Introduction to Learned Systems

Jan 15 2026

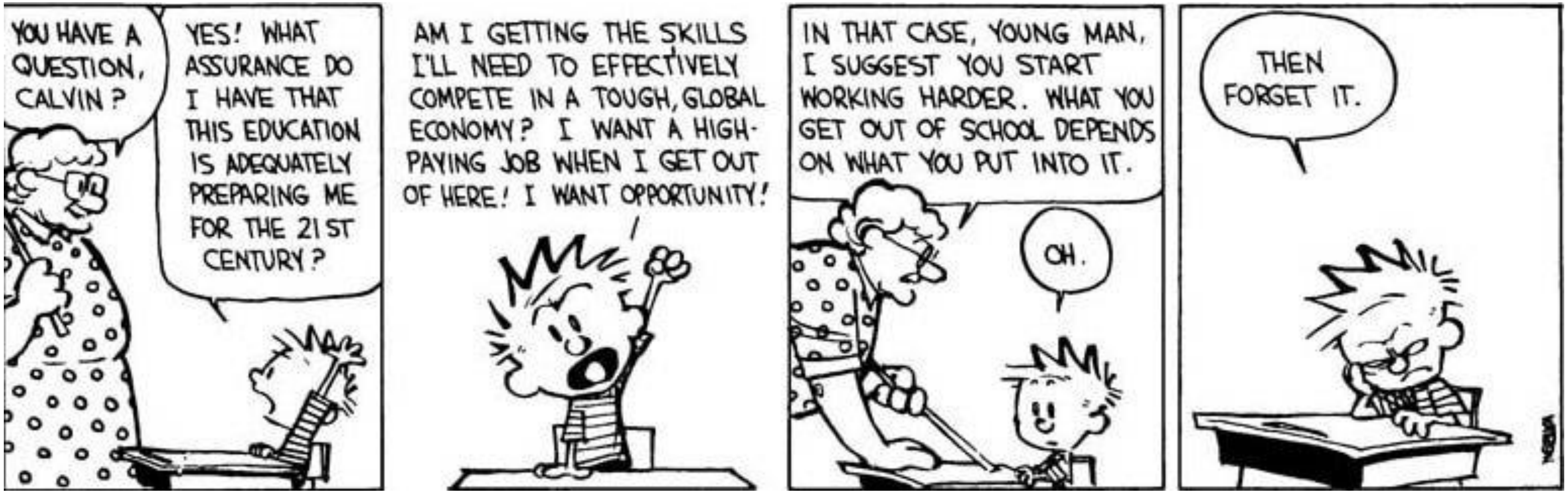
Administrivia

- Review submission site (HotCRP) is up!
  - Reach out to me if you are not able to access
- Presentation signup sheet is up
  - Looking for presenters next week
- Leave of absence
  - Give me an heads up
  - Missing more than 3 classes will require permission

AI/LLM tools

- Use tools wisely
  - Use to understand/clarify concepts
  - Use to find related work

- Do not use to:
  - Write paper reviews
  - Generate project ideas


YOU HAVE A QUESTION, CALVIN?
YES! WHAT ASSURANCE DO I HAVE THAT THIS EDUCATION IS ADEQUATELY PREPARING ME FOR THE 21ST CENTURY?
AM I GETTING THE SKILLS I'LL NEED TO EFFECTIVELY COMPETE IN A TOUGH, GLOBAL ECONOMY? I WANT A HIGH-PAYING JOB WHEN I GET OUT OF HERE! I WANT OPPORTUNITY!
IN THAT CASE, YOUNG MAN, I SUGGEST YOU START WORKING HARDER. WHAT YOU GET OUT OF SCHOOL DEPENDS ON WHAT YOU PUT INTO IT.
OH.
THEN FORGET IT.

Goals

- Understand different types of system problems
- ML refresher
- Learned Operating System
- Park: platform for learned systems

“Systems”

1. Operating systems
2. Cluster management
3. Data systems

What is an operating system?

Applications

Operating System

Hardware

Operating system tasks

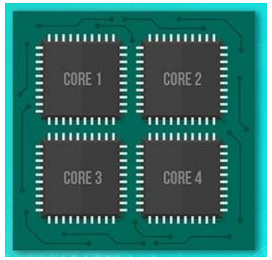
- Resource sharing
  - Multiple users
  - Multiple applications
  - Multiple devices

- Abstraction
  - Ease of use
  - Reuse common facilities
  - Support different hardware

Goals:

1. Performance
2. Fairness
3. Protection
4. Efficiency
5. Reliability

OS policies



1. CPU scheduling
2. Load balancing
3. Voltage/frequency scaling

1. Page allocation
2. Swapping
3. NUMA migrations

1. I/O scheduling
2. Admission control
3. Page cache

1. Packet/flow scheduling
2. Congestion control

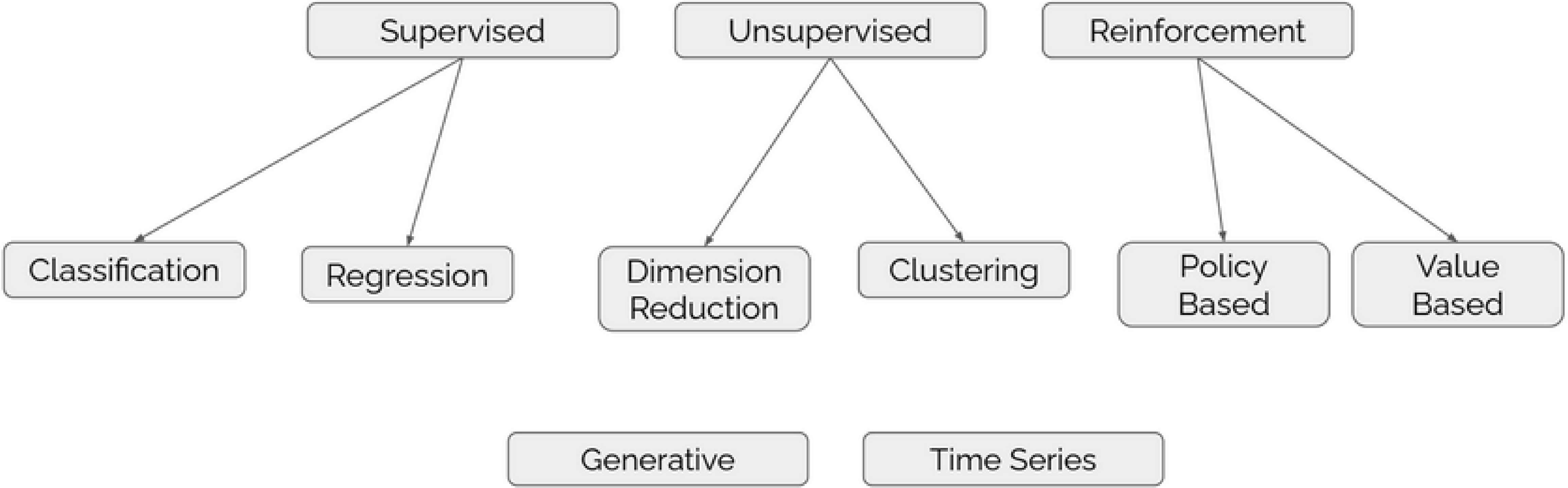
Datacenter

- Resource allocation
  - Placement/bin-packing
  - Scheduling
  - Overprovisioning
- Performance isolation
- Microservice scaling

Data systems

- Buffer management
- Configuration selection
- Query planning
- Distributed systems
  - Consistency, availability, partitioning

Machine Learning



ML algorithms

- Decision trees and random forests
- Support Vector Machines (SVM)
- Baye’s
- K-NN
- Neural nets
- Linear and logistic regression
- K-means, DBSCAN, BIRCH
- Time series models: ARIMA, transformers

ML development process

- Feature selection
- Model selection
- Training and iteration
- Hyperparameter tuning
- Analysis

Where ML can help systems

- Richer decision space
- Recommend/offer richer inputs
- Configuration selection/tuning
- Design algorithms
  - Simulations
  - Trace synthesis

Existing ML for systems use-cases

- Learned decisions
- Anomaly detection
- Forecasting
- Discovery
- Optimization

Challenges in using ML

- Overheads
  - Inference costs
  - Data collection costs
- Implementation challenges
  - Policy update or replacement
- Explainability/debuggability
- Robustness

Learned Operating System

Motivation

- Hardware and applications rapidly changing

- Existing OS are rigid
  - Limited configurations
  - Difficult to change

- Config selection difficult – 17K parameters in Linux

- No adaptivity

Proposal

[Parameter Tuning in Memory Tiering Systems]

Learning configurations	e.g.: I/O readahead, scheduler timeslice, frequency of swapping
Learning policies	e.g.: block placement, page replacement, I/O hedging
Learning mechanisms	e.g.: replace virtual-physical mapping entirely with an ML model

Challenges

- Too many knobs

Set-ting	File System	Blk Size	Inode Size	Block Grp	Jour-nal	Flex Grp	Read-ahead	XFS Sctr Size	Allc Grp Cnt	Log Buf Cnt	Log Buf Size	Allc Size	Node Size	Spec Opt	Atime Opt	I/O Schd	Drty Bg Ratio	Drty Ratio	Dev	Total
S1	Ext2	3	7	6	–	–	–	–	–	–	–	–	–	–	2	3	dflt	dflt	4	2,208
S1	Ext3	3	7	6	3	–	–	–	–	–	–	–	–	–	2	3	dflt	dflt	4	6,624
S1	Ext4	3	7	6	3	dflt	dflt	–	–	–	–	–	–	–	2	3	dflt	dflt	4	6,624
S1	XFS	3	5	–	–	–	–	dflt	9	dflt	dflt	dflt	–	–	2	3	dflt	dflt	4	2,592
S1	Btrfs	–	5	–	–	–	–	–	–	–	–	–	3	4	2	3	dflt	dflt	4	288
S1	Nilfs2	3	–	9	2	–	–	–	–	–	–	–	–	–	2	3	dflt	dflt	4	1,944
S1	Reiserfs	dflt	–	–	3	–	–	–	–	–	–	–	–	2	2	3	dflt	dflt	4	192
S2	Ext4	3	3	dflt	3	3	3	–	–	–	–	–	–	–	dflt	3	2	3	SSD	3,888
S2	XFS	3	2	–	–	–	–	3	4	2	2	2	–	–	dflt	3	2	3	SSD	5,184

Carver: Finding Important Parameters for Storage System Tuning

Challenges

- Problem formulation and model selection
  - Feedback loop is long
  - Single large model vs multiple smaller models per instance
- Training
  - Data collection: overheads of tracing
  - No ground truth? Don’t know optimal?
- Inference
  - Overheads (time and space)

Challenges

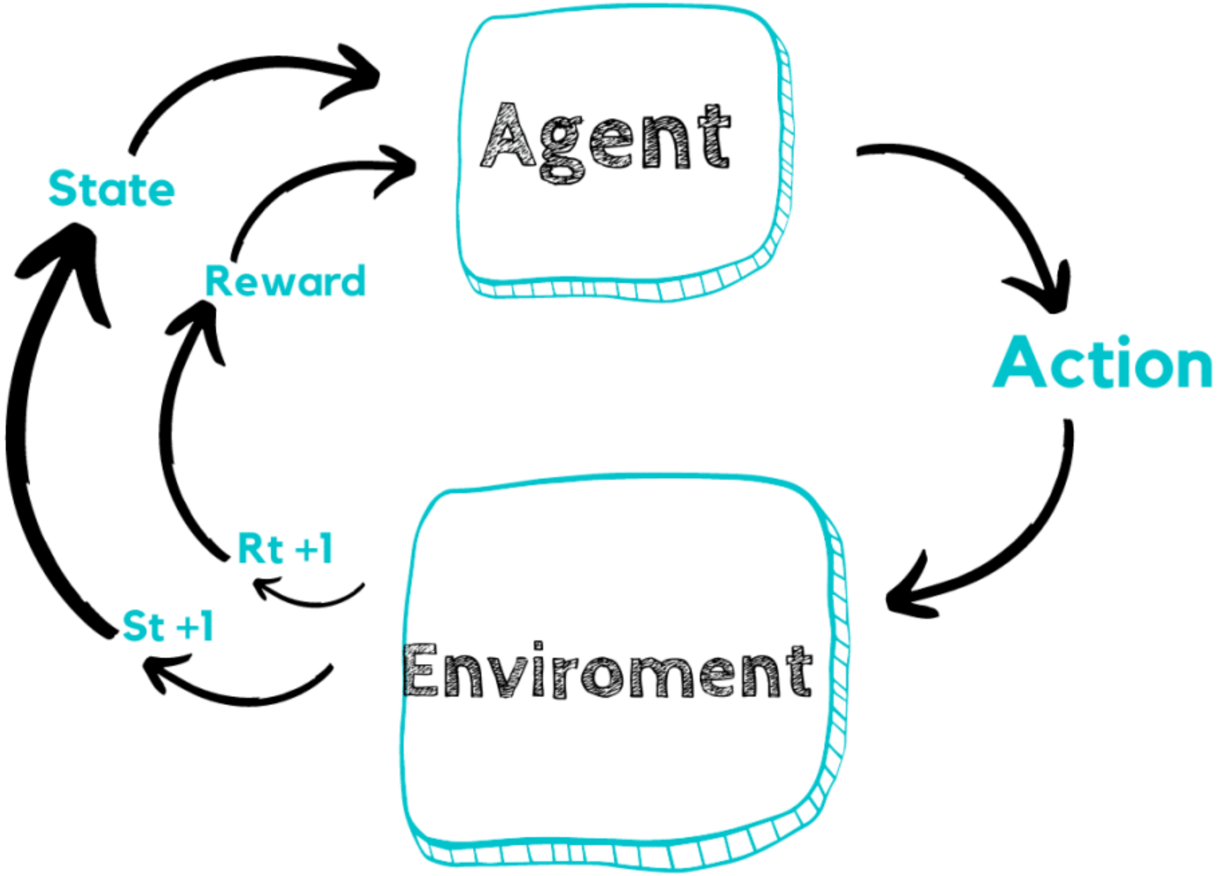
- Integration
- Security

Park: An Open Platform for Learning-Augmented Systems

Motivation

- Many tasks sequential decision-making tasks
  - Caching, scheduling, congestion control
- Modeling challenging (systems are complex)
- RL perfect fit

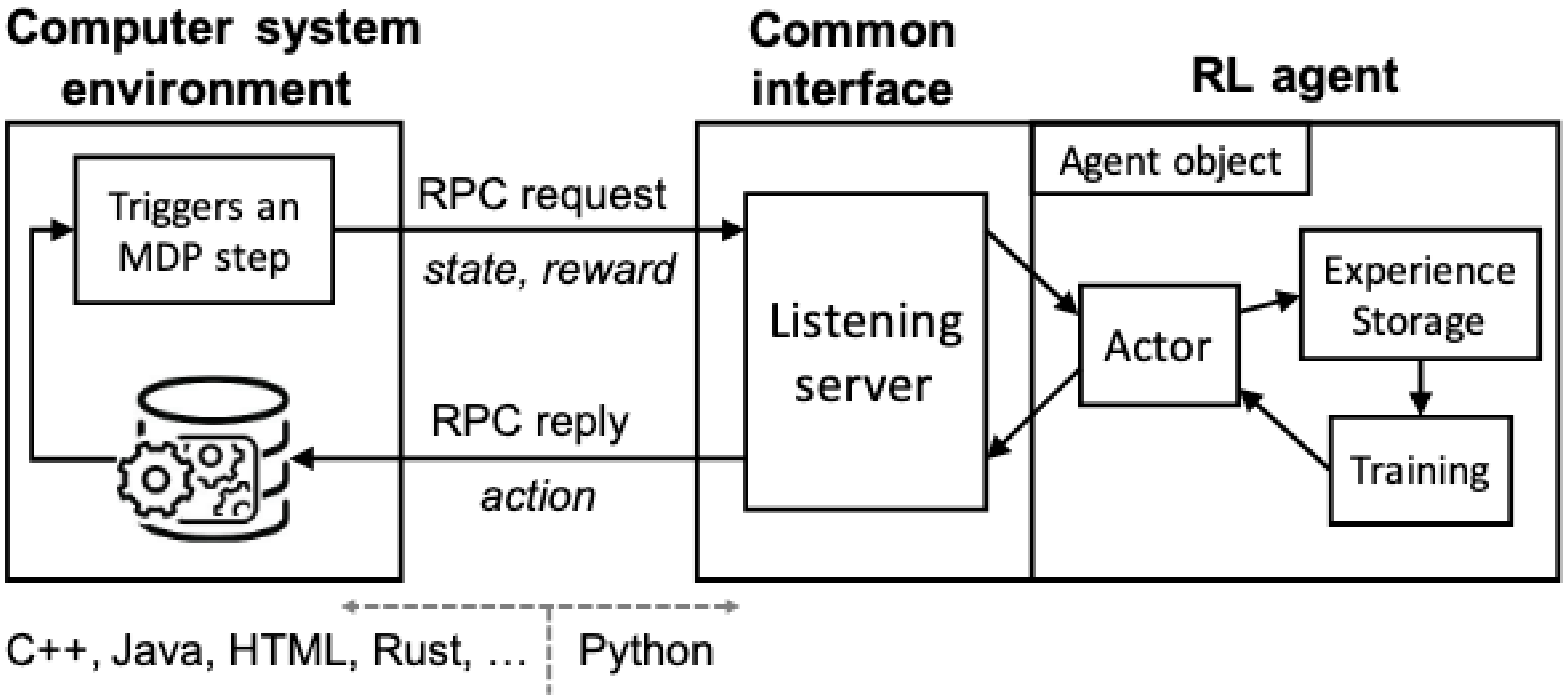
Reinforcement Learning



Challenges with RL

- Needle in a haystack
  - Majority state-action yields little difference in reward
- Large state-action space
- Input-driven variance
  - Difficult to tell if reward due to action or input
- Infinite horizon problems
  - No episodic training since no terminal state

Park platform



Environment	Type	State space	Action space	Reward	Step time	Challenges (§3)
Adaptive video streaming	Real/sim	Past network throughput measurements, playback buffer size, portion of unwatched video	Bitrate of the next video chunk	Combination of resolution and stall time	Real: ~3s Sim: ~1ms	Input-driven variance, slow interaction time
Spark cluster job scheduling	Real/sim	Cluster and job information as features attached to each node of the job DAGs	Node to schedule next	Runtime penalty of each job	Real: ~5s Sim: ~5ms	Input-driven variance, state representation, infinite horizon, reality gap
SQL database query optimization	Real	Query graph with predicate and table features on nodes, join attributes on edges	Edge to join next	Cost model or actual query time	~5s	State representation, reality gap
Network congestion control	Real	Throughput, delay and packet loss	Congestion window and pacing rate	Combination of throughput and delay	~10ms	Sparse space for exploration, safe exploration, infinite horizon
Network active queue management	Real	Past queuing delay, enqueue/dequeue rate	Drop rate	Combination of throughput and delay	~50ms	Infinite horizon, reality gap
Tensorflow device placement	Real/sim	Current device placement and runtime costs as features attached to each node of the job DAGs	Updated placement of the current node	Penalty of runtime and invalid placement	Real: ~2s Sim: ~10ms	State representation, reality gap
Circuit design	Sim	Circuit graph with component ID, type and static parameters as features on the node	Transistor sizes, capacitance and resistance of each node	Combination of bandwidth, power and gain	~2s	State representation, sparse space for exploration
CDN memory caching	Sim	Object size, time since last hit, cache occupancy	Admit/drop	Byte hits	~2ms	Input-driven variance, infinite horizon, safe exploration
Multi-dim database indexing	Real	Query workload, stored data points	Layout for data organization	Query throughput	~30s	State/action representation, infinite horizon
Account region assignment	Sim	Account language, region of request, set of linked websites	Account region assignment	Serving cost in the future	~1ms	State/action representation
Server load balancing	Sim	Current load of the servers and the size of incoming job	Server ID to assign the job	Runtime penalty of each job	~1ms	Input-driven variance, infinite horizon, safe exploration
Switch scheduling	Sim	Queue occupancy for input-output port pairs	Bijection mapping from input ports to output ports	Penalty of remaining packets in the queue	~1ms	Action representation

Takeaways

- Learning can help system design and policies
- Key challenges include
  - Problem formulation
  - Model design
  - Overheads

Discussion topics

- How could foundation models help OS?
- What directions should we take to mitigate security risks, privacy risks, and potential biases when using user/system data to train ML models?

Before next class …

- **Signup for presentations!!**
- Submit paper reviews on HotCRP by 9AM Tuesday