Supplementary Information

A multi-robot-multi-task scheduling system for autonomous chemistry laboratories

Junyi Zhou[1,2,†], Man Luo[1,†], Linjiang Chen[1,†,*], Qing Zhu[1,3], Shan Jiang[1], Fei Zhang[2,*], Weiwei Shang[2,*], & Jun Jiang[1,4,*]

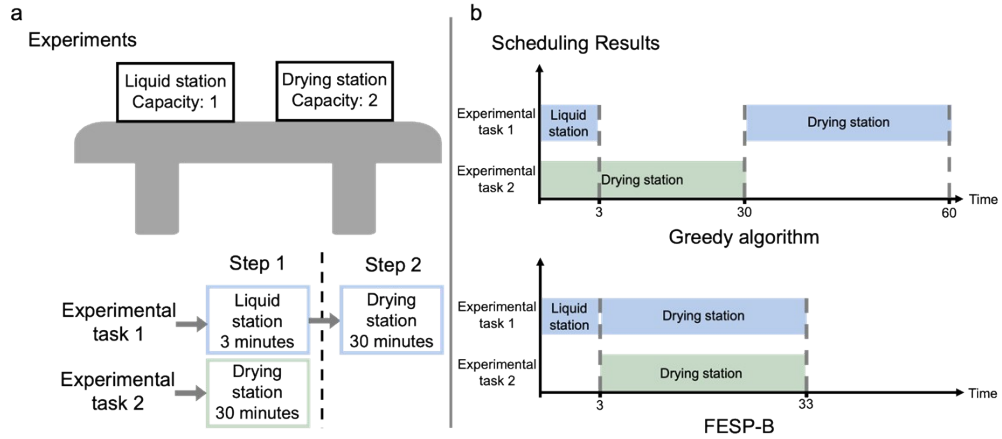
[1] Key Laboratory of Precision and Intelligent Chemistry, Hefei National Research Center for Physical Sciences at the Microscale, School of Chemistry and Materials Science, University of Science and Technology of China, Hefei, China
[2] School of Information Science and Technology, University of Science and Technology of China, Hefei, China
[3] Institute of Intelligent Innovation, Henan Academy of Sciences, Zhengzhou, China
[4] Hefei National Laboratory, University of Science and Technology of China, Hefei, China

† These authors contributed equally: J.Z., M.L., L.C.
E-mails: linjiangchen@ustc.edu.cn (L.C.); zfei@ustc.edu.cn (F.Z.); wwshang@ustc.edu.cn (W.S.); jiangj1@ustc.edu.cn (J.J.)

Table of Contents

1. Additional figures

A simple example for comparing the greedy algorithm with the FESP-B algorithm



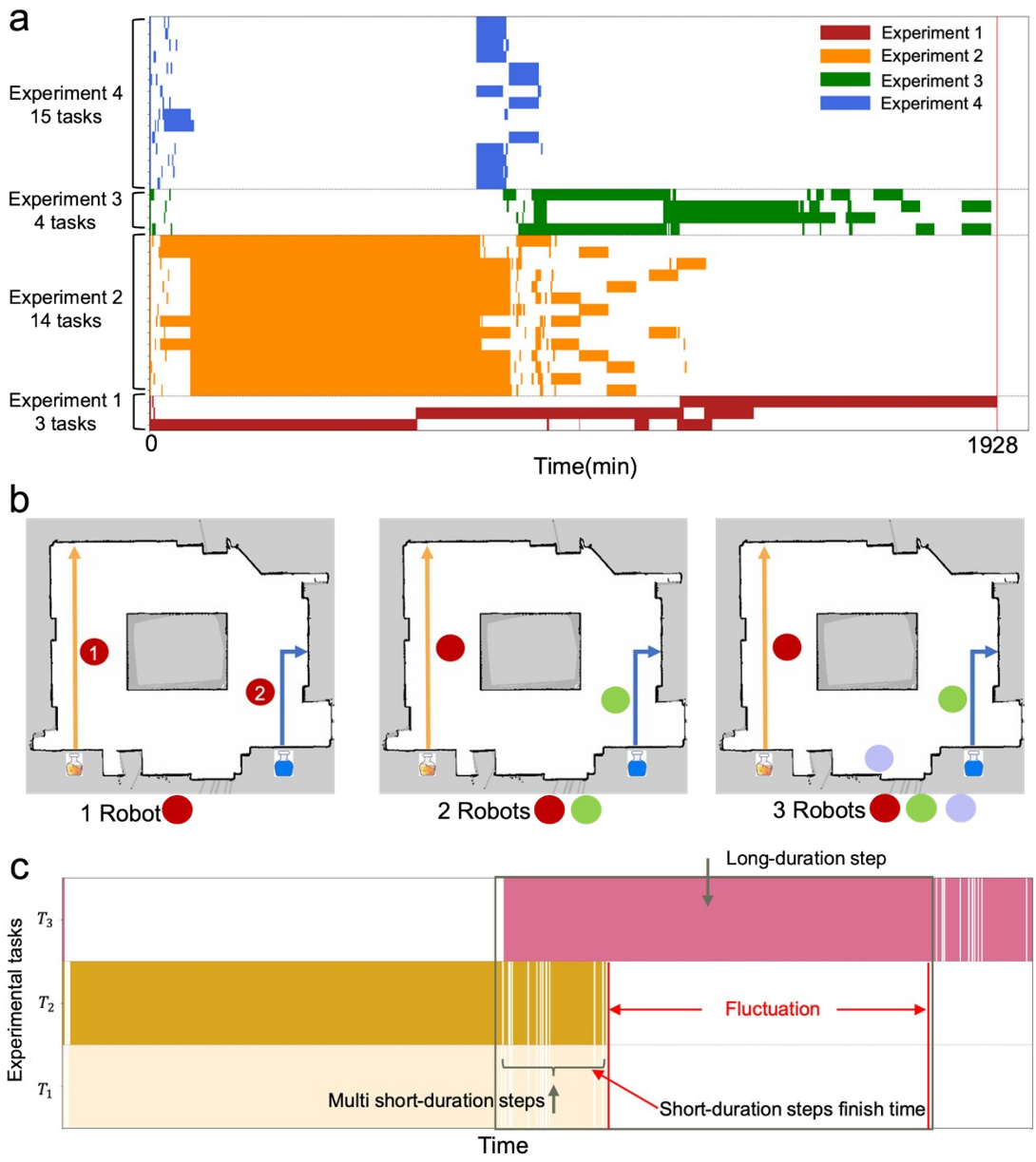
Supplementary Fig. S1. **A simple example for comparing the greedy algorithm with the FESP-B algorithm.** (a) Experimental conditions. (b) Comparison of scheduling results between FESP-B and greedy algorithm.

As illustrated in Supplementary Fig. S1a, two experimental tasks need to be performed. Experimental task 1 consists of two steps: a 3-minute liquid dispensing operation and a 30-minute drying operation. Task 2 includes only one step: a 30-minute drying operation. The liquid station and drying station have capacities of 1 and 2, respectively. Note that the drying station's capacity of 2 means it can process up to two samples simultaneously, but due to the nature of the drying operation, the two samples must start and finish drying simultaneously. If the drying station is already processing one sample, the second sample cannot be added midway.

Supplementary Fig. S1b shows a comparison of scheduling results between the greedy algorithm and FESP-B. The greedy algorithm, following the "first-come-first-served" principle, directly assigns tasks to available stations as needed. Experimental task 1 and task 2 require different stations for their first steps, so there is no resource conflict, and the greedy algorithm allows both tasks to begin simultaneously. However, when task 1's liquid dispensing step finishes, task 2's drying step is still in progress. Task 1 must wait for task 2 to complete before starting its drying step. Due to the local optimization nature of the greedy algorithm, it fails to identify the opportunity for synchronizing the drying steps of task 1 and task 2, resulting in wasted time. The total makespan is 60 minutes.

In contrast, the FESP-B algorithm models the task scheduling as an optimization problem, aiming to minimize the total makespan. By comprehensively optimizing the entire taskflow, the FESP-B algorithm identifies and leverages parallelization opportunities. During task 1's liquid dispensing step, the algorithm delays the start of task 2's drying operation to synchronize it with task 1's drying operation. This approach ensures that the drying steps of both tasks occur simultaneously, significantly reducing the total makespan to only 33 minutes.

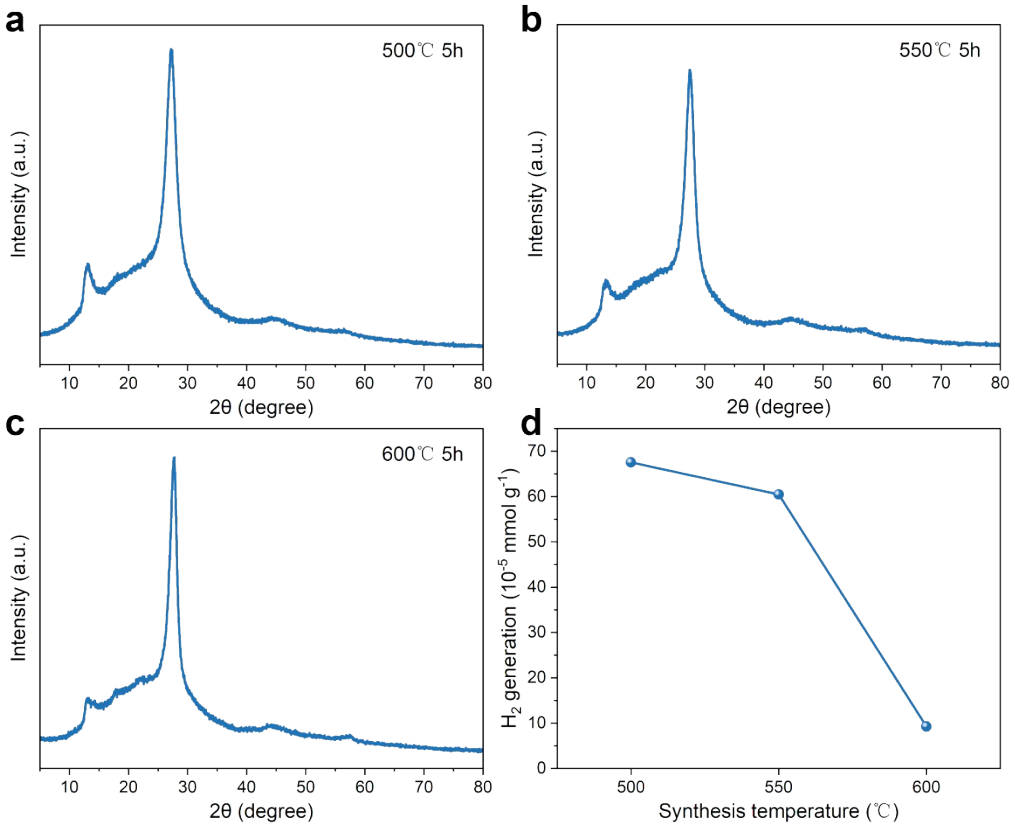
Reasons why time reduction effect diminishes as the number of robots increases



Supplementary Fig. S2. The bottleneck in optimizing the overall experimental completion time. (a) Multi-task scheduling table without considering sample transfer time. (b) Impact of different numbers of robots on the same sample transfer requirements. Suppose at a certain time step, there are two samples that need to be transferred. If there is only 1 robot, it will need to transfer both samples sequentially. If there are 2 robots, they can each transfer a sample simultaneously. If there are 3 robots, 2 robots will be used for the sample transfer, while the third robot will remain idle and not participate in the transfer for that time step. (c) Scheduling diagram of a certain experiment considering sample transfer time, with each tick on the vertical axis representing an experimental task, showing three tasks. It can be seen that short-duration completion times for tasks T_1 and T_2 within the fluctuation time range constrained by the red vertical line do not affect the final completion time of the experiment.

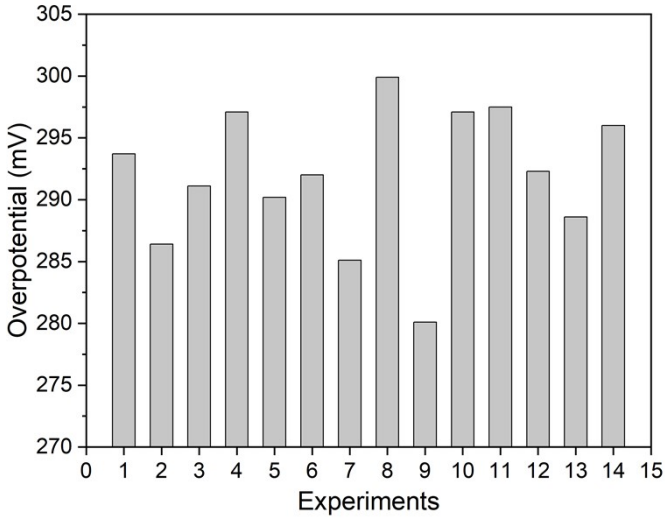
2. Chemical experiments' results

Results for experiment 1



Supplementary Fig. S3. XRD patterns of g-C_3N_4 prepared by holding at different synthesis temperatures for 5 hours: **a,** 500℃; **b,** 550℃; **c,** 600℃. **d,** Hydrogen generation of g-C_3N_4 samples synthesized at different temperatures. The H_2 amount decreases with increasing synthesis temperatures.

Results for experiment 2

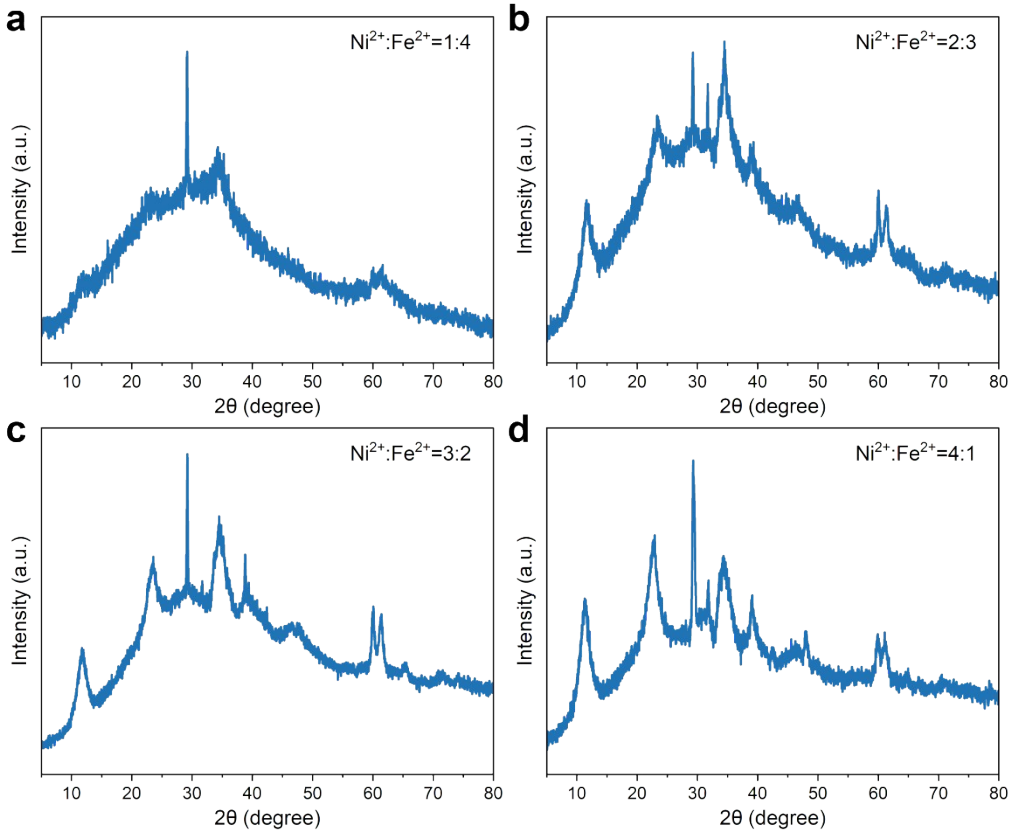


Supplementary Fig. S4. Overpotential values of MO-HECs with 14 different components at a current density of 10 mA/cm^2. The specific synthetic formula of MO-HECs and their corresponding overpotentials are shown in Table S1.

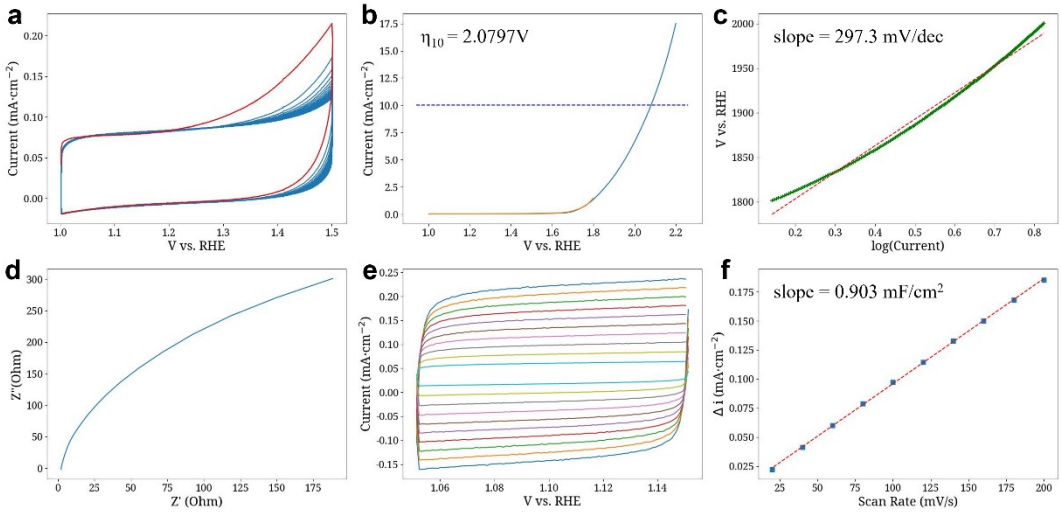
Supplementary Table S1. Summary of the synthetic formula and measured OER overpotential values of MO-HECs.

Co	Cu	Fe	Mn	Ni	overpotential
0.08	0.08	0.33	0.2	0.31	293.7
0.12	0.07	0.32	0.22	0.27	286.4
0.14	0.07	0.3	0.3	0.19	291.1
0.16	0.3	0.1	0.1	0.34	297.1
0.17	0.21	0.19	0.12	0.31	290.2
0.19	0.06	0.21	0.19	0.35	292
0.23	0.1	0.26	0.15	0.26	285.1
0.24	0.21	0.29	0.05	0.21	299.9
0.3	0.17	0.12	0.21	0.2	280.1
0.3	0.17	0.12	0.31	0.1	297.1
0.3	0.17	0.22	0.26	0.05	297.5
0.31	0.22	0.05	0.17	0.25	292.3
0.32	0.34	0.19	0.07	0.08	288.6
0.33	0.16	0.06	0.26	0.19	296

Results for experiment 3

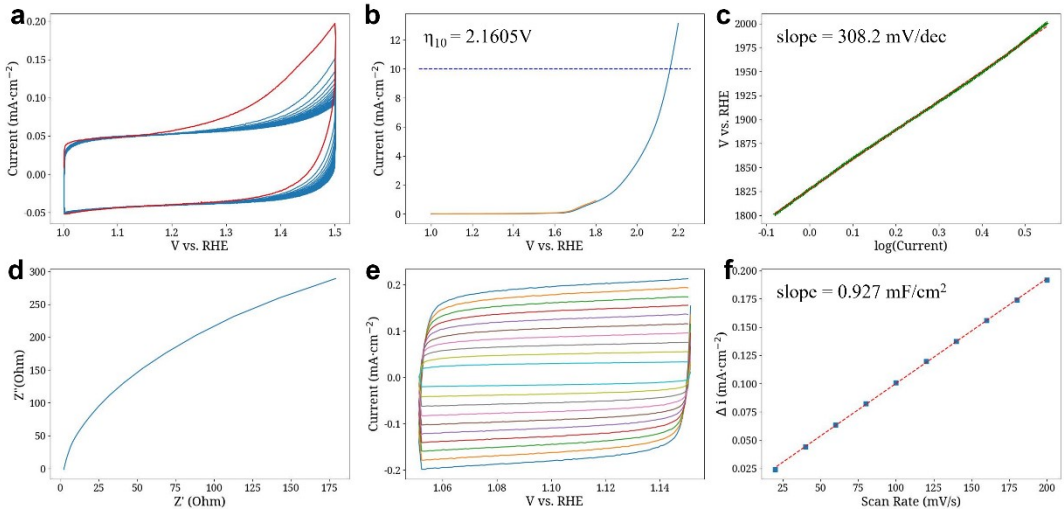


Supplementary Fig. S5. XRD patterns of NiFe-LDH with four different Ni^{2+}/Fe^{2+} ratios. The characteristic peaks of NiFe-LDH become more obvious as the Ni^{2+}/Fe^{2+} ratio increases, indicating that high Fe content makes the LDH structure tend to be amorphous.

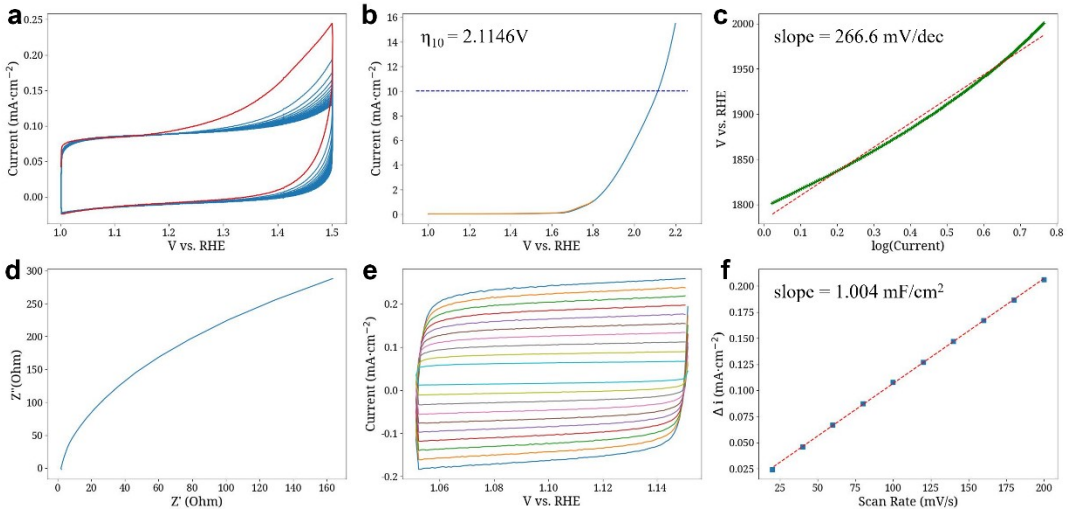


Supplementary Fig. S6. The experiment results of NiFe-LDH obtained when the feeding ratio of Ni^{2+}/Fe^{2+} was 1:4. **a,** Cyclic voltammetry(CV) curves. The CV scan rate for cyclic voltammetry curves is set at 50 mV/s, with a total of 40 cycles. RHE refers to Reversible Hydrogen Electrode. The red line indicates the first lap CV activation, and the blue line indicates the curve of CV activation for laps 2 to 40. As the number of activation turns increases, the CV curve tends to stabilize, indicating that the catalyst reaches a stable state. **b,** Polarization curves. η_{10} denotes the potential at which a current density of 10 mA/cm^2 is achieved (as indicated by the blue dashed line). **c,** Tafel slopes derived from the cyclic

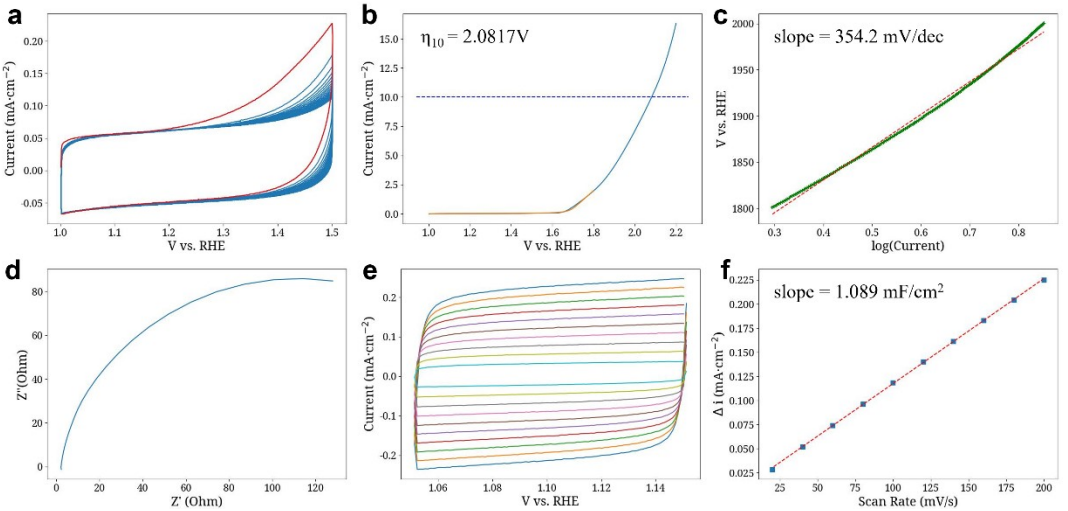
voltammetry on stationary electrodes (fitted by the red dashed line). **d,** Electrochemical impedance spectroscopy. **e,** Electrochemical cyclic voltammetry(CV) curve. The electrochemical CV curves shows the CV curves at different scan rates, with the voltage range restricted to the non-Faradaic region. A total of 10 scan rates are applied, including 20 mV/s, 40 mV/s, 60 mV/s, 80 mV/s, 100 mV/s, 120 mV/s, 140 mV/s, 160 mV/s, 180 mV/s, and 200 mV/s. As the scan rate increases, the non-Faradaic current also increases. **f,** Electrochemical active surface area (EASA). The peak current density for a 1.10 V forward scan at the same scan speed is called i_0, and the reverse scan current is called i_1. The slope of the curve fitted with $|i_0\text{-}i_1|/2$ as the vertical coordinate and the scan speed as the horizontal coordinate is Cdl, which denotes the double layer capacitance.



Supplementary Fig. S7. The experiment results of NiFe-LDH obtained when the feeding ratio of Ni^{2+}/Fe^{2+} was 2:3. **a,** Cyclic voltammetry(CV) curves. The CV scan rate for cyclic voltammetry curves is set at 50 mV/s, with a total of 40 cycles. RHE refers to Reversible Hydrogen Electrode. The red line indicates the first lap CV activation, and the blue line indicates the curve of CV activation for laps 2 to 40. As the number of activation turns increases, the CV curve tends to stabilize, indicating that the catalyst reaches a stable state. **b,** Polarization curves. η_{10} denotes the potential at which a current density of 10 mA/cm^2 is achieved (as indicated by the blue dashed line). **c,** Tafel slopes derived from the cyclic voltammetry on stationary electrodes (indicated by the red dashed line). **d,** Electrochemical impedance spectroscopy. **e,** Electrochemical cyclic voltammetry(CV) curve. The electrochemical CV curves shows the CV curves at different scan rates, with the voltage range restricted to the non-Faradaic region. A total of 10 scan rates are applied, including 20 mV/s, 40 mV/s, 60 mV/s, 80 mV/s, 100 mV/s, 120 mV/s, 140 mV/s, 160 mV/s, 180 mV/s, and 200 mV/s. As the scan rate increases, the non-Faradaic current also increases. **f,** Electrochemical active surface area (EASA). The peak current density for a 1.10 V forward scan at the same scan speed is called i_0, and the reverse scan current is called i_1. The slope of the curve fitted with $|i_0\text{-}i_1|/2$ as the vertical coordinate and the scan speed as the horizontal coordinate is Cdl, which denotes the double layer capacitance.



Supplementary Fig. S8. The experiment results of NiFe-LDH obtained when the feeding ratio of Ni^{2+}/Fe^{2+} was 3:2. **a,** Cyclic voltammetry(CV) curves. The CV scan rate for cyclic voltammetry curves is set at 50 mV/s, with a total of 40 cycles. RHE refers to Reversible Hydrogen Electrode. The red line indicates the first lap CV activation, and the blue line indicates the curve of CV activation for laps 2 to 40. As the number of activation turns increases, the CV curve tends to stabilize, indicating that the catalyst reaches a stable state. **b,** Polarization curves. η_{10} denotes the potential at which a current density of 10 mA/cm^2 is achieved (as indicated by the blue dashed line). **c,** Tafel slopes derived from the cyclic voltammetry on stationary electrodes (indicated by the red dashed line). **d,** Electrochemical impedance spectroscopy. **e,** Electrochemical cyclic voltammetry(CV) curve. The electrochemical CV curves shows the CV curves at different scan rates, with the voltage range restricted to the non-Faradaic region. A total of 10 scan rates are applied, including 20 mV/s, 40 mV/s, 60 mV/s, 80 mV/s, 100 mV/s, 120 mV/s, 140 mV/s, 160 mV/s, 180 mV/s, and 200 mV/s. As the scan rate increases, the non-Faradaic current also increases. **f,** Electrochemical active surface area (EASA). The peak current density for a 1.10 V forward scan at the same scan speed is called i_0, and the reverse scan current is called i_1. The slope of the curve fitted with $|i_0\text{-}i_1|/2$ as the vertical coordinate and the scan speed as the horizontal coordinate is Cdl, which denotes the double layer capacitance.



Supplementary Fig. S9. The experiment results of NiFe-LDH obtained when the feeding ratio of Ni2+/Fe2+ was 4:1. **a,** Cyclic voltammetry(CV) curves. The CV scan rate for cyclic voltammetry curves is set at 50 mV/s, with a total of 40 cycles. RHE refers to Reversible Hydrogen Electrode. The red line indicates the first lap CV activation, and the blue line indicates the curve of CV activation for laps 2 to 40. As the number of activation turns increases, the CV curve tends to stabilize, indicating that the

catalyst reaches a stable state. **b,** Polarization curves. η_{10} denotes the potential at which a current density of 10 mA/cm^2 is achieved (as indicated by the blue dashed line). **c,** Tafel slopes derived from the cyclic voltammetry on stationary electrodes (indicated by the red dashed line). **d,** Electrochemical impedance spectroscopy. **e,** Electrochemical cyclic voltammetry(CV) curve. The electrochemical CV curves shows the CV curves at different scan rates, with the voltage range restricted to the non-Faradaic region. A total of 10 scan rates are applied, including 20 mV/s, 40 mV/s, 60 mV/s, 80 mV/s, 100 mV/s, 120 mV/s, 140 mV/s, 160 mV/s, 180 mV/s, and 200 mV/s. As the scan rate increases, the non-Faradaic current also increases. **f,** Electrochemical active surface area (EASA). The peak current density for a 1.10 V forward scan at the same scan speed is called i_0, and the reverse scan current is called i_1. The slope of the curve fitted with $|i_0\text{-}i_1|/2$ as the vertical coordinate and the scan speed as the horizontal coordinate is Cdl, which denotes the double layer capacitance.

The detailed parameters of the electrocatalytic experiment are as follows:

- Parameter setting of dropping volumes: 0.2 mL.
- Parameter setting of CV activation:

Init E (V)	High E (V)	Low E (V)	Init P/N	Scan Rate (V/s)	Segment	Sample Interval (V)	Quiet Time (sec)	Sensitivity (A/V)
0	0.5	0	P	0.05	40	0.001	2	0.1

- Parameter setting of electrochemical active surface area (EASA):

Init E (V)	High E (V)	Low E (V)	Init P/N	Scan Rate (V/s)	Segment	Sample Interval (V)	Quiet Time (sec)	Sensitivity (A/V)
0.05	0.15	0.05	P	0.02	20	0.001	2	0.1

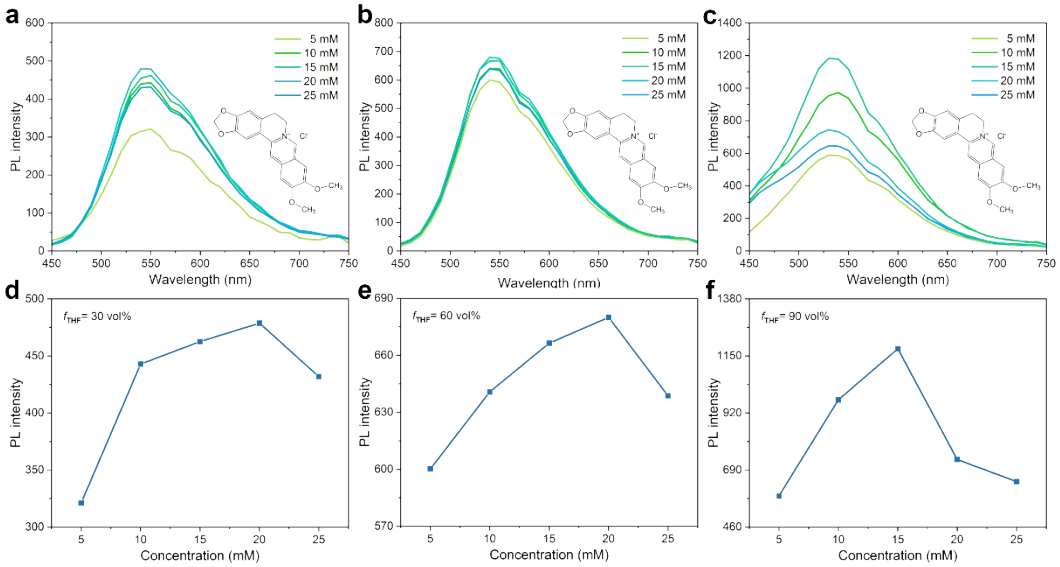
- Parameter setting of electrochemical impedance spectroscopy (EIS):

Init E (V)	High Frequency (Hz)	Low Frequency (Hz)	Amplitude (V)	Quiet Time (sec)	Cycles (.1-1Hz)
0.6	1e+5	0.1	0.005	2	1

- Parameter setting of linear sweep voltammetry (LSV):

Init E (V)	Final E (V)	Scan Rate (V/s)	Sample Interval (V)	Quiet Time (sec)	Sensitivity (A/V)
0	1.2	0.005	0.001	2	0.1

Results for experiment 4



Supplementary Fig. S10. Relationship between AIE luminescence and tetrahydrofuran (THF) concentration. **a-c,** PL spectra of BBR chloride in THF/water mixtures with different THF fractions (**a,** 30 vol.%; **b,** 60 vol.%; **c,** 90 vol.%). Excitation wavelength: 405nm. **d-f,** Peak fluorescence intensity with increased concentrations of BBR.

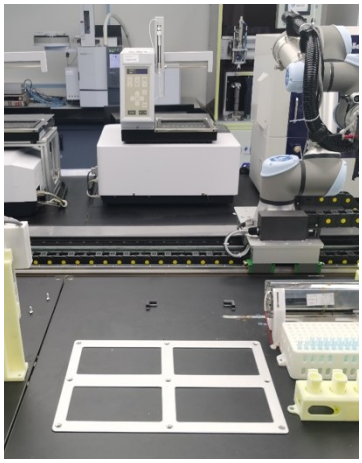
3. Robots and stations

The multi-robot-multi-task experiments involve two mobile robots and one benchtop robot, with the stations including the sample rack station, the multi-robot exchange station, the liquid station, the solid station, the magnetic stirring station, the centrifuge station, the calcination station, the aspiration station, the drying station, the encapsulation station, the photocatalytic station, the gas chromatography station, the fluorescence spectroscopy station, the PXRD station, and the electrocatalytic station.

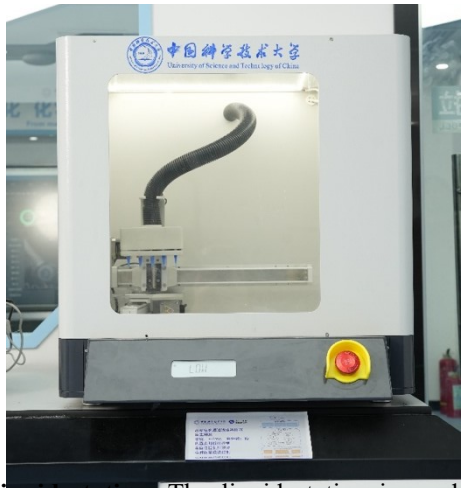
Supplementary Fig. S11. **Benchtop robot.** The benchtop robot is used to flexibly clamp and transfer samples on an integrated cl eparation and measurement operations.

Supplementary Fig. S12. **Mobile robot.** The mobile robot is used to carry and transport samples in an automated lab for an all-round process from sample preparation to characterization and testing.

Supplementary Fig. S13. **Sample rack station.** The sample rack station is used to retrieve and return samples.



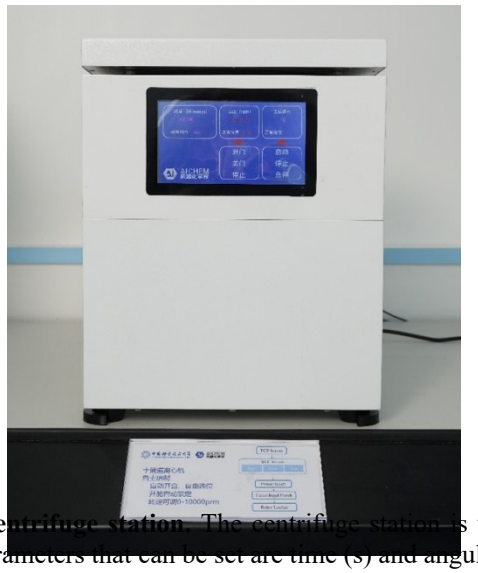
Supplementary Fig. S14. **Multi-robot exchange station.** The multi-robot exchange station is used for transferring samples between the high-throughput platform and other experimental stations in the laboratory.



Supplementary Fig. S15. **Liquid station.** The liquid station is used to add pre-configured liquid reagents (ten different reagents can be placed at the same time). Parameters that can be set are mass (g) and the type of liquid.

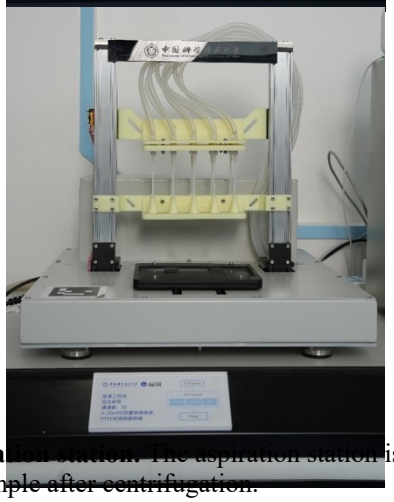
Supplementary Fig. S16. **Solid station.** The solid station is used to add pre-filled solid reagents. Parameters that can be set are mass (g) and the type of solid.

Supplementary Fig. S17. **Magnetic stirring station.** The magnetic stirring station is used to mix and stir different reagents (heating permitted) to obtain homogeneous solutions or sample dispersions. Parameters that can be set are time (s), temperature (°C) and angular velocity (rpm).

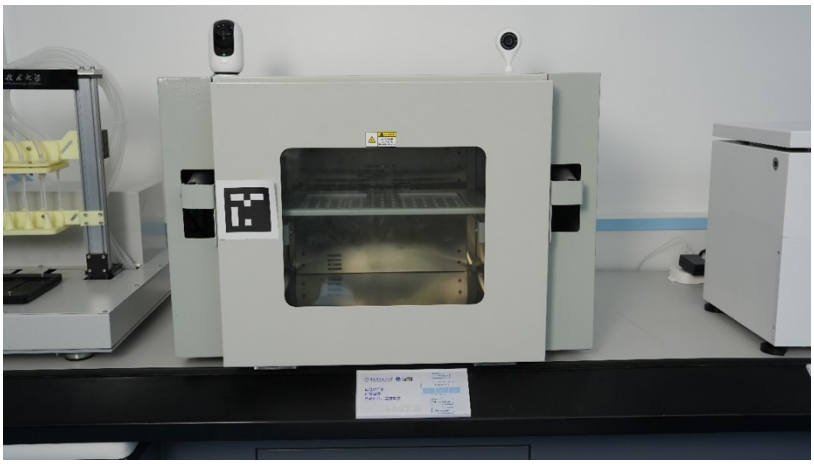


Supplementary Fig. S18. **Centrifuge station.** The centrifuge station is used to separate mixtures of liquids and solid particles. Parameters that can be set are time (s) and angular velocity (rpm).

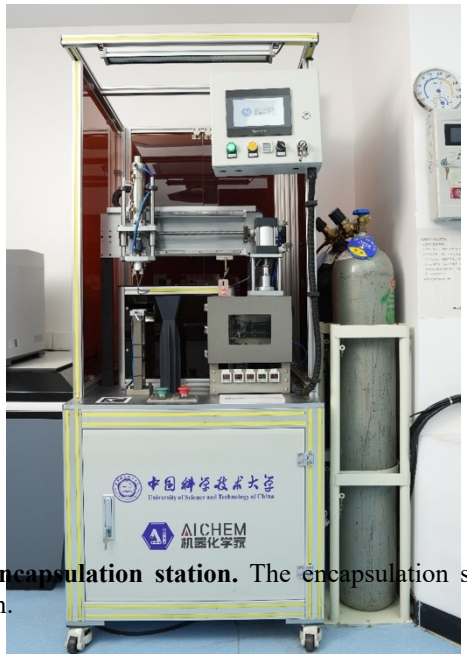
Supplementary Fig. S19. **Calcination station.** The calcination station is used to calcine and heat samples. Parameters that can be set are time (s), temperature (°C) and heating rate (°C/min).



Supplementary Fig. S20. **Aspiration station.** The aspiration station is used to aspirate the waste liquid from the upper part inside the sample after centrifugation.

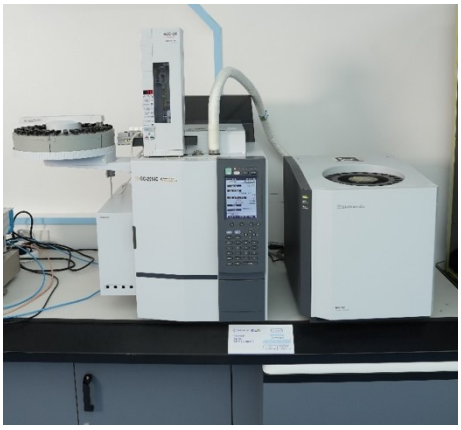


Supplementary Fig. S21. **Drying station.** The drying station is used to dry samples. Parameters that can be set are time (s) and temperature (°C).

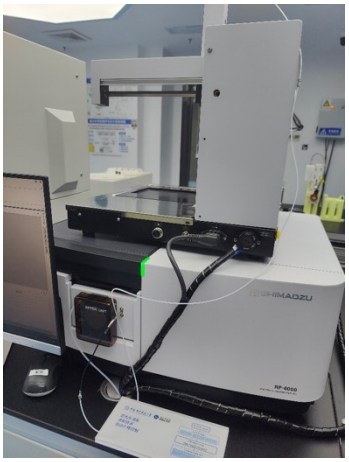


Supplementary Fig. S22. **Encapsulation station.** The encapsulation station is used to encapsulate reaction vessels under vacuum.

Supplementary Fig. S23. **Photocatalytic station.** The photocatalytic station is used to carry out light irradiation with simultaneous agitation. The parameter that can be set is the time (s).

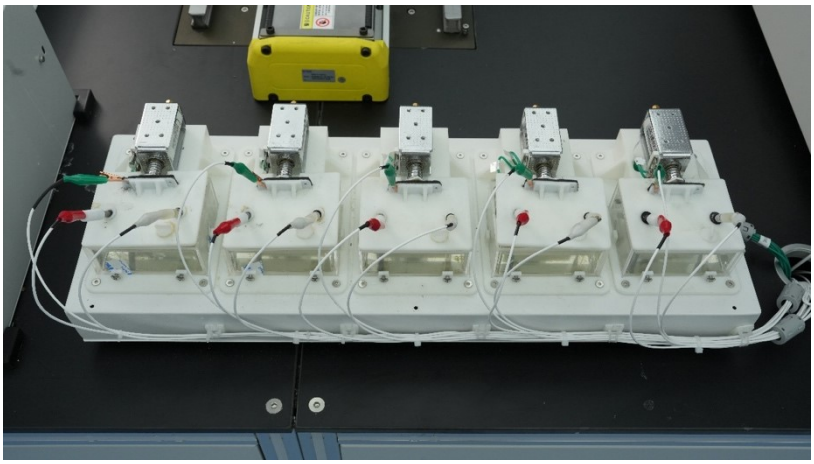


Supplementary Fig. S24. **Gas chromatography station.** The gas chromatography station is used to measure the gas production of reaction systems.



Supplementary Fig. S25. **Fluorescence spectroscopy station.** The fluorescence spectroscopy station is used to measure the fluorescence spectrum of samples (with samples prepared on quartz substrates).

Supplementary Fig. S26. **PXRD station.** The PXRD station is used to determine the phase composition of samples (with samples prepared on quartz substrates).



Supplementary Fig. S27. **Electrocatalytic station.** The electrocatalytic station is used to perform electrocatalytic tests by a three-electrode system (with samples prepared on carbon paper).

4. A constraint programming approach for FESP-B

Regarding the constraint characteristics of chemical experiment tasks and the scalability of solution approach, we have chosen to modify the CP approach based on FJSP-B[32] to address the FESP-B. We use the IBM ILOG CP Optimizer[36] to solve this problem, and for details on the modeling language, we refer the reader to [36]. Below, we will provide the pseudocode and corresponding explanations for the CP solution procedure.

The algorithm receives all the experimental sequence information, $expr_{data}$, which includes detailed parameters for each experimental step, a list of stations in the laboratory, *all_stations*, and the capacity information of all stations, *capacity*, where $capacity_m$ represents the capacity of station m, as inputs. The output of the algorithm is a scheduling table that includes the start time, end time, and the name of the processing station for each experimental step. In lines 1-5 of the pseudo-code, variables related to constraint programming are initialized: $scheduling_{res}$ represents the scheduling result, containing interval variables for each experimental step; *s2m* contains interval variables for eligible stations corresponding to each experimental step, set as *optional* because each step can only be assigned to one station from the list of eligible stations; *m2itv* contains sequences of interval variables for each station, indicating the experimental steps each station needs to process; $load_m$ is a cumulative function representing the sample capacity of each station m at each moment, which should not exceed the station's capacity; $status_m$ is a state function representing the state of each station m, which must be the same within the same batch. Lines 6-12 assign values to $scheduling_{res}$, *s2m*, and *m2itv* variables based on the experimental sequence information. Lines 13-16 ensure that each experimental step is assigned to exactly one station. Lines 17-20 maintain the sequential execution of operations in the experimental sequence, ensuring the logical coherence of the experimental task flow. Lines 21-24 enforce concurrent consistency by requiring that experiments processed in the same batch start and end simultaneously. Lines 25-33 guarantee that the maximum number of samples processed simultaneously by a station does not exceed its capacity. Lines 34-43 ensure chemical experimental operation precision by allowing parallel processing only for experimental steps with identical parameters. Lines 44-45 define runtime constraints for stations, where new constraints need to be added for stations with special runtime requirements, such as a centrifuge requiring a specific number of samples to be placed symmetrically to maintain balance. If the centrifuge has an even number of channels, this can be achieved by simply imposing a large penalty on cases where an odd number of samples are allocated in the same batch. Finally, lines 46-47 define the objective function to evaluate the scheduling results.

Algorithm 1: Constraint programming approach for FESP-B

Input: instances of all experimental tasks, $expr_{data}$, a list of stations in laboratory, *all_stations*, and capacity information of all stations, *capacity*, where the capacity of station m is denoted as $capacity_m$.

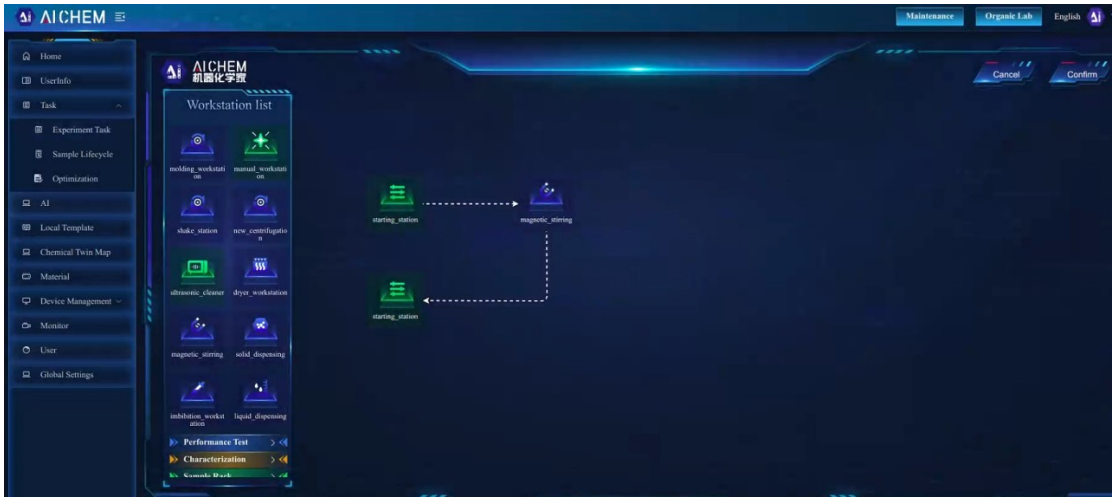
Output: scheduling result, include the start and end times of each step for each experimental task, as well as the names of the assigned stations.

Initial result: $scheduling_{res} \leftarrow \emptyset$
Initial varible: $s2m \leftarrow \emptyset$
Initial varible: $m2itv \leftarrow \emptyset$
Initial cumulative loads: $load_m \leftarrow step\_at(0, 0)$
Initial station status: $status_m \leftarrow status\_function(m)$
for *task* in $expr_{data}$ and *step* in *task* **do**:
  $scheduling_{res}[task, step] \leftarrow interval\_var()$
  for *m* in $step_{eligible\_stations}$ **do**:
    $s2m[task, step, m] \leftarrow interval\_var(size=step_{processing\ time}, optional=True)$
    $m2itv[m].append(s2m[task, step, m])$
  end for
end for
# *Routing*
for *task* in $expr_{data}$ and *step* in *task* **do**:
  $alternative(scheduling_{res}[task, step], [s2m[task, step, m]\ for\ all\ m])$
end for
# *Sequencing*
for *task* in $expr_{data}$ and *task* in *expr* **do**:
  $endBeforeStart(scheduling_{res}[task, step], scheduling_{res}[task, step+1])$
end for
# *Synchronizing*
for *m* in *all_stations* and *itv* in $m2itv[m]$ **do**:
  $always\_equal(state_m, itv, m, isStartAligned=True, isEndAligned=True)$
end for
# *Batching*
for *task* in $expr_{data}$ and *step* in *task* **do**
  for *m* in *all_stations* **do**
    $load_m \mathrel{+}= pulse(s2m[task, step, m], 1)$
  end for
end for
for *m* in *all_stations* **do**
    $load_m \leq capacity_m$
end for
# *Exclusive*
for *task* in $expr_{data}$ and *step* in *task* **do**
  for *task'* in $expr_{data}$ and *step'* in *task'* **do**
    for *m* in *all_stations* **do**
      if $step_{param} \neq step'_{param}$ **do**

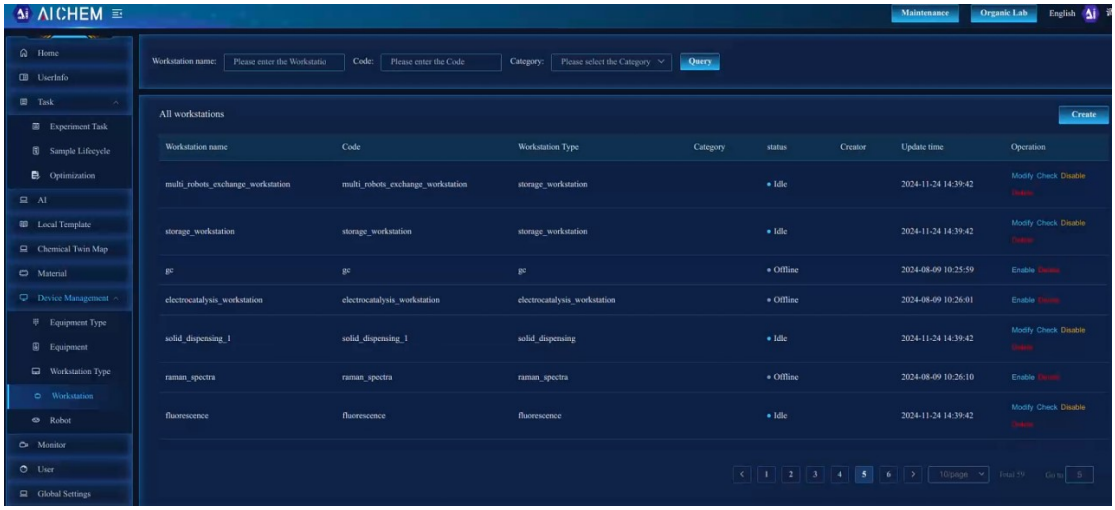
```
            noOverlap(s2m[task, step, m], s2m[task', step', m])
          end if
        end for
      end for
    end for
    # Runtime Constraint
    # add runtime constraint
    # Objective
    minmize max[end_of(scheduling_res[task, step]) for all task, step]
    return scheduling_res, m2itv
```

5. Interactive GUI screenshots

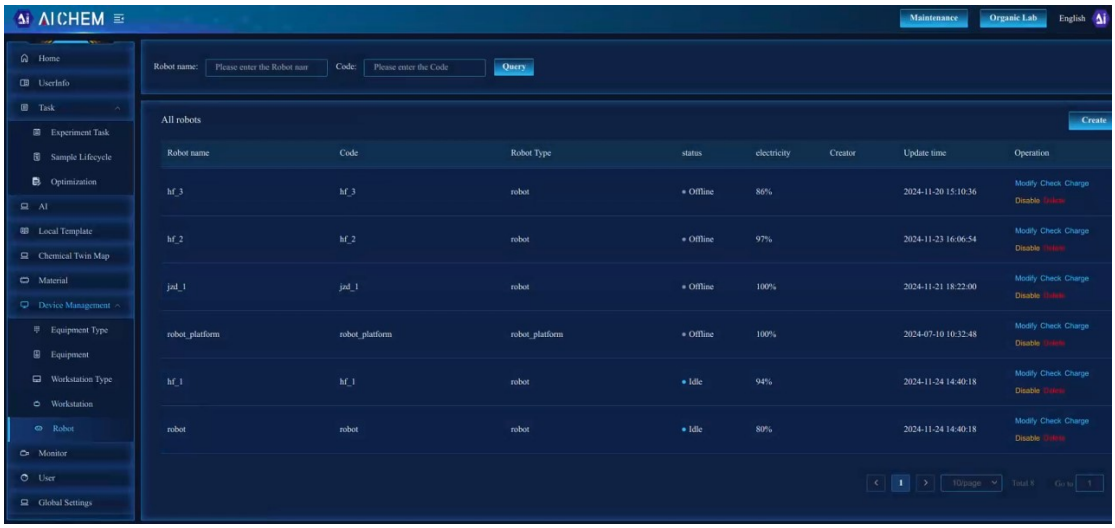
The interactive task management GUI acts as the interface bridge between the multi-robot-multi-task scheduling system and research users, providing an intuitive platform for experimental task management. Here, we present some interface screenshots to illustrate the functionalities of the GUI.



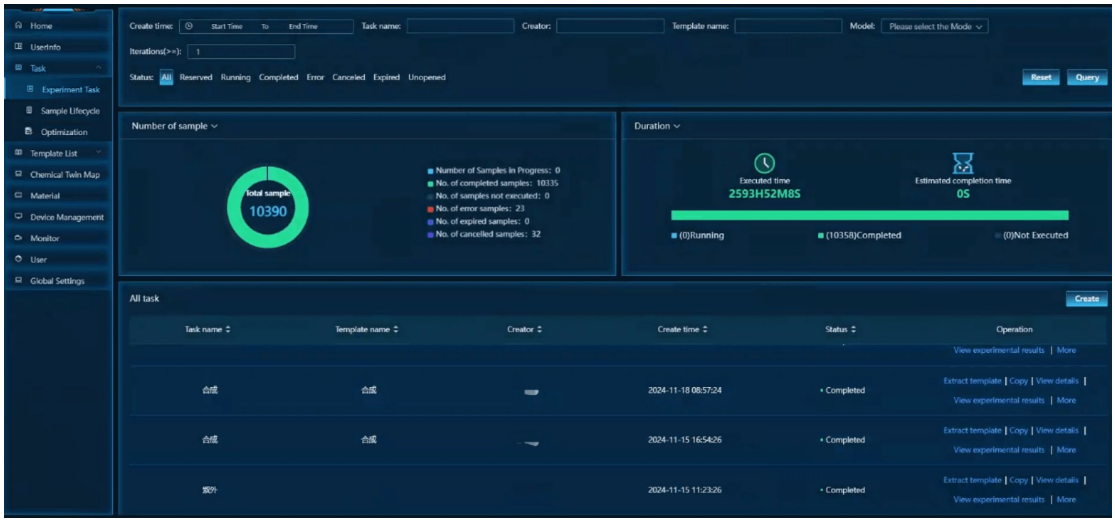
Supplementary Fig. S28. **The interface for users to assign experiments.** Users can select components from the left panel, drag them into the central area to connect and form a task flow, and confirm the assignment in the top-right corner.



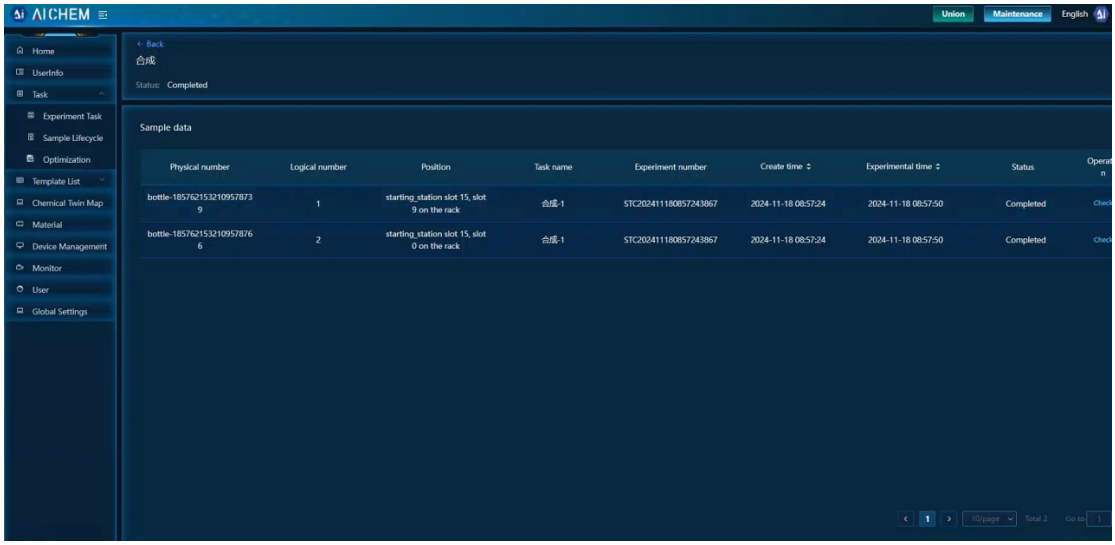
Supplementary Fig. S29. **The panel for monitoring station status.** Users can monitor the status of stations through this panel.



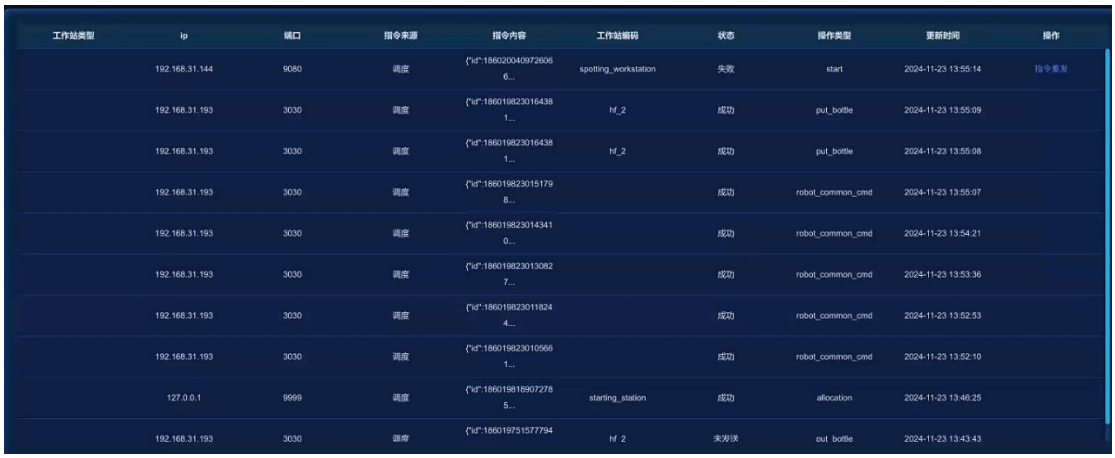
Supplementary Fig. S30. **The panel for monitoring robot status.** Users can monitor the status of robots through this panel.



Supplementary Fig. S31. **The interface for viewing assigned experimental tasks.** Users can see previously assigned tasks here.



Supplementary Fig. S32. **The interface for viewing detailed experiment information.** Users can see the locations of the samples (bottles) corresponding to the experiment.



Supplementary Fig. S33. **The interface for viewing operation records.** Users can view the status of commands through the command management page. If there are commands with a failure status, users can identify the issues and then click the "Resend Command" button to continue the experiment.

6. The content of information encoding for robots and workstations, opertation records and task pool.

The information of robots and workstations, task pool, and operation command records are serialized into JSON character streams during management and transmission. Below are some examples.

```
{
    "id": 1602558355258351616, # Workstation ID, globally unique, representing a
    workstation.
    "workstationType": "liquid_dispensing", # Workstation Type, not unique, different
    workstations can belong to the same type. For example, Workstation A_1 and A_2
    can both belong to Type A.
    "name": "liquid_dispensing", # Workstation Name. Can be any value, using user-
    friendly language.
    "code": "liquid_dispensing", # Workstation Code. Globally unique, similar to ID
    but can use user-friendly language.
    "status": "IDLE", # Workstation Status, indicating whether the workstation is
    idle or not.
    "electricityQuantity": null, # Battery Level, not applicable to workstations that
    are plugged in; used for robots, which are logically considered workstations.
    "bottleSlotCount": 160, # Workstation Capacity, indicating the maximum number of
    bottles the workstation can handle simultaneously in a batch.
    "bottleCount": 1, # Number of bottles currently at the workstation.
    "isRobot": 0, # Indicates whether the workstation is a robot.
    "machineList": [ # Components in the workstation, with each member of the list
    representing a component of the workstation.
        {
        "machineTypeCode": "liquid_dispensing", # Component Type.
        "channelList": [ # Channels. For example, information about the channels for
        different materials in a liquid dispensing station can be stored here.
                {
                "machineId": 1598229081986572288,
                "materialCode": "d02eae53-1819-4a04-a4b5-8fd16edc14b2", # Identify
                materials in the database.
                "rest": 996.0, # Material stock level.
                "formula": "C2H5OH", # formula of materials
                "alias": "C2H5OH", # user-friendly consideration
                "concentration": 1.0, # Concentration of materials.
                "concentrationUnitCode": "mol/l", # The concentration unit,
                material, concentration, and concentration unit together determine
                the properties of a channel in the liquid workstation.
                }
            ],
        }
    ],
    "bottleList": [ # Information of bottles currently at the workstation, used for
    tracking bottle locations.
        {
        "bottleCode": "bottle-1809399055726551045", # Unique bottle code.
        "rackId": 1809399055726551043, # Unique rack code.
        }
    ],
    "sectionList": [ # Section within the Laboratory. Laboratory equipment may be
    divided into multiple sections. In our laboratory, workstations are divided into
    laboratory sections and high-throughput platform sections, as described in the
    paper.
        {"sectionCode": "lab", "sectionName": "lab"}
    ],
    "remark": null, # Personalized remarks for the workstation.
},
```

Supplementary Fig. S34. **An example of the information encoding for a general station.** Each keyword in the JSON stores a piece of information.

```
{
"id": 1612694273754807297, # Robot ID, globally unique, representing a robot.
"workstationType": "robot", # # Robot Type, not unique, different workstations can
belong to the same type. In terms of logical encoding, we treat the robot as a
workstation as well.
"name": "robot_0", # Robot Name. Can be any value, using user-friendly language.
"code": "robot_0", # Robot Code. Globally unique, similar to ID but can use user-
friendly language.
"status": "IDLE", # Robot status.There are generally three states: idle, busy, and
charging. When either the chassis or the robotic arm is moving, the robot's status is
considered busy.
"electricityQuantity": 95.0, # Robot battery status
"currentStation": "starting_station", # Robot current station
"bottleSlotCount": 40, # The number of samples the robot can carry.
"bottleCount": 0, # The number of samples the robot is carrying.
"isRobot": 1, # Indicates the workstation is a robot.
"machineList": [{"machineTypeCode": "robot", "channelList": null}], # This field is
compatible with the workstation and is described using a unified template, set to null.
"bottleList": [ # Information of bottles currently at the robot, used for tracking
bottle locations.
    {
        "bottleCode": "bottle-1809399055726551045", # Unique bottle code.
        "rackId": 1809399055726551043, # Unique rack code.
    }
],
"sectionList": [{"sectionCode": "lab", "sectionName": "lab"}], # Indicates which
laboratory the robot belongs to.
"remark": null, # Personalized remarks for the workstation.
}
```

Supplementary Fig. S35. **An example of the information encoding for a general robot.** Each keyword in the JSON stores a piece of information. To maintain the generality of the template, we treat the robot as a station as well. Therefore, many parameters in the descriptions of robots and stations are consistent.

```
{
"name": "simple_example",# Experiment Name.
"steps": [# Specific steps of the Experiments.
    {
        "id": 5609691746369538,# step ID, unique
        "index": 1,# Sequence number of the step in the entire experiment.
        "workstation": "starting_station",# Workstation name for performing the
        step.
        "detail": [# Detailed information of the step.
            {
            "material": null, # If it is a dispensing workstation, material
            information will be filled in.
            "concentration": null,# If it is a dispensing workstation, material
            concentration information will be filled in.
            "value": null,# If it is a dispensing workstation, material value
            information will be filled in.
            "vials_no": "1", # Relative number of the bottle to be processed in
            the experiment.
            "actual_no": null,# When taking bottles from the sample rack, the
            system assigns the absolute number, i.e., bottle-code, to the bottles
            included in the task. From this point on, the relative number
            vials_no of the bottle in the task will be bound to the absolute
            number.
            "material_code": null,# Each material corresponds to a material_code.
            "concentration_unit_code": null # The same material may have
            different concentrations, and the scheduling algorithm needs to
            consider concentration information to assign bottles from different
            steps to different dispensing stations.
            }
        ],
        "time": null, # Time parameter corresponding to the step.
        "parameters": [# Other parameters corresponding to the step.
                {
                    "machineTypeCode": "starting_station",
                    "param": {"bottle_count": "1",}
                }
            ],
        },
        {
        "id": 5609691746369537,
        "index": 2,
        "workstation": "magnetic_stirring",
        "detail": [
            {
                "material": null,
                "concentration": null,
                "value": null,
                "vials_no": "1",
                "actual_no": null,
                "material_code": null,
                "concentration_unit_code": null
            }
        ],
        "time": 10.0,
        "parameters": [
            {
            "machineTypeCode": "magnetic_stirring",
            "param": {
            "temperature": "20",
            "speed": "500"
            }
        }
        ],
    }
],
"expr_no": "STC202411110956235774",# Experiments ID, unique.
"vials_count": 1 # Number of Bottles Included in the Experiment. Task scheduling is
done on a per-bottle basis.
}
```

Supplementary Fig. S36. **An example of the information encoding for a experiment.** Each keyword in the JSON stores a piece of information.

```
{
"id": 1859477102911426560, # Identifies a unique record.
"createdTime": "2024-11-11 14:01:03", # Creation time of the Instruction.
"updatedTime": "2024-11-11 14:01:05", # Update time of the Instruction, recorded when
the instruction is created or when the instruction status changes from processing to
finish.
"stepId": 6000871142883328, # The step_id described in the experimental task
corresponding to the operation for this instruction.
"index": 1, # Corresponds to the step index of the experiment.
"operation": "take", # Operation, as described in the paper, generally includes three
types: take, put, and start. Among them, take and put are performed by robots, while
start is performed by the workstation.
"status": "finish",# Operation Status, typically represented by processing and finish.
"workstation": "starting_station", # Workstation code, explained in the workstation
info.
"workstationType": "starting_station",# Workstation type, explained in the workstation
info.
"finishTime": "1732168865482", # Finish time of the Instruction.
"time": null, # Time parameter of the Instruction, typically only present in start
instructions for workstations that have a time parameter.
"scheduleId": 00000001, # Instruction Number for the Scheduling Algorithm.
"robot": "robot_1",# Which robot participated in this instruction.
"expr_no": "STC202411211400538312",# Corresponds to the no of the experiment.
"bottle_code": "bottle-12345678", #Bottle code of the sample operated by the
instruction.
"material_list": []# If it is a dispensing workstation, material information will be
stored here.
}
```

Supplementary Fig. S37. **An example of the information encoding for a operation record.** Each keyword in the JSON stores a piece of information.

7. The instruction set for robots and stations.

The high-level API functions for robots, include definitions for four types of operations, such as visual positioning, motion planning, experimental operations, and station communication. For stations, the high-level API is encapsulated in a “start” command. The detailed APIs are as follows:

Supplementary Table S2. The high-level API functions for robots

API type	API function
Visual positioning	**1. locate(station_name: str) -> pose** #Get the manipulation pose of the station_name station.
Motion planning	**2. move(station_name: str) -> None** #Move to the station_name station.
Experimental operations	**3. take_bottle(rack_idx: int, bottle_idx: int, station_name: str) -> None** #Take the bottle defined by rack_idx and bottle_idx from the station_name station to the robot. **4. put_bottle(rack_idx: int, bottle_idx: int, station_name: str) -> None** #Put the bottle defined by rack_idx and bottle_idx from the robot to the station_name station.

Supplementary Table S3. The high-level API functions for stations

API type	API function
Experimental operations	**1. start(ip_info: str, param: str) -> None** #Send a start command with parameters “param” to the station at IP address “ip_info”.

8. Supplementary information for the four experiments

The detailed procedural information for the four real experiments included in the manuscript is listed here.

Supplementary Table S4. The detailed information of the four real experiments in the paper.

<table>
<tr><th>Experiment</th><th>Experimental steps information</th></tr>
<tr><td>Experiment 1
(3 samples)</td><td>1. Sample rack station.
2. Solid station: add 5 g of dicyandiamide powder.
3. Calcination station, calcination for a total of 10 hours, including heating and cooling time (3 groups).
<table>
<tr><td>500℃</td><td>550℃</td><td>600℃</td></tr>
</table>
4. Liquid station: add 2 g of ethanol.
5. Magnetic stirring station: 20°C, 5 minutes.
6. PXRD station.
7. Liquid station: add 1g of triethanolamine and 9g of aqueous (containing 9 mg H2PtCl6)
8. Encapsulation station.
10. Photocatalytic station, 1 hour.
11. Gas chromatography station.
12. Sample rack station.</td></tr>
<tr><td>Experiment 2
(14 samples)</td><td>1. Sample rack station.
2. Liquid station: add:
(1) A 1 g mixed solution of DMF, anhydrous ethanol, and water in a 1:1:1 volume ratio, containing 0.0364 g of triethylamine.
(2) A 10 g metal salt-DMF solution (14 groups).
(3) A 3 g DMF solution containing 0.5 g of trimesic acid.
<table>
<tr><th>Copper Nitrate/g</th><th>Iron Nitrate/g</th><th>Nickel Nitrate/g</th><th>Manganese Nitrate/g</th><th>Cobalt Nitrate/g</th></tr>
<tr><td>0.8</td><td>3.3</td><td>3.1</td><td>2</td><td>0.8</td></tr>
<tr><td>0.7</td><td>3.2</td><td>2.7</td><td>2.2</td><td>1.2</td></tr>
<tr><td>0.7</td><td>3</td><td>1.9</td><td>3</td><td>1.4</td></tr>
<tr><td>3</td><td>1</td><td>3.4</td><td>1</td><td>1.6</td></tr>
<tr><td>2.1</td><td>1.9</td><td>3.1</td><td>1.2</td><td>1.7</td></tr>
<tr><td>0.6</td><td>2.1</td><td>3.5</td><td>1.9</td><td>1.9</td></tr>
<tr><td>1</td><td>2.6</td><td>2.6</td><td>1.5</td><td>2.3</td></tr>
<tr><td>2.1</td><td>2.9</td><td>2.1</td><td>0.5</td><td>2.4</td></tr>
<tr><td>1.7</td><td>1.2</td><td>2</td><td>2.1</td><td>3</td></tr>
<tr><td>1.7</td><td>1.2</td><td>1</td><td>3.1</td><td>3</td></tr>
<tr><td>1.7</td><td>2.2</td><td>0.5</td><td>2.6</td><td>3</td></tr>
<tr><td>2.2</td><td>0.5</td><td>2.5</td><td>1.7</td><td>3.1</td></tr>
<tr><td>3.4</td><td>1.9</td><td>0.8</td><td>0.7</td><td>3.2</td></tr>
<tr><td>1.6</td><td>0.6</td><td>1.9</td><td>2.6</td><td>3.3</td></tr>
</table>
3. Magnetic stirring station: 20°C, 720 minutes.</td></tr>
</table>

<table>
<tr><td></td><td>4. Centrifuge station: 5000 rpm, 5 minutes.
5. Aspiration station.
6. Liquid station: add 3g of anhydrous ethanol.
7. Magnetic stirring station: 20°C, 5 minutes.
8. Centrifuge station: 5000 rpm, 5 minutes.
9. Aspiration station.
10. Liquid station: add 6 g of Nafion solution.
11. Electrocatalytic station.
12. Sample rack station.</td></tr>
<tr><td>Experiment 3
(4 samples)</td><td>1. Sample rack station.
2. Liquid station: 0.3 M nickel nitrate solution (adjusted to pH ~10) and 0.3 M iron nitrate solution (adjusted to pH ~10) (4 groups).
<table><tr><th>Nickel Nitrate /g</th><th>Iron Nitrate /g</th></tr><tr><td>1</td><td>4</td></tr><tr><td>2</td><td>3</td></tr><tr><td>3</td><td>2</td></tr><tr><td>4</td><td>1</td></tr></table>3. Magnetic stirring station: 20°C, 10 minutes.
4. Liquid station: add 3 g of sodium hydroxide-sodium carbonate aqueous solution.
5. Magnetic stirring station: 20°C, 60 minutes.
6. Drying station, 80°C, 5 hours.
7. Centrifuge station: 5000 rpm, 5 minutes.
8. Aspiration station.
9. Liquid station: add 5 g of deionized water.
10. Magnetic stirring station: 20°C, 5 minutes.
11. Centrifuge station: 5000 rpm, 5 minutes.
12. Aspiration station.
13. Liquid station: add 5 g of ethanol.
14. Magnetic stirring station: 20°C, 5 minutes.
15. Centrifuge station: 5000 rpm, 5 minutes.
16. Aspiration station.
17. Liquid station: add 1.5 g of ethanol.
18. Magnetic stirring station: 20°C, 5 minutes.
19. PXRD station.
20. Liquid station: add 3 g of Nafion solution.
21. Electrocatalytic station.
22. Sample rack station.</td></tr>
<tr><td>Experiment 4
(15 samples)</td><td>1. Sample rack station.
2. Liquid station: add ultrapure water and THF (3 groups).
<table><tr><th>H2O/g</th><th>THF/g</th></tr><tr><td>7</td><td>2.66</td></tr><tr><td>4</td><td>5.32</td></tr><tr><td>1</td><td>7.98</td></tr></table>3. Solid station: add Berberine (5 groups).
<table><tr><th>Berberine/g</th></tr><tr><td>0.0168</td></tr></table></td></tr>
</table>

	0.0336
	0.0504
	0.0673
	0.0841
	4. Magnetic stirring station: 20°C, 60 minutes. 5. Fluorescence spectroscopy station. 6. Sample rack station.