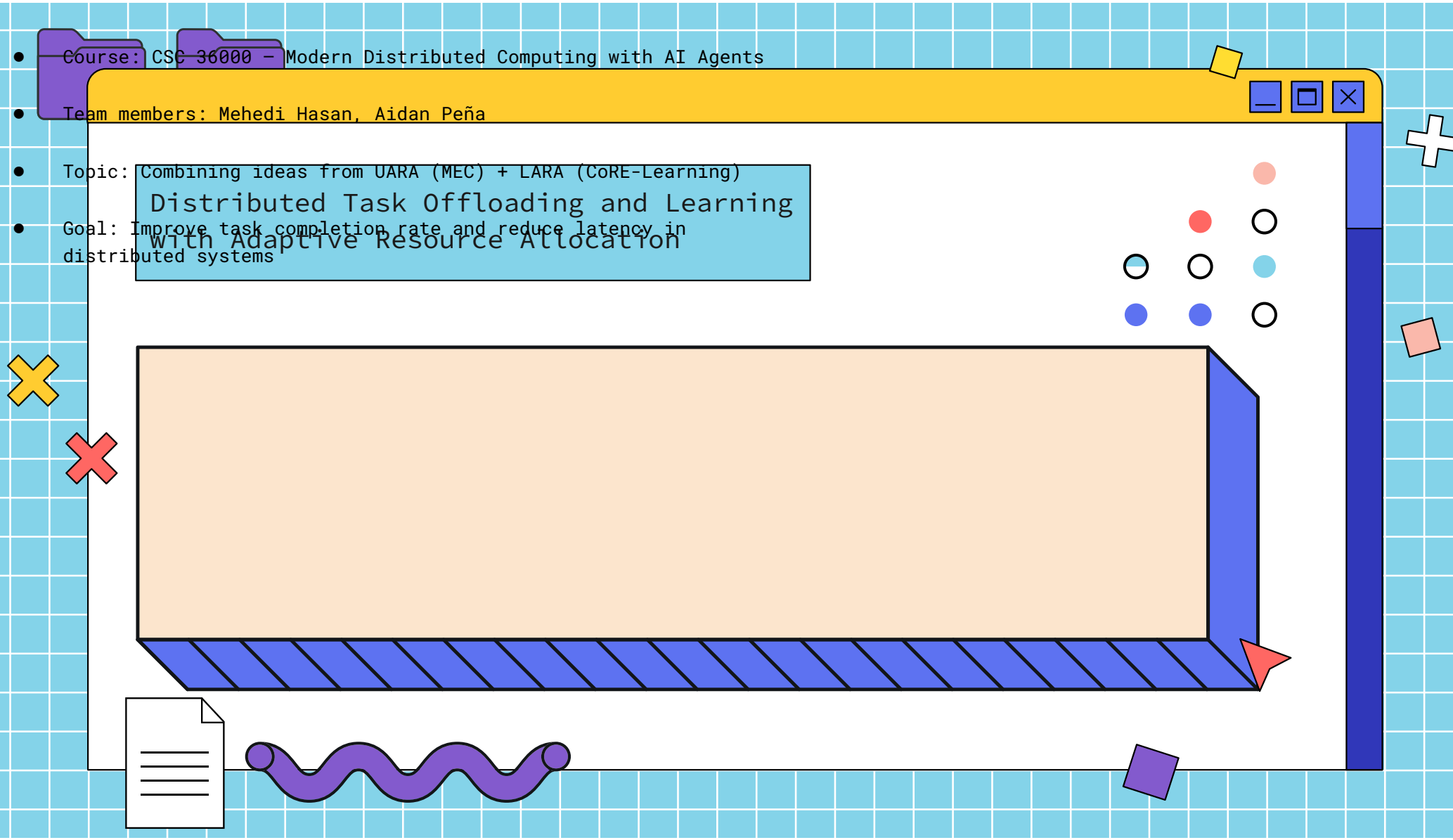
Distributed Task Offloading and Learning with Adaptive Resource Allocation

- Course: CSC 36000 – Modern Distributed Computing with AI Agents
- Team members: Mehedi Hasan, Aidan Peña
- Topic: Combining ideas from UARA (MEC) + LARA (CoRE-Learning)
- Goal: Improve task completion rate and reduce latency in distributed systems



Baselines Used in My Project

- **Uniform scheduling**
  Each task gets equal resources

- **Shortest Deadline First (SDF)**
  Task with closest deadline gets all resources

- **LARA-style scheduling**
  Estimates remaining work and allocates smartly

Why These Baselines Are Weak

Uniform Scheduler

- Wastes resources on “easy” or “impossible” tasks
- No intelligence

Shortest Deadline First

- Focuses only on time
- Ignores how hard the task is

Problem:

- They do NOT consider learning speed or resource needs per task

My Coding Progress

Created a Task class with:

- arrival time
- Deadline
- target loss
- learning curve

Implemented 3 schedulers:

- Uniform
- SDF
- LARA-style (adaptive)

Code tracks:

- progress per task
- success vs missed tasks
- final success rate

Formula I Used:

I used the LARA paper's idea:

Loss curve:

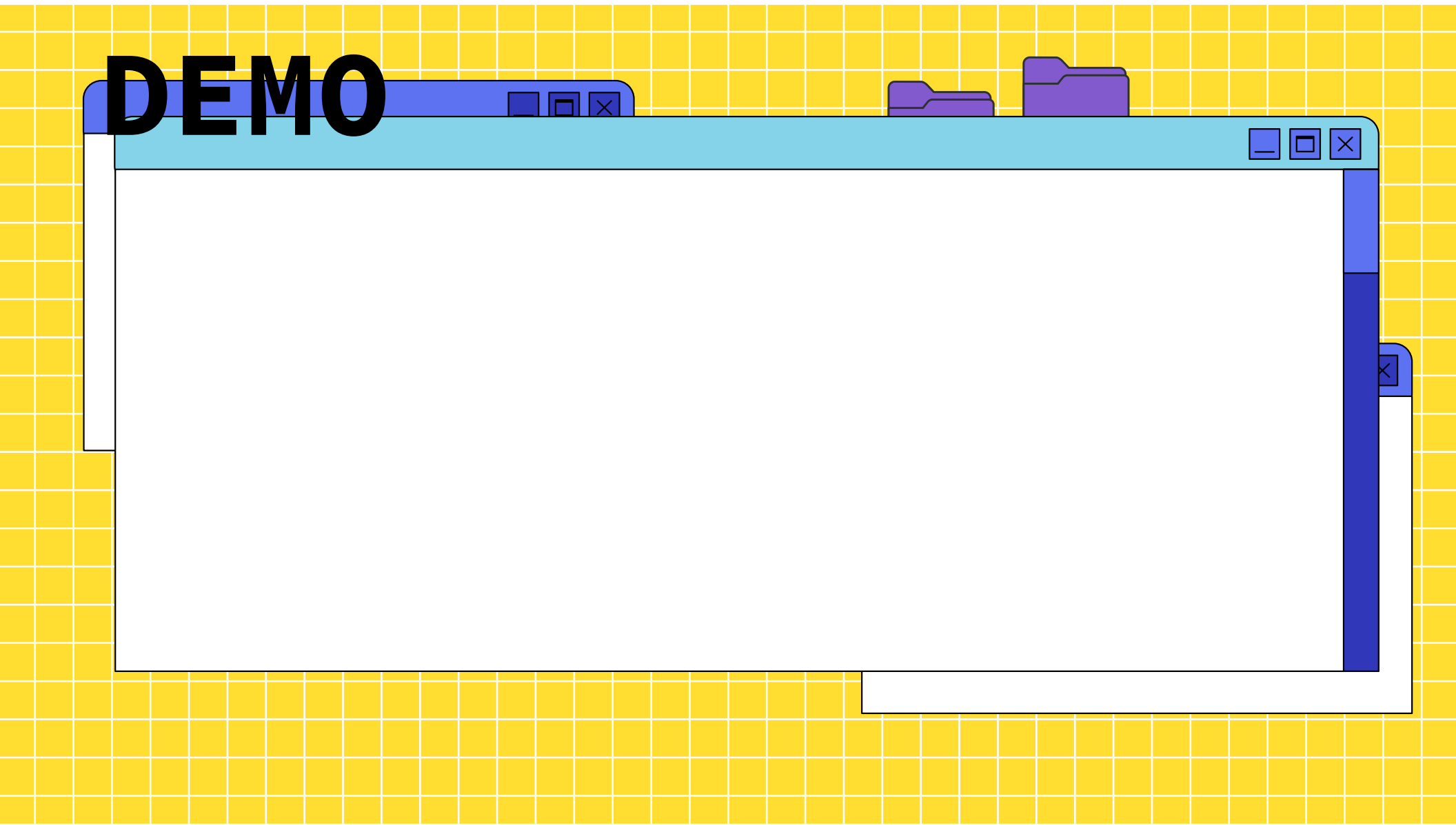
$$\text{loss} = a \cdot s^{-b}$$

To compute required samples:

$$s = \left(\frac{a}{\epsilon}\right)^{\frac{1}{b}}$$

Meaning:

- a and b describe learning difficulty
- s = number of samples needed
- ε = target loss
- Used to estimate remaining work for each task


DEMO

What Reliability Means in My Project

Inputs:

- Total tasks
- Number finished before deadline

Measures:

- How dependable the system is
- If tasks finish on time in a dis

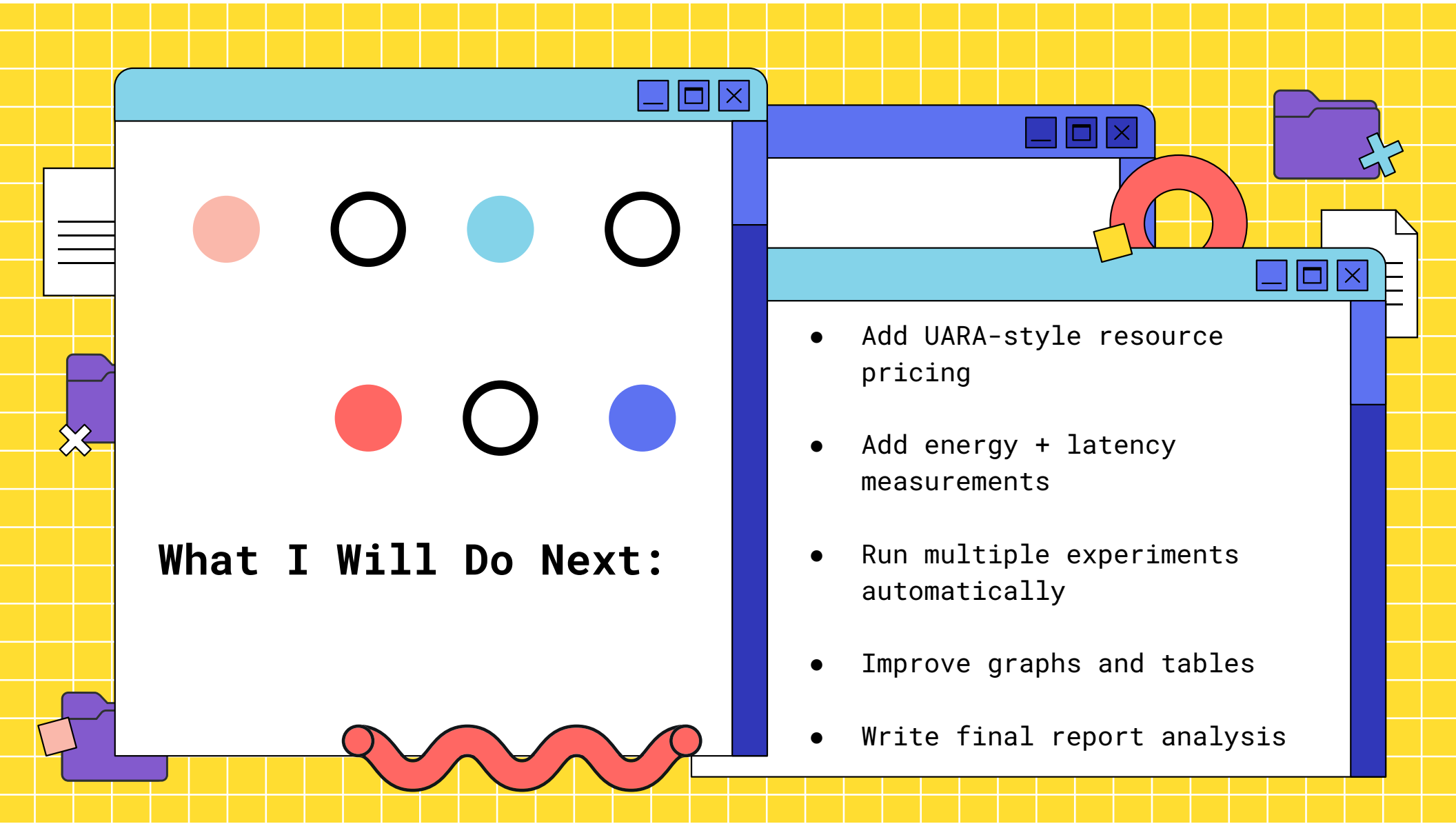
$$\text{Reliability} = \frac{\text{Completed Tasks}}{\text{Total Tasks}}$$

Challenges

- Estimating loss curve with noisy data
- Tuning exploration threshold
- Occasionally unstable estimates
- Debugging allocation logic

What I did:

- Added error handling
- Added exploration-first stage
- Used weighted least squares


What I Will Do Next:
Add UARA-style resource pricing
Add energy + latency measurements
Run multiple experiments automatically
Improve graphs and tables
Write final report analysis