

# Supporting Information

# **Integrating Machine Learning and Large Language Models to Advance Exploration of Electrochemical Reactions**

Z. Zheng, F. Florit, B. Jin, H. Wu, S.-C. Li, K. Y. Nandiwale, C. A. Salazar, J. G. Mustakis, W. H. Green, K. F. Jensen\*

# **Supporting Information**

# Integrating Machine Learning and Large Language Models to Advance Exploration of Electrochemical Reactions

Zhiling Zheng,<sup>†</sup> Federico Florit,<sup>†</sup> Brooke Jin,<sup>†</sup> Haoyang Wu,<sup>†</sup> Shih-Cheng Li,<sup>†</sup> Kakasaheb Y. Nandiwale,<sup>§</sup> Chase A. Salazar,<sup>§</sup> Jason G. Mustakis,<sup>§</sup> William H. Green,<sup>†</sup> and Klavs F. Jensen<sup>†</sup>,\*

<sup>†</sup> Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02142, United StatesBu4ClO4Bu4ClO4

<sup>§</sup> Chemical Research & Development, Pfizer Worldwide Research and Development, Groton, CT 06340, United States

\* To whom correspondence should be addressed: kfjensen@mit.edu

# **Table of Contents**

Section S1. Materials and Methods	S3
Section S2. General Reactor Setup	S5
S2.1. Design and Assembly of Electrochemistry Batch Reactor	S5
S2.2. Reactivity Screening Experiments	S10
Section S3. Semantic Literature Analysis	S20
S3.1. Dataset Curation	S20
S3.2. Prompt Engineering	S27
S3.3. Evaluation Metrics and Results Analysis	S30
Section S4. Reactivity and Selectivity Prediction	\$35
S4.1. Reactivity Model	\$35
S4.2. Selectivity Model	\$37
S4.3. Virtual Screening	S38
Section S5. LLM-Assisted Code Generation	
Section S6. Optimization of Synthesis Conditions	S68
Section S7. NMR Data	S102
References	S110

# Section S1. Materials and Methods

### Chemicals

Acetonitrile (ACN), 1,3,5-trimethoxybenzene, and pyridine (anhydrous, purity  $\geq$  99.8%) were purchased from Sigma-Aldrich. Valerophenone (purity  $\geq$  98%), pregnenolone acetate (purity  $\geq$ 99%), norambreinolide (purity  $\geq$  97%), hexafluoro-2-propanol (HFIP, purity  $\geq$  99%) 2,2,6,6teramethyl-1-piperidinyloxy (TEMPO, purity  $\geq$  97%), quinuclidine (QD, purity  $\geq$  97%), tetrabutylammonium perchlorate (Bu<sub>4</sub>CNlO<sub>4</sub>, purity  $\geq$ 99%), tetrabutylammonium tetrafluoroborate (Et<sub>4</sub>NBF<sub>4</sub>, purity  $\geq$  97%), and tetrabutylammonium hexafluorophosphate (Bu<sub>4</sub>NPF<sub>6</sub>, purity  $\geq$  98%) were purchased from Aaron Chemicals. Theaspirane (purity  $\geq$  96%), *N*hydroxyphthalimide (NHPI,  $\geq$ 4,5,6,7-tetrachloro-2-hydroxyisoindoline-1,3-dione 98%), (TCNHPI, purity  $\geq$  97%), and 1,4-diazabicyclo(2.2.2)octane (DABCO, purity  $\geq$  98%) were purchased from Ambeed.  $\alpha$ -Pinene (purity  $\geq$  98%), ambroxide (purity  $\geq$  99%), alpha-ionone (purity lithium perchlorate anhydrous (LiClO<sub>4</sub>, purity  $\geq$  99%), and lithium > 95%), trifluoromethanesulfonate (LiOTf, purity  $\geq$  98%) were purchased from AK Scientific. 1,4dioxaspiro[4.5]dec-6-ene (purity  $\geq$  95%) was purchased from ChemScence. Valencene (purity  $\geq$ 95%) was purchased from Aurum Pharmatech LLC. Deuterated solvents were obtained from Cambridge Isotope Laboratories. The commercially available reagents were used without purification.

Large language models (LLMs) and data availability

In this research, several large language models were utilized, including GPT-3.5 and GPT-4 (developed and maintained by OpenAI), Claude 3 (developed and maintained by Anthropic), and the open-sourced Llama 3 (developed by Meta). These models fall under the category of autoregressive language models and are based on the transformer architecture.<sup>1</sup> Specifically, the GPT models (*gpt-3.5-turbo*, *gpt-4-0613*, *gpt-4-turbo*, *gpt-4o*, and *gpt-4o-mini*) and OpenAI o1 models (*o1-preview* and *o1-mini*) were accessed via application programming interface (API) from OpenAI, the Claude models (*claude-3.5-sonnet*, *claude-3-sonnet*, and *claude-3-opus*) via API from Anthropic, and the Llama 3 model (*llama-3.1-405b*, *llama-3-70b*) via API on Replicate.

For function-calling tasks, the backbone models *gpt-4o-2024-05-13* and *claude-3-5-sonnet-20240620* were employed, whereas the Llama model was not tested for this purpose. For semantic literature analysis, the *gpt-4o-2024-05-13* backbone model was used. We note that the GPT-40 model can also be accessed via the web user interface (WUI) chatbot ChatGPT (powered by GPT-40, version dated May 13, 2024), as the underlying base model is the same. Further documentation on the function calling, also known as tool-use, feature of those LLMs can be found at:

- https://platform.openai.com/docs/guides/function-calling

- https://docs.anthropic.com/en/docs/build-with-claude/tool-use

Detailed information on literature data mining is available in Section S3 of the Supporting Information. Unless specified, all code generated in this document was produced by LLMs through human prompting, and no substantial coding experience was required. Conversations between human users and LLMs for literature mining, code generation, and synthesis planning are publicly accessible in the Spreadsheets SF3 and SF4 of the Supporting information. These spreadsheets include both the human input prompts and the LLM output answers in JSON format.

Additional source code can be found in the Github repository (https://github.com/zach-zhiling-zheng/EChem-Explorations).

#### Nuclear magnetic resonance (NMR) spectroscopy

<sup>1</sup>H and <sup>13</sup>C NMR spectra were acquired on Bruker NEO-500 MHz spectrometers at Department of Chemistry Instrumentation Facility, Massachusetts Institute of Technology. Data were reported as chemical shifts in ppm relative to CDCl<sub>3</sub> (7.26 ppm) for <sup>1</sup>H and CDCl<sub>3</sub> (77.2 ppm) for <sup>13</sup>C. Chemical shifts are reported in ppm, and multiplicities are indicated by s (singlet), d (doublet), t (triplet), q (quartet), p (pentet), h (hextet), m (multiplet) and br (broad). Coupling constants, *J*, are reported in Hertz, and integration is provided and assignments are indicated.

# Section S2. General Reactor Setup

## S2.1. Design and Assembly of Electrochemistry Batch Reactor

The electrochemistry reactor was prepared in several steps. First, the anode and cathode connectors, made from Al 6061, were precisely cut to the specified dimensions and attached to the reactor top. This reactor top plate, crafted from HDPE, was then placed over the alignment plate, ensuring the connectors were properly aligned. The alignment plate, also made of HDPE, was essential for accurately positioning the electrodes within the reactor. Various electrodes, including nickel, graphite, and RVC, were custom-cut to fit the reactor's design. The reactor has the capacity to hold 48 anodes and 48 cathodes. These electrodes were then inserted into the 4mL vials, which were located within the vial locator, a component also fabricated from HDPE. Finally, a gasket was placed between the reactor top and the vial locator to ensure a secure and leak-proof assembly.

Description	Dimensions	Vendor	Part No.	Notes
Anode connector	69.13mm x 108.08mm x 1mm	Xometry	-	Al 6061
Reactor top	77mm x 111mm x 3.18mm	Protolabs	-	HDPE
Cathode connector	69.13mm x 105.46mm x 1mm	Xometry	-	Al 6061
Alignment plate	77mm x 111mm x 12mm	Protolabs	-	HDPE
Nickel electrode	1mm x 60mm	Sigma Aldrich	GF30807097	Custom cut
Graphite electrode	3.175mm x 60mm	McMaster-Carr	9121K86	Custom cut
RVC electrode	6.35mm x 152.4mm	McMaster-Carr	3947K141	Custom cut
Gasket	77mm x 111mm x 0.79mm	McMaster-Carr	5787T78	Custom cut
Vial locator	77mm x 111mm x 46mm	Protolabs	-	HDPE





Figure S1. 3D model and assembled photograph of the electrochemistry batch reactor.



Figure S2. Illustration of the reactor (without reactor top) and electrode connectors.



**Figure S3.** Drawings of the reactor top. The anode and cathode connectors can be embedded in the grooves of the plate. All distances are in millimeters.



Figure S4. Drawings of the anode connector. All distances are in millimeters.



Figure S5. Drawings of the cathode connector. All distances are in millimeters.



Figure S6. Drawings of the alignment plate. All distances are in millimeters.



Figure S7. Drawings of the vial locator. All distances are in millimeters.



**Figure S8.** Photographs of using a liquid handler to prepare 24 reaction mixtures in 4 ml glass vials simultaneously. (A) The side view shows the liquid handler adding solvent to multiple vials. (B) The top view shows the arrangement and filling of the vials.



Figure S9. Photographs of the assembled electrochemistry batch reactor. (A) Top view of the reactor box with all 96 electrodes inserted. (B) Connecting the reactor to the power supply while placed on top of a stir plate. (C) Side view showing the insertion of electrodes into the vials, demonstrating the alignment

#### **S2.2. Reactivity Screening Experiments**

To obtain the reactivity screening results, we adopted a strategy similar to that reported in the previous literature<sup>2</sup>, but instead of a droplet system, we employed the batch reactor. As described in the previous section, the batch reactor contained 24 glass vials, each has a maximum reaction volume of 4 mL. Therefore, we set 4mL to be the reaction volume for all 336 screening experiments. A stock solution was prepared containing 50 mM of the electrocatalyst N-hydroxyphthalimide (NHPI), 50 mM of the electrolyte tetraethylammonium tetrafluoroborate (Et<sub>4</sub>NBF<sub>4</sub>), and 50 mM pyridine. This stock solution was made by dissolving 5 mmol of each compound in 100 mL of acetonitrile-d<sub>3</sub> and stirring the mixture for 10 minutes, resulting in a light yellow, clear solution. The choice of NHPI as a mediator (electrocatalyst) was based on an analysis of a literature textmining dataset reported in Section S3, which identified NHPI as the most commonly used electrocatalyst for similar types of reactions. Following the preparation of the stock solution, 0.2 mmol of the substrate, listed in Table S2, was dissolved in 4 mL of the stock deuterium acetonitrile solution to achieve a concentration of 50 mM, and this ensured a 1:1:1 ratio between the substrate, mediator, and electrolyte. We note that acetonitrile can be used in replace of deuterium acetonitrile for the screening experiments, the reason we decided to use deuterium solvent is to accelerate the later analysis process. In addition, each vial was equipped with a PTFE magnetic stir bar (10 mm/0.39 inch) to facilitate mixing. We note that the substrates for the reactivity screening in this study were selected randomly from the laboratory inventory to maximize chemical diversity, and all chemicals used were commercially available. For the electrodes, both graphite and reticulated vitreous carbon (RVC) electrodes were tested for each substrate. Key reaction parameters, including the distance between the 4 electrodes in a vial (3.6 mm and 8.13 mm, respectively), stirring speed (600 rpm), reaction time (12 hours), and applied voltage (3.5 V), were kept constant across all reactions. It is important to note that an advantage of the use of a batch reactor in this study is that it minimized human error by providing a standardized setup, ensuring consistent control of all variables, and enhancing the efficiency of the screening process. Additionally, the batch reactor design reduced the number of wires and power sources needed for parallel reactions and simplified the handling of a large number of reactions on a robotic platform.

The main objective of the screening experiment was to assign a reactivity label to each compound for  $C(sp^3)$ -H oxidation to alcohol or ketone. Therefore, the primary focus was on whether any related product was formed as a result of C-H oxidation, regardless of the yield. To this end, proton and carbon NMR were chosen as the methods to determine if the expected oxidation occurred. Immediately after the reaction completed, the reactor top and the alignment plate of the reactor were removed, and 650 µL of the solution was taken directly for NMR analysis. Each time, 48 NMR tubes were placed on the autosampler to collect both proton and carbon NMR spectra for each sample before and after electrolysis. It should be notes that, in many cases, multiple products formed, and the identity of the products was unknown. Overlapping peaks in the proton NMR spectra and the inability of carbon NMR to provide indicative information, such as the appearance of carbonyl groups, complicated yield quantification. However, by manually comparing the proton and carbon NMR spectra of the reaction mixtures before and after electrolysis, it was possible to determine whether new species containing hydroxyl or carbonyl groups had formed as a result of C-H oxidation. Due to the nature of this analysis, the output was recorded as a Boolean variable either conversion (1) or no conversion (-1). This binary approach was adequate for the target modeling task designed to accommodate this experimental protocol, and in fact negative data points are more important as they are rarely reported in literature. Preliminary results are presented in Table S2, and the data is also available in Spreadsheet SF2 in the Supporting Information.



**Figure S10.** Photographs of electrochemical batch reactor setup for reactivity screening experiments. (A) Batch reactor: Top and alignment plate with connectors and electrodes (left); vials and vial locator (right). (B) Connecting wires to the reactor to supply electricity and turning on the magnetic stir plate. (C) Reaction mixtures before the electrolysis. (D) Reaction mixtures after the electrolysis. A total of 24 reactions were run every 12 hours, taking approximately two weeks to complete the screening of the substrates mentioned in the table below.

CAS Number	SMILES String	Reactivity
84-11-7	C1=CC=C2C(=C1)C3=CC=C3C(=O)C2=O	-1
94-09-7	CCOC(=O)C1=CC=C(C=C1)N	-1
98-06-6	CC(C)(C)C1=CC=CC=C1	-1
3074-03-1	C1=CC=C2C=C3C(=CC2=C1)C4=CC=CC=C4C3=O	-1
3988-03-2	C1=CC(=CC=C1C(=O)C2=CC=C(C=C2)Br)Br	-1
4452-06-6	C=CC(=O)C1=CC2=CC=CC=C2C=C1	-1
5118-06-9	COC(=0)C1=C(C=CS1)O	-1
100077-38-1	C1=C2C3=CC(=C(C=C3C4=CC(=C(C=C4C2=CC(=C1S)S)S)S)S)S	-1
100-09-4	COC1=CC=C(C=C1)C(=O)O	-1
100-47-0	C1=CC=C(C=C1)C#N	-1
100-55-0	C1=CC(=CN=C1)CO	1
100-68-5	CSC1=CC=CC=C1	-1
10075-50-0	C1=CC2=C(C=CN2)C=C1Br	-1
1013-88-3	C1=CC=C(C=C1)C(=N)C2=CC=C2	-1
1018-79-7	C1=CC=C2C(=C1)C(=C(C(=C2O)C#N)C#N)O	-1
10275-58-8	CC(C)(C)C1=CC2=C(C=C1)C=CC(=C2)C(C)(C)C	-1
10310-21-1	C1=NC2=C(N1)C(=NC(=N2)N)Cl	-1
10345-87-6	C1CC(=CC(=O)C1)C2=CC=CC=C2	-1
103-50-4	C1=CC=C(C=C1)COCC2=CC=C2	1
1035-52-5	C1CCC(CC1)N(C2CCCC2)C(=O)C3=CC=CC=C3	-1
104054-27-5	CCC1(CC2=CC=CC=C2C1)C3=CN=CN3	-1

Table S2. Boolean outcomes of initial screening results.

104-94-9	COC1=CC=C(C=C1)N	-1
106-40-1	C1=CC(=CC=C1N)Br	-1
106-47-8	C1=CC(=CC=C1N)Cl	-1
106-49-0	CC1=CC=C(C=C1)N	1
106-51-4	C1=CC(=O)C=CC1=O	-1
1090-13-7	C1=CC=C2C=C3C(=CC2=C1)C(=O)C4=CC=CC=C4C3=O	-1
1092390-02- 7	B1(OC(C(O1)(C)C)(C)C)C2=CC=C(C=C2)C3=CC4=CC=C4C=C3	-1
1121-76-2	C1=C[N+](=CC=C1CI)[O-]	-1
1122-54-9	CC(=O)C1=CC=NC=C1	-1
1122-58-3	CN(C)C1=CC=NC=C1	-1
1122-62-9	CC(=O)C1=CC=CC=N1	-1
116-63-2	C1=CC=C2C(=C1)C(=CC(=C2N)O)S(=O)(=O)O	-1
116-82-5	C1=CC=C2C(=C1)C(=O)C3=C(C2=O)C(=C(C=C3O)Br)N	-1
116-85-8	C1=CC=C2C(=C1)C(=O)C3=C(C=CC(=C3C2=O)O)N	-1
117-12-4	C1=CC2=C(C(=C1)O)C(=O)C3=C(C2=O)C(=CC=C3)O	-1
117-77-1	C1=CC=C2C(=C1)C(=O)C3=CC(=C(C=C3C2=O)O)N	-1
117-78-2	C1=CC=C2C(=C1)C(=O)C3=C(C2=O)C=C(C=C3)C(=O)O	-1
118129-60-5	C1=CC2=C3C(=CC(=C4C3=C1C5=C(C=C6C7=C(C=CC4=C57)C(=O)OC6=O)Br)Br)C(=O)OC2=O	-1
1198-14-7	C1=CC2=C(C=CC(=C2N=C1)O)Br	-1
121-69-7	CN(C)C1=CC=CC	-1
1221722-10- 6	CC1=C(C=CC=C1C(=O)O)C2=CC=C(C=C2)OC(F)(F)F	1
1222-05-5	CC1COCC2=CC3=C(C=C12)C(C(C3(C)C)C)(C)C	-1
123-31-9	C1=CC(=CC=C1O)O	-1
127926-65-2	C1=CC=C2C(=C1)C3=CC=CC=C3C2(C4=CC(=C(C=C4)N)F)C5=CC(=C(C=C5)N)F	-1
128-64-3	C1=CC=C2C(=C1)C3=C4C(=CC=C5C4=C(C=C3)C6=CC=C7C8=C(C=CC5=C68)C9=CC=CC=C9C7=O)C2=O	-1
128-67-6	C1=CC=C2C(=C1)C(=O)C3=C(C2=O)C(=C(C=C3)C(=O)O)[N+](=O)[O-]	-1
128-80-3	CC1=CC=C(C=C1)NC2=C3C(=C(C=C2)NC4=CC=C(C=C4)C)C(=O)C5=CC=CC=C5C3=O	-1
128-93-8	CNC1=C2C(=C(C=C1)Br)C(=O)C3=CC=C3C2=O	-1
128-95-0	C1=CC=C2C(=C1)C(=O)C3=C(C=CC(=C3C2=O)N)N	-1
129112-25-0	C1=CC=C(C(=C1)OS(=O)(=O)C(F)(F)F)Br	-1
129-39-5	C1=CC2=C(C(=C1)[N+](=O)[O-])C(=O)C3=C(C2=O)C=CC=C3[N+](=O)[O-]	-1
129-44-2	C1=CC2=C(C(=C1)N)C(=O)C3=C(C2=O)C(=CC=C3)N	-1
129-56-6	C1=CC=C2C(=C1)C3=NNC4=CC=CC(=C43)C2=O	-1
130-15-4	C1=CC=C2C(=O)C=CC(=O)C2=C1	-1
130-20-1	C1=CC=C2C(=C1)C(=O)C3=CC(=C4C(=C3C2=O)NC5=C(C=C6C(=C5N4)C(=O)C7=CC=CC=C7C6=O)CI)CI	-1
130-26-7	C1=CC2=C(C(=C(C=C2CI)I)O)N=C1	-1
130-95-0	COC1=CC2=C(C=CN=C2C=C1)C(C3CC4CCN3CC4C=C)O	1
131-14-6	C1=CC2=C(C=C1N)C(=O)C3=C(C2=O)C=C(C=C3)N	-1
131-56-6	C1=CC=C(C=C1)C(=O)C2=C(C=C2)O)O	-1
131-57-7	COC1=CC(=C(C=C1)C(=O)C2=CC=CC=C2)O	-1
13361-64-3	C[Si](C)(C)CC#C	1
134-81-6	C1=CC=C(C=C1)C(=O)C(=O)C2=CC=C2	-1

13534-97-9	C1=CC(=NC=C1N)Br	-1
135-88-6	C1=CC=C(C=C1)NC2=CC3=CC=C3C=C2	-1
13629-22-6	C1=CC=C2C(=C1)C3=CC=C3C2=NN	-1
13734-36-6	CC(C)(C)OC(=O)N(C)CC(=O)O	-1
137-58-6	CCN(CC)CC(=O)NC1=C(C=CC=C1C)C	1
14289-45-3	C1=CC=C2C=C(C=CC2=C1)C(=O)O	-1
14348-75-5	C1=CC2=C(C=C1Br)C(=O)C3=C2C=CC(=C3)Br	-1
1446-61-3	CC(C)C1=CC2=C(C=C1)C3(CCCC(C3CC2)(C)CN)C	1
14548-46-0	C1=CC=C(C=C1)C(=O)C2=CC=NC=C2	-1
147-85-3	C1CC(NC1)C(=O)O	1
148893-10-1	CN(C)C(=[N+](C)C)N1C2=C(N=CC=C2)[N+](=N1)[O-].F[P-](F)(F)(F)F	-1
14918-69-5	C1=CC(=O)C2=C(C1=O)C(=C(C(=C2O)CI)CI)O	-1
14923-84-3	C1=CC2=C(C=CC3=C2C4=C1C=CC(=C4C=C3)N)N	-1
1503-49-7	C1=CC=C(C=C1)C(=O)C2=CC=C(C=C2)C#N	-1
15499-84-0	C1=CC=C2C(=C1)C3=CC=CC=C3C2(C4=CC=C(C=C4)N)C5=CC=C(C=C5)N	-1
156-38-7	C1=CC(=CC=C1CC(=O)O)O	-1
157864-32 <i>-</i> 9	C1=CC=C(C=C1)CNC2=NC(=NC3=CC=C32)Cl	1
1579-40-4	CC1=CC=C(C=C1)OC2=CC=C(C=C2)C	-1
1592-95-6	C1=CC=C2C(=C1)C3=C(N2)C=CC(=C3)Br	-1
160591-91-3	B(C1=C(C=C(C=C1)CI)F)(O)O	-1
1612793-07- 3	C=CC1=CC2=C(C=C1)C(=C(C#N)C#N)C3=CC=CC=C3C2=C(C#N)C#N	-1
16294-75-0	C1=CC=C2C(=C1)C3=CC=C4C5=C(C=CC(=C35)C2=O)C6=CC=C6S4	-1
164295-94-7	C1=CC=C2C=C(C=CC2=C1)C3=NNC(=C3)C(=O)O	-1
167869-21-8	COC1=CC=CC(=C1N)C2=CC(=O)C3=CC=C3O2	-1
1693-28-3	C1=CC=C2C(=C1)C(=O)C3=C(S2)C=CC(=C3)C(F)(F)F	-1
171364-82-2	B1(OC(C(O1)(C)C)(C)C)C2=CC=C(C=C2)C#N	-1
17217-57-1	COC1=CC(=NC=C1)C2=NC=CC(=C2)OC	-1
1729-99-3	C1=CC2=C(C(=C1)C(=O)O)C(=CC=C2)Br	-1
17418-58-5	C1=CC=C(C=C1)OC2=C(C3=C(C(=C2)O)C(=O)C4=CC=CC=C4C3=O)N	-1
17480-69-2	CC(C1=CC=CC=C1)NCC2=CC=C2	1
1800-42-6	C1=CC=C2C=C(C=CC2=C1)C(=O)C(F)(F)F	-1
18162-48-6	CC(C)(C)[Si](C)(C)Cl	-1
18173-64-3	CC(C)(C)[Si](C)(C)O	-1
1827-27-6	C1=CC(=NC=C1N)F	-1
18801-00-8	CC(C)(C)C1=CC2=CC3=CC=C3C=C2C=C1	-1
188425-85-6	C1=CC=C(C(=C1)C2=CC=C(C=C2)Cl)NC(=O)C3=C(N=CC=C3)Cl	-1
188-94-3	C1=CC2=C3C=CC4=C5C=CC6=C7C=CC=CC7=C8C6=C5C(=C9C4=C3C(=C2C=C1)C=C9)C=C8	-1
191-07-1	C1=CC2=C3C4=C1C=CC5=C4C6=C(C=C5)C=CC7=C6C3=C(C=C2)C=C7	-1
191-24-2	C1=CC2=C3C(=C1)C4=CC=CC5=C4C6=C(C=C5)C=CC(=C36)C=C2	-1
192198-85- <del>9</del>	C1=CC=C(C=C1)N2C3=CC=CC=C3N=C2C4=CC(=CC(=C4)C5=NC6=CC=CC=C6N5C7=CC=CC=C7)C8=NC9=CC=CC=C9 N8C1=CC=CC=C1	-1
1948-33-0	CC(C)(C)C1=C(C=CC(=C1)O)O	-1
19694-02-1	C1=CC2=C3C(=C1)C=CC4=C(C=CC(=C43)C=C2)C(=O)O	-1

2018-61-3	CC(=O)NC(CC1=CC=CC=C1)C(=O)O	1
206-44-0	C1=CC=C2C(=C1)C3=CC=CC4=C3C2=CC=C4	-1
207-08-9	C1=CC=C2C=C3C4=CC=CC5=C4C(=CC=C5)C3=CC2=C1	-1
2074732-95- 7	C1CC2(C1C2)S(=O)(=O)C3=CC=C(C=C3)C(F)(F)F	-1
, 21022-17-3	C1=CC=C(C=C1)C2=CC=C2NC(=O)C(=O)NC3=CC=CC=C3C4=CC=CC=C4	-1
2103-99-3	C1=CC(=CC=C1C2=CSC(=N2)N)Cl	-1
21145-77-7	CC1CC(C2=C(C1(C)C)C=C(C(=C2)C(=O)C)C)(C)C	1
21331-43-1	C1=CC=C2C=C(C=CC2=C1)C3=CSC(=N3)N	-1
214360-44-8	B1(OCC(CO1)(C)C)C2=CC=C(C=C2)C#N	-1
214360-73-3	B1(OC(C(O1)(C)C)(C)C)C2=CC=C(C=C2)N	-1
215-58-7	C1=CC=C2C=C3C4=CC=CC=C4C5=CC=C5C3=CC2=C1	-1
2157-52-0	C1=CC=C2C(=C1)C3=CC=CC=C3C2=NO	-1
217-68-5	C1=CC=C2C(=C1)C3=CC=CC=C3C4=NC=CN=C24	-1
218-01-9	C1=CC=C2C(=C1)C=CC3=C2C=CC4=CC=CC43	-1
2243590-42- 1	C1=C(SC2=C1C3=C(C=C(S3)C=O)C4=C2C=C(S4)C=O)C=O	-1
23593-75-1	C1=CC=C(C=C1)C(C2=CC=C2)(C3=CC=CC=C3Cl)N4C=CN=C4	-1
236389-21-2	B(C1=CC2=C(C=C1)C3=CC=CC=C3C24C5=CC=CC=C5C6=CC=C46)(0)O	-1
23676-08-6	CCOC1=CC=C(C=C1)C(=O)OC	-1
23746-81-8	C1=CC=C2C=C(C=CC2=C1)C3=CC4=CC=CC=C4N3	-1
239-01-0	C1=CC=C2C(=C1)C=CC3=C2NC4=CC=CC=C34	-1
23912-79-0	C1=CC=C2C(=C1)C(=O)C3=CC4=C(C=C3C2=O)C(=O)C5=CC=C5C4=O	-1
2420-87-3	C1=CC2=C(C=C1C3=CC4=C(C=C3)C(=O)OC4=O)C(=O)OC2=O	-1
2446-83-5	CC(C)OC(=O)N=NC(=O)OC(C)C	-1
2475-44-7	CNC1=C2C(=C(C=C1)NC)C(=O)C3=CC=C3C2=O	-1
2475-45-8	C1=CC(=C2C(=C1N)C(=O)C3=C(C=CC(=C3C2=O)N)N)N	-1
2627-86-3	CC(C1=CC=CC=C1)N	-1
2648079-79-	C1=CC=C2C(=C1)SC3=CC=CC=C3[S+]2C(F)(F)F.C(F)(F)S(=O)(=O)[O-]	-1
2675-89-0	CN(C)C(=O)CCI	-1
2714-87-6	C1=CC=C(C=C1)C2=CC=C(C=C2)C(=O)F	-1
2719-27-9	C1CCC(CC1)C(=O)Cl	-1
27485-15-0	C1=CC=C2C(=C1)C(=O)C3=CC(=C(C=C3C2=O)C(=O)O)C(=O)O	-1
2834-92-6	C1=CC=C2C(=C1)C=CC(=C2N)O	-1
2872-48-2	COC1=C(C2=C(C(=C1)N)C(=O)C3=CC=C3C2=O)N	-1
28736-42-7	C1=CC=C2C(=C1)C(=O)C3=C(C=CC(=C3C2=O)F)F	-1
2958-36-3	C1=CC=C(C(=C1)C(=O)C2=C(C=CC(=C2)CI)N)CI	-1
299914-63- <del>9</del>	C1=CC(=CC=C1C(=C(C2=CC=C(C=C2)I)C3=CC=C(C=C3)I)C4=CC=C(C=C4)I)I	-1
3029-32-1	C1=CC=C2C=C3C(=CC2=C1)C(=O)C4=CC5=CC=C5C=C4C3=O	-1
30379-58-9	C1=CC=C(C=C1)COC(=O)CO	-1
3096-56-8	C1=CC=C2C(=C1)C3=C(C2=O)C=C(C=C3)Br	-1
3147-58-8	C1=CC=C2C(=C1)C=C(C(=C2O)C(=O)O)O	-1
31519-22-9	C1=CC=C2C(=C1)C(=CC(=C2O)C(=O)O)O	-1
31574-87-5	C1=CC2=C(C=C1Br)C3=C(S2)C=CC(=C3)Br	-1

318-98-9 CC(C)NCC(COC1=CC=CC2=CC=C21)O.Cl   320-51-4 C1=CC(=C(C=C1N)C(F)(F)F)Cl   321-60-8 C1=CC=C(C=C1)C2=CC=C2F   3236-71-3 C1=CC=C2C(=C1)C3=CC=CC=C3C2(C4=CC=C(C=C4)O)C5=CC=C(C=C5)O   32795-84-9 C1=CC=C2C(=C1)C=CC3=C(C4=CC=C4C=C32)Br   333432-28-3 B(C1=CC2=C(C=C1)C3=CC=CC=C3C2(C)C)(O)O   3280-34-5 C1=CC(=C1C)O(C3=C(C=C1)C)C)C(C=C(C=C2)C)(C)	-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
320-51-4 C1=CC(=C(C=C1N)C(F)(F)F)CI   321-60-8 C1=CC=C(C=C1)C2=CC=C2F   3236-71-3 C1=CC=C2C(=C1)C3=CC=C3C2(C4=CC=C(C=C4)O)C5=CC=C(C=C5)O   32795-84-9 C1=CC=C2C(=C1)C=CC3=C(C4=CC=C4=C32)Br   333432-28-3 B(C1=CC2=C(C=C1)C3=CC=C3C2(C)O)O   3280-34-5 C1=CC=C2C(=C1)C3=CC=C3C2(C)O)O	-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
321-60-8 C1=CC=C(C=C1)C2=CC=C2F   3236-71-3 C1=CC=C2C(=C1)C3=CC=C3C2(C4=CC=C4)O)C5=CC=C(C=C5)O   32795-84-9 C1=CC=C2C(=C1)C=CC3=C(C4=CC=C4C=C32)Br   333432-28-3 B(C1=CC2=C(C=C1)C3=CC=CC=C3C2(C)C)(O)O   2380-34-5 C1=CC(=C1C)O(C3=CC=C2C)C)(O)O	-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
3236-71-3 C1=CC=C2C(=C1)C3=CC=C3C2(C4=CC=C(C=C4)O)C5=CC=C(C=C5)O   32795-84-9 C1=CC=C2C(=C1)C=CC3=C(C4=CC=C4C=C32)Br   333432-28-3 B(C1=CC2=C(C=C1)C3=CC=CC=C3C2(C)C)(O)O   2380-34-5 C1=CC(=C1C)O(C=C1C)O(C)=C(C=C2)C)(C)	-1 -1 -1 -1 -1 -1 -1 -1
32795-84-9 C1=CC=C2C(=C1)C=CC3=C(C4=CC=C4C=C32)Br   333432-28-3 B(C1=CC2=C(C=C1)C3=CC=CC=C3C2(C)C)(O)O   2380-34-5 C1=CC(=C1C)O(C2=C(C=C1)C)C(C=C)C)(O)O	-1 -1 -1 -1 -1 -1 -1
B(C1=CC2=C(C=C1)C3=CC=C3C2(C)C)(O)O	-1 -1 -1 -1 -1
	-1 -1 -1 -1
5380-54-5 CI-CC(-C(C-CIC))0/0CZ-C(C-CZ)CI/CI	-1 -1 -1 -1
3405-77-4 CC1=CC(=NO1)C(=O)O	-1 -1 -1
34316-15-9 C[N+]1=C2C(=C3C=CC(=C(C3=C1)OC)OC)C=CC4=CC5=C(C=C42)OCO5	-1 -1
34941-90-7 C1=CN=C(C=C1F)F	-1
34993-56-1 C#CC1=C2C=CC3=CC=CC4=C3C2=C(C=C4)C=C1	
35438-63-2 C1=CC2=C3C(=C1)C4=C5C(=C(C=C4)C=O)C=CC=C5C3=CC=C2	-1
35825-57-1 CC1COC2=C1C(=O)C(=O)C3=C2C=CC4=C3CCCC4(C)C	1
360-03-2 C1=CC=C(C=C1)C(C(=O)O)(F)F	-1
360-97-4 C1=NC(=C(N1)C(=O)N)N	-1
3641-13-2 C1(=NC(=NN1)N)C(=O)O	-1
367-21-5 C1=CC(=C(C=C1N)CI)F	-1
371-40-4 C1=CC(=CC=C1N)F	-1
3737-95-9 C1=CC=C2C(=C1)C=C(C(=C2N=NC3=C(C=C(C4=CC=CC43)S(=O)(=O)O)O)O)C(=O)O	-1
3905-64-4 CC(C)(C)C1=CC2=C(C=C1)C=C(C=C2)C(C)(C)C	-1
3910-35-8 CC1(CC(C2=CC=CC1)(C)C3=CC=C3)C	-1
3920-79-4 C1=CC=C(C=C1)NC2=CC3=CC=C3C4=CC=C42	-1
39215-21-9 C1=CC=C2C(=C1)C=C(C(=C2C3=C(C(=CC4=CC=C43)O)O)O)O	-1
3934-20-1 C1=CN=C(N=C1Cl)Cl	-1
3963-95-9 CN(C)C1C2C(C3C(=C)C4=C(C(=CC=C4)O)C(=C3C(=O)C2(C(=C(C1=O)C(=O)N)O)O)O)O.Cl	-1
39674-97-0 CC1(C2=C(C3=CC=C3C(=O)C2=O)NC1=O)C	-1
402-43-7 C1=CC(=CC=C1C(F)(F)F)Br	-1
40291-26-7 C1CC(=O)N(C1C(=O)ON2C(=O)CCC2=O)C(=O)OCC3=CC=C3	-1
42019-78-3 C1=CC(=CC=C1C(=O)C2=CC=C(C=C2)CI)O	-1
42823-46-1 C1=CC(=CC=C1C(=O)O)N=C(N)N.Cl	-1
42924-53-8 CC(=O)CCC1=CC2=C(C=C1)C=C(C=C2)OC	-1
434-45-7 C1=CC=C(C=C1)C(=O)C(F)(F)F	-1
434-64-0 C1(=C(C(=C1F)F)F)F)F)C(F)(F)F	-1
434-85-5 C1=CC=C2C(=C1)C(=C3C4=CC=CC=C4C(=O)C5=CC=CC=C53)C6=CC=C6C2=O	-1
434-90-2 C1(=C(C(=C1F)F)F)F)C2=C(C(=C(C(=C2F)F)F)F)F)	-1
443-69-6 C1=CC2=C(C=C1F)C(=O)N2	-1
4487-59-6 C1=CC(=NC=C1[N+](=O)[O-])Br	-1
4548-45-2 C1=CC(=NC=C1[N+](=O)[O-])Cl	-1
455-14-1 C1=CC(=CC=C1C(F)(F)F)N	-1
456-24-6 C1=CC(=NC=C1[N+](=O)[O-])F	-1
4569-45-3 CC1(C2=CC=CC=C2C3=CC=C31)C	-1
4630-82-4 COC(=O)C1CCCCC1	-1
4674-50-4 CC1CC(=O)C=C2C1(CC(CC2)C(=C)C)C	-1

467-69-6	C1=CC=C2C(=C1)C3=CC=CC=C3C2(C(=O)O)O	-1
470-17-7	CC12CCCC(=C)C1CC3C(C2)OC(=O)C3=C	-1
475-38-7	C1=CC(=C2C(=O)C=CC(=O)C2=C1O)O	-1
475-71-8	C1=CC=C2C(=C1)C3=NC4=C5C6=C(C=C4)C(=O)C7=CC=CC=C7C6=NC8=C5C3=C(C2=O)C=C8	-1
476-60-8	C1=CC=C2C(=C1)C(=C3C(=CC=C(C3=C2O)O)O)O	-1
480-22-8	C1=CC2=CC3=C(C(=CC=C3)O)C(=C2C(=C1)O)O	-1
480-40-0	C1=CC=C(C=C1)C2=CC(=O)C3=C(C=C(C=C3O2)O)O	-1
480-41-1	C1C(OC2=CC(=CC(=C2C1=O)O)O)C3=CC=C(C=C3)O	-1
481-06-1	CC1C2CCC3(C=CC(=O)C(=C3C2OC1=O)C)C	-1
484-17-3	C1=CC=C2C(=C1)C=C(C3=CC=C23)O	-1
485-47-2	C1=CC=C2C(=C1)C(=O)C(C2=O)(O)O	-1
485-72-3	COC1=CC=C(C=C1)C2=COC3=C(C2=O)C=CC(=C3)O	-1
486-25-9	C1=CC=C2C(=C1)C3=CC=C3C2=O	-1
486-66-8	C1=CC(=CC=C1C2=C0C3=C(C2=O)C=CC(=C3)O)O	-1
4877-80-9	C1=C2C3=CC(=C(C=C3C4=CC(=C(C=C4C2=CC(=C1O)O)O)O)O)O)O	-1
491-67-8	C1=CC=C(C=C1)C2=CC(=O)C3=C(O2)C=C(C(=C3O)O)O	-1
492-27-3	C1=CC=C2C(=C1)C(=O)C=C(N2)C(=O)O	-1
50-28-2	CC12CCC3C(C1CCC2O)CCC4=C3C=CC(=C4)O	1
50-32-8	C1=CC=C2C3=C4C(=CC2=C1)C=CC5=C4C(=CC=C5)C=C3	-1
51726-83-1	C1=CC=C2C(=C1)C=CC3=C2NC=C(C3=O)C(=O)O	-1
525-82-6	C1=CC=C(C=C1)C2=CC(=O)C3=CC=CC=C3O2	-1
529-86-2	C1=CC=C2C(=C1)C=C3C=CC3=C2O	-1
530-62-1	C1=CN(C=N1)C(=O)N2C=CN=C2	-1
53-16-7	CC12CCC3C(C1CCC2=O)CCC4=C3C=CC(=C4)O	1
53226-42-9	C1CCC(C1)NC(=O)C2=CC=C2	-1
53348-04-2	C1=CC=C2C(=C1)C3=CC=C3C(=C2N)N	-1
53348-05-3	C1=CC2=C(C=C1Br)C3=C(C=CC(=C3)Br)C(=O)C2=O	-1
534-17-8	C(=O)([O-])[O-].[Cs+].[Cs+]	-1
5350-93-6	C1=CC(=NC=C1N)Cl	-1
538-92-2	CC(C)CC1=CC=C1	1
5417-63-0	C1=CC=C2C=C(C(=CC2=C1)N)O	-1
5437-45-6	C1=CC=C(C=C1)COC(=O)CBr	1
5451-40-1	C1=NC2=C(N1)C(=NC(=N2)CI)Cl	-1
5468-77-9	CN(C)C(=O)CBr	-1
547-91-1	C1=CC2=C(C(=C(C=C2S(=O)(=O)O)I)O)N=C1	-1
5521-31-3	CN1C(=0)C2=C3C(=CC=C4C3=C(C=C2)C5=C6C4=CC=C7C6=C(C=C5)C(=O)N(C7=O)C)C1=O	-1
556-90-1	C1=CC(=CC=C1C=C2C(=O)N=C(S2)N)O	-1
5625-67-2	C1CNC(=O)CN1	-1
56-45-1	C(C(C(=O)O)N)O	-1
56752-35-3	C1=CC2=C3C(=C1)C(=CC=C3C4=C5C2=CC=C(C5=CC=C4)Br)Br	-1
568-72-9	CC1=COC2=C1C(=O)C(=O)C3=C2C=CC4=C3CCCC4(C)C	-1
571-60-8	C1=CC=C2C(=C1)C(=CC=C2O)O	-1
572-83-8	C1=CC=C2C(=C1)C(=O)C3=C(C2=O)C=C(C=C3)Br	-1

574-09-4	CCOC(C1=CC=CC)C(=O)C2=CC=C2	-1
57486-69-8	COC(=O)CC1=CC=C1Br	1
57-63-6	CC12CCC3C(C1CCC2(C#C)O)CCC4=C3C=CC(=C4)O	1
58-14-0	CCC1=C(C(=NC(=N1)N)N)C2=CC=C(C=C2)Cl	-1
58-63-9	C1=NC2=C(C(=O)N1)N=CN2C3C(C(C(O3)CO)O)O	1
59-67-6	C1=CC(=CN=C1)C(=O)O	-1
59721-16-3	CN(C)C(=O)COC(=O)CC1=CC=C(C=C1)O	-1
602-25-5	C1=CC=C2C(=C1)C(=O)C3=C(C=CC(=C3C2=O)CI)CI	-1
605-32-3	C1=CC=C2C(=C1)C(=O)C3=C(C2=O)C=C(C=C3)O	-1
605-55-0	C1=CC=C2C(=C1)C=CC3=C2C=CC(=C3)O	-1
607-68-1	C1=CC=C2C(=C1)C(=NC(=N2)CI)CI	-1
608-31-1	C1=CC(=C(C(=C1)CI)N)CI	-1
610-49-1	C1=CC=C2C=C3C(=CC2=C1)C=CC=C3N	-1
612-78-2	C1=CC=C2C=C(C=CC2=C1)C3=CC4=CC=CC=C4C=C3	-1
613-13-8	C1=CC=C2C=C3C=C(C=CC3=CC2=C1)N	-1
6217-22-7	C1=CC2=C3C(=C1)C(=O)C(=O)C4=CC=CC(=C43)C=C2	-1
622-31-1	C1=CC=C(C=C1)C=NO	-1
6223-83-2	C1=CC=C2C(=C1)C3=C(C2=O)C=CC=C3C(=O)O	-1
622-62-8	CCOC1=CC=C(C=C1)O	-1
623-00-7	C1=CC(=CC=C1C#N)Br	-1
632-83-7	C1=CC=C2C(=C1)C(=O)C3=C(C2=O)C(=CC=C3)Br	-1
635-12-1	C1=CC=C2C=C3C(=O)C=CC(=O)C3=CC2=C1	-1
6364-19-8	C1=CC2=C3C(=C1)C(=CC=C3C4=C5C2=CC=C(C5=CC=C4)C(=O)O)C(=O)O	-1
6398-98-7	CCN(CC)CC1=C(C=CC(=C1)NC2=C3C=CC(=CC3=NC=C2)CI)O.O.O.CI.CI	-1
644-13-3	C1=CC=C(C=C1)C(=O)C2=CC3=CC=C3C=C2	-1
6485-55-8	CC1CNCC(O1)C	1
6485-97-8	C1=CC=C(C=C1)C2=CC3=C(C=C2)C(=O)C4=CC=CC=C4C3=O	-1
66-25-1	CCCCC=0	-1
6674-22-2	C1CCC2=NCCCN2CC1	-1
693-13-0	CC(C)N=C=NC(C)C	-1
708-06-5	C1=CC=C2C(=C1)C=CC(=C2C=O)O	-1
7087-68-5	CCN(C(C)C)C(C)C	1
71125-38-7	CC1=CN=C(S1)NC(=O)C2=C(C3=CC=C3S(=O)(=O)N2C)O	-1
7153-22-2	CCOC(=O)C1=CC=C(C=C1)C#N	-1
719-59-5	C1=CC=C(C=C1)C(=O)C2=C(C=CC(=C2)CI)N	-1
72-48-0	C1=CC=C2C(=C1)C(=O)C3=C(C2=O)C(=C(C=C3)O)O	-1
7254-19-5	C1=CC2=C(C=C1Br)C=C(N2)C(=O)O	-1
7323-62-8	CC(=O)NC1=C2C(=C(C=C1)O)C(=O)C3=CC=C3C2=O	-1
7427-09-0	CC1=C(C(=O)C2=CC=C2C1=O)N	-1
7473-98-5	CC(C)(C(=O)C1=CC=CC=C1)O	-1
7601-89-0	[O-]CI(=O)(=O)=O.[Na+]	-1
762-72-1	C[Si](C)(C)CC=C	1
7632-00-0	N(=O)[O-].[Na+]	-1

7649-92-5	C1=CC=C(C(=C1)C(=O)C2=CC=C(C=C2)F)C(=O)O	-1
765-30-0	C1CC1N	-1
76-83-5	C1=CC=C(C=C1)C(C2=CC=C2)(C3=CC=C3)Cl	-1
771-97-1	C1=CC=C2C=C(C(=CC2=C1)N)N	-1
773-76-2	C1=CC2=C(C(=C(C=C2CI)CI)O)N=C1	-1
773873-77-1	COC(=O)C1=CNC2=C1C=C(C=C2)Br	-1
807-19-2	C1=CC(=CC=C1C(=O)O)P(=O)(C2=CC=C(C=C2)C(=O)O)C3=CC=C(C=C3)C(=O)O	-1
808-57-1	COC1=C(C=C2C(=C1)C3=CC(=C(C=C3C4=CC(=C(C=C24)OC)OC)OC)OC)OC	-1
81-32-3	C1=CC(=C2C(=CC=C3C2=C1C4=C5C3=CC=C(C5=C(C=C4)C(=O)O)C(=O)C(=	-1
81-33-4	C1=CC2=C3C(=CC=C4C3=C1C5=C6C4=CC=C7C6=C(C=C5)C(=O)NC7=O)C(=O)NC2=O	-1
81-41-4	C1=CC=C2C(=C1)C(=O)C3=C(C(=C(C(=C3C2=O)N)C#N)C#N)N	-1
81-42-5	C1=CC=C2C(=C1)C(=O)C3=C(C2=O)C(=C(C(=C3N)CI)CI)N	-1
81-49-2	C1=CC=C2C(=C1)C(=O)C3=C(C2=O)C(=C(C=C3Br)Br)N	-1
81-54-9	C1=CC=C2C(=C1)C(=O)C3=C(C2=O)C(=C(C=C3O)O)O	-1
81-55-0	C1=CC(=C2C(=C1[N+](=O)[O-])C(=O)C3=C(C=CC(=C3C2=O)O)[N+](=O)[O-])O	-1
81-58-3	C1=CC(=C2C(=C1Cl)C(=O)C3=C(C=CC(=C3C2=O)Cl)Cl)Cl	-1
81-64-1	C1=CC=C2C(=C1)C(=O)C3=C(C=CC(=C3C2=O)O)O	-1
81-77-6	C1=CC=C2C(=C1)C(=O)C3=C(C2=O)C4=C(C=C3)NC5=C(N4)C=CC6=C5C(=O)C7=CC=C7C6=O	-1
81-96-9	C1=CC=C2C(=C1)C3=C4C(=C(C=C3)Br)C=CC=C4C2=O	-1
82-22-4	C1=CC=C2C(=C1)C(=O)C3=C(C2=O)C(=CC=C3)NC4=CC=CC5=C4C(=O)C6=CC=CC=C6C5=O	-1
82-34-8	C1=CC=C2C(=C1)C(=O)C3=C(C2=O)C(=CC=C3)[N+](=O)[O-]	-1
82-38-2	CNC1=CC=CC2=C1C(=O)C3=CC=C3C2=O	-1
82-43-9	C1=CC2=C(C(=C1)Cl)C(=O)C3=C(C2=O)C=CC=C3Cl	-1
82-45-1	C1=CC=C2C(=C1)C(=O)C3=C(C2=O)C(=CC=C3)N	-1
82-46-2	C1=CC2=C(C(=C1)Cl)C(=O)C3=C(C2=O)C(=CC=C3)Cl	-1
82717-96-2	CCOC(=O)C(CCC1=CC=CC1)NC(C)C(=O)O	-1
84405-44-7	C1=CC2=C(C=C1Br)C(=O)C(=O)C3=C2C=CC(=C3)Br	-1
84-47-9	CC(C)(C)C1=CC2=C(C=C1)C(=O)C3=CC=C3C2=O	-1
84-60-6	C1=CC2=C(C=C1O)C(=O)C3=C(C2=O)C=C(C=C3)O	-1
85721-33-1	C1CC1N2C=C(C(=O)C3=CC(=C(C=C32)N4CCNCC4)F)C(=O)O	-1
865-47-4	CC(C)(C)[O-].[K+]	-1
86636-92-2	CN(C)C1=CC2=C(C=C1)C=C(C=C2)C(=O)C=C	-1
870259-02 <i>-</i> 2	C#CC1=CC(=C2C=CC3=C(C=C(C4=C3C2=C1C=C4)C#C)C#C)C#C	-1
870774-29-1	B(C1=CC(=CC=C1)C2=CC3=CC=C3C=C2)(O)O	-1
872050-52-7	B(C1=CC=C(C=C1)C2=C3C=CC4=CC=CC5=C4C3=C(C=C5)C=C2)(O)O	-1
87-42-3	C1=NC2=C(N1)C(=NC=N2)Cl	-1
88-17-5	C1=CC=C(C(=C1)C(F)(F)F)N	-1
882-09-7	CC(C)(C(=O)O)OC1=CC=C(C=C1)Cl	-1
883835-33-4	C1=CC(=CC=C1C=O)C2=CC(=C(C=C2C3=CC=C(C=C3)C=O)C4=CC=C(C=C4)C=O)C5=CC=C(C=C5)C=O	-1
88426-33-9	CC(C)(C)C1CCC(CC1)CC2=C(C3=CC=C3C(=O)C2=O)O	1
90-15-3	C1=CC=C2C(=C1)C=CC=C2O	-1
90299-04-0	CC(C)(C1=CC2=CC=C2C=C1)N	-1
90-43-7	C1=CC=C(C=C1)C2=CC=C2O	-1

90-47-1	C1=CC=C2C(=C1)C(=O)C3=CC=CC=C3O2	-1
92149-07-0	COC1=C2C=CC3=C(C=CN=C3C2=NC=C1)OC	-1
92-44-4	C1=CC=C2C=C(C(=CC2=C1)O)O	-1
92-66-0	C1=CC=C(C=C1)C2=CC=C(C=C2)Br	-1
927-58-2	C(CC(=O)CI)CBr	-1
93655-34-6	COC(=O)C1=CC=CC2=CC3=C(C=C21)C(=CC=C3)C(=O)OC	-1
93-92-5	CC(C1=CC=CC=C1)OC(=O)C	-1
939-97-9	CC(C)(C)C1=CC=C(C=C1)C=O	-1
947-73-9	C1=CC=C2C(=C1)C=C(C3=CC=C23)N	-1
95233-18-4	C1CC(CCC1C2=CC=C(C=C2)Cl)C3=C(C4=CC=CC=C4C(=O)C3=O)O	-1
95-51-2	C1=CC=C(C(=C1)N)Cl	-1
957-68-6	CC(=O)OCC1=C(N2C(C(C2=O)N)SC1)C(=O)O	-1
959085-68-8	C1CC(C1)OC(=O)C2=CC=C(C=C2)C(F)(F)F	-1
98-16-8	C1=CC(=CC(=C1)N)C(F)(F)F	-1
99548-54-6	CC1=C(C=CC=C1Br)C(=O)OC	-1
99614-02-5	CC1=NC=CN1CC2CCC3=C(C2=O)C4=CC=CC=C4N3C	1
99-90-1	CC(=O)C1=CC=C(C=C1)Br	-1
99-92-3	CC(=O)C1=CC=C(C=C1)N	-1

# Section S3. Semantic Literature Analysis

## **S3.1.** Dataset Curation

To build a targeted reaction dataset for training machine learning models, we employ semantic literature analysis to identify suitable publications based on human-defined criteria. This approach allows for highly customizable searches, tailored to specific reaction descriptions, which are often difficult to capture using traditional database queries. Traditional database queries, while effective for broad searches, often struggle with specificity. For instance, selecting reactions labeled as "electrochemical" is feasible, but pinpointing electrochemical oxidation on aliphatic carbons presents significant challenges. Importantly, large language models LLMs can be customized to meet specific research needs. For niche areas of chemistry, LLMs can prioritize pertinent information more effectively than traditional queries and natural language processing methods, which may have limitations in search algorithms and data categorization. Furthermore, LLMs allow chemists to define criteria in everyday language, minimizing the need for extensive coding expertise. We began by broadly identifying potential publications using keyword searches. In this study, our focus was on compiling a dataset for electrochemical C-H oxidation with a mediator. We initiated our search on the Web of Science using the keywords "Electrochemical," "C-H oxidation," and "Hydrogen atom transfer," yielding an initial list of 140 publications. The selected papers for this study, along with their DOIs, publication years, and titles, are detailed in Table S4.

However, simple keyword searches often include literature that does not meet specific criteria. For example, computational studies without any synthesis or other types of electrochemical reactions might be included. Manually filtering through each publication is challenging and time-consuming. Therefore, the primary purpose of using LLMs for semantic analysis is to create a customizable filter that can efficiently process large corpora of text. LLMs consider all aspects discussed in the manuscripts, including reactions, analyses, and discussions, to determine if a paper meets the defined criteria, providing a "yes" or "no" answer. For highly specific reactions, the final number of relevant manuscripts is typically small. LLMs expedite the identification process by quickly eliminating unsuitable papers, saving time, and enhancing accuracy. In this study, we downloaded all manuscripts in PDF format following the guidelines and permissions outlined in the agreement between MIT and the publishers. Data handling was conducted through an API, ensuring the content of the papers remained private to the models and that conversation history was not stored elsewhere for training purposes. The full information of all the papers can be found on the Spreadsheet SF1. Literature Screening Dataset in the Supporting Information.



Figure S11. Overview of the key steps in reaction data mining assisted by the LLM.

Publishers	Prefix	Count
Wiley	10.1002	32
Springer	10.1007, 10.1134	6
Elsevier	10.1016	21
ACS	10.1021	50
Nature	10.1038	4
RSC	10.1039	16
Thieme	10.1055	3
J-STAGE	10.1248	3
Other publishers	10.1070, 10.3390, 10.3866, 10.6023	5

**Table S3.** List of publishers, and the associated DOI prefix, along with the number of publications used for the dataset generation.



Figure S12. Statistics on the distribution of the publications from different publishers.

DOI Number	Title	Year
10 1002/tor 202200221	Electrochemical C-H/C-C Bond Oxygenation: A Potential Technology for Plastic	;
10.1002/101.202300331	Electrochemical Coupling Reactions Using Non-Transition Metal Mediators:	2023
10.1002/ejoc.202300835	Recent Advances	2023
10.1021/jacs.3c09907	the Aid of Imidazolium Radical Mediators	2023
10.1021/acs.orglett.3c02668	HAT-Mediated Electrochemical C(sp2)-HAcylation of Quinolines with Alcohols	2023
10.1021/jacs.3c06477	Competition between C-C and C-H Bond Fluorination: A Continuum of Electron Transfer and Hydrogen Atom Transfer Mechanisms	2023
10.1002/anie.202310138	Oxidation	2023
10.1021/acs.orglett.3c03036	Dehalogenative Arylation of Unactivated Alkyl Halides via Electroreduction	2023
10.1002/anie.202310072	Organocatalytic C-H Functionalization of Simple Alkanes	2023
10.1039/d3ac02153i	Electrochemical cascade sequences for remote C7-H bond thiocyanation of quinoxalin-2(1H)-ones with ammonium thiocyanate	2023
10.1002/cioc.202300288	Chloride-Promoted Photoelectrochemical C-H Silvlation of Heteroarenes	2023
10 1002/adec 202300644	HAT-Mediated Electrochemical Cross Dehydrogenative Coupling Reaction of Outpoxalin-2(1H)-ones with Ethers	2023
10.1002/aust.202300044	Hypervalent Halogen Compounds in Electrochemical Reactions: Advantages	2023
10.1002/adsc.202300412	and Prospects	2023
10.1021/acscatal.3c01221	C(sp3)/Si-H Functionalization and Hydrofunctionalization of Alkenes Degradation pathways of atrazine by electrochemical oxidation at different	2023
10.1016/j.envpol.2023.121987	Current densities: Identifications from compound-specific isotope analysis and DFT calculation Selective C-H Acyloxylation of Sulfides/Disulfides Enabled by Hypervalent	2023
10.1002/adsc.202300360	lodine Reagents	2023
10.1016/j.tetlet.2023.154483	N-Heterocyclic carbene-catalyzed remote C(sp3)-H acylation of amides	2023
10.1016/j.checat.2023.100582	C(sp3)-H amination	2023
10.1002/cctc.202300009	Sulfide Electrocatalyst	2023
10.1021/acs.inorgchem.2c04168	Structural and Spectroscopic Characterization of Copper(III) Complexes and Subsequent One-Electron Oxidation Reaction and Reactivity Studies	2023
10.1002/adsc.202201395	Dehydrogenative Coupling	2023
10.1007/s40242-023-2323-y	Solvent-dependence of KI Mediated Electrosynthesis of Imidazo[1,2-a]pyridines	2023
10.1016/j.checat.2022.100491	Copper-catalyzed electrochemical C-H fluorination	2023
10.1038/s41586-022-05667-0	Electro chemical reactor dictates site selectivity in N-heteroarene carboxylations	2023
10.1055/a-1992-7066	Electrochemical Benzylic C(sp 3)-H Amidation via Ritter-Type Reaction in the Absence of External Mediator and Oxidant	2023
10.1021/acs.jpcc.2c07736	Methane to Methanol Conversion over N-Doped Graphene Facilitated by Electrochemical Oxygen Evolution: A First-Principles Study	2023
10.1021/acs.joc.2c02363	Direct Hydroxylarylation of Benzylic Carbons (sp3/sp2/sp) via Radical-Radical Cross-Coupling Powered by Paired Electrolysis	2023
10.1039/d2qo01498j	NHPI-catalyzed electrochemical C-H alkylation of indoles with alcohols to access di(indolyl)methanes via radical coupling	2022
10.1016/j.apsadv.2022.100319	Influence of temperature, surface composition and electrochemical environment on 2-propanol decomposition at the Co3O4 (001)/ H2O interface	2022
10.1016/j.jelechem.2022.116835	Electrochemical quinuclidine-mediated C-H activation: Intermediates and mechanism	2022
10.1021/acscatal.2c03424	A Horseradish Peroxidase-Mediator System for Benzylic C-H Activation	2022
10.1002/anie.202210201	Metal-Free Electrochemical Carboxylation of Organic Halides in the Presence of Catalytic Amounts of an Organomediator	2022
10 1002/acto 202200820	How to Harness Electrochemical Mediators for Photocatalysis - A Systematic	2022
10.1002/0010.202200030	Experied and Reactivity of a Electing Nill Risphaneyy/ Direction Sector	2023
10.1002/dille.202211343	An efficient electrochemical oxidation of C(sp3)-H bond for the synthesis of	2022
10.1016/J.mcat.2022.112633	aryiketones	2022

# Table S4. Title of papers, the associated DOI number, along with publication years.

10.1021/acs.accounts.2c00444	N-O Bond Activation by Energy Transfer Photocatalysis	2022
10.1002/ejoc.202200596	Cross-Coupling of C-H and N-H Bonds: A Hydrogen Evolution Strategy for the Construction of C-N Bonds Electrochemical C(sp3)-H Eurocionalization of y-L actams Based on Hydrogen	
10.1021/acs.orglett.2c01528	Atom Transfer	2022
10.1016/j.jechem.2022.03.036	A new pathway for formic acid electro-oxidation: The electro-chemically decomposed hydrogen as a reaction intermediate	2022
10.1002/chem.202104493	Sulfonyl Nitrene and Amidyl Radical: Structure and Reactivity	2022
10.1021/jacs.1c13063	Imines	2022
10.1039/d1cs00494h	Chemical versatility of azide radical: journey from a transient species to synthetic accessibility in organic transformations Electrochemical Activation of C-C Bonds through Mediated Hydrogen Atom	2022
10.1002/cssc.202102317	Transfer Reactions	2022
10.1039/d1fd00074h	Electrocatalytic alcohol oxidation by a molecular iron complex	2022
10.1016/j.tetlet.2022.153647	An electrochemical $\gamma$ -C-H arylation of amines in continuous flow	2022
10.1021/acs.chemrev.1c00374	Photochemical and Electrochemical Applications of Proton-Coupled Electron Transfer in Organic Synthesis	2022
10.1016/j.tetlet.2021.153593	of aromatic α-diketones	2022
10.1055/sos-SD-126-00120	Synthesis of Ketones by Oxidation of Alkanes	2022
10.1002/anie.202115178	Electrochemical Desaturative β-Acylation of Cyclic N-Aryl Amines	2022
10.1021/jacs.1c07983	An Expanded SET Model Associated with the Functional Hindrance Dominates the Amide-Directed Distal sp3 C-H Functionalization	2021
10.1002/chem.202103384	C(sp2)-H Activation	2022
10.6023/cjoc202106001	Applications of Proton-Coupled Electron Transfer in Organic Synthesis	2021
10.1021/jacs.1c08671	Secondary Acyclic Amines with Ketones	2021
10.1021/jacs.1c06473	Electrochemical Borylation of Alkyl Halides: Fast, Scalable Access to Alkyl Boronic Esters	2021
10.1002/celc.202100784	TEMPO-Mediated Electrochemical N-demethylation of Opiate Alkaloids	2021
10.1038/s41467-021-24203-8	Electrochemical activation of C-H by electron-deficient W2C nanocrystals for simultaneous alkoxylation and hydrogen evolution	2021
10.1021/acs.organomet.1c00188	Stability, and Carbon-Hydrogen Bond Activations	2021
10.1021/jacs.1c03780	N-Ammonium Ylide Mediators for Electrochemical C-H Oxidation Mediator-Enabled Electrocatalysis with Ligandless Copperfor Angeropic Chan-	2021
10.1021/jacs.1c02103	Lam Coupling Reactions	2021
10.1002/anie.202012707	Efficient Aerobic Oxidation of Organic Molecules by Multistep Electron Transfer	2021
10.1039/d0ob02444a	ketones	2021
10.1002/ejic.202000955	Dirhodium(II,II) Paddlewheel Complexes	2021
10.1002/anie.202010437	Selective Electrochemical Hydrolysis of Hydrosilanes to Silanols via Anodically Generated Silyl Cations	2021
10.1039/d0sc03616a	and Cxe001:C bonds using water as the oxygen source	2020
10.3390/catal10090982	Recent Advances in Asymmetric Catalytic Electrosynthesis	2020
10,1020/d0a.01224b	Electrifying green synthesis: recent advances in electrochemical annulation	2020
10.1039/00gc01324b	Reactions	2020
10.1002/anie.201913767	Synthetic Photoelectrochemistry Determining Proton-Coupled Standard Potentials and X-H Bond Dissociation Free Energies in Nonagueous Solvents Using Open-Circuit Potential	2020
10.1021/jacs.0c01032	Measurements	2020
10.1039/d0sc01898h	Carboxylate breaks the arene C-H bond via a hydrogen-atom-transfer mechanism in electrochemical cobalt catalysis	2020
10 1021/acs in orachem 9603767	Catalytic C-H Bond Oxidation Using Dioxygen by Analogues of Heme	2020
10.1021/203.0019010000000	Reductive Electrochemical Activation of Molecular Oxygen Catalyzed by an Iron-Tungstate Oxide Capsule: Reactivity Studies Consistent with Compound I	2020
10.1021/acscatal.0c00897	Type Oxidants	2020

10.1002/anie.202001149	Electrochemical Synthesis of Thienoacene Derivatives: Transition-Metal-Free Dehydrogenative C-S Coupling Promoted by a Halogen Mediator Electrochemical Oxidation of Organic Molecules at Lower Overpotential: Accessing Broader Functional Group Compatibility with Electron-Proton	2020
10.1021/acs.accounts.9b00544	Transfer Mediators	2020
10.1021/acs.accounts.9b00603	and Transition Metal Catalysis	2020
10.1021/jacs.9b11472	Electrophotocatalytic C-H Functionalization of Ethers with High Regioselectivity	2020
10.1021/acs.inorgchem.9b02080	Conducting Coronene Cation Radical Salt Containing Magnetic Metal lons	2019
10.1002/anie.201903902	O-O Bond Formation and Liberation of Dioxygen Mediated by N5-Coordinate Non-Heme Iron(IV) Complexes	2019
10.1021/acs.inorgchem.9b00247	Oxidation Catalysis and Ligand Transformation	2019
10.1021/acscatal.9b01394	Atom Transfer	2019
10.1002/chem.201806092	Remote Hydrogen Atom Transfer in Batch and Continuous Flow	2019
10.1002/anie.201813960	Merging Photochemistry with Electrochemistry: Functional-Group Tolerant Electrochemical Amination of C(sp3)-H Bonds	2019
10.1021/acscatal.9b00287	Visible Light Uranyl Photocatalysis: Direct C-H to C-C Bond Conversion	2019
10.1021/jacs.8b08460	$\beta$ -Agostic Ethyl: Structure, Dynamics and Mechanism	2018
10.1002/anie.201809611	Catalyst Cooperation	2018
10.1021/jacs.8b07380	Amines	2018
10.1038/s41467-018-06020-8	of saturated O-heterocycles	2018
10.1002/elan.201800147	Tetrachloro-Phthalimido-N-Oxyl (Cl4PINO) Catalyst	2018
10.1055/s-0036-1591558	Acids	2018
10.1039/c8ob00063h	Recent advances in iodine mediated electrochemical oxidative cross-coupling	2018
10.1039/c7sc05224c	cis-Oxoruthenium complexes supported by chiral tetradentate amine (N4) ligands for hydrocarbon oxidations	2018
10.1039/c7sc04032f	Selective electrochemical generation of benzylic radicals enabled by ferrocene- based electron-transfer mediators	2018
	&ITN&IT-Hydroxyphthalimide-Mediated Electrochemical Iodination of Methylarenes and Comparison to Electron-Transfer-Initiated C-H	
10.1021/jacs.7b09744	Functionalization	2018
10.1021/acs.joc.7b01686	lodine	2017
10.1007/s11696-017-0201-0	Manganese(II) complexes with Bn-tpen as powerful catalysts of cyclohexene oxidation	2017
10.1021/acs.accounts.7b00343	Manganese-Oxygen Intermediates in O-O Bond Activation and Hydrogen-Atom Transfer Reactions	2017
10.1016/j.jmgm.2017.08.016	Catalytic hydrogenation of CO2 over Pt- and Ni-doped graphene: A comparative DFT study	2017
10 1016/i molstruc 2017 05 044	Synthesis, characterization, crystal structure and electrochemical studies of ionic iron(III) dipicolinate complex	2017
	Electrochemical and electron paramagnetic resonance study of the mechanism	2011
	electrocatalytic oxidation of isopropyl alcohol at glassy carbon and single-	
10.1016/j.jelechem.2017.04.031	walled carbon nanotube electrodes	2017
10.1021/jacs.7b03539	Scalable, Electrochemical Oxidation of Unactivated C-H Bonds	2017
10.1021/acs.inorgchem.7b00448	FeV=O and FeIV=O Units	2017
10.1021/jacs.6b13091	Fast Hydrogen Atom Abstraction by a Hydroxo Iron(III) Porphyrazine Tuning the Reactivity of Terminal Nickel(III)-Oxygen Adducts for C-H Bond	2017
10.1021/jacs.6b08406	Activation	2016
10.1038/nature17431	Scalable and sustainable electrochemical allylic C-H oxidation Combining Nanosecond and Millisecond Time Scale Techniques:	2016
	Determination of Thermodynamic and Kinetic data of Primary Alkyl Amine	
10.1021/jp5109366	Cation Radicals	2015

10.1039/c5dt00863h	Iridium-based complexes for water oxidation	2015
10.1016/j.electacta.2014.08.039	Electrochemical and Electron Paramagnetic Resonance Study of the Mechanism of Oxidation of Phenazine-di-N-oxide in the Presence of Isopropyl alcohol at Glassy Carbon and Single-Walled Carbon Nanotube Electrodes A comparative study of organic electron transfer redox mediators: electron transfer kinetics for trianylimidazole and trianylamine mediators in the oxidation	2014
10.1016/j.electacta.2014.07.105	of 4-methoxybenzyl alcohol	2014
10.1021/om401225r	Homogeneous Catalysis and Selectivity in Electrochemistry	2014
10.1021/ic500943k	OH Complex Derived from Dioxygen	2014
10.1021/ja505136j	Direct, Catalytic Monofluorination of sp3 C-H Bonds: A Radical-Based Mechanism with Ionic Selectivity Electrochemical and ESR Study of the Mechanism of Oxidation of Phenazine-	2014
10.1134/S1023193514010066	di-N-oxide in the Presence of Cyclohexanol on Glassy Carbon and Single- Walled Carbon Nanotube Electrodes Efficient Indirect Electrochemical Synthesis of 2-Substituted Benzoxazoles	2014
10.1002/adsc.201300502	using Sodium Iodide as Mediator	2013
10.1039/c2sc21318d	electrochemistry, and oxidative reactivity	2013
10.1134/S1023193512040088	oxides in the presence of methanol and its deuterated derivatives	2012
10.1021/ic202462a	Platinum(II) Dimine Complexes with Halide/Pseudohalide Ligands and Dangling Trialkylamine or Ammonium Groups	2012
10.1021/oI300195c	and Applicability to Electrocxidative C-H Activation	2012
10.1021/ja211005g	Role of Water or Methanol	2012
10.1039/c0ob01257b	structural effects on the intrinsic electron transfer reactivities	2011
10.3866/PKU.WHXB20100925	Photoinduced Electron and Hydrogen Transfer Reactions of Thioxanthone with Amines, Phenols and Alcohols Electrochemical and ESR studies of tert-butanol oxidation mechanism in the	2010
10.1134/S1023193510090119	presence of radical cations pyrazine-di-N-oxide and its substituted derivatives as mediators Redox Potential and C-H Bond Cleaving Properties of a Nonheme FelV-O	2010
10.1021/ja909923w	Complex in Aqueous Solution	2010
10.1016/j.matchemphys.2008.12.0 33	) The electron transfer behavior of the hydrogen-terminated boron-doped diamond film electrode	2009
10.1016/j.ica.2007.10.053	Complexes of oxomethoxovanadium(V) with tridentate thiobenzhydrazide based Schiff bases	2008
10.1002/chem.200801615	Intervalent Bis(µ-aziridinato)MII-MI Complexes (M = Rh, Ir): Delocalized Metallo-Radicals or Delocalized Aminyl Radicals? Electrochemical and ESR-study of the mechanism of organic compound	2008
10.1134/S1023193507110031	oxidation in the presence of mediators-Radical cations of substituted pyrazin-di- N-oxydes Syntheses, characterization, and dioxygen reactivities of Cu(I) complexes with	2007
10.1021/ic062206s	cis,cis-1,3,5-triaminocyclohexane derivatives:: A Cu(III)2O2 intermediate exhibiting higher C-H activation Hydrocarbon oxidation by $\beta$ -halogenated dioxoruthenium(VI) porphyrin	2007
10.1002/chem.200500814	complexes:: Effect of reduction potential (RuVI/V) and C-H bond-dissociation energy on rate constants	2005
10.1039/b507657a	the oxidation of non-phenolic substrates	2005
10.1016/S1381-1177(03)00014-6	Free radical versus electron-transfer routes of oxidation of hydrocarbons by laccase/mediator systems catalytic or stoichiometric procedures	2003
10.1021/om020621w	C-H oxidative addition with a (PCP)Ir(III)-pincer complex	2002
10.1016/S0013-4686(02)00448-6 10.1070/rc2002v071n03abeh0007	oxidation with participation of radical cation of phenazine-di-N-oxide	2002
04	Electrochemical activation of reactions involving organometallic compounds Oxidative single-electron transfer activation of $\sigma$ -bonds in alightic halogenation	2002
10.1021/ja000193q	reactions Molecular meccano - Part 58 - The electrochemically-driven	2000
10.1021/jo991467z	decomplexation/recomplexation of inclusion adducts of ferrocene derivatives with an electron-accepting receptor	2000

10.1016/S0022-0728(98)00206-X	Electrochemical reduction of acids in dimethyl sulfoxide. Comparison of weak C-H, N-H and O-H acids	1998
10.1016/0304-5102(93)E0230-E	Electrocatalytic functionalization of alkanes using aqueous platinum salts Anodic-oxidation of amides and lactams using N-hydroxyphthalimide as a	1994
10.1248/cpb.34.975	mediator	1986
10.1248/cpb.33.4798	Electrochemical oxidation of olefins using N-hydroxyphthalimide as a mediator An odic-oxidation of compounds having benzylic or allylic carbon and alpha-	1985
10.1248/cpb.31.4209	carbon to hetero atom using N-hydroxyphthalimide as a mediator N-hydroxyphthalimide as an effective mediator for the oxidation of alcohols by	1983
10.1039/c39830000479	electrolysis	1983



**Figure S13.** Statistics on the length of manuscripts for data mining. The manuscripts can be classified into two main categories: shorter communication-style manuscripts and longer full articles. The average word count for communication-style manuscripts, depicted by the purple dashed line, is 2297 words. In contrast, the average word count for full articles, indicated by the green dashed line, is 4743 words. The supporting information associated with these manuscripts is not included in this analysis. The overall average word count for all manuscripts is 3665 words, represented by the red dashed line.

### **S3.2.** Prompt Engineering

**User**: Please answer the following question truthfully, based on the complete text of the literature provided below.

Question: Does this paper report experimental findings related to the electrochemical synthesis conditions for aliphatic C-H oxidation reactions?

Note that to answer "Yes" to this question, all three criteria must be met: (a) wet lab experimental electrochemical synthesis, (b) transformation from C-H to C-OH or C=O, and (c) the reaction occurs on aliphatic carbon. If any of these criteria are not mentioned, the overall answer should be "No." In your response, adhere strictly to the format below and address each part with specific references to the text:

Overall Answer: {Yes/No} Part A: {Yes/No} Reasoning A: {A brief justification for this part of your answer, citing relevant sections of the paper} Part B: {Yes/No} Reasoning B: {A brief justification for this part of your answer, citing relevant sections of the paper} Part C: {Yes/No} Reasoning C: {A brief justification for this part of your answer, citing relevant sections of the paper} Make sure you consider all sentences in the manuscript and always keep the following questions in mind: Does the paper involve wet lab experimental procedures in electrochemical synthesis? Part A: Does the paper include experimental procedures and non-computational/non-simulation data in electrochemical synthesis? Part B: Does the paper detail C-H oxidation specifically leading to C-H bonds (alcohol) or C=O bonds (carbonyl) compounds? Note that coupling products where C-H becomes C-OR group do not count. Also note that only functionalization of C-H bonds without clear indication of C-OH or C=O bonds does not count. Part C: Is the reaction happening in aliphatic carbon? Take a deep breath and read it, consider all the expectations, analyze and give reasoning in detail step by step. Manuscript: {full text}

**Figure S14.** A representative zero-shot conversation using the *gpt-4o* backbone model, tasked with classifying the content of a paper based on user-defined criteria. The full manuscript text was converted to plain text and presented following this prompt. The LLM was provided with a template answer and asked to generate a structured output. The prompt was crafted by a human using previously described principles and techniques.<sup>3–5</sup> It was demonstrated that slight typographical variations in the input prompt do not significantly impact binary classification output from the LLM.

#### Assistant: Overall Answer: Yes

#### Part A: Yes

Reasoning A: The paper discusses electrochemical methods for the selective oxidation of C(sp3)-C/H(sp3) bonds, which is a part of experimental electrochemical synthesis (sections: Abstract, 1 Introduction).

#### Part B: Yes

Reasoning B: The paper describes the conversion of carbon-hydrogen (C–H) bonds to carbon-oxygen (C–O) bonds, indicating a transformation from C-H to C-OH or C=O. This is evident in several sections, including the Abstract and the Introduction, where C–H oxygenation and the production of aldehydes and ketones (C=O compounds) from C–H bonds are mentioned.

#### Part C: Yes

Reasoning C: The reactions discussed in the paper focus on the selective activation of inert C(sp3)-C/H(sp3) bonds, indicating that the reaction occurs on aliphatic carbon. This is stated in the Abstract and further elaborated in the section "Electrochemical C-H/C-C Bond Oxygenation."

**Figure S15.** A representative response provided by the LLM, featuring an overall binary label (Yes/No) for the manuscript<sup>6</sup> and answers for each section (A, B, C) accompanied by a paragraph of reasoning. This detailed explanation helps facilitate subsequent human analysis and enhances the understanding of the decision-making process. Note that while figures and graphs from the literature can be optionally incorporated into the LLM's evaluation,<sup>7,8</sup> this study only relied on textual input for examining semantic analysis purposes. The complete collection of the output answers for all 140 manuscripts for this study is available in Spreadsheet SF1 Literature Screening Dataset in the Supporting Information.

#### Assistant: Overall Answer: No

#### Part A: No

Reasoning A: The study involves the use of density functional theory (DFT) calculations to investigate the hydrogenation of CO2 on Pt- and Ni-doped graphene surfaces. This method falls under computational simulation, not wet lab experimental electrochemical synthesis (Section 2, "Computational details").

#### Part B: No

Reasoning B: The paper focuses on the hydrogenation of CO2 into formic acid (HCOOH) over Pt- and Ni-doped graphene. This involves the transformation of CO2 (with C=O bonds) into formic acid. However, there is no specific mention of the transformation of C-H to C-OH or C=O, as required by the question's criteria (Sections 3.1, 3.4).

#### Part C: No

Reasoning C: The reactions discussed in the paper involve CO2 molecules and their conversion to formic acid on Pt- and Ni-doped graphene surfaces. The carbon in these reactions is not aliphatic but originates from CO2 molecules, which does not fulfill the criterion of reactions occurring on aliphatic carbon (Sections 3.1. 3.4).

**Figure S16.** A representative response provided by the LLM, featuring an overall binary label (Yes/No) for the manuscript<sup>9</sup> and answers for each section (A, B, C) accompanied by a paragraph of reasoning.

**User**: Question: Does this paper report experimental findings related to the electrochemical synthesis conditions for aliphatic C-H oxidation reactions?

Note that to answer "Yes" to this question, all three criteria must be met: (a) wet lab experimental electrochemical synthesis, (b) transformation from C-H to C-OH or C=O, and (c) the reaction occurs on aliphatic carbon. If any of these criteria are not mentioned, the overall answer should be "No."

In your response, adhere strictly to the format below and address each part with specific references to the text:

\*\*\*

Overall Answer: {Yes/No} Part A: {Yes/ No} Reasoning A: {A brief justification}

Part B: {Yes/ No} Reasoning B: {A brief justification}

Part C: {Yes/ No} Reasoning C: {A brief justification}

\*\*\*

Manuscript: {full text}

**Figure S17.** An input prompt designed with vague and simplistic instructions for an ablation study, omitting crucial elements such as directives to adhere strictly to the provided text when answering and not including specific examples for the LLM. This lack of detail introduces greater uncertainty regarding the objectives, as the prompt does not provide sufficient guidance. This prompt was tested using the same task and model for semantic literature analysis to evaluate performance.

#### S3.3. Evaluation Metrics and Results Analysis

We first analyzed the processing time required for large language models to handle 140 manuscripts stored in a common folder. Each manuscript was converted to plain text format, and the text was combined with the pre-designed prompt described in the previous section. The conversion process was performed sequentially, resulting in 140 instances of interaction with the LLM via the API. The response time for the LLM was recorded for each manuscript, correlating with the length of tokens in both the input (manuscript reading) and output (LLM's response). As previously noted, the response for the LLM adheres to a specific template, ensuring the length of the output remains relatively constant. Therefore, the primary factor influencing the response time is the length of the input prompt, which is a combination of a fixed prompt and the manuscript's length. Consequently, longer manuscripts lead to extended response times. The figure below illustrates that the LLM's processing time ranges from 5 to 23 seconds, with an average of 14.59 seconds. It is important to highlight that the same fixed prompt was utilized for all interactions, and no human intervention was necessary during the LLM's processing of the manuscripts.



Figure S18. The variation in processing times across the 140 manuscripts.

Once the automated semantic literature analysis on 140 publications was completed, we aimed to evaluate the performance of large language models to determine their reliability in filtering less relevant literature and pinpointing those that chemists are interested in based on human natural language prompts. We also wanted to assess the extent to which LLMs can perform accurate classification and quantify this accuracy. An inadequately prompted model can produce erroneous outputs, known as hallucinations, where the model generates information that is not accurate, factual, or consistent with the input or real-world data. In our study, the primary objective is to assign a "Yes" or "No" label to each piece of literature to see if it fits the scope the human is looking for, making the model's precision and reliability crucial. This is especially true given the need for semantic analysis based on the content of the literature.

First of all, we separated the entire search into three criteria: (i) determine if this paper is a wet lab experiment paper, (ii) determine if this paper is for C-H oxidation reaction, and (iii) determine if this reaction is happening on the aliphatic carbon. An overall label is assigned as "Yes" only if all three sections have a "Yes" answer. For the selected 140 publications, we created the ground truth labels for each of them for both the overall answer and the three sections by having human volunteers review the papers and assign a "Yes" or "No" to each individual section for each paper. A paper with three "Yes" answers in the three sections is given a "Yes" for the overall answer, while the rest receive a "No" for the overall answer. The complete list of ground truth labels associated with each publication can be found in Spreadsheet SF1. Literature Screening Dataset in the Supporting Information. Once the relevant papers were identified, we used their DOI, title, and abstract to query the Reaxys database (https://www.reaxys.com/) to retrieve the reaction and product SMILES. It is important to note that, in this study, we did not specifically ask large language models to extract reactions. Instead, we focused on filtering irrelevant papers and identifying the relevant papers based on their content and then used those papers to link to the reactions. Since the dataset was relatively small, the retrieved reactions and substrates could be conveniently verified by humans using an existing curated chemical database. However, it is worth noting that prior studies have shown that LLMs are capable of performing both identification and extraction tasks for chemical synthesis and organic reactions, yielding structured outputs related to the reactions discussed in the papers.<sup>10–12</sup>

Additionally, we aimed to investigate the importance of well-designed prompts in guiding LLMs for semantic analysis. We sought to understand the impact of different levels of detail in prompts and demonstrate how applying specific prompt engineering strategies<sup>3,4</sup> can enhance model performance. To this end, we designed two versions of prompts based on established principles for writing effective prompts: (a) detailed prompt: This version includes comprehensive instructions and examples to provide clear guidance to the LLM. (b) simple prompt: This version is shorter and less detailed, providing minimal guidance. By conducting an ablation study using these two versions of prompts, we aimed to evaluate the effectiveness of prompt engineering strategies. If these strategies enhance the model's understanding of the tasks, we should observe a difference in accuracy between the two versions. It is anticipated that the detailed instructions and reduce uncertainty regarding the goals of semantic analysis. Overall, this evaluation not only assesses the performance of LLMs in classifying relevant literature but also explores how prompt design can influence the accuracy and reliability of semantic analysis.

Next, to evaluate the LLMs' performance, we used the following metrics. Before explaining these metrics, it's important to define the terms True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN):

(1) **True Positive** (TP): This occurs when the LLM assigns a "Yes" overall answer to a paper, and the ground truth confirms that the paper fits all the criteria (it is a relevant paper that satisfies chemist's criteria).

(2) **False Positive** (FP): This happens when the LLM assigns a "Yes" overall answer to a paper, but the ground truth indicates that the paper does not fit all the criteria (it is not a relevant paper).

(3) **True Negative** (TN): This occurs when the LLM assigns a "No" overall answer to a paper, and the ground truth confirms that the paper does not fit all the criteria (it is not a relevant paper).

(4) **False Negative** (FN): This happens when the LLM assigns a "No" overall answer to a paper, but the ground truth indicates that the paper does fit all the criteria (it is a relevant paper).

Using these definitions, we can evaluate the LLM's performance with the following metrics: (1) **Accuracy**: Accuracy measures the overall correctness of the model's predictions. It is the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances.

(2) **Precision**: Precision quantifies the accuracy of positive predictions, indicating the proportion of true positive results out of all positive results predicted by the model.

(3) **Recall**: Recall evaluates the model's ability to identify all relevant instances, indicating the proportion of true positive results out of all actual positive instances.

(4) **F1 Score**: The F1 score is the harmonic mean of Precision and Recall, providing a balance between the two metrics. It is useful when you need to account for both false positives and false negatives.

Metric	Equation
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
F1 Score	$2 \times \frac{Precision \times Recall}{Precision + Recall}$

Table S5. Equations used to calculate the performance metrics.

Prompt	<b>TP Count</b>	<b>TN Count</b>	<b>FP</b> Count	FN Count	Total
Detailed	21	114	3	2	140
Simple	10	111	4	15	140

**Table S6.** Performance of LLM based on different prompts (Figure S14 and Figure S17) for the same list of literature.

**Table S7.** Overall evaluation metrics for two different prompts. The calculations are based on the values shown in Table S6 using the equations from Table S5.

Prompt	Accuracy	Precision	Recall	F1 Score
Detailed	96.43%	87.50%	91.30%	93.80%
Simple	86.43%	71.43%	40%	54.69%



**Figure S19.** Overall evaluation metrics for two different prompts. A significant loss in the recall performance resulted in a lower F1 score.

Prompt	Part A Accuracy	Part B Accuracy	Part C Accuracy
Detailed	97.14%	90%	95.71%
Simple	80.71%	81.43%	77.14%

Table S8. The breakdown accuracy performance for each part.



Yercentage by Prompt for Part A Accuracy, Part B Accuracy, and Part C Accuracy

Figure S20. Breakdown of accuracy performance for each part. Without proper prompt engineering strategies, the LLM can have a poorer understanding of human instructions, which can be ambiguous.

# Section S4. Reactivity and Selectivity Prediction

# S4.1. Reactivity Model

## **DFT** calculations

Density functional theory (DFT) quantum mechanical (QM) descriptors were calculated using a modified version of the automated workflow detailed in prior research.<sup>13,14</sup> For each molecule examined in this study, we began by generating its 3D molecular conformers from input SMILES strings using the MMFF94s<sup>15</sup> force field in RDKit<sup>16</sup>. We produced 20 conformers per molecule and selected the one with the lowest MMFF94s energy for further optimization at the GFN2-xTB<sup>17</sup> level of theory, followed by a frequency calculation at the same level to confirm the absence of imaginary frequencies. Subsequently, species that passed the frequency check were optimized again using the wB97XD<sup>18</sup> functional and def2-svp basis set<sup>19</sup> in Gaussian 16<sup>20</sup>. Due to potential DFT optimization failures, this procedure was repeated with different initial conformer sampling. Structural and vibrational frequency checks were conducted to ensure the final optimized geometries at the DFT level had converged. Natural bond orbitals and associated descriptors were then computed using NBO 7.0. The computed QM descriptors used in this study are listed in Table S9 below, and their values are provided in the attached file. In the end, 548 out of 575 (95.3%) species successfully generated the QM descriptors, while the remaining species failed due to DFT optimization issues.

Catagory	Descriptors
Atom	NPA charge, shielding constant, 1s valence orbital occupancy, 2s valence orbital occupancy, 2p valence orbital occupancy, 3s valence orbital occupancy, 3p valence orbital occupancy, 4s valence orbital occupancy, 4p valence orbital occupancy
Bond	Bond order, bond length, natural ionicity
Molecule	Dipole moment, traceless quadrupole moment, HOMO-3/LUMO, HOMO-3/LUMO+1, HOMO-3/LUMO+2, HOMO-3/LUMO+3, HOMO-2/LUMO, HOMO-2/LUMO+1, HOMO-2/LUMO+2, HOMO-2/LUMO+3, HOMO-1/LUMO, HOMO-1/LUMO+1, HOMO-1/LUMO+2, HOMO/LUMO, HOMO/LUMO+1, HOMO/LUMO+2, HOMO/LUMO+3

Table S9. List of Quantum Mechanical Descriptors used for Chemprop.

# Data processing and model training

We employed a directed message-passing neural network (D-MPNN) using Chemprop 1.7.1<sup>21</sup> to train machine learning models for predicting oxidation reactivity. The D-MPNN architecture enables the extraction of molecular features by iteratively updating bond representations through message passing, effectively capturing the dependencies and interactions within the molecular graph. This learned representation is subsequently fed into a fully connected neural network, which refines the feature space to predict oxidation reactivity. As a baseline, we utilized a random forest
model with Morgan fingerprints of radius 2 and 2048 bits, providing a robust comparative benchmark to evaluate the performance of the D-MPNN model.

All models were trained with default hyperparameters for 200 epochs across 5-fold cross-validation and 5 independent model ensembles. We employed different data-splitting strategies, including random splits and Bemis-Murcko scaffold splits, maintaining an 80:10:10 ratio for training, validation, and testing sets. This approach allowed us to assess the model's robustness and generalizability across diverse molecular scaffolds.

To enhance the model's predictive power, we incorporated a set of QM descriptors as additional features.<sup>14</sup> These descriptors provide detailed information about the electronic properties of molecules, enriching the feature space and potentially improving prediction accuracy. To leverage all the QM-calculated descriptors for every atom, all the hydrogen atoms in the molecular graph are considered. Due to computational limitations, we were able to use only 548 out of 575 data points to train the machine learning models, as some DFT calculations failed. Despite this, the dataset remained sufficiently comprehensive for model training and evaluation.

The performance of D-MPNN models is illustrated in the table below, both with and without the inclusion of QM descriptors, alongside Random Forest models, across various data splits. The models exhibit high accuracy on the test sets for both random splits and scaffold splits, with most metrics exceeding 90%. Notably, incorporating QM descriptors into the D-MPNN model marginally enhances its performance, indicating that QM information can improve the model's generalizability beyond that of a conventional graph neural network. Overall, both D-MPNN and Random Forest models are highly effective in predicting reaction activity.

**Table S10.** Overall evaluation metrics for reactivity models trained on different data splits. The error bars represent the standard deviation of the values across 5-fold.

		Accuracy	ROC-AUC	Precision	Recall	F1 Score
Random Forest		0.92 +/- 0.01	0.97 +/- 0.02	0.88 +/- 0.04	0.97 +/- 0.02	0.92 +/- 0.02
D-MPNN	Random	0.94 +/- 0.02	0.98 +/- 0.01	0.90 +/- 0.02	0.96 +/- 0.03	0.93 +/- 0.02
D-MPNN + QM		0.94 +/- 0.02	0.98 +/- 0.02	0.92 +/- 0.05	0.94 +/- 0.04	0.93 +/- 0.02
Random Forest		0.93 +/- 0.02	0.97 +/- 0.02	0.86 +/- 0.04	0.99 +/- 0.02	0.92 +/- 0.02
D-MPNN	Scaffold	0.91 +/- 0.01	0.96 +/- 0.02	0.87 +/- 0.03	0.94 +/- 0.04	0.90 +/- 0.01
D-MPNN + QM		0.93 +/- 0.04	0.96 +/- 0.03	0.87 +/- 0.05	0.98 +/- 0.02	0.92 +/- 0.04

## Interpretability method

The interpretability method<sup>22</sup> implemented in Chemprop identifies substructures, known as rationales, that significantly contribute to specific properties within molecules using Monte Carlo Tree Search (MCTS). A rationale is a key substructure that drives the desired properties in target molecules. The process begins with a positive molecule that satisfies the property constraint. The molecule is then iteratively pruned by removing peripheral bonds or rings, creating a search tree where each state represents a subgraph.

### S4.2. Selectivity Model

For site-selectivity prediction, we employed a bond-property prediction model implemented in Chemprop. Unlike molecular property prediction, where the entire molecular representation is considered, this approach focuses on each bond within the molecule. After message passing, each bond embedding is processed through a fully connected neural network to predict the reactivity at each site. The models were trained using the same data splits and hyperparameters as the reactivity prediction models. However, a key difference for site-selectivity prediction is that hydrogen atoms were excluded from the molecular featurization process, as reaction sites are not located on hydrogen atoms. This refinement ensures that the model accurately targets the relevant reactive sites, enhancing its predictive accuracy. Since only the bond embedding is used as input to the fully connected neural network, the molecule-level QM descriptors cannot be used in this model; only atom and bond descriptors are utilized. All training splits and scripts used in this study are provided in the Supporting Information.

As shown in table below, models trained on different splits demonstrate reasonable accuracy. Given that the number of oxidizable sites is significantly lower than the number of non-oxidizable sites, the model achieves high accuracy. However, this is accompanied by low precision, recall, and F1 scores. In contrast, the ROC-AUC serves as a robust metric for evaluating model performance in highly imbalanced datasets. Our model achieves a commendable ROC-AUC value exceeding 90%, indicating strong discriminative capability.

		Accuracy	ROC-AUC	Precision	Recall	F1 Score
D-MPNN	Random	0.98 +/- 0.01	0.98 +/- 0.01	0.70 +/- 0.09	0.72 +/- 0.09	0.70 +/- 0.08
D-MPNN + QM	Random	0.98 +/- 0.01	0.98 +/- 0.01	0.71 +/- 0.11	0.72 +/- 0.07	0.71 +/- 0.07
D-MPNN	Cooffold	0.98 +/- 0.01	0.98 +/- 0.01	0.73 +/- 0.15	0.61 +/- 0.24	0.63 +/- 0.17
D-MPNN + QM	Scarroid	0.98 +/- 0.01	0.98 +/- 0.01	0.69 +/- 0.13	0.65 +/- 0.18	0.66 +/- 0.14

**Table S11.** Overall evaluation metrics for reactivity models trained on different data splits. The error bars represent the standard deviation of the values across 5-fold.

# S4.3. Virtual Screening



**Figure S21.** Visualization of the chemical space between the training dataset (red and blue) and the 500,000 commercially available compound dataset (grey) provided courtesy of chemical suppliers (i) Ambeed, (ii) ChemScene, and (iii) Arctom\_Scientific, using Morgan fingerprints and t-SNE.

# Section S5. LLM-Assisted Code Generation



**Figure S22.** An overview of 4 tasks examined in this study and examples of code generated by the LLM for Tasks 1 to 4.

Model Name	Avg Time (sec)	Code Length	Code Correctness	Code Correctness (with reflection)	Code Accuracy	ML Model Performance
llama-3	24	218	54%	76%	72%	96%
llama-3.1	25	224	52%	89%	80%	92%
claude-3-sonnet	16	165	75%	91%	89%	95%
claude-3-opus	37	188	84%	100%	99%	96%
claude-3.5-sonnet	14	216	100%	100%	99%	96%
gpt-3.5-turbo	7	189	75%	90%	88%	95%
gpt-40-mini	9	243	60%	91%	90%	96%
gpt-4-0613	29	180	70%	98%	92%	95%
gpt-4-turbo	19	209	86%	99%	99%	96%
gpt-40	11	218	92%	100%	99%	96%
o1-mini	25	249	94%	100%	100%	96%
o1-preview	37	244	91%	100%	100%	96%

**Table S12.** Summary of performance of different base models on Task 1.

Note: Code correctness is defined as the percentage of code that can be executed without any error messages when repeating the same task 100 times. Code accuracy is defined as the code being both executable and achieving the expected outcome, specifically resulting in a machine learning model with more than 85% accuracy on the test dataset. A higher rate of code correctness and accuracy generally indicates fewer hallucinations, as models prone to hallucination often generate code that either fails to execute or produces incorrect results. The number of times of reflection is set to two, which means there are three attempts in total when self-reflection mode turns on. The average time is measured by counting the response time for the model to generate the entire code, and the code length includes comments.

Model Name	Avg Time (sec)	Code Length	Code Correctness	Code Correctness (with reflection)	Code Accuracy
llama-3	28	476	1%	9%	8%
llama-3.1	22	471	6%	16%	15%
claude-3-sonnet	22	514	0%	18%	14%
claude-3-opus	39	424	18%	60%	56%
claude-3.5-sonnet	15	450	5%	48%	41%
gpt-3.5-turbo	8	322	20%	52%	34%
gpt-40-mini	12	560	5%	27%	27%
gpt-4-0613	33	419	18%	42%	30%
gpt-4-turbo	30	468	26%	66%	64%
gpt-40	16	548	43%	79%	75%
o1-mini	23	764	35%	75%	74%
o1-preview	66	661	50%	90%	85%

**Table S13.** Summary of performance of different base models on Task 2.

Note: Code correctness is defined as the percentage of code that can be executed without any error messages when repeating the same task 100 times. Code accuracy is defined as the code being both executable and achieving the expected outcome, specifically resulting in qualified csv file with correct format and contents. A higher rate of code correctness and accuracy generally indicates fewer hallucinations, as models prone to hallucination often generate code that either fails to execute or produces incorrect results. The average time is measured by counting the response time for the model to generate the entire code, and the code length includes comments.

Model Name	Avg Time (sec)	Code Length	Code Correctness	Code Correctness (with reflection)	Code Accuracy
llama-3	10	121	90%	90%	76%
llama-3.1	9	105	98%	98%	98%
claude-3-sonnet	12	176	95%	95%	94%
claude-3-opus	26	219	90%	98%	94%
claude-3.5-sonnet	11	336	98%	99%	99%
gpt-3.5-turbo	6	205	28%	54%	52%
gpt-40-mini	6	271	80%	81%	81%
gpt-4-0613	25	249	80%	94%	92%
gpt-4-turbo	30	374	76%	96%	94%
gpt-40	11	368	92%	98%	94%
o1-mini	25	358	79%	98%	98%
o1-preview	36	285	91%	100%	99%

**Table S14.** Summary of performance of different base models on Task 3.

Note: Code correctness is defined as the percentage of code that can be executed without any error messages when repeating the same task 100 times. Code accuracy is defined as the code being both executable and achieving the expected outcome, specifically resulting in qualified csv file with correct format and contents. A higher rate of code correctness and accuracy generally indicates fewer hallucinations, as models prone to hallucination often generate code that either fails to execute or produces incorrect results. The average time is measured by counting the response time for the model to generate the entire code, and the code length includes comments.

#### User:

You are an expert in coding, and you always write code in a single, whole Python code block so that your code can be copied and pasted directly into a Jupyter Notebook for execution. Your task is to help a chemist write an ML program to train several models for reactivity prediction. The chemist has a list of compounds in "echem\_train.xlsx" in the same folder, and the first 10 rows look like this:

```
CAS
         Canonical SMILES
                                   Reactivity Oxidation Site
2049-96-9
         CCCCCOC(=0)c1ccccc1
                                       1
                                              2.3
98-51-1 Cc1ccc(C(C)(C)C)cc1
                                   1
                                         1
870259-02-2 C#Cc1cc(C#C)c2ccc3c(C#C)cc(C#C)c4ccc1c2c43 0
                                                      -1
121-69-7 CN(C)c1ccccc1
                                   0
                                       -1
82-38-2 CNc1cccc2c1C(=O)c1ccccc1C2=O
                                          0
                                                -1
1209228-43-2 CCCCOc1ccc(CC)cc1
                                       1
                                              10
2459764-19-1 CCc1ccc(CCNC(=O)C2CC(F)(F)C2)cc1
                                                   2,7
                                           1
853-23-6 CC(=0)OC1CCC2(C)C(=CCC3C4CCC(=0)C4(C)CCC32)C1 1 12
91-17-8
          C1CCC2CCCC2C1
                                     1
                                            4,9
6553-83-9 CCCCCOC(=O)C1CCCCC1
                                          1
                                                11,12,13
```

The reactivity can be 1 or 0.

The goal of the ML model is to predict, based on a given SMILES string of a chemical, using its fingerprint, whether this compound will react (1) or will not react (0).

Please propose five ML models suitable for this case, and perform 5-fold cross-validation for each model.

For the trained models, there is a file called "echem\_testxlsx" with the same layout as the train file, which you can use to evaluate your model performance.

At the end, print out the accuracy performance of each model under cross-validation and on the test set. For example:

RandomForestClassifier: Cross-validation accuracy: 0.924 RandomForestClassifier: Test set accuracy: 0.930

```
SVC: Cross-validation accuracy: 0.915
SVC: Test set accuracy: 0.957
```

**Figure S23.** A representative user prompt used for tasking the LLM for Task 1. The prompt was crafted by a human using previously described prompt engineering strartegies.<sup>3,23</sup> The LLM is tasked with processing a given Excel spreadsheet, making decisions on the machine learning model to choose, performing cross-validation, evaluating the model on a separate test set, and finally reporting the model's performance.

```
import pandas as pd
from rdkit import Chem
from rdkit.Chem import AllChem
from sklearn.model selection import train test split, cross val score
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear model import LogisticRegression
from sklearn.metrics import accuracy score
# Load training data
train data = pd.read excel("echem train.xlsx")
# Load test data
test data = pd.read excel("echem test.xlsx")
# Function to convert SMILES string to Morgan fingerprint
def smiles to fingerprint(smiles):
   mol = Chem.MolFromSmiles(smiles)
    if mol is not None:
        # Calculate fingerprint
        fingerprint = AllChem.GetMorganFingerprintAsBitVect(mol, 2, nBits=2048)
        return list(fingerprint)
    else:
        return None
# Prepare training data
X train = train data['Canonical SMILES'].apply(smiles to fingerprint)
y train = train data['Reactivity']
# Prepare test data
X test = test data['Canonical SMILES'].apply(smiles to fingerprint)
y test = test data['Reactivity']
# Remove rows with None fingerprints (invalid SMILES parsing)
X train = X train.dropna()
y train = y train[X train.index]
X test = X test.dropna()
y_test = y_test[X_test.index]
# Convert the DataFrames to lists
X train = list(X train)
X test = list(X test)
# List of models to train and evaluate
models = [
    ("RandomForestClassifier", RandomForestClassifier()),
    ("SVC", SVC()),
    ("KNeighborsClassifier", KNeighborsClassifier()),
    ("DecisionTreeClassifier", DecisionTreeClassifier()),
    ("LogisticRegression", LogisticRegression(max iter=1000))
]
```

```
# Evaluate each model using 5-fold cross-validation
results = []
for name, model in models:
    print(f"Training {name}")
    cv_scores = cross_val_score(model, X_train, y_train, cv=5, scoring='accuracy')
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    test_accuracy = accuracy_score(y_test, y_pred)
    results.append((name, cv_scores.mean(), test_accuracy))
# Print out the results
for name, cv_accuracy, test_accuracy in results:
    print(f"{name}: Cross-validation accuracy: {cv_accuracy:.3f}")
    print(f"{name}: Test set accuracy: {test_accuracy:.3f}")
```

**Figure S24.** A representative example of the output generated by the LLM *gpt-4o* based on the user's prompt for Task 1. In this example, we assume the user can have any level of background knowledge in machine learning and may not be familiar with all the functions. Therefore, we intentionally exclude detailed information about the machine learning models and relevant parameters, allowing the LLM to make decisions independently. This approach results in variability in the choice of models, parameters, and training methods, as the LLM may produce slightly different answers each time. It is important to repeat the same task multiple times to assess the robustness and reliability of the answers, rather than judging based on a single performance. The task is repeated for all different base models 100 times, and the full conversation, along with all generated code is provided in Spreadsheet SF3. Auto Coding Dataset in the Supporting Information.

```
import numpy as np
import pandas as pd
from rdkit import Chem, DataStructs
from rdkit.Chem import AllChem
from sklearn.model selection import cross val score, train test split
from sklearn.metrics import accuracy score
from sklearn.linear model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn import svm
from sklearn.neural network import MLPClassifier
# Load train and test data
train = pd.read excel('echem train.xlsx')
test = pd.read excel('echem test.xlsx')
# function to convert canonical SMILES strings into ECFP6 fingerprints
def smiles to ecfp6 array (smiles str):
   mol = Chem.MolFromSmiles(smiles str)
   arr = np.zeros((1,), dtype=bool)
    fp = AllChem.GetMorganFingerprintAsBitVect(mol, 2, useFeatures=True)
    DataStructs.ConvertToNumpyArray(fp, arr)
    return arr
# Apply the conversion function to convert canonical SMILES strings into ECFP6 fingerprints
X train = np.array([smiles to ecfp6 array(s) for s in train['Canonical SMILES']])
y train = train['Reactivity'].values
X_test = np.array([smiles_to_ecfp6_array(s) for s in test['Canonical SMILES']])
y test = test['Reactivity'].values
# Define models
models = [(LogisticRegression(), 'Logistic Regression'),
          (RandomForestClassifier(), 'Random Forest Classifier'),
          (GradientBoostingClassifier(), 'Gradient Boosting Classifier'),
          (svm.SVC(), 'SVC'),
          (MLPClassifier(), 'MLP Classifier')]
for model, name in models:
    # Do 5-fold cross-validation
    scores = cross val score(model, X train, y train, cv=5)
    # Fit and predict
    model.fit(X train, y train)
    y pred = model.predict(X test)
    # Calculate accuracy of the model
    accuracy cv = scores.mean()
    accuracy_test = accuracy_score(y_test, y_pred)
    # Print the accuracy performance
    print(f'{name}: Cross-validation accuracy: {accuracy cv:.3f}\n{name}: Test set accuracy:
{accuracy test:.3f}\n')
```

**Figure S25.** Another example of the output generated by the LLM *llama-3.1* based on the user's prompt for Task 1, showcasing a different choice of model and code.

```
import pandas as pd
from sklearn.model selection import train test split, cross val score
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.naive bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
# Load the training dataset
train data = pd.read excel("echem train.xlsx")
# Load the test dataset
test data = pd.read excel ("echem test.xlsx")
X train = train data['Canonical SMILES']
y train = train data['Reactivity']
X_test = test_data['Canonical SMILES']
y test = test data['Reactivity']
# Feature extraction for SMILES strings can be done using libraries like RDKit
# Here, we will simulate by using the length of SMILES string as a feature
X train simulated = X train.apply(len).values.reshape(-1, 1)
X test simulated = X test.apply(len).values.reshape(-1, 1)
models = {
   'RandomForestClassifier': RandomForestClassifier(),
    'SVC': SVC(),
    'GaussianNB': GaussianNB(),
    'LogisticRegression': LogisticRegression(),
    'KNeighborsClassifier': KNeighborsClassifier()
}
# For each model, perform 5-fold cross-validation and print the accuracy
for name, model in models.items():
    cv scores = cross val score(model, X train simulated, y train, cv=5)
    test scores = model.fit(X train simulated, y train).score(X test simulated, y test)
    print(f"{name}: Cross-validation accuracy: {cv scores.mean():.3f}")
    print(f"{name}: Test set accuracy: {test scores:.3f}")
```

**Figure S26.** An example of incorrect code output generated by the LLM *gpt-3.5-turbo* based on the user's prompt for Task 1. Although the code appears correct and executable, the resulting training yields low accuracy performance.

Gathering data I'm pulling together the training data from 'echem\_train.xlsx', focusing on compounds and their SMILES strings to estimate 'Reactivity'. This step forms the basis for subsequent model training.

Weighing model options Considering five ML models: Logistic Regression, Random Forest, SVM, KNN, and Gradient Boosting. Also looking at alternatives like Naive Bayes and Neural Networks. Planning to use 5-fold cross-validation for training and evaluating model performance.

**Preparing the tools** I'm thinking through the essentials: importing RDKit, pandas, numpy, and scikit-learn to compute fingerprints from SMILES strings.

Mapping the code I'm outlining the plan to import libraries, read data, compute fingerprints, convert them to numpy arrays, and define models for cross-validation, training, and testing with print statements for accuracies.

**Converting data** I'm translating SMILES strings into fingerprints, filtering out invalid ones, and isolating the features and labels for both training and test datasets. It's interesting to see the data transformation unfold.

**Preparing for scaling** I'm setting up the StandardScaler from sklearn to preprocess the data. Models like LogisticRegression, SVC, and KNeighborsClassifier benefit from this. Scaling is separately defined for LogisticRegression, SVC, and KNeighborsClassifier.

Assessing model performance I'm evaluating models with and without scaling. Using crossvalidation scores and test set accuracy, I aim to pinpoint which approach yields the best results.

Handling data I'm pulling data from Excel files, converting SMILES to fingerprints, standardizing features, and setting up models for cross-validation. Observing patterns improves model performance.

```
import pandas as pd
import numpy as np
from rdkit import Chem
from rdkit.Chem import AllChem, DataStructs
from sklearn.model_selection import cross val score, StratifiedKFold
from sklearn.metrics import accuracy score
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
# Read the training and test data
train df = pd.read excel('echem train.xlsx')
test df = pd.read excel('echem test.xlsx')
# Function to compute fingerprints from SMILES
def smiles to fp(smiles):
   mol = Chem.MolFromSmiles(smiles)
   if mol is not None:
        fp = AllChem.GetMorganFingerprintAsBitVect(mol, 2, nBits=2048)
        arr = np.zeros((1,))
        DataStructs.ConvertToNumpyArray(fp, arr)
```

```
return arr
    else:
        return None # Handle invalid SMILES
# Compute fingerprints for training data
train df['Fingerprint'] = train df['Canonical SMILES'].apply(smiles to fp)
train df = train df.dropna(subset=['Fingerprint'])
# Extract features and labels
X_train = np.array(list(train_df['Fingerprint']))
y train = train df['Reactivity'].values
# Compute fingerprints for test data
test df['Fingerprint'] = test df['Canonical SMILES'].apply(smiles to fp)
test_df = test_df.dropna(subset=['Fingerprint'])
X test = np.array(list(test df['Fingerprint']))
y test = test df['Reactivity'].values
# Initialize scaler
scaler = StandardScaler()
# Scale features
X train scaled = scaler.fit transform(X train)
X test scaled = scaler.transform(X test)
# Define models
models with scaling = {
    'LogisticRegression': LogisticRegression(max iter=1000),
    'SVC': SVC(),
    'KNeighborsClassifier': KNeighborsClassifier()
}
models without scaling = {
    'RandomForestClassifier': RandomForestClassifier(),
    'GradientBoostingClassifier': GradientBoostingClassifier()
}
# Cross-validation setup
skf = StratifiedKFold(n splits=5, shuffle=True, random state=42)
# Evaluate models with scaling
for name, model in models_with_scaling.items():
   cv scores = cross val score(model, X train scaled, y train, cv=skf, scoring='accuracy')
    cv accuracy = np.mean(cv scores)
   model.fit(X train scaled, y train)
   y pred = model.predict(X test scaled)
    test accuracy = accuracy score(y test, y pred)
    print(f"{name}: Cross-validation accuracy: {cv accuracy:.3f}")
    print(f"{name}: Test set accuracy: {test accuracy:.3f} \n")
# Evaluate models without scaling
for name, model in models without scaling.items():
    cv scores = cross val score(model, X train, y train, cv=skf, scoring='accuracy')
    cv_accuracy = np.mean(cv_scores)
    model.fit(X train, y train)
    y pred = model.predict(X test)
    test accuracy = accuracy_score(y_test, y_pred)
    print(f"{name}: Cross-validation accuracy: {cv accuracy:.3f}")
    print(f"{name}: Test set accuracy: {test_accuracy:.3f} \n")
```

**Figure S27.** An example of the reasoning and the output code generated by the LLM *OpenAI o1preview* based on the user's prompt for Task 1, showcasing the enhanced reasoning ability.

```
import re
import tempfile
import os
import shutil
import anthropic
import openai
import pandas as pd
import time
import traceback
import glob
import numpy as np
from IPython.utils.io import capture output
import subprocess
from skopt import Optimizer
from skopt.space import Real, Categorical, Integer
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import KFold, cross val score
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.linear model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy score
from rdkit import Chem
from rdkit.Chem import Descriptors
def generate and execute code (user prompts, model='gpt-3.5-turbo', num calls=100,
max reflection=3):
   results = []
    for in range(num calls):
        conversation history = []
        row data = {'Model': model}
        execution result, num reflections = 0, 0
        for i in range(len(user prompts)):
            if i > 0 and execution result != 1:
                row data[f'Generated Code {i+1}'] = 'N/A'
                row data[f'Execution Result {i+1}'] = 'N/A'
                row data[f'Response Time {i+1} (s)'] = 'N/A'
                row data[f'Number of Reflections {i+1}'] = 'N/A'
                continue
            prompt conversation history = conversation history.copy()
            response content, response time = chat(model, user prompts[i],
prompt conversation history)
            prompt conversation history.append({"role": "user", "content": user prompts[i]})
            prompt conversation history.append({"role": "assistant", "content":
response content})
            code_block = re.search(r'```python\n(.*?) \n```', response_content, re.DOTALL)
            if code block:
                code = code block.group(1)
            else:
                code = response content
               print (f"Warning: No code block found in the response for prompt {i+1}.
Attempting to execute the entire response.")
```

```
for j in range(max_reflection + 1):
                execution result, error message = run code (code)
                if execution result == 1:
                   break
                if j < max reflection:</pre>
                    reflection prompt = f"Please reflect on the code you previously wrote.
There is an error and I cannot run it on my Jupyter Notebook. The error message
is:\n{error message}\nPlease try to catch any bugs or failures to follow the user
instruction. In your answer, give me the full revised code. Do not just give the revised
part, but the whole code that can be directly copy and paste to run. Make sure you give full
code in the python code block, do not miss, comment or abbreviation anything."
                    print (reflection prompt)
                    prompt conversation history.append({"role": "user", "content":
reflection_prompt})
                    revised response content, revised response time = chat(model,
reflection prompt, prompt conversation history)
                    prompt conversation history.append({"role": "assistant", "content":
revised_response_content})
                    revised code block = re.search(r'``python\n(.*?)\n``',
revised response content, re.DOTALL)
                    if revised code block:
                        code = revised code block.group(1)
                    response time += revised response time
                    num reflections += 1
            row data[f'Generated Code {i+1}'] = code
            row data[f'Execution Result {i+1}'] = execution result
            row_data[f'Response Time {i+1} (s)'] = response_time
            row data[f'Number of Reflections {i+1}'] = num reflections
            conversation history = prompt conversation history.copy()
        row data['Conversation History'] = str(conversation history)
        results.append(row data)
    df = pd.DataFrame (results)
    df.to excel('Results '+model+'.xlsx', index=False)
    return df
def run code(code):
    with tempfile.NamedTemporaryFile(mode='w', delete=False, suffix='.py') as temp file:
        temp file.write(code)
        temp file path = temp file.name
    try:
        result = subprocess.run(['python', temp file path], capture output=True, text=True)
        os.unlink(temp file path)
        if result.returncode == 0:
           return 1, None
        else:
            return 0, result.stderr
    except Exception as e:
```

```
error_message = traceback.format_exc()
        os.unlink(temp_file_path)
        return 0, error message
def chat (model, user prompt, conversation history, max retries=3):
   retry count = 0
   pre prompt = "You are a helpful coding assistant who always writes detailed and
executable code without human implementation. Please ensure that you write the complete code
so I can copy and paste it directly into a Jupyter notebook to run. Please write all codes in
one code block; do not separate them by text explanations. When explanations are necessary,
include them as comments in the code. Make sure you use ```python to mark the start of the
python code."
    while retry_count < max_retries:</pre>
        try:
            start time = time.time()
            if model.startswith('claude'):
                client = anthropic.Anthropic(api_key=anthropic_key)
                response = client.messages.create(
                    model=model,
                    max tokens=4096,
                    messages=[
                        * conversation history,
                        {"role": "user", "content": user prompt}
                    1
                )
                response content = response.content[0].text
            elif model.startswith('gpt'):
                response = openai.ChatCompletion.create(
                    model=model,
                    messages=[
                        {"role": "system", "content": pre prompt},
                        *conversation history,
                        {"role": "user", "content": user prompt}
                    ]
                )
                response content = response.choices[0].message.content
            elif model.startswith('llama'):
                formatted history = ""
                for message in conversation history:
                    role = message["role"]
                    if role == "user":
                        formatted history += f"User: {message['content'] } \n"
                    elif role == "assistant":
                        formatted history += f"Assistant: {message['content']}\n"
                formatted prompt = f"{formatted history}User: {user prompt} \nAssistant:"
                if model == "llama-3":
                    model name = "meta/meta-llama-3-70b-instruct"
                response = replicate.run(
                    model name,
                    input={
                        "system prompt": pre prompt,
                        "max tokens": 4096,
                        "prompt": formatted prompt
                    }
                )
```

```
response_content = ''.join(response)
                print(response content)
            else:
                raise ValueError(f"Unknown model: {model}")
            end time = time.time()
            response time = end time - start time
            return response content, response time
        except Exception as e:
            retry count += 1
            if retry count == max retries:
                error message = f"Error occurred during code generation: {str(e)}"
                print(error message)
                return error message, 0
user prompts = ["""You are an expert in coding and you always write code .....
.....
]
```

**Figure S28.** An example of Python code used to automate the repetitive task of prompting an LLM 100 times and saving the output code. If the generated code is not executable, the error message is sent back to the LLM for correction. This logic is applied to all tasks in this study by simply changing the model choice and the user prompt. The complete history of all generated code can be found in Spreadsheet SF3. Auto Coding Dataset of the Supporting Information.

```
import pandas as pd
import re
import tempfile
import subprocess
import os
import traceback
import time
def process excel files(file names):
    summary data = []
    for file name in file names:
        # Check if the "acc " file already exists
        output file name = "acc " + file name
        if os.path.isfile(output file name):
            # If the file exists, read it instead of the original file
            df = pd.read excel (output file name)
        else:
            # If the file doesn't exist, read the original file
            df = pd.read excel(file name)
            # Create a new column "Performance" and initialize it with an empty string
            df["Performance"] = ""
            # Filter rows where "Execution Result 1" is 1
            filtered df = df[df["Execution Result 1"] == 1]
            total rows = len(df)
            print(f"Processing {file name} with {total rows} rows")
            # Iterate over the filtered rows
            for index, row in filtered df.iterrows():
                print(f"Processing row {index + 1}/{total rows}")
                code = row["Generated Code 1"]
                # Create a temporary file to write the code
                with tempfile.NamedTemporaryFile(mode='w', delete=False, suffix='.py') as
temp file:
                    temp file.write(code)
                    temp file path = temp file.name
                try:
                    # Execute the code using subprocess
                    result = subprocess.run(['python', temp file path], capture output=True,
text=True, timeout=100)
                    os.unlink(temp file path)
                    if result.returncode == 0:
                        # Code executed successfully
                        output = result.stdout.split('\n')
                        # Find the accuracy value from the captured output
                        accuracy values = []
                        for line in output:
```

```
match = re.search(r"accuracy:\s*(\d+(\.\d+)?)", line,
re.IGNORECASE)
                            if match:
                                accuracy values.append(float(match.group(1)))
                        # Update the "Performance" column with the highest accuracy value
                        if accuracy values:
                            df.at[index, "Performance"] = max(accuracy values)
                    else:
                        # Code execution encountered an error
                        print(f"Error executing code at index {index} in file {file name}:
{result.stderr}")
                except subprocess.TimeoutExpired:
                    print(f"Error: Code execution at index {index} in file {file name}
exceeded time limit and was terminated.")
                    os.unlink(temp file path)
                except Exception as e:
                    # Error occurred while executing the code
                    error message = traceback.format exc()
                    os.unlink(temp file path)
                    print(f"Error executing code at index {index} in file {file name}:
{error message}")
            # Save the updated DataFrame back to the Excel file
            df.to excel(output file name, index=False)
        # Generate summary data for the current file
        model name = file name.split(" ")[1]
        total rows = len(df)
        executed rows = len(df[df["Execution Result 1"] == 1])
        # Count rows where "Number of Reflections 1" is 0 within the filtered DataFrame
        one conversation rows = len(df[(df["Execution Result 1"] == 1) & (df["Number of
Reflections 1"] == 0)])
        # Handle empty or non-numeric values in the "Performance" column
        avg accuracy = pd.to numeric(df["Performance"], errors='coerce').mean()
        # Calculate the average time considering the "Number of Reflections 1"
        df["Adjusted Time"] = df["Response Time 1 (s)"] / (df["Number of Reflections 1"] + 1)
        avg time = df["Adjusted Time"].mean()
        correctness = len(df[pd.to numeric(df["Performance"], errors='coerce') > 0.85]) /
total rows #executed rows
        # Calculate the average code length
        code length = df["Generated Code 1"].apply(lambda x: len(str(x).split())).mean()
        summary row = {
           "Model Name": model name,
            "Avg Time": avg time,
            "Code Length": code length,
```

**Figure S29.** An example of Python code used to automate the evaluation of code generated by repeatedly prompting an LLM 100 times for Task 1. The generated code is analyzed and graded based on the performance of the trained machine learning models. The evaluation outcomes are merged and provided in Spreadsheet SF3 of the Supporting Information.

#### User:

You are an expert in coding and chemistry, and you are designing an experiment with a synthetic chemist to write an active machine learning program in Python to improve the yield of an electrochemical C-H bond oxidation. The same substrate compound will be subjected to different concentrations, mediator types and equivalents, and different electrolytes at 1 eq, with the option of adding a co-solvent. The product will be analyzed to determine if the specific C-H bond is oxidized to C-OH or C=O.

The search space for this optimization has been previously determined and is as follows:

Substrate Concentration: 0.025 M, 0.05 M, 0.075 M, 0.1 M, 0.125 M Mediator eq.: 0, 0.25, 0.5, 0.75, 1 Mediator type: NHPI, TCNHPI, QD, DABCO, TEMPO Electrolyte type: LiClO4, LiOTf, Bu4NClO4, Et4NBF4, Bu4NPF6 Co-solvent HFIP: 0, 1

Please write code to generate a CSV with the entire search space, where each row is a unique combination of synthesis parameters. The first 5 columns will be these parameters.

Once you generate the reaction scope, move to the next step: initialization. This step provides the chemist with 5 conditions to start with by randomly picking 5 conditions. To do this, create a new CSV file called "echem\_reaction\_opt\_round0.csv" with the same 5 columns. Then, add a column called "Yield" and assign "PENDING" to all conditions, as we have not yet started to run the actual reactions. The last column should be called "Priority", which indicates the past/next experiments. For completed experiments, the number is -1; for suggested experiments, the number is 1; and for the rest, it is 0. This CSV should have the exact same number of rows as the search space CSV. Since we have not yet started, the priority is 0 for all experiments at the beginning. Then, randomly select 5 initial reaction conditions and assign priority 1 to them. Finally, sort all rows based on priority, so that the "1" comes first, followed by 0. Please write all your code in one code block so that I can copy and paste it into a Jupyter Notebook to execute directly.

**Figure S30.** A representative user prompt used for tasking the LLM for the first part of Task 2. The prompt was crafted by a human using previously described prompt engineering strartegies.<sup>3,23</sup> In this case, the LLM is tasked with generating an initial search space.

#### User:

I have completed the first 5 reactions with yields "6, 5, 5, 8, 0". Please write the code to suggest the next 5 experiments. You need two functions:

1. update\_yield: Takes a CSV file, row number, and yield as input, updates the CSV by changing the pending yield for priority 1 conditions to observed yields.

2. reaction\_optimization: Takes a CSV file ("echem\_reaction\_opt\_roundxxx.csv"), ensures all priority 1 experiments are completed (with observed yields), changes their priority to -1, and suggests 5 new experiments using the Optimizer class from skopt with a Gaussian process and expected improvement. The new experiments should come from priority 0 rows. Create a new CSV with round n+1, update priorities, and sort by priority.

Please write the code in one block and import the necessary modules.experiments, the number is -1; for suggested experiments, the number is 1; and for the rest, it is 0. This CSV should have the exact same number of rows as the search space CSV. Since we have not yet started, the priority is 0 for all experiments at the beginning. Then, randomly select 5 initial reaction conditions and assign priority 1 to them. Finally, sort all rows based on priority, so that the "1" comes first, followed by 0. Please write all your code in one code block so that I can copy and paste it into a Jupyter Notebook to execute directly.

**Figure S31.** A representative user prompt used for tasking the LLM for the second part of Task 2. Once the LLM-generated code for the first part is executable, this prompt is sent to the LLM as a follow-up request, tasking the LLM with starting from the previously written code and files and making suggestions on the synthesis conditions.

```
import pandas as pd
import itertools
import random
# Define the parameter space
substrate concentration = [0.025, 0.05, 0.075, 0.1, 0.125]
mediator eq = [0, 0.25, 0.5, 0.75, 1]
mediator_type = ['NHPI', 'TCNHPI', 'QD', 'DABCO', 'TEMPO']
electrolyte type = ['LiClO4', 'LiOTf', 'Bu4NClO4', 'Et4NBF4', 'Bu4NPF6']
co solvent HFIP = [0, 1]
# Generate the search space
search space = list(itertools.product(substrate concentration, mediator eq, mediator type,
electrolyte type, co solvent HFIP))
search space df = pd.DataFrame (search space, columns=["Substrate Concentration", "Mediator
eq.", "Mediator type", "Electrolyte type", "Co solvent HFIP"])
# Save the entire search space to a CSV
search space df.to csv("echem reaction search space.csv", index=False)
# Initialize the optimization round
search space df["Yield"] = "PENDING"
search space df["Priority"] = 0
# Randomly select 5 initial conditions
initial conditions indices = random.sample(range(len(search space df)), 5)
# Update priority for the initial conditions
search space df.loc[initial conditions indices, "Priority"] = 1
# Sort by Priority to move the initial conditions at the top
search space df = search space df.sort values(by="Priority",
ascending=False).reset index(drop=True)
# Save the initialization to a CSV
search space df.to csv("echem reaction opt round0.csv", index=False)
# Display the first few rows of both generated DataFrames
print("Search Space:")
print(search space df.head())
print("\nInitial Optimization Round:")
opt round df = search space df.query("Priority == 1")
print(opt round df)
```

**Figure S32.** A representative example of the output generated by the LLM based on the user's prompt for the first part of Task 2. The task is repeated for all different base models 100 times, and the full conversation, along with all generated code is provided in Spreadsheet SF3. Auto Coding Dataset in the Supporting Information.

```
import pandas as pd
from skopt import Optimizer
from skopt.space import Real, Categorical, Integer
# Function to update yield
def update yield (csv file, row num, yield value):
   df = pd.read csv(csv file)
   df.loc[row num, 'Yield'] = yield value
   df.to csv(csv file, index=False)
# Function to suggest the next 5 experiments
def reaction optimization (csv file):
   df = pd.read csv(csv file)
    # Check all priority 1 experiments have yields and update their status to -1
    assert df.loc[df['Priority'] == 1, 'Yield'].apply(lambda x: x != 'PENDING').all(), "Not
all priority 1 experiments have yields."
   df.loc[df['Priority'] == 1, 'Priority'] = -1
    # Prepare data for the optimizer
   X = df.loc[df['Priority'] == -1, ['Substrate Concentration', 'Mediator eq.', 'Mediator
type', 'Electrolyte type', 'Co solvent HFIP']].values
   y = df.loc[df['Priority'] == -1, 'Yield'].astype(float).values
    # Initialize the optimizer
    opt = Optimizer(dimensions=space, base estimator='GP', acq func='EI')
    # Tell the optimizer the observed data
    opt.tell(X.tolist(), y.tolist())
    # Ask for the next 5 points
    new points = opt.ask(n points=5)
    # Convert the new points into a DataFrame
    new points df = pd.DataFrame(new points, columns=['Substrate Concentration', 'Mediator
eq.', 'Mediator type', 'Electrolyte type', 'Co solvent HFIP'])
    # Ensure suggested experiments are unique and not already completed
    for index, row in new points df.iterrows():
        matched row = df[(df['Substrate Concentration'] == row['Substrate Concentration']) &
                         (df['Mediator eq.'] == row['Mediator eq.']) &
                         (df['Mediator type'] == row['Mediator type']) &
                         (df['Electrolyte type'] == row['Electrolyte type']) &
                         (df['Co solvent HFIP'] == row['Co solvent HFIP'])]
        assert matched row.empty or matched row['Priority'].iloc[0] != -1, "Suggested
experiment overlaps with completed one."
    # Update the original DataFrame with new priorities
    for index, row in new points df.iterrows():
        df.loc[(df['Substrate Concentration'] == row['Substrate Concentration']) &
               (df['Mediator eq.'] == row['Mediator eq.']) &
               (df['Mediator type'] == row['Mediator type']) &
               (df['Electrolyte type'] == row['Electrolyte type']) &
```

```
(df['Co solvent HFIP'] == row['Co solvent HFIP']), 'Priority'] = 1
# Generate the new round CSV file name
round_number = int(csv_file.split('_')[-1].split('.')[0].replace('round', ''))
new_csv_file = f"echem_reaction_opt_round{round_number + 1}.csv"
# Sort by priority and save the new DataFrame
df = df.sort_values(by='Priority', ascending=False)
df.to_csv(new_csv_file, index=False)
# Update the yield for the initial experiments
update_yield("echem_reaction_opt_round0.csv", 0, 6)
update_yield("echem_reaction_opt_round0.csv", 3, 8)
update_yield("echem_reaction_opt_round0.csv", 4, 0)
# Suggest the next 5 experiments
reaction_opt_mization("echem_reaction_opt_round0.csv")
```

**Figure S33.** A representative example of the output generated by the LLM based on the user's prompt for the second part of Task 2.

```
def check csv conditions(file names):
    for file name in file names:
        try:
            df = pd.read csv(file name)
            # Convert all column names to lower case to handle case insensitivity
            df.columns = [col.lower() for col in df.columns]
            if "priority" not in df.columns or "yield" not in df.columns:
                print (f"{file name} does not pass: 'priority' or 'yield' column missing")
                continue # Skip to the next file
            # Convert 'yield' values to string for a consistent comparison across types
           negative priority yields = df[df['priority'] == -1]['yield'].astype(str).tolist()
           valid yields = ['6', '5', '8', '0']
            if not all(yield value in valid yields for yield value in
negative priority yields):
                print (f"{file name} does not pass: Incorrect 'yield' values for priority -1")
                continue # Skip to the next file
            # Check the first five rows for columns where priority is 1
           priority one rows = df[df['priority'] == 1].head(5)
            # Define acceptable values for each column as floats
            substrate concentrations = [0.025, 0.05, 0.075, 0.1, 0.125]
           mediator eqs = [0, 0.25, 0.5, 0.75, 1]
           mediator types = ["NHPI", "TCNHPI", "QD", "DABCO", "TEMPO"]
           electrolyte types = ["LiClO4", "LiOTf", "Bu4NClO4", "Et4NBF4", "Bu4NPF6"]
           co solvents = [0, 1]
            # Prepare to check 'yield' column for all rows where priority is 1
            yield values = df[df['priority'] == 1]['yield'].astype(str).tolist()
            if not all(yield_value == 'PENDING' for yield_value in yield_values):
                print(f"{file name} does not pass: 'yield' values for priority 1 are not
'PENDING'")
                continue # Skip to the next file
            # Iterating through each row to check conditions
            for , row in priority one rows.iterrows():
                # Convert numeric column values to float for comparison
                try:
                   row concentration = float(row[0])
                   row eqs = float(row[1])
                   row co solvent = float(row[4])
                except ValueError:
                    print(f"{file name} does not pass: Numeric conversion error in data.
row concentration = {row[0]}; row eqs = {row[1]}; row co solvent = {row[4]}")
                    continue
                # Check each column against its respective allowed values
                if row concentration not in substrate concentrations:
                    print(f"{file name} does not pass: Incorrect substrate concentration
{row[0]} in the first column")
                   continue # Skip to the next file
```

```
if row eqs not in mediator eqs:
                   print(f"{file name} does not pass: Incorrect mediator equivalents
{row[1]} in the second column")
                   continue # Skip to the next file
               if row[2] not in mediator types:
                   print(f"{file name} does not pass: Incorrect mediator type {row[2]} in
the third column")
                   continue # Skip to the next file
               if row[3] not in electrolyte types:
                   print(f"{file name} does not pass: Incorrect electrolyte type {row[3]} in
the fourth column")
                   continue # Skip to the next file
               if row co solvent not in co solvents:
                   print(f"{file name} does not pass: Incorrect co-solvent {row[4]} in the
fifth column")
                   continue # Skip to the next file
           return True # Return True immediately upon successful validation of a file
        except Exception as e:
           print(f"Error processing file {file name}: {e}")
   return False # If no files pass the test, return False
```

**Figure S34.** An example of Python code used to automate the evaluation of code generated by repeatedly prompting an LLM 100 times for Task 2. The generated code is analyzed and graded based on the content of the csv files generated from the code given by LLMs. The evaluation outcomes, similar to those in Task 1, are merged into SF3. Auto Coding Dataset via function *process\_excel\_files* described previously, and provided in the Supporting Information.

User: Below is an example for the EDBO package (EDBO+: Bayesian reaction optimization as a tool for chemical synthesis) from https://github.com/doyle-lab-ucla/edboplus. Please carefully read it: from edbo.plus.optimizer\_botorch import EDBOplus #### This tutorial covers the basics of running EDBO+: from designing a combinatorial space to running the Bayesian Optimizer. ## 1. Creating a search scope using EDBO+. ##### Again the samples suggested by EDBO+ have priority = +1. In addition, we assign priority = -1 to the experiments that we have already run (these are at the bottom of the dataset). Please read the above example and help me write code for the optimization of my reaction yield. Below are the possible choices of synthesis parameters: Substrate Concentration: 0.025 M, 0.05 M, 0.075 M, 0.1 M, 0.125 M Mediator eq.: 0, 0.25, 0.5, 0.75, 1 Mediator type: NHPI, TCNHPI, QD, DABCO, TEMPO Electrolyte type: LiCIO4, LiOTf, Bu4NCIO4, Et4NBF4, Bu4NPF6 Co-solvent HFIP: 0, 1 First, the code should suggest the 5 initial experiments I should start with, and I only want to maximize the reaction yield. If possible, make sure you write all your code in one code block so that I can copy and paste it into Jupyter Notebook to execute directly. You do not need to have further explanation on the code or other instructions.

**Figure S35.** A representative user prompt used for tasking the LLM for the first part of Task 3. The prompt was crafted by a human using previously described prompt engineering strartegies.<sup>3,23</sup> The LLM is tasked with generating an initial search space with EDBO<sup>22</sup> by reading its documentation. The abbreviated portion (...) includes content from the tutorial examples provided by the authors of the EDBO package (https://github.com/doyle-lab-ucla/edboplus).

```
User:
```

```
Now I have completed the first 5 reactions in the lab. The observed yields are "6, 5, 5, 8, 0" for suggested reactions 1 to 5. Their conditions are:

....
0.1 0.5 TCNHPI Bu4NPF6 0
0.075 0.5 QD Et4NBF4 1
0.075 0.5 QD LiOTf 1
0.075 0.5 NHPI LiCIO4 1
0.05 0.5 TEMPO Et4NBF4 0
....
Please write the code to help me suggest the next 5 experiments. Make sure you import related modules and write the code in one code block.
```

**Figure S36.** A representative user prompt for tasking the LLM in the second part of Task 2. After verifying that the LLM-generated code for the first part is executable, this follow-up prompt requests the LLM to build upon the existing code and files, providing suggestions on the synthesis conditions.

```
# Import necessary libraries
from edbo.plus.optimizer botorch import EDBOplus
import pandas as pd
# Define the reaction components for the search space
reaction components = {
    'substrate concentration': [0.025, 0.05, 0.075, 0.1, 0.125],
    'mediator eq': [0, 0.25, 0.5, 0.75, 1],
    'mediator type': ['NHPI', 'TCNHPI', 'QD', 'DABCO', 'TEMPO'],
    'electrolyte type': ['LiClO4', 'LiOTf', 'Bu4NClO4', 'Et4NBF4', 'Bu4NPF6'],
    'co solvent HFIP': [0, 1]
}
# Generate the reaction scope and save it to a CSV file
EDBOplus().generate reaction scope(
   components=reaction components,
   filename='optimization scope.csv',
    check overwrite=False
)
# Load the generated reaction scope using Pandas
df scope = pd.read csv('optimization scope.csv')
# Print the number of combinations in the reaction scope
n combinations = len(df scope)
print(f"Your reaction scope has {n combinations} combinations.")
# Run EDBO+ to suggest initial experiments
EDBOplus().run(
   filename='optimization scope.csv', # Previously generated scope
    objectives=['yield'], # Objective to be optimized
    objective_mode=['max'], # Maximize reaction yield
   batch=5, # Number of experiments in parallel that we want to perform in this round
    columns_features='all', # Features to be included in the model
    init sampling method='cvtsampling' # Initialization method
)
# Load and display the first few experiments from the updated CSV file
df edbo = pd.read_csv('optimization_scope.csv')
print(df edbo.head(10))
```

**Figure S37.** A representative example of the output generated by the LLM based on the user's prompt for the first part of Task 3. The task is repeated for all different base models 100 times, and the full conversation, along with all generated code is provided in Spreadsheet SF3. Auto Coding Dataset in the Supporting Information.

```
# Import necessary libraries
from edbo.plus.optimizer botorch import EDBOplus
import pandas as pd
# Load the created reaction scope file
df edbo = pd.read csv('my optimization yield.csv')
# Fill in the observed yields for the first 5 reactions
df edbo.loc[df edbo.index[0], 'yield'] = 6
df edbo.loc[df edbo.index[1], 'yield'] = 5
df edbo.loc[df edbo.index[2], 'yield'] = 5
df edbo.loc[df edbo.index[3], 'yield'] = 8
df edbo.loc[df edbo.index[4], 'yield'] = 0
# Save the updated data with observations to a new CSV file
df edbo.to csv('my optimization round1.csv', index=False)
# Check the updated data
print(df edbo.head(10))
# Run EDBO+ again with the updated data to suggest the next experiments
EDBOplus().run(
    filename='my optimization roundl.csv', # Previous scope (including observations).
    objectives=['yield'], # Objective to be optimized.
    objective mode=['max'], # Maximize yield.
   batch=5, # Number of experiments in parallel to perform in this round.
    columns features='all', # Include all features in the model.
    init sampling method='cvtsampling' # Initialization method.
)
# Load and display the suggested experiments
df_edbo_next = pd.read_csv('my_optimization_round1.csv')
print(df_edbo_next.head(10)) # Display the first 10 rows to check the high priority
experiments
```

**Figure S38.** A representative example of the output generated by the LLM based on the user's prompt for the second part of Task 3.

```
def check csv conditions(file names):
    for file_name in file_names:
        try:
           df = pd.read csv(file name)
            # Convert all column names to lower case to handle case insensitivity
           df.columns = [col.lower() for col in df.columns]
           if "priority" not in df.columns or "yield" not in df.columns:
                print(f"{file name} does not pass: 'priority' or 'yield' column missing")
                continue # Skip to the next file
            # Convert 'yield' values to string for a consistent comparison across types
           negative priority yields = df[df['priority'] == -1]['yield'].astype(str).tolist()
           valid yields = ['6', '5', '8', '0']
           if not all(yield value in valid yields for yield value in
negative priority yields):
                print(f"{file_name} does not pass: Incorrect 'yield' values for priority -1.
{negative priority yields }")
                continue # Skip to the next file
            # Define acceptable values for each column as floats
            substrate concentrations = [0.025, 0.05, 0.075, 0.1, 0.125]
           mediator eqs = [0, 0.25, 0.5, 0.75, 1]
           mediator types = ["NHPI", "TCNHPI", "QD", "DABCO", "TEMPO"]
            electrolyte types = ["LiClO4", "LiOTf", "Bu4NClO4", "Et4NBF4", "Bu4NPF6"]
           co solvents = [0, 1]
            # Check the first five rows for columns where priority is 1
           priority one rows = df[df['priority'] == 1].head(5)
            # Iterating through each row to check conditions
            for , row in priority one rows.iterrows():
                # Convert numeric column values to float for comparison
                try:
                    row concentration = float(row[0])
                   row eqs = float(row[1])
                   row co solvent = float(row[4])
                except ValueError:
                   print(f"{file name} does not pass: Numeric conversion error in data.
row concentration = {row[0]}; row eqs = {row[1]}; row co solvent = {row[4]}")
                    continue
                # Check each column against its respective allowed values
                if row concentration not in substrate concentrations:
                   print(f"{file name} does not pass: Incorrect substrate concentration
{row[0]} in the first column")
                    continue # Skip to the next file
                if row eqs not in mediator eqs:
                   print(f"{file name} does not pass: Incorrect mediator equivalents
{row[1]} in the second column")
                   continue # Skip to the next file
                if row[2] not in mediator types:
                   print(f"{file name} does not pass: Incorrect mediator type {row[2]} in
```

**Figure S39.** An example of Python code used to automate the evaluation of code generated by repeatedly prompting an LLM 100 times for Task 3. The generated code is analyzed and graded based on the content of the csv files generated from the code given by LLMs. The evaluation outcomes, similar to those in Task 1 and Task 2, are merged Spreadsheet SF3. Auto Coding Dataset via function *process\_excel\_files* described previously, and provided in the Supporting Information.

## Section S6. Optimization of Synthesis Conditions

Prior to initiating the optimization of synthesis conditions, it is essential to define the search space systematically. To achieve this, we advocate a hybrid search approach. Initially, we instructed an LLM assistant to read and analyze a series of relevant literature identified in Section S2. This analysis allowed the extraction of synthesis conditions, which are detailed in Spreadsheet SF4. EChem Reaction Optimization Dataset in the Supporting Information. The LLM was then employed to propose hypotheses by considering a set of plausible configurations or ranges for the search space. Through this process, the following parameters were identified as critical for optimization of the electrochemical reaction relevant to this study: (1) concentration of the substrate, (2) type of redox mediator, (3) concentration of the mediator, (4) type of electrolyte, (5) additives in the solvent, (6) type of base, (7) electrode material, (8) reaction time, (9) current density, and (10) reaction atmosphere. This knowledge is primarily derived from summarizing the discussion sections of previous literature and examining preliminary trial-and-error efforts reported in the previous literature for optimizing reaction yields.

For the purposes of this study, and to align with the design of our reactor, we focused on the first five parameters as primary variables, hypothesizing that each individually influences the yield. Additionally, certain combinations between the two parameters could also be significant. To ensure consistency, other variables, such as voltage and reaction time, were held constant. Thus, we selected parameters 1–5 as the dimensions defining our reaction space.

Each of these parameters offers numerous possible choices. To manage the complexity and practicality of the experiments, we pruned the reaction space to a feasible number of experiments. Specifically, we considered five concentrations for the substrate, five types of mediators with five different equivalents, five types of electrolytes, and two different solvent systems. This selection captures the reaction space defined by availability and amenability, resulting in a total of 1,250 combinations of reaction conditions for each substrate (Table S15). Note that for the consistency of this study, all 8 substrates (Table S16) are assigned the same search space, resulting in a possible 10,000 electrochemical C-H oxidation reactions that can be potentially examined, and the optimization process is independent for each substrate, leading to optimal synthesis conditions for each of them rather than having a condition that fits all.

Synthesis Paramter	Choice			
Concentration (mM)	25, 50, 75, 100, 125			
Electrocatalyst	NHPI, TCNHPI, Quinuclidine, DABCO, TEMPO			
Equivalence of Electrocatalyst	0, 0.25, 0.5, 0.75, 1			
Electrolyte	LiClO <sub>4</sub> , LiOTf , Bu <sub>4</sub> NClO <sub>4</sub> , Et <sub>4</sub> NBF <sub>4</sub> , Bu <sub>4</sub> NPF <sub>6</sub>			
Solvent	ACN, ACN/HFIP			

Table S15. Choice of each synthesis parameter.

No.	Substrate	CAS Number	SMILES String
1	(+)-a-Pinene	7785-70-8	CC1=CCC2CC1C2(C)C
2	1,4-Dioxaspiro[4.5]dec-6-ene 4683-24-3 C1CC2(C=CC1=O)OCCO2		C1CC2(C=CC1=0)OCCO2
3	alpha-Ionone	127-41-3	CC1=CCCC(C1C=CC(=O)C)(C)C
4	Valencene	4630-07-3	CC1CCC=C2C1(CC(CC2)C(=C)C)C
5	Theaspirane	36431-72-8	CC1=CCCC(C)(C)C12CCC(C)O2
6	Pregnenolone acetate	1778-02-5	CC(=0)OC1CCC2(C)C(=CCC3C2CCC2 (C)C(C(C)=0)CCC32)C1
7	(-)-Ambroxide	6790-58-5	CC1(CCCC2(C1CCC3(C2CCO3)C)C)C
8	Valerophenone	1009-14-9	CCCCCC(=0)C1=CC=CC=C1

**Table S16.** Eight compounds chosen for synthesis parameter optimization.

## Reaction setup

The electrolysis was conducted using a 24-well batch reactor equipped with graphite or RVC anodes and nickel cathodes (Table S1 and Figure S1). Each vial contained two pairs of anodes and cathodes (a total of four electrodes) with interelectrode distances of 3.6 mm and 8.13 mm, respectively. The solution containing the substrate (Table S16, 0.025 mM – 0.125 mM), mediator (0 - 1 eq.), electrolyte (1 eq.), and pyridine (1 eq.) was prepared by mixing the stock solution of each individual component accordingly. The typical stock solution needed for each reaction). The five mediators ranged from 0.1 to 0.3M (typically 0.5 to 2 mL), and the five electrolytes and pyridine were dissolved together in acetonitrile to achieve a concentration of 0.8M (typically 100  $\mu$ L to 500  $\mu$ L). Additional acetonitrile, with or without 10 eq. HFIP, was added to make the total volume 4 mL. No precautions were taken to exclude air or moisture, and the reactions were run at room temperature. The voltage was set to 3.5 V, resulting in a current of 1 mA for every 5 mM concentration of reaction (typically starting current is around 200 mA for 24 vials running together, each approximately 50 mM). Each vial contained a magnetic stir bar, and the reactions were stirred at 600 rpm for 12 hours.

## Reaction analysis

The resulting solution was evaporated by blowing air through it for 2 hours, and the residue was treated with 0.7 mL CDCl<sub>3</sub> and subjected to <sup>1</sup>H NMR analysis using 1,3,5-trimethoxybenzene (420.5 mg in 100 mL of acetonitrile-d<sub>3</sub>, 25  $\mu$ mol/mL) added to each well as the internal standard. Both proton and carbon NMR analyses were conducted. For purification, typically four vials were run under the same optimized conditions (Table S18), and the products were combined. After removing the solvent, the residue was redissolved in a small amount of CH<sub>2</sub>Cl<sub>2</sub> and passed through a short silica-gel plug with ethyl acetate to remove the electrolyte. The solution was concentrated, and the crude material was chromatographed through silica gel, eluting with ethyl acetate/hexanes to obtain the product.

**Table S17.** Summary of synthesis parameters under different reaction conditions. The best two conditions (A and B) for compounds 1 to 8 are denoted as IA, IB, IIA, IIB, etc. For most compounds, the top yield conditions can be similar, although variations are observed across different compounds and conditions. Figure 4 primarily reflects the results based on set A conditions. Detailed yield data can be found in Tables S18–26.

Procedure	Concentration (M)	Mediator eq.	Mediator	Electrolyte	HFIP
IA	0.025	1	TCNHPI	LiOTf	1
IB	0.025	1	TCNHPI	Et <sub>4</sub> NBF <sub>4</sub>	1
IIA	0.125	1	NHPI	Bu <sub>4</sub> NPF <sub>6</sub>	0
IIB	0.125	0.75	NHPI	Bu4NClO4	1
IIIA	0.025	1	NHPI	LiClO <sub>4</sub>	1
IIIB	0.025	1	TCNHPI	LiClO <sub>4</sub>	1
IVA	0.05	1	NHPI	LiOTf	0
IVB	0.025	1	TCNHPI	LiOTf	0
VA	0.025	1	NHPI	Et <sub>4</sub> NBF <sub>4</sub>	1
VB	0.025	1	NHPI	LiOTf	1
VIA	0.025	0.25	NHPI	Bu <sub>4</sub> NPF <sub>6</sub>	1
VIB	0.025	0.75	QD	LiClO <sub>4</sub>	0
VIIA	0.125	0.5	QD	Bu4NClO4	1
VIIB	0.125	0.5	QD	Et <sub>4</sub> NBF <sub>4</sub>	1
VIIIA	0.05	1	QD	Bu4NClO4	1
VIIIB	0.125	1	QD	LiOTf	1



**Figure S40.** Illustration of reaction space and yield from 455 experiments conducted using the 4 approaches in this study. For the search space on this study, 5 different synthesis parameters were chosen: concentration of the substrates and electro-catalysts, choice of electrocatalyst and electrolyte, and solvent, resulting in 1250 different combinations, while reaction time, voltage, and electrodes were held constant.
Characterization data for the electrolysis products



**Verbenone (1b).** The title compound was obtained using the general procedure for synthesis optimization, and the synthesis conditions and reaction yields are summarized in Table S18. The desired product was obtained as a clear oil. Spectral data matched the values reported in the literature.<sup>24</sup>

<sup>1</sup>H NMR (500 MHz, Chloroform-*d*): δ 5.64 (s, 1H), 2.73 (dt, *J* = 9.2, 5.5 Hz, 1H), 2.56 (td, *J* = 5.9, 1.7 Hz, 1H), 2.35 (td, *J* = 6.2, 1.5 Hz, 1H), 1.99 (d, *J* = 9.2 Hz, 1H), 1.94 (s, 3H), 1.42 (s, 3H), 0.93 (s, 3H);

<sup>13</sup>C NMR (151 MHz, Chloroform-*d*): δ 203.9, 170.1, 121.1, 57.4, 53.9, 49.6, 40.8, 26.5, 23.5, 22.0.



**1,4-Dioxaspiro[4.5]dec-6-en-8-one (2b).** The title compound was obtained using the general procedure for synthesis optimization, and the synthesis conditions and reaction yields are summarized in Table S19. The desired product was obtained as a clear oil. Spectral data matched the values reported in the literature.<sup>25</sup>

<sup>1</sup>H NMR (500 MHz, Chloroform-*d*): δ 6.62 (d, J = 10.2 Hz, 1H), 6.01 (d, J = 10.2 Hz, 1H), 4.10 – 3.99 (m, 4H), 2.64 (t, J = 6.6 Hz, 2H), 2.21 (t, J = 6.5 Hz, 2H); <sup>13</sup>C NMR (151 MHz, Chloroform-*d*): δ 198.8, 146.4, 130.6, 65.1, 35.3, 32.9.



**3,5,5-Trimethyl-4-(3-oxobut-1-en-1-yl)cyclohex-2-en-1-one** (**3b**). The title compound was obtained using the general procedure for synthesis optimization, and the synthesis conditions and reaction yields are summarized in Table S20. The desired product was obtained as a colorless oil. Spectral data matched the values reported in the literature.<sup>26,27</sup>

<sup>1</sup>H NMR (500 MHz, Chloroform-*d*):  $\delta$  6.68 (dd, J = 15.8, 9.5 Hz, 1H), 6.19 (d, J = 15.8 Hz, 1H), 5.99 (s, 1H), 2.72 (d, J = 9.5 Hz, 1H), 2.37 (d, J = 16.9 Hz, 1H), 2.29 (s, 3H), 2.16 (d, J = 16.9 Hz, 1H), 1.90 (s, 3H), 1.08 (s, 3H), 1.01 (s, 3H);

<sup>13</sup>C NMR (151 MHz, Chloroform-*d* ): δ 198.3, 197.6, 159.2, 143.6, 133.7, 126.9, 55.4, 47.3, 36.7, 27.9, 27.6, 27.3, 23.5.



**Nootkatone** (4b). The title compound was obtained using the general procedure for synthesis optimization, and the synthesis conditions and reaction yields are summarized in Table S21. The desired product was obtained as a yellow liquid. Spectral data matched the values reported in the literature.<sup>24</sup>

<sup>1</sup>H NMR (500 MHz, Chloroform-*d*):  $\delta$  5.76 (s, 1H), 4.76 – 4.71 (m, 2H), 2.56 – 2.46 (m, 1H), 2.41 – 2.18 (m, 4H), 2.06 – 1.94 (m, 3H), 1.71 (s, 3H), 1.41 – 1.28 (m, 1H), 1.11 (s, 3H), 1.40 – 1.35 (m, 2H), 0.95 (d, *J* = 6.8 Hz, 3H);

<sup>13</sup>C NMR (151 MHz, Chloroform-*d* ): δ 199.6, 170.5, 149.0, 124.7, 109.3, 43.9, 42.1, 40.5, 40.3, 39.3, 33.0, 31.6, 20.8, 16.8, 14.9.



**Theaspirone (5b).** The title compound was obtained using the general procedure for synthesis optimization, and the synthesis conditions and reaction yields are summarized in Table S22. The desired product was obtained as a colorless oil. Spectral data matched the values reported in the literature<sup>24</sup>.

<sup>1</sup>H NMR (500 MHz, Chloroform-*d*):  $\delta$ .5.75 (t, J = 1.4 Hz, 1H), 2.24 – 2.15 (m, 1H), 2.40 (d, J = 17.0 Hz, 1H), 2.32 (dd, J = 13.7, 8.4 Hz, 1H), 2.20 (d, J = 17.1 Hz, 1H), 2.18 – 2.10 (m, 1H), 2.05 – 1.99 (m, 1H), 1.97 (d, J = 9.1 Hz, 3H), 1.84 – 1.76 (m, 1 H), 1.54 – 1.45 (m, 1H), 1.30 (d, J = 6.0 Hz, 3H), 1.07 – 0.97 (m, 6H),;

<sup>13</sup>C NMR (151 MHz, Chloroform-*d*): δ 198.4, 168.4, 124.9, 88.5, 77.7, 50.2, 40.8, 34.3, 32.7, 24.4, 23.0, 20.5, 19.0.



**7,20-Dioxopregn-5-en-3-beta-yl acetate (6b).** The title compound was obtained using the general procedure for synthesis optimization, and the synthesis conditions and reaction yields are summarized in Table S23. The desired product was obtained as a white solid. Spectral data matched the values reported in the literature.<sup>24,28</sup>

<sup>1</sup>H NMR (500 MHz, Chloroform-*d*):  $\delta$  5.75 (s, 1H), 4.72 (tt, *J* = 11.6, 4.6 Hz, 1H), 2.73 (ddd, *J* = 13.7, 8.9, 5.6 Hz, 1H), 2.62 – 2.56 (m, 1H), 2.52 – 2.37 (m, 3H), 2.17 – 2.08 (m, 1H), 2.05 (s, 3H), 2.02 – 1.24 (m, 18H), 0.89 (s, 3H);

<sup>13</sup>C NMR (151 MHz, Chloroform-*d* ): δ 209.8, 201.2, 170.3, 126.5, 72.1, 62.3, 50.0, 45.2, 44.4, 38.4, 37.6, 36.0, 31.6, 27.3, 26.5, 23.6, 21.2, 17.3, 13.3.



**Norambreinolide (7b).** The title compound was obtained using the general procedure for synthesis optimization, and the synthesis conditions and reaction yields are summarized in Table S24. The desired product was obtained as a white solid. Spectral data Spectroscopic properties were identical to a commercial sample.

<sup>1</sup>H NMR (500 MHz, Chloroform-*d*):  $\delta$  2.41 (dd, J = 16.3, 14.7 Hz, 1H), 2.23 (dd, J = 16.2, 6.5 Hz, 1H), 2.08 (dt, J = 11.9, 3.4 Hz, 1H), 1.97 (dd, J = 14.8, 6.5 Hz, 1H), 1.88 (dq, J = 14.3, 3.5 Hz, 1H), 1.72 – 1.61 (m, 2H), 1.48 – 1.36 (m, 3H), 1.33 (s, 3H), 1.27 – 1.14 (m, 2H), 1.08 – 1.01 (m, 2H), 0.92 (s, 3H), 0.89 (s, 3H), 0.84 (s, 3H);

<sup>13</sup>C NMR (151 MHz, Chloroform-*d*): δ 176.9, 86.3, 59.0, 56.5, 42.3, 39.5, 38.7, 36.0, 33.2, 33.1, 28.7, 21.6, 20.9, 20.6, 18.1, 15.1.



**1-Phenyl-1,4-pentanedione (8b).** The title compound was obtained using the general procedure for synthesis optimization, and the synthesis conditions and reaction yields are summarized in Table S25. The desired product was obtained as a clear oil. Spectral data matched the values reported in the literature.<sup>29</sup>

<sup>1</sup>H NMR (500 MHz, Chloroform-*d*): δ 7.99 – 7.94 (m, 2H), 7.58 – 7.52 (m, 1H), 7.48 – 7.40 (m, 1H), 3.31 - 3.22 (m, 2H), 2.90 - 2.83 (m, 2H), 2.25 (s, 3H); <sup>13</sup>C NMR (151 MHz, Chloroform-*d*): δ 207.3, 198.5, 136.6, 133.4, 128.3, 37.0, 32.4, 30.1.

		Active Lea	rning				I	Random Samp	ling		
Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield	Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield
0.075	0.5	QD	Et4NBF4	1	5	0.1	0.25	QD	Bu4NClO4	0	0
0.075	0.5	NHPI	LiClO4	1	8	0.125	0.25	DABCO	Et4NBF4	1	0
0.1	0.5	TCNHPI	Bu4NPF6	0	6	0.1	0.25	DABCO	Et4NBF4	1	0
0.05	0.5	TEMPO	LiClO4	0	0	0.05	0.25	TCNHPI	LiClO4	0	7
0.125	0.5	NHPI	Et4NBF4	1	5	0.1	0.25	NHPI	Bu4NPF6	1	12
0.125	0	NHPI	LiOTf	1	0	0.1	0.5	TEMPO	Bu4NPF6	0	0
0.025	1	NHPI	Bu4NPF6	1	5	0.1	0.5	DABCO	LiOTf	1	0
0.025	0.75	TCNHPI	Et4NBF4	1	55	0.1	0.75	DABCO	LiOTf	1	0
0.1	0.25	TEMPO	Bu4NClO4	0	0	0.05	0.0	DABCO	LiClO4	0	0
0.05	1	TCNHPI	LiClO4	1	37	0.1	1.0	TEMPO	LiOTf	1	0
0.025	0	TCNHPI	Et4NBF4	1	0	0.1	0.75	NHPI	LiClO4	1	5
0.025	0	TCNHPI	Bu4NClO4	1	0	0.05	0.25	NHPI	Et4NBF4	1	17
0.075	1	TCNHPI	Et4NBF4	0	15	0.05	0.5	DABCO	Et4NBF4	1	0
0.05	0.75	TCNHPI	Et4NBF4	0	27	0.075	0.0	TEMPO	LiOTf	0	0
0.025	1	TCNHPI	LiOTf	1	60	0.125	0.5	TCNHPI	Et4NBF4	0	7
0.025	1	TCNHPI	Et4NBF4	1	59	0.05	1.0	DABCO	Bu4NPF6	0	0
0.075	1	TCNHPI	Bu4NPF6	0	19	0.1	0.0	DABCO	LiOTf	1	0
0.025	1	TCNHPI	Bu4NPF6	1	33	0.05	0.0	QD	LiClO4	0	0
0.025	0.75	TCNHPI	LiOTf	1	54	0.05	0.5	TEMPO	Bu4NClO4	1	0
0.025	0.75	QD	LiOTf	0	5	0.1	0.25	TCNHPI	LiOTf	0	14
0.125	0	DABCO	LiClO4	0	0	0.075	0.25	TCNHPI	Et4NBF4	1	7
0.075	0.5	QD	Et4NBF4	0	0	0.125	0.75	NHPI	LiOTf	0	5
0.05	0.75	TCNHPI	Bu4NClO4	0	15	0.025	0.5	NHPI	LiClO4	0	5
0.025	1	TCNHPI	Bu4NClO4	1	40	0.05	0.0	DABCO	Et4NBF4	1	0
0.125	1	TCNHPI	LiOTf	1	31	0.025	0.0	DABCO	Bu4NClO4	1	0
0.125	1	QD	Bu4NClO4	0	0	0.025	0.0	TCNHPI	LiClO4	0	0
0.125	1	NHPI	LiOTf	1	12	0.075	0.25	TCNHPI	Bu4NPF6	0	8
0.125	0.75	DABCO	LiOTf	1	0	0.1	1.0	TCNHPI	LiOTf	0	22
0.125	1	NHPI	Et4NBF4	0	13	0.025	0.5	NHPI	Bu4NClO4	0	0
0.1	0.75	TEMPO	LiClO4	1	0	0.075	0.0	NHPI	Bu4NClO4	1	0
0.1	0	NHPI	LiClO4	0	0	0.1	0.75	TCNHPI	Bu4NPF6	1	16
0.05	1	TCNHPI	Et4NBF4	1	43	0.125	0.25	DABCO	Bu4NClO4	1	0
0.1	0.5	TEMPO	LiOTf	0	0	0.05	0.5	DABCO	LiClO4	0	0
0.1	0.25	TEMPO	LiClO4	1	0	0.075	1.0	NHPI	LiOTf	0	23
0.05	1	TCNHPI	LiOTf	1	31	0.05	0.25	TEMPO	Bu4NClO4	1	0
0.05	0.75	NHPI	Bu4NClO4	0	10	0.125	0.25	NHPI	Et4NBF4	0	7
0.125	1	TEMPO	LiOTf	1	0	0.05	0.25	NHPI	Et4NBF4	0	24
0.125	0.25	QD	Bu4NClO4	0	0	0.075	1.0	TEMPO	Et4NBF4	1	0
0.075	0	TCNHPI	LiOTf	1	0	0.075	1.0	TCNHPI	Bu4NClO4	0	21
0.05	0.75	TCNHPI	Et4NBF4	1	28	0.1	0.25	TCNHPI	LiOTf	0	14
0.1	0.25	TEMPO	Et4NBF4	1	5	0.075	0.5	QD	Bu4NClO4	0	0
0.075	1	DABCO	Bu4NClO4	1	0	0.125	0.0	QD	Et4NBF4	1	0
0.05	0.75	TCNHPI	Bu4NClO4	1	10	0.1	0.5	NHPI	Et4NBF4	1	5
0.025	1	TCNHPI	Et4NBF4	0	42	0.1	0.75	TCNHPI	Bu4NClO4	1	17

**Table S18.** Synthesis conditions and reaction yield for compound 1 (four reactions per iteration).

		Active Lea	urning				Ι	Random Samp	oling		
Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield	Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield
0.075	0.5	QD	Et4NBF4	1	22	0.1	0	TEMPO	Et4NBF4	1	0
0.075	0.5	NHPI	LiClO4	1	9	0.125	0.25	TEMPO	Et4NBF4	1	0
0.1	0.5	TCNHPI	Bu4NPF6	0	23	0.1	0	TEMPO	Bu4NClO4	1	0
0.05	0.5	TEMPO	LiClO4	0	0	0.025	0.5	TCNHPI	LiClO4	1	10
0.125	1	TCNHPI	LiOTf	0	14	0.05	1	TCNHPI	LiClO4	1	5
0.125	0.25	NHPI	Bu4NPF6	0	45	0.05	1	TEMPO	LiClO4	1	0
0.125	0	TCNHPI	Bu4NPF6	1	0	0.025	0.75	QD	Et4NBF4	1	8
0.025	0	TCNHPI	Bu4NPF6	0	0	0.05	1	QD	LiClO4	0	4
0.125	1	NHPI	Et4NBF4	0	14	0.075	0	NHPI	Bu4NClO4	1	0
0.125	0.75	DABCO	Bu4NPF6	1	0	0.075	0	DABCO	Bu4NPF6	1	0
0.125	0	NHPI	LiOTf	0	0	0.025	0.75	TEMPO	Bu4NClO4	0	0
0.1	0.75	NHPI	LiClO4	0	6	0.125	0.25	NHPI	Bu4NPF6	1	33
0.1	1	NHPI	Bu4NClO4	0	50	0.1	0.25	TEMPO	Bu4NPF6	0	0
0.1	0.5	NHPI	Bu4NPF6	0	42	0.1	0.75	TEMPO	Bu4NPF6	1	0
0.05	0.5	NHPI	Et4NBF4	0	13	0.05	0.25	TCNHPI	Bu4NClO4	1	5
0.05	0.25	QD	LiOTf	0	7	0.05	0.25	QD	LiOTf	1	3
0.125	1	NHPI	Bu4NPF6	0	59	0.075	0	QD	LiClO4	0	0
0.05	0.25	NHPI	LiClO4	0	3	0.125	0.5	QD	Bu4NPF6	1	10
0.025	1	NHPI	Bu4NPF6	0	12	0.05	0.25	TCNHPI	LiClO4	1	8
0.025	0.5	NHPI	Bu4NPF6	0	12	0.1	0.75	DABCO	LiClO4	1	0
0.125	1	NHPI	Bu4NClO4	0	40	0.075	0.75	NHPI	LiOTf	1	32
0.125	0.75	NHPI	Bu4NClO4	1	55	0.075	0.75	DABCO	LiOTf	1	0
0.125	0	DABCO	LiClO4	0	0	0.1	0.75	TEMPO	LiClO4	1	0
0.1	1	NHPI	Bu4NPF6	0	34	0.125	0.75	TEMPO	LiOTf	1	0
0.125	1	NHPI	Bu4NPF6	1	12	0.05	0.25	NHPI	Bu4NClO4	0	14
0.125	0.25	NHPI	Bu4NClO4	0	28	0.1	0.5	DABCO	Bu4NPF6	0	0
0.1	1	QD	Bu4NClO4	0	10	0.125	0.5	DABCO	Bu4NPF6	0	0
0.075	1	NHPI	Et4NBF4	1	22	0.1	1	NHPI	Bu4NPF6	0	34
0.125	1	NHPI	Bu4NPF6	1	43	0.1	0.75	QD	LiClO4	0	4
0.125	1	NHPI	Bu4NClO4	1	47	0.075	1	QD	LiOTf	1	7
0.125	0.5	TCNHPI	Bu4NPF6	0	15	0.025	1	TCNHPI	Bu4NPF6	1	15
0.1	1	QD	Bu4NClO4	1	16	0.075	1	QD	Et4NBF4	1	8
0.125	1	TEMPO	Et4NBF4	1	0	0.125	0	QD	Bu4NClO4	1	0
0.125	0.5	QD	Bu4NPF6	1	10	0.075	1	TEMPO	LiOTf	1	0
0.125	0	NHPI	Bu4NClO4	1	0	0.05	1	QD	LiClO4	1	0
0.075	0.75	TCNHPI	Et4NBF4	0	17	0.025	0.5	TEMPO	Bu4NPF6	0	0
0.125	0.75	NHPI	Bu4NPF6	1	53	0.125	1	TEMPO	Bu4NClO4	0	0
0.125	0.75	NHPI	Bu4NPF6	0	53	0.05	0.75	TCNHPI	Bu4NClO4	0	18
0.075	0.5	NHPI	Bu4NClO4	0	30	0.125	0.75	TEMPO	Bu4NClO4	1	3
0.05	0.5	TEMPO	Bu4NClO4	1	0	0.025	0.5	NHPI	LiOTf	1	17
0.125	1	DABCO	Et4NBF4	0	0	0.075	1	NHPI	Bu4NClO4	0	38
0.125	0.75	NHPI	LiOTf	0	46	0.125	0.25	TCNHPI	LiClO4	0	5
0.125	0.5	NHPI	Bu4NClO4	0	42	0.125	0.5	TEMPO	Bu4NPF6	1	0
0.025	0.25	DABCO	Et4NBF4	0	0	0.075	0.25	NHPI	Bu4NPF6	0	29

**Table S19.** Synthesis conditions and reaction yield for compound 2 (four reactions per iteration).

		Active Lea	rning				1	Random Samp	ling		
Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield	Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield
0.075	0.5	QD	Et4NBF4	1	22	0.075	0	NHPI	Bu4NClO4	0	0
0.075	0.5	NHPI	LiClO4	1	15	0.05	0	DABCO	Bu4NClO4	1	0
0.1	0.5	TCNHPI	Bu4NPF6	0	14	0.05	0.5	TEMPO	Et4NBF4	0	0
0.05	0.5	TEMPO	LiClO4	0	3	0.025	0.5	TCNHPI	Bu4NClO4	1	20
0.125	1	NHPI	Bu4NPF6	1	19	0.025	0	TCNHPI	Et4NBF4	1	0
0.125	0.75	TCNHPI	LiOTf	0	0	0.05	0.5	QD	LiOTf	1	5
0.125	0.5	NHPI	Et4NBF4	0	2	0.025	0.5	TCNHPI	Bu4NClO4	1	20
0.125	0	TCNHPI	LiOTf	1	0	0.1	1	NHPI	Et4NBF4	0	14
0.125	1	TCNHPI	Bu4NPF6	1	18	0.075	0.5	QD	LiClO4	0	3
0.125	1	QD	Bu4NPF6	1	0	0.1	0.5	TEMPO	Et4NBF4	0	0
0.05	0.25	NHPI	Bu4NPF6	1	25	0.075	0.5	QD	Et4NBF4	0	0
0.025	1	NHPI	Bu4NPF6	1	41	0.05	1	NHPI	LiClO4	0	26
0.125	0.75	TEMPO	LiClO4	1	0	0.125	0.75	NHPI	Et4NBF4	1	13
0.125	0.5	TEMPO	LiOTf	0	0	0.05	0.25	TEMPO	Et4NBF4	1	0
0.125	0	TCNHPI	LiClO4	1	0	0.025	0.5	TCNHPI	Bu4NClO4	1	20
0.025	1	NHPI	Bu4NPF6	0	23	0.1	0.25	NHPI	Et4NBF4	0	6
0.125	1	TCNHPI	Bu4NPF6	0	21	0.05	0	NHPI	Bu4NClO4	1	0
0.125	0.5	TEMPO	LiClO4	1	0	0.075	0.75	QD	LiClO4	1	10
0.025	0.75	TEMPO	LiOTf	1	0	0.1	1	TEMPO	Bu4NClO4	1	0
0.025	0.25	DABCO	Bu4NPF6	1	0	0.1	0.5	TEMPO	Bu4NClO4	0	0
0.125	0	QD	Et4NBF4	0	0	0.025	0.25	NHPI	LiClO4	1	35
0.1	1	TEMPO	LiOTf	1	0	0.05	0.75	NHPI	Bu4NClO4	0	24
0.075	1	TCNHPI	LiOTf	0	14	0.05	0	NHPI	LiClO4	0	0
0.025	0.75	NHPI	LiOTf	1	49	0.025	0.75	TCNHPI	Bu4NClO4	0	20
0.075	0.25	NHPI	LiOTf	1	36	0.025	0.5	DABCO	Et4NBF4	1	0
0.05	0.5	NHPI	Et4NBF4	0	11	0.025	0.25	TEMPO	LiClO4	1	0
0.025	1	NHPI	LiOTf	1	27	0.075	0	TCNHPI	LiClO4	0	0
0.025	0.25	TCNHPI	Bu4NPF6	1	16	0.125	0.5	DABCO	Et4NBF4	0	0
0.1	0.75	TEMPO	Et4NBF4	1	0	0.075	0	TEMPO	LiOTf	0	0
0.1	0	NHPI	Bu4NPF6	1	0	0.05	0	TCNHPI	Bu4NClO4	0	0
0.05	1	TCNHPI	Bu4NClO4	0	13	0.125	0.25	DABCO	LiOTf	0	0
0.025	0	NHPI	LiOTf	0	0	0.025	0.5	TCNHPI	LiOTf	0	22
0.05	1	DABCO	Bu4NClO4	1	0	0.05	1	QD	LiClO4	1	15
0.025	1	NHPI	LiClO4	1	56	0.05	0.5	TCNHPI	Et4NBF4	1	19
0.025	1	NHPI	Et4NBF4	1	27	0.05	1	TEMPO	Et4NBF4	1	3
0.025	0.75	TCNHPI	LiClO4	1	29	0.125	0.5	TCNHPI	Bu4NPF6	1	12
0.125	1	TEMPO	LiOTf	1	0	0.025	0.5	NHPI	LiOTf	0	33
0.125	1	TEMPO	LiOTf	0	0	0.125	0.5	NHPI	Et4NBF4	0	2
0.05	1	NHPI	LiClO4	1	29	0.025	0.75	QD	LiClO4	1	15
0.025	1	TCNHPI	LiClO4	1	55	0.075	0	TCNHPI	Bu4NClO4	0	0
0.125	1	TCNHPI	LiOTf	1	25	0.05	0.5	TEMPO	LiOTf	1	0
0.125	1	TCNHPI	LiOTf	0	19	0.025	0	NHPI	LiOTf	1	0
0.125	1	TCNHPI	LiClO4	1	15	0.025	0.25	TCNHPI	LiOTf	1	16
0.125	1	TCNHPI	LiClO4	0	7	0.075	0	TEMPO	Bu4NPF6	1	0

**Table S20.** Synthesis conditions and reaction yield for compound 3 (four reactions per iteration).

		Active Lea	urning				I	Random Samp	ling		
Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield	Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield
0.075	0.5	QD	Et4NBF4	1	22	0.125	0	QD	Et4NBF4	0	0
0.075	0.5	NHPI	LiClO4	1	22	0.025	0.25	QD	Bu4NPF6	0	0
0.1	0.5	TCNHPI	Bu4NPF6	0	11	0.025	1	QD	Bu4NPF6	0	0
0.05	0.5	TEMPO	LiClO4	0	5	0.075	0.25	QD	Bu4NPF6	1	4
0.125	0	NHPI	LiOTf	1	0	0.1	0.25	TCNHPI	Bu4NPF6	1	11
0.1	0.5	DABCO	LiOTf	0	0	0.075	0.5	NHPI	Et4NBF4	0	20
0.025	1	NHPI	Bu4NPF6	1	18	0.1	0.5	DABCO	LiClO4	1	0
0.025	0.75	TCNHPI	Bu4NClO4	1	21	0.1	0	NHPI	LiOTf	0	0
0.05	0	NHPI	Et4NBF4	1	0	0.075	0.5	TEMPO	Bu4NClO4	0	0
0.025	1	TCNHPI	LiClO4	1	29	0.075	0	NHPI	Bu4NClO4	0	0
0.025	1	TCNHPI	Bu4NPF6	1	6	0.05	0.5	DABCO	LiClO4	1	0
0.025	0.75	TEMPO	Et4NBF4	0	0	0.1	1	DABCO	Bu4NClO4	1	0
0.125	1	TCNHPI	LiClO4	1	0	0.075	0.25	DABCO	LiClO4	1	0
0.1	1	NHPI	LiClO4	1	0	0.125	0.5	DABCO	LiClO4	1	0
0.1	1	NHPI	LiClO4	0	0	0.05	0.5	NHPI	Et4NBF4	1	25
0.075	0.25	TCNHPI	Et4NBF4	0	0	0.125	0.5	TCNHPI	Et4NBF4	0	4
0.125	0.5	QD	Et4NBF4	1	6	0.1	0	TCNHPI	LiOTf	0	0
0.025	0.75	QD	LiClO4	1	8	0.025	0.75	NHPI	Bu4NPF6	1	15
0.025	0.25	QD	LiOTf	0	6	0.125	0.75	QD	Et4NBF4	1	5
0.025	0	QD	LiClO4	1	0	0.125	0.5	TCNHPI	Et4NBF4	1	4
0.1	0.75	DABCO	LiOTf	0	0	0.075	0	QD	Et4NBF4	0	0
0.05	1	NHPI	Et4NBF4	1	56	0.025	1	DABCO	LiClO4	0	0
0.025	1	TCNHPI	LiOTf	1	61	0.1	0	TCNHPI	LiClO4	1	0
0.025	0.75	DABCO	Et4NBF4	0	0	0.1	0.25	QD	LiOTf	0	5
0.125	0.75	NHPI	Bu4NPF6	0	24	0.075	0	TCNHPI	Bu4NPF6	1	0
0.1	0.75	DABCO	LiClO4	1	0	0.05	0.75	TEMPO	Et4NBF4	0	0
0.05	1	TCNHPI	Et4NBF4	1	25	0.075	0.5	DABCO	LiClO4	1	0
0.025	1	TCNHPI	LiOTf	0	60	0.125	0.75	QD	Bu4NPF6	0	21
0.125	0	TCNHPI	LiOTf	0	0	0.125	0.5	NHPI	Bu4NClO4	1	7
0.075	0.75	TEMPO	Et4NBF4	1	0	0.125	0	TEMPO	LiOTf	0	0
0.05	1	TCNHPI	LiOTf	1	55	0.1	0	TEMPO	Et4NBF4	1	0
0.025	0	TCNHPI	LiOTf	1	0	0.05	1	DABCO	Et4NBF4	0	0
0.1	0.5	NHPI	LiClO4	1	11	0.125	1	DABCO	Bu4NClO4	1	0
0.025	1	TCNHPI	Bu4NClO4	0	30	0.075	0.25	DABCO	Bu4NClO4	0	0
0.025	1	NHPI	LiOTf	1	59	0.025	1	DABCO	LiClO4	1	0
0.025	1	NHPI	Et4NBF4	1	48	0.075	1	TCNHPI	Bu4NPF6	1	9
0.125	0.25	TEMPO	LiOTf	1	3	0.125	0.75	NHPI	LiClO4	0	0
0.075	0.75	NHPI	LiOTf	0	26	0.075	0.75	DABCO	LiOTf	0	0
0.05	1	NHPI	LiOTf	1	61	0.125	0.5	DABCO	LiClO4	0	0
0.05	1	NHPI	LiOTf	0	61	0.025	0	TCNHPI	Et4NBF4	0	0
0.125	0	DABCO	Bu4NPF6	1	11	0.075	0	TEMPO	Bu4NClO4	1	0
0.05	1	NHPI	LiClO4	1	18	0.075	0.25	TCNHPI	LiOTf	1	29
0.05	1	NHPI	Bu4NClO4	1	45	0.025	0	TEMPO	Et4NBF4	1	0
0.025	1	NHPI	LiOTf	0	58	0.125	0.25	QD	LiOTf	0	5

 Table S21. Synthesis conditions and reaction yield for compound 4 (four reactions per iteration).

Active Learning							I	Random Samp	oling		
Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield	Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield
0.075	0.5	QD	Et4NBF4	1	22	0.05	0	DABCO	Et4NBF4	0	0
0.075	0.5	NHPI	LiClO4	1	17	0.125	0	QD	Bu4NClO4	0	0
0.1	0.5	TCNHPI	Bu4NPF6	0	6	0.1	1	NHPI	Bu4NPF6	1	24
0.05	0.5	TEMPO	LiClO4	0	0	0.075	0.25	QD	Bu4NClO4	1	8
0.125	0.5	NHPI	Et4NBF4	1	24	0.05	0	DABCO	Bu4NPF6	0	0
0.125	0	NHPI	LiOTf	1	0	0.075	1	TEMPO	Bu4NClO4	1	0
0.025	1	NHPI	Bu4NPF6	1	21	0.075	0	QD	LiOTf	0	0
0.025	0.75	TCNHPI	Et4NBF4	1	5	0.025	0.5	NHPI	LiClO4	0	18
0.125	1	NHPI	Et4NBF4	0	10	0.025	0.75	DABCO	Bu4NPF6	1	0
0.125	0.25	NHPI	Bu4NPF6	1	6	0.125	0	TEMPO	Bu4NPF6	1	0
0.125	0	NHPI	Et4NBF4	0	0	0.05	0.25	NHPI	Et4NBF4	1	19
0.05	0	NHPI	Et4NBF4	0	0	0.025	1	NHPI	LiOTf	0	30
0.125	1	NHPI	LiClO4	1	15	0.1	1	QD	Et4NBF4	0	17
0.125	1	NHPI	LiClO4	0	16	0.1	1	TEMPO	LiClO4	0	0
0.075	1	TEMPO	LiOTf	1	0	0.075	0	TEMPO	LiOTf	0	0
0.025	1	QD	Bu4NPF6	1	34	0.125	1	DABCO	Bu4NPF6	0	0
0.125	1	QD	Bu4NPF6	1	12	0.125	0.5	TCNHPI	Et4NBF4	0	6
0.1	0.75	QD	Bu4NClO4	1	19	0.075	0.75	TEMPO	Bu4NPF6	1	0
0.075	0.5	TCNHPI	Et4NBF4	1	30	0.05	0.25	TCNHPI	Bu4NClO4	1	10
0.05	1	QD	LiClO4	1	23	0.025	0.5	QD	Bu4NClO4	1	16
0.125	1	TCNHPI	Et4NBF4	1	0	0.075	0	NHPI	Bu4NPF6	0	0
0.1	0.5	TCNHPI	Bu4NClO4	1	28	0.05	0.25	DABCO	Bu4NClO4	0	0
0.025	1	NHPI	LiOTf	0	30	0.1	0.25	NHPI	LiClO4	0	7
0.025	0.75	DABCO	LiOTf	0	0	0.125	0.5	TCNHPI	LiOTf	0	8
0.05	1	QD	Bu4NClO4	1	15	0.05	0.75	TEMPO	LiClO4	1	0
0.025	1	NHPI	Bu4NPF6	0	17	0.025	0.75	TEMPO	LiClO4	0	0
0.025	0.25	NHPI	Et4NBF4	0	11	0.125	0.75	TEMPO	Bu4NClO4	1	3
0.025	0	QD	Bu4NPF6	1	0	0.125	0.5	NHPI	Bu4NClO4	0	11
0.1	0.75	TEMPO	LiOTf	0	0	0.1	0.75	NHPI	Bu4NClO4	1	23
0.1	0	NHPI	Et4NBF4	0	0	0.05	0.25	NHPI	LiClO4	1	26
0.05	1	TCNHPI	Bu4NClO4	0	17	0.025	0.25	DABCO	LiClO4	1	0
0.025	1	NHPI	LiClO4	1	41	0.025	0	DABCO	Et4NBF4	0	0
0.1	0.5	TEMPO	LiClO4	1	0	0.125	1	DABCO	LiOTf	1	0
0.1	0.5	NHPI	LiOTf	1	39	0.1	0	QD	Bu4NClO4	0	0
0.05	1	QD	LiClO4	0	37	0.05	0.5	QD	Et4NBF4	1	15
0.025	1	NHPI	LiOTf	1	52	0.025	0.75	TCNHPI	LiOTf	1	20
0.125	0.75	TCNHPI	LiOTf	0	17	0.1	1	TEMPO	LiOTf	1	0
0.1	0.75	NHPI	Bu4NPF6	0	23	0.1	1	NHPI	LiOTf	0	39
0.1	0	TCNHPI	LiOTf	0	0	0.025	0.5	DABCO	LiOTf	1	0
0.025	0.25	DABCO	LiClO4	1	0	0.1	0.5	NHPI	Bu4NClO4	0	14
0.125	0.5	NHPI	LiClO4	1	10	0.075	0.75	QD	Et4NBF4	1	17
0.1	0.75	TCNHPI	Bu4NClO4	1	6	0.125	1	QD	LiClO4	0	15
0.05	1	NHPI	LiOTf	0	9	0.075	0.25	TEMPO	Bu4NPF6	1	0
0.025	1	NHPI	Et4NBF4	1	54	0.075	0.25	TEMPO	LiClO4	1	0

 Table S22. Synthesis conditions and reaction yield for compound 5 (four reactions per iteration).

Active Learning						Random Sampling					
Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield	Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield
0.075	0.5	QD	Et4NBF4	1	22	0.075	1	TCNHPI	LiOTf	0	20
0.075	0.5	NHPI	LiClO4	1	20	0.05	0.75	TEMPO	LiOTf	1	0
0.1	0.5	TCNHPI	Bu4NPF6	0	8	0.05	1	TEMPO	Bu4NClO4	1	0
0.05	0.5	TEMPO	LiClO4	0	0	0.125	0.25	TEMPO	Bu4NPF6	0	0
0.125	0.5	NHPI	Et4NBF4	1	10	0.125	0.75	TCNHPI	Bu4NPF6	0	5
0.125	0	NHPI	LiOTf	1	0	0.025	0	DABCO	LiClO4	0	0
0.025	0.75	TCNHPI	Et4NBF4	1	17	0.025	0.75	TEMPO	LiOTf	1	0
0.025	0.5	QD	LiOTf	1	45	0.075	0	TEMPO	Et4NBF4	1	0
0.125	0.75	DABCO	Et4NBF4	1	0	0.1	0.75	DABCO	LiOTf	0	21
0.1	0	QD	Bu4NPF6	1	0	0.075	0.5	NHPI	Bu4NClO4	1	32
0.05	1	QD	Et4NBF4	1	47	0.075	0.5	NHPI	Bu4NClO4	0	32
0.025	1	QD	LiOTf	1	34	0.025	0.5	DABCO	Et4NBF4	0	0
0.075	0	QD	Bu4NClO4	0	0	0.025	0	QD	Bu4NClO4	0	10
0.025	1	QD	LiOTf	0	71	0.1	0	TEMPO	Bu4NClO4	0	0
0.025	1	QD	Bu4NPF6	1	38	0.075	0.25	TEMPO	Et4NBF4	0	0
0.025	0.75	QD	LiClO4	0	73	0.05	0	DABCO	LiOTf	1	0
0.1	0.75	QD	LiClO4	0	0	0.025	0	QD	Bu4NPF6	0	0
0.025	1	QD	LiClO4	0	4	0.05	1	NHPI	Bu4NClO4	0	51
0.025	1	NHPI	Bu4NClO4	0	60	0.075	0	QD	Bu4NPF6	1	0
0.025	0.75	QD	LiOTf	0	12	0.1	0.25	TEMPO	LiClO4	1	0
0.025	1	NHPI	Et4NBF4	0	50	0.05	0.5	NHPI	Bu4NPF6	1	41
0.025	1	NHPI	Bu4NPF6	0	35	0.025	1	TCNHPI	Et4NBF4	0	24
0.025	0.75	TEMPO	Bu4NPF6	1	0	0.025	1	TEMPO	Et4NBF4	0	3
0.025	0.25	TEMPO	LiClO4	1	0	0.025	0	TEMPO	LiOTf	0	0
0.125	0.75	NHPI	Bu4NPF6	1	12	0.1	1	DABCO	Bu4NClO4	0	0
0.05	1	TEMPO	Et4NBF4	0	0	0.075	0.75	DABCO	Bu4NPF6	1	0
0.025	1	NHPI	LiOTf	1	70	0.05	0	NHPI	Bu4NPF6	0	0
0.025	0.25	NHPI	Bu4NPF6	1	81	0.05	0.5	TCNHPI	Bu4NClO4	1	22
0.1	1	DABCO	Bu4NClO4	1	0	0.1	0	DABCO	Bu4NPF6	1	0
0.05	0.5	NHPI	LiOTf	0	75	0.1	0.5	DABCO	Bu4NPF6	0	0
0.025	0.75	NHPI	LiOTf	1	57	0.125	0.5	NHPI	Bu4NClO4	0	18
0.025	0	NHPI	LiOTf	1	0	0.1	0	NHPI	Et4NBF4	1	0
0.075	0.75	NHPI	LiClO4	0	25	0.025	0.25	DABCO	Et4NBF4	1	0
0.05	0.5	NHPI	Bu4NPF6	1	41	0.05	0.25	DABCO	LiOTf	0	0
0.025	0	NHPI	LiClO4	0	0	0.125	0.25	TCNHPI	LiClO4	0	0
0.025	0	NHPI	Et4NBF4	0	0	0.125	0	TEMPO	Bu4NClO4	0	0
0.125	0.25	TCNHPI	LiClO4	0	0	0.1	0	TEMPO	LiOTf	1	0
0.025	1	NHPI	LiOTf	0	68	0.025	0.75	TEMPO	Bu4NClO4	1	0
0.025	0.5	TCNHPI	Bu4NClO4	0	24	0.075	0.25	QD	LiClO4	0	8
0.025	0.25	NHPI	Et4NBF4	0	43	0.125	1	NHPI	LiClO4	1	14
0.075	0	QD	LiClO4	0	0	0.05	1	DABCO	Bu4NClO4	0	0
0.05	0.75	NHPI	LiOTf	0	50	0.025	0	DABCO	Bu4NPF6	1	0
0.05	0	DABCO	Bu4NPF6	0	0	0.05	0	TEMPO	LiOTf	0	0
0.025	1	NHPI	LiClO4	0	49	0.125	0.75	TEMPO	LiOTf	0	0

 Table S23. Synthesis conditions and reaction yield for compound 6 (four reactions per iteration).

		Active Lea	rning				1	Random Samp	oling		
Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield	Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield
0.075	0.5	QD	Et4NBF4	1	22	0.075	1	DABCO	LiClO4	1	15
0.075	0.5	NHPI	LiClO4	1	13	0.125	0.75	TEMPO	Bu4NClO4	0	0
0.1	0.5	TCNHPI	Bu4NPF6	0	0	0.125	0	TCNHPI	Et4NBF4	0	0
0.05	0.5	TEMPO	LiClO4	0	0	0.025	0.5	TCNHPI	LiOTf	0	5
0.125	1	QD	LiOTf	1	0	0.025	1	DABCO	LiClO4	1	0
0.125	0.75	QD	Et4NBF4	1	36	0.05	0.25	TCNHPI	Et4NBF4	1	5
0.025	0	QD	Et4NBF4	1	0	0.075	0.75	DABCO	Bu4NPF6	0	0
0.025	0	DABCO	Bu4NClO4	1	0	0.125	1	NHPI	Bu4NClO4	1	16
0.125	1	NHPI	Et4NBF4	1	18	0.075	0.25	QD	LiOTf	0	15
0.125	0	NHPI	Et4NBF4	0	0	0.125	0	QD	LiClO4	1	0
0.05	1	TCNHPI	LiClO4	0	4	0.125	0	NHPI	LiClO4	0	0
0.05	0.5	TCNHPI	Bu4NClO4	1	14	0.025	0.5	QD	Bu4NClO4	1	11
0.075	0.25	TCNHPI	Et4NBF4	1	0	0.075	0.5	TCNHPI	LiOTf	0	7
0.025	1	QD	Et4NBF4	1	0	0.05	0.25	QD	LiClO4	0	15
0.025	0.75	QD	Et4NBF4	1	0	0.05	0	DABCO	Bu4NClO4	1	0
0.025	0.75	NHPI	Et4NBF4	1	3	0.05	0.75	QD	Bu4NPF6	1	4
0.125	1	QD	LiClO4	1	35	0.125	0.5	TEMPO	Bu4NPF6	1	0
0.125	1	QD	Bu4NPF6	1	0	0.1	0.75	DABCO	LiOTf	1	0
0.125	0.5	QD	Bu4NClO4	1	41	0.125	0	QD	Bu4NClO4	0	0
0.125	1	QD	Bu4NClO4	1	28	0.125	1	NHPI	Bu4NPF6	1	0
0.125	1	QD	Bu4NClO4	0	18	0.05	0.25	TCNHPI	Et4NBF4	0	4
0.125	0	QD	LiClO4	1	0	0.1	0.75	DABCO	Bu4NPF6	1	0
0.1	0	QD	Bu4NClO4	1	0	0.125	1	TCNHPI	Et4NBF4	1	0
0.075	0.5	QD	Bu4NClO4	0	32	0.025	0	DABCO	LiOTf	1	0
0.125	0.75	QD	LiClO4	1	41	0.125	0.25	QD	Bu4NPF6	1	17
0.125	0.75	QD	Bu4NClO4	1	19	0.025	0.5	QD	Bu4NClO4	1	11
0.125	0.5	QD	Et4NBF4	0	41	0.125	1	TEMPO	Et4NBF4	1	0
0.1	0.75	NHPI	LiClO4	1	34	0.075	0	NHPI	Et4NBF4	1	0
0.125	0.75	QD	LiClO4	0	18	0.125	0.5	DABCO	LiClO4	1	0
0.125	0.5	QD	LiClO4	0	26	0.075	0	TEMPO	Et4NBF4	0	0
0.1	0.75	QD	LiClO4	0	40	0.1	0.75	DABCO	Bu4NPF6	1	0
0.05	0.25	QD	LiOTf	0	13	0.075	0.25	QD	LiOTf	0	15
0.125	0.75	TEMPO	LiClO4	0	0	0.1	0.5	TCNHPI	Et4NBF4	1	22
0.125	0.75	TEMPO	Et4NBF4	1	0	0.1	0.25	TEMPO	Bu4NClO4	1	0
0.125	0.25	TCNHPI	Et4NBF4	0	7	0.025	0.25	TEMPO	Bu4NClO4	0	0
0.025	1	QD	Et4NBF4	0	5	0.075	1	TCNHPI	LiClO4	0	8
0.125	0.5	QD	Et4NBF4	1	40	0.125	0	TCNHPI	LiOTf	0	0
0.075	0.75	NHPI	Bu4NClO4	0	22	0.025	0.5	NHPI	LiOTf	0	5
0.075	0	TCNHPI	Bu4NPF6	0	0	0.1	0.75	QD	Bu4NClO4	0	29
0.05	0	TEMPO	LiOTf	0	0	0.125	0.75	TEMPO	Bu4NPF6	0	0
0.125	0.75	NHPI	LiClO4	0	31	0.05	0	TCNHPI	LiOTf	1	0
0.1	1	QD	Et4NBF4	1	38	0.05	0.75	TEMPO	Bu4NPF6	1	0
0.1	0.75	TCNHPI	LiOTf	1	5	0.075	0.5	NHPI	LiClO4	1	13
0.1	0.75	QD	Et4NBF4	1	33	0.05	0.25	QD	Bu4NPF6	0	9

 Table S24. Synthesis conditions and reaction yield for compound 7 (four reactions per iteration).

		Active Lea	rning				I	Random Samp	oling		
Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield	Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield
0.075	0.5	QD	Et4NBF4	1	16	0.05	0.75	TEMPO	Bu4NClO4	1	0
0.075	0.5	NHPI	LiClO4	1	0	0.05	1	QD	Bu4NPF6	0	10
0.1	0.5	TCNHPI	Bu4NPF6	0	0	0.1	1	NHPI	Bu4NPF6	1	3
0.05	0.5	TEMPO	LiClO4	0	0	0.125	0.75	NHPI	Bu4NClO4	1	0
0.125	0.75	QD	Et4NBF4	0	19	0.025	0.25	DABCO	LiClO4	1	0
0.125	0.75	QD	LiOTf	1	21	0.125	1	DABCO	Et4NBF4	0	0
0.025	0	QD	Et4NBF4	0	0	0.1	0.5	QD	LiOTf	1	14
0.025	0	DABCO	Bu4NClO4	0	0	0.1	0	NHPI	Bu4NPF6	1	0
0.125	1	QD	LiOTf	1	33	0.025	0	NHPI	Et4NBF4	1	0
0.125	1	QD	Et4NBF4	1	9	0.075	0.25	TEMPO	Et4NBF4	0	0
0.125	1	QD	Bu4NPF6	1	5	0.025	1	QD	LiClO4	0	16
0.125	0	TEMPO	LiOTf	0	0	0.1	1	NHPI	Bu4NPF6	1	3
0.125	0.75	TEMPO	LiClO4	1	0	0.05	0	TCNHPI	Bu4NPF6	0	0
0.125	0.5	TEMPO	LiOTf	0	0	0.075	0.5	DABCO	Bu4NClO4	1	0
0.05	0.5	DABCO	LiOTf	0	0	0.125	0	TEMPO	LiOTf	0	0
0.025	1	QD	LiOTf	0	24	0.125	0	TEMPO	LiOTf	1	0
0.125	1	QD	LiOTf	0	0	0.125	0.25	NHPI	Bu4NPF6	0	0
0.125	0.5	NHPI	LiOTf	0	2	0.025	0	NHPI	Bu4NPF6	0	0
0.125	0	QD	LiOTf	0	0	0.05	0.75	TCNHPI	Et4NBF4	1	5
0.075	0.75	QD	LiOTf	0	14	0.1	0.25	TCNHPI	Bu4NPF6	1	0
0.1	0.75	TCNHPI	LiOTf	1	0	0.075	0.5	TCNHPI	Et4NBF4	1	0
0.075	0.75	QD	LiOTf	1	14	0.125	0	TEMPO	Bu4NPF6	1	0
0.05	1	QD	Bu4NClO4	1	45	0.1	0.75	QD	Bu4NPF6	0	7
0.025	1	TCNHPI	LiOTf	1	4	0.025	0	DABCO	Et4NBF4	0	0
0.125	1	TEMPO	LiOTf	1	0	0.1	0.75	QD	LiClO4	0	11
0.125	1	TEMPO	LiOTf	0	0	0.025	0	NHPI	LiOTf	1	0
0.125	1	TEMPO	LiClO4	1	0	0.1	1	DABCO	Et4NBF4	0	0
0.125	1	TEMPO	LiClO4	0	0	0.125	0.5	TCNHPI	Bu4NPF6	0	0
0.125	1	TEMPO	Et4NBF4	1	0	0.075	1	DABCO	Et4NBF4	0	0
0.125	1	TEMPO	Et4NBF4	0	0	0.1	0.5	NHPI	LiOTf	1	5
0.125	1	TEMPO	Bu4NPF6	1	0	0.075	0.25	DABCO	Bu4NClO4	0	0
0.125	1	TEMPO	Bu4NPF6	0	0	0.05	0.75	QD	LiClO4	0	17
0.1	0.25	TCNHPI	LiClO4	0	2	0.1	0.5	TCNHPI	Bu4NClO4	1	0
0.075	0.75	DABCO	Bu4NClO4	0	0	0.05	0	DABCO	Bu4NClO4	0	0
0.05	1	TEMPO	Bu4NClO4	1	0	0.05	0.5	NHPI	Bu4NPF6	1	0
0.025	1	QD	LiClO4	1	20	0.025	0.5	QD	LiClO4	1	19
0.125	1	TEMPO	Bu4NClO4	1	0	0.05	0.75	TEMPO	Bu4NClO4	0	0
0.125	1	TEMPO	Bu4NClO4	0	0	0.05	1	NHPI	Et4NBF4	1	7
0.125	1	TCNHPI	LiOTf	1	0	0.025	1	TEMPO	Bu4NPF6	1	0
0.125	1	TCNHPI	LiOTf	0	0	0.025	0.25	NHPI	Bu4NPF6	0	0
0.125	1	TCNHPI	LiClO4	1	0	0.125	0.75	TEMPO	Bu4NClO4	0	0
0.125	1	TCNHPI	LiClO4	0	0	0.05	0	TEMPO	Bu4NPF6	0	0
0.125	1	TCNHPI	Et4NBF4	1	0	0.05	0.75	TEMPO	LiOTf	0	0
0.125	1	TCNHPI	Et4NBF4	0	0	0.025	1	TCNHPI	Bu4NClO4	1	8

Table S25. Synthesis conditions and reaction yield for compound 8 (four reactions per iteration).

		Active Lea	rning				F	Random Samp	ling		
Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield	Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield
0.1	0.5	TCNHPI	Bu4NPF6	0	6	0.125	1.0	TEMPO	LiClO4	0	0
0.075	0.5	QD	Et4NBF4	1	5	0.1	0.25	TEMPO	Et4NBF4	0	0
0.075	0.5	QD	LiOTf	1	5	0.025	0.5	QD	Bu4NClO4	0	0
0.075	0.5	NHPI	LiClO4	1	8	0.025	1.0	QD	Et4NBF4	1	5
0.05	0.5	TEMPO	Et4NBF4	0	0	0.1	0.25	DABCO	Bu4NClO4	1	0
0.125	1	NHPI	Bu4NPF6	1	10	0.05	0.0	TCNHPI	Bu4NClO4	0	0
0.125	0.5	TCNHPI	LiClO4	1	7	0.1	0.0	TEMPO	Bu4NPF6	1	0
0.125	0	NHPI	LiClO4	1	0	0.075	0.0	TCNHPI	LiClO4	0	0
0.025	1	TCNHPI	Bu4NPF6	1	33	0.025	0.5	QD	LiClO4	0	0
0.025	0	NHPI	LiClO4	0	0	0.1	0.75	NHPI	LiClO4	0	0
0.125	0.5	DABCO	Bu4NPF6	1	0	0.1	1.0	DABCO	Bu4NClO4	1	0
0.075	0.25	TEMPO	Bu4NPF6	1	0	0.075	0.75	DABCO	LiClO4	0	0
0.025	1	DABCO	Bu4NPF6	1	0	0.05	0.25	QD	Bu4NPF6	0	0
0.025	0.75	TEMPO	LiOTf	0	0	0.1	0.0	TEMPO	Bu4NClO4	1	0
0.025	0.25	TCNHPI	Bu4NPF6	1	12	0.05	0.0	NHPI	Et4NBF4	1	0
0.1	1	TCNHPI	Bu4NPF6	0	19	0.05	0.0	NHPI	LiClO4	1	0
0.05	0.75	TCNHPI	Et4NBF4	0	27	0.125	0.75	QD	Bu4NClO4	0	0
0.025	1	TCNHPI	LiOTf	1	60	0.1	0.25	DABCO	Bu4NPF6	0	0
0.025	1	TCNHPI	LiClO4	1	51	0.075	0.0	TEMPO	Bu4NPF6	1	0
0.025	1	TCNHPI	Et4NBF4	1	59	0.075	0.25	TEMPO	LiOTf	1	0
0.1	0.75	NHPI	Bu4NPF6	1	7	0.125	0.25	TCNHPI	Bu4NClO4	0	5
0.075	0.5	QD	Bu4NPF6	1	6	0.075	0.75	NHPI	Bu4NPF6	1	19
0.025	1	TCNHPI	Bu4NCIO4	1	40	0.075	0.75	NHPI	LiOIf	0	22
0.025	0.75	ICNHPI	Bu4NCIO4	1	31	0.1	0.75	TEMPO	Et4NBF4	0	0
0.025	0.75	NHPI	LIOIT	1	10	0.1	0.75	TEMPO	Bu4NPF6	0	0
0.125	0.25	TEMPO	LIUIT BudNCIO4	1	0	0.05	0.0	NHPI	Et4NBF4	1	0
0.1	1	UD	L:OT	1	21	0.03	1.0	TEMPO	D:: 4NDE6	1	25
0.05	0.75	NHDI	Bu4NCIO4	1	10	0.025	0.5	TEMPO	Bu4NPF0	1	0
0.03	1	TCNHDI	LiOTf	0	10	0.025	0.75		Bu4NClO4	0	0
0.025	1	TCNHDI	BudNCIO	1	20	0.075	0.5	NHDI	LiOTE	1	20
0.125	0.25	NHPI	LiClO4	1	5	0.075	0.0	TEMPO	LIOTI	1	0
0.1	0.75	TCNHPI	Et4NBF4	1	18	0.1	0.5	TEMPO	LiClO4	0	0
0.05	1	TCNHPI	Et4NBF4	1	43	0.1	0.5	DABCO	Bu4NPF6	0	0
0.025	0	DABCO	LiOTf	1	0	0.075	1.0	DABCO	LiOTf	0	0
0.125	1	TEMPO	LiOTf	1	0	0.075	0.25	NHPI	Bu4NPF6	0	24
0.125	1	TEMPO	LiOTf	0	0	0.075	0.75	TEMPO	Bu4NPF6	1	0
0.125	1	TEMPO	LiClO4	1	0	0.1	0.25	DABCO	LiClO4	0	0
0.125	1	TEMPO	LiClO4	0	0	0.125	0.5	TEMPO	LiClO4	1	0
0.125	1	TEMPO	Et4NBF4	1	0	0.1	1.0	QD	Et4NBF4	1	0
0.125	1	TEMPO	Et4NBF4	0	0	0.075	0.0	ТЕМРО	Bu4NPF6	1	0
0.125	1	TEMPO	Bu4NPF6	1	0	0.025	0.5	QD	Et4NBF4	1	5
0.125	1	TEMPO	Bu4NPF6	0	0	0.125	0.0	DABCO	LiClO4	0	0
0.125	1	TEMPO	Bu4NClO4	1	0	0.05	1.0	DABCO	Bu4NClO4	1	0
0.125	1	TEMPO	Bu4NClO4	0	0	0.075	1.0	TEMPO	Et4NBF4	1	0

 Table S26. Synthesis conditions and reaction yield for compound 1 (five reactions per iteration).

0.125	1	TCNHPI	LiOTf	1	31	0.1	0.0	TEMPO	LiOTf	1	0
0.125	1	TCNHPI	LiOTf	0	30	0.05	1.0	DABCO	Et4NBF4	0	0
0.125	1	TCNHPI	LiClO4	1	23	0.1	0.25	TCNHPI	Et4NBF4	1	5
0.125	1	TCNHPI	LiClO4	0	20	0.075	0.25	TEMPO	LiClO4	1	0
0.125	1	TCNHPI	Et4NBF4	1	32	0.025	1.0	TCNHPI	Bu4NClO4	0	26
0.125	1	TCNHPI	Et4NBF4	0	16	0.1	1.0	TCNHPI	Et4NBF4	1	19
0.125	1	TCNHPI	Bu4NPF6	1	5	0.05	0.5	TCNHPI	Bu4NPF6	1	15
0.125	1	TCNHPI	Bu4NPF6	0	5	0.025	0.75	DABCO	Et4NBF4	1	0
0.125	1	TCNHPI	Bu4NClO4	0	5	0.05	0.5	DABCO	LiClO4	0	0
0.125	1	QD	LiOTf	1	0	0.05	0.75	TCNHPI	Bu4NPF6	1	27

## Table S27. Synthesis conditions and reaction yield for compound 1 (three reactions per iteration).

		Active Lea	rning				F	Random Samp	ling		
Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield	Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield
0.075	0.5	QD	Et4NBF4	1	5	0.125	0.25	TEMPO	Et4NBF4	1	0
0.075	0.5	NHPI	LiClO4	1	8	0.025	0.0	QD	Bu4NPF6	1	5
0.1	0.5	TCNHPI	Bu4NPF6	0	6	0.125	0.25	TEMPO	Et4NBF4	0	0
0.125	1	TEMPO	LiOTf	1	0	0.025	0.75	QD	Et4NBF4	0	0
0.125	0	DABCO	LiClO4	0	0	0.1	0.25	NHPI	Et4NBF4	1	7
0.025	0	NHPI	LiClO4	0	0	0.1	0.75	DABCO	Bu4NPF6	1	0
0.125	0.75	NHPI	LiOTf	0	5	0.075	0.5	DABCO	Et4NBF4	0	0
0.05	1	NHPI	Bu4NPF6	1	58	0.125	0	TCNHPI	LiClO4	1	0
0.025	0.75	TCNHPI	Bu4NPF6	1	39	0.075	0.25	DABCO	LiOTf	1	0
0.05	0.5	NHPI	Bu4NPF6	1	60	0.075	0.5	TCNHPI	Bu4NClO4	0	10
0.025	1	QD	Bu4NPF6	1	3	0.05	0.25	DABCO	Et4NBF4	1	5
0.025	0	NHPI	Bu4NPF6	1	0	0.05	1.0	DABCO	LiOTf	0	0
0.1	1	NHPI	Et4NBF4	1	16	0.075	0.75	NHPI	Bu4NClO4	1	12
0.1	0.5	TEMPO	Bu4NPF6	0	0	0.075	1.0	QD	LiClO4	1	5
0.025	0.75	TEMPO	Et4NBF4	1	0	0.075	0.25	QD	Et4NBF4	0	0
0.125	1	NHPI	Bu4NPF6	1	10	0.1	0.25	TCNHPI	Bu4NPF6	1	3
0.025	1	NHPI	Bu4NPF6	1	5	0.075	0.75	TEMPO	LiClO4	1	0
0.025	0.75	DABCO	Et4NBF4	1	0	0.1	0.0	TEMPO	Bu4NClO4	1	0
0.125	0.25	NHPI	LiClO4	0	5	0.1	0.25	DABCO	Bu4NPF6	0	0
0.05	0.75	TCNHPI	LiClO4	1	25	0.125	0.75	DABCO	Bu4NClO4	1	0
0.025	0	NHPI	LiClO4	1	0	0.075	0.0	DABCO	LiOTf	1	0
0.025	1	QD	LiClO4	1	0	0.05	0.25	QD	LiOTf	0	5
0.025	0.5	DABCO	Bu4NClO4	1	0	0.1	1.0	QD	LiOTf	1	7
0.025	0	TCNHPI	LiOTf	0	0	0.025	1.0	DABCO	Bu4NClO4	1	0
0.125	1	TEMPO	LiOTf	0	0	0.05	0.5	TEMPO	Bu4NClO4	1	0
0.125	1	TCNHPI	Bu4NPF6	1	5	0.05	0.25	DABCO	Et4NBF4	0	0
0.125	0.5	TEMPO	Et4NBF4	0	0	0.1	0.25	DABCO	Bu4NPF6	1	0
0.125	1	TEMPO	LiClO4	1	0	0.025	0.5	NHPI	Bu4NPF6	0	13
0.125	1	TEMPO	LiClO4	0	0	0.05	0	DABCO	Bu4NPF6	1	0
0.125	1	TEMPO	Et4NBF4	1	0	0.05	0.75	DABCO	Et4NBF4	0	0
0.125	1	TEMPO	Et4NBF4	0	0	0.1	0.75	QD	Bu4NClO4	0	0
0.125	1	TEMPO	Bu4NPF6	1	0	0.1	1.0	QD	Bu4NPF6	1	0
0.125	1	TEMPO	Bu4NPF6	0	0	0.1	1.0	DABCO	LiClO4	0	0



**Figure S41.** Highest reaction yield for compound **1** achieved in each iteration, using the active learning approach and random sampling as demonstrated in Table S18.



Figure S42. Highest reaction yield for compound 2 achieved in each iteration, using the active learning approach and random sampling as demonstrated in Table S19.



Figure S43. Highest reaction yield for compound 3 achieved in each iteration, using the active learning approach and random sampling as demonstrated in Table S20.



**Figure S44.** Highest reaction yield for compound **4** achieved in each iteration, using the active learning approach and random sampling as demonstrated in Table S21.



**Figure S45.** Highest reaction yield for compound **5** achieved in each iteration, using the active learning approach and random sampling as demonstrated in Table S22.



**Figure S46.** Highest reaction yield for compound **6** achieved in each iteration, using the active learning approach and random sampling as demonstrated in Table S23.



**Figure S47.** Highest reaction yield for compound **7** achieved in each iteration, using the active learning approach and random sampling as demonstrated in Table S24.



Figure S48. Highest reaction yield for compound 8 achieved in each iteration, using the active learning approach and random sampling as demonstrated in Table S25.



**Figure S49.** Distribution of outcomes in C-H oxidation reaction **1**. A total of 455 reaction conditions were investigated for this compound by exploring different batch sizes and active learning approaches. Around half of the reactions resulted in a yield lower than 5%, indicating little to no desired product formation from alpha-pinene to verbenone, while the rest achieved yields of up to 60%.

**Table S28.** Summary of reaction yields optimized for each compound in this study, achieved by combining ML and LLM without inputting any chemical knowledge. The results are compared with the yield range obtained by humans, either by reproducing electrochemical synthesis conditions reported in the literature or through manual optimization. The range is determined by running the conditions in 4 parallel vials under similar conditions, and the yield reported in the original literature is also provided.

Compound	Optimized Reaction Yield (%)	Reproducing Using Literature Condition or Manual Optimization (%)	Electrochemical Condition Reported by Literature (%)
1	60	57-67	67 <sup>24</sup>
2	59	48-62	N/A
3	56	50-60	N/A
4	61	60-74	$77^{24}$
5	54	47-57	$49^{24}$ and $63^{24}$
6	81	75-90	81 <sup>24</sup>
7	41	40-63	N/A
8	45	45-55	58 <sup>30</sup>

Iteration	Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield
0	0.075	0.5	QD	Et4NBF4	1	5
0	0.075	0.5	NHPI	LiClO4	1	8
0	0.1	0.5	TCNHPI	Bu4NPF6	0	6
0	0.05	0.5	TEMPO	LiClO4	0	0
1	0.1	1	NHPI	LiClO4	1	0
1	0.05	0.25	DABCO	Bu4NPF6	0	0
1	0.125	0.5	TEMPO	Et4NBF4	1	0
1	0.1	0.75	TCNHPI	LiOTf	1	17
2	0.075	1	TCNHPI	Bu4NPF6	1	21
2	0.125	0.75	NHPI	LiOTf	0	5
2	0.025	0.5	QD	Et4NBF4	1	5
2	0.05	1	DABCO	LiClO4	1	0
3	0.1	0.75	NHPI	Bu4NClO4	1	5
3	0.125	1	TCNHPI	LiClO4	1	23
3	0.075	0.5	TEMPO	Bu4NPF6	1	0
3	0.05	0.25	QD	LiOTf	1	0
4	0.1	1	TCNHPI	Bu4NClO4	1	20
4	0.125	0.75	NHPI	Et4NBF4	1	6
4	0.075	1	DABCO	Bu4NPF6	1	0
4	0.05	0.5	TEMPO	LiOTf	1	0
5	0.125	0.75	TCNHPI	LiOTf	1	29
5	0.1	0.75	NHPI	Et4NBF4	1	6
5	0.075	0.25	TCNHPI	LiClO4	1	0
5	0.05	1	NHPI	Bu4NPF6	1	58
6	0.075	1	NHPI	Bu4NClO4	1	12
6	0.05	0.75	NHPI	LiOTf	1	28
6	0.125	1	TCNHPI	Bu4NPF6	0	5
6	0.1	0.5	NHPI	LiClO4	1	0
7	0.025	1	NHPI	Bu4NPF6	1	5
7	0.075	0.75	TCNHPI	Et4NBF4	1	17
7	0.1	1	NHPI	LiOTf	1	12
7	0.05	0.25	NHPI	Bu4NClO4	1	10
8	0.075	1	NHPI	Et4NBF4	1	13
8	0.1	0.75	TCNHPI	LiOTf	1	17
8	0.05	0.75	NHPI	LiClO4	1	18
8	0.125	0.5	TCNHPI	Bu4NClO4	1	5
9	0.1	1	NHPI	Bu4NClO4	1	7
9	0.075	0.5	TCNHPI	LiOTf	1	14
9	0.05	1	TCNHPI	Bu4NPF6	1	33
9	0.1	0.25	NHPI	Et4NBF4	1	7
10	0.075	1	TCNHPI	LiClO4	1	23
10	0.1	0.75	NHPI	Bu4NPF6	1	7
10	0.125	0.25	TCNHPI	LiOTf	1	0
10	0.05	1	NHPI	LiOTf	1	31

Table S29. Synthesis conditions suggested by LLM without ML and measured reaction yields for compound 1 (four reactions per iteration).

Iteration	Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield
0	0.1	0.5	TCNHPI	Bu4NPF6	0	6
0	0.075	0.5	QD	Et4NBF4	1	5
0	0.075	0.5	QD	LiOTf	1	5
0	0.075	0.5	NHPI	LiClO4	1	8
0	0.05	0.5	TEMPO	Et4NBF4	0	0
1	0.1	0.75	NHPI	LiClO4	1	5
1	0.125	0.5	NHPI	Bu4NPF6	1	0
1	0.075	0.5	DABCO	LiClO4	1	0
1	0.1	0.5	NHPI	LiOTf	1	10
1	0.05	1	TCNHPI	Bu4NPF6	0	5
2	0.1	0.5	NHPI	Et4NBF4	1	5
2	0.075	0.5	QD	Bu4NClO4	1	0
2	0.05	0.5	TCNHPI	LiClO4	1	11
2	0.1	0.75	NHPI	Bu4NClO4	1	5
2	0.125	0.5	QD	LiOTf	1	0
3	0.05	0.5	TCNHPI	LiOTf	1	20
3	0.025	0.75	NHPI	LiOTf	1	10
3	0.1	0.5	NHPI	Bu4NPF6	1	7
3	0.075	0.25	DABCO	LiOTf	1	0
3	0.1	0.5	TEMPO	LiClO4	1	0
4	0.05	0.75	TCNHPI	LiOTf	1	31
4	0.05	0.75	NHPI	LiOTf	1	28
4	0.075	0.5	TCNHPI	Bu4NClO4	1	5
4	0.025	0.5	TCNHPI	LiClO4	1	58
4	0.1	0.75	TCNHPI	LiClO4	1	20
5	0.025	1	TCNHPI	LiClO4	1	51
5	0.025	0.75	NHPI	LiClO4	1	0
5	0.05	1	NHPI	LiOTf	1	31
5	0.025	0.5	TCNHPI	Et4NBF4	1	59
5	0.05	0.5	NHPI	Bu4NClO4	1	10
6	0.025	1	TCNHPI	Et4NBF4	1	59
6	0.05	0.5	TCNHPI	Bu4NPF6	1	15
6	0.05	0.75	NHPI	Bu4NPF6	1	59
6	0.025	0.5	TCNHPI	Bu4NPF6	1	10
6	0.05	0.75	NHPI	Et4NBF4	1	20
7	0.025	1	NHPI	LiOTf	1	6
7	0.025	1	TCNHPI	Bu4NClO4	1	20
7	0.05	1	TCNHPI	LiClO4	1	37
7	0.05	1	NHPI	LiClO4	1	32
7	0.05	1	NHPI	Bu4NClO4	1	23
8	0.025	1	NHPI	Et4NBF4	1	24
8	0.025	1	TCNHPI	LiOTf	1	60
8	0.05	0.5	NHPI	Et4NBF4	1	3
8	0.075	1	TCNHPI	Et4NBF4	1	14
8	0.075	1	NHPI	Bu4NPF6	1	18

**Table S30.** Synthesis conditions suggested by LLM without ML and measured reaction yields for compound **1** (five reactions per iteration).

9	0.025	0.75	TCNHPI	Bu4NPF6	1	39
9	0.05	1	TCNHPI	Bu4NClO4	1	5
9	0.025	0.5	NHPI	Bu4NPF6	1	7
9	0.1	1	TCNHPI	Bu4NPF6	1	0
9	0.05	0.25	NHPI	LiClO4	1	16
10	0.025	0.5	NHPI	Et4NBF4	1	0
10	0.025	0.25	TCNHPI	LiClO4	1	43
10	0.05	0	TCNHPI	LiOTf	1	0
10	0.1	0.25	NHPI	LiOTf	1	9
10	0.125	0.5	NHPI	LiClO4	1	0

**Table S31.** Synthesis conditions suggested by LLM without ML and measured reaction yields for compound **1** (three reactions per iteration).

Iteration	Substrate Conc.	Mediator eq.	Mediator	Electrolyte	Co solvent	Yield	
0	0.075	0.5	QD	Et4NBF4	1	5	
0	0.075	0.5	NHPI	LiClO4	1	8	
0	0.1	0.5	TCNHPI	Bu4NPF6	0	6	
1	0.1	0.75	NHPI	LiClO4	1	5	
1	0.05	0.5	TEMPO	Bu4NPF6	1	0	
1	0.125	1	TCNHPI	Et4NBF4	0	16	
2	0.1	1	DABCO	LiOTf	1	0	
2	0.125	0.5	QD	Bu4NClO4	0	0	
2	0.05	0.75	TCNHPI	LiClO4	1	25	
3	0.025	1	TCNHPI	LiClO4	0	56	
3	0.1	0.75	NHPI	Bu4NClO4	0	5	
3	0.075	0.5	TEMPO	LiOTf	1	0	
4	0.05	1	TCNHPI	LiClO4	1	37	
4	0.075	1	TCNHPI	Bu4NClO4	0	0	
4	0.025	0.75	NHPI	Et4NBF4	1	8	
5	0.05	0.75	TCNHPI	LiOTf	1	31	
5	0.025	0.5	NHPI	Bu4NPF6	1	7	
5	0.1	1	QD	Et4NBF4	1	0	
6	0.075	1	TCNHPI	LiOTf	1	8	
6	0.025	1	TCNHPI	Bu4NClO4	1	40	
6	0.1	0.75	DABCO	LiClO4	0	0	
7	0.125	1	TCNHPI	LiClO4	1	23	
7	0.05	1	NHPI	LiOTf	1	31	
7	0.075	0.75	TEMPO	Bu4NPF6	1	0	
8	0.05	1	TCNHPI	Et4NBF4	0	36	
8	0.1	0.5	NHPI	LiOTf	0	5	
8	0.075	1	QD	LiClO4	1	0	
9	0.125	0.75	TCNHPI	LiOTf	0	19	
9	0.05	0.5	NHPI	Bu4NPF6	0	59	
9	0.075	0.25	TEMPO	LiOTf	0	0	
10	0.025	0.75	NHPI	Bu4NPF6	0	0	
10	0.05	0.75	NHPI	Bu4NPF6	1	59	
10	0.1	1	TCNHPI	Bu4NPF6	0	19	

**User**: Act as a professional researcher in organic chemistry. You are asked to analyze the data and propose synthesis parameters to run experiments to maximize the yield for electrochemical C-H oxidation of alpha-pinene (CC1=CCC2CC1C2(C)C). The reaction varies with different substrate concentrations, different mediator compounds and equivalents related to the substrate, various electrolytes, and the choice of using acetonitrile only or an acetonitrile/HFIP cosolvent. Below are the options for each synthesis parameter; note that due to experimental limitations, they are discrete. The other synthesis parameters, such as voltage, electrodes, and temperature, are kept constant throughout the experiment.

Substrate Concentration (M): 0.025, 0.05, 0.075, 0.1, 0.125 Mediator eq.: 0, 0.25, 0.5, 0.75, 1 Mediator type: NHPI, TCNHPI, QD, DABCO, TEMPO Electrolyte type: LiClO4, LiOTf , Bu4NClO4, Et4NBF4, Bu4NPF6 Cosolvent HFIP: 0, 1

where NHPI = N-Hydroxyphthalimide; TCNHPI = Tetrachloro-N-hydroxyphthalimide; QD = Quinuclidine; DABCO = 1,4-Diazabicyclo[2.2.2]octane; TEMPO = 2,2,6,6-Tetramethylpiperidine 1-oxyl HFIP - Hexafluoro-2-propanol

You are working with an experimental chemist. First, start with five initial random experiments, then based on your understanding of these experiments, propose five new experimental conditions to try. In the next conversation, the chemist will run the experiments and report back the yields. Based on the results from the initial ten experiments, you will propose five more. In total, there are ten iterations. The goal is to behave like an expert chemist and make proposals not only based on yield observations but also on your understanding of each synthesis parameter and this specific type of reaction.

Now, here is the first 4 randomly chosen initial experiments, the reaction yield is a number between 0 to 100:

Iteration Substrate Concentration			Mediato	r eq.	Mediator type	Electrolyte type	Co solvent HFIP	yield	
0	0.1	0.5	TCNHPI	Bu4NPF6	0	6			
0	0.075	0.5	QD	Et4NBF4	1	5			
0	0.075	0.5	QD	LiOTf	1	5			
0	0.075	0.5	NHPI	LiClO4	1	8			

Please follow strictly to below template to answer your questions, make sure you include the {} and comma in the template.

Previous Reactions and Yields:

Iteration 0 Reaction 1: {Concentration, Mediator eq, Mediator Type, Electrolyte Type, Cosolvent HFIP} Reaction Yield: {Yield} Iteration 0 Reaction 2: {Concentration, Mediator eq, Mediator Type, Electrolyte Type, Cosolvent HFIP} Reaction Yield: {Yield} Iteration 0 Reaction 3: {Concentration, Mediator eq, Mediator Type, Electrolyte Type, Cosolvent HFIP} Reaction Yield: {Yield}

Iteration m Reaction n: {Concentration, Mediator eq, Mediator Type, Electrolyte Type, Cosolvent HFIP, reaction yield}

Your Observation:{include your observations such as any trends or patterns you have identified, and summarize how you rationale the selection of conditions for the subsequent experiments}

Suggested Condition 1: {Concentration, Mediator eq, Mediator Type, Electrolyte Type, Cosolvent HFIP}

Reasoning 1: {Your reasoning for suggested condition 1}

Suggested Condition 2: {Concentration, Mediator eq, Mediator Type, Electrolyte Type, Cosolvent HFIP} Reasoning 2: {Your reasoning for suggested condition 2}

Suggested Condition 3: {Concentration, Mediator eq, Mediator Type, Electrolyte Type, Cosolvent HFIP} Reasoning 3: {Your reasoning for suggested condition 3}

Suggested Condition 4: {Concentration, Mediator eq, Mediator Type, Electrolyte Type, Cosolvent HFIP} Reasoning 4: {Your reasoning for suggested condition 4}

**Figure S50.** A representative user prompt in a zero-shot conversation where the user directly asks an LLM to mimic a human analyzing data by trends and providing suggestions. The user specifies the exact chemical context and the choice of synthesis parameters and provides a template for the output to formulate the answer from the LLM. Note that no machine learning is involved in this process; the LLM acts as a black box optimizer, and the response is based entirely on empirical knowledge. The prompt can be adjusted to have three / four / five reactions per iteration.

## Assistant:

Previous Reactions and Yields: Iteration 0 Reaction 1: {0.075, 0.5, QD, Et4NBF4, 1} Reaction Yield: {5} Iteration 0 Reaction 2: {0.075, 0.5, NHPI, LiCIO4, 1} Reaction Yield: {8} Iteration 0 Reaction 3: {0.1, 0.5, TCNHPI, Bu4NPF6, 0} Reaction Yield: {6} Iteration 0 Reaction 4: {0.05, 0.5, TEMPO, LiCIO4, 0} Reaction Yield: {0}

Your Observation: From the initial data, it appears that the yields are generally low across all conditions tested, suggesting room for optimization in multiple parameters. Notably, the use of HFIP as a cosolvent correlates with some increase in yield (experiments 1 and 2). TEMPO, in the absence of HFIP, led to a yield of zero, indicating it might require specific conditions to be effective or could be a less suitable mediator for this reaction. The electrolyte LiCIO4 showed mixed results, which could merit further exploration with different mediators or substrate concentrations.

Suggested Condition 1: {0.1, 1, NHPI, LiCIO4, 1}

Reasoning 1: Increasing the concentration of the substrate to 0.1 M and mediator equivalence to 1 with NHPI, which had some success in earlier trials, could enhance reaction efficiency. HFIP is included as it showed potential in boosting yields. Suggested Condition 2: {0.05, 0.25, DABCO, Bu4NPF6, 0}

Reasoning 2: Reducing mediator equivalence to 0.25 and using DABCO, a different mediator type, might provide new insights into mediator performance. The lower substrate concentration and the absence of HFIP are intended to assess DABCO's intrinsic activity.

Suggested Condition 3: {0.125, 0.5, TEMPO, Et4NBF4, 1}

Reasoning 3: Increasing substrate concentration to the maximum and reintroducing TEMPO with HFIP might improve TEMPO's activity, which was ineffective in a prior test. Et4NBF4 is selected based on its prior use but paired differently. Suggested Condition 4: {0.1, 0.75, TCNHPI, LiOTf, 1}

Reasoning 4: Increasing mediator equivalence to 0.75 and changing the electrolyte to LiOTf could provide a different ionic environment for TCNHPI, potentially enhancing its effectiveness. The presence of HFIP aims to stabilize the reaction mixture.

**Figure S51.** A representative assistant answer based on the user's prompt. The LLM is instructed to first summarize the previously observed yields for various conditions to verify its understanding of the prompt and its ability to interpret the data. Following this, the assistant should provide a narrative observation to analyze the trend in yields and, based on this analysis, suggest additional conditions aimed at enhancing the yield. Each suggestion should be accompanied by reasoning.

```
import re
import pandas as pd
import os
def process input string(input string):
    # Extract the iteration number
   iteration number = int(re.findall(r'Iteration (\d+)', input string)[-1]) + 1
    # Extract the observation
    observation = re.findall(r'Your Observation: (.*)', input string, re.DOTALL)[0].strip()
    # Extract the suggested conditions and reasonings
    conditions = re.findall(r'Suggested Condition \d+: {(.*?)}', input string)
    reasonings = re.findall(r'Reasoning \d+: (.*)', input string)
    # Create a list to store the data
    data = []
    # Iterate over the conditions and reasonings
    for condition, reasoning in zip(conditions, reasonings):
        values = [iteration number, observation] + condition.split(', ') + [reasoning]
        data.append(values)
    # Create a DataFrame
    columns = ['Iteration', 'Observation', 'Substrate Concentration', 'Mediator eq.',
'Mediator type', 'Electrolyte type', 'Co solvent HFIP', 'Reasoning']
    df = pd.DataFrame(data, columns=columns)
    # Save the DataFrame to an Excel file
   output file = 'output.xlsx'
   df.to excel(output file, index=False)
   # Open the Excel file
   os.startfile(output file)
    # Generate the text
    text = f"Here is the yield for your iteration {iteration number} \n\n"
    for i, condition in enumerate(conditions, start=1):
        text += f"Iteration {iteration number} Reaction {i}: {{{condition}}} Reaction Yield:
{ {X} }\n\n"
    text +=f"Please suggest {i} new reaction using the provided template. "
    text += """Make sure to include all given reactions in "Previous Reactions and Yields"
and make your reasoning as detailed as possible. Please do not suggest the reaction condition
that has been completed before. Below is the choice of each parameter.
Substrate Concentration (M): 0.025, 0.05, 0.075, 0.1, 0.125
Mediator eq.: 0, 0.25, 0.5, 0.75, 1
Mediator type: NHPI, TCNHPI, QD, DABCO, TEMPO
Electrolyte type: LiClO4, LiOTf , Bu4NClO4, Et4NBF4, Bu4NPF6
Cosolvent HFIP: 0, 1
    ......
    return text
```

**Figure S52.** A parser written by the LLM that can extract the suggestions it made into formulated conditions and generate the user prompt for the next iteration of the conversation. By doing so, the human user will only need to provide the observed yield for each condition, and the prompt will be automatically generated and sent to the LLM for the next round of suggestions.

User: Here is the yield for your iteration 1 Iteration 1 Reaction 1: {0.1, 1, NHPI, LiCIO4, 1} Reaction Yield: {0} Iteration 1 Reaction 2: {0.05, 0.25, DABCO, Bu4NPF6, 0} Reaction Yield: {0} Iteration 1 Reaction 3: {0.125, 0.5, TEMPO, Et4NBF4, 1} Reaction Yield: {0} Iteration 1 Reaction 4: {0.1, 0.75, TCNHPI, LiOTf, 1} Reaction Yield: {17} Please suggest 4 new reactions using the provided template. Make sure to include all given reactions in "Previous Reactions and Yields" and make your reasoning as detailed as possible. Please do not suggest the reaction condition that has been completed before. Below is the choice of each parameter.

Substrate Concentration (M): 0.025, 0.05, 0.075, 0.1, 0.125 Mediator eq.: 0, 0.25, 0.5, 0.75, 1 Mediator type: NHPI, TCNHPI, QD, DABCO, TEMPO Electrolyte type: LiCIO4, LiOTf, Bu4NCIO4, Et4NBF4, Bu4NPF6 Cosolvent HFIP: 0, 1

~~~

Assistant: Previous Reactions and Yields:

Iteration 0 Reaction 1: {0.075, 0.5, QD, Et4NBF4, 1} Reaction Yield: {5} Iteration 0 Reaction 2: {0.075, 0.5, NHPI, LiCIO4, 1} Reaction Yield: {8} Iteration 0 Reaction 3: {0.1, 0.5, TCNHPI, Bu4NPF6, 0} Reaction Yield: {6} Iteration 0 Reaction 4: {0.05, 0.5, TEMPO, LiCIO4, 0} Reaction Yield: {0} Iteration 1 Reaction 1: {0.1, 1, NHPI, LiCIO4, 1} Reaction Yield: {0} Iteration 1 Reaction 2: {0.05, 0.25, DABCO, Bu4NPF6, 0} Reaction Yield: {0} Iteration 1 Reaction 3: {0.125, 0.5, TEMPO, Et4NBF4, 1} Reaction Yield: {0} Iteration 1 Reaction 4: {0.1, 0.75, TCNHPI, LiOTf, 1} Reaction Yield: {17}

Your Observation: The highest yield obtained so far is 17% with TCNHPI, suggesting that this mediator might have promising potential under the right conditions. Most other setups resulted in zero yield, indicating either incompatible mediator -electrolyte combinations or issues with the parameters like concentration or mediator equivalence. It's noteworthy that the only setups yielding results involved either NHPI or TCNHPI. This could imply specific mediator and electrolyte interactions that are beneficial, or the need for optimal mediator equivalence and substrate concentrations.

## Suggested Condition 1: {0.075, 1, TCNHPI, Bu4NPF6, 1}

Reasoning 1: Doubling down on the combination of TCNHPI with another high-performance electrolyte, Bu4NPF6, and using HFIP which may provide a better medium for the reaction. Higher mediator equivalence and consistent substrate concentration from previous effective experiments might enhance yield.

Suggested Condition 2: {0.125, 0.75, NHPI, LiOTf, 0}

Reasoning 2: NHPI previously showed some activity; increasing the substrate concentration to the maximum might boost its performance. Using LiOTf as the electrolyte could provide a different ionic influence, and adjusting mediator equivalence to 0.75 might strike a balance between reactivity and stability.

Suggested Condition 3: {0.025, 0.5, QD, Et4NBF4, 1}

Reasoning 3: Lowering the substrate concentration could potentially increase the relative impact of the mediator, and using QD in the presence of HFIP with a previously used effective electrolyte might provide insights into the optimal operating range for QD.

Suggested Condition 4: {0.05, 1, DABCO, LiClO4, 1}

Reasoning 4: Exploring the upper limit of mediator equivalence for DABCO, which had not been tested at higher equivalence levels. LiCIO4 was selected due to previous modest success with NHPI, and including HFIP might stabilize or facilitate the reaction process, potentially uncovering hidden synergies.

**Figure S53.** A representative user prompt generated by the parser and sent to the LLM to request suggestions for the next round, accompanied by the corresponding answer from the LLM. The complete history of all conversations for all experiments in this study is provided in Spreadsheet SF4. EChem Reaction Optimization Dataset in the Supporting Information.

User: Here is the yield for your iteration 2 Iteration 2 Reaction 1: {0.075, 1, TCNHPI, Bu4NPF6, 1} Reaction Yield: {21} Iteration 2 Reaction 2: {0.125, 0.75, NHPI, LiOTf, 0} Reaction Yield: {5} Iteration 2 Reaction 3: {0.025, 0.5, QD, Et4NBF4, 1} Reaction Yield: {5} Iteration 2 Reaction 4: {0.05, 1, DABCO, LiClO4, 1} Reaction Yield: {0} Please suggest 4 new reaction using the provided template. Make sure to include all given reactions in "Previous Reactions and Yields" and make your reasoning as detailed as possible. Please do not suggest the reaction condition that has been completed before. Below are the choice of each parameter.

Substrate Concentration (M): 0.025, 0.05, 0.075, 0.1, 0.125 Mediator eq.: 0, 0.25, 0.5, 0.75, 1 Mediator type: NHPI, TCNHPI, QD, DABCO, TEMPO Electrolyte type: LiCIO4, LiOTf, Bu4NCIO4, Et4NBF4, Bu4NPF6 Cosolvent HFIP: 0, 1

Assistant: Previous Reactions and Yields: Iteration 0 Reaction 1: {0.075, 0.5, QD, Et4NBF4, 1} Reaction Yield: {5} Iteration 0 Reaction 2: {0.075, 0.5, NHPI, LiCIO4, 1} Reaction Yield: {8} Iteration 0 Reaction 3: {0.1, 0.5, TCNHPI, Bu4NPF6, 0} Reaction Yield: {6} Iteration 0 Reaction 4: {0.05, 0.5, TEMPO, LiCIO4, 0} Reaction Yield: {0} Iteration 1 Reaction 1: {0.1, 1, NHPI, LiCIO4, 1} Reaction Yield: {0} Iteration 1 Reaction 2: {0.05, 0.25, DABCO, Bu4NPF6, 0} Reaction Yield: {0} Iteration 1 Reaction 3: {0.125, 0.5, TEMPO, Et4NBF4, 1} Reaction Yield: {0} Iteration 1 Reaction 4: {0.1, 0.75, TCNHPI, LiOTf, 1} Reaction Yield: {17} Iteration 2 Reaction 1: {0.075, 1, TCNHPI, Bu4NPF6, 1} Reaction Yield: {21} Iteration 2 Reaction 2: {0.125, 0.5, NHPI, LiOTf, 0} Reaction Yield: {5} Iteration 2 Reaction 3: {0.025, 0.5, QD, Et4NBF4, 1} Reaction Yield: {5} Iteration 2 Reaction 4: {0.05, 1, DABCO, LiCIO4, 1} Reaction Yield: {0}

Your Observation: TCNHPI continues to show promising results, especially with higher mediator equivalence and the use of HFIP. There seems to be a trend where higher substrate concentrations or higher mediator equivalents generally result in increased yields, particularly when combined with certain electrolytes like Bu4NPF6. Notably, experiments with HFIP tend to yield better results, suggesting its significant role in stabilizing or activating the reaction environment.

Suggested Condition 1: {0.1, 0.75, NHPI, Bu4NClO4, 1}

Reasoning 1: Building on the modest success of NHPI at high substrate concentration, pairing it with Bu4NCIO4 might enhance ionic interactions, potentially improving the reaction's efficiency. Using HFIP could further assist in stabilizing the active species or improving solubility.

Suggested Condition 2: {0.125, 1, TCNHPI, LiClO4, 1}

Reasoning 2: Maximizing both the substrate concentration and mediator equivalence for TCNHPI, which has shown the highest yields. LiClO4 is revisited with HFIP to explore whether the ionic environment could further enhance yields under these maximized conditions.

Suggested Condition 3: {0.075, 0.5, TEMPO, Bu4NPF6, 1} Reasoning 3: Re-testing TEMPO with different conditions, given the zero yield previously. A lower mediator equivalence and the inclusion of HFIP, combined with a well-performing electrolyte (Bu4NPF6), might alter the dynamics favorably.

Suggested Condition 4: {0.05, 0.25, QD, LiOTf, 1}

Reasoning 4: Lower substrate and mediator levels might optimize QD's performance in a different electrolyte environment. LiOT f could provide an interesting contrast to Et4NBF4, and including HFIP might enhance the mediator's activity or stability in the reaction system.

**Figure S54.** A representative user prompt generated by the parser and sent to the LLM to request suggestions for the next round, accompanied by the corresponding answer from the LLM.

```
import pandas as pd
import numpy as np
from itertools import product
from skopt import Optimizer
from skopt.space import Integer, Categorical
# Generate the entire search space and save to CSV
def generate search space():
   substrate concentration = [0.025, 0.05, 0.075, 0.1, 0.125]
   mediator eq = [0, 0.25, 0.5, 0.75, 1]
   mediator types = ['NHPI', 'TCNHPI', 'QD', 'DABCO', 'TEMPO']
   electrolyte_types = ['LiClO4', 'LiOTf', 'Bu4NClO4', 'Et4NBF4', 'Bu4NPF6']
   co solvent = [0, 1]
   all combinations = list(product(substrate concentration, mediator eq, mediator types,
electrolyte types, co solvent))
   df = pd.DataFrame (all combinations, columns=['Substrate Concentration', 'Mediator eq',
'Mediator Type', 'Electrolyte Type', 'Co-solvent HFIP'])
   df.to csv('echem search space.csv', index=False)
   initialize experiments(df)
   print('echem search space.csv generated')
# Initialize the experiment csv
def initialize experiments(df):
   df['Yield'] = 'PENDING'
   df['Priority'] = 0
    # Randomly select 3/4/5 conditions to start
   initial indices = np.random.choice(df.index, size=3, replace=False)
   df.loc[initial indices, 'Priority'] = 1
   df.sort values('Priority', ascending=False, inplace=True)
   df.to csv('echem reaction opt round0.csv', index=False)
# Function to update yield for multiple experiments
def update yield(csv file, row numbers, yield values):
   df = pd.read csv(csv file)
    for row num, yield val in zip(row numbers, yield values):
        if df.at[row num, 'Priority'] == 1:
            df.at[row num, 'Yield'] = yield val
    df.to csv(csv file, index=False)
# Function for reaction optimization
def reaction optimization (csv file):
    df = pd.read csv(csv file)
    if df[df['Priority'] == 1]['Yield'].eq('PENDING').any():
        raise ValueError ("Some experiments with Priority 1 have not completed yet.")
   df.loc[df['Priority'] == 1, 'Priority'] = -1
    # Setup the optimizer
   dimensions = [Integer(1, 5), Integer(0, 4), Categorical(['NHPI', 'TCNHPI', 'QD', 'DABCO',
'TEMPO']),
                  Categorical(['LiCl04', 'LiOTf', 'Bu4NCl04', 'Et4NBF4', 'Bu4NPF6']),
Integer(0, 1)]
   optimizer = Optimizer(dimensions, base estimator="GP", acq func="EI")
    # Encode completed experiments
    completed = df[df['Priority'] == -1]
```

```
for , row in completed.iterrows():
        encoded = [int(row['Substrate Concentration']/0.025), int(row['Mediator eq']/0.25),
                  row['Mediator Type'], row['Electrolyte Type'], row['Co-solvent HFIP']]
        optimizer.tell(encoded, -float(row['Yield'])) # negative yield because we minimize
in optimization
    # Get suggestions
    suggested points = optimizer.ask(n points=3)
    for point in suggested points:
        decoded = [point[0] * 0.025, point[1] * 0.25, point[2], point[3], point[4]]
        existing = df[(df['Substrate Concentration'] == decoded[0]) & (df['Mediator eq'] ==
decoded[1]) &
                      (df['Mediator Type'] == decoded[2]) & (df['Electrolyte Type'] ==
decoded[3]) &
                      (df['Co-solvent HFIP'] == decoded[4])]
        if existing.empty or existing.iloc[0]['Priority'] == 0:
            row idx = existing.index
            if len(row idx) > 0:
                df.at[row idx[0], 'Priority'] = 1
                df.at[row idx[0], 'Yield'] = 'PENDING'
                print ("Added")
    # Prepare new CSV for next round
    new round number = int(csv file.split('round') [1].split('.')[0]) + 1
    new csv file = f'echem reaction opt round {new round number}.csv'
    df.sort values('Priority', ascending=False, inplace=True)
    df.to csv(new csv file, index=False)
    return new csv file
```

**Figure S55.** LLM-generated Python code designed for function calling to assist in reaction optimization. This code uses *skopt* as an external machine learning tool to suggest reaction conditions based on user input and prior experimental data. This helper function can be integrated into the LLM as an external tool using detailed methods reported in previous literature.<sup>5</sup> Briefly, the helper function's name, a description of each paragraph, and its return content are provided. The LLM is informed of the function's existence and its potential as a tool. Based on the user prompt, the LLM decides whether to use this tool, resulting in the extraction of tool input, execution of code, and return of results. Finally, the LLM uses the tool's results to formulate its response. Further details and examples are available at platform.openai.com/docs/guides/function-calling, https://docs.llama-api.com/essentials/function and docs.anthropic.com/en/docs/build-with-claude/tool-use.

```
"name": "generate search space",
    "description": "Generate the search space for reaction optimization and save it to a CSV
file.",
    "parameters": {
         "type": "object",
        "properties": {}
    },
    "required": []
}
{
    "name": "update yield",
    "description": "Update the yields for specific reactions in the given CSV file.",
    "parameters": {
    "type": "object",
         "properties": {
             "csv_file": {
    "type": "string",
                  "description": "The path to the CSV file where yields are recorded."
             },
             "row_numbers": {
    "type": "array",
                  "items": { "type": "integer" },
                  "description": "The row numbers of the experiments to update."
             },
"yield_values": {
    "arra"
                 "type": "array",
"items": { "type": "number" },
                  "description": "The yield values to update in the corresponding rows."
             }
         },
         "required": ["csv_file", "row_numbers", "yield_values"]
    }
}
{
    "name": "reaction_optimization",
    "description": "Perform reaction optimization using skopt and suggest new reaction
conditions.",
    "parameters": {
         "type": "object",
"properties": {
             "csv file": {
                  "type": "string",
                  "description": "The path to the CSV file containing reaction data for
optimization."
             }
         },
         "required": ["csv file"]
    }
}
```

Figure S56. Example of JSON schema for function calling.

```
from edbo.plus.optimizer_botorch import EDBOplus
import pandas as pd
def initialize experiment (reaction components, filename, batch size=4):
    # Generate the reaction scope and save it to a CSV file
    EDBOplus().generate reaction scope(
        components=reaction components, # Components used to generate the scope
        filename=filename,# Filename for saving the generated scopecheck_overwrite=False# Avoid overwriting existing files
    )
    # Run the optimizer to suggest initial experiments
    EDBOplus().run(
                                           # Use the generated scope from the filename
        filename=filename,
        filename=filename,  # Use the generated scope from the filename
objectives=['yield'],  # Set 'yield' as the objective to optimize
objective_mode=['max'],  # Maximize the objective (yield)
batch=batch_size,  # Set the number of experiments per batch
         columns features='all',  # Use all columns as features
        init_sampling_method='cvtsampling' # Initialization method for sampling
    )
    # Load and display the first few experiments
    df init = pd.read csv(filename)
    return df init.head(batch size)
def update experiments with yield (filename, conditions, observed yields, batch size=4):
    # Load the existing reaction scope
    df = pd.read csv(filename)
    # Update the DataFrame with observed yields
    for condition, yield val in zip(conditions, observed yields):
        mask = (
             (df['substrate concentration'] == condition[0]) &
             (df['mediator eq'] == condition[1]) &
             (df['mediator type'] == condition[2]) &
             (df['electrolyte type'] == condition[3]) &
             (df['co solvent HFIP'] == condition[4])
         )
        df.loc[mask, 'yield'] = yield val # Update the yield for the experiment
         df.loc[mask, 'priority'] = -1  # Mark as completed
    # Save the updated scope
    updated filename = 'updated ' + filename
    df.to csv(updated filename, index=False)
    # Run the optimizer using the updated scope to suggest the next experiments
    EDBOplus().run(
        filename=updated_filename,  # Use the updated scope
objectives=['yield'],  # Objective to be optimized
        objectives=['yield'],
objective_mode=['max'],
                                             # Objective mode to maximize yield
        objective_mode=['max'],
                                                # Number of experiments to suggest
        batch=batch size,
         columns features='all', # Features to include in the model
        init sampling method='cvtsampling' # Sampling method for initialization
    # Load and display the suggested next experiments
    df next = pd.read csv(updated filename)
    return df next.head(batch size)
```

**Figure S57.** LLM-generated Python code for a helper function to use *edbo*<sup>22,31</sup> as the ML tool to make suggestions based on user input.



Figure S59. <sup>13</sup>C NMR spectrum of 1b (126 MHz, CDCl<sub>3</sub>).



Figure S60. <sup>1</sup>H NMR spectrum of 2b (500 MHz, CDCl<sub>3</sub>).



Figure S61. <sup>13</sup>C NMR spectrum of 2b (126 MHz, CDCl<sub>3</sub>).



Figure S63. <sup>13</sup>C NMR spectrum of 3b (126 MHz, CDCl<sub>3</sub>).





Figure S65. <sup>13</sup>C NMR spectrum of 4b (126 MHz, CDCl<sub>3</sub>).





Figure S67. <sup>13</sup>C NMR spectrum of 5b (126 MHz, CDCl<sub>3</sub>).



Figure S69. <sup>13</sup>C NMR spectrum of 6b (126 MHz, CDCl<sub>3</sub>).




Figure S71. <sup>13</sup>C NMR spectrum of 7b (126 MHz, CDCl<sub>3</sub>).





Figure S73. <sup>13</sup>C NMR spectrum of 8b (126 MHz, CDCl<sub>3</sub>).

## References

- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. arXiv August 1, 2023. https://doi.org/10.48550/arXiv.1706.03762.
- (2) Zahrt, A. F.; Mo, Y.; Nandiwale, K. Y.; Shprints, R.; Heid, E.; Jensen, K. F. Machine-Learning-Guided Discovery of Electrochemical Reactions. J. Am. Chem. Soc. 2022, 144 (49), 22599–22610. https://doi.org/10.1021/jacs.2c08997.
- (3) Zheng, Z.; Rong, Z.; Rampal, N.; Borgs, C.; Chayes, J. T.; Yaghi, O. M. A GPT-4 Reticular Chemist for Guiding MOF Discovery. *Angew. Chem. Int. Ed.* **2023**, *62* (46), e202311983. https://doi.org/10.1002/anie.202311983.
- (4) Zheng, Z.; Zhang, O.; Borgs, C.; Chayes, J. T.; Yaghi, O. M. ChatGPT Chemistry Assistant for Text Mining and the Prediction of MOF Synthesis. *J. Am. Chem. Soc.* **2023**, *145* (32), 18048–18062. https://doi.org/10.1021/jacs.3c05819.
- (5) Zheng, Z.; Alawadhi, A. H.; Chheda, S.; Neumann, S. E.; Rampal, N.; Liu, S.; Nguyen, H. L.; Lin, Y.; Rong, Z.; Siepmann, J. I.; Gagliardi, L.; Anandkumar, A.; Borgs, C.; Chayes, J. T.; Yaghi, O. M. Shaping the Water-Harvesting Behavior of Metal–Organic Frameworks Aided by Fine-Tuned GPT Models. *J. Am. Chem. Soc.* **2023**, *145* (51), 28284–28295. https://doi.org/10.1021/jacs.3c12086.
- (6) Rani, S.; Aslam, S.; Lal, K.; Noreen, S.; Alsader, K. A. M.; Hussain, R.; Shirinfar, B.; Ahmed, N. Electrochemical C-H/C-C Bond Oxygenation: A Potential Technology for Plastic Depolymerization. *Chem. Rec.* 2024, 24 (3), e202300331. https://doi.org/10.1002/tcr.202300331.
- (7) Zheng, Z.; He, Z.; Khattab, O.; Rampal, N.; Zaharia, M. A.; Borgs, C.; Chayes, J. T.; Yaghi, O. M. Image and Data Mining in Reticular Chemistry Powered by GPT-4V. *Digit. Discov.* 2024. https://doi.org/10.1039/D3DD00239J.
- (8) Yang, Z.; Li, L.; Lin, K.; Wang, J.; Lin, C.-C.; Liu, Z.; Wang, L. The Dawn of LMMs: Preliminary Explorations with GPT-4V(Ision). arXiv October 11, 2023. https://doi.org/10.48550/arXiv.2309.17421.
- (9) Esrafili, M. D.; Sharifi, F.; Dinparast, L. Catalytic Hydrogenation of CO2 over Pt- and Ni-Doped Graphene: A Comparative DFT Study. J. Mol. Graph. Model. 2017, 77, 143–152. https://doi.org/10.1016/j.jmgm.2017.08.016.
- (10) Zhang, W.; Wang, Q.; Kong, X.; Xiong, J.; Ni, S.; Cao, D.; Niu, B.; Chen, M.; Li, Y.; Zhang, R.; Wang, Y.; Zhang, L.; Li, X.; Xiong, Z.; Shi, Q.; Huang, Z.; Fu, Z.; Zheng, M. Fine-Tuning Large Language Models for Chemical Text Mining. *Chem. Sci.* **2024**, *15* (27), 10600–10611. https://doi.org/10.1039/D4SC00924J.
- (11) Leong, S. X.; Pablo-García, S.; Zhang, Z.; Aspuru-Guzik, A. Automated Electrosynthesis Reaction Mining with Multimodal Large Language Models (MLLMs). *ChemRxiv Prepr.* 2024. https://doi.org/10.26434/chemrxiv-2024-7fwxv.
- (12) Ai, Q.; Meng, F.; Shi, J.; Pelkie, B.; Coley, C. W. Extracting Structured Data from Organic Synthesis Procedures Using a Fine-Tuned Large Language Model. *ChemRxiv Prepr.* **2024**. https://doi.org/10.26434/chemrxiv-2024-979fz.
- (13) Regio-selectivity prediction with a machine-learned reaction representation and on-the-fly quantum mechanical descriptors Chemical Science (RSC Publishing). https://pubs.rsc.org/en/content/articlelanding/2021/sc/d0sc04823b (accessed 2024-07-02).

- (14) Li, S.-C.; Wu, H.; Menon, A.; Spiekermann, K.; Li, Y.-P.; Green, W. When Do Quantum Mechanical Descriptors Help Graph Neural Networks Predict Chemical Properties? ChemRxiv July 1, 2024. https://doi.org/10.26434/chemrxiv-2024-7q438-v3.
- (15) Halgren, T. A. Merck Molecular Force Field. I. Basis, Form, Scope, Parameterization, and Performance of MMFF94. J. Comput. Chem. 1996, 17 (5–6), 490–519. https://doi.org/10.1002/(SICI)1096-987X(199604)17:5/6<490::AID-JCC1>3.0.CO;2-P.
- (16) RDKit. https://www.rdkit.org/ (accessed 2024-07-02).
- (17) Bannwarth, C.; Ehlert, S.; Grimme, S. GFN2-xTB—An Accurate and Broadly Parametrized Self-Consistent Tight-Binding Quantum Chemical Method with Multipole Electrostatics and Density-Dependent Dispersion Contributions. J. Chem. Theory Comput. 2019, 15 (3), 1652– 1671. https://doi.org/10.1021/acs.jctc.8b01176.
- (18) Chai, J.-D.; Head-Gordon, M. Long-Range Corrected Hybrid Density Functionals with Damped Atom–Atom Dispersion Corrections. *Phys. Chem. Chem. Phys.* **2008**, *10* (44), 6615. https://doi.org/10.1039/b810189b.
- (19) Weigend, F.; Ahlrichs, R. Balanced Basis Sets of Split Valence, Triple Zeta Valence and Quadruple Zeta Valence Quality for H to Rn: Design and Assessment of Accuracy. *Phys. Chem. Chem. Phys.* **2005**, 7 (18), 3297–3305. https://doi.org/10.1039/B508541A.
- (20) Gaussian 16 Rev. C.01/C.02 Release Notes / Gaussian.com. https://gaussian.com/relnotes/ (accessed 2024-07-02).
- (21) Heid, E.; Green, W. H. Machine Learning of Reaction Properties via Learned Representations of the Condensed Graph of Reaction. *J. Chem. Inf. Model.* **2022**, *62* (9), 2101–2110. https://doi.org/10.1021/acs.jcim.1c00975.
- (22) Torres, J. A. G.; Lau, S. H.; Anchuri, P.; Stevens, J. M.; Tabora, J. E.; Li, J.; Borovika, A.; Adams, R. P.; Doyle, A. G. A Multi-Objective Active Learning Platform and Web App for Reaction Optimization. *J. Am. Chem. Soc.* **2022**, *144* (43), 19999–20007. https://doi.org/10.1021/jacs.2c08592.
- (23) Zheng, Z.; Zhang, O.; Nguyen, H. L.; Rampal, N.; Alawadhi, A. H.; Rong, Z.; Head-Gordon, T.; Borgs, C.; Chayes, J. T.; Yaghi, O. M. ChatGPT Research Group for Optimizing the Crystallinity of MOFs and COFs. ACS Cent. Sci. 2023, 9 (11), 2161–2170. https://doi.org/10.1021/acscentsci.3c01087.
- (24) Horn, E. J.; Rosen, B. R.; Chen, Y.; Tang, J.; Chen, K.; Eastgate, M. D.; Baran, P. S. Scalable and Sustainable Electrochemical Allylic C–H Oxidation. *Nature* **2016**, *533* (7601), 77–81. https://doi.org/10.1038/nature17431.
- (25) Yu, J.-Q.; Corey, E. J. A Mild, Catalytic, and Highly Selective Method for the Oxidation of  $\alpha,\beta$ -Enones to 1,4-Enediones. *J. Am. Chem. Soc.* **2003**, *125* (11), 3232–3233. https://doi.org/10.1021/ja0340735.
- (26) Khachik, F.; Chang, A.-N. Total Synthesis of (3R,3'R,6'R)-Lutein and Its Stereoisomers. J. Org. Chem. 2009, 74 (10), 3875–3885. https://doi.org/10.1021/jo900432r.
- (27) Jiang, Y.; Chen, S.; Chen, Y.; Gu, A.; Tang, C. Sustainable Aerobic Allylic C–H Bond Oxidation with Heterogeneous Iron Catalyst. J. Am. Chem. Soc. 2024, 146 (4), 2769–2778. https://doi.org/10.1021/jacs.3c12688.
- (28) Liu, C.; Liu, H.; Zheng, X.; Chen, S.; Lai, Q.; Zheng, C.; Huang, M.; Cai, K.; Cai, Z.; Cai, S. Visible-Light-Enabled Allylic C–H Oxidation: Metal-Free Photocatalytic Generation of Enones. ACS Catal. 2022, 12 (2), 1375–1381. https://doi.org/10.1021/acscatal.1c05124.

- (29) Schweitzer-Chaput, B.; Demaerel, J.; Engler, H.; Klussmann, M. Acid-Catalyzed Oxidative Radical Addition of Ketones to Olefins. *Angew. Chem. Int. Ed.* **2014**, *53* (33), 8737–8740. https://doi.org/10.1002/anie.201401062.
- (30) Kawamata, Y.; Yan, M.; Liu, Z.; Bao, D.-H.; Chen, J.; Starr, J. T.; Baran, P. S. Scalable, Electrochemical Oxidation of Unactivated C–H Bonds. J. Am. Chem. Soc. 2017, 139 (22), 7448–7451. https://doi.org/10.1021/jacs.7b03539.
- (31) Shields, B. J.; Stevens, J.; Li, J.; Parasram, M.; Damani, F.; Alvarado, J. I. M.; Janey, J. M.; Adams, R. P.; Doyle, A. G. Bayesian Reaction Optimization as a Tool for Chemical Synthesis. *Nature* 2021, 590 (7844), 89–96. https://doi.org/10.1038/s41586-021-03213-y.