Supplementary Information (SI) for Digital Discovery.
This journal is © The Royal Society of Chemistry 2024

Supplementary Information:

Knowledge Discovery from Porous Organic Cages Literature Using a Large Language Model

Yaoyi Su,[a] Siyuan Yang,[a,b] Yuanhan Liu,[a] Aiting Kai,[a,b] Linjiang Chen,[c,d] and Ming Liu [a,b]**

a. Department of Chemistry, Zhejiang University, Hangzhou, Zhejiang 310058, China.

b. ZJU-Hangzhou Global Scientific and Technological Inovation Center, Zhejiang University, Hangzhou, Zhejiang 311200, China

c. Key Laboratory of Precision and Intelligent Chemistry, University of Science and Technology of China, Hefei, Anhui 230026, China

d. School of Chemistry, School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK

*Corresponding Authors: mingliu@zju.edu.cn , linjiangchen@ustc.edu.cn

Table of Contents

S1. General Information
S2. Prompt Engineering
S3. Python Code
S4. Detailed experimental data
S5. References

Section S1. General Information

Raw data, the resulting database of POCs and source codes including directly runnable chatbot are available at the following URL: https://github.com/syy1213/LLMs-GPT-4-Cage. We have encapsulated our Python code into a Docker-based container, preserving the necessary Python packages required for reproducing our work. You can access this resource on https://hub.docker.com/r/syy12137059/cage_gpt/tags.

Large Language Models

Building on the success of its predecessors, GPT-4 exhibits even more refined capabilities in natural language understanding and generation.[1] We use the GPT-4 for text classification. For extraction of tabular information and construction of interactive chatbots, we use GPT-4o, a recently released model from OpenAI.

The GPT was used on 26 June 2024. It is necessary to record this date, as the performance of the model may undergo changes over time.

Section S2. Prompt Engineering

Prompt for text classification

System Prompt
You are a text classifier that classifies paragraphs into seven categories based on what they contain. 1. this paper's author: Containing information about the author of this paper (There are many consecutively occurring names, the last of which must be followed by an asterisk). Generally this category of paragraph appears earlier in the article. 2. extra author: Containing authors of other articles, such as background descriptions. 3. comprehensive synthesis: Containing comprehensive experimental conditions of the chemical reaction. The chemical reaction conditions must appear with clear information about the reaction temperature, reaction time, reactants, products, solvents, and their amounts. Note in particular that the amount is mandatory. This means that if neither volume, weight nor molarity is mentioned in the text, it cannot be attributed to this category. 4. incomprehensive synthesis: Containing incomprehensive experimental conditions of the chemical reaction. (also a reaction condition, but without very complete information) 5. CCDC: Containing CCDC number (The text generally contains the word "CCDC" followed by a string of numbers.) 6. surface area: Containing information on the specific surface area of a compound. (The text generally contains the word "m2/g".) 7. affiliation: Containing information about the author's organization, city, or nationality, etc. 8. Ref: Containing references. 9. others: paragraphs which fall into none of the previously mentioned categories.

Some examples of paragraphs are given below, along with the corresponding categories, so that you can study them in order to later classify the paragraphs given to you.

paragraph:
Yinghua Jin,[†] Bret A. Voss,[‡] Athena Jin,[†] Hai Long,[§] Richard D. Noble,[‡] and Wei Zhang[*],
†† Department of Chemistry and Biochemistry and ‡ Department of Chemical and Biological Engineering, University of Colorado, Boulder, Colorado 80309, United States § National Renewable Energy Laboratory, Golden, Colorado 80401, United States
category: this paper's author

paragraph:
Avinash S. Bhat, Sven M. Elbert, Wen-Shan Zhang, Frank Rominger, Michael Dieckmann, Rasmus R. Schrçder, and Michael Mastalerz[*]
category: this paper's author

paragraph:
a Department of Chemistry, Imperial College London, South Kensington, London, SW7 2AZ, UK. E-mail: k.jelfs@imperial.ac.ukb Department of Chemistry and Materials Innovation Factory, University of Liverpool, 51 Oxford Street, Liverpool, L7 3NY, UK
category: this paper's author

paragraph:
Enrico Berardo, [‡a] Rebecca L. Greenaway, [‡b] Lukas Turcani, [a] Ben M. Alston, [b] Michael J. Bennison, [b] Marcin Miklitz, [a] Rob Clowes, [b] Michael E. Briggs, [b] Andrew I. Cooper [b] and Kim E. Jelfs [*]
category: this paper's author

paragraph:
[a] J. C. Lauer, Dr. F. Rominger, Prof. Dr. M. Mastalerz Organisch-Chemisches Institut Ruprecht-Karls-Universiteit Heidelberg Im Neuenheimer Feld 270 69120 Heidelberg, Germany E-mail: michael.mastalerz@oci.uni-heidelberg.de [b] Dr. Wen-Shan Zhang, Prof. Dr. Rasmus R. Schroder Centre for Advanced Materials Ruprecht-Karls -Universiteit Heidelberg Im Neuenheimer Feld 225 69120 Heidelberg, Germany
category: this paper's author

paragraph:
For instance, Cooper et al. studied the effect of connecting aliphatic tethers of different lengths of diamines in the imine condensation with aromatic trialdehydes.
category: extra author

paragraph:
Sindelar and Beer independently employed water-soluble macrocyclic bambusuril or rotaxane to recognize a variety of anions in water.
category: extra author

paragraph:
Feigel et al. employed amino acid precursors in synthesizing a trefoil knot, whose formation is driven by hydrogen bonding between the amide building blocks.
category: extra author

paragraph:
Mukherjee et al. have recently observed a self-selection process between multiple structural isomers when an unsymmetrical ditopic building block was employed for the synthesis of imine cages, but such processes are very hard to anticipate. In the area of frameworks, Zhang et al. showed that it was possible to obtain COFs with heterogeneous pore structures by using a C_{2v} tritopic building block
category: extra author

paragraph:
The cage is significantly larger than our previous largest example CC5 (see comparison in Fig. S1) and approximately the same size as the highly porous adamantoid cage reported by Mastalerz et al. (2.9 nm vs 2.84 nm diameter).
category: extra author

paragraph:
C4RACHO (0.10 mmol, 82 mg) and 1,3-diaminopropane (0.2 mmol, 14.8 mg) were added to $PhNO_2$ (6mL). The mixture was sealed in a 20 mL glass vial, which was heated at 100 °C for 3 days. Red block single crystals of CPOC-101 were then obtained with ~ 38% yield and washed with MeOH.
category: comprehensive synthesis

paragraph:
In a 20 mL glass vial 5 mL ethanolic solution of 1,3,5-tris(aminomethyl)-2,4,6-trimethylbenzene (X, 2.1 mg, 0.010 mmol) was added drop wise to the solution of 1,3-bis(4-formylphenylethynyl)benzene (A, 5.0 mg, 0.015 mmol) in a mixture of 5mL ethanol and 1 mL chloroform with stirring at room temperature.
category: comprehensive synthesis

paragraph:
To a suspension of BTDP (40.0 mg, 0.10 mmol) in dry dichloromethane (20.0 mL), a solution of chiral (R,R)-cyclohexanediamine (22.8 mg, 0.20 mmol) in dry dichloromethane (10.0 mL) was added slowly. The mixture was stirred at room temperature for 5 days. Then the reaction mixture was filtered, and organic phase was evaporated under reduced pressure. The yellow product was obtained by the recrystallization of crude product in dichloromethane and methanol, in a yield of 72% (40.0 mg).
category: comprehensive synthesis

paragraph:

Syntheses of cage 7 under different reaction conditions 2-hydroxy -1,3,5 -triformylbenzene (178 mg, 1 mmol) and (1R,2R)-diphenylethylene diamine (318 mg, 1.5 mmol) were dissolved in appropriate mixture of solvents (total volume = 50 mL). In each case, the resulting mixture was stirred under argon atmosphere at room temperature for 14 days. The formed yellow precipitate was filtered off, dried and then activated.
category: comprehensive synthesis

paragraph:
Trifluoroacetic acid (26.9 mg, 0.236 mmol, 0.05 equiv.) was charged to a mixture of the (1R,2R)-(+)-1,2-diphenylethylenediamine (1.50 g, 7.06 mmol, 1.5 equiv.) and 1,3,5-triformylbenzene (0.764 g, 4.71 mmol, 1.0 equiv.) in dichloromethane (60 ml) and the mixture was stirred at 15 ºC for 50 h.
category: comprehensive synthesis

paragraph:
Reduction of CC7. An oven-dried round-bottom flask (500 mL) was cooled down to room temperature under N_2; to this was added CC7(350 mg, 0.1 mmol), $CHCl_3$ (30 mL), and MeOH (150 mL). The resulting suspension was stirred at room temperature for 10 min, $NaBH_4$ (185 mg, 50 mmol) was added slowly. The reaction mixture was then stirred at room temperature under N_2 overnight.
category: comprehensive synthesis

paragraph:
Synthesis of Pd@CC1r Catalyst. In a typical synthetic protocol, 40mg (0.03 mmol) of CC1r dissolved in 6 mL of $CHCl_3$ was treated with 4 mL of a $CHCl_3$ solution of 20 mg (0.09 mmol) of $Pd(OAc)_2$ for 1 h. Into this reaction mixture was added a methanolic solution of $NaBH_4$ (24.3 mg, 0.18 mmol) dropwise at room temperature, followed by stirring for 20 min.
category: comprehensive synthesis

paragraph:
The cages were obtained through [8+12] imine condensation reactions between two relatively simple precursors, tris(4-formylphenyl)amine (Scheme 1, A) and the chiral diamines (R,R)-1,2-cyclohexanediamine (B1) and (R,R)-1,2-cyclohex-4-enediamine (B2) to yield CC7 and CC8, respectively. The reaction occurs without any additional template or catalyst (Scheme 1)to generate the A_8B_{12} cage in good yield (85 – 90%) through formation of 24 new imine bonds.
category: incomprehensive synthesis

paragraph:
To achieve this first synthetic goal, hexamethoxytriptycene 1 was partially oxidized to the quinone 2 and condensed with phenylene diamine 3 to give phenazine 4 in 79% yield.
category: incomprehensive synthesis

paragraph:
CC20 was obtained by the self-assembly of 8 molecules of 2-hydroxy-1,3,5-triformylbenzene

(HO-TFB) with 12 molecules of cis-1,3-CHDA.
category: incomprehensive synthesis

paragraph:
The organic cages are formed by reacting 1,3,5-triformylbenzene with different vicinal diamines (Scheme 1a). For CC2, CC3 and CC4, the chirality of the diamine controls the helical chirality of the cage molecule, whereas CC1 and CC13 are synthesised using achiral diamines and prepared as helical racemates.
category: incomprehensive synthesis

paragraph:
We report herein a [2+3] phosphate organic cage (cage 1) via the Schiff-base condensation between a phosphate based trialdehyde with a unique tetrahedron geometry and (1R,2R)-(-)-1,2-cyclohexanediamine (Scheme 1).
category: incomprehensive synthesis

paragraph:
One can also carry out chemistry on these systems: for example, CC7 can be cleanly reduced to its amine form at room temperature using $NaBH_4$ in a mixture of $CHCl_3$ and MeOH, as supported by NMR spectroscopy and mass spectrometry (Figures S29 and S30).
category: incomprehensive synthesis

paragraph:
S51 Crystal structure of compound 3-Et-H CCDC number : 1588273 Solvent: CH_2Cl_2 Table 2: Crystal data and structure refinement for jol8.
category: CCDC

paragraph:
E-mail: mengxianggao@ccnu.edu.cnd Technology R&D Center, Hubei Tobacco (Group) Co., Ltd, Wuhan, 430070, China. E-mail: zhangyh@hbtobacco.cn Electronic supplementary information (ESI) available: Synthesis details, NMRdata, crystal data, fluorescence and UV data and FTIR. CCDC 2077306 and 2077307.
category: CCDC

paragraph:
X-ray data have been deposited at the Cambridge Crystallographic Data Centre (CCDC 948612 contains the supplementary crystallographic data for this paper. These data can be obtained free of charge from The Cambridge Crystallographic Data Centre.
category: CCDC

paragraph:
Crystallographic data have been submitted to the Cambridge Crystallographic Database with reference numbers CCDC 707056 (1, EtOAc solvate), CCDC 720848 (1, desolvated), CCDC 720849 (2, desolvated), CCDC 720851 (3, $CHCl_3$ solvate) and CCDC 720850 (3, desolvated, data

collected at 400K) and are available free of charge at www.ccdc.cam.ac.uk/data_request/cif.
category: CCDC

paragraph:
The apparent BET surface area of the rotary evaporated sample (720 m^2/g) was more than twice as high as the slowly crystallized sample (330 m^2/g).
category: surface area

paragraph:
The Langmuir surface area calculated for 2 from the N_2 isotherm was 600 m^2/g and the Brunauer Emmett Teller surface area (SABET) was 533 m^2/g. Cage 3 also shows a type I N_2 sorption isotherm with a corresponding Langmuir surface area of 730 m^2/g (SABETD 624 m^2/g).
category: surface area

paragraph:
The specific BET surface area is with 443 m^2/g much larger than accessible by thermal activation and comparable to smaller cages e.g. from Cooper et al. , but significantly lower than for larger cages. Furthermore, the material adsorbs 0.93 wt% H_2 (at 77 K and 1 bar), 13.8 wt% CO_2 and 2.17 wt% CH_4 (both at 273 K, 1 bar). All derived polymer materials have been investigated by nitrogen sorption at 77 K, turning out to be of low porosity with specific surface areas below 26 m^2/g.
category: surface area

paragraph:
Therefore, both cage compounds have been investigated by gas sorption. Thermal treatment (120 °C for 3 h) under vacuum led not to a reasonable activation of the pores. In each case, the apparent specific surface area (Brunauer-Emmett-Teller model) never exceeded 11 m^2/g (3-Et-Et) or 27 m^2/g (3-Et-H) as measured by nitrogen sorption at 77 K.
category: surface area

paragraph:
[a] Department of Chemistry, Zhejiang University, Hangzhou 310027 (P.R.China). E-mail: lihao2015@zju.edu.cn.
category: affiliation

paragraph:
Department of Chemistry, Imperial College London, South Kensington, London, SW7 2AZ, United Kingdom Department of Chemistry and Materials Innovation Factory, University of Liverpool, 51 Oxford Street, Liverpool, L7 3NY, United Kingdom §E.B. and R.L.G. contributed equally.
category: affiliation

paragraph:
Organisch-Chemisches Institut Ruprecht-Karls-Universiteit Heidelberg, Im Neuenheimer Feld

270, 69120 Heidelberg (Germany).
category: affiliation

paragraph:
d. Chemical Refining Laboratory, Refining Department, Egyptian Petroleum Research Institute, Nasr City, 11727, Egypt.
category: affiliation

paragraph:
McMahon, D. P.; Stephenson, A.; Chong, S. Y.; Little, M. A.; Jones, J. T. A.; Cooper, A. I.; Day, G. M. Faraday Discuss. 2018, 211, 383−399.
category: Ref

paragraph:
A. G. Slater, M. A. Little, A. Pulido, S. Y. Chong, D. Holden, L. Chen, C. Morgan, X. Wu, G. Cheng, R. Clowes, M. E. Briggs, T. Hasell, K. E. Jelfs, G. M. Day, A. I. Cooper, Nat. Chem. 2017, 9, 17–25.
category: Ref

paragraph:
Hehre, W. J.; Radom, L.; Schleyer, P. v. R.; Pople, J. A. Ab Initio Molecular Orbital Theory; Wiley: New York, 1986.
category: Ref

paragraph:
A. Blanco-Gómez, I. Neira, J. L. Barriada, M. Melle-Franco, C. Peinador, M. D. García, Chem. Sci. 2019, 10, 10680–10686.
category: Ref

paragraph:
Key learning points (1) The concept of dynamic covalent chemistry (DCvC) (2) Different types of dynamic covalent reactions (3) Characteristic features of a thermodynamically controlled process (4) Applications of DCvC in development of organic 2-D and 3-D molecular architectures and responsive polymers.
category: others

paragraph:
The simulations of the possible structures were carried out in Accelrys Material Studio 7.0 software package. Before the simulations, the structures were firstly optimized in Gaussian 09 package by semi-empirical calculations at PM3 level. The stimulated PXRD patterns were determined by the Reflex module. P1 space group was chosen for the primitive models in the initial simulations. The Pawley refinement of the experimental PXRD was conducted by the Reflux module in the Material Studio 7.0.
category: others

paragraph:
Photoirradiation of a non-agitated solution containing hexamerfibers 200 induced homolytic cleavage of disulfide and radical disulfide exchange reaction, resulting in the lateral cross-linking of fibers and formation of a gel 201 from a free flowing solution. In the presence of excess dithiothreitol, quantitative recovery of monomers was obtained from the gel, indicating that the system is not only photo-responsive but alsoredox-responsive.
category: others

paragraph:
1H NMR ($CDCl_3$, 400 MHz): d=8.35 (s, 12H), 8.25 (s, 12H), 7.61 (m,48H), 7.07 (m, 48H), 3.52 (brm, 24H), 1.77–1.88 (brs, 72H), 1.55 ppm (brs, 24H). 13C NMR ($CDCl_3$, 100 MHz): d = 159.4, 159.3, 148.9, 148.6, 131.8, 131.6, 129.4, 124.1, 123.9, 75.1, 74.5, 33.6, 24.6 ppm. ESI-MS ($MeOH/CHCl_3$): m/z=1785.5 for C240H240N32 $[M+2H]^{2+}$, 1190.7 for C240H240N32 $[M+3H]^{3+}$; MALDI-TOF MS: m/z=3573 for C240H240N32 $[M+H]^{+}$.
category: others

paragraph:
CC7 crystallizes in the cubic space group P213 with two crystallographically independent molecules, each located on a 3-fold axis (Figure S3). CC8 crystallizes isostructurally, with slightly reduced unit cell parameters. The cage molecules CC7 and CC8 are packed only loosely into the three-dimensional structure (Figure 2), giving rise to large internal and external voids which are filled with the solvent dichloro-methane.
category: others

paragraph:
Gas sorption analysis. All samples were tested with gas of the following purities: carbon dioxide (SCF grade, BOC gases) and methane (ultrahigh purity, BOC gases) using a Micromeritics ASAP 2020 volumetric adsorption analyzer. Samples were degassed offline at 100 °C for 15 h under vacuum (10^{-5} bar) before analysis, followed by degassing on the analysis port under vacuum, also at 100 °C. Carbon dioxide isotherms were measured at 273 K.
category: others

paragraph:
Figure S48. CO_2 and C_2H_2 adsorption isotherms of calix[4] resorcinarene at 298 K.
category: others

paragraph:
Both cages have tetrahedral Tsymmetry, and by defining thecentral amine nitrogen atoms in the 8 Aunits (Scheme 1) asnodes, its topology is equivalent to that of the [8+12] chiralnanocube structure proposed by Warmuth. The 8 triphenyl-amine moieties occupy the vertices and the 12 diamine linkers occupy the edges of this highly distorted cube (Figure S4). By contrast, the molecular shape, as opposed to the topology, has more similarities with the structural model proposed for arhombicuboctahedral capsule.
category: others

paragraph:
Nevertheless, this result indicates the feasibility of operating dynamic imine and olefin metathesis simultaneously to achive complex molecular architectures.
category: others

paragraph:
In both polymorphs, the cages are packing mainly via CH-π and dispersion interactions (Figure 1b).In polymorph α, a three-dimensional pore system can be found whereas for polymorph β pores are isolated and thus not accessible. The pores in polymorph α are strongly winded and contain a number of dead-ends.
category: others

paragraph:
The observed [2+3] over the expected [4+6] selectivity remains opposed to the previously postulated “odd–even” rule and is attributed to the presence of three OH groups in each aromatic ring.
category: others

Prompt for data extraction

Assistant Prompt
Follow the instructions, extract the information related to organic molecular cages from the given text and organize them into a table.
System Prompt
Give you some labeled texts containing information about a new research paper related to organic molecular cages. Please extract from it the name of the molecular cage, the specific surface area (give the whole sentence which describe it), the ccdc number, the topology (eg. [2+3], [4+6] etc. , you need to give the whole sentence which describe it), the comprehensive synthesis procedures, the name of the reactant needed for the synthesis, and the the comprehensive synthesis procedures of the reactant, and then organize these information into the following table: |name of the organic molecular cage|topology|ccdc number|specific surface area|cage' s synthesis procedure|reactant1|reactant1' s synthesis procedure|reactant2|reactant2' s synthesis procedure| please note: 1. If some information is not given, use 'None' 2. The comprehensive synthesis procedures must contain amounts of reactant, solvent and catalysts.

Section S3. Python Code

Code for classification of texts

The function of the following code is to convert the pdf file into text format, with the enter key as the split symbol will be divided into a number of parts.

```
import PyPDF2
def count_tokens(text):
    """Returns the number of tokens in a text string."""
    encoding = tiktoken.get_encoding("cl100k_base")
    num_tokens = len(encoding.encode(text))
    return num_tokens

def get_txt_from_pdf(pdf_files,filter_ref = False, combine=False):
    data = []
    pdf_content_dict = {}
    for pdf in pdf_files:
        all_pdf_txt = ''
        print(pdf)
        with open(pdf, 'rb') as pdf_content:
            pdf_reader = PyPDF2.PdfReader(pdf_content)
            for page_num in range(len(pdf_reader.pages)):
                page = pdf_reader.pages[page_num]
                page_text = page.extract_text()
                all_pdf_txt += page_text
            all_pdf_txt = re.split('\n',all_pdf_txt)
            all_pdf_txt = [txt1 for txt in all_pdf_txt for txt1 in re.split('\n \n',txt)]
            for i, para_part in enumerate(all_pdf_txt):
                data.append({
                    'file name': pdf,
                    'paragraph number': i+1,
                    'content': para_part,
                    'tokens': count_tokens(para_part)
                })
    df = pd.DataFrame(data)
    return df
```

The function of the following code is to implement text integration. It first determines whether each section ends with a period or an asterisk. If so, join it with the next part.

```
import pandas as pd
start_index = 0
all_groups = []
data = []
for i in range(len(df)):
    pdf = df['file name'][i]
    content_split = df['content'][i].split()
    if content_split != []:
        content_end = content_split[-1][-1]
        if content_end == '.' or content_end == '*':
            all_groups.append(range(start_index,i+1))
            new_content = ''
            for j in range(start_index,i+1):
                new_content += df['content'][j]
            data.append({
                        'file name': pdf,
                        'content': new_content,
                        'tokens': count_tokens(new_content)
                        })
            start_index = i+1
new_df = pd.DataFrame(data)
```

The following code functions as a packet of several paragraphs.

```
index = 1
part_num = 1
txt = ''
all_paper = {}
for i in range(len(new_df)):
    content = new_df.loc[i,'content']
    filename1 = new_df.loc[i,'file name']
    if i>0:
        filename2 = new_df.loc[i-1,'file name']
        if   filename1!=filename2   or i == len(new_df)-1:
            index = 1
            all_paper[filename2+'part'+str(part_num)] = txt
            txt = ''
            part_num = 1
        if   token_num > 3300-count_tokens(content) or index >= 20:
            index = 1
            all_paper[filename1+'part'+str(part_num)] = txt
            part_num += 1
            txt = ''
    txt += str(index)+'、'+content+'\n \n'
    token_num = count_tokens(txt)
    new_df.loc[i,'new_content'] = str(index)+'、'+content
    new_df.loc[i,'part'] = filename1+'part'+str(part_num)
    index += 1
```

The following code is the core part that sends the packaging of the paragraphs to the GPT-4 model to generate the classification labels.

```
from openai import OpenAI
api_key = "*****"
organization = "*****"
client = OpenAI(api_key=api_key,organization=organization)
```

```
all_answers = {}
data = []
for index, key in enumerate(all_paper_list):
    text = all_paper[key]
    try:
        response = client.chat.completions.create(model='gpt-4',
                                        messages=[ {"role": "assistant", "content": assistant},
                                                   {"role": "system", "content": prompt},
                                                   {"role": "user", "content": text}
                                                   ]
                                    )
        answers = response.choices[0].message.content
        all_answers[key] = answers
        print(index+1,key)
        data.append({
                    'file name': key,
                    'content': text,
                    'categories': answers
                    })
    except Exception as e:
        print(f"Error: {str(e)}")
```

Code for tabulated data organization[3]

The following code uses the GPT-4 API to extract valid information from text.

```
from openai import OpenAI
api_key = "*****"
organization = "*****"
client = OpenAI(api_key=api_key,organization=organization)

assistant = client.beta.assistants.create(
  name="cage table organizer",
  instructions='Follow the instructions, extract the information related to organic molecular cages from the given text and organize them into a table',
  model="gpt-4o"
)
import numpy as np
import pandas as pd
tables = []
csv_files = glob.glob("./CSV_file/*/*.csv")
for input_csv in csv_files:
    thread = client.beta.threads.create()
    key = input_csv.split('\\')[-1].replace('.csv','')
    with open(input_csv,'r',encoding='gb18030') as f:
        input_content = f.read()
    try:
        message=client.beta.threads.messages.create(thread_id=thread.id,
  role='user',content=description+input_content)
        manual_answer = manual_dict[key]
        run = client.beta.threads.runs.create(
        thread_id=thread.id,
        assistant_id=assistant.id
        )
        sleep(30)
        print(run.id)
        run_retrive = client.beta.threads.runs.retrieve(
        thread_id=thread.id,
        run_id=run.id)
        print(run_retrive.status)
        while(run_retrive.status == 'in_progress'):
            sleep(10)
            run_retrive = client.beta.threads.runs.retrieve(thread_id=thread.id,run_id=run.id)
        print(run_retrive.status)
        messages = client.beta.threads.messages.list(thread_id=thread.id)
        output_message = messages.data[0].content[0].text.value
        i = output_message.find("|")
        j = output_message.rfind("|")
        gpt_answer = output_message[i:j+1].replace('--','').replace('|-|','')
        tables.append([input_csv,manual_answer,gpt_answer])
    except Exception as e:
        print(str(e))
df_gpt_manual = pd.DataFrame(tables)
df_gpt_manual.to_csv('./CSV_file/gpt_manual_log.csv')
```

The following code is used to calculate the similarity between the answer given by GPT and the reference answer.[2]

```
import numpy as np
import pandas as pd
import evaluate
all_answers = pd.read_csv('./###.csv',encoding='gb18030')
bertscore = evaluate.load("bertscore")

def get_monomers(text):
    text_split = text.split('\n')
    matrix = []
    for each_paragraph in text_split:
        each_paragraph = each_paragraph.split('|')
        each_paragraph = list(filter(None, each_paragraph))
        length = len(each_paragraph)
        if length == 9:
            monomer1 = each_paragraph[-4]
            monomer2 = each_paragraph[-2]
            synthesis1 = each_paragraph[-3]
            synthesis2 = each_paragraph[-1]

            matrix.append([monomer1+';'+monomer2,synthesis1+';'+synthesis2])
        else:
            print('error',length,each_paragraph)
    matrix = list(map(list, zip(*matrix)))
    return matrix

def get_others(text):
    text_split = text.split('\n')
    matrix = []
    for each_paragraph in text_split:
        each_paragraph = each_paragraph.split('|')
        each_paragraph = list(filter(None, each_paragraph))
        length = len(each_paragraph)
        if length == 9:
            matrix.append(each_paragraph)
        else:
            print('error',length,each_paragraph)
    matrix = list(map(list, zip(*matrix)))
    return matrix

for i in range(len(all_answers)):
    standard = all_answers['standard'][i]
    GPT = all_answers['GPT'][i]
    predictions = [GPT]
    references = [standard]
    results=bertscore.compute(predictions=predictions,references=references,
lang="en",model_type="distilbert-base-uncased")
    recall = results['recall']
    all_answers.loc[i, "Similarity ofavg"] = recall[0]
    standard_monomers = get_monomers(standard)
    GPT_monomers = get_monomers(GPT)
    standard_monomers = ['|'.join(each_part) for each_part in standard_monomers]
    GPT_monomers = ['|'.join(each_part) for each_part in GPT_monomers]
    results=bertscore.compute(predictions=GPT_monomers, references=standard_monomers,
lang="en", model_type = "distilbert-base-uncased")
    monomers_recall = str(results['recall'])
    standard_matrix = get_others(standard)
    GPT_matrix = get_others(GPT)
    standard_matrix = ['|'.join(each_part) for each_part in standard_matrix]
    GPT_matrix = ['|'.join(each_part) for each_part in GPT_matrix]
    results = bertscore.compute(predictions=GPT_matrix, references=standard_matrix, lang="en", model_type = "distilbert-base-uncased")
    others_recall = str(results['recall'])
    all_answers.loc[i, "Similarity of monomers"] = monomers_recall
    all_answers.loc[i, "Similarity of others"] = others_recall
```

Code for building chatbots

The following code builds a chat system using tkinter, an interface through which users can send messages and receive replies generated by GPT-4o.

```
import tkinter as tk
from tkinter import simpledialog
from time import sleep
from tkinter import ttk
import threading
from PIL import Image, ImageTk
from openai import OpenAI
api_key = "***"
organization = “***”
client = OpenAI(api_key=api_key,organization=organization)
file = client.files.create(
  file=open("original-database.json", "rb"),
  purpose='assistants'
)
VectorStore = client.beta.vector_stores.create(file_ids=[file.id],name='cage-database')
assistant = client.beta.assistants.create(
  instructions="""A table summarizing many organic molecular cages is given to you. \
  From left to right are the title of the article, the author, the organization, \
  the name of the molecular cage, the topology, the CCDC number, the specific surface area, \
  the synthesis conditions, the name of the monomer, and the conditions under which the monomer was synthesized. \
  Please answer the user's questions based on the table.""",
  model="gpt-4o",
  tools=[{"type": 'file_search'}],tool_resources={"file_search": {"vector_store_ids": [VectorStore.id]}})

def write_log(txt):
    with open('./chatbot_log.txt','a',encoding='gb18030') as f:
        f.write(txt)

def get_gpt_response(prompt):
    try:
        message=client.beta.threads.messages.create(thread_id=thread.id, role='user',content=prompt)
        run = client.beta.threads.runs.create(
                thread_id=thread.id,
                assistant_id=assistant.id
                )
        run_retrive = client.beta.threads.runs.retrieve(
                thread_id=thread.id,
                run_id=run.id)
        while(run_retrive.status == 'in_progress'):
            sleep(10)
            run_retrive=client.beta.threads.runs.retrieve(thread_id=thread.id,run_id=run.id)
        messages = client.beta.threads.messages.list(thread_id=thread.id)
        output_message = messages.data[0].content[0].text.value
        return output_message
    except Exception as e:
        return str(e)
def handle_message():
    message = entry.get()
    if message:
        display_message("User", message, 'right', 'yellow')
        entry.delete(0, tk.END)
        display_message("GPT", "Please wait...", 'left', 'light pink')
        threading.Thread(target=update_with_gpt_response, args=(message,)).start()
def update_with_gpt_response(message):
    response = get_gpt_response(message)
    chat_box_frame.winfo_children()[-1].destroy()
    display_message("GPT", response, 'left', 'light pink')
def display_message(sender, message, side, bg_color):
    message_frame = tk.Frame(chat_box_frame, bg=bg_color, padx=10, pady=5, bd=1, relief='solid')
    message_label = tk.Label(message_frame, text=f"{sender}: {message}", bg=bg_color, font=("Times New Roman", 10), wraplength=500, justify='left')
    message_label.pack()
    if side == 'left':
        message_frame.pack(anchor='w', pady=10, padx=10)
    else:
        message_frame.pack(anchor='e', pady=10, padx=10)
    chat_box_canvas.yview_moveto(1)
root = tk.Tk()
root.title("GPT Dialog")
root.geometry("400x600")
chat_box_canvas = tk.Canvas(root, bg='light green')
```

```
chat_box_frame = tk.Frame(chat_box_canvas, bg='light green')
chat_box_scrollbar = ttk.Scrollbar(root, orient='vertical', command=chat_box_canvas.yview)
chat_box_canvas.configure(yscrollcommand=chat_box_scrollbar.set)
chat_box_scrollbar.pack(side='right', fill='y')
chat_box_canvas.pack(side='left', fill='both', expand=True)
chat_box_canvas.create_window((0, 0), window=chat_box_frame, anchor='nw')
def on_frame_configure(event):
    chat_box_canvas.configure(scrollregion=chat_box_canvas.bbox("all"))
chat_box_frame.bind("<Configure>", on_frame_configure)
entry_frame = tk.Frame(root, bg='lightgray', bd=2)
entry = tk.Entry(entry_frame, width=30, font=("Times New Roman", 12))
entry.pack(side='left', padx=5, pady=5)
send_button=tk.Button(entry_frame,text="Send",command=handle_message, font=("Times New Roman", 12, 'bold'), bg='#4CAF50', fg='white')
send_button.pack(side='left', padx=5, pady=5)
entry_frame.pack(side='bottom', fill='x')
root.mainloop()
```

Section S4. Detailed experimental data

Detailed data for text classification

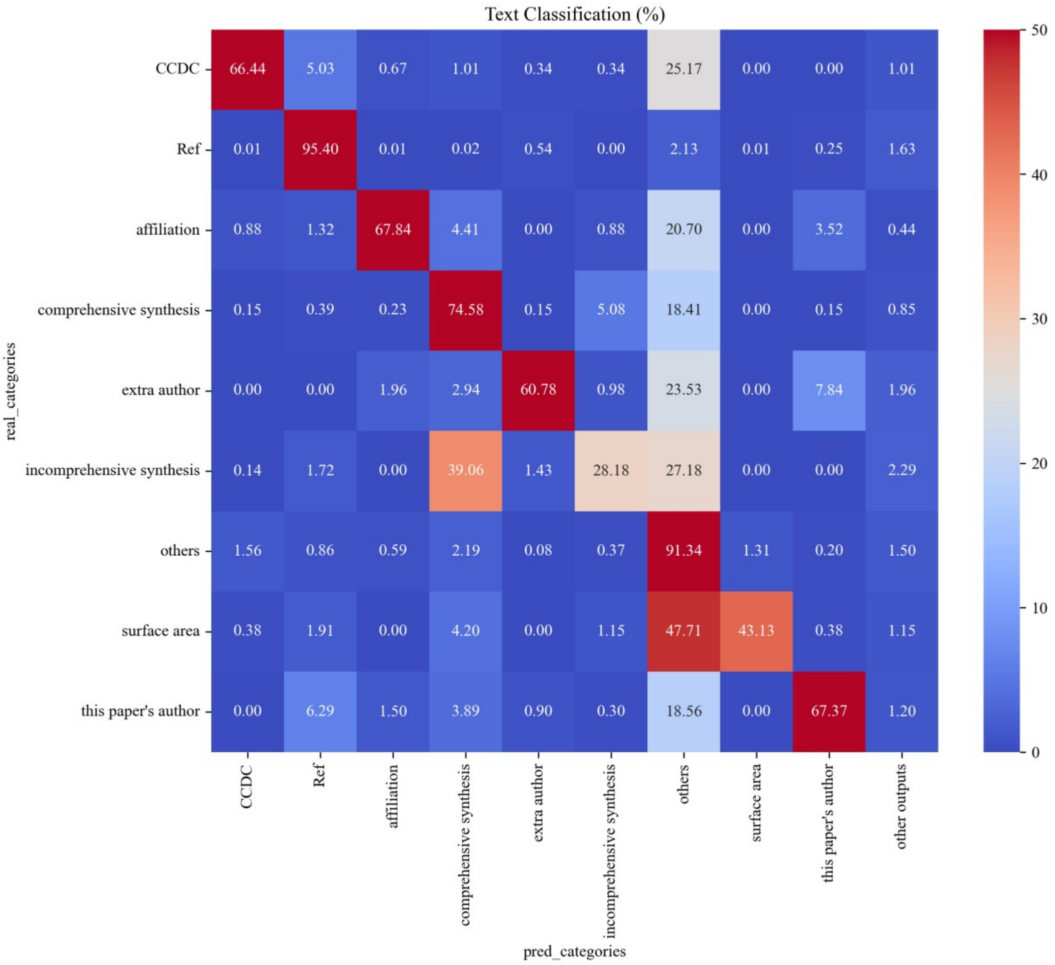


Fig S1. Details pertaining to text classification utilizing GPT-4 are described herein. For instance, the datum located in the first column of the first row indicated that the proportion of texts belonging to category "CCDC" and accurately classified is nearly 0.66. Conversely, the value situated in the fourth column of the sixth row signifies that the proportion of texts attributed to the category "incomprehensive synthesis" and erroneously classified as "comprehensive synthesis" is nearly 0.40.

Similarity of information extraction

Table S1. The similarity of information extracted by GPT-4 compared to manually extracted information from text was evaluated, encompassing both category-specific similarity metrics and overall similarity.

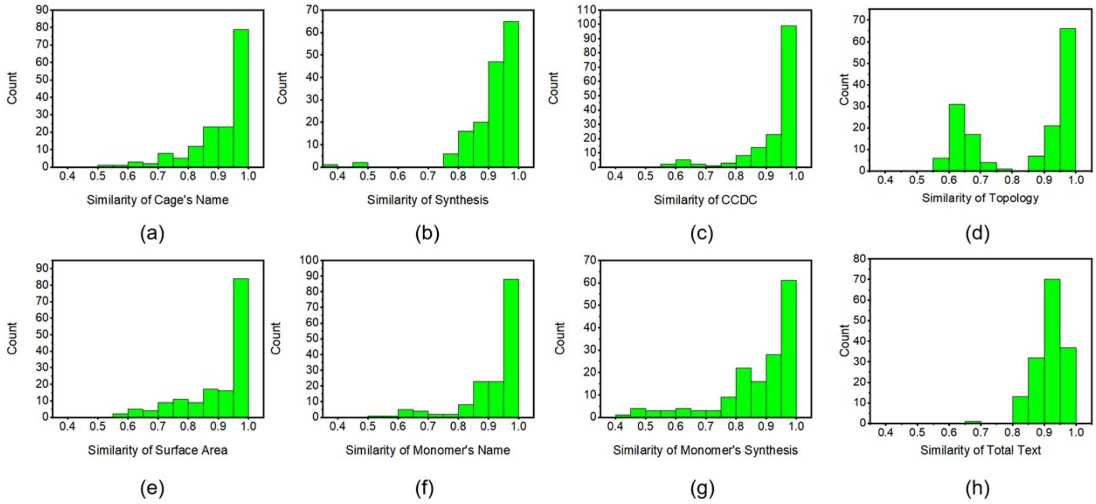
Index	Overall	Name	Synthesis	BET	CCDC	Topology	Monomer's Name	Monomer's Synthesis
1	0.9128	0.8819	0.9395	1.0000	1.0000	0.5970	0.8378	0.8221
2	0.9340	1.0000	0.4672	1.0000	0.9628	0.6064	0.9767	0.6663
3	0.9107	0.7542	0.9396	1.0000	0.8211	0.6064	0.9459	0.9419
4	0.9456	1.0000	0.9954	0.9419	1.0000	1.0000	1.0000	0.9233
5	0.9439	1.0000	0.9719	0.8855	1.0000	1.0000	1.0000	0.8148
6	0.9855	0.9735	0.9902	1.0000	1.0000	0.6030	0.9777	0.9965
7	0.9836	0.7278	0.9949	1.0000	1.0000	1.0000	1.0000	1.0000

8	0.9143	0.7334	0.8005	1.0000	1.0000	0.6064	0.9162	0.8367
9	0.8709	1.0000	0.8874	0.7127	1.0000	1.0000	1.0000	1.0000
10	0.9729	1.0000	0.9792	1.0000	1.0000	0.6064	1.0000	0.9782
11	0.9727	1.0000	0.9979	1.0000	1.0000	0.6064	1.0000	0.9321
12	0.9334	1.0000	0.9686	0.8249	0.9609	0.6030	0.9840	1.0000
13	0.9397	0.9933	0.9684	0.5931	0.6072	1.0000	0.7724	0.9580
14	0.9478	0.9605	0.9485	1.0000	1.0000	0.6064	1.0000	1.0000
15	0.9816	0.9575	0.9968	1.0000	0.9689	0.9677	1.0000	0.9966
16	0.9138	1.0000	0.9491	0.7432	1.0000	0.9632	1.0000	1.0000
17	0.8859	0.8606	0.8402	0.8979	0.8115	1.0000	0.9205	0.9313
18	0.9064	1.0000	0.8616	0.7581	1.0000	0.6030	0.9806	0.8333
19	0.9627	1.0000	0.9730	1.0000	0.6293	1.0000	0.9617	0.7157
20	0.9826	1.0000	0.9948	1.0000	1.0000	0.6064	1.0000	0.9868
21	0.9617	1.0000	0.9966	1.0000	0.9630	0.5980	1.0000	0.9518
22	0.8640	1.0000	0.8040	0.8848	1.0000	1.0000	0.6099	0.7140
23	0.9226	1.0000	0.9419	0.8536	0.9258	0.9652	1.0000	1.0000
24	0.9232	1.0000	0.9543	0.8692	0.9588	1.0000	1.0000	1.0000
25	0.9856	1.0000	0.9905	1.0000	0.8794	1.0000	1.0000	1.0000
26	0.9782	0.9404	0.9751	1.0000	0.8614	0.6064	0.9379	0.9757
27	0.9600	0.8846	0.9699	1.0000	0.8841	0.6064	0.8652	0.9636
28	0.9083	0.7611	0.9213	1.0000	0.8761	0.6060	0.8644	1.0000
29	0.9844	1.0000	0.9936	0.9340	1.0000	1.0000	1.0000	1.0000
30	0.8361	0.9311	0.8178	0.9327	0.9669	0.5970	0.8181	0.8463
31	0.9434	0.7801	0.9351	1.0000	0.8587	0.6095	0.9772	0.9757
32	0.8986	0.8854	0.9018	1.0000	1.0000	0.6064	0.8513	1.0000
33	0.9080	0.7152	0.9247	1.0000	0.7828	0.6064	0.8282	0.9450
34	0.9542	0.8865	1.0000	1.0000	0.8967	0.6064	0.7482	0.4960
35	0.9799	0.9079	0.9893	0.9965	1.0000	1.0000	1.0000	0.8419
36	0.9345	0.5960	0.8892	1.0000	1.0000	0.6096	0.9584	0.6281
37	0.9863	1.0000	0.9805	1.0000	0.8764	1.0000	1.0000	0.9900
38	0.9804	0.8495	0.9905	1.0000	1.0000	0.5970	0.9957	1.0000
39	0.9459	1.0000	0.9845	0.9072	0.6166	0.6096	0.9728	1.0000
40	0.8815	1.0000	0.8617	1.0000	1.0000	1.0000	0.9683	0.8650
41	0.9118	1.0000	0.9562	0.8511	1.0000	1.0000	1.0000	0.8331
42	0.9355	0.9600	0.8846	1.0000	0.9664	0.6030	0.9778	0.8418
43	0.9509	0.8338	0.9592	1.0000	0.5983	0.6064	0.8232	0.9693
44	0.9011	0.9245	0.9517	0.7970	0.9229	1.0000	0.9940	1.0000
45	0.9639	1.0000	0.9516	0.8875	1.0000	1.0000	0.8733	0.9524
46	0.9435	0.7241	0.9895	1.0000	0.8576	1.0000	0.9924	1.0000
47	0.9757	1.0000	0.9290	0.9010	0.9811	0.9306	0.8652	0.9466
48	0.9324	0.9085	0.8840	0.7728	0.9637	1.0000	1.0000	0.9715
49	0.9050	1.0000	0.8490	1.0000	1.0000	0.9764	0.9326	0.5201
50	0.8818	0.8574	0.9783	1.0000	0.6054	0.6064	1.0000	0.9446
51	0.9588	0.7109	0.8857	1.0000	1.0000	0.9446	0.6151	1.0000

52	0.8868	1.0000	0.9740	1.0000	1.0000	0.9400	0.6528	0.9606
53	0.9412	1.0000	0.9848	0.8666	1.0000	0.9306	1.0000	1.0000
54	0.9638	0.9069	0.9862	1.0000	1.0000	1.0000	0.5525	0.9038
55	0.9143	0.9059	0.9615	1.0000	0.9715	1.0000	0.8914	0.9250
56	0.8922	0.8189	0.9154	1.0000	0.5952	0.6064	1.0000	0.7955
57	0.8485	1.0000	0.9579	1.0000	0.9139	0.6064	0.9781	0.9295
58	0.9445	1.0000	0.9893	1.0000	1.0000	0.9779	0.9962	0.9789
59	0.9837	1.0000	0.8874	0.6236	1.0000	0.9306	0.9774	1.0000
60	0.8919	1.0000	0.8199	1.0000	1.0000	1.0000	0.9590	0.9740
61	0.9514	0.8109	0.8153	0.9520	0.9520	1.0000	0.8560	0.7566
62	0.8046	1.0000	0.9556	0.8729	1.0000	0.9286	0.9806	0.9761
63	0.9137	1.0000	0.8742	0.7926	0.8850	1.0000	0.9934	0.8419
64	0.8585	0.6810	0.9612	1.0000	1.0000	0.6758	0.9718	0.8679
65	0.9436	0.9191	0.9247	0.7014	0.8397	0.6769	0.9681	0.7925
66	0.8877	0.9700	0.9788	0.8912	0.9928	0.9817	0.9670	0.9396
67	0.9641	0.8684	0.9272	0.8403	0.9338	0.6702	0.9453	0.9274
68	0.9348	0.6426	0.3875	0.6848	0.6848	0.9282	0.7363	0.4262
69	0.6599	0.9894	0.9120	0.9976	0.9764	0.7604	0.9887	0.8641
70	0.9018	0.8787	0.9660	0.7897	1.0000	0.9264	0.8876	1.0000
71	0.9409	0.7993	0.9757	0.8198	1.0000	0.6883	0.9756	0.9773
72	0.9221	0.8628	0.9356	0.9160	0.9160	1.0000	0.8108	0.8252
73	0.9338	1.0000	0.9016	0.7213	1.0000	0.5799	0.9724	0.9333
74	0.9233	1.0000	0.9813	1.0000	1.0000	1.0000	0.9871	0.9631
75	0.9710	0.8044	0.9953	0.7674	1.0000	0.6894	0.9889	0.7733
76	0.8364	0.9491	0.9463	0.9856	0.9856	0.6729	0.9631	0.9525
77	0.9509	0.9827	0.9705	1.0000	0.9827	0.9587	0.9750	0.9303
78	0.9484	0.8107	0.9897	0.6851	0.9818	0.9791	1.0000	1.0000
79	0.9313	0.9240	0.9156	0.9753	0.9753	1.0000	0.9762	0.8898
80	0.9422	0.7323	0.9715	0.9941	0.9941	0.6482	0.9874	0.8919
81	0.9573	1.0000	0.9785	0.7002	0.6324	0.9944	0.9812	0.8497
82	0.9247	0.8512	0.8472	0.7555	0.8570	0.9824	0.9251	0.8412
83	0.8515	0.9907	0.9403	0.6350	1.0000	0.9492	0.9594	1.0000
84	0.8180	1.0000	0.9768	0.9727	0.9252	0.9778	1.0000	0.9729
85	0.9634	0.8789	0.9659	0.7079	0.9494	0.9603	0.6301	0.9580
86	0.9320	0.8777	0.8202	0.8619	0.9441	1.0000	0.8946	0.8736
87	0.8492	0.9804	0.7993	0.9485	0.9724	0.9314	0.9455	0.9976
88	0.8327	0.5240	0.9392	0.6341	0.8276	0.9724	0.9219	0.9371
89	0.9131	0.9577	0.7904	1.0000	0.8959	0.9205	0.6592	0.8056
90	0.8081	0.8087	0.8294	0.9518	0.9351	0.9492	0.9300	0.7975
91	0.8687	0.9007	0.9613	0.9987	0.9603	0.7339	0.5052	0.9988
92	0.9602	0.9935	0.9569	1.0000	0.9954	0.6645	0.9588	0.6812
93	0.9146	1.0000	1.0000	1.0000	1.0000	1.0000	0.9435	0.7999
94	0.9842	1.0000	0.8932	0.5918	1.0000	0.6689	1.0000	0.8154
95	0.8604	1.0000	0.9378	0.8030	1.0000	1.0000	0.9890	1.0000

96	0.9439	0.9320	0.9265	0.9944	0.9631	1.0000	0.8701	0.8587
97	0.9010	0.9981	0.9229	0.7120	0.9786	0.9680	0.9895	0.9980
98	0.9140	1.0000	0.9981	1.0000	1.0000	0.9091	1.0000	0.9871
99	0.9892	0.9051	0.9455	1.0000	0.9301	0.9791	0.9189	0.9163
100	0.9647	1.0000	0.9283	0.8095	1.0000	0.9862	1.0000	1.0000
101	0.8941	0.9530	0.9799	0.9022	1.0000	1.0000	1.0000	0.8318
102	0.8628	0.9634	0.9134	1.0000	1.0000	0.9496	0.9633	0.9520
103	0.9540	0.9931	0.9695	0.6575	0.9729	0.8892	0.8448	0.8697
104	0.8520	0.7836	0.9408	0.9778	0.6715	0.9870	0.9379	0.9281
105	0.9502	0.9868	0.9102	0.9941	0.9700	0.6247	0.9906	0.9968
106	0.9020	0.8845	0.9624	1.0000	0.9707	0.8924	1.0000	0.8627
107	0.9201	0.8669	0.9287	0.9428	0.9428	0.9547	0.9080	0.9553
108	0.9332	0.8925	0.9247	1.0000	1.0000	0.9653	0.9721	0.9908
109	0.9039	0.8404	0.8393	0.9753	0.9775	0.6467	0.8678	0.6454
110	0.8830	1.0000	0.9735	0.7436	0.9399	1.0000	0.9654	1.0000
111	0.8976	0.9961	0.8955	0.8458	0.9613	0.6813	0.9901	1.0000
112	0.8884	1.0000	0.9742	0.7375	1.0000	1.0000	1.0000	0.8160
113	0.8296	0.9691	0.8079	0.9979	0.9979	0.6524	0.9820	0.7885
114	0.9071	0.8627	0.8783	1.0000	0.9514	1.0000	0.9932	1.0000
115	0.8759	1.0000	0.4574	0.8973	0.9637	0.6894	0.9775	0.9034
116	0.9085	0.9733	0.9155	0.9878	0.9729	0.7124	0.9749	0.9326
117	0.9028	0.8285	0.9651	0.7908	0.9473	0.9906	1.0000	0.9608
118	0.9054	0.6817	0.9023	0.6267	0.7820	0.9464	0.8943	0.9012
119	0.8711	0.9935	0.9631	0.9976	0.9765	0.7053	0.9599	0.9049
120	0.9068	0.9817	0.9004	0.9327	0.9196	0.9096	0.9719	0.4880
121	0.9303	0.8949	0.9071	0.9928	0.9444	0.9683	0.9733	0.9205
122	0.8513	0.9265	0.8922	0.9833	0.8776	0.9773	0.9436	0.6039
123	0.9172	1.0000	0.9388	0.8074	1.0000	1.0000	0.9821	0.9378
124	0.9800	0.8147	0.9830	1.0000	1.0000	0.6867	0.9154	0.9683
125	0.9372	1.0000	0.9730	0.8771	0.9492	0.9131	0.9545	0.9226
126	0.8211	1.0000	0.8452	0.6865	1.0000	1.0000	1.0000	0.9229
127	0.8569	1.0000	0.8277	1.0000	1.0000	0.9919	1.0000	1.0000
128	0.8671	0.9700	0.8725	0.6475	0.9753	0.9558	0.9800	0.8125
129	0.9574	0.9795	0.9649	1.0000	0.9720	0.6902	0.9825	0.9587
130	0.8437	0.8182	0.7820	0.9028	0.7853	0.8854	0.8880	0.8730
131	0.8461	0.8876	0.7905	0.9367	0.9367	0.6056	0.9052	0.7839
132	0.9745	0.9248	0.9419	0.9753	0.9753	0.9812	0.9797	0.5527
133	0.8559	0.9838	0.8620	0.9988	0.9988	0.8782	0.8775	0.5064
134	0.8918	0.8913	0.8987	0.9723	0.9723	0.6262	0.8912	0.8288
135	0.9453	0.9452	0.9480	1.0000	1.0000	0.6758	0.8552	1.0000
136	0.8844	0.8811	0.8973	0.9674	0.9674	0.6171	0.6463	0.4914
137	0.9159	0.9908	0.9702	0.7711	0.9941	0.9292	0.8671	0.7865
138	0.8876	0.9572	0.7699	0.9520	0.9520	0.9559	0.8527	0.8597
139	0.9071	0.9272	0.8958	0.9946	1.0000	0.9909	0.8917	0.5546

140	0.8846	0.9091	0.9063	0.8771	0.8325	0.6702	0.8832	0.8395
141	0.9306	0.8503	0.9399	0.7949	0.9408	0.9957	0.9341	0.5710
142	0.9445	0.6193	0.9170	0.8914	0.8813	0.8526	0.6253	0.8709
143	0.9409	0.8427	0.9309	0.9692	0.8897	0.9361	0.8480	0.8868
144	0.9031	0.9771	0.9055	0.7804	1.0000	0.6758	0.9508	0.4992
145	0.8805	0.6482	0.8279	0.9753	0.7362	0.6467	0.7741	0.8531
146	0.9483	1.0000	0.9812	1.0000	0.9696	0.9791	0.9604	0.6440
147	0.9011	0.8707	0.9034	0.8276	0.8276	0.5548	0.6841	0.9231
148	0.8565	0.7436	0.8244	0.8600	0.8276	0.8541	0.8916	0.8244
149	0.9231	0.7148	0.9309	0.9022	0.9601	0.7156	0.9252	0.6943
150	0.9398	1.0000	0.9774	1.0000	1.0000	0.9345	0.6968	0.8489
151	0.9237	0.9027	0.8893	0.8276	0.8276	0.9219	0.8539	0.9173
152	0.8353	0.9453	0.7957	0.9856	0.9398	0.8957	0.8401	0.8302
153	0.9082	1.0000	0.9565	1.0000	0.9604	0.6642	0.9434	0.5190



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig S2. The similarity distribution of each item, including cage's name(a), synthesis(b), CCDC number(c), topology(d), surface area(e), name(f) and synthesis(g) of the monomer molecule, and the overall similarity distribution(h).

Example of questions and completed answers on Chatbot

Input:
What is the reducing agent for the conversion of CC3 to RCC3?
Output:
The reducing agent used for the conversion of CC3 to RCC3 is sodium borohydride ($NaBH_4$).
Input:
Please tell me briefly the exact synthesis route.
Output:
1. Dissolve the imine cage CC3-R (926 mg, 0.83 mmol) in a chloroform/methanol mixture (1:1 v/v, 50 mL) and stir until the solution becomes clear. 2. Add sodium borohydride (1.00 g, 26.5 mmol) to the clear solution and stir for 12 hours at

room temperature.
3. Add water (2 mL) to the reaction mixture and stir for an additional 12 hours.
4. Remove the solvent under vacuum.
5. Extract the resulting white solid with chloroform (2 × 50 mL).
6. Combine the organic phase and wash it with water (2 × 100 mL).
7. Dry the chloroform phase using anhydrous magnesium sulfate (MgSO4) before removing it under vacuum.
8. RCC3 (crude yield = 900 mg, 95.1%) is obtained as a white solid.
9. Purify RCC3 using a reversible reaction with acetone:
    - Dissolve 1000 mg of crude RCC3 in 100 mL of acetone and allow the solution to stand.
    - Crystals start to appear on the wall of the flask after 30 minutes.
    - Collect the crystals (AT-RCC3) after one day by filtration.
    - Dissolve the crystals again in a chloroform/methanol mixture (1:1 v/v) and stir.
    - Add several drops of distilled water to the solution and stir for another 12 hours.
    - Remove the solvents to recover pure RCC3 (680 mg, 70.4%).

This process effectively converts CC3 to RCC3 using sodium borohydride as the reducing agent.

Input:

Please list all molecular cages reported by Prof. Andy Cooper (name and article title only)

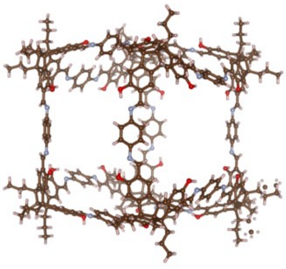
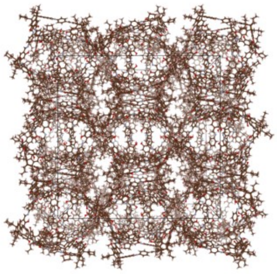
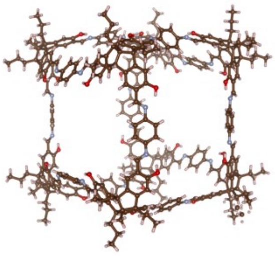
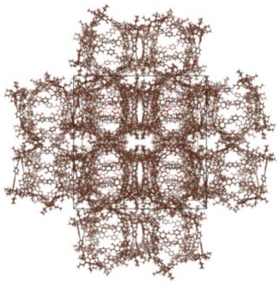
Output:

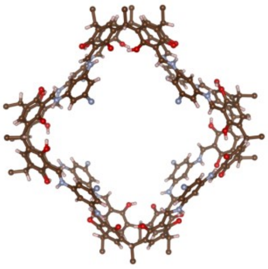
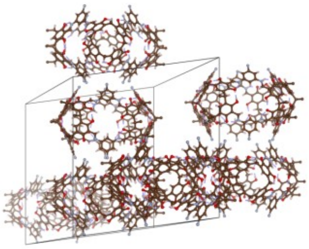
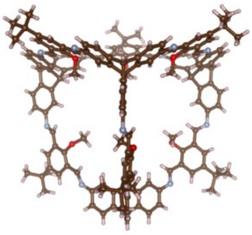
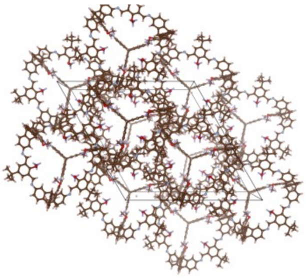
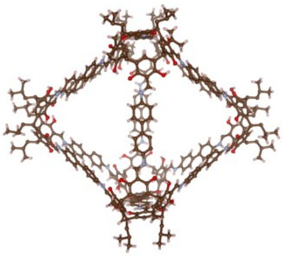
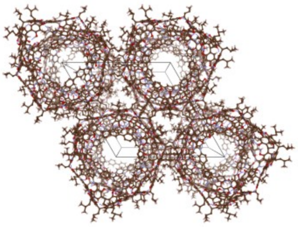
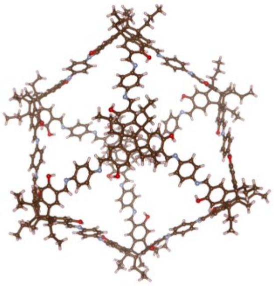
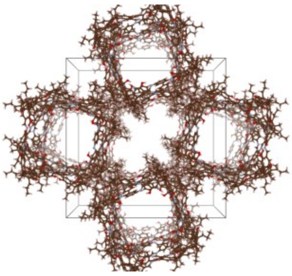
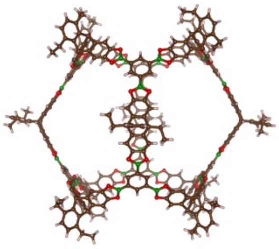
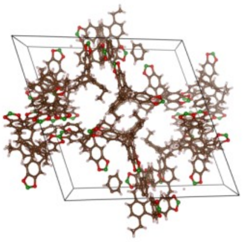
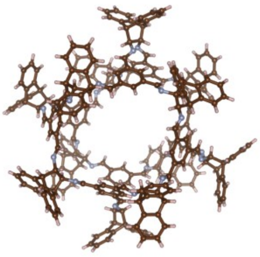
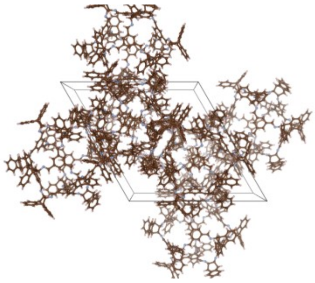
Here are the molecular cages reported by Prof. Andy Cooper along with the respective article titles:
1. **Cage Name: CC7**
    - **Article Title:** Large Self-Assembled Chiral Organic Cages: Synthesis, Structure, and Shape Persistence
2. **Cage Name: CC8**
    - **Article Title:** Large Self-Assembled Chiral Organic Cages: Synthesis, Structure, and Shape Persistence
3. **Cage Name: TCC1 [3+6]**
    - **Article Title:** Cage Doubling: Solvent-Mediated Re-equilibration of a [3 + 6] Prismatic Organic Cage to a Large [6+12] Truncated Tetrahedron
4. **Cage Name: CC20**
    - **Article Title:** Synthesis of a Large, Shape-Flexible, Solvatomorphic Porous Organic Cage
5. **Cage Name: CC13**
    - **Article Title:** Controlling the Crystallization of Porous Organic Cages: Molecular Analogs of Isoreticular Frameworks Using Shape-Specific Directing Solvents
6. **Cage Name: CC10**
    - **Article Title:** Supramolecular engineering of intrinsic and extrinsic porosity in covalent organic cages
7. **Cage Name: CC9**
    - **Article Title:** Supramolecular engineering of intrinsic and extrinsic porosity in covalent organic cages
8. **Cage Name: Cage 3-R**
    - **Article Title:** Modular and Predictable Assembly of Porous Organic Molecular Crystals
9. **Cage Name: CC12**

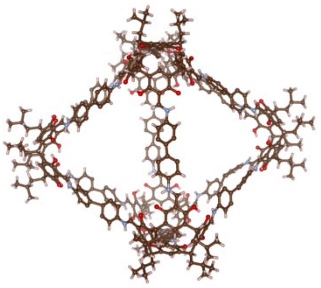
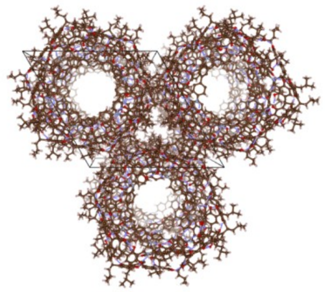
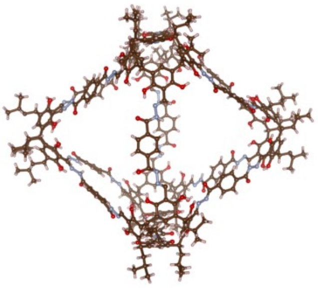
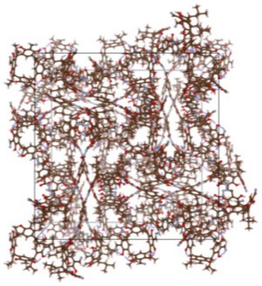
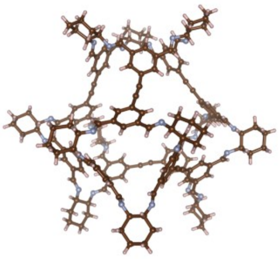
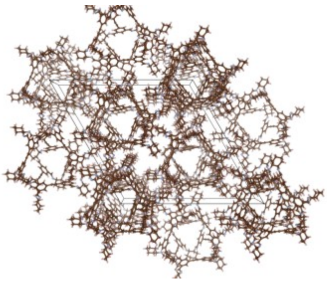
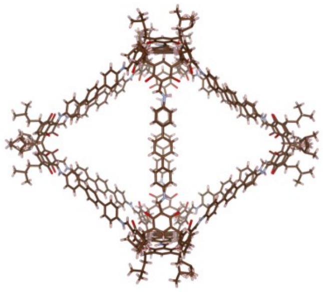
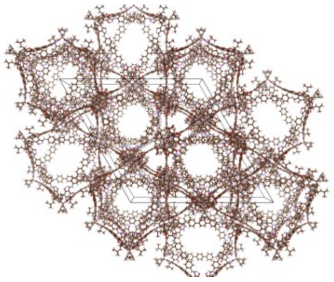
   - **Article Title:** Shape Prediction for Supramolecular Organic Nanostructures: [4 + 4] Macrocyclic Tetrapods
10. **Cage Name: Cage 5-R**
   - **Article Title:** Supramolecular engineering of intrinsic and extrinsic porosity in covalent organic cages
11. **Cage Name: CC11**
   - **Article Title:** Shape Prediction for Supramolecular Organic Nanostructures: [4 + 4] Macrocyclic Tetrapods
12. **Cage Name: CC6**
   - **Article Title:** Selective Gas Sorption in a [2+3] "Propeller" Cage Crystal
13. **Cage Name: CC18**
   - **Article Title:** Periphery-Functionalized Porous Organic Cages
14. **Cage Name: Cage 3**
   - **Article Title:** Porous Organic Cages
15. **Cage Name: TCC1-R**
   - **Article Title:** Reticular Synthesis of Porous Molecular 1-Dimensional Nanotubes and 3-Dimensional Networks
16. **Cage Name: TCC3-R**
   - **Article Title:** Reticular Synthesis of Porous Molecular 1-Dimensional Nanotubes and 3-Dimensional Networks

Representative crystal structure with high calculated surface area and density

Table S2. Representative crystal structure, molecular structure, density, surface area, CCDC number.

CCDC Number	Density(g/cm^3)	ASA(m^2/g)	Molecular Structure	Crystal Structure
2022634	0.390228	5104.11		
2022633	0.386877	5102.87		

2113665	0.436129	4531.72		
904717	0.555444	4515.21		
2014846	0.433226	4048.25		
2221442	0.464375	3773.4		
948612	0.457181	3707.89		
1481846	0.558249	3651.02		

2113144	0.471088	3543.63		
2097232	0.477613	3532.98		
1578448	0.558173	3510.71		
2014847	0.484538	3499.93		

Section S5. References

1 OpenAI, **https://openai.com/**, accessed December 2024.

2 J. Shin, Y. Lee and K. Jung, *Pr. Mach. Learn. Res.*, 2019, **101**, 1081-1093.

3 Z. L. Zheng, O. F. Zhang, C. Borgs, J. T. Chayes and O. M. Yaghi, *J. Am. Chem. Soc.*, 2023, **145**, 18048-18062.