Decentralized Online Convex Optimization in Networked Systems

Yiheng Lin, Judy Gan, Guannan Qu, Yash Kanoria, and Adam Wierman

Thirty-ninth International Conference on Machine Learning

Selling a single product

*Picture credits: HP.

Selling a single product

Price at time t: x_t

*Picture credits: HP.

Selling a single product

Price at time t: x_t

Demand at time t: $d_t(x_t)$

*Picture credits: HP.

Selling a single product

Price at time t: x_t

Demand at time t: $d_t(x_t)$

Revenue at time t: $d_t(x_t) \cdot x_t$

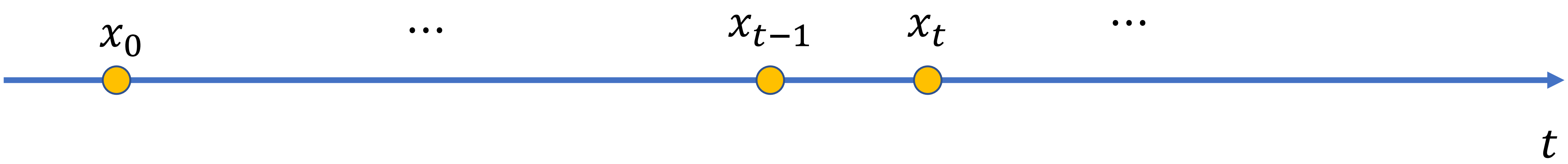
*Picture credits: HP.

Selling a single product

Price at time t: x_t

Demand at time t: $d_t(x_t)$

Revenue at time t: $d_t(x_t) \cdot x_t$



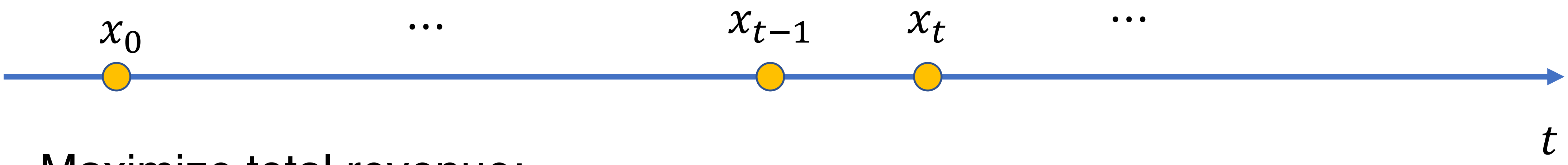
*Picture credits: HP.

Selling a single product

Price at time t: x_t

Demand at time t: $d_t(x_t)$

Revenue at time t: $d_t(x_t) \cdot x_t$



Maximize total revenue:

$$\max_{x_t} \sum_{t=1}^{T} d_t(x_t) \cdot x_t$$

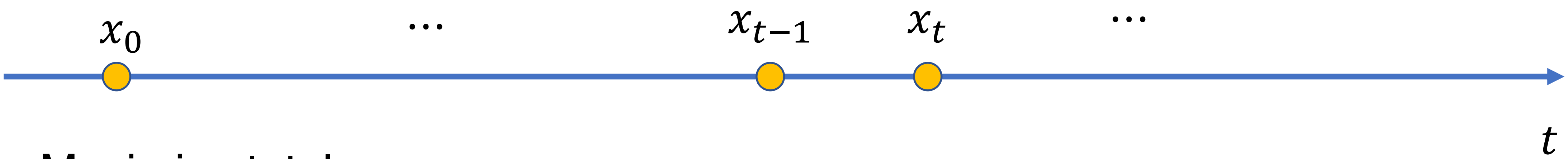
*Picture credits: HP.

Selling a single product

Price at time t: x_t

Demand at time t: $d_t(x_t)$

Revenue at time t: $d_t(x_t) \cdot x_t$



Maximize total revenue:

$$\max_{x_t} \sum_{t=1}^{T} d_t(x_t) \cdot x_t$$

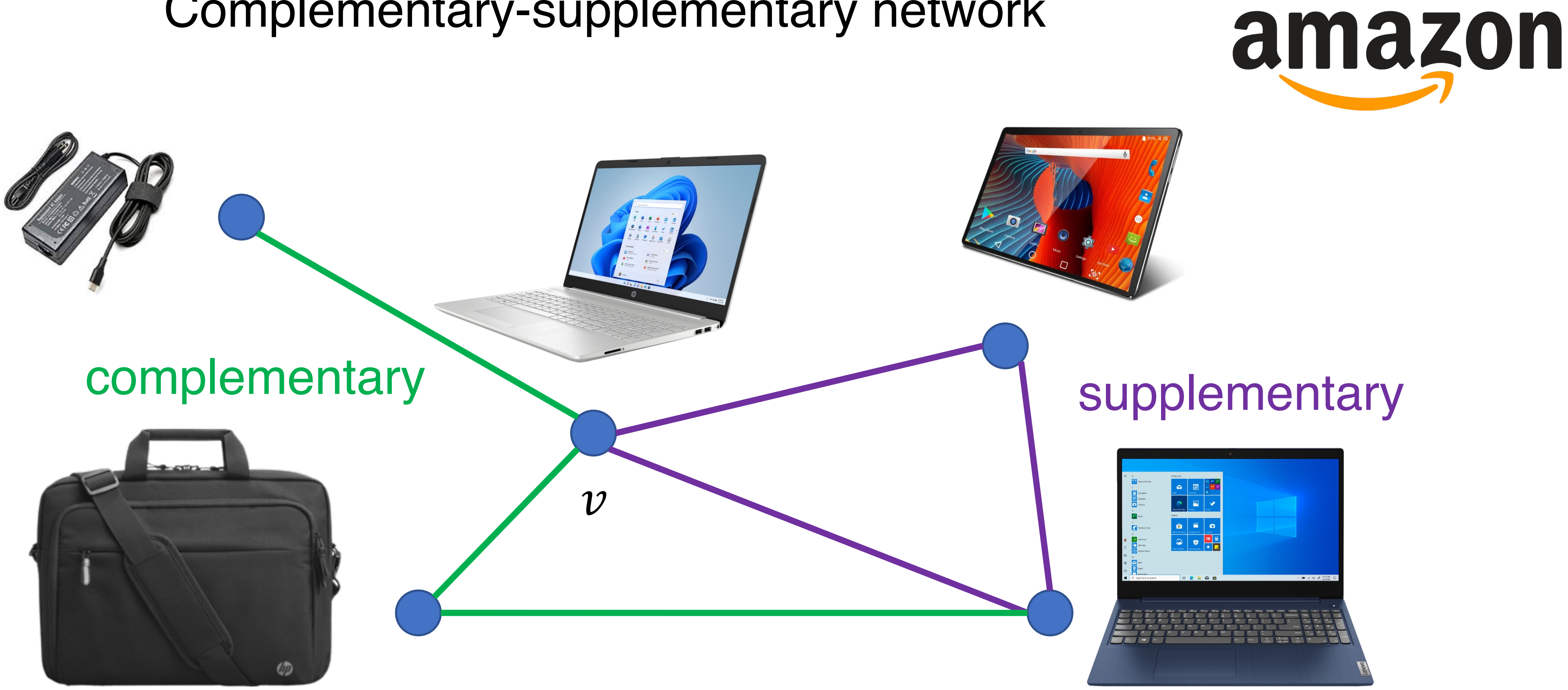
Online Convex Optimization (OCO):

$$\min_{x_t} \sum_{t=1}^{T} f_t(x_t)$$

*Picture credits: HP.

Selling multiple products with temporal dynamics

Complementary-supplementary network

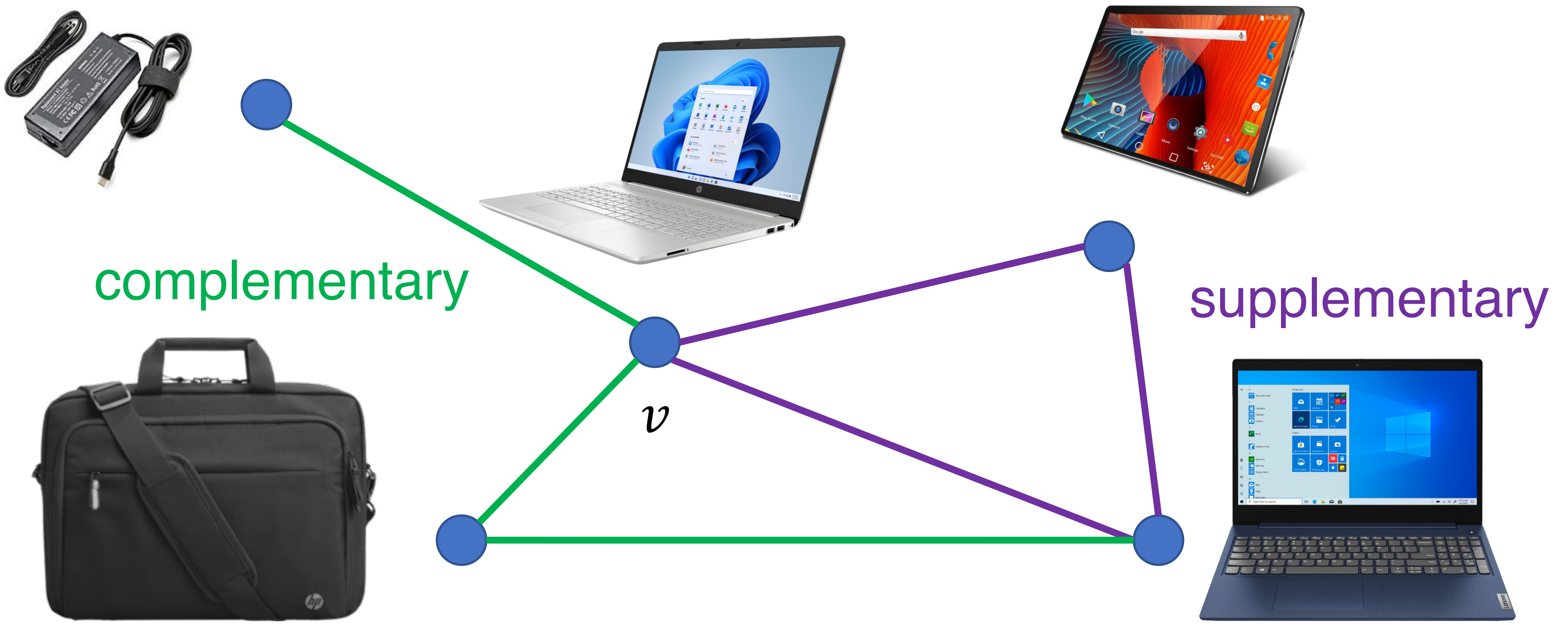


*Picture credits: HP, Walmart, Amazon.

Selling multiple products with temporal dynamics

Complementary-supplementary network

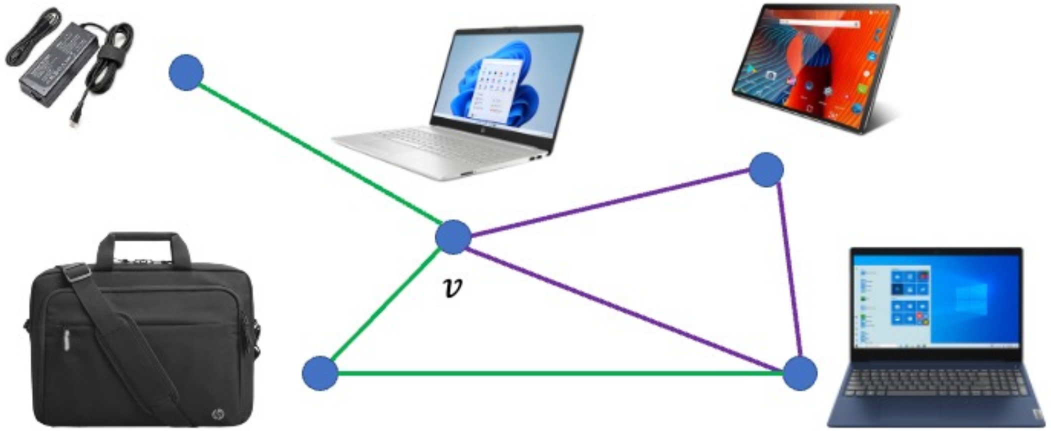
amazon



The network can be **huge**: More than 12 million products on Amazon!

*Picture credits: HP, Walmart, Amazon.

Selling multiple products with temporal dynamics

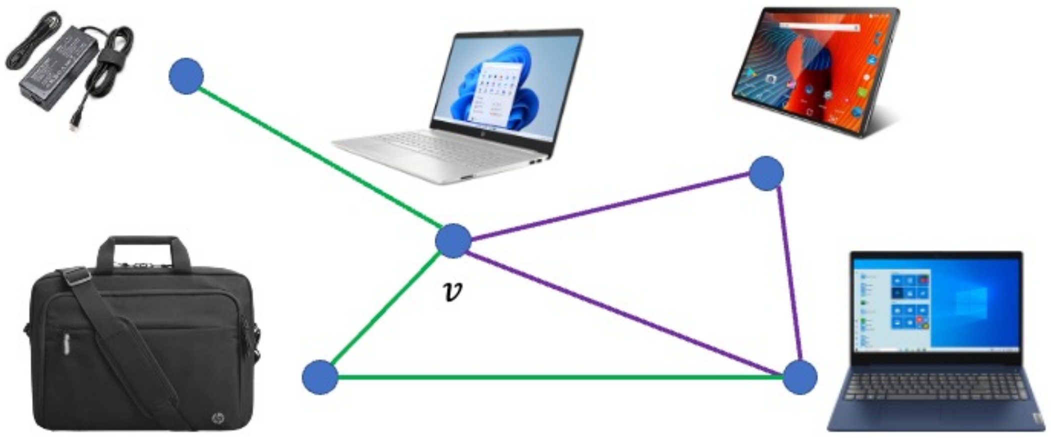


For each product v:

Price at time t: x_t^v

*Picture credits: HP, Walmart, Amazon.

Selling multiple products with temporal dynamics



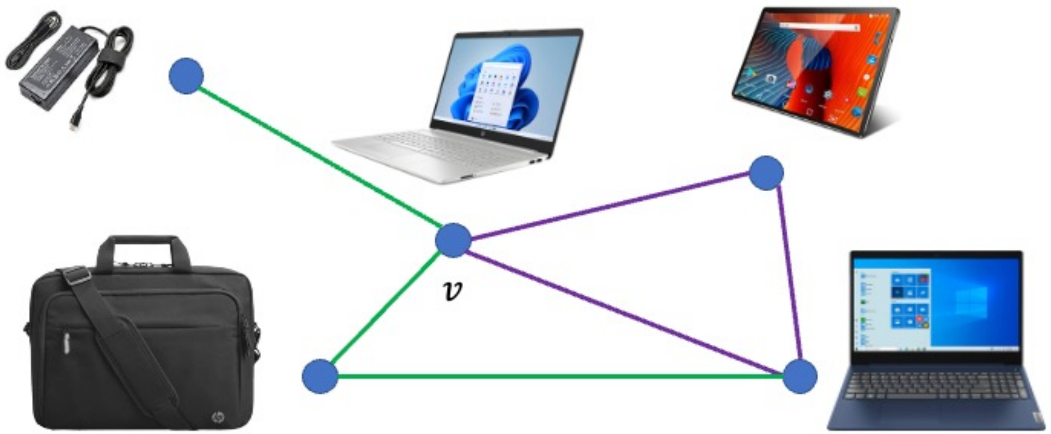
For each product v:

Price at time t: x_t^v

Demand at time t: $d_t^v\left(x_t^{N_v}, x_{t-1}^v\right)$

*Picture credits: HP, Walmart, Amazon.

Selling multiple products with temporal dynamics



For each product v:

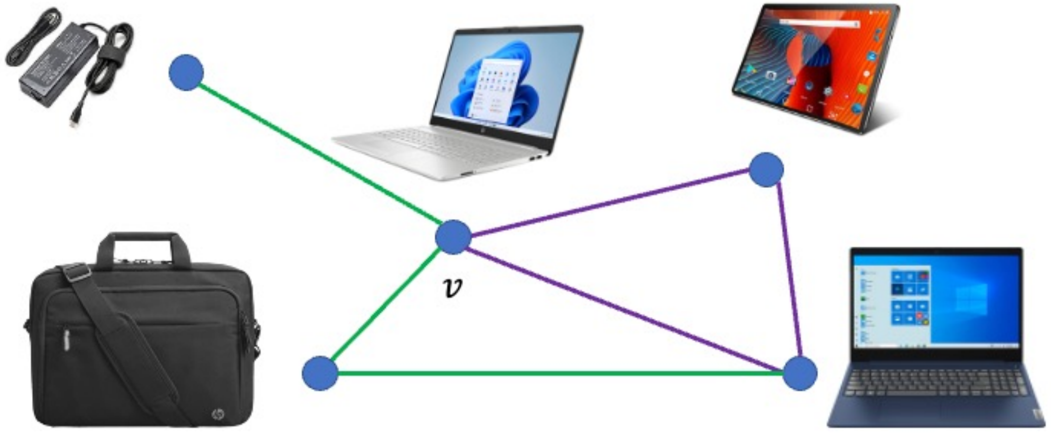
Price at time t: x_t^v

Neighbors of v

Demand at time t: $d_t^v\left(x_t^{N_v}, x_{t-1}^v\right)$

*Picture credits: HP, Walmart, Amazon.

Selling multiple products with temporal dynamics



For each product v:

Price at time t: x_t^v

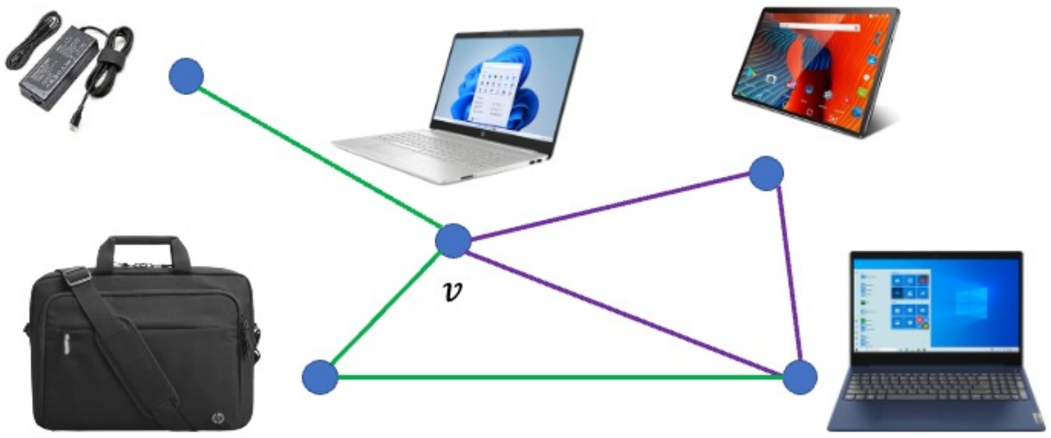
Neighbors of v

Demand at time t: $d_t^v\left(x_t^{N_v}, x_{t-1}^v\right)$

Revenue at time t: $d_t^v\left(x_t^{N_v}, x_{t-1}^v\right) \cdot x_t^v$

*Picture credits: HP, Walmart, Amazon.

Selling multiple products with temporal dynamics



For each product v:

Price at time t: x_t^v

Neighbors of v

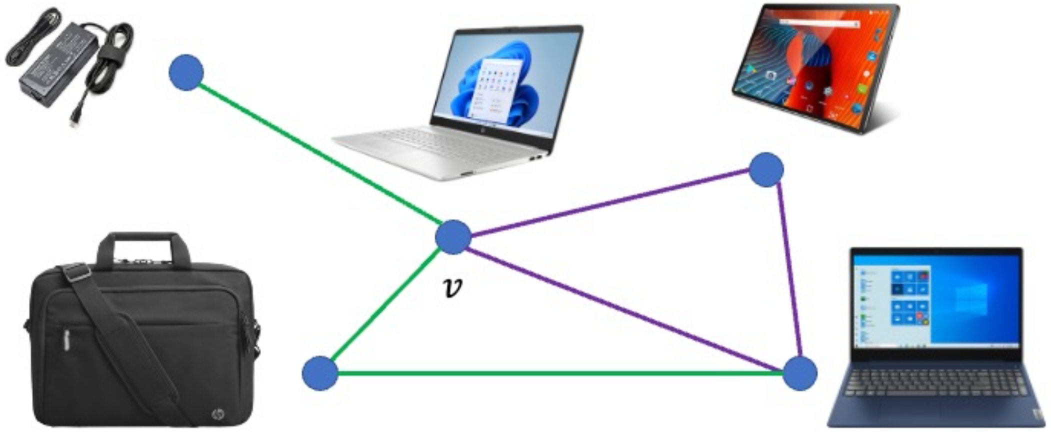
Demand at time t: $d_t^v\left(x_t^{N_v}, x_{t-1}^v\right)$

Revenue at time t: $d_t^v\left(x_t^{N_v}, x_{t-1}^v\right) \cdot x_t^v$

Goal: Maximize the total revenue of all products.

*Picture credits: HP, Walmart, Amazon.

Selling multiple products with temporal dynamics



For each product v:

Price at time t: x_t^v

Neighbors of v

Demand at time t: $d_t^v(x_t^{N_v}, x_{t-1}^v)$

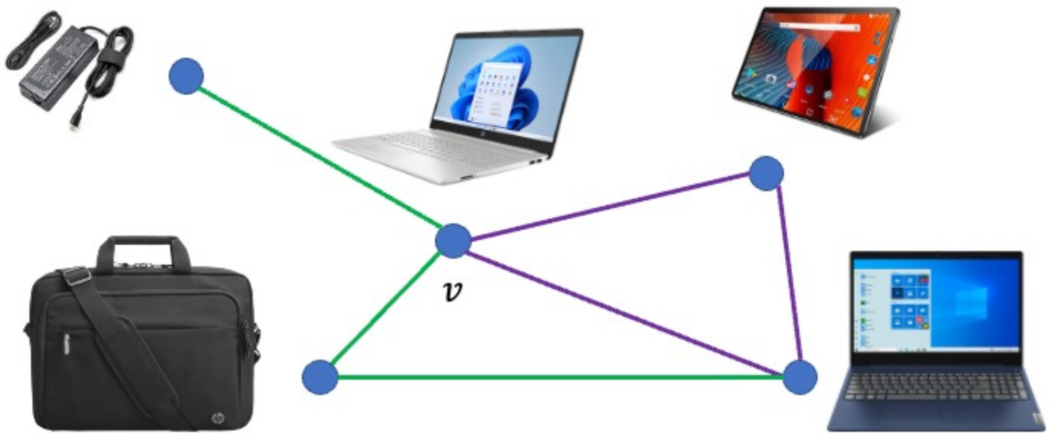
Revenue at time t: $d_t^v(x_t^{N_v}, x_{t-1}^v) \cdot x_t^v$

Goal: Maximize the total revenue of all products.

Applications: Selling a single product + multiple products & temporal dynamics

*Picture credits: HP, Walmart, Amazon.

Selling multiple products with temporal dynamics



For each product v:

Price at time t: x_t^v

Neighbors of v

Demand at time t: $d_t^v\left(x_t^{N_v}, x_{t-1}^v\right)$

Revenue at time t: $d_t^v\left(x_t^{N_v}, x_{t-1}^v\right) \cdot x_t^v$

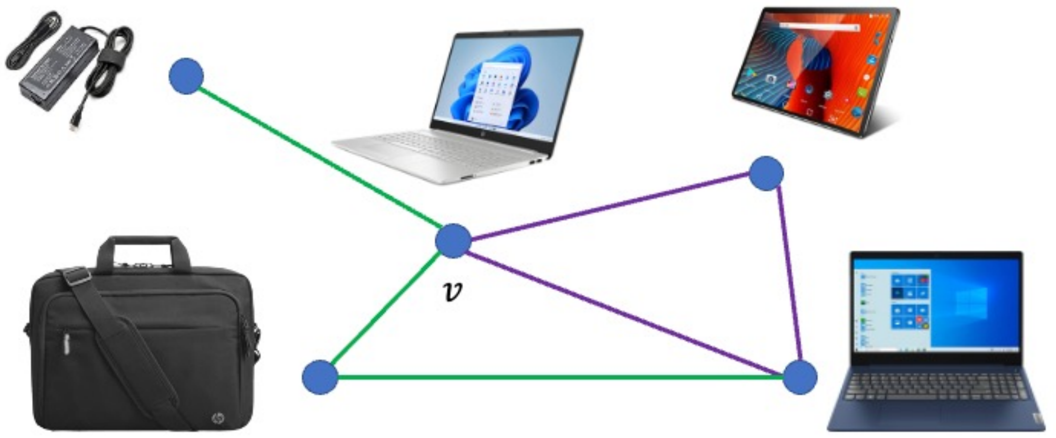
Goal: Maximize the total revenue of all products.

Applications: Selling a single product + multiple products & temporal dynamics

Tools: OCO

*Picture credits: HP, Walmart, Amazon.

Selling multiple products with temporal dynamics



For each product v:

Price at time t: x_t^v

Neighbors of v

Demand at time t: $d_t^v\left(x_t^{N_v}, x_{t-1}^v\right)$

Revenue at time t: $d_t^v\left(x_t^{N_v}, x_{t-1}^v\right) \cdot x_t^v$

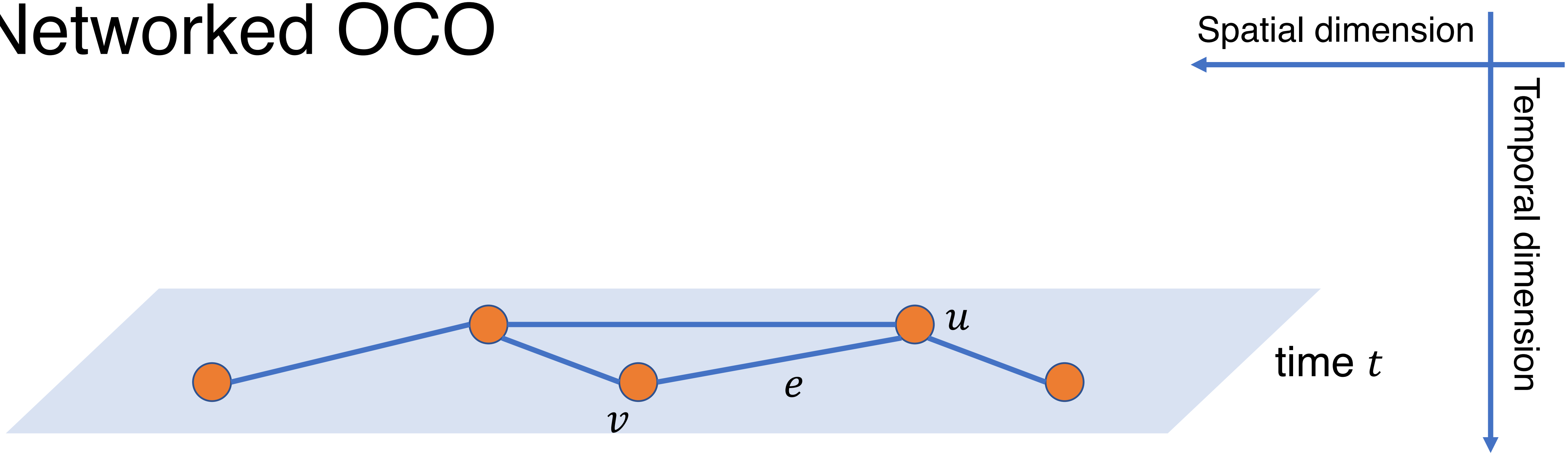
Goal: Maximize the total revenue of all products.

Applications: Selling a single product | + multiple products & temporal dynamics

Tools: OCO $\longrightarrow$ Networked OCO

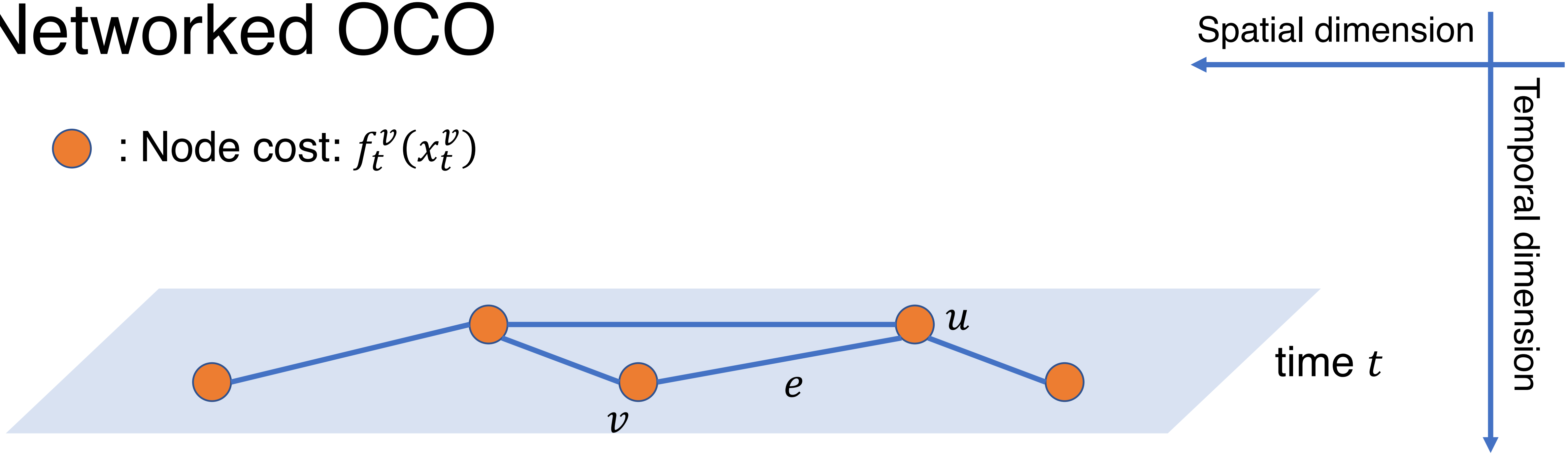
*Picture credits: HP, Walmart, Amazon.

Networked OCO



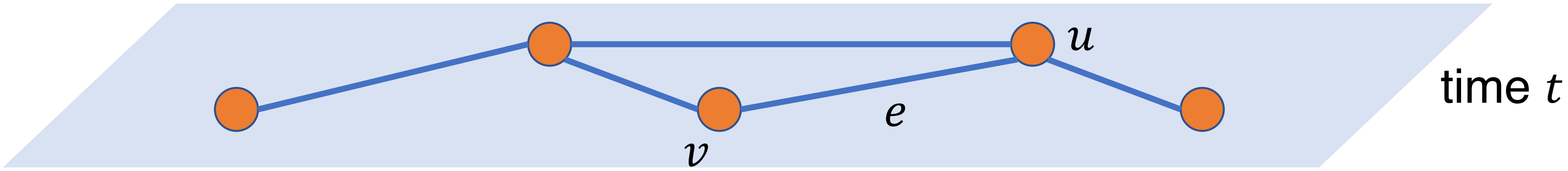
Networked OCO

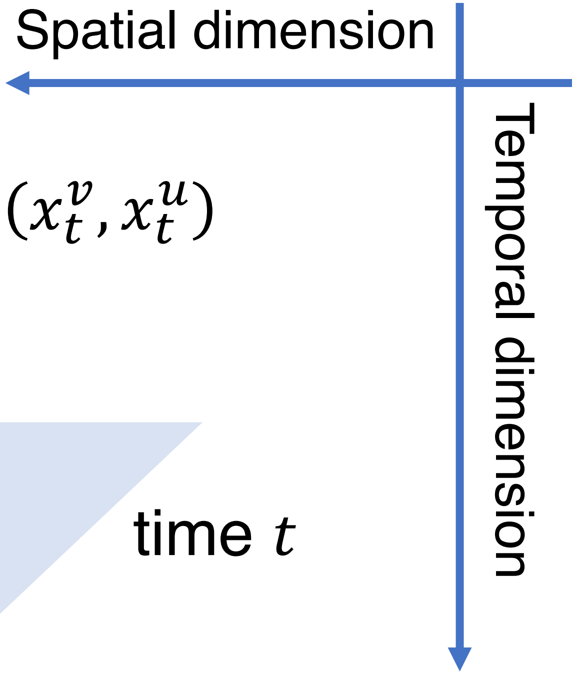
: Node cost: $f_t^v(x_t^v)$



Networked OCO

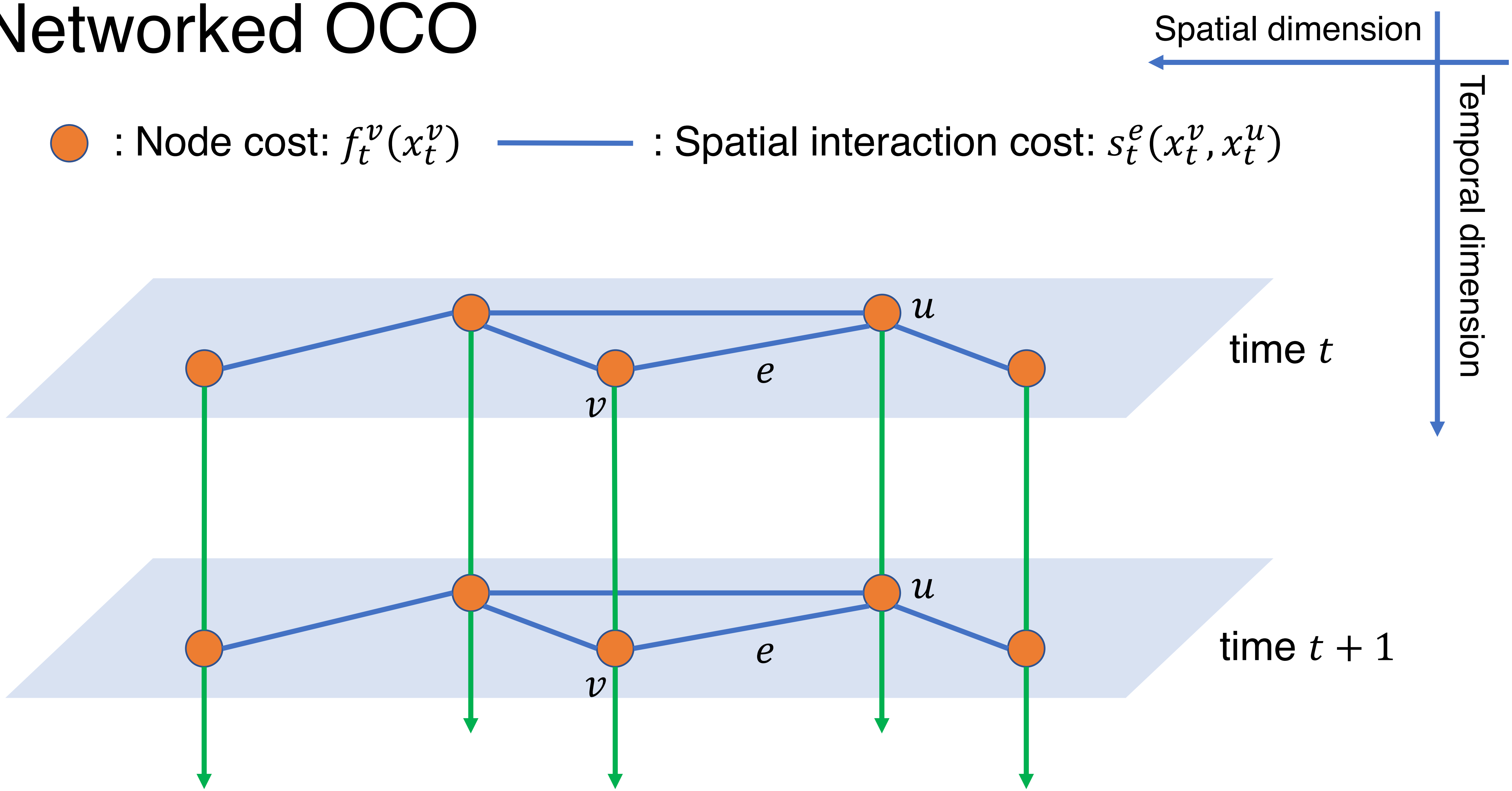
: Node cost: $f_t^v(x_t^v)$: Spatial interaction cost: $s_t^e(x_t^v, x_t^u)$



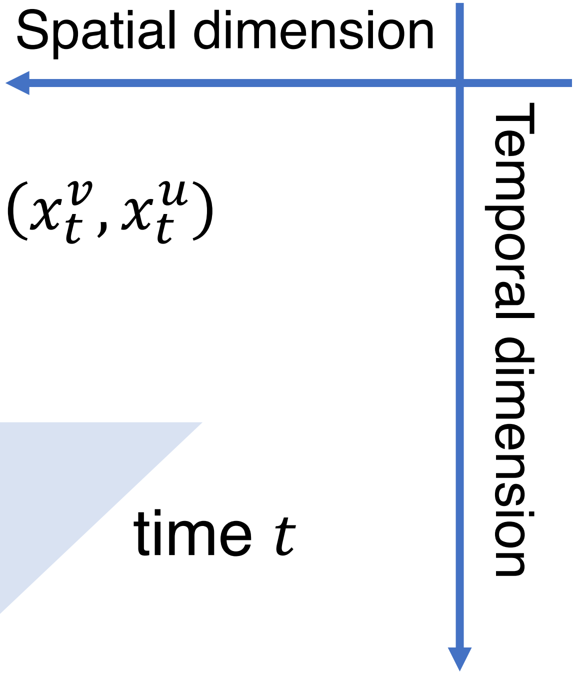


Networked OCO

: Node cost: $f_t^v(x_t^v)$: Spatial interaction cost: $s_t^e(x_t^v, x_t^u)$



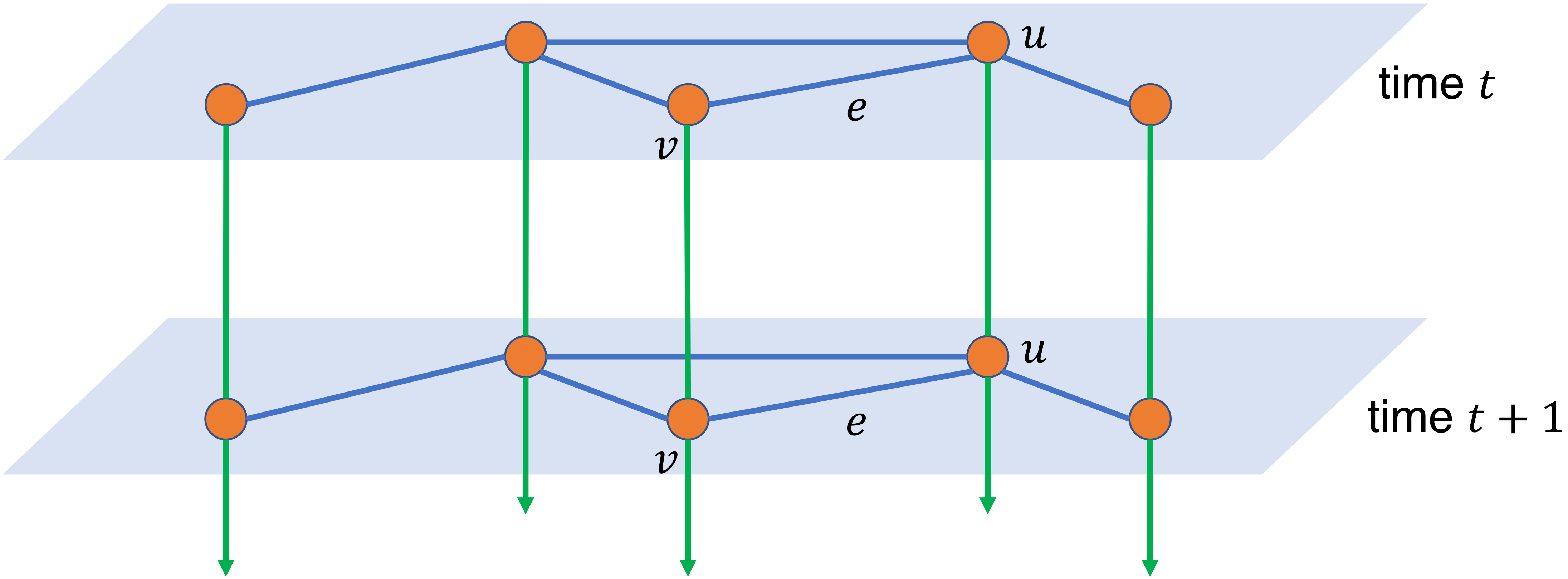
Networked OCO



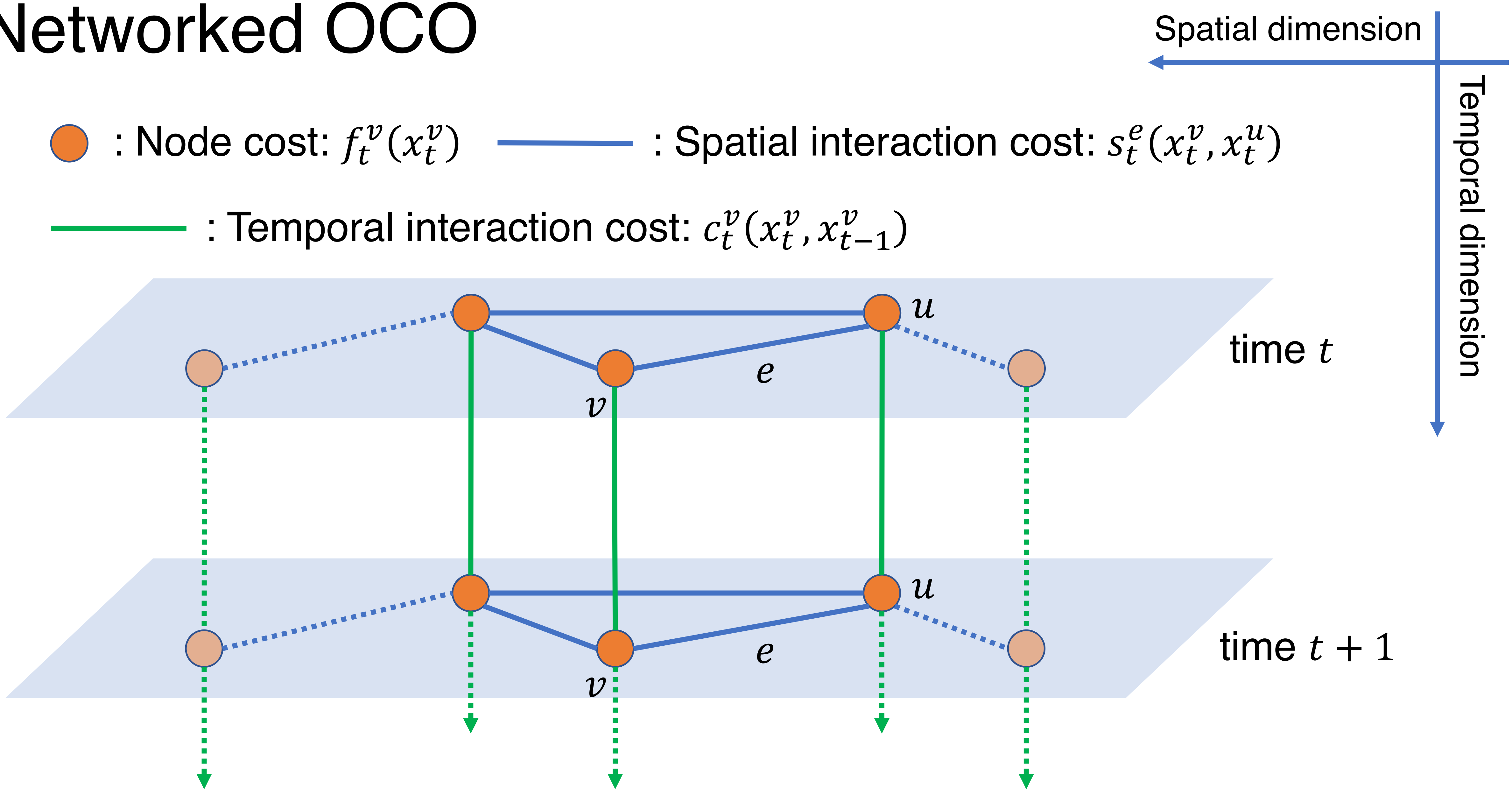
: Node cost: $f_t^v(x_t^v)$

: Spatial interaction cost: $s_t^e(x_t^v, x_t^u)$

: Temporal interaction cost: $c_t^v(x_t^v, x_{t-1}^v)$



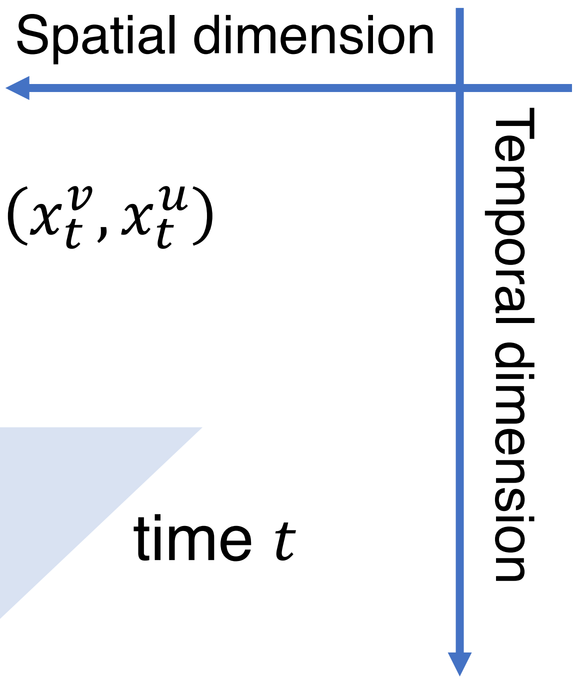
Networked OCO

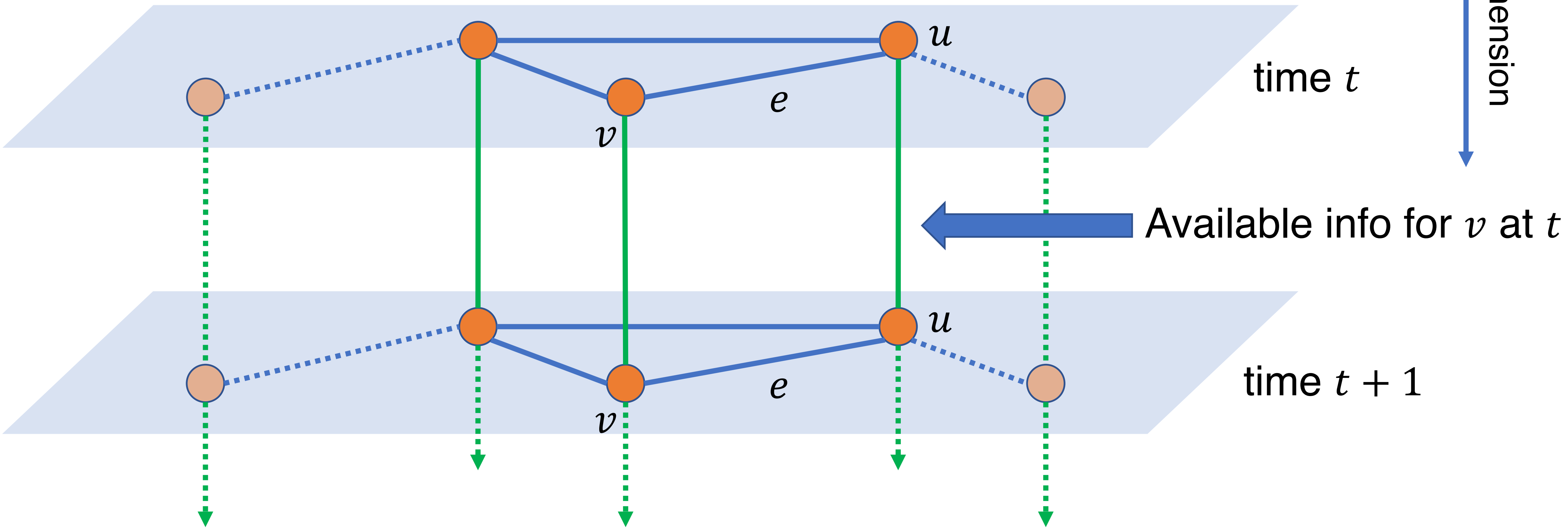


Networked OCO

: Node cost: $f_t^v(x_t^v)$: Spatial interaction cost: $s_t^e(x_t^v, x_t^u)$

: Temporal interaction cost: $c_t^v(x_t^v, x_{t-1}^v)$





Main Results

Theorem (upper bound): We proposed a decentralized online algorithm, Localized Predictive Control, that achieves the competitive ratio of $1 + O\left({\rho_T}^k\right) + O({\rho_S}^r)$.

Main Results

Theorem (upper bound): We proposed a decentralized online algorithm, Localized Predictive Control, that achieves the competitive ratio of $1 + O\left({\rho_T}^k\right) + O\left({\rho_S}^r\right)$.

k: Prediction horizon. r: Communication radius.

Main Results

Theorem (upper bound): We proposed a decentralized online algorithm, Localized Predictive Control, that achieves the competitive ratio of $1 + O\left({\rho_T}^k\right) + O({\rho_S}^r)$.

k: Prediction horizon. r: Communication radius.

Theorem (lower bound): The competitive ratio of any decentralized online algorithm is lower bounded by $1 + \Omega\left({\lambda_T}^k\right) + \Omega({\lambda_S}^r)$.

Main Results

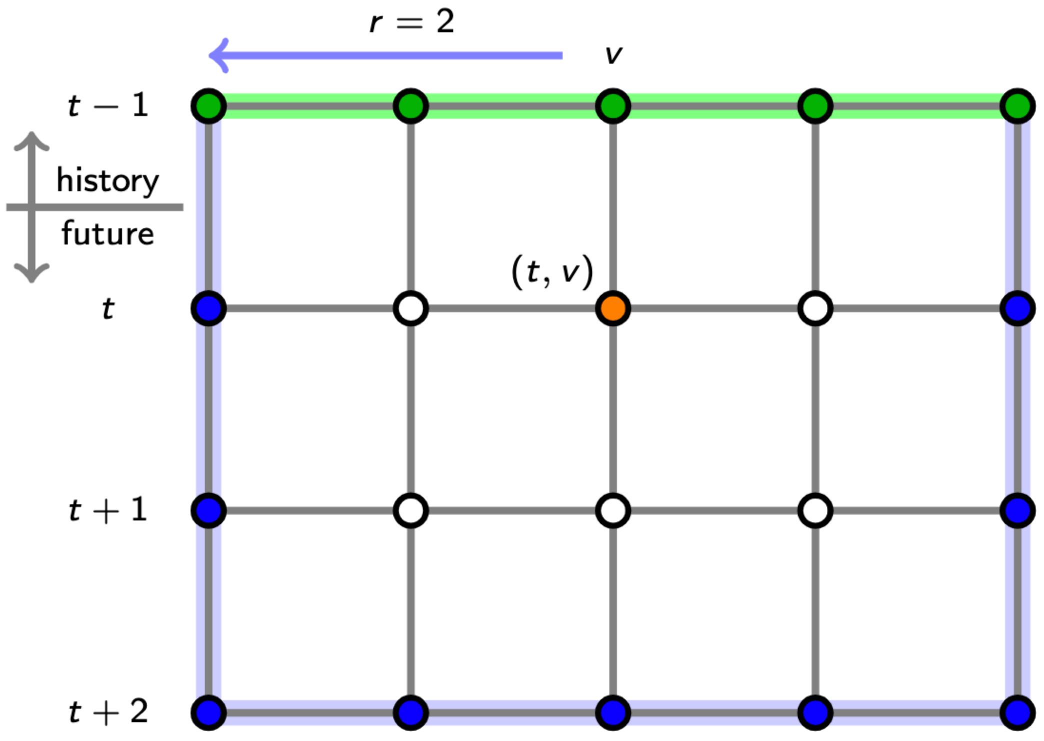
Theorem (upper bound): We proposed a decentralized online algorithm, Localized Predictive Control, that achieves the competitive ratio of $1 + O\left({\rho_T}^k\right) + O\left({\rho_S}^r\right)$.

k: Prediction horizon. r: Communication radius.

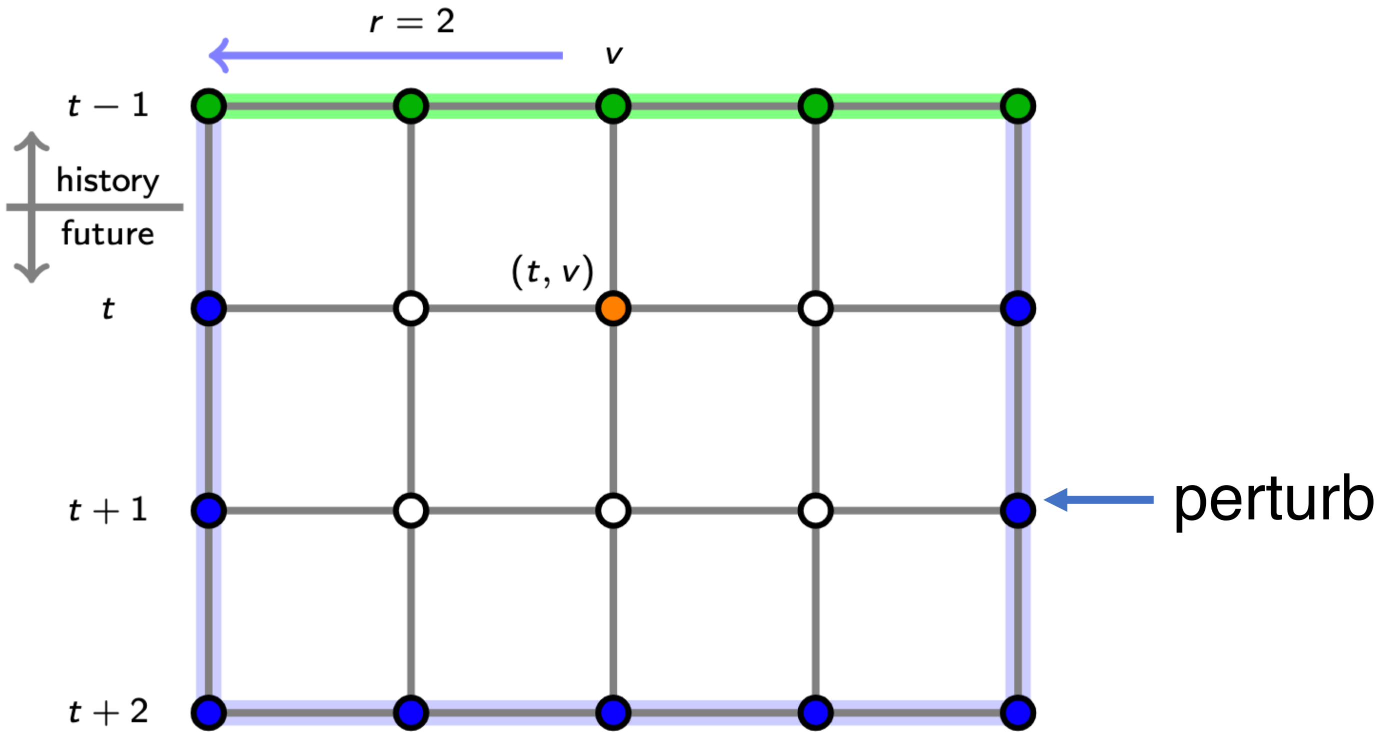
Theorem (lower bound): The competitive ratio of any decentralized online algorithm is lower bounded by $1 + \Omega\left({\lambda_T}^k\right) + \Omega\left({\lambda_S}^r\right)$.

The competitive ratio bound for LPC is near optimal!

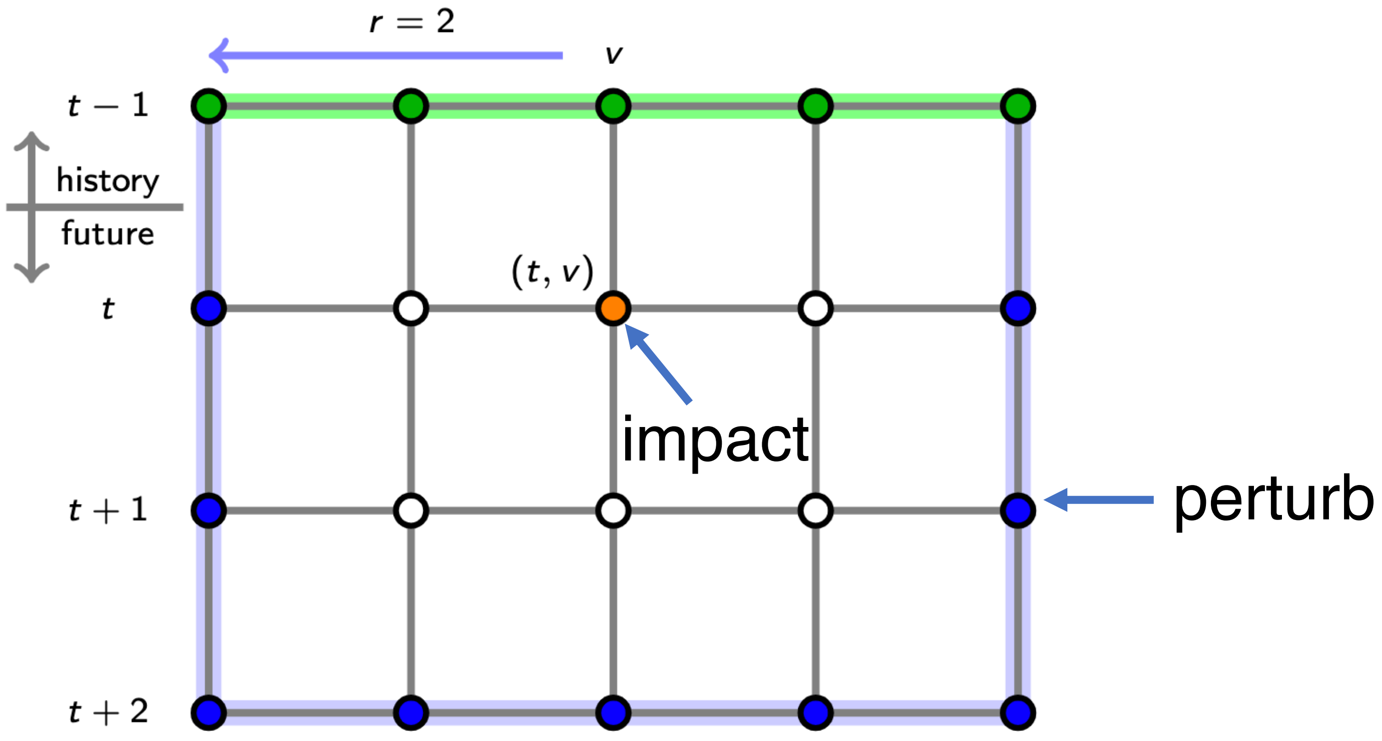
Method: Perturbation Analysis



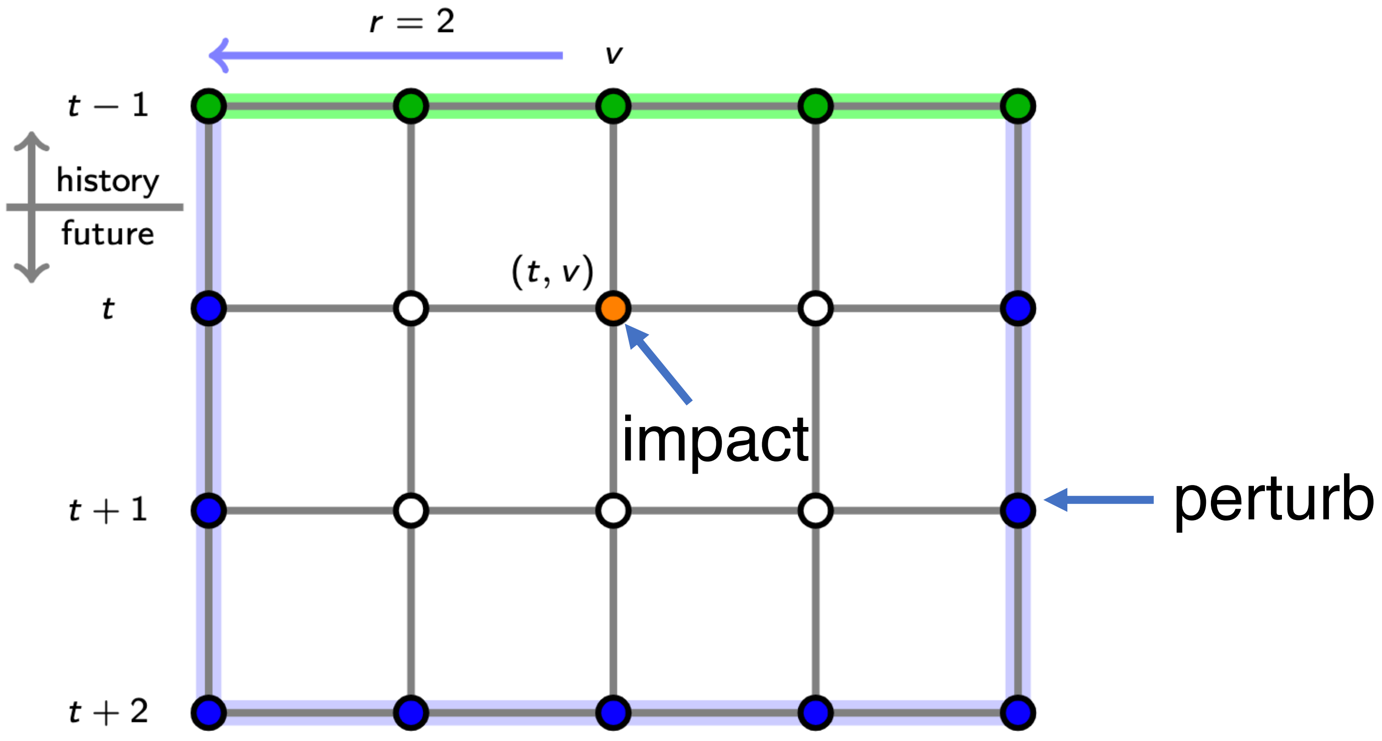
Method: Perturbation Analysis



Method: Perturbation Analysis



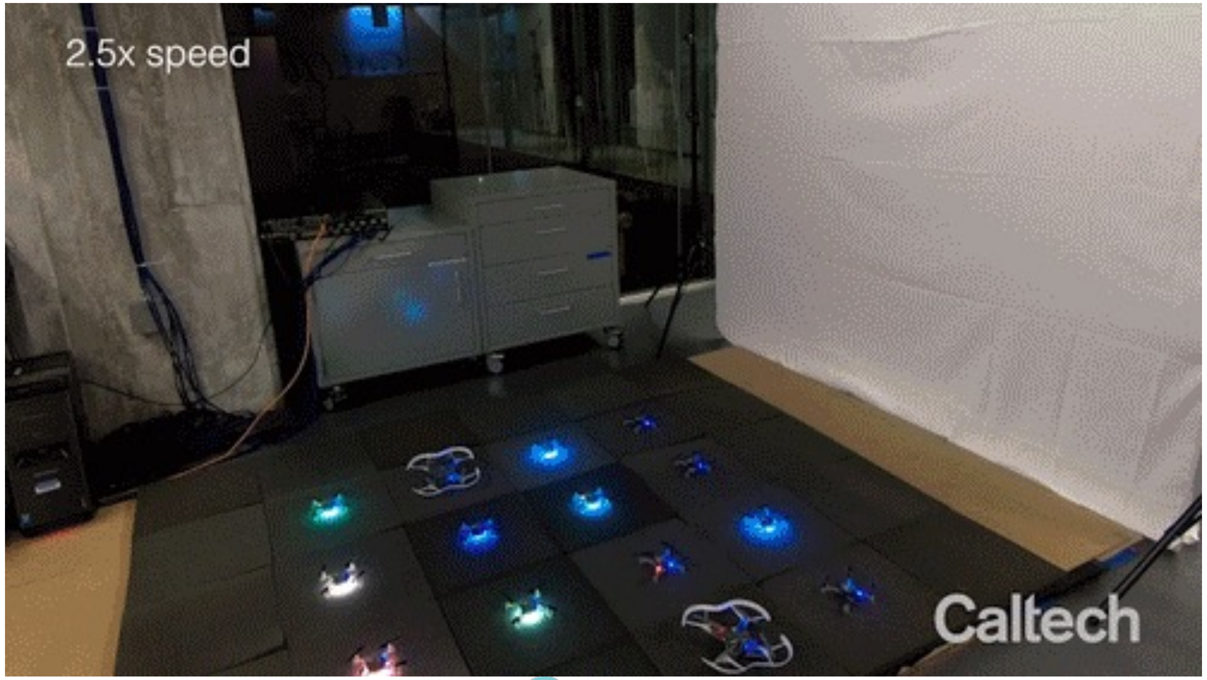
Method: Perturbation Analysis



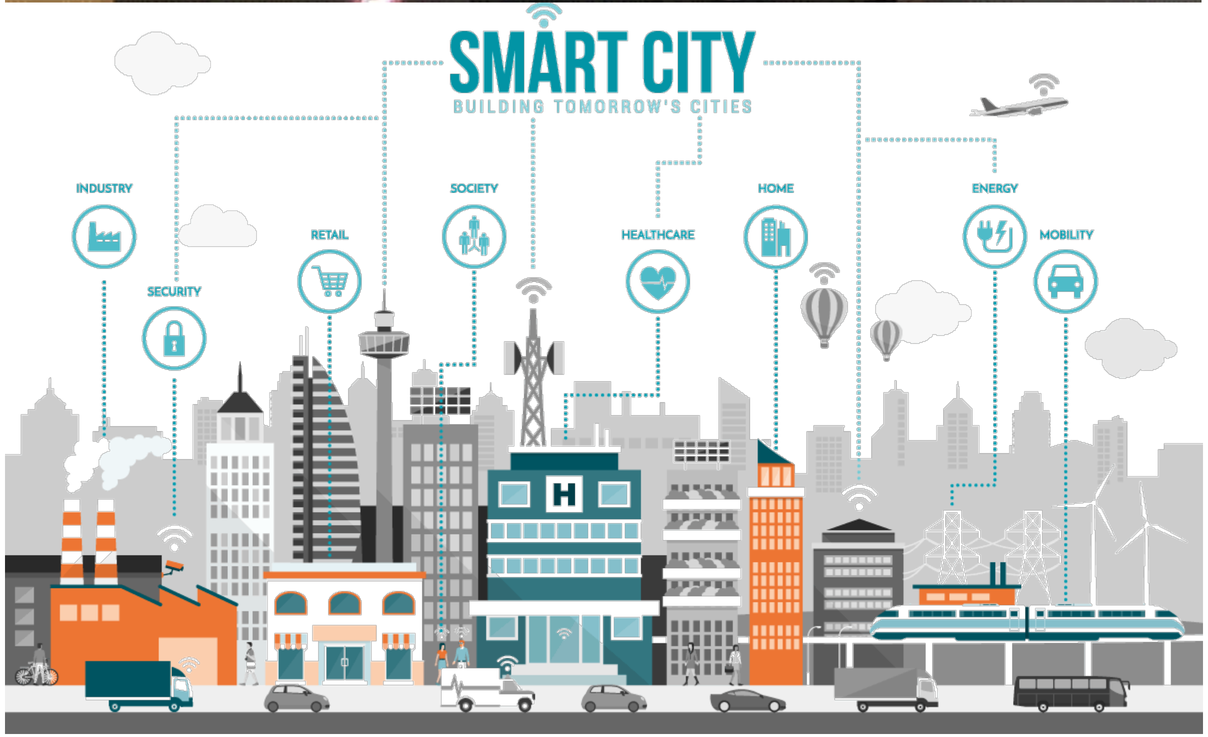
Perturbation bounds for offline optimal ➡ Regret & Competitive ratio

Other Applications

Drone swarm



Power grids



Smart city

*Picture credits: Guanya Shi, Malwarebytes Labs, ARC Advisory Group.

Decentralized Online Convex Optimization in Networked Systems

Yiheng Lin, Judy Gan, Guannan Qu, Yash Kanoria, and Adam Wierman

Email: yihengl@caltech.edu