Electronic Supplementary Material (ESI) for Digital Discovery.
This journal is © The Royal Society of Chemistry 2024

Supporting Information

Image and Data Mining in Reticular Chemistry Powered by GPT-4V

Zhiling Zheng,[abc] Zhiguo He,[ab] Omar Khattab,[d] Nakul Rampal,[abc] Matei A. Zaharia,[e]
Christian Borgs,[ce] Jennifer T. Chayes,[cefgh] and Omar M. Yaghi*[abci]

[a] Department of Chemistry, University of California, Berkeley, California 94720, United States

[b] Kavli Energy Nanoscience Institute, University of California, Berkeley, California 94720, United States

[c] Bakar Institute of Digital Materials for the Planet, University of California, Berkeley, California 94720, United States

[d] Department of Computer Science, Stanford University, Stanford, California 94305, United States

[e] Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, California 94720, United States

[f] Department of Mathematics, University of California, Berkeley, California 94720, United States

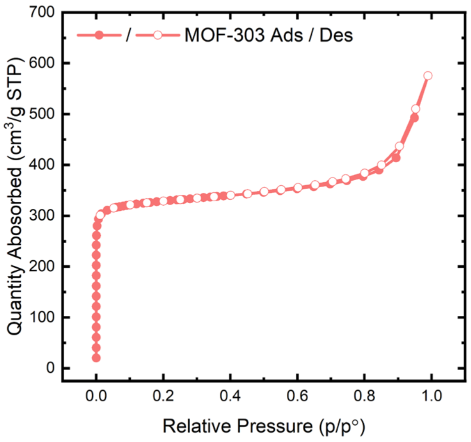
[g] Department of Statistics, University of California, Berkeley, California 94720, United States

[h] School of Information, University of California, Berkeley, California 94720, United States

[i] KACST–UC Berkeley Center of Excellence for Nanomaterials for Clean Energy Applications, King Abdulaziz City for Science and Technology, Riyadh 11442, Saudi Arabia

* To whom correspondence should be addressed: yaghi@berkeley.edu

Prompt: Describe the image.



GPT-4V:

The image is a graph representing the adsorption and desorption isotherms of a material named MOF-303. On the x-axis, we have "Relative Pressure (p/p°)" which is dimensionless, and it ranges from 0 to 1. The y-axis represents "Quantity Absorbed (cm^3/g STP)" which is a volume per mass at standard temperature and pressure (STP), and it ranges from 0 to 700 cm^3/g.

The adsorption part of the isotherm (as pressure increases) appears to have a steep rise at low pressures indicating a rapid uptake of the gas. After a certain point, the graph plateaus, suggesting that the material has reached its maximum adsorption capacity.

The markers are open circles, connected by a line, which is common in such types of graphs to represent experimental data points. The color of the data points and line is a shade of pink or light red. The background is white, and the axes are labeled in black with a standard font.

Figure S1. Interaction with GPT-4V using a nitrogen sorption isotherm image. The provided image was combined with a brief text prompt to evaluate GPT-4V's proficiency in interpreting scientific figures from literature. The data depicted in the test image are a replot from a previous publication[1] to avoid copyright concerns. In its response, GPT-4V demonstrated the ability to discern not only the content but also to interpret the colors, characters, and numbers within the graph. Note that the full prompt used in this study (Figure S15) is different from the one shown in this example.

Prompt: Describe the image.

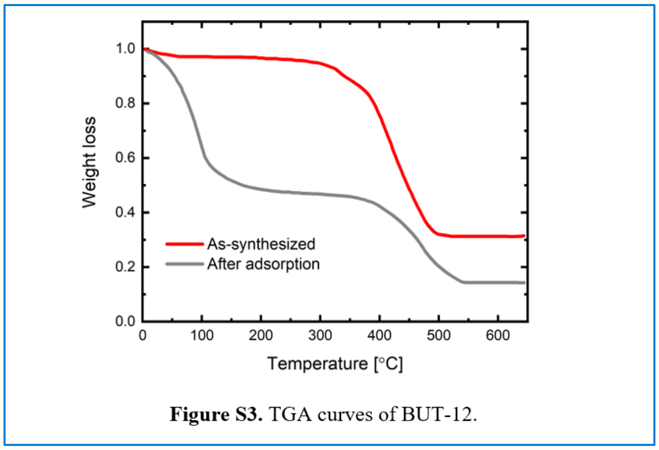


Figure S3. TGA curves of BUT-12.

GPT-4V:
The image is a graph, specifically a Thermogravimetric Analysis (TGA) curve. TGA is a type of testing that is used to determine changes in weight in relation to change in temperature. The graph shows two curves:

1. The red curve represents the "As-synthesized" sample, which indicates the weight loss of the material as it was originally produced.
2. The grey curve represents the sample "After adsorption", showing the weight loss after the material has presumably adsorbed some substance.

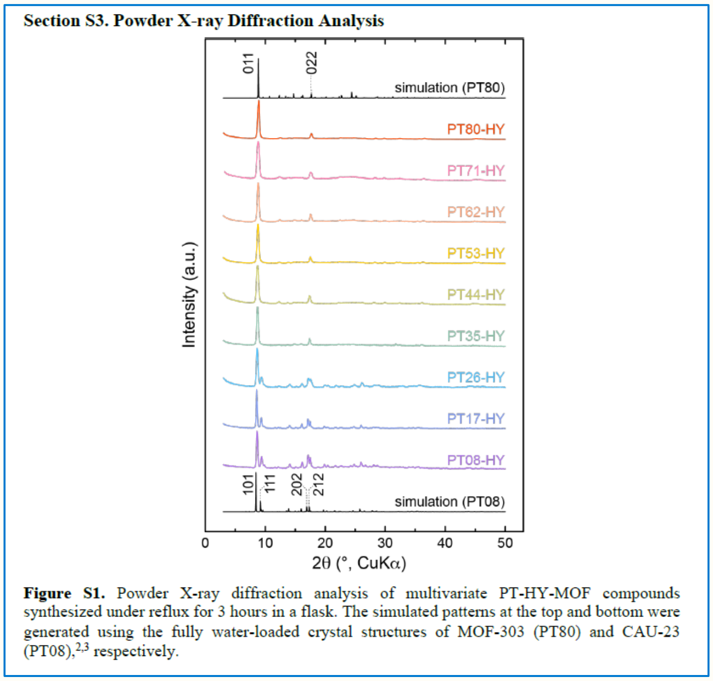
The x-axis of the graph indicates the temperature in degrees Celsius (°C), ranging from 0 to 600°C. The y-axis represents the weight loss, scaled from 0 to 1, which is likely a normalized or relative scale rather than absolute weight.

The red curve starts at a weight loss value close to 1 and remains stable until approximately 200°C, after which it begins to decline, indicating that the material is losing weight as the temperature increases, likely due to the loss of volatile components or decomposition. The grey curve starts also close to 1 but begins to decline slightly earlier than the red curve, and continues to do so at a gradual rate, suggesting that the adsorbed material affects the weight loss pattern.

At the bottom, there's a caption "Figure S3. TGA curves of BUT-12," which identifies this as the third supplemental figure in a series, related to a material or compound referred to as BUT-12. This kind of graph is commonly found in scientific publications, particularly in materials science and chemistry to report the thermal stability of materials.

Figure S2. Interaction with GPT-4V using a plot containing TGA curves. The provided image was combined with a brief text prompt to evaluate GPT-4V's proficiency in interpreting scientific figures from literature. Note that the data depicted are a replot from a previous publication[2] to avoid copyright concerns. In its response, GPT-4V demonstrated the ability to discern not only the content but also to interpret the colors, characters, and numbers within the graph.

Prompt: Describe the image.



Section S3. Powder X-ray Diffraction Analysis

Figure S1. Powder X-ray diffraction analysis of multivariate PT-HY-MOF compounds synthesized under reflux for 3 hours in a flask. The simulated patterns at the top and bottom were generated using the fully water-loaded crystal structures of MOF-303 (PT80) and CAU-23 (PT08),[2,3] respectively.

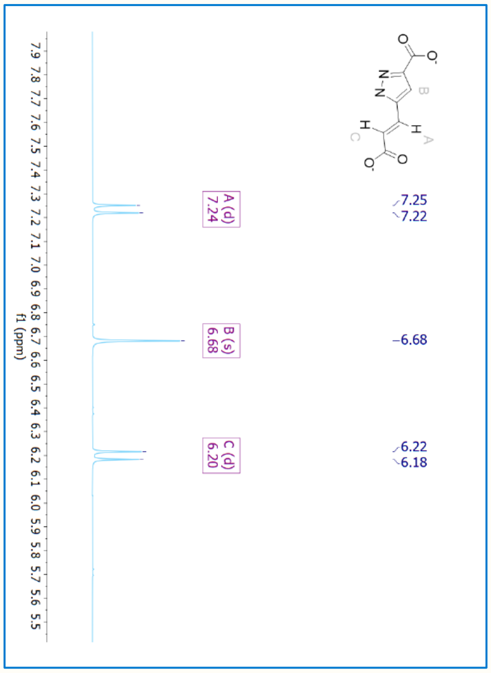
GPT-4V:
The image is of a scientific figure from a research paper or report, specifically Section S3 titled "Powder X-ray Diffraction Analysis". It shows a series of X-ray diffraction (XRD) patterns for different samples, which are labeled PT80-HY, PT71-HY, PT62-HY, PT53-HY, PT44-HY, PT35-HY, PT26-HY, PT17-HY, and PT08-HY. Each pattern displays the intensity of diffracted X-rays as a function of the angle (2θ, with CuKα radiation), which is a common method for characterizing the crystalline structures of materials.

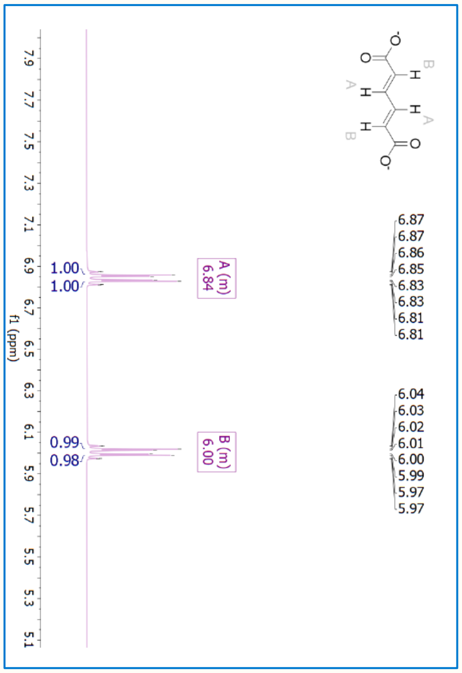
At the top and bottom of the graph, there are simulated XRD patterns labeled as "simulation (PT80)" and "simulation (PT08)", which serve as references for comparing the experimental data. These simulated patterns are for the fully water-loaded crystal structures of two different materials referred to as MOF-303 (PT80) and CAU-23 (PT08), respectively. The intensity is plotted on the y-axis (in arbitrary units, a.u.), and the angle 2θ is on the x-axis, ranging from 0 to 50 degrees. The peaks in each pattern correspond to the crystalline planes within the materials, and their positions and intensities are unique to the specific structure of the material being analyzed.

The text below the figure describes the conditions under which the PT-HY-MOF compounds were synthesized (under reflux for 3 hours in a flask) and mentions the simulated patterns that correspond to the water-loaded crystal structures of MOF-303 (PT80) and CAU-23 (PT08). The footnote numbers "2,3" after "PT08" refer to citations within the main text of the document.

Figure S3. Interaction with GPT-4V using a PXRD pattern sourced from a previous publication[3]. Copyright © 2022 American Chemical Society.

Prompt: Describe the images.





GPT-4V:

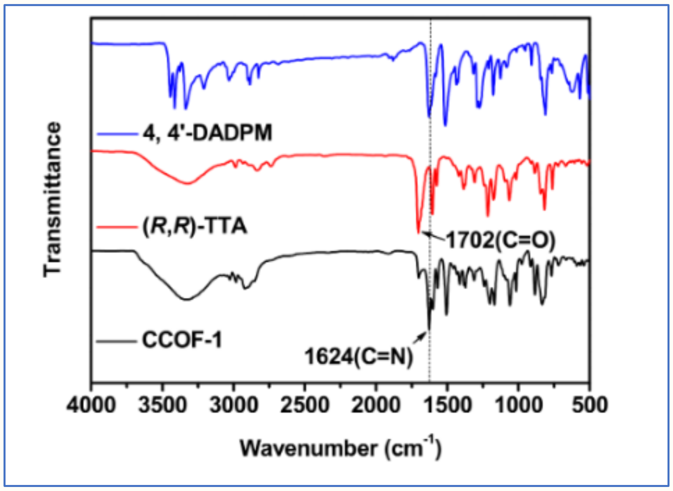
The images you've provided appear to be screenshots of nuclear magnetic resonance (NMR) spectra, which are used in chemistry to determine the structure of molecules. Each peak or set of peaks in the spectrum corresponds to particular groups of atoms within the molecule, and their position (chemical shift) on the horizontal axis (typically measured in parts per million, or ppm) provides information about the electronic environment of these atoms.

The accompanying chemical structures suggest that these spectra are being used to analyze these specific molecules, with annotations indicating the peaks corresponding to particular atoms or groups within the molecules. The letters (A, B, C, etc.) next to the peaks likely correlate to the labeled atoms or groups in the molecular structure shown in the image.

Such data is often used by chemists for compound verification, structure elucidation, and in studies of molecular dynamics. Without specific details on the molecules or the context of the spectra, this is a general interpretation of the type of information that these images represent.

Figure S4. Interaction with GPT-4V using two NMR spectra images rotated 90 degrees. GPT-4V demonstrated the ability to process multiple images simultaneously and identify the plots even when rotated.

Prompt: Describe the image.



GPT-4V:
The image shows an infrared (IR) spectroscopy graph with three overlaid spectra, each represented by a different color. From top to bottom, there is a blue, a red, and a black spectrum.

- The blue spectrum is labeled as "4, 4'-DADPM".
- The red spectrum is labeled as "(R,R)-TTA".
- The black spectrum is labeled as "CCOF-1".

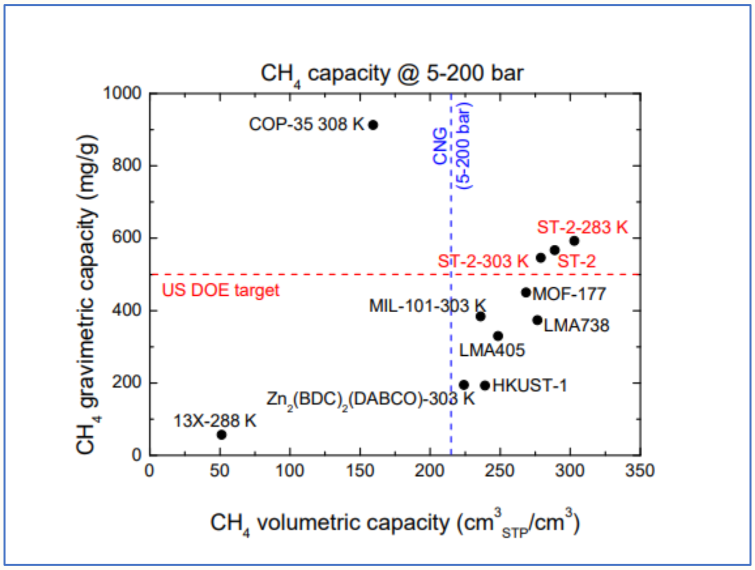
There are two peaks in the spectra with annotations:
- One peak in the red spectrum is labeled with "1702 (C=O)", indicating the transmittance peak corresponding to a carbonyl functional group.
- A peak in the black spectrum is labeled with "1624 (C=N)", indicating a transmittance peak for a carbon-nitrogen double bond.

The x-axis is labeled as "Wavenumber (cm^-1)", which is the standard unit of wavenumber for IR spectra, ranging from approximately 4000 to 500 cm^-1. The y-axis is labeled as "Transmittance", which measures how much light is transmitted at each wavenumber. The transmittance is plotted as a percentage, where a higher transmittance indicates less absorption by the sample at that particular wavenumber.

This type of graph is typically used in analytical chemistry to identify functional groups in molecules and to characterize chemical compounds. Each of the labeled peaks corresponds to a different molecular vibration that is characteristic of a certain chemical bond or functional group within the molecule.

Figure S5. Interaction with GPT-4V using the IR spectra sourced from a previous publication[4]. Copyright © 2016 American Chemical Society.

Prompt: Describe the image.



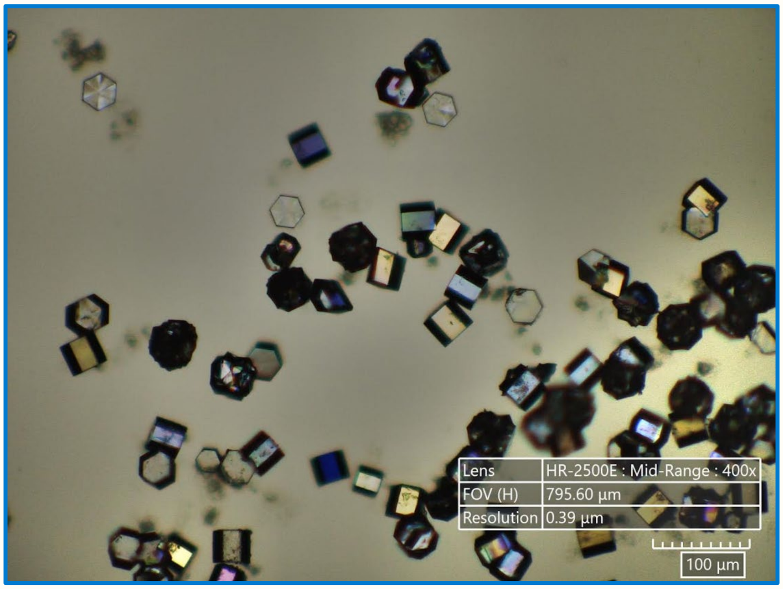
GPT-4V:
The image is a graph plotting methane (CH4) storage capacities of various materials under a pressure range of 5-200 bars. The x-axis represents the volumetric capacity (measured in cubic centimeters of standard temperature and pressure gas per cubic centimeter of material, cm^3_STP/cm^3), while the y-axis shows the gravimetric capacity (measured in milligrams of gas per gram of material, mg/g).

Different materials are plotted as points on the graph, such as COP-35, Zn2(BDC)2(DABCO), MIL-101, MOF-177, HKUST-1, and others, each represented by a distinct marker. The U.S. Department of Energy (DOE) target for methane storage is indicated by a dashed horizontal red line, while a dashed vertical blue line may indicate a benchmark or target for volumetric capacity.

The graph is used to compare the performance of these materials against the DOE targets for methane storage, which is a key metric for applications like natural gas vehicles where storage efficiency can determine the range and practicality of the technology.

Figure S6. Interaction with GPT-4V using a scatter plot sourced from a previous publication[5]. GPT-4V demonstrated the capability to identify key elements within the plot and to interpret them accurately. Copyright © 2017 American Chemical Society.

Prompt: Describe the image.

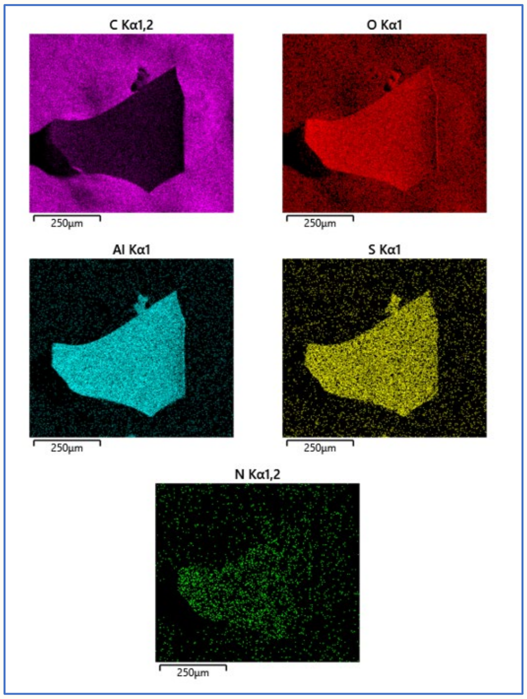


GPT-4V:
This image appears to be taken through a microscope, showing a variety of crystal-like structures, possibly a salt or some mineral crystalline forms, scattered across the field of view. The crystals have different shapes, with some exhibiting more complex geometries, and many of them display dark edges, likely due to their refractive properties bending light. The colors vary from clear to dark shades, possibly due to the thickness or orientation of the crystals.

On the bottom of the image, there's a scale bar indicating 100 micrometers (µm), which provides a reference for the size of the crystals. The information overlaid on the image indicates that this is taken with a high-resolution lens (HR-2500E) at a mid-range magnification of 400x. The field of view (FOV) is noted as 795.60 micrometers horizontally, with a resolution of 0.39 micrometers, suggesting that the image is highly detailed.

Figure S7. Interaction with GPT-4V using an image taken with a microscope. The image was accompanied by a textual prompt to assess GPT-4V's ability to interpret real experimental result images, including labels and scale bars. Notably, the image is from unpublished work, ensuring that GPT-4V's interpretation is based on an image it has not previously encountered, thus testing its ability to recognize new crystal images.

Prompt: Describe the image.

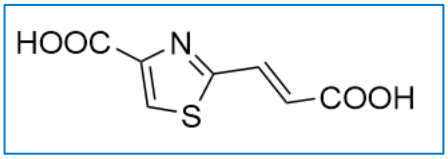


GPT-4V:
The image appears to be a composite of five separate images, each one showing a different Energy-dispersive X-ray spectroscopy (EDX or EDS) map of a sample. These maps are used to illustrate the distribution of elements within the sample. Each of the images is labeled with the element it represents and the characteristic X-ray line used in the analysis (Kα line). The elements displayed are carbon (C), oxygen (O), aluminum (Al), sulfur (S), and nitrogen (N).

Each map is color-coded for a specific element, and the intensity of the color represents the concentration of the element in the sample. The colors used are purple for carbon, red for oxygen, cyan for aluminum, yellow for sulfur, and green for nitrogen. Each image also has a scale indicator in the lower left corner, stating "250μm," which implies that the field of view is 250 micrometers wide.

Figure S8. Interaction with GPT-4V using a series of scanning electron microscopy (SEM) with energy dispersive X-ray spectroscopy (EDS) images. GPT-4V was able to identify and interpret the provided image. Note that the images are from unpublished work, ensuring that GPT-4V's interpretation is based on an image it has not previously encountered, thus testing its ability to recognize SEM images.

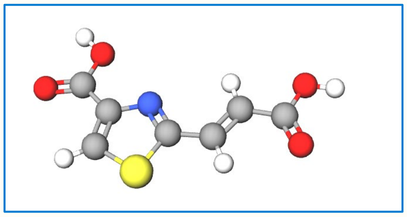
Prompt: Describe the image.



GPT-4V:
The image you've uploaded appears to be a structural chemical formula. It represents an organic molecule with a specific arrangement of atoms and bonds. The molecule contains a ring structure that includes nitrogen (N) and sulfur (S) atoms, indicating it might be a heterocyclic compound. There are two carboxylic acid groups (-COOH) attached to the ring, suggesting the compound is a dicarboxylic acid. The precise angles and lengths of the lines in the diagram depict the bonding structure and the molecular geometry of the compound. This type of diagram is commonly used in chemistry to represent the structure of molecules.

Figure S9. Interaction with GPT-4V using a chemical structure of an organic linker that has never been published in any paper or dataset before.

Prompt: Describe the image.

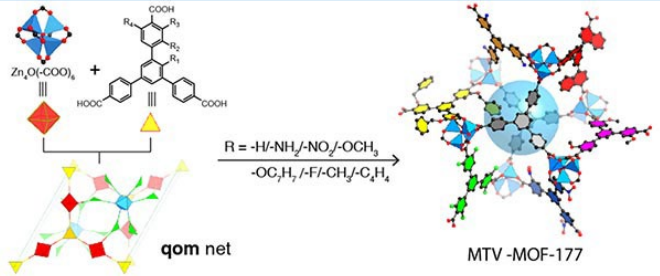


GPT-4V:
The image appears to be a three-dimensional molecular structure model, which is commonly used in chemistry to represent the arrangement of atoms in a molecule. This particular model shows a molecule with various colored spheres representing different atoms. The grey spheres typically represent carbon atoms, the white spheres are hydrogen, the red spheres are oxygen, and the blue sphere is likely nitrogen. The yellow sphere could represent a sulfur atom. These spheres are connected by sticks which represent chemical bonds; single bonds are typically depicted by a single stick, and double bonds by two sticks parallel to each other.

This kind of representation is known as a ball-and-stick model and is used to provide a visual interpretation of a molecule's structure, showing how the atoms are connected and their relative positions in three-dimensional space.

Figure S10. Interaction with GPT-4V using an image visualized by MolView, which shows 3D version of the same molecule with different chemical elements in default color settings.

Prompt: Describe the image.



GPT-4V: The image is a graphical representation of chemical processes and structures. On the left side, there is a reaction scheme showing a metal ion, Zn(II), reacting with a ligand that is a complex organic molecule with a central benzene ring substituted with various functional groups and carboxylic acid groups. The ligand is depicted with R groups, indicating variable substituents, which include -H, $-NH_2$, $-NO_2$, $-OCH_3$, $-OC_6H_4$, -F, and $-CH_3-C_6H_4$. Below this reaction scheme, there is a colorful illustration of a "qom net," which seems to represent a type of molecular framework or structure resulting from this reaction.

On the right side, there is a more complex molecular structure labeled "MTV-MOF-177," which is likely a type of Metal-Organic Framework (MOF). This part of the image shows a 3D ball-and-stick model with various colors representing different atoms or groups within the structure. The central part of the structure is highlighted with a semi-transparent blue sphere, indicating a cavity or pore within the MOF, which is typical for these materials known for their porosity.

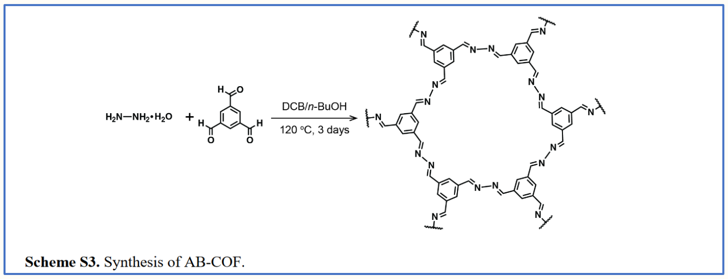
The image is likely used in a scientific context, possibly in materials science or chemistry, to illustrate the synthesis of a Metal-Organic Framework and the resulting porous structure that can have applications in gas storage, separation, or catalysis.

Prompt: What does MTV mean here?

GPT-4V: In the context of Metal-Organic Frameworks (MOFs), MTV stands for "multivariate," which refers to the incorporation of multiple organic functionalities on the ligands within a single framework. In MTV-MOFs, different functionalities occupy the same kind of positions in the framework, and the percentage of each functionality can be varied.

Figure S11. Interaction with GPT-4V using a TOC image from a publication abstract, which details the schematic synthesis of a MOF described in the publication[6]. GPT-4V's response accurately identifies key components: the metal ion, the organic linker, and the resulting MOF structure and topology. Additionally, the dialogue demonstrates GPT-4V's capability to delve deeper into specific aspects, such as explaining the abbreviation 'MTV' in the context of MOFs. Copyright © 2015 American Chemical Society.

Prompt: Describe the image.



Scheme S3. Synthesis of AB-COF.

GPT-4V: The image displays a chemical synthesis scheme labeled "Scheme S3. Synthesis of AB-COF." It illustrates a chemical reaction where two molecules on the left side are reacting to form a more complex molecule on the right side. On the left, there are two reactants:

1. A molecule of hydrazine hydrate ($H_2N—NH_2*H_2O$).
2. A molecule that appears to be a derivative of terephthalaldehyde.

These reactants are subjected to a reaction condition using a catalyst or reagent (DCB/n-BuOH) at a temperature of 120°C for 3 days. The product on the right is a large, ring-shaped organic molecule, which likely represents a covalent organic framework (COF), given the label "AB-COF." This molecule is composed of repeating units that are connected by nitrogen atoms, forming a hexagonal mesh-like structure. The repeating units include aromatic rings with nitrogen and carbon atoms, indicating a possible conjugated system which could contribute to the properties of the AB-COF.

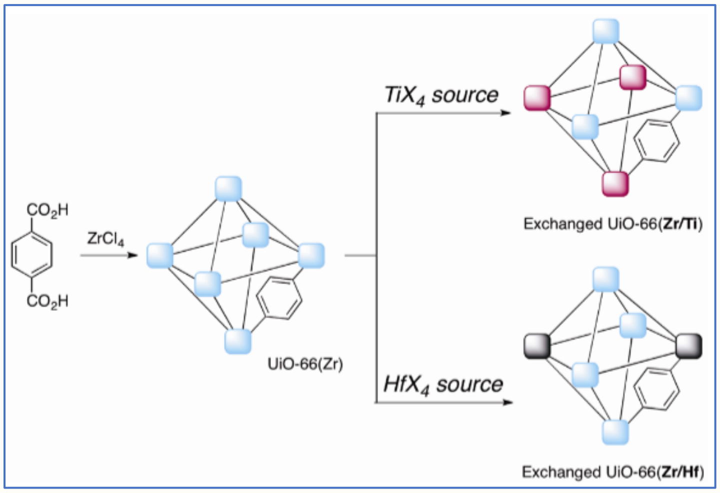
The specific arrangement of the atoms and the bonds suggests that the COF has been synthesized through a condensation reaction, where smaller molecules join together, releasing smaller by-products such as water, and creating a larger, more complex structure. The presence of nitrogen in the hexagonal rings indicates that the reaction involves the formation of azine or imine bonds.

Prompt: Tell me more about the reaction condition.

GPT-4V: The reaction condition "DCB/n-BuOH at 120°C, 3 days" indicates that the synthesis of the AB-COF is performed in a mixed solvent system consisting of dichlorobenzene (DCB) and n-butanol (n-BuOH) at a high temperature of 120 degrees Celsius, maintained for three days. DCB is a non-polar solvent, while n-BuOH is a moderately polar solvent; their combination may be used to optimize the solubility of the reactants and control the reaction rate. The high temperature and extended time suggest a thermally driven reaction that may require such conditions for the reactants to fully convert into the covalent organic framework (COF).

Figure S12. Interaction with GPT-4V using a synthesis scheme from a previous publication[7]. GPT-4V's response identifies the starting materials, reaction conditions and products. In addition, it is able to further explain chemicals' abbreviations. Copyright © 2019 American Chemical Society.

Prompt: Describe the image.



GPT-4V:
The image is a schematic representation of a chemical process, specifically the synthesis and post-synthetic modification of a metal-organic framework (MOF) known as UiO-66.

On the left, there is a chemical structure with two carboxylic acid groups (CO_2H) attached to a benzene ring, which is the organic linker used to synthesize the MOF. This linker reacts with $ZrCl_4$ (zirconium tetrachloride), represented by blue spheres, to form the UiO-66(Zr) framework. The blue spheres likely represent Zr^{4+} ions, and the lines between them and the organic linker represent coordination bonds.

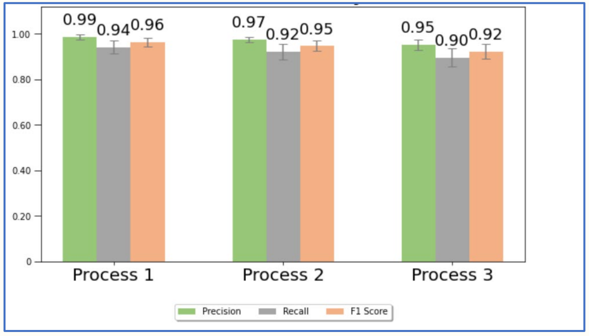
On the right side, there are two pathways shown for post-synthetic modification of UiO-66(Zr). The upper pathway shows the MOF reacting with TiX_4 (where X represents halogen atoms), resulting in an "Exchanged UiO-66(Zr/Ti)" where some Zr^{4+} ions in the framework have been replaced by Ti^{4+} ions, indicated by pink spheres.

The lower pathway shows the MOF reacting with HfX_4 (where X again represents halogen atoms), resulting in "Exchanged UiO-66(Zr/Hf)" where some Zr^{4+} ions have been replaced by Hf^{4+} ions.

The image is used to illustrate the synthetic pathway of UiO-66 and how its metal sites can be modified post-synthesis with other metal sources like Ti and Hf. The representations are stylized rather than realistic, with colors and shapes indicating different components of the MOF structure.

Figure S13. Interaction with GPT-4V using a post synthetic cation exchange scheme from previous publication[8]. GPT-4V's response clearly explains the scheme, starting from the synthesis to create the pristine MOF and detailing the metal exchange process. Copyright © 2012 American Chemical Society.

Prompt: Describe the image.



GPT-4V: The image is a bar chart with three groups of bars, each group representing a different process: Process 1, Process 2, and Process 3. For each process, there are three bars colored differently to represent Precision, Recall, and F1 Score, respectively.

- For Process 1, Precision is approximately 0.99, Recall is about 0.94, and the F1 Score is around 0.96.
- In Process 2, Precision drops slightly to around 0.97, Recall is roughly 0.92, and the F1 Score is about 0.95.
- For Process 3, the Precision is around 0.95, Recall is approximately 0.90, and the F1 Score is close to 0.92.

Each bar has a black line at the top, representing the error bar, which indicates the variability or uncertainty in the measurements of each metric. The scale on the y-axis goes from 0 to 1 in increments of 0.2, and each bar reaches up to a value between 0.9 and 1.0, indicating high performance across all three metrics for all processes.

Prompt: Write python codes to plot a similar image.

GPT-4V:

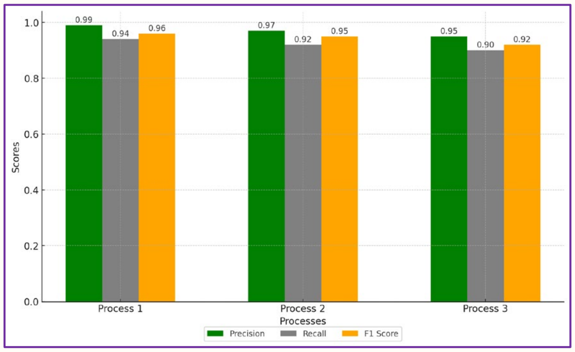
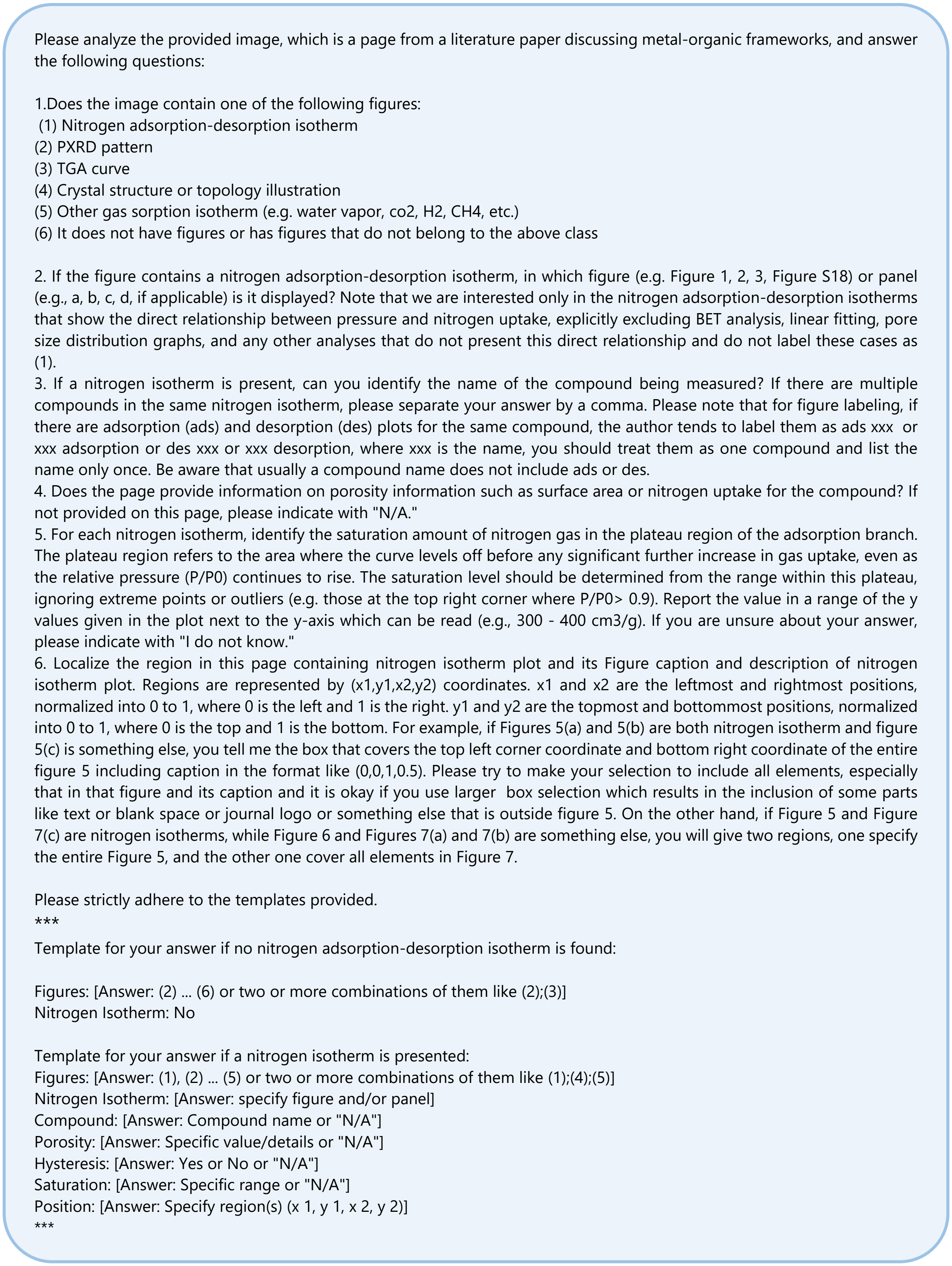


Figure S14. Interaction with GPT-4V using a performance metrics plot sourced from a previous publication[9]. Further requests can be made to replot the image based on the information seen by GPT-4V in the provided image.



Please analyze the provided image, which is a page from a literature paper discussing metal-organic frameworks, and answer the following questions:

1.Does the image contain one of the following figures:
 (1) Nitrogen adsorption-desorption isotherm
(2) PXRD pattern
(3) TGA curve
(4) Crystal structure or topology illustration
(5) Other gas sorption isotherm (e.g. water vapor, co2, H2, CH4, etc.)
(6) It does not have figures or has figures that do not belong to the above class

2. If the figure contains a nitrogen adsorption-desorption isotherm, in which figure (e.g. Figure 1, 2, 3, Figure S18) or panel (e.g., a, b, c, d, if applicable) is it displayed? Note that we are interested only in the nitrogen adsorption-desorption isotherms that show the direct relationship between pressure and nitrogen uptake, explicitly excluding BET analysis, linear fitting, pore size distribution graphs, and any other analyses that do not present this direct relationship and do not label these cases as (1).
3. If a nitrogen isotherm is present, can you identify the name of the compound being measured? If there are multiple compounds in the same nitrogen isotherm, please separate your answer by a comma. Please note that for figure labeling, if there are adsorption (ads) and desorption (des) plots for the same compound, the author tends to label them as ads xxx or xxx adsorption or des xxx or xxx desorption, where xxx is the name, you should treat them as one compound and list the name only once. Be aware that usually a compound name does not include ads or des.
4. Does the page provide information on porosity information such as surface area or nitrogen uptake for the compound? If not provided on this page, please indicate with "N/A."
5. For each nitrogen isotherm, identify the saturation amount of nitrogen gas in the plateau region of the adsorption branch. The plateau region refers to the area where the curve levels off before any significant further increase in gas uptake, even as the relative pressure (P/P0) continues to rise. The saturation level should be determined from the range within this plateau, ignoring extreme points or outliers (e.g. those at the top right corner where P/P0> 0.9). Report the value in a range of the y values given in the plot next to the y-axis which can be read (e.g., 300 - 400 cm3/g). If you are unsure about your answer, please indicate with "I do not know."
6. Localize the region in this page containing nitrogen isotherm plot and its Figure caption and description of nitrogen isotherm plot. Regions are represented by (x1,y1,x2,y2) coordinates. x1 and x2 are the leftmost and rightmost positions, normalized into 0 to 1, where 0 is the left and 1 is the right. y1 and y2 are the topmost and bottommost positions, normalized into 0 to 1, where 0 is the top and 1 is the bottom. For example, if Figures 5(a) and 5(b) are both nitrogen isotherm and figure 5(c) is something else, you tell me the box that covers the top left corner coordinate and bottom right coordinate of the entire figure 5 including caption in the format like (0,0,1,0.5). Please try to make your selection to include all elements, especially that in that figure and its caption and it is okay if you use larger box selection which results in the inclusion of some parts like text or blank space or journal logo or something else that is outside figure 5. On the other hand, if Figure 5 and Figure 7(c) are nitrogen isotherms, while Figure 6 and Figures 7(a) and 7(b) are something else, you will give two regions, one specify the entire Figure 5, and the other one cover all elements in Figure 7.

Please strictly adhere to the templates provided.
***
Template for your answer if no nitrogen adsorption-desorption isotherm is found:

Figures: [Answer: (2) ... (6) or two or more combinations of them like (2);(3)]
Nitrogen Isotherm: No

Template for your answer if a nitrogen isotherm is presented:
Figures: [Answer: (1), (2) ... (5) or two or more combinations of them like (1);(4);(5)]
Nitrogen Isotherm: [Answer: specify figure and/or panel]
Compound: [Answer: Compound name or "N/A"]
Porosity: [Answer: Specific value/details or "N/A"]
Hysteresis: [Answer: Yes or No or "N/A"]
Saturation: [Answer: Specific range or "N/A"]
Position: [Answer: Specify region(s) (x 1, y 1, x 2, y 2)]
***

Figure S15. Full prompt guiding GPT-4 for literature figure analysis in this study.

```
import fitz, os, csv

def pdf_to_images(pdf_path, output_folder, dpi=300):
    # Open PDF and set up image conversion settings
    doc, scale_factor = fitz.open(pdf_path), dpi / 72.0
    if not os.path.exists(output_folder): os.makedirs(output_folder)

    # Convert each page to image and collect data
    image_data = [(f"{os.path.basename(pdf_path).replace('.pdf', '')}_page_{page_num+1}.png",
page_num+1)
                  for page_num, page in enumerate(doc) if
(page.get_pixmap(matrix=fitz.Matrix(scale_factor,
scale_factor)).save(os.path.join(output_folder, f"{os.path.basename(pdf_path).replace('.pdf',
'')}_page_{page_num+1}.png")))]
    return image_data

base_folder, folders_to_process = base, ["10.1016", "10.1021", "10.1038", "10.1039",
"10.1107"]
# Generate unique CSV file name
csv_path = next(os.path.join(os.getcwd(), f"dictionary_{counter}.csv") for counter in
range(1000) if not os.path.exists(os.path.join(os.getcwd(), f"dictionary_{counter}.csv")))
csv_data = []

# Process each folder and PDF, and collect image data for CSV
for foldername in folders_to_process:
    folder_path = os.path.join(base_folder, foldername)
    if os.path.isdir(folder_path):
        csv_data += [(f"{foldername}/{filename.replace('-si.pdf', '').replace('.pdf', '')}",
foldername, filename, image_name, page_num)
                     for filename in os.listdir(folder_path) if filename.endswith(".pdf")
                     for image_name, page_num in pdf_to_images(os.path.join(folder_path,
filename), folder_path)]

# Write data to CSV
with open(csv_path, 'w', newline='') as csvfile:
    csv.writer(csvfile).writerows([["DOI", "Folder Name", "PDF Name", "Image Name", "Page
Number"]] + csv_data)
```

Figure S16. Python code to convert PDF literature files into PNG image files by page, with each PDF file divided into multiple images, one image per page. The code was generated with the assistance of ChatGPT.

```python
import pandas as pd
import requests
import base64

def encode_image(image_path): # Function to encode the image for OpenAI API
    with open(image_path, "rb") as image_file:
        return base64.b64encode(image_file.read()).decode('utf-8')

def analyze_image(image_path, api_key): # Function to analyze the image using OpenAI's API
    prompt = """..."""  # (Place prompt here)
    base64_image = encode_image(image_path)
    headers = {"Content-Type": "application/json", "Authorization": f"Bearer {api_key}"}
    payload = {
        "model": "gpt-4-vision-preview",
        "messages": [{"role": "user", "content": [{"type": "text", "text": prompt}, {"type":
"image_url", "image_url": {"url": f"data:image/jpeg;base64,{base64_image}"}}]}],
        "max_tokens": 300
    }
    try:
        response = requests.post("https://api.openai.com/v1/chat/completions",
headers=headers, json=payload)
        response.raise_for_status()
        return response.json()['choices'][0]['message']['content']
    except Exception as e:
        print(e)
        return ""

def main_process(api_key, csv_file_path): # Main processing function
    img_api_df = pd.read_csv(csv_file_path)
    for index, row in img_api_df.iterrows():
        img_path = row['img']
        if pd.isna(row['text']) or "Error:" in str(row['text']).strip():
            print(f"Processing Image No. {index+1}: {img_path}")
            text = analyze_image(img_path, api_key)
            if text:
                img_api_df.at[index, 'text'] = text
                img_api_df.to_csv(csv_file_path, index=False)
            else:
                print(f"Failed to analyze image: {img_path}")
        else:
            print(f"Image No. {index+1} already processed.")
    print("Processing completed!")

if __name__ == '__main__':
    api_key = "your-api-key"  # Replace with your actual API key
    csv_file_path = 'img_API.csv'  # Path to your CSV file
    main_process(api_key, csv_file_path)
```

Figure S17. Python code to access GPT-4 via API. This script encodes images into a base64 format and sends them, along with a textual prompt, to GPT-4V for analysis. The script was generated with the assistance of ChatGPT and designed to iterate through a CSV file, processing images referenced by their file names.

```python
import pyautogui, pandas as pd, json, pyperclip, time, sys

# Define constants
TimeSleep, exit_flag, PointFileName= 60, False, "./config/points.json"

# Function to get region coordinates
def get_region(screen_width, screen_height, region_name):
    region_width, region_height = screen_width // 2, screen_height // 2
    regions = {'left': (0, 0, region_width, screen_height), 'right': (region_width, 0,
region_width, screen_height),
               'down': (0, region_height, screen_width, region_height)}
    if region_name in regions: return regions[region_name]
    raise ValueError("Invalid region name.")

# Function to find 'Enter' button on screen
def findEnter():
    match = pyautogui.locateOnScreen('./config/enter.png',
region=get_region(*pyautogui.size(), 4))
    return pyautogui.center(match) if match else None

# Function to process image and get result
def GetResult(imgPath):
    points, Recjson, error = json.load(open(PointFileName)), json.load(open(RecFileName)),
False
    pyautogui.click(points['create_dialogue']) # Create a new dialogue
    pyautogui.click(points['click_image_upload_button'], clicks=2) # Upload the image
    pyautogui.click(points['click_text_input_box'])
    pyperclip.copy(imgPath)
    pyautogui.hotkey('ctrl', 'v') # Paste image path to the input box for uploading
    pyautogui.press('enter')
    for _ in range(20):
        if exit_flag: sys.exit()
        time.sleep(1)
        if findEnter(): break
    else: return "Error: Upload failed.", True
    pyautogui.click(findEnter()[0] - 100, findEnter()[1]) # Find the input text box
    pyperclip.copy(question)
    pyautogui.hotkey('ctrl', 'v')
    pyautogui.press('enter')
    for _ in range(100): # Image uploading is successful when enter bottom turns grey
        if exit_flag or pyautogui.locateOnScreen('./config/grayEnter.png',
region=get_region(*pyautogui.size(), 4)): break
        pyautogui.click(points['blank_area']) # Click blank area on the right and scroll down
        pyautogui.scroll(-3000)
        time.sleep(1)
    pyautogui.click(points['blank_area']) # Scroll all the way down to the bottom
    pyautogui.scroll(-3000)
    for _ in range(20): # Detect the “Copy” icon next to the output text
        if exit_flag or pyautogui.locateOnScreen('./config/copy.png',
region=get_region(*pyautogui.size(), "right")): break
```

```
        pyautogui.click(points['blank_area'])
        pyautogui.scroll(-3000)
        time.sleep(0.5)
    else: return "Error: Output Not Found", True
    pyautogui.click(pyautogui.center(pyautogui.locateOnScreen('./config/copy.png',
region=get_region(*pyautogui.size(), "right"))))
    return pyperclip.paste(), error

# Main execution block
if __name__ == '__main__':
    t = threading.Thread(target=check_keyboard_input)
    t.daemon = True
    t.start()
    df = pd.read_csv('img.csv', encoding="utf-8")
    ErrorTime = 0
    errorFlag = False
    for index, imgPath in enumerate(df['img']):
        if errorFlag == True:
            errorFlag = False
            time.sleep(TimeSleep)
            if ErrorTime == 3 :
                time.sleep(60*30) # Reach time limit, retry after 30 minutes
                ErrorTime = 0 #reset counting
        if (df.loc[index, "text"].strip() == "" or pd.isna(df.loc[index, 'text']) or
len(df.loc[index, "text"].strip()) <= 5 or any(keyword in df.loc[index, 'text'] for keyword
in ["Error:", "upload the image", "analyze", "Sorry", "sorry"]) or ("Figures" not in
df.loc[index, 'text'] and "Nitrogen Isotherm" not in df.loc[index, 'text'])):
            text, errorFlag = GetResult(imgPath)
            df.loc[index, 'text'] = text
            df.to_csv("img.csv", encoding='utf-8', index=False)
            if errorFlag == True:
                ErrorTime +=1
                print(text)
                continue
            else:
                ErrorTime = 0 # reset error time
```

Figure S18. Python code to access GPT-4 via web interface. The script automates interactions with a web-based platform to retrieve GPT-4 responses. It utilizes 'config.png' files for graphical element recognition, guiding the script to locate specific buttons and input fields on the screen. 'Points' refer to pre-defined coordinates for dialogue creation, image upload buttons, and text input fields in the script, corresponding to these graphical elements, ensuring accurate navigation and interaction within the web interface.

```python
import pandas as pd
import re

def extract_choices(output):
    # Normalize the string to handle different cases and remove square brackets
    normalized_output = output.replace('\n', ' ').replace('[', '').replace(']', '').strip()

    # Check if 'Figures:' is in the output
    if 'Figures:' not in normalized_output:
        return [6]  # Default to choice 6 in these cases

    # Extract the part of the output before 'Nitrogen Isotherm:' if it's present
    nitrogen_isotherm_index = normalized_output.find('Nitrogen Isotherm:')
    if nitrogen_isotherm_index != -1:
        normalized_output = normalized_output[:nitrogen_isotherm_index]

    # Extract the part of the output starting from 'Figures:'
    figures_index = normalized_output.find('Figures:')
    if figures_index != -1:
        normalized_output = normalized_output[figures_index:]

    # Find all occurrences of choices in the format (number), making sure they are within 1-6
    choices_str_list = re.findall(r'\((\d)\)', normalized_output)
    choices = sorted(set([int(num) for num in choices_str_list if int(num) in range(1, 7)]))

    return choices

if __name__ == '__main__':

    # Read the CSV file
    df = pd.read_csv('img.csv') # Read the CSV file
    # Apply the function to each row in 'text' column and store the result in a new column
    df['GPT-4V Output'] = df['text'].apply(extract_choices)
    # Write the updated DataFrame back to the CSV file
    df.to_csv('img.csv', index=False)
```

Figure S19. Python code to extract the label from GPT-4V's responses. This script processes textual output from GPT-4V, specifically targeting and extracting categorized choices. It normalizes the response text for consistent processing, removes extraneous characters, and identifies distinct numerical labels indicating different response types or classifications. The code was generated with the assistant of ChatGPT and the outputs are available in the ESI data set.

```
import pandas as pd
import re

def extract_nitrogen_info(row):
    output = row['text']
    info = {'Nitrogen Isotherm': "N/A", 'Compound': "N/A", 'Porosity': "N/A", 'Hysteresis':
"N/A",
            'Saturation': "N/A", 'Position': "N/A"}

    # Extracting and cleaning each piece of information
    for key in info.keys():
        try:
            # Extracting the value
            value = output.split(f'{key}: ')[1].split('\\n')[0]

            # Removing the "[Answer XXXXX]" pattern
            clean_value = re.sub(r'\\[\\s*Answer\\s*:\\s*(.*?)\\s*\\]', r'\\1', value)

            # Further cleaning the value by removing unwanted characters
            clean_value = ''.join(filter(lambda char: char.isalnum() or char in " .,-;/()[]",
clean_value))

            # Updating the info dictionary
            info[key] = clean_value.strip() if clean_value else "N/A"
        except IndexError:
            continue  # If the key is not found, it remains "N/A"

    return pd.Series(info)

if __name__ == '__main__':

    # Load the CSV file
    file_path = 'img.csv'
    data = pd.read_csv(file_path)

    # Applying the extraction function to each row
    extracted_data = data.apply(extract_nitrogen_info, axis=1)

    # Updating the dataframe with the extracted data
    data.update(extracted_data)

    # Saving the updated dataframe to the CSV file
    data.to_csv(file_path, index=False)
```

Figure S20. Python code to extract the nitrogen isotherm information from GPT-4V's responses. This script eliminates irrelevant characters and extracts specific parameters from GPT-4V's formulated answers. These parameters include the figure number of the nitrogen isotherm, the name of the compound, the indicated porosity, the presence of hysteresis, the saturation plateau, and the position of the bounding box. The code was generated with the assistant of ChatGPT.

Table S1. Equations used to calculate the performance metrics.

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

Note: TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative.

Table S2. Comparative evaluation of accuracy in recognizing different plot types from published literature. This table presents a comparison of accuracy rates between the web interface and the API methods for recognizing various plot types. Note that the API method was tested on the first 1,000 images due to API request limitations, whereas the web interface results reflect an analysis of all 6,240 images.

Plot Type	Web Accuracy	API Accuracy
Nitrogen Isotherm	99.5%	97.9%
Power X-Ray Diffraction	99.2%	95.3%
Thermogravimetric Analysis	99.2%	94.9%
Crystal Structure or Topology	98.1%	91.8%
Other Gas Sorption Isotherm	95.0%	96.9%
None of Above	94.3%	88.7%

Table S3. Comparative analysis of image processing capabilities across 6240 pages from 346 peer-reviewed publications.

Parameter	Web	API
Cost (USD)	\$20	\$125
Processing Time (days)	12	0.4
Number of Images	6240	6240
Number of Context Tokens	N/A	122,221,798
Number of Generated Tokens	203,266	118,866
Average Cost per Paper (USD)	\$0.057	\$0.361

Note: This study analyzed a total of 346 papers, collectively encompassing 6,240 pages. For the Web Interface, a single account, priced at \$20 per month, can process approximately 500 images/pages daily. The input context token is not applicable, as the token count for uploads via the web interface depends on the resolution and size of the figure. The count of generated tokens for the Web Interface is based on the outputs collected. For the API, each request costs approximately 2000 tokens, equating to \$0.02 per image/page, with an average processing time of 5.2 seconds per request.

Table S4. Comparative analysis of image processing capabilities within a fixed budget of $100.

Parameter	Web	API
Cost (USD)	$100	$100
Processing Time (days)	30	0.3
Estimated Number of Images	75,000	5,000
Estimated Number of Papers	4,167	277
Average Cost per Paper (USD)	$0.024	$0.361

Note: On average, a single paper, inclusive of Supplementary Information and the main text, contains 18 pages. This was deduced from a sample of 346 papers in this study, which collectively accounted for 6,240 pages. For the Web Interface, a single account priced at $20 per month is capable of processing around 500 images/pages daily. Utilizing five such accounts escalates the monthly image processing capacity to a total of 75,000 images. For the API, each request costs approximately 2000 tokens, equating to $0.02 per image/page, with an average processing time of 5.2 seconds per request.

References

1. Z. Zheng, H. L. Nguyen, N. Hanikel, K. K.-Y. Li, Z. Zhou, T. Ma and O. M. Yaghi, *Nat. Protoc.*, 2023, **18**, 136–156.
2. B. Wang, X.-L. Lv, D. Feng, L.-H. Xie, J. Zhang, M. Li, Y. Xie, J.-R. Li and H.-C. Zhou, *J. Am. Chem. Soc.*, 2016, **138**, 6204-6216.
3. Z. Zheng, N. Hanikel, H. Lyu and O. M. Yaghi, *J. Am. Chem. Soc.*, 2022, **144**, 22669-22675.
4. X. Wang, X. Han, J. Zhang, X. Wu, Y. Liu and Y. Cui, *J. Am. Chem. Soc.*, 2016, **138**, 12332-12335.
5. C.-C. Liang, Z.-L. Shi, C.-T. He, J. Tan, H.-D. Zhou, H.-L. Zhou, Y. Lee and Y.-B. Zhang, *J. Am. Chem. Soc.*, 2017, **139**, 13300-13303.
6. Y.-B. Zhang, H. Furukawa, N. Ko, W. Nie, H. J. Park, S. Okajima, K. E. Cordova, H. Deng, J. Kim and O. M. Yaghi, *J. Am. Chem. Soc.*, 2015, **137**, 2641-2650.
7. H. L. Nguyen, C. Gropp, N. Hanikel, A. Möckel, A. Lund and O. M. Yaghi, *ACS Cent. Sci.*, 2022, **8**, 926-932.
8. M. Kim, J. F. Cahill, H. Fei, K. A. Prather and S. M. Cohen, *J. Am. Chem. Soc.*, 2012, **134**, 18082-18088.
9. Z. Zheng, O. Zhang, C. Borgs, J. T. Chayes and O. M. Yaghi, *J. Am. Chem. Soc.*, 2023, **145**, 18048–18062.