A. Appendix

In this supplemental material, we provide ablation study results (Table 5), Resnet18 results (Table 6) and visualization of trajectory path of microscope movement planned by CryoRL (Fig. 7) as well as polices learned from CryoRL (Fig. 8). We also include the pseudo code for the action eliminaion algorithm (Alg. 1) and more details of the experimental setup for Genetic Algorithm (Cryo-GA) and Simulated Annealing (Cryo-SA), the two baseline methods in the performance comparison of the main paper.

Training Duration	Test Duration			
	τ=120	τ=240	τ=360	τ=480
τ=120	40.4	82.1	123.1	163.4
τ=240	41.1	87.5	**130.0**	**165.5**
τ=360	**45.7**	**90.2**	125.7	163.5

(a) Effects of time duration on cryoRL performance.

Rewards		Duration (minutes)			
square-level	grid-level	τ=120	τ=240	τ=360	τ=480
0.23 (default)	0.09 (default)	41.1	86.6	**132.0**	171.4
0.23 (×2)	0.09	**43.0**	**87.0**	131.1	**172.0**
0.23	0.09 (×2)	41.6	86.9	129.5	165.9
0.23 (×2)	0.09 (×2)	41.8	80.8	124.7	163.3

(b) Effects of different rewards on cryoRL's performance.

Table 5. Ablation study of CryoRL

Methods	classifier	τ=120	τ=240	τ=360	τ=480
CryoRL-A2C	Resnet18	37.0±6.7	71.9±9.5	104.1±9.4	144.8±9.3
CryoRL-C51		37.2±4.2	70.6±5.0	98.1±5.0	128.0±3.6
CryoRL-DQN		42.9±3.6	80.8±3.0	123.3±5.9	168.5±2.0
CryoRL-DQN (dueling)		42.9±4.2	86.9±5.2	125.2±5.3	159.5±6.9
CryoRL-DQN (prioritized)		42.3±4.1	86.0±4.3	128.3±3.6	174.1±5.5
CryoRL-A2C†	Resnet18	46.0±2.6(+24.3%)	86.4±1.2(+20.2%)	124.4±2.2(+19.5%)	158.8±4.5(+9.7%)
CryoRL-C51†		46.5±0.8(+25.0%)	78.1±1.3(+10.6%)	116.7±1.0(+18.9%)	138.2±2.8(+8.0%)
CryoRL-DQN†		**47.4**±2.0(+10.5%)	**91.0**±2.5(+12.6%)	132.8±2.1(+7.7%)	176.5±3.5(+4.7%)
CryoRL-DQN† (dueling)		47.2±1.0(+10.0%)	89.1±2.7(+10.0%)	129.2±1.8(+3.2%)	166.2±5.0(+4.2%)
CryoRL-DQN† (prioritized)		47.1±2.3(+11.3%)	90.4±2.3(+2.5%)	**133.0**±3.0(+3.7%)	**177.4**±4.1(+1.9%)
CryoRL-DQN	Resnet50	41.7±3.1	86.6±3.0	132.0±2.3	171.4±2.0
CryoRL-DQN†		47.4±0.5(+13.7%)	89.0±3.1(+5.1%)	131.8±1.8(+0.0%)	172.6±2.0(+1.0%)

Table 6. Performance of different CryoRL variants on the Y1 dataset using Resnet18 as the offline hole classifier († indicates action elimination.). The performance gains from action elimination are highlighted by numbers in parentheses. The numbers in bold mark the best performance achieved by CryoRL under different time durations using Resnet18 as the classifier.

Resnet18 Results. We adopted Resnet18 as the offline classifier for CryoRL, which achieves better low-CTF classification accuracy than Resnet50 (91.0% v.s 83.9%), but lower high-CTF classification accuracy (87.5% v.s 91.2%). This suggests that Resnet18 yield more falsely classified good holes. As a result, CryoRL based on Resnet18 underperforms its counterpart based on Resnet50 (Table 6). However, when action elimination is applied, the performance of Resnet18 is significantly boosted and even gets slightly better than that of Resnet50. Additionally, action elimination greatly improves A2C and C51, similar to what's shown in the main paper.

Trajectory Path and RL Policy Visualization. We plot one trajectory path of the microscope movement on the atlas planned by our CryoRL at square level (left) and patch level (right), respectively, in a 8-hour data collection session. The trajectory within a specific patch (right) illustrates that cryoRL can identify patches with more good holes (CTF≤6.0) in a global sense and prioritize their visits first. It is also noticed that some patches with a few good holes are left untouched in the square. This is because moving to a patch in another square (not shown here) is more rewarding than staying.

We further compare and visualize the policies learned by our approach as well as the strategies used by human users. Specifically, we count how often the microscope visits a pair of hole-level images (i.e patches) in the 50 trials of our results and illustrate such information by an undirected graph. A node of the graph represents a patch and a blue edge between two patches indicates the frequency of them being visited by the microscope. Note that the node size here denotes the quality of a patch determined by the number of good holes in the patch, and the node color indicates the grid the patch belongs to. Intuitively, a good policy should show strong connections between large-sized nodes. As observed in Fig 8a), our learned RL policy favors larger-size nodes, clearly demonstrating that CryoRL enables efficient data collection. Oppositely, the behavior

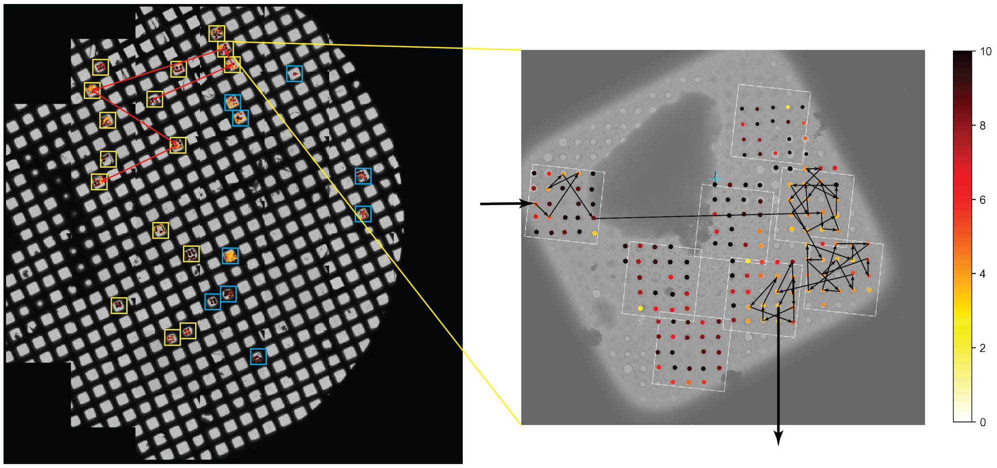
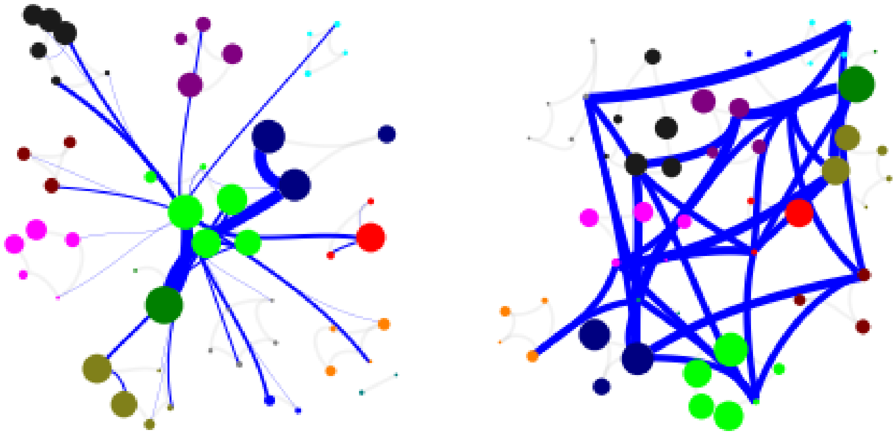


Figure 7. A trajectory of microscope movement planned by CryoRL at square level (left) and patch level (right), respectively, in a 8-hour data collection session. The blue and yellow boxes show part of the training and validation sets while the color bar represents the ground-truth CTF value. The trajectory within a specific patch (right) illustrates that cryoRL can identify patches with more good holes (CTF≤6.0) in a global sense and prioritize their visits first. It is also noticed that some patches with a few good holes are left untouched in the square. This is because moving to a patch in another square (not shown here) is more rewarding than staying.



a) RL policy

b) user policy

Figure 8. Illustration of data collection policies from cryoRL and human subjects. Here a graph node denotes a patch in our data and the size of the node indicates the quality of the patch (i.e the number of low-CTF holes). Patches from the same grid are grouped by color and linked by light grey edges. A blue edge between a pair of patches shows how often the two patches are visited by the microscope. Intuitively, an effective policy should demonstrate strong connections between large-sized nodes, which is the case for the learned policy by our approach. As opposed to the RL policy, the human users presents random behaviors (b)).

of human users is random, with a lot of more patches being explored. This is because that the users were not penalized for switching different patches in the human study, and may also be due to the large variance in the user expertise.

Algorithm for Action Elimination. The psudo code for action elimination is illustrated in Alg. 1. In the algorithm, **Action_Elim** returns a list of valid actions, which are provided to the standard **QLearning** procedure or other policy learners for policy learning. The procedure **max_lCTF** finds an upper limit of the number of low-CTF holes within a time duration τ under the assumption that all holes are in good quality. The elimination coefficient β controls the size of the valid action set. During training, β should be set large to ensure sufficient training data with diversity. However, in test, β can be set smaller to eliminate bad microscope movements while making action execution efficient.

Alg. 1 Fast CryoRL with Action Elimination

```
Require: States $S$, Actions $A$, Rewards $R$
Require: Learning Rate $\alpha$, Discounting factor $\gamma$, Elimination coefficient $beta$
Require: Switching costs $C$, Duration $\tau$
procedure QLearning_AE($S, A, R, C, \alpha, \beta, \gamma, \tau$)
    $P \leftarrow [p_0, p_1, \cdots, p_n]$                                  ▷ Patches
    $L \leftarrow [l_0, l_1, \cdots, l_n]$                                  ▷ # of predicted lCTFs in each patch
    $A' \leftarrow Action\_Elim(P, L, C, \tau)$
    $Q \leftarrow QLearning(S, A', R, \alpha, \gamma)$                      ▷ standard Q_learning
     return $Q$
end procedure
```

```
procedure Action_Elim($P, L, C, \beta, \tau$)
    $N_{max} \leftarrow \beta * max\_lCTF$(P,C,$\tau$)          ▷ maximum lCTFs found assuming that all holes are good
    $n \leftarrow 0$
    $A' \leftarrow \{\}$
    for $p_i$ in $P$ do
        $n \leftarrow n + l_i$
        $A' \leftarrow A' \bigcup \{h_j \in p_i | j = 1 \cdots m_i\}$
        if $n \geq N_{max}$ then
            $break$
        end if
    end for
     return $A'$
end procedure
```

Experimental Setup for Genetic Algorithm (GA) and Simulated Annealing (SA) As mentioned in the main paper (Section 5.2), the solutions of both GA and SA are assessed based on the same objective function used for RL, i.e Eq. 1 in the main paper. We implemented CryoRL-GA based on pyGAD [8] and Cryo-SA base on SimAnneal [24]. For CryoRL-GA, we set the number of generations to 40 and the solutions per population to 10. We use single-point crossover and and random mutation. For CryoRL-SA, the minimum and maximum temperatures are chosen as $1e - 8$ and $\sqrt{N}$, respectively, where N is the total number of training samples. The temperature reduction rate is set to 0.995.