1 or 2 options
Each option has distinct bandwidth & latency needs

3.1 The Challenge

Minimize Total Deployment Cost

- Computing + Routing + Reconfiguration costs

Decision Variables (per RH)

1. Functional split selection
2. vDU placement (ES)
3. vCU placement (RC)

Subject to Constraints

- Connectivity, Capacity, Latency, Bandwidth

→ NP-hard Integer Linear Programming problem

3.2 Motivation: The Need for Learning-Based Solutions

Comparing solution approaches:

ILP Solvers

- Optimal solutions
- Intractable at scale
- Static only

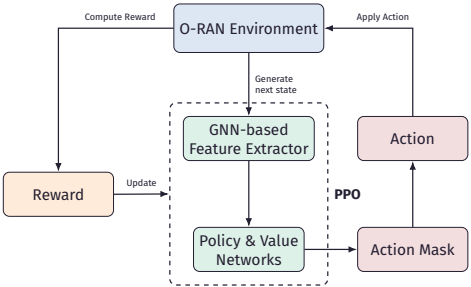
Heuristics

- Fast execution
- Suboptimal
- No learning

Deep RL

- Near-optimal
- Scalable
- Adaptive

4.1 Our Solution: Graph-Augmented PPO (GPPO)



Three Key Components

1. GNN Feature Extractor

- Topology-aware learning
- Graph representation

2. PPO Agent [Sch+17]

- Stable policy updates
- Efficient exploration

3. Action Masking

- Filter invalid actions
- Faster convergence

4.2 GNN Feature Extractor: Graph Representation

Network modeled as graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$

ES/RC Nodes

Node features $\mathrm{h_i}$:

- Remaining capacity
- Node index
- Processing cost

RH Nodes

Node features $\mathrm{h_i}$:

- Traffic demand
- Latency requirement
- Service type

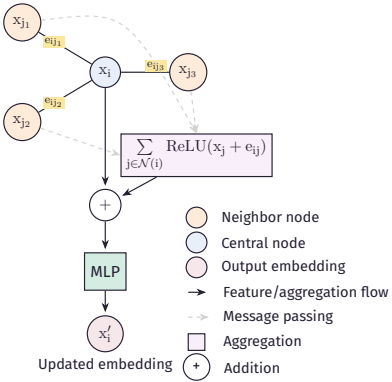
Link Edges

Edge features $\mathrm{e_{ij}}$:

- Bandwidth capacity
- Link delay
- Routing cost

→ All features projected to common embedding dimension

4.3 GNN Feature Extractor: Architecture



GINEConv Layer [Hu+20]

- Aggregates neighbors
- Adds edge features
- MLP transformation
- Two sequential layers

Global Pooling

- Mean over all nodes
- Fixed-size embedding

Output

- Fed to policy & value nets

4.4 Action Masking: Enforcing Constraints

Challenge: Large combinatorial action space with many invalid actions

Masking Strategy

1. Mask invalid actions in logits
2. Apply before softmax
3. Sample only valid actions

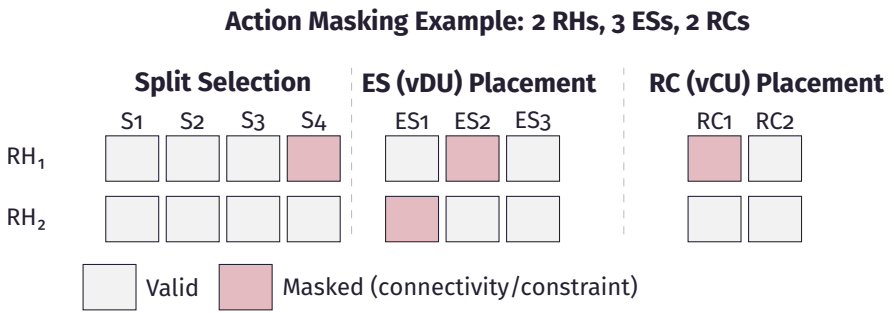
Masked Actions

- Disconnected RH-ES and RH-RC pairs
- Split 4 without direct RC link

Benefits

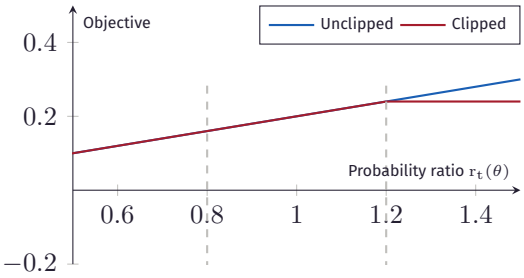
- **Faster convergence**
  No wasted exploration
- **Guaranteed feasibility**
  Hard constraints satisfied
- **Valid gradients**
  Proven in [HO20]

4.5 Action Masking: Example



Action vector: $\mathrm{a} = [\mathrm{a}_1^{\text{split}}, \mathrm{a}_2^{\text{split}}, \mathrm{a}_1^{\text{DU}}, \mathrm{a}_2^{\text{DU}}, \mathrm{a}_1^{\text{CU}}, \mathrm{a}_2^{\text{CU}}]$

4.6 PPO Training: Stable Policy Updates



Prevents policy from changing too much per update

Probability Ratio

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$$

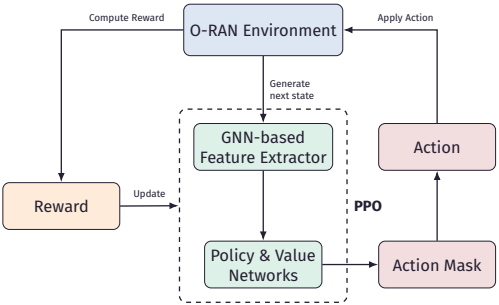
Clipped Objective

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(r_t \hat{A}_t, \text{clip}(r_t, 1 \pm \epsilon) \hat{A}_t \right) \right]$$

Why PPO? [Sch+17]

- Stable training
- Sample efficient
- Robust performance

4.7 Solution Architecture: GPPO



Key Design Principles

Topology Awareness
GNN processes network structure to capture spatial constraints and dependencies

Stable Learning
PPO ensures robust policy updates with clipped objectives

Constraint Satisfaction
Action masking eliminates invalid actions, reducing the effective action space.

5.1 Experimental Setup

Network Topologies

- **Small**: 8 RHs, 3 ES, 2 RC
- **Large**: 64 RHs, 4 ES, 2 RC

Traffic Demands

- eMBB: 250-300 Mbps
- mMTC: 150-200 Mbps
- URLLC: 20-40 Mbps

Training

- 600K steps, 32 parallel envs
- 288 timeslots per episode

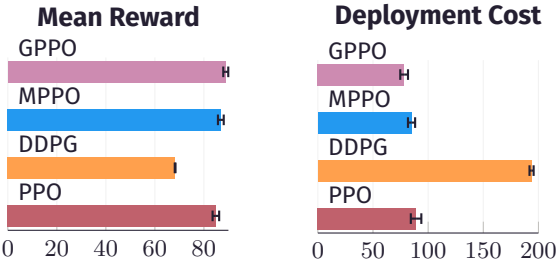
Baseline Methods

- **DDPG**: Actor-critic MLP
- **PPO**: Standard, no masking
- **MPPO**: PPO + Masking
- **GPPO**: GNN + PPO + Masking

Metrics

- Mean reward
- Deployment cost
- Success rate (6 runs)

5.2 Results: Small-Scale (8 RHs)



✔ GPPO: Best reward & lowest cost
✔ +2.3% reward, -7.9% cost vs MPPO
✔ All methods: 100% success

Observations

- **All succeed**: Simple topology
- **DDPG worst**: Continuous action inefficient
- **Masking helps**: MPPO > PPO
- **GPPO best**: Topology awareness

→ **Modest GNN gain**

5.3 Results: Large-Scale (64 RHs)

Performance

Method	Reward	Cost
DDPG	-1.57	—
PPO	-1.13	—
MPPO	24.28	7,983
GPPO	**52.48**	**6,529**

✗ DDPG & PPO fail (0/6)
! MPPO: 50% success (3/6)
✓ GPPO: 100% success; -18.2% cost

Critical Findings

- **Scalability crisis**: Traditional DRL completely fails
- **Masking insufficient**: MPPO unreliable
- **GNN essential**: Only GPPO achieves perfect reliability
- **Quality**: Best cost among successful methods

→ **Topology awareness critical at scale**

5.4 Multi-Topology Evaluation

Protocol: Train on 5 different 8-RH topologies simultaneously

Cross-Topology Performance

Method	Reward	Cost
MPPO	48.47	261.46
GPPO	**60.57**	**211.80**

✔ +25% reward
✔ -19% cost
✔ Lower variance

Key Insights

- **Robust**: Handles diverse topologies
- **Consistent**: GPPO wins across all structures
- **Transferable**: Graph patterns generalize

→ **Single model for multiple topologies**

6.1 Key Contributions & Results

Main Contributions

1. **GPPO Framework**: GNN + PPO for joint split & placement
2. **Action Masking**: Guarantees constraints, accelerates learning
3. **Scalability**: Proven on 8 & 64 RH networks

→ **Graph awareness enables scalable O-RAN management**

6.2 Future Work

Technical Extensions

- Multi-agent GPPO for distributed control
- Temporal GNNs for traffic prediction
- Transfer learning across operators

Practical Deployment

- Integration with O-RAN RIC
- Real-world testbed validation

Thank You!

Questions?

duc-thinh.ngo@imt-atlantique.fr

7.1 References (I)

[HO20] Shengyi Huang and Santiago Ontañón. “A closer look at invalid action masking in policy gradient algorithms”. In: *arXiv preprint arXiv:2006.14171* (2020).

[Hu+20] W Hu et al. “Strategies For Pre-training Graph Neural Networks”. In: *International Conference on Learning Representations (ICLR)*. 2020.

[Li+24] Haiyuan Li et al. “NetMind: Adaptive RAN Baseband Function Placement by GCN Encoding and Maze-solving DRL”. In: *2024 IEEE Wireless Communications and Networking Conference (WCNC)*. 2024, pp. 1–6.

7.1 References (II)

[Mur+24] Fahri Wisnu Murti et al. “Deep Reinforcement Learning for Orchestrating Cost-Aware Reconfigurations of vRANs”. In: *IEEE Transactions on Network and Service Management* 21.1 (2024), pp. 200–216.

[Sch+17] John Schulman et al. “Proximal Policy Optimization Algorithms”. In: *CoRR* (2017).

7.2 Split Requirements & Trade-offs

Key Observations

- Each split = **2 split options** (high-layer + low-layer)
- Each option has distinct **bandwidth** and **latency** requirements
- Example: S1 uses O2 (λ traffic, 10ms) + O7 (10 Gbps, 0.25ms)

Fundamental Trade-off: Centralization

Advantages

- Lower processing cost
- Simpler deployment
- Better resource pooling

Drawbacks

- Higher bandwidth demand
- Stricter latency constraints
- May violate SLA

→ Need intelligent, adaptive split selection

Duc-Thinh Ngo

IEEE NCA 2025, Lisbon, Portugal, November 5th, 2025

duc-thinh.ngo@imt-atlantique.fr