Automating cryo-EM data collection with machine learning (Discussion)

NEW YORK STRUCTURAL BIOLOGY CENTER

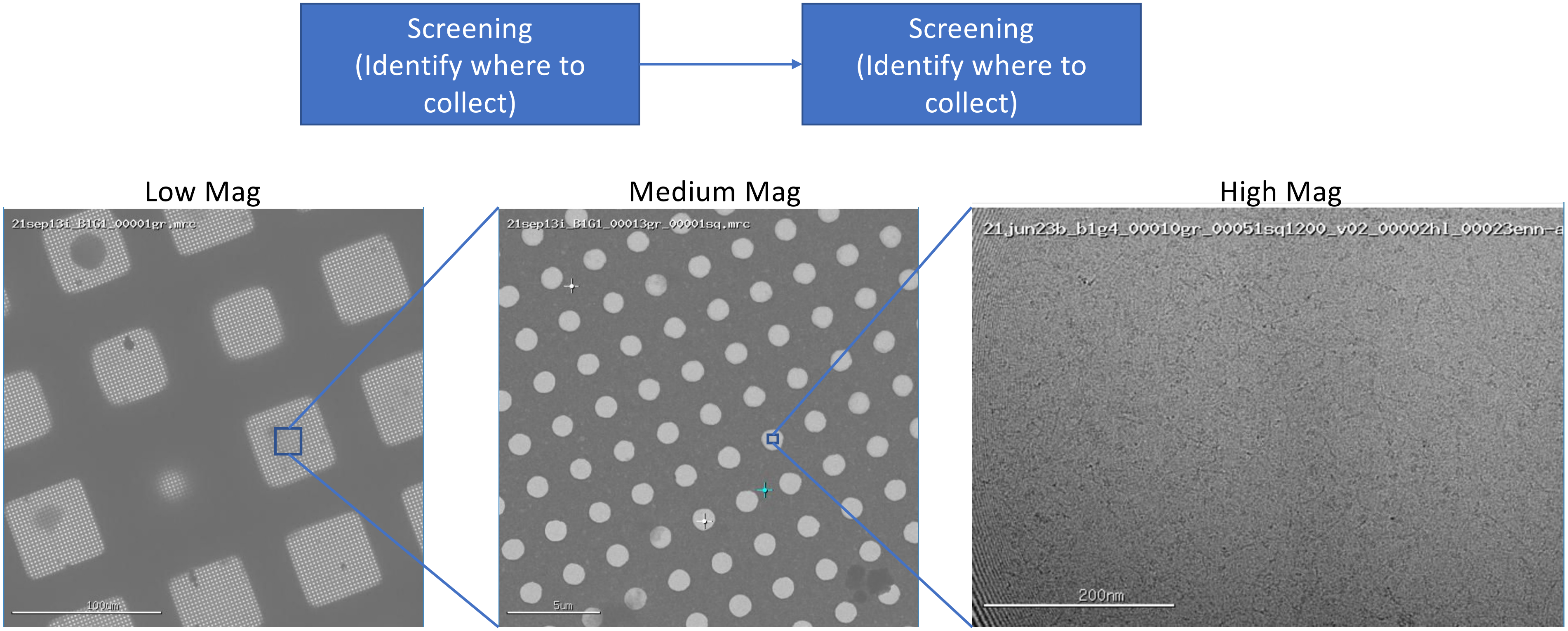
SIMONS ELECTRON MICROSCOPY CENTER

Tristan Bepler

SMART data collection workshop 2022

SIMONS MACHINE LEARNING CENTER

The cryoEM data collection process involves many steps that require human intervention



Screening (Identify where to collect)

Screening (Identify where to collect)

Low Mag

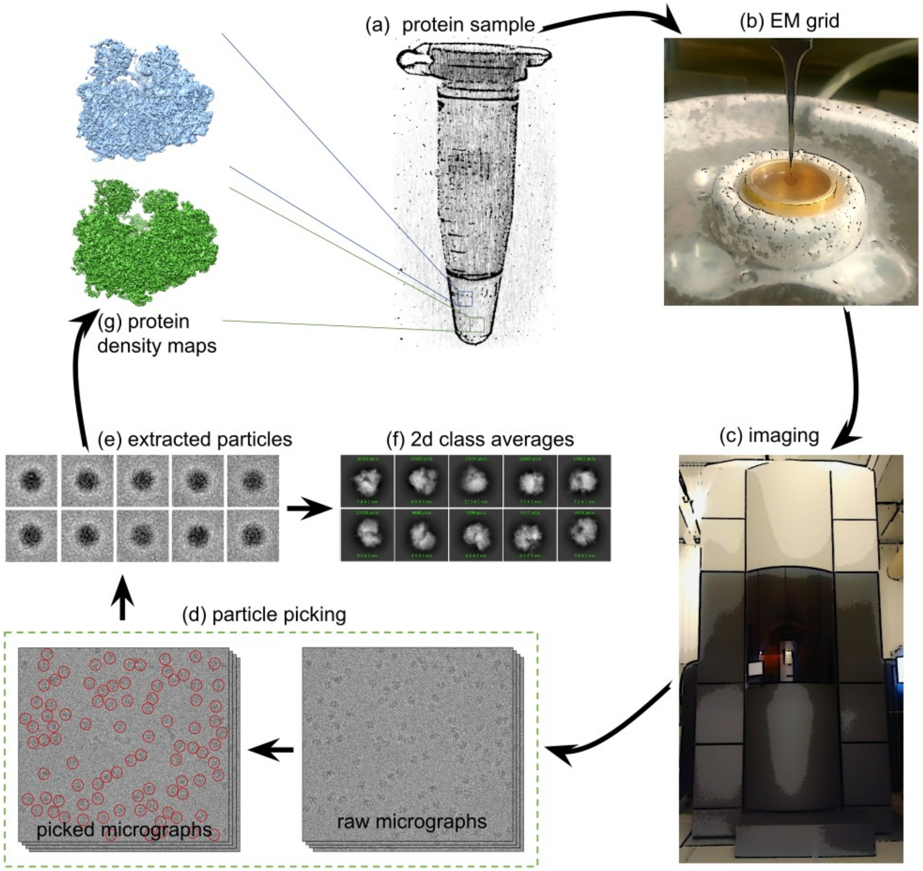
Medium Mag

High Mag

21sep13i_B1G1_00001gr.mrc
100um
21sep13i_B1G1_00013gr_00001sq.mrc
5um
21jun23b_b1g4_00010gr_00051sq1200_v02_00002hl_00023enn-a
200nm

In single particle cryoEM, the goal is to collect enough projection images to solve a high resolution structure

- More data (= more particles)
- Better quality data (= less noise)



Single particle is not the only use case for cryoEM, even in structural biology

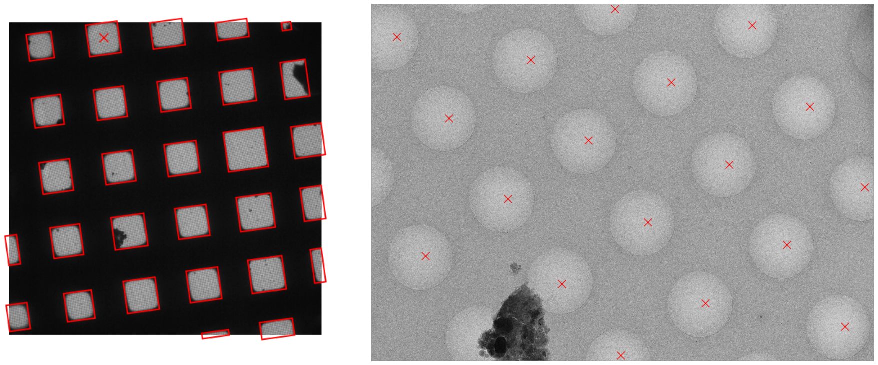
- Do automation algorithms need to be specific for single particle?
- Can they support other applications of cryoEM?

What is the goal of data collection?

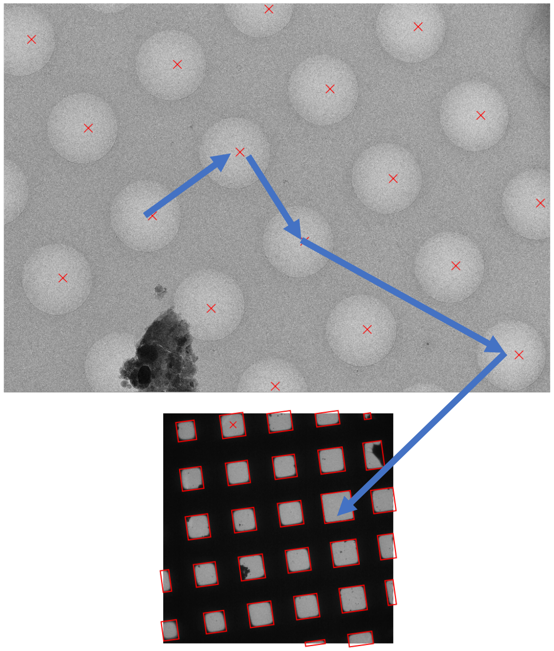
- Collect enough “high quality” data to answer the scientific question
  - High resolution reconstruction of the protein structure
  - High resolution CTF
  - Some other characterization of the system
- Operation is expensive, so throughput is important
- Not just cost – we want to increase throughput to increase the pace of science

Breaking this down into two “problems”

1) Computer Vision
Where can we target?



2) Planning
What order should we target?



Identifying possible targets

- This is an object detection problem.
- Labeled data seems to be scarce.
- We have incompletely labeled data from historical sessions.
    - Not all squares/holes are collected (“labeled”)
- How to work with this kind of data?
- Need to build strong priors into these models.

- Do we need efforts to build more complete datasets?

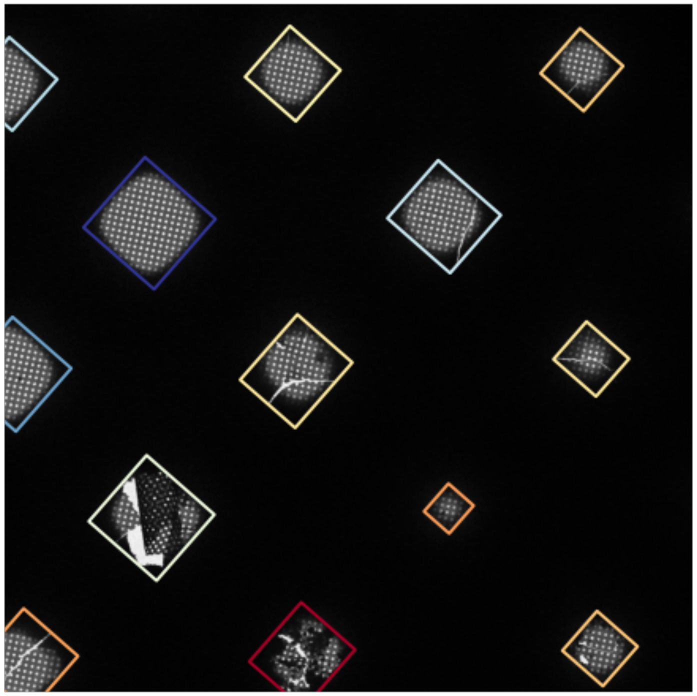
Planning – what order should we collect targets?

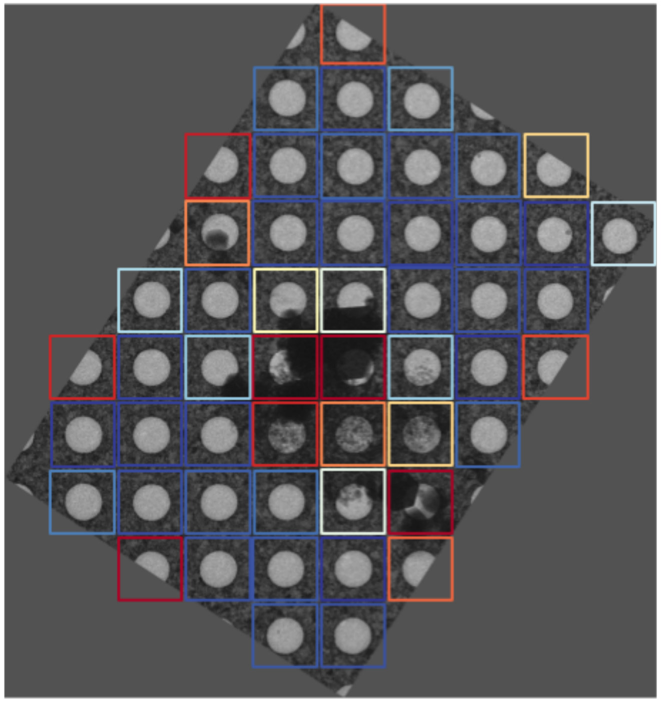
- Collect as much good data as possible in as little time as possible
- What is the optimal collection strategy to achieve this?

A simple way to plan: rank targets by quality

Can we score targets without knowing anything about the sample?

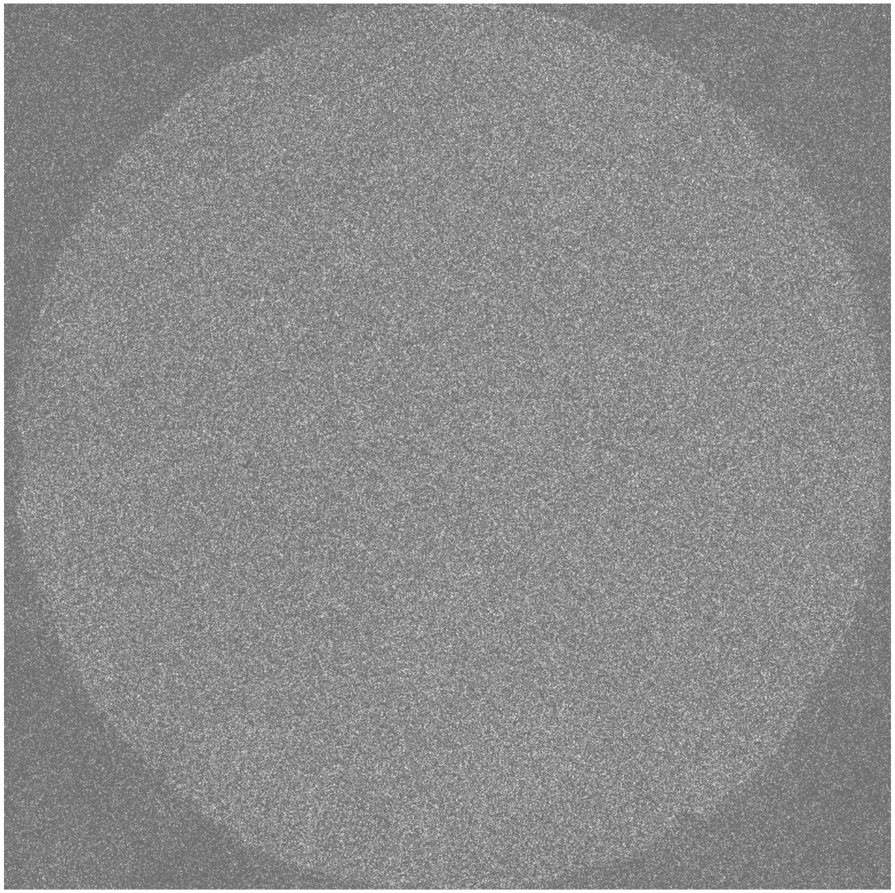
Are there squares or holes that should never be collected?





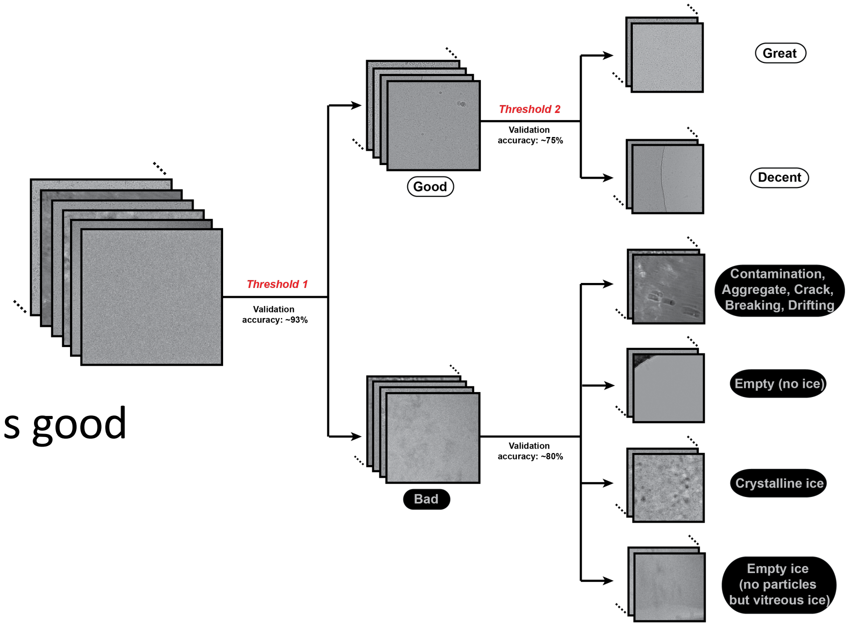
Do medium and low mag images contain enough information to know whether a high mag image will contain particles?

- Ice thickness is important
- Can we see particles at medium mag?
- What about high resolution medium mag images?
- Can we make this sample agnostic?



Interlude: what makes a target “good”?

- How do we know that a target was good after collecting it?
- Number of particles?
    - Number of good particles?
    - Do we need to do reconstruction on the fly?
        - Is resolution even a good measure of quality?
    - What about non-single particle cryoEM?
- Are there simpler ways?
    - CTF?
    - Eye test – can humans tell if a high mag image is good by looking at it?
        - Learn from this: e.g., MicAssess (Li *et al*)


Threshold 1
Validation accuracy: ~93%
Good
Threshold 2
Validation accuracy: ~75%
Great
Decent
Bad
Validation accuracy: ~80%
Contamination, Aggregate, Crack, Breaking, Drifting
Empty (no ice)
Crystalline ice
Empty ice (no particles but vitreous ice)

What if there is no sample agnostic way to tell which targets are best?

Changing the policy on the fly – learn features of good squares/holes as we go

- Different samples have different characteristics about which squares and holes contain the best data
- Without knowing all of these conditions *a priori*, what can we do?

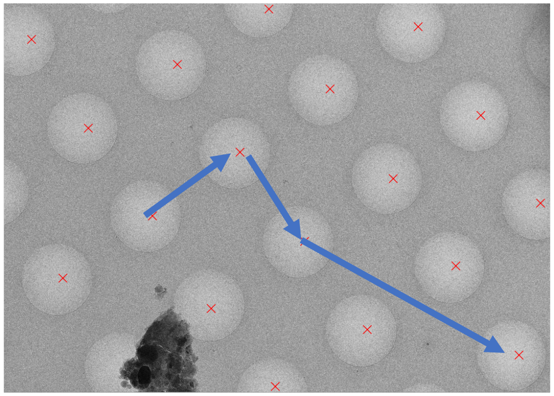
- Learn them on-the-fly!
- Use data collected during the session to update our belief about which squares and holes are best and use that to optimize collection
- Multi-armed bandit problem

What is the multi-armed bandit problem?

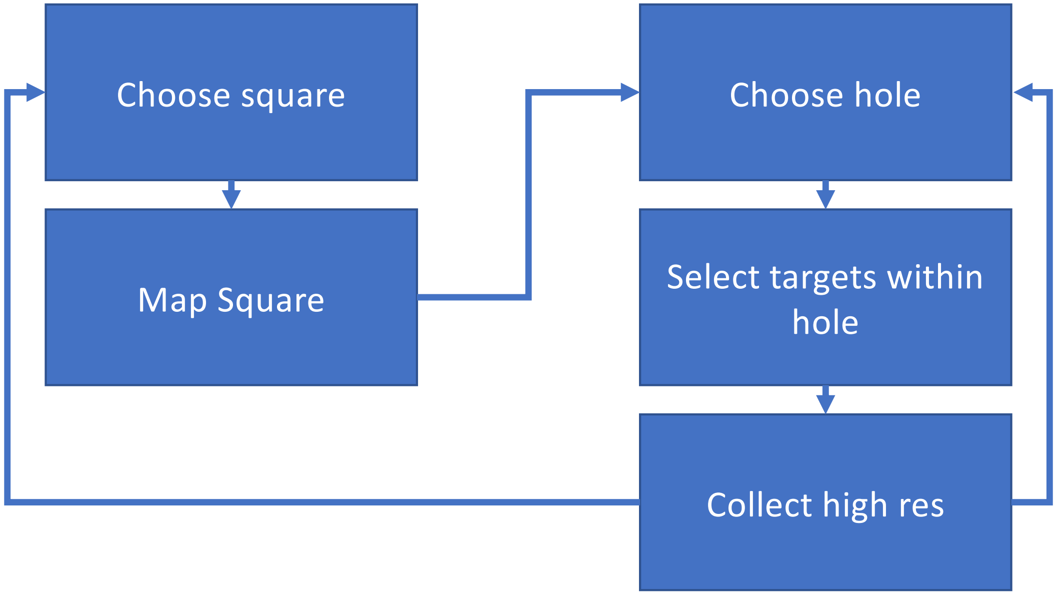
- We are presented with a bank of slot machines. Each machine pays out different amounts with different odds, but we don’t know what they are.
- Our goal is to earn as much money as possible in as few pulls as possible (cumulative returns – aka minimize the gap between our earnings and the maximum possible earnings, cumulative regret)
- How do we optimize this?
- We need to learn which machine has the best odds and then pull that lever – trade off exploration (learning about the machines) vs. exploitation (pull the lever we think has the best odds)

In cryoEM, we can think about maximizing the number of good micrographs collected

1) Pick the hole we predict to be best
2) Collect that hole to receive “reward”
3) Update our predictions about which holes are good
4) Back to 1)



But this is actually a hierarchical process



Choose square

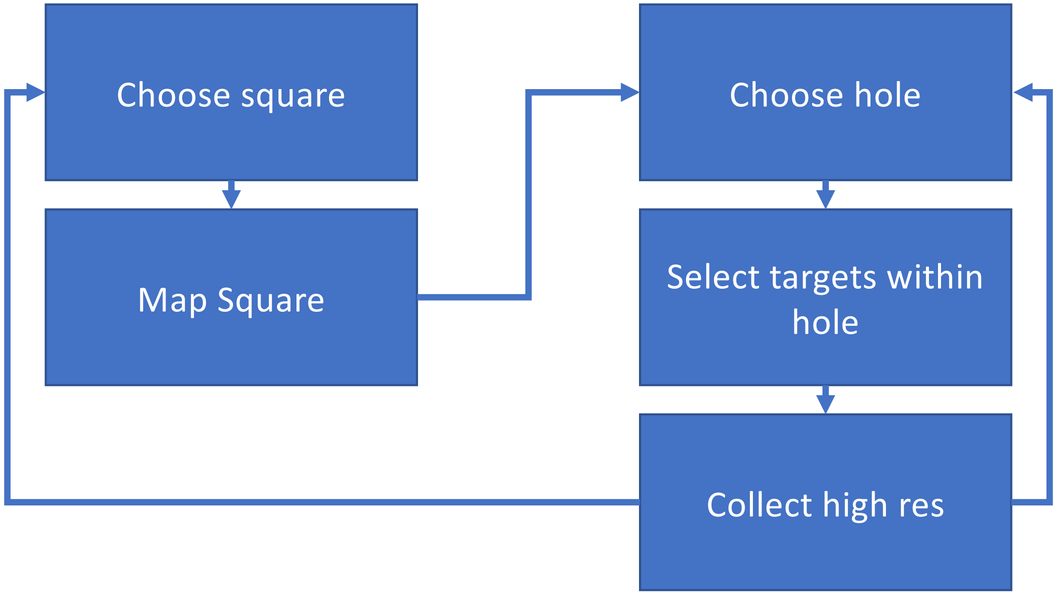
Map Square

Choose hole

Select targets within hole

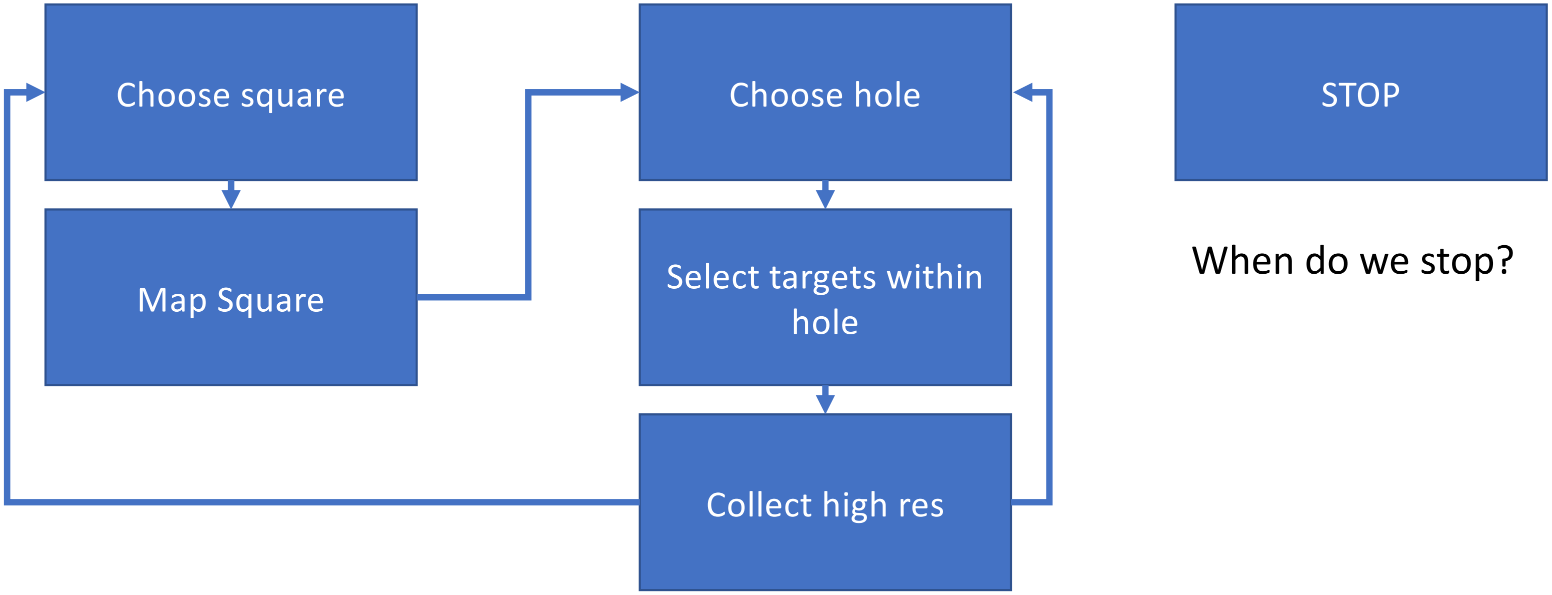
Collect high res

But this is actually a hierarchical process



Choices need to include STOP, and GO UP A LEVEL

But this is actually a hierarchical process

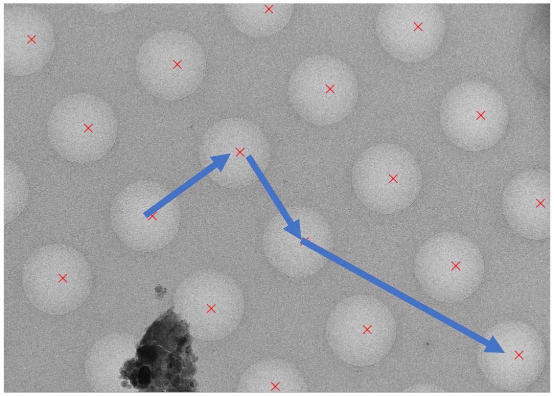


When do we stop?

Choices need to include STOP, and GO UP A LEVEL

What about time?

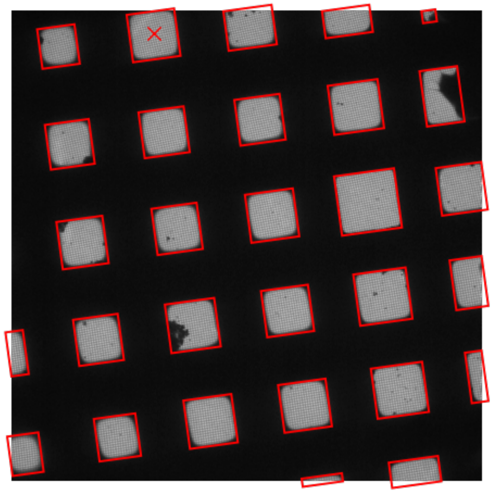
- We don’t just want to minimize the number of images collected.
  - Just a surrogate for time spent collecting.
- We also need to consider *how long each collection takes*!
  - Moving the stage is slow
  - Changing the grid is very slow
  - Etc.

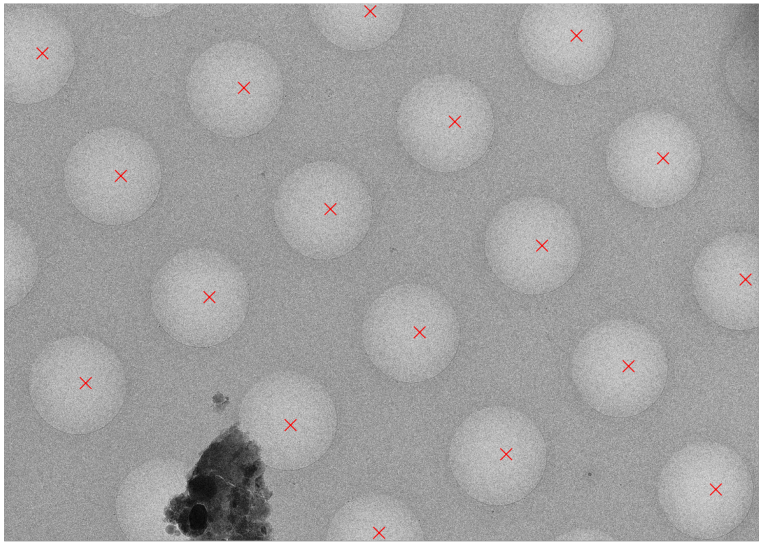


How do we minimize moving the stage?

- Requires high level planning and understanding the time cost of each collection
- Each image doesn’t have a fixed time cost, because the stage can be moved once for multiple shots
- Therefore, need to plan with multiple collections in mind
  - This is much harder than shot-by-shot planning!
  - Need to consider possible outcomes of all holes within range

Side note – can we choose better locations *within* squares and holes?





Other important considerations

- How do we interface ML with the microscope and existing control software?
- What information do users/operators want to see from ML systems?
  - Is explainability important? What does it mean in this context?
  - Can we make prediction scores interpretable?
  - Does feature importance matter?
- How should we measure the performance of these systems?

How do we train and evaluate ML models for microscope automation?

- Historical sessions
  - Biased data collection means low quality data is often missing
  - Needs full history of images and collection locations
  - Lack of downstream info

- Specific data collection
  - Requires collecting data specifically for model training
  - Can collect without bias
  - Hard to collect large number of datasets across microscopes and samples

Do we need A/B testing?

- Evaluating methods is difficult, because we need to do prospective studies
- How can we efficiently determine if an algorithm improves collection?
- Reproducing operator decisions is limited

Discussion