# Oracle: QoS-aware Online Service Provisioning in Non-Terrestrial Networks with Safe Transfer Learning

**Shengyu Zhang, Songshi Dou, Zhenglong Li, Lawrence K. Yeung, Tony Q.S. Quek**

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

香 港 大 學
THE UNIVERSITY OF HONG KONG

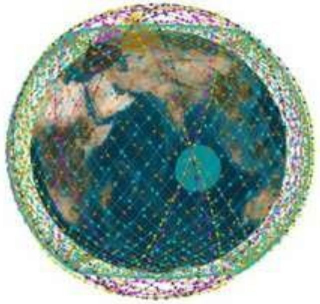
연세대학교
YONSEI UNIVERSITY

May 22, 2025

# Background

- **Utilizing constellations consisting of numerous Low Earth Orbit (LEO) satellites**
  - Increasing number of satellites launched in recent years
  - Providing *Internet service from space*, particularly in remote areas
  - Mega-constellations: consisting of *thousands of satellites* in LEO
- **Many companies are entering the market - SpaceX, OneWeb, Telesat, etc.**
  - Functioning as space-based Internet Service Providers (ISPs)
  - Having the capability to offer pervasive Internet connectivity worldwide
  - For example, as of 2024, SpaceX's Starlink has
    - Over 7,000 Starlink satellites launched
    - More than 4 million subscribers

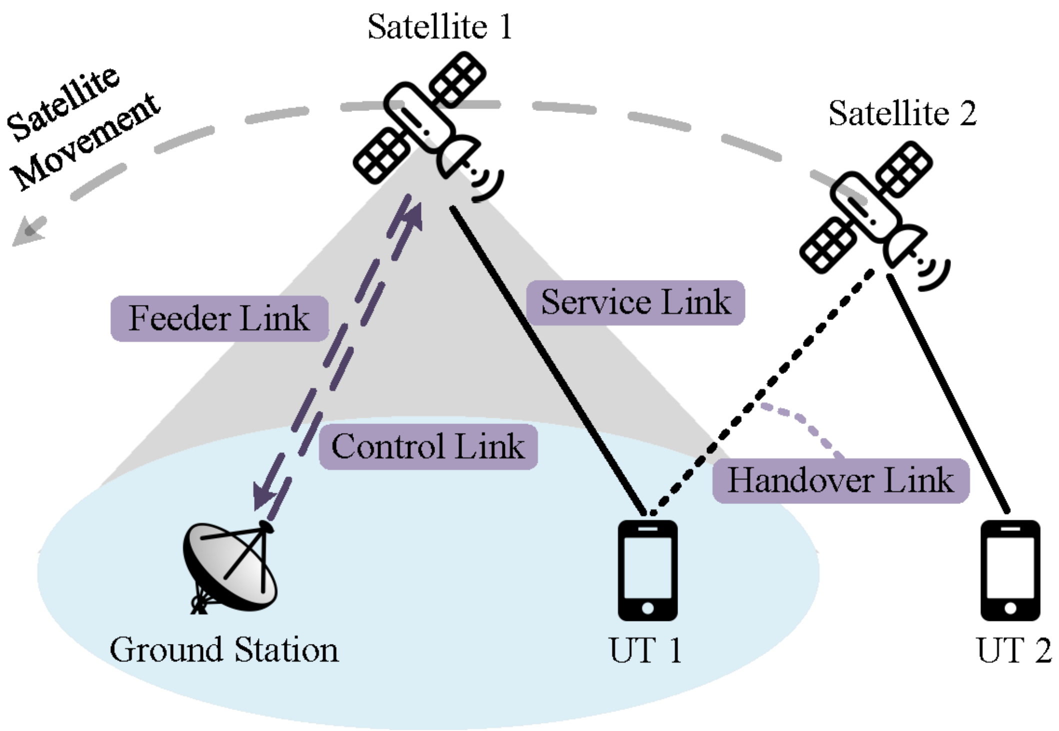
STARLINK



Starlink Constellation

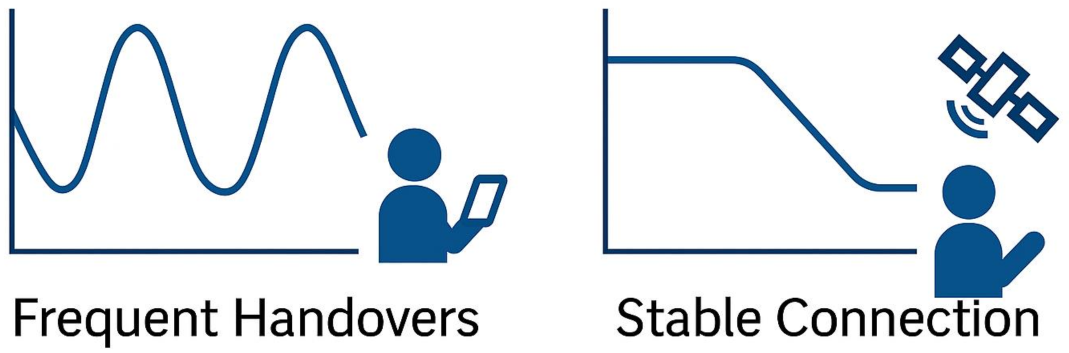
# Motivation

- ⚠ **Why QoS Is Challenging in Non-Terrestrial Networks (NTNs)**
  - High satellite mobility → Frequent handovers every 2–3 minutes
  - Dynamic channel conditions caused by
    - satellite movement
    - atmospheric fading
    - obstructions
  - Measurement shows up to *80% throughput loss* during handovers
  - **Key challenge**: maintaining service continuity and QoS in such environments



# Problem Definition and Challenges

- **QoS-aware Service Provisioning: ⚠ What Makes It Hard**
  - QoS in NTNs = {transmission rate, handover frequency}
  - Tradeoff:
    - Frequent switching → better rate, worse reliability
    - Fewer handovers → more stable, possibly lower throughput
  - Need to optimize this tradeoff in real time under dynamic channel conditions
  - Must meet service constraints: per-user rate, satellite capacity



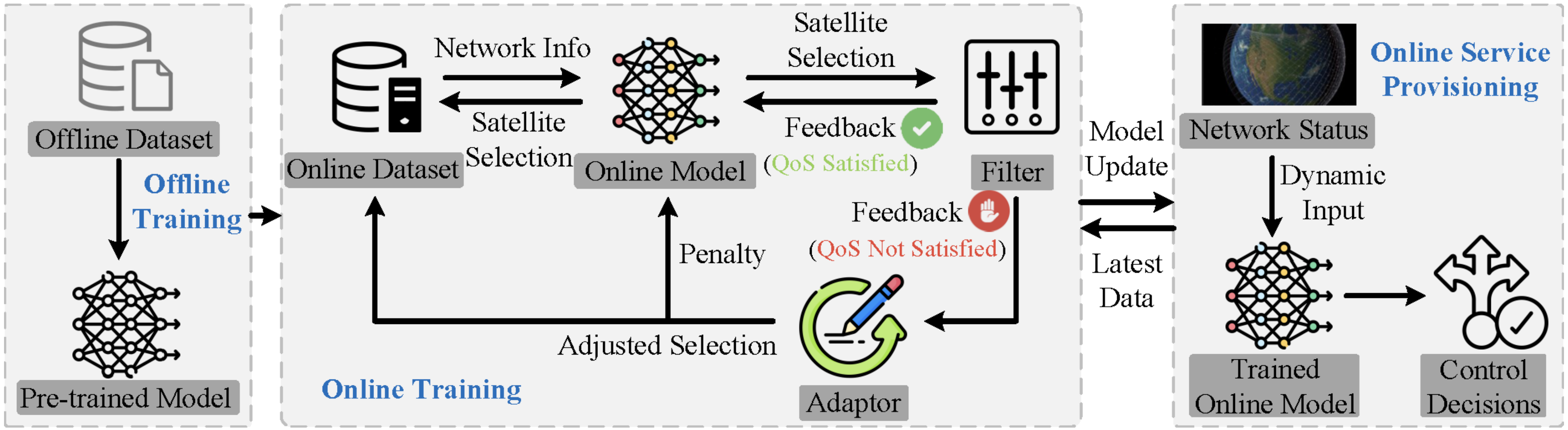
# Limitations of Existing Works

- ⚠ **Why Existing Methods Are Not Enough**
  - Many works optimize throughput or delay but assume *static or slowly varying Channel State Information (CSI)*
  - Offline-trained models (*e.g.*, MADRL) suffer from out-of-distribution issues
  - No capability for *online adaptation*
  - *Infeasible decisions* may violate QoS or system constraints
  - Real-time, constraint-aware, adaptive decision-making is missing

| Capability | Traditional | Offline RL | Oracle |
| --- | --- | --- | --- |
| Handles CSI dynamics | ✗ | ✓ | ✓ |
| Online Adaptation | ✗ | ✗ | ✓ |
| Feasibility Guarantee | ✗ | ✗ | ✓ |

# Oracle Framework Overview

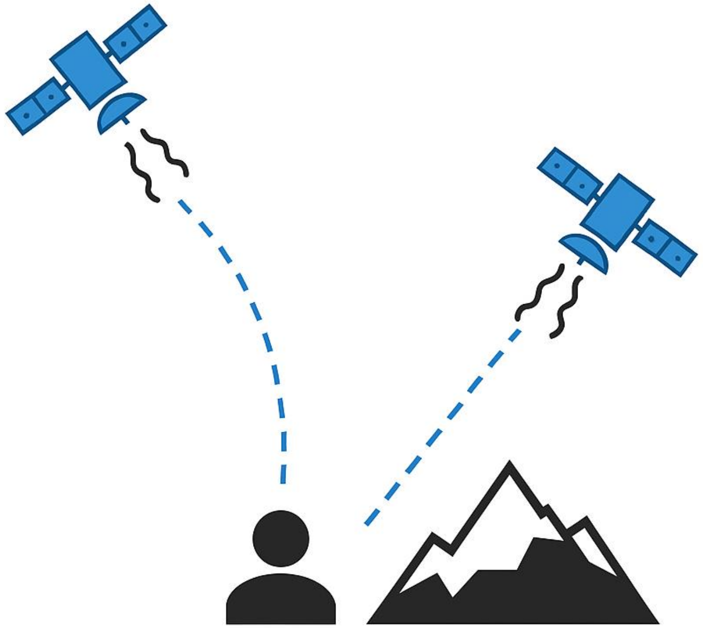
- **Oracle System Architecture**




  - **Offline Training**: General spatial-temporal patterns learned
  - **Online Adaptation**: Few-shot fine-tuning with live data
  - **MPC Controller**: Predicts short-horizon decisions
  - **Safe Transfer Learning**: Ensures feasibility of actions

# System and Channel Model

- **Channel Model in Dynamic NTN Environment**
  - ⚠ Why is this modeling necessary?
    - High satellite velocity → frequent Doppler shift
    - Weather and obstructions → highly variable loss
    - Mobility and rotation → dynamic antenna gain
    - → All of these require frame-level CSI modeling
  - Signal model: $y_{k,m}(t) = \mathbf{h}_{k,m}^{H}(t) P_k \mathbf{s}_k(t) + n_k$
  - Channel coefficient includes:
    - Antenna gains
    - Free-space path loss
    - Atmospheric fading
    - Shadowed-Rician fading
  - SNR: $\gamma_{k,m}(t) = \frac{P_k \left\| \mathbf{h}_{k,m}(t) \right\|^2}{\sigma_k^2}$
  - Rate: $R_{k,m}(t) = B \log_2\left(1 + \gamma_{k,m}(t)\right)$

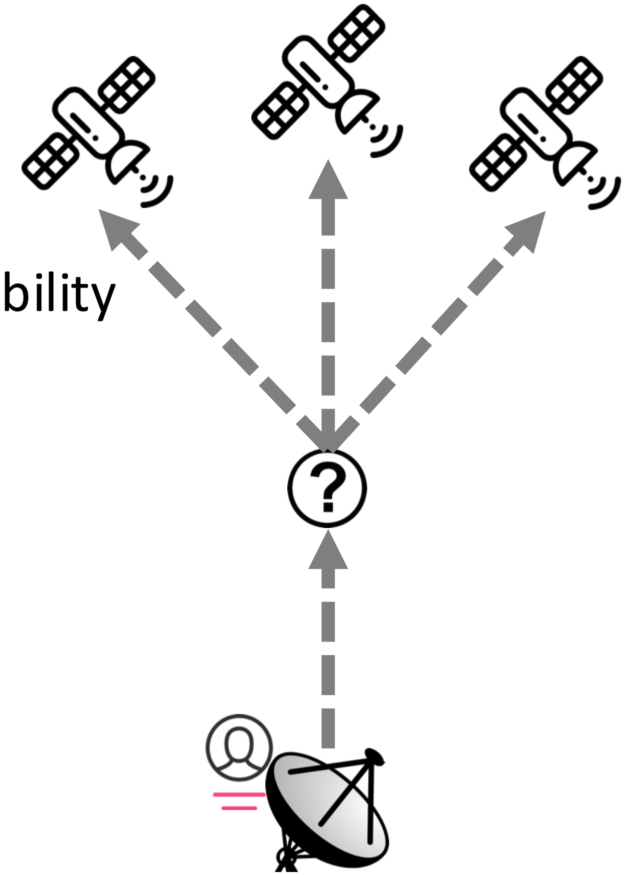


# QoS Optimization Problem

- **Formulation of the QoS-Aware Provisioning Problem**
  - Objective Function:

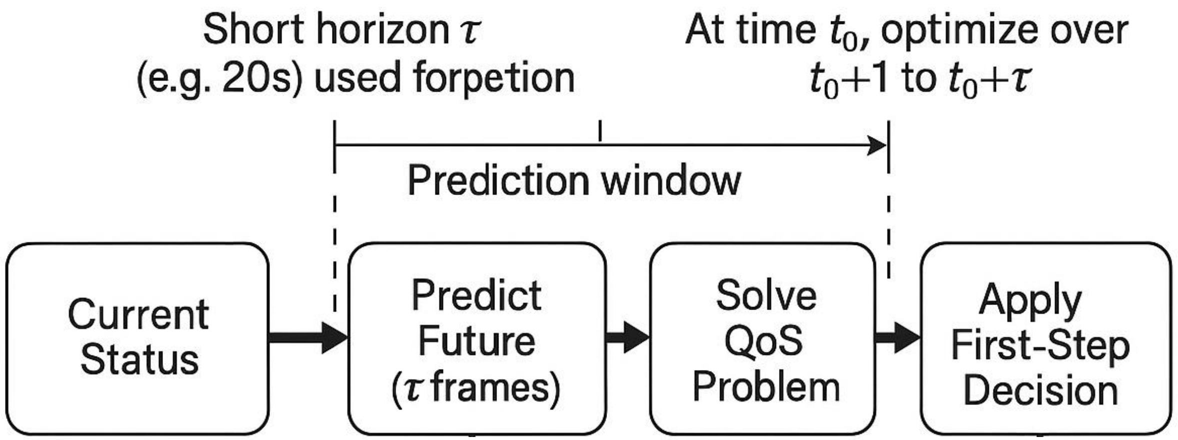
    $$\Phi = \lambda \cdot \sum x_{k,m}(t) \cdot \hat{R}_{k,m}(t) - (1-\lambda) \cdot \text{HandoverCost}$$

    - First term → encourages high data rate assignments
    - Second term → penalizes frequent satellite switching
    - λ ∈ [0, 1] allows system to control trade-off between rate and stability
  - Constraints:
    - Each user connects to at most one satellite
    - Minimum rate requirement: $\sum_{m} x_{k,m}(t) \cdot \hat{R}_{k,m}(t) \geq R_k^{\text{th}}(t)$
    - Satellite capacity: $\sum_{k} x_{k,m}(t) \cdot \hat{R}_{k,m}(t) \leq C_m$
  - Explain λ:
    - λ = 0 → minimize handovers
    - λ = 1 → maximize rate



# Model Predictive Control (MPC)

- **Short-Horizon Optimization with MPC**
  - Short horizon τ (*e.g.*, 20s) used for prediction
  - At time $t_0$, optimize over $t_0$+1 to $t_0$+τ
  - Apply only the first decision $x(t_0 + 1)$
  - Repeat process at each new time frame with updated status

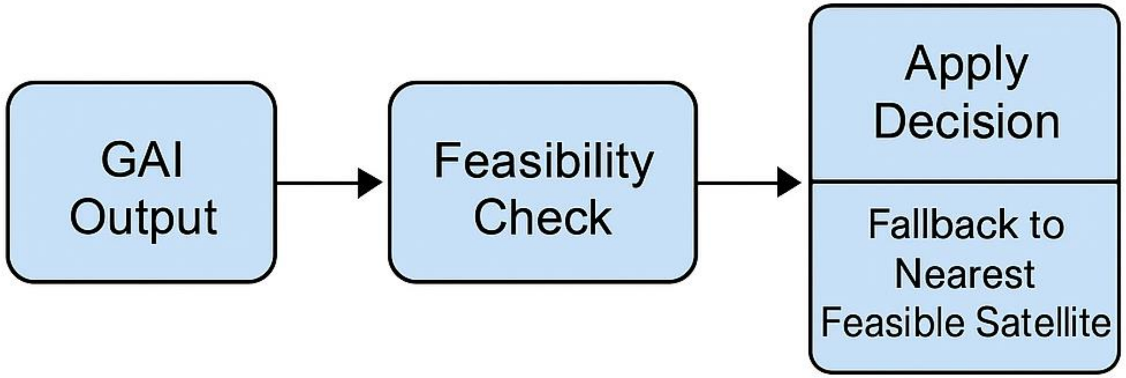


# Model Predictive Control (MPC)

- **Extend-MPC Algorithm - Using History to Enhance Prediction**
  - Basic MPC only uses current network state
    - → Prone to reacting to transient fluctuations
    - → Misses underlying temporal trends (e.g., motion patterns, load buildup)
  - History-enhanced MPC = Extend-MPC
    1) Collect historical state vector at time frame $t_0$
    2) Predict network state over short horizon $\tau$
    3) Solve the same MPC problem as in Slide 9
    4) Apply only the first decision
    5) Slide the window forward → repeat at next frame
  - Benefits
    - Allows model to capture trends
    - Reduces over-reaction to short-term noise
    - Achieves better balance between reactiveness and stability

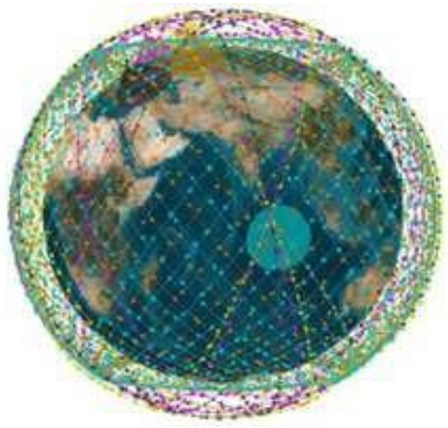
# Safe Transfer Learning Framework

- **Fast Decision Making with Safe Generative AI**
  - GAI predicts decisions over control window directly
  - Safety filter ensures:
    - Minimum QoS (Does it violate rate requirement?)
    - Capacity constraints (Does it exceed satellite capacity?)
  - If invalid → fallback to nearest feasible satellite
  - Loss function with penalty terms: $\mathrm{L} = -\Phi + \mu_1 \cdot (\text{QoS violation}) + \mu_2 \cdot (\text{Capacity overload})$



# Simulation Setup

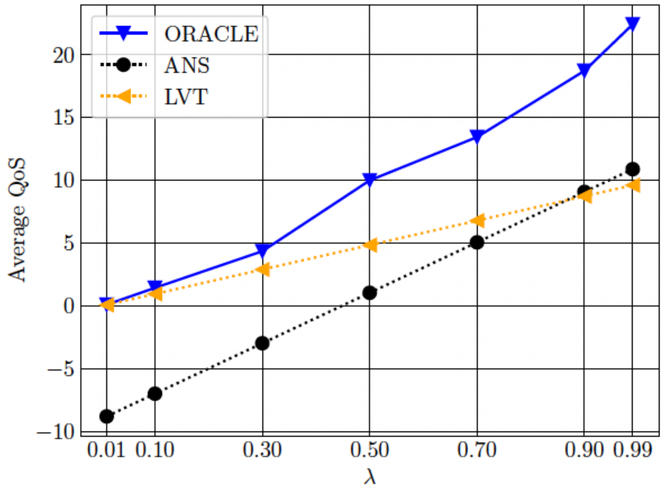
- **Evaluation Setup Based on Starlink Constellation**
  - Key Parameters:
    - Carrier frequency: 28 GHz
    - Bandwidth: 500 MHz
    - Satellites: 1584 (72 orbits × 22 satellites)
    - Users: 10 UTs
    - Horizon: $\tau$ = 20s, History: $T_0$ = 10s
    - Weather: $\chi$ ~ Uniform[0, 4]



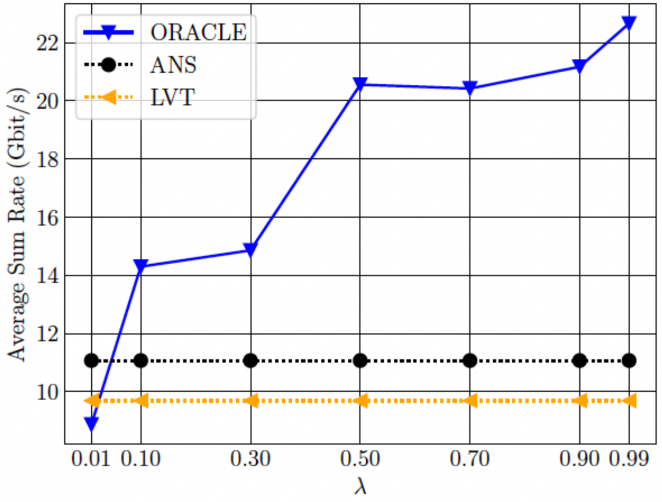
- **Benchmark algorithm**
  - Always Nearest Satellite (**ANS**)
    - This scheme always finds the nearest satellite for each user and assigns each user to its nearest satellite.
  - Longest Visible Time (**LVT**)
    - This scheme finds the satellite with the longest remaining visible time.

# Results – λ Sensitivity

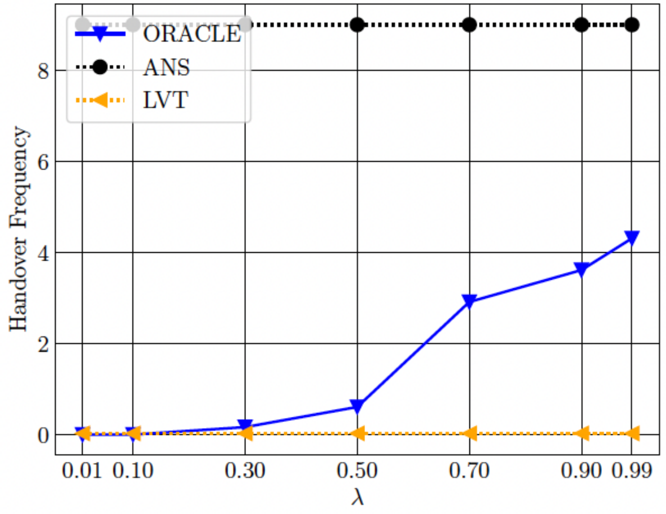
- **Oracle Performance Across QoS Priorities (λ)**
  - Fig. a: QoS
  - Fig. b: Sum Rate
  - Fig. c: Handover Frequency




(a) Performance of average QoS.




(b) Performance of average sum rates.

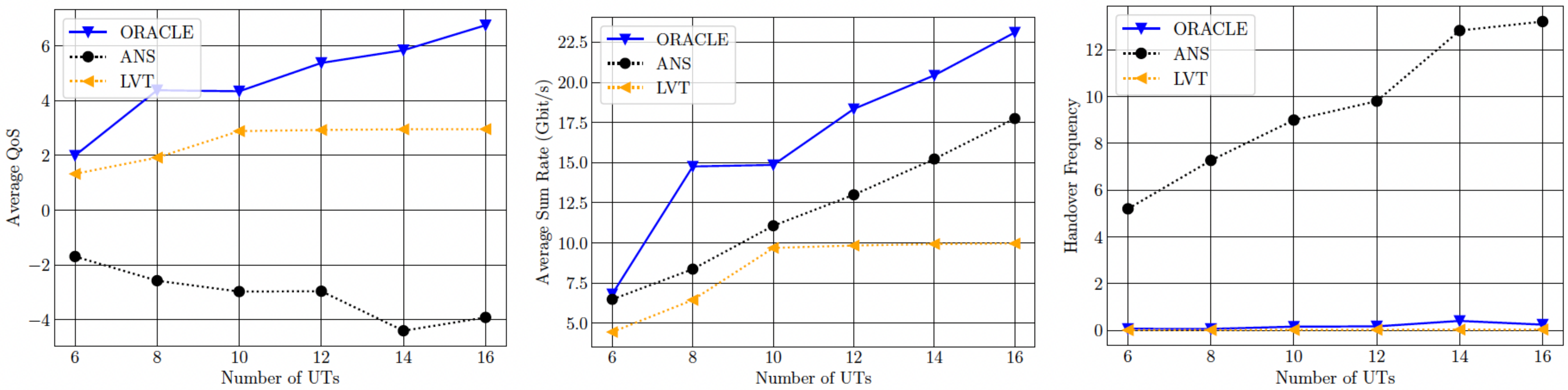



(c) Performance of handover frequency.

- **Summary:**
  - Oracle matches or outperforms ANS and LVT across various settings
  - Dynamically balances performance depending on λ

# Results – Scalability

- **Oracle Performance Across Different Number of UTs**
  - Fig. a: QoS
  - Fig. b: Sum Rate
  - Fig. c: Handover Frequency

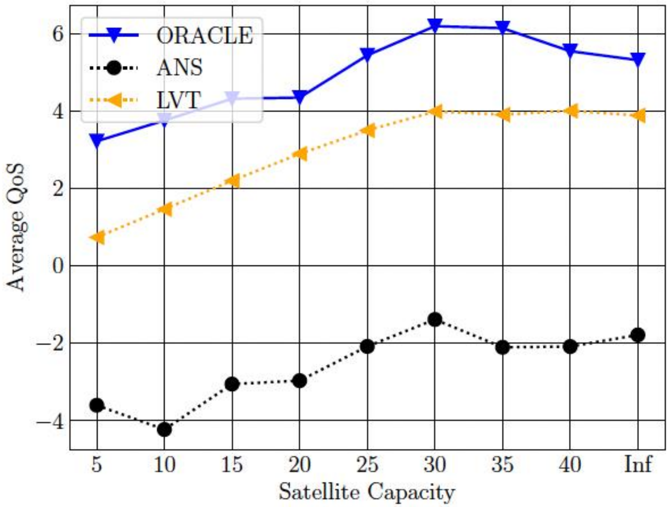



(a) Performance of average QoS. (b) Performance of average sum rates. (c) Performance of handover frequency.

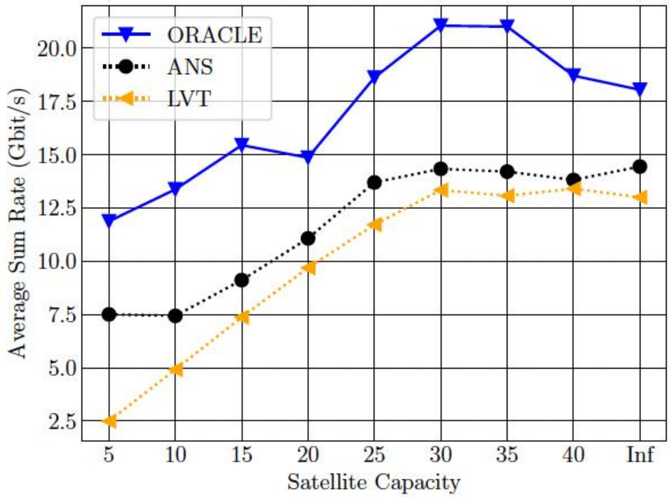
- **Summary:**
  - As the number of UTs increases, ORACLE consistently achieves higher QoS.
  - This demonstrates Oracle's scalability and robustness under increasing load.

# Results – Robustness

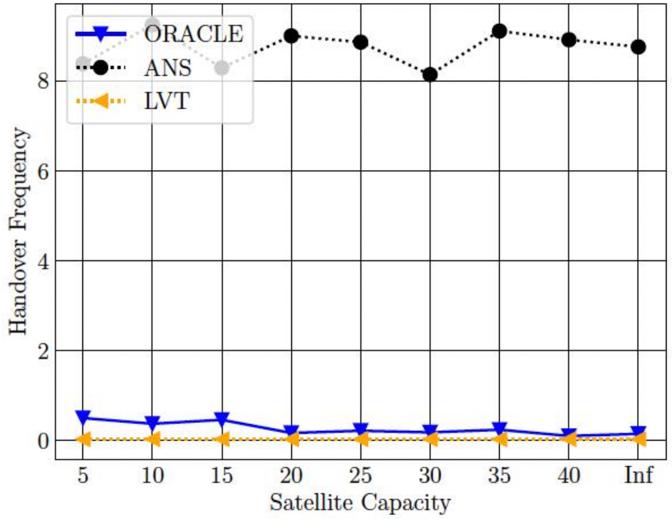
- **Oracle Performance Across Different Satellite Capacities**
  - Fig. a: QoS
  - Fig. b: Sum Rate
  - Fig. c: Handover Frequency




(a) Performance of average QoS.




(b) Performance of average sum rates.

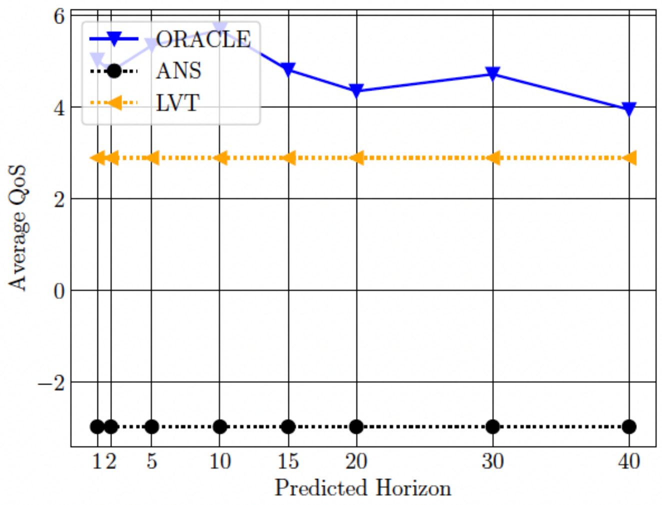



(c) Performance of handover frequency.

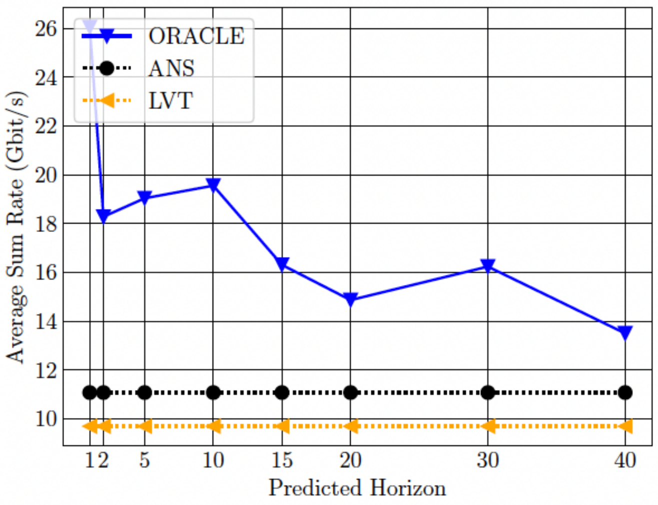
- **Summary:**
  - When capacity is constrained, performance gaps are larger, showing Oracle's ability to optimize resource allocation under scarcity.

# Results – Prediction Horizon

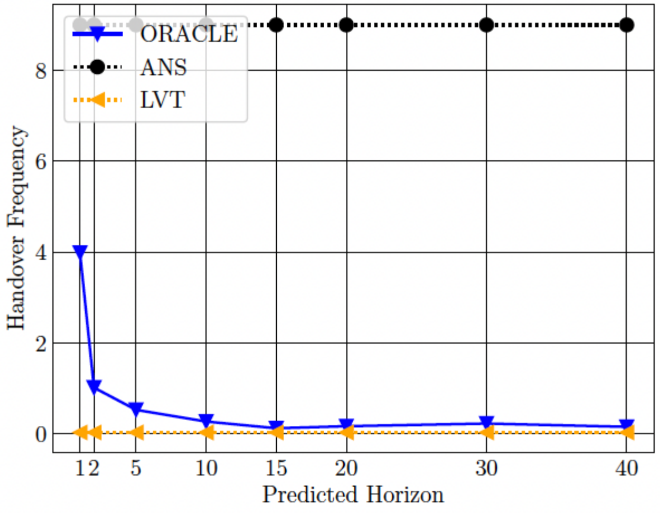
- **Oracle Performance Across Different Prediction Horizon (τ)**
  - Fig. a: QoS
  - Fig. b: Sum Rate
  - Fig. c: Handover Frequency




(a) Performance of average QoS




(b) Performance of average sum rates.




(c) Performance of handover frequency.

- **Summary:**
  - This shows Oracle's sensitivity to prediction horizon and supports the choice of a short, controlled horizon in MPC.

# Conclusion

- **Oracle Summary:**
  - A fully online, adaptive framework for *QoS-aware service provisioning* in NTNs
  - Integrates *Model Predictive Control* and *Safe Transfer Learning*
  - Designed for *real-time*, *constraint-aware*, and *scalable* deployment
- **Achievements:**
  - Up to *3× improvement in QoS* over ANS and LVT baselines
  - Maintains stable performance under varying:
    - User loads
    - Satellite capacity limits
    - QoS preference (via $\lambda$)
- **Benefits:**
  - *Online adaptation* to time-varying CSI
  - *Tradeoff optimization* between handover and throughput
  - *Safe decisions* under real-world system constraints

# Thanks for your attention!

# Q&A