

# MATVIX: Multimodal Information Extraction from Visually Rich Articles

Ghazal Khalighinejad<sup>1</sup>, Sharon Scott<sup>2</sup>, Ollie Liu<sup>3</sup>, Kelly Anderson<sup>2</sup>, Rickard Stureborg<sup>1</sup>, Aman Tyagi<sup>2</sup>, Bhuwan Dhingra<sup>1</sup>

<sup>1</sup>Duke University

<sup>2</sup>The Procter & Gamble Company

<sup>3</sup>University of Southern California

## Abstract

Multimodal information extraction (MIE) is crucial for scientific literature, where valuable data is often spread across text, figures, and tables. In materials science, extracting structured information from research articles can accelerate the discovery of new materials. However, the multimodal nature and complex interconnections of scientific content present challenges for traditional text-based methods. We introduce MATVIX, a benchmark consisting of 324 full-length research articles and 1,688 complex structured JSON files, carefully curated by domain experts. These JSON files are extracted from text, tables, and figures in full-length documents, providing a comprehensive challenge for MIE. We introduce an evaluation method to assess the accuracy of curve similarity and the alignment of hierarchical structures. Additionally, we benchmark vision-language models (VLMs) in a zero-shot manner, capable of processing long contexts and multimodal inputs, and show that using a specialized model (DePlot) can improve performance in extracting curves. Our results demonstrate significant room for improvement in current models. Our dataset and evaluation code can be found at <https://matvix-bench.github.io/>

## 1 Introduction

Multimodal information extraction (MIE) has become a key research focus, aiming to extract structured information from both text and visual content (Liu et al., 2019; Dong et al., 2020; Oka et al., 2021; Sun et al., 2024). This is particularly important in scientific literature, where valuable details are often spread across text, figures, and tables. The complex nature of scientific content, combined with the need to combine information from multiple sources, presents substantial challenges for traditional text-based extraction methods.

In materials science, MIE is crucial as research articles contain valuable data that can accelerate the discovery of new materials. Tools like

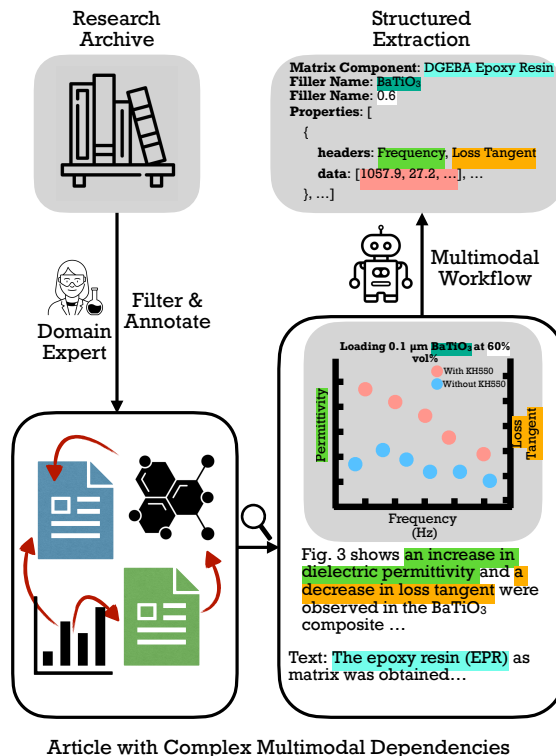


Figure 1: Example of a materials research article illustrating interconnected data between text and figures, with a JSON structure capturing sample properties and composition details. Note that the Matrix component is extracted from the text.

GNoME (Merchant et al., 2023) show how extracting structured databases from these publications can improve discovery efficiency. However, this process is complicated by the multimodal nature of scientific articles and the complex connections between data points. **Figures are particularly critical, as they often contain essential information about material properties not present in the text,** making their accurate extraction vital for comprehensive information retrieval (Polak and Morgan, 2024).

Recent methods like DePlot (Liu et al., 2023a)

tackle visual plot reasoning by converting plot images into linearized tables to enable pretrained LLMs to reason over visual data with minimal training. This approach allows LLMs to leverage few-shot reasoning for tasks like chart QA (Masry et al., 2022). However, DePlot’s focus on simple plot-to-table conversion makes it less suitable for the complex MIE needed to handle interconnected data spanning long contexts, including text, tables, and multiple figures within full-length scientific documents.

To address this gap, we introduce a novel benchmark, MATVIX, focused on obtaining structured information from materials science articles within the domains of polymer nanocomposites and polymer biodegradation. Existing datasets such as FUNSD (Jaume et al., 2019), CORD (Kim et al., 2021), and Kleister (Stanisławek et al., 2021) have made significant contributions to document analysis and information extraction, especially in handling complex layouts and long documents. However, **they do not address the complexities of  $N$ -ary relation extraction**, focusing instead on simpler tasks like named entity recognition (NER), which typically involves identifying predefined entities without capturing intricate relationships between them. Furthermore, these datasets do not consider scientific documents, which often contain specialized language and complex figures. Previous work (Dagdelen et al., 2024; Cheung et al., 2024) in materials science has explored  $N$ -ary relation extraction, but primarily from text-only abstracts or short texts. PNCEXtract (Khalighinejad et al., 2024) represents a step forward by considering full-length articles, yet it remains limited to text-based content. In contrast, **MATVIX considers all elements in long scientific documents, including text, tables, and figures**, providing a comprehensive challenge for MIE research (see Table 1 for a comparison).

Previous work using pretrained models like LayoutLM (Xu et al., 2020), LayoutLMv2 (Xu et al., 2022), and domain-specific models such as MatSciBERT (Gupta et al., 2022a) has advanced the extraction of structured data from visually rich documents. However, these models are not suitable for our task, as extracting complex  $N$ -ary relations from long documents exceeds the token limit of 512 tokens in BERT and LayoutLM models. Additionally, our task involves generating hierarchical JSON files, which is more complex than simple entity recognition or relation extraction and requires

a sequence-to-sequence approach. Splitting the documents to fit these models is not feasible, as it disrupts the extraction of interconnected relations across the entire document.

**MATVIX addresses the challenge of long document processing** by utilizing VLMs in a zero-shot manner. These models can process long input contexts (OpenAI, 2023) without sacrificing accuracy and have demonstrated strong performance in tasks requiring both textual and visual reasoning. Given their capabilities, one might wonder if we can simply input a scientific paper, complete with all its images, into a VLM to obtain structured data. In this paper, we benchmark these models and compare them against simpler baselines, showing that there is still substantial room for improvement.

**Our task involves extracting complex hierarchical structures from scientific documents**, where traditional evaluation metrics fall short. Our method first aligns compositions, which serve as the identity of each sample, and then evaluates the properties represented by curves. To measure the closeness between predicted and ground truth curves, we use the Fréchet distance, which captures how similar the overall trends are.

Our results show that while VLMs show potential, significant improvements are needed. We also demonstrate that combining the best-performing VLMs with the specialized DePlot model enhances information extraction from figures.

## 2 The MATVIX Benchmark

In this section, we first describe our dataset, including the problem definition and dataset preparation, and then explain our evaluation method for the task. MATVIX focuses on two critical domains: Polymer Nanocomposites (PNC) and Polymer Biodegradation (PBD). Structured data in these fields is crucial for accelerating research and discovery, as it allows scientists to efficiently analyze relationships between material compositions and properties, which are often spread across text, figures, and tables within research articles. The emphasis on PNC and PBD reflects their significant representation within the field of materials science. The focus on PNC and PBD is justified by their significant presence in the field; a Google Scholar search yields approximately 95,600 results for “materials science”, 32,100 for “polymer nanocomposites”, and 17,200 for “polymer biodegradation”.

Dataset	Complex Layout	Long Doc	$N$ -ary RE	Scientific	Multimodal
FUNSD (Jaume et al., 2019)	✓	×	×	×	×
CORD (Kim et al., 2021)	✓	×	×	×	×
Kleister (Stanisławek et al., 2021)	✓	✓	×	×	×
PolyIE (Cheung et al., 2024)	×	×	✓	✓	×
PNCEXtract (Khalighinejad et al., 2024)	×	×	✓	✓	×
Ours	✓	✓	✓	✓	✓

Table 1: Comparison of Dataset Characteristics. Ours requires  $N$ -ary relation extraction and is the only dataset that requires extraction from the scientific domain and reasoning over plots.

Each PNC and PBD sample in the dataset is represented as a structured JSON object that captures both the chemical composition and associated property data. A sample refers to a specific instance of a material with a defined chemical composition and measured properties. This structured format provides detailed information about the materials’ compositions and includes the numerical property data needed for training machine learning models. These numerical data points are particularly important for developing models that can predict the relationship between material composition and performance. By leveraging these structured data representations, researchers can conduct large-scale analysis and modeling to advance material discovery and optimization (Ward et al., 2016).

## 2.1 Problem Definition

Let  $\mathcal{D} = D_1, D_2, \dots, D_N$  denote our dataset, which consists of  $N$  total articles, where  $N = 324$ . Among these articles, 231 are from the PNC domain and 93 are from the PBD domain. For each article  $D_i \in \mathcal{D}$ , there is an associated list of samples  $\mathcal{S}_i$ , comprising various PNC or PBD samples. Formally,  $\mathcal{S}_i$  is defined as:

$$\mathcal{S}_i = s_{i1}, s_{i2}, \dots, s_{in_i}, \quad (1)$$

where  $s_{ij}$  represents the  $j$ -th sample (either PNC or PBD) in the sample list of the  $i$ -th article, and  $n_i$  denotes the total number of samples in  $\mathcal{S}_i$ . Each sample  $s_{ij}$  is a JSON object. The structure of the JSON object of interest is provided in Appendix C.

The goal is to extract the relevant information from each article  $D_i$  to populate the corresponding sample list  $\mathcal{S}_i$ . This involves identifying and extracting the values for each of the entries in the JSON object for every PNC or PBD sample mentioned in the article.

Statistic	$\mathcal{D}_{PNC}$	$\mathcal{D}_{PBD}$
Total Papers	231	93
Total Samples	1396	292
Avg. Samples per Paper	6	3
Avg. Tokens per Paper	8905	8456

Table 2: Data Statistics for  $\mathcal{D}_{PNC}$  and  $\mathcal{D}_{PBD}$

## 2.2 Polymer Nanocomposites (PNC)

### 2.2.1 Overview

Our PNC dataset, derived from the Nanomine data repository (Zhao et al., 2018), extends PNCEXtract (Khalighinejad et al., 2024) by including both compositions and properties of PNC samples.

Each PNC sample  $s_{ij}$  is represented as a structured JSON object comprising two main sections: composition and properties. The composition section specifies the matrix and filler materials along with their attributes, while the properties section contains specific characteristics of the sample, including their names, measurement conditions, and corresponding data points (see Appendix C for the JSON format).

We focus on six key properties frequently studied in the dataset: Thermal, Electrical, Mechanical, Viscoelastic, Volumetric, and Rheological. These properties are prioritized because they are not only the most commonly reported in research papers but also critical in determining the performance of polymer nanocomposites. Each property includes numerical data collected under various experimental conditions, specified in the JSON headers, with the actual data points listed. Examples of property representations and their associated plots are provided in Appendix 2.

An analysis of the Nanomine data repository reveals the distribution of these properties across 4,186 samples: Thermal (26.4%), Electrical (29.6%), Mechanical (14.1%), Viscoelastic (21.0%), Volumetric (3.3%), Rheological (5.4%), and Others (0.1%). Therefore, we concentrate on

the six key properties due to their significance in the dataset.

### 2.2.2 Data Collection

Our dataset is based on data from the NanoMine repository (Zhao et al., 2018), a comprehensive resource for PNC data structured around an XML-based schema for the representation and sharing of nanocomposite materials information. The original data in NanoMine was collected and stored using Excel templates provided to materials researchers. However, this structure is not consistent and includes a large template with 43 attributes in the Materials Composition section and over 20 different properties, all organized in formats that are challenging to process.

To address these inconsistencies, we standardized and cleaned the NanoMine repository data. We categorized the 20 properties into six main categories—Thermal, Electrical, Mechanical, Viscoelastic, Volumetric, and Rheological—ensuring that the data points within these properties were restructured and aligned accordingly. The process involved organizing the data into a structured JSON format suitable for our analysis and modeling purposes.

This categorization and cleaning effort were validated by the experts in the field, to ensure the accuracy of the structured data.

## 2.3 Polymer Biodegradation (PBD)

### 2.3.1 Overview

Our PBD dataset focuses on extracting information related to the biodegradation of polymers. The dataset was collected by experts in biodegradable polymers, ensuring high accuracy through meticulous data collection and verification. The dataset includes 47 research papers in the test set and 46 papers in the validation set, with a total of 159 samples for testing and 133 samples for validation.

Each PBD sample  $s_{ij}$  is captured as a structured JSON object that details both the composition and biodegradation characteristics of the polymer sample. The structure captures essential information regarding the polymer’s type, characteristics, and biodegradation data, including conditions and corresponding measurements. For the detailed structure of the JSON format, please refer to Appendix C. The biodegradation results are typically presented in figures, showing plots of conditions versus biodegradation percentage (refer to Appendix 3).

### 2.3.2 Data Collection

Two materials science experts curated a collection of 93 research papers focused on biodegradable materials, selecting high-quality articles from reputable journals. They first identified key compositional attributes consistent across polymer biodegradation samples. One expert extracted these details, while the second verified their accuracy.

After validating compositions, the experts extracted properties, which were often presented in text, tables, or figures. For plots—commonly showing biodegradation rates—the PlotDigitizer<sup>1</sup> tool was used to trace curves and extract (x, y) data points. This process involved calibrating axes and converting visual information into structured JSON files. For a detailed explanation of the data collection and digitization process, see the Appendix 4.

## 3 Evaluation

Each paper contains a set of samples, and each sample is characterized by its composition and properties. The composition is represented as a set of strings, while the properties are captured as a list of curves. To evaluate the accuracy of the predicted samples against the ground truth, we follow a two-step process: first, we evaluate the alignment of the compositions within the samples, and then we assess the matching of the properties for the aligned samples. The reason for this approach is that the composition defines what the sample is, providing its identity, while the properties describe the characteristics of that specific sample. Therefore, it is crucial to first match the compositions correctly, ensuring that we are comparing the same types of samples, before evaluating the properties within those matched samples.

The evaluation employs the F1 Score for composition alignment and two specialized metrics, the Curve Similarity Score (CSS) and the Curve Alignment Score (CAS), for property evaluation.

**Composition Alignment** To assess the alignment between predicted and ground truth compositions within each sample, we treat this as a maximum bipartite matching problem. Each composition consists of a set of strings, and we aim to find the best correspondence between the predicted and ground truth compositions. We use the Munkres algorithm (Hungarian algorithm) (Kuhn, 1955) to solve this bipartite matching problem, optimizing

<sup>1</sup><https://plotdigitizer.com/>

for the highest possible F1 Score. If there are more ground truth samples for a paper than predicted samples, the unmatched ground truth samples are considered false negatives. Conversely, if there are more predicted samples than ground truth samples, the unmatched predicted samples are considered false positives.

**Curve Similarity Score (CSS)** Once compositions are aligned, we evaluate the properties within these matched samples. Each property is represented by a curve, which is defined as a list of (x, y) points, and the CSS is introduced as a quantitative measure of similarity between the predicted and ground truth properties. These properties are typically plotted as the relationship between a variable and its corresponding response, such as how dielectric permittivity changes with frequency or temperature, which are commonly reported in research papers. The trends captured in these curves are crucial, as they convey important information about the material’s behavior. Therefore, accurately extracting and evaluating these curves is essential.

To quantify the similarity, we use the Levenshtein distance to compare the headers (x and y-axis labels) and the Fréchet distance to compare the ground-truth data points of the curves. The Fréchet distance measures the similarity between two curves by finding the smallest of the maximum pairwise distances. To compute this for polygonal curves, the discrete Fréchet distance is used, as shown by [Wien et al. \(1994\)](#), which involves determining the shortest path through a coupling sequence that matches points between the curves while maintaining their order.

The CSS, given a predicted curve  $c_p$  and a ground truth curve  $c_t$ , is calculated as follows:

$$\text{CSS}(c_p, c_t) = \left(1 - \text{nl}_{\text{lev}}(d_{\text{lev}}(h_p, h_t))\right) \left(1 - \text{nl}_{\text{frech}}(d_{\text{frech}}(c_p, c_t))\right), \quad (2)$$

where  $d_{\text{lev}}(h_p, h_t)$  is the Levenshtein distance between the headers of the predicted ( $h_p$ ) and ground truth ( $h_t$ ) curves, and  $d_{\text{frech}}(c_p, c_t)$  is the Fréchet distance between the predicted and ground truth curves. The normalization functions  $\text{nl}_{\text{lev}}$  and  $\text{nl}_{\text{frech}}$  are defined as follows:

$$\text{nl}_{\text{lev}}(d_{\text{lev}}) = \min\left(1, \frac{d_{\text{lev}}(h_p, h_t)}{\max(\text{len}(h_p), \text{len}(h_t))}\right)$$

$$\text{nl}_{\text{frech}}(d_{\text{frech}}) = \min\left(1, \frac{d_{\text{frech}}(c_p, c_t)}{\|c_t\|}\right)$$

where  $\text{len}(h)$  represents the length of the header, and  $\|c_t\|$  denotes the norm of the ground truth curve data.

This approach addresses several limitations inherent in the metric introduced in DePlot, which misses critical information about the alignment of trends. By incorporating the Fréchet distance, the CSS provides a more comprehensive evaluation, capturing both the trend similarities.

**Curve Alignment Score (CAS)** The CAS metric identifies the best match between predicted and ground truth curves when multiple curves are present within a sample. Let  $X \in \mathbb{R}^{N \times M}$  be a binary matrix where  $X_{ij}$  indicates the assignment of the  $i$ -th predicted curve to the  $j$ -th ground truth curve, based on the Munkres algorithm. The CAS is calculated as follows:

$$\text{CAS} = \frac{1}{\max(N, M)} \sum_{i=1}^N \sum_{j=1}^M X_{ij} \cdot \text{CSS}(c_{p_i}, c_{t_j}),$$

where  $\text{CSS}(c_{p_i}, c_{t_j})$  represents the Curve Similarity Score between the  $i$ -th predicted curve ( $c_{p_i}$ ) and the  $j$ -th ground truth curve ( $c_{t_j}$ ).

### 3.1 Human Evaluation

We conduct a human evaluation study to assess the effectiveness of our evaluation metric. A total of 50 plot-prediction pairs (GPT-4o predictions) are randomly sampled from our dataset, representing a range of different scores.

Three human annotators, each with relevant expertise in the field, evaluate these pairs. For each sample, the annotators are presented with both the ground-truth plot and the model’s predicted plot. They are asked to assess the quality of the prediction based on two specific questions: (1) Are the axes labeled correctly? (2) Is the trend of the predicted curve consistent with the ground truth? (See [Appendix 9](#) for details.)

The human scores for the first question (regarding headers) were averaged and compared to the automated header scores after alignment, calculated as  $1 - \text{nl}_{\text{lev}}(d_{\text{lev}}(h_p, h_t))$ . Similarly, the human scores for the second question (regarding curves) were averaged and compared to the automated curve scores after alignment, calculated as  $1 - \text{nl}_{\text{frech}}(d_{\text{frech}}(c_p, c_t))$ .

For comparison, we used both Pearson’s  $r$  and Spearman’s  $\rho$  correlation coefficients. [Table 3](#) in-

icates positive correlations between the human judgments and the automated metric scores.

	Coefficient	p-value
Pearson $r$ (curves)	0.887	$4.17 \times 10^{-7}$
Spearman $\rho$ (curves)	0.717	0.00055
Pearson $r$ (headers)	0.930	$8.54 \times 10^{-9}$
Spearman $\rho$ (headers)	0.921	$2.28 \times 10^{-8}$

Table 3: Correlation results between model scores and human evaluations for header and curve scores, using Pearson’s  $r$  and Spearman’s  $\rho$ . Human scores represent the average ratings from three annotators.

## 4 Benchmarking VLMs

The objective is to extract structured data from materials science documents. To achieve this, we first convert each PDF document into LaTeX format using Mathpix (<https://mathpix.com/>). This approach generates a TeX file that shows the structure of the paper, including sections, subsections, and all images. We then employ Visual Language Models (VLMs) in a zero-shot manner to extract structured JSON data.

During our preliminary experiments, we observed that providing both the entire LaTeX file and all associated images as input to the VLMs leads to suboptimal results. Additionally, since we must input images one at a time, it becomes costly. To address this, we devised a multi-step pipeline (see Appendix 5):

- **Text Information Extraction:** First, we use an LLM to extract structured information from the text in the LaTeX document.
- **Information Expansion:** For each image, we then prompt the VLM to expand the extracted information based on the text and the images. This expansion is handled individually for each image.
- **Information Integration:** Given that multiple images are typically associated with a document, we merge all the expanded information from the different images to create a comprehensive, structured dataset.

Formally, the steps can be described as follows:

$$\hat{S}_i^{\text{text}} = \text{LLM}(D_i^{\text{text}}) \quad (3)$$

$$\hat{S}_i^{\text{img},k} = \text{VLM}(\hat{S}_i^{\text{text}}, I_k), \quad \forall k \in [K] \quad (4)$$

$$\hat{S}_i = \text{Merge}(\hat{S}_i^{\text{text}}, \{\hat{S}_i^{\text{img},k}\}_{k=1}^K) \quad (5)$$

where  $D_i^{\text{text}}$  is the textual data in document  $i$ ,  $\hat{S}_i^{\text{text}}$  is the predicted sample list derived only from the textual data of document  $i$ , and  $I_k$  is the  $k$ th image in the document.  $\hat{S}_i^{\text{img},k}$  represents the expanded information obtained by the VLM for the  $k$ -th image using  $\hat{S}_i^{\text{text}}$  as context. Finally,  $\hat{S}_i$  merges the textual and image-based information to form a comprehensive structured dataset for document  $i$ .

## 5 Experiments

In this section, we present the results of modeling with VLMs on MATVIX.

### 5.1 Models and Setup

We use GPT-4-Turbo, GPT-4o, Claude-3-Haiku, Claude-3.5-Sonnet, and Gemini-Pro-1.5 in our experiments (OpenAI, 2023; Anthropic, 2024; Reid et al., 2024), which are instruction-tuned models and are prompted in a zero-shot manner. We also conducted preliminary experiments with the open-sourced Vicuna-7b-v1.5-16k (Chiang et al., 2023) model, but it failed to capture any meaningful structure. Additionally, we evaluate open-source models, including Qwen2.5-VL-7B-Instruct and Qwen2.5-VL-72B-Instruct (Yang et al., 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) and Mixtral-8x22B-Instruct-v0.1 (Jiang et al., 2024), and Llama-3.2-11B-Vision-Instruct-Turbo (et al., 2024). We evaluate these models against a custom baseline approach called Majority Vote. The baseline method selects the most common header and curve predictions from the validation set.

### 5.2 Baseline for Headers, Curves, and CAS

First, for the curves associated with each property (note that there are six properties in the PNC dataset), we calculate the average Fréchet Distance among all curves in the validation set. The curve that is the closest to all others (i.e., has the smallest cumulative Fréchet Distance) is selected as the baseline curve for that property. For the headers, we examine the validation set to determine the most common x-header and y-header for each property. These most frequent headers serve as the baseline headers.

Next, we consider the predictions from the LLM ( $\hat{S}_i^{\text{text}}$ ). We then add the baseline properties to each predicted composition. Specifically, for PNC, we calculate the average occurrence of each of the six properties per sample and include that many for each property. For PBD, we expand by adding

Model	Config	Polymer Nanocomp						Polymer Biodegradation					
		P	R	F1	Head.	Curve	CAS	P	R	F1	Head.	Curves	CAS
<b>API-based VLMs</b>													
GPT-4o	Base	–	–	–	06.17	13.96	02.42	–	–	–	93.39	15.66	14.57
	T-Only	74.51	48.87	58.80	14.94	14.54	04.70	35.53	18.80	23.60	55.37	31.07	21.78
	T+Img	70.48	46.37	55.75	11.29	12.64	03.23	35.99	18.90	23.76	43.70	28.60	15.18
GPT-4-Turbo	Base	–	–	–	06.93	13.42	02.43	–	–	–	94.84	15.56	15.04
	T-Only	68.95	45.10	54.28	12.83	11.68	03.21	26.58	16.37	19.37	46.49	23.78	13.43
	T+Img	68.34	43.14	52.41	14.92	10.67	01.87	31.50	20.61	23.98	39.45	26.76	13.21
Claude 3.5	Base	–	–	–	06.50	12.25	02.20	–	–	–	94.03	16.57	16.09
	T-Only	55.55	35.95	43.33	13.74	10.99	03.45	26.12	13.46	16.70	47.82	33.11	20.07
	T+Img	52.45	33.16	40.21	15.50	18.36	04.50	24.25	12.27	15.27	30.85	21.15	11.21
Claude 3	Base	–	–	–	06.15	11.77	02.00	–	–	–	95.56	16.81	16.35
	T-Only	51.93	3.03	40.07	11.06	04.88	01.54	41.66	20.25	26.01	24.90	20.21	06.78
	T+Img	51.46	31.62	38.74	12.80	16.88	03.79	44.71	21.12	27.29	13.98	18.09	03.30
Gemini 1.5	Base	–	–	–	06.65	13.98	02.44	–	–	–	93.20	17.61	16.69
	T-Only	72.28	47.69	57.37	18.29	07.07	02.70	23.08	16.55	18.52	52.36	25.63	16.02
	T+Img	71.17	46.76	56.32	14.96	19.08	03.98	18.98	12.70	14.43	18.29	05.65	02.97
<b>Open-source VLMs</b>													
Qwen2.5-VL-7B-Instruct	Base	–	–	–	06.47	12.20	02.18	–	–	–	91.12	13.30	14.92
	T-Only	54.37	35.73	42.85	12.46	14.20	03.82	02.04	01.53	01.75	47.30	22.23	10.72
	T+Img	53.83	35.46	42.53	09.99	13.23	03.71	02.04	01.53	01.75	47.30	11.78	05.86
Qwen2.5-VL-72B-Instruct	Base	–	–	–	06.54	12.99	02.29	–	–	–	93.81	15.44	16.35
	T-Only	69.81	46.18	55.43	14.19	12.65	04.42	38.33	33.57	35.61	40.41	44.25	14.58
	T+Img	70.24	46.48	55.79	13.64	18.12	05.40	28.03	27.27	27.60	36.32	26.71	13.11
Mistral-7B-Instruct-v0.3	Base	–	–	–	06.86	13.33	02.47	–	–	–	93.75	14.43	15.64
	T-Only	47.74	27.07	33.88	12.04	08.10	02.96	20.26	10.29	12.70	55.20	28.65	16.22
Mixtral-8x22B-Instruct-v0.1	Base	–	–	–	06.61	11.75	02.21	–	–	–	93.84	15.64	15.65
	T-Only	56.25	27.22	35.67	12.05	11.08	04.39	24.06	12.08	15.30	52.31	18.93	08.73
Llama-3.2-11B-Vision-Instruct-Turbo	Base	–	–	–	06.52	16.15	02.74	–	–	–	93.52	14.53	14.42
	T-Only	72.51	47.86	57.52	07.23	06.07	02.05	29.58	13.75	17.98	48.69	15.30	07.26
	T+Img	67.62	45.25	53.99	07.49	10.49	02.98	27.81	12.79	16.71	15.63	12.88	02.94

Table 4: Evaluation results for predicting compositions (P, R, F1) and properties (Headers, Curves, CAS) on the Polymer Nanocomposite (PNC) and Polymer Biodegradation (PBD) datasets under different configurations.

one baseline property directly, as there is only one property.

### 5.3 Results

Table 4 presents model performance in composition extraction, curve and header extraction, and curve alignment.

**Challenges in Curve Alignment.** Across both PNC and PBD datasets, models demonstrate stronger performance in composition extraction, with GPT-4o achieving the highest F1-Scores on the PNC dataset and Qwen2.5-VL leading on PBD. Curve extraction is more challenging, particularly in the PNC domain; the best model achieves only 5.40. This lower performance reflects the complexity of curve extraction, as it requires interpreting data from tables and figures, extracting curves accurately, and then aligning them with the appropriate composition. This process is inherently more complex than composition extraction, which requires fewer reasoning steps and draws from more straightforward data sources.

**Baseline Sometimes Outperforms in CAS.** The baseline configuration in some models provides better results. However, the best-performing models

outperform the baseline on both the PNC and PBD datasets. Interestingly, most models outperform the baseline in curve extraction but struggle with header extraction, revealing a gap in their ability to fully integrate and interpret all data components.

**T+Img Configuration Does Not Always Enhance Curve Extraction.** Surprisingly, incorporating both text and images (T+Img) does not consistently lead to better performance in curve extraction. While some information is only present in the images and not in the text, current VLMs seem more influenced by the noise from the images than by the useful data they contain. As a result, the T-Only configuration is often more effective, as it relies on focused textual information without the interference introduced by noisy visual inputs. Note that, in the T-Only case, since the entire LaTeX file is provided as input to the LLM, the tables are included, and in many cases, the important results from the figures are mentioned in the tables.

#### 5.3.1 Specialized Tools for Chart-to-Text Extraction.

While specialized tools for curve extraction exist, they are insufficient when information is intercon-

Table 5: Performance comparison of different configurations with and without DePlot. Results are from the PNC dataset.

Model	Head.	Curve	CAS
<b>GPT-4o</b>			
T-Only	14.94	14.54	04.70
T-Only (+DePlot)	15.69	19.72	06.27
T+Img	11.29	12.64	03.23
<b>Qwen2.5-VL-72B</b>			
T-Only	14.19	12.65	04.42
T-Only (+DePlot)	12.61	16.14	05.30
T+Img	13.64	18.12	05.40

nected with text, tables, and figures. Tools like PlotDigitizer can manually extract data points but lack automation, requiring annotators to calibrate axes and mark values manually. To our knowledge, no fully automated solutions are available.

We further test whether integrating DePlot with models can enhance results. Our approach involves first passing all images through DePlot to obtain its output. We then replace the figure section of the LaTeX file with DePlot’s output, which is formatted as a linearized table. This modified LaTeX file is fed into an LLM for evaluation. For the T+Img configuration, we provide the LLM’s prediction along with the original images to assess the combined performance. Table 5 shows that using DePlot can improve the results of the best-performing configurations.

## 6 Related Work

A plethora of prior works have devoted to unimodal information extraction (IE) with LLMs. See (Tchoua et al., 2019; Oka et al., 2021; Xie et al., 2023; Shetty et al., 2023; Dagdelen et al., 2024, *inter alia*) for an overview of their applications in scientific texts. In contrast, there lacks specialized IE systems that jointly operate on scientific documents that contain texts, tables, and images (Dong et al., 2020; Gupta et al., 2022b; Sun et al., 2024).

General-purpose foundation models (OpenAI, 2023; Reid et al., 2024; Anthropic, 2024) are appealing alternatives for such tasks; yet directly applying these models often yields subpar performance due to the complexity of document structures (Khalighinejad et al., 2024), their inability to reason over long contexts and/or multiple images (Reid et al., 2024), and performance differences across modalities (Li et al., 2024; Fu et al., 2024a).

Several recent works have endeavoured to adapt pre-trained foundation models for material science

(Gupta et al., 2022a,b; Song et al., 2023), but extending this fine-tuning approach to images has been challenging. This is attributed to subpar performance of open-weight VLMs (Yue et al., 2023) – noted for their lack of faithfulness (Fu et al., 2024b) and compositionality (Kong et al., 2023) compared to API-access models – as well as a lack of high-quality multimodal datasets in the material science domain (Miret and Krishnan, 2024). MATVIX aims to bridge this gap by contributing an expert-annotated, multimodal dataset over full-length scientific documents, and a workflow that achieves nontrivial performance according to a curated evaluation suite across different modalities.

In this respect, MATVIX connects more broadly to a growing body of works that evaluate LLMs as agentic systems (Liu et al., 2023b; Mialon et al., 2023; Koh et al., 2024; Liu et al., 2024; Xie et al., 2024). Compared to knowledge-intensive benchmarks, they arguably evaluate model capabilities more akin to daily workflows, and are robust against data contamination. MATVIX subsidizes this research with a dataset driven by scientific use cases, and offers a suite of *partial* evaluation metrics that enable users to identify areas of improvements compared to binary success metrics.

## 7 Conclusion and Future Work

We introduce MATVIX, an expert-annotated, multimodal information extraction benchmark developed from scholarly articles. These articles receive a minimal amount of pre-processing; they are thus endowed with diverse textual, tabular, and visual structures, all of which contain information important for scientific applications. A general workflow is proposed, and is evaluated against a suite of automatic evaluation metrics that ensure the accuracies of extracted data across all modalities. Results validate the performance of our workflow, and our automatic metrics agree with human evaluations.

There are many avenues for future work. One such example is exploring an agentic framework where the model utilizes various smaller models or tools to assist with the extraction task. For instance, as shown in Table 5, DePlot is helpful for image-to-table extraction. Additionally, we hypothesize that a BERT model specifically trained for NER and RE may achieve higher recall than generative LLMs. Therefore, integrating these components into an agentic framework could be a promising next step. Another direction is to validate the usefulness of



extracted information for domain scientists. In materials science, much of the extracted data is used to train downstream machine learning models (Xu et al., 2023). To assess the effectiveness of our system, we can compare the performance of models trained on the extracted data with those trained on ground truth data.

## 8 Limitations

The MATVIX benchmark provides valuable insights into multimodal information extraction in the PNC and PBD domains. However, there are several limitations to consider.

First, our benchmark is limited to these two specific domains within materials science. While this focus is important for advancing research in these fields, the findings may not generalize well to other scientific disciplines. Future work should explore expanding the dataset to include additional areas of science.

Additionally, we only considered a zero-shot approach in this paper. While this is effective for evaluating the generalization capabilities of VLMs, fine-tuning these models on domain-specific data could further improve their performance, though this was outside the scope of our current study.

Finally, our evaluation metrics, particularly for curve extraction, do not take into account the units of measurement, which can be critical for scientific analysis. While the Fréchet distance helps measure trend similarity, the absence of unit considerations limits the metric's ability to fully assess the accuracy of the extracted data. Future work should explore more domain-specific metrics that account for both trends and units to provide a deeper understanding of model performance.

## 9 Ethics Statement

We do not believe there are significant ethical issues associated with this research.

## 10 Acknowledgement

This research was supported by a gift from Procter & Gamble.

## References

Anthropic. 2024. [The claude 3 model family: Opus, sonnet, haiku](#).

Jerry Cheung, Yuchen Zhuang, Yinghao Li, Pranav Shetty, Wantian Zhao, Sanjeev Gramppurohit, Rampi

Ramprasad, and Chao Zhang. 2024. [POLYIE: A dataset of information extraction from polymer material scientific literature](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2370–2385, Mexico City, Mexico. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. [Vicuna: An open-source chatbot impressing gpt-4 with 90%\\* chatgpt quality](#).

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418.

Zhi-Min Dang, Yan-Fei Yu, Hai-Ping Xu, and Jinbo Bai. 2008. [Study on microstructure and dielectric property of the batio3/epoxy resin composites](#). *Composites Science and Technology*, 68(1):171–177.

Xin Luna Dong, Hannaneh Hajishirzi, Colin Lockard, and Prashant Shiralkar. 2020. [Multi-modal information extraction from text, semi-structured, and tabular data on the web](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 23–26, Online. Association for Computational Linguistics.

Aaron Grattafiori et al. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.

Deqing Fu, Ruohao Guo, Ghazal Khalighinejad, Ollie Liu, Bhuwan Dhingra, Dani Yogatama, Robin Jia, and Willie Neiswanger. 2024a. [Isobench: Benchmarking multimodal foundation models on isomorphic representations](#). In *First Conference on Language Modeling*.

Deqing Fu, Tong Xiao, Rui Wang, Wang Zhu, Pengchuan Zhang, Guan Pang, Robin Jia, and Lawrence Chen. 2024b. [Tldr: Token-level detective reward model for large vision language models](#). *arXiv preprint arXiv:2410.04734*.

Tanishq Gupta, Mohd Zaki, N. M. Anoop Krishnan, and Mausam. 2022a. [Matscibert: A materials domain language model for text mining and information extraction](#). *npj Computational Materials*, 8(1):102.

Tanishq Gupta, Mohd Zaki, NM Krishnan, et al. 2022b. [Discomat: distantly supervised composition extraction from tables in materials science articles](#). *arXiv preprint arXiv:2207.01079*.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. [Funsd: A dataset for form understanding in noisy scanned documents](#). *Preprint*, arXiv:1905.13538.

- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.
- Ghazal Khalighinejad, Defne Circi, L. Brinson, and Bhuwan Dhingra. 2024. [Extracting polymer nanocomposite samples from full-length documents](#). In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13163–13175, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.
- Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2021. Ocr-free document understanding transformer. *arXiv preprint arXiv:2111.15664*.
- Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. 2024. [VisualWebArena: Evaluating multimodal agents on realistic visual web tasks](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–905, Bangkok, Thailand. Association for Computational Linguistics.
- Xianghao Kong, Ollie Liu, Han Li, Dani Yogatama, and Greg Ver Steeg. 2023. Interpretable diffusion via information decomposition. *arXiv preprint arXiv:2310.07972*.
- H. W. Kuhn. 1955. [The hungarian method for the assignment problem](#). *Naval Research Logistics Quarterly*, 2(1-2):83–97.
- Zekun Li, Xianjun Yang, Kyuri Choi, Wanrong Zhu, Ryan Hsieh, HyeonJung Kim, Jin Hyuk Lim, Sungyong Ji, Byungju Lee, Xifeng Yan, et al. 2024. Mmsci: A multimodal multi-discipline dataset for phd-level scientific comprehension. *arXiv preprint arXiv:2407.04903*.
- Fangyu Liu, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhui Chen, Nigel Collier, and Yasemin Altun. 2023a. [DePlot: One-shot visual language reasoning by plot-to-table translation](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10381–10399, Toronto, Canada. Association for Computational Linguistics.
- Ollie Liu, Deqing Fu, Dani Yogatama, and Willie Neiswanger. 2024. [Dellma: Decision making under uncertainty with large language models](#). *Preprint*, arXiv:2402.02392.
- Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2023b. Agent-bench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*.
- Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. 2019. [Graph convolution for multimodal information extraction from visually rich documents](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 32–39, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. [ChartQA: A benchmark for question answering about charts with visual and logical reasoning](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.
- Amil Merchant, Simon Batzner, Samuel S. Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. 2023. [Scaling deep learning for materials discovery](#). *Nature*, 624(7990):80–85.
- Grégoire Mialon, Clémentine Fourier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. *arXiv preprint arXiv:2311.12983*.
- Santiago Miret and NM Krishnan. 2024. Are llms ready for real-world materials discovery? *arXiv preprint arXiv:2402.05200*.
- Hiroyuki Oka, Atsushi Yoshizawa, Hiroyuki Shindo, Yuji Matsumoto, and Masashi Ishii. 2021. Machine extraction of polymer data from tables using xml versions of scientific articles. *Science and Technology of Advanced Materials: Methods*, 1(1):12–23.
- OpenAI. 2023. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Maciej P. Polak and Dane Morgan. 2024. [Extracting accurate materials data from research papers with conversational language models and prompt engineering](#). *Nature Communications*, 15(1):1569.
- Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soriccut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Pranav Shetty, Arunkumar Chitteth Rajan, Chris Kueneth, Sonakshi Gupta, Lakshmi Prerana Panchumarti, Lauren Holm, Chao Zhang, and Rampi Ramprasad. 2023. A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing. *npj Computational Materials*, 9(1):52.

- Yu Song, Santiago Miret, Huan Zhang, and Bang Liu. 2023. [Honeybee: Progressive instruction finetuning of large language models for materials science](#). *Preprint*, arXiv:2310.08511.
- Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2021. [Kleister: Key Information Extraction Datasets Involving Long Documents with Complex Layouts](#), page 564–579. Springer International Publishing.
- Lin Sun, Kai Zhang, Qingyuan Li, and Renze Lou. 2024. [Umie: Unified multimodal information extraction with instruction tuning](#). *Preprint*, arXiv:2401.03082.
- Roselyne B Tchoua, Aswathy Ajith, Zhi Hong, Logan T Ward, Kyle Chard, Alexander Belikov, Debra J Audus, Shrayesh Patel, Juan J de Pablo, and Ian T Foster. 2019. Creating training data for scientific named entity recognition with minimal human effort. In *Computational Science—ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part I 19*, pages 398–411. Springer.
- C.G. van Ginkel and C.A. Stroo. 1992. [Simple method to prolong the closed bottle test for the determination of the inherent biodegradability](#). *Ecotoxicology and Environmental Safety*, 24(3):319–327.
- Logan Ward, Ankit Agrawal, Alok Choudhary, and Christopher Wolverton. 2016. [A general-purpose machine learning framework for predicting properties of inorganic materials](#). *npj Computational Materials*, 2(1):16028.
- Technische Universität Wien, Thomas Eiter, Thomas Eiter, Heikki Mannila, and Heikki Mannila. 1994. [Computing discrete fréchet distance](#). Technical report.
- Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. 2024. [Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments](#). *arXiv preprint arXiv:2404.07972*.
- Tong Xie, Yuwei Wan, Wei Huang, Yufei Zhou, Yixuan Liu, Qingyuan Linghu, Shaozhou Wang, Chunyu Kit, Clara Grazian, Wenjie Zhang, and Bram Hoex. 2023. [Large language models as master key: Unlocking the secrets of materials science with gpt](#). *Preprint*, arXiv:2304.02213.
- Pengcheng Xu, Xiaobo Ji, Minjie Li, and Wencong Lu. 2023. Small data machine learning in materials science. *npj Computational Materials*, 9(1):42.
- Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2022. [Layoutlmv2: Multi-modal pre-training for visually-rich document understanding](#). *Preprint*, arXiv:2012.14740.
- Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. [Layoutlm: Pre-training of text and layout for document image understanding](#). In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20*. ACM.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.
- Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 2023. [MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi](#). *arXiv preprint arXiv:2311.16502*.
- He Zhao, Yixing Wang, Anqi Lin, Bingyin Hu, Rui Yan, James McCusker, Wei Chen, Deborah L. McGuinness, Linda Schadler, and L. Catherine Brinson. 2018. [NanoMine schema: An extensible data representation for polymer nanocomposites](#). *APL Materials*, 6(11):111108.

## A Terms of Use

We used OpenAI Models, Claude, Gemini, and the NanoMine data repository in accordance with their licenses and terms of use.

## B Computational Experiments Details

**Hyperparameter Settings** The models used in our experiments, OpenAI Models, Claude, and Gemini, have been evaluated for their performance in multimodal information extraction tasks within the MATVIX benchmark while the temperature parameter is set to zero to ensure consistent evaluation.

## C JSON Formats

### Polymer Nanocomposites

```
{
  "Matrix Component": "",
  "Matrix Abbreviation": "",
  "Filler Chemical Name": "",
  "Filler Abbreviation": "",
  "Filler PST": "",
```

```

"Filler Mass": "",
"Filler Volume": "",
"Properties": [
  {
    "data": [
      ["", ""],
      ["", ""],
    ],
    "headers": [
      "x-label",
      "y-label"
    ],
    "property name": ""
  },
]
}

```

### **Polymer Biodegradation**

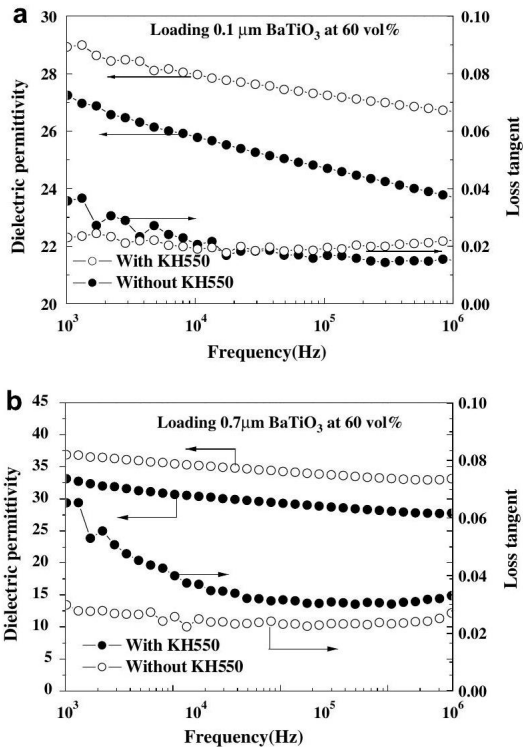
```

{
  "Polymer Type": "",
  "Substitution Type": "",
  "Degree of Substitution": "",
  "Comonomer Type": "",
  "Degree of Hydrolysis": "",
  "Molecular Weight": "",
  "Molecular Weight Unit": "",
  "Biodegradation Test Type": "",
  "Biodegradation": {
    "header": [
      "x-label",
      "y-label"
    ],
    "data": [
      [x1, y1],
      [x2, y2],
      ...
    ]
  }
}

```

## D Properties Examples

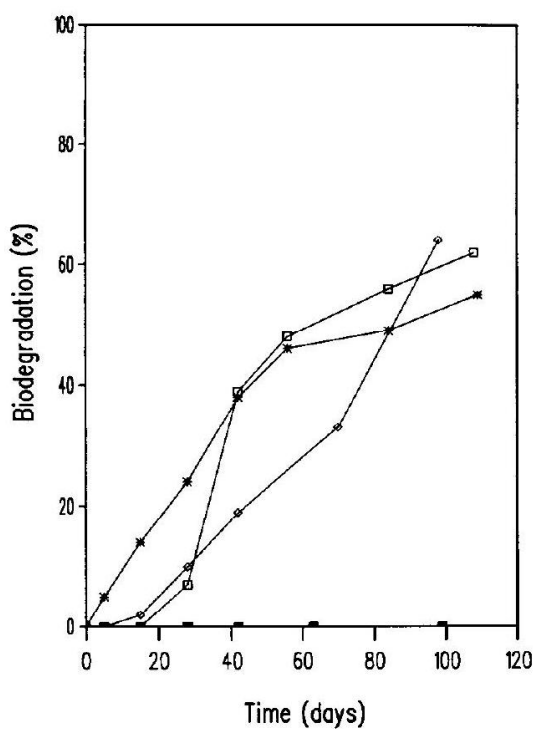
### D.1 PNC



```
{
  "Matrix Component": "DGEBA Epoxy Resin",
  "Matrix Abbreviation": "EPR",
  "Filler Chemical Name": "Barium titanate",
  "Filler Abbreviation": "BaTiO3",
  "Filler PST": "phosphated ester",
  "Filler Mass": 0.888779527559055,
  "Filler Volume": 0.6,
  "Properties": [
    {
      "data": [
        [
          "1057.89229",
          "27.25528"
        ],
        [
          "1352.11121",
          "26.98656"
        ],
        [
          "1762.75495",
          "26.8906"
        ],
        [
          "2209.0395",
          "26.5643"
        ],
        [
          "2851.93852",
          "26.52591"
        ]
      ],
      "headers": [
        "Frequency (Hz)",
        "Dielectric Permittivity"
      ],
      "property name": "Electrical"
    },
    {
      "data": [
        [
          "1009.90286",
          "0.03615"
        ],
        [
          "1304.81583",
          "0.03711"
        ],
        [
          "1685.84961",
          "0.02768"
        ]
      ],
      "headers": [
        "Frequency (Hz)",
        "Loss Tangent"
      ],
      "property name": "Electrical"
    }
  ]
}
```

Figure 2: A figure and its corresponding sample. Note how the data points in the properties are coming from the plot in the image. Also note that the data points in the JSON are shortened to fit on the page; the actual JSON is much larger. Some information in the JSON, like the full name of the filler PST, is not shown in the figure but can be found in the text. See the original article (Dang et al., 2008).

## D.2 PBD



```

{
  "Polymer Type": "carboxymethyl cellulose",
  "Substitution Type": "carboxymethyl",
  "Degree of Substitution": "0.7",
  "Comonomer Type": "null",
  "Degree of Hydrolysis": "null",
  "Molecular Weight": "null",
  "Molecular Weight Unit": "null",
  "Biodegradation Test Type": "Closed bottle",
  "Biodegradation": {
    "header": [
      "Time (days)",
      "Biodegradation (%)"
    ],
    "data": [
      [
        0,
        0
      ],
      [
        5.45454162633676,
        5.144192138244883
      ],
      [
        16.10389353031044,
        13.873732925104242
      ],
      [
        28.311688022159935,
        24.162120174857645
      ],
      [
        42.07792014773282,
        37.80202672518049
      ],
      [
        56.493502938741265,
        46.531567512039864
      ],
      [
        84.54545961220134,
        49.72720157166718
      ],
      [
        109.48053110990973,
        55.105221571325245
      ]
    ]
  }
}

```

Figure 3: A figure and its corresponding sample. Note how the data points in the properties are derived from the plot in the image. There are three types of data points in this plot; while these are not explicitly labeled in the image, the figure title specifies which samples each type corresponds to. See the original article ([van Ginkel and Stroo, 1992](#)).

## E Annotation Guidelines for PBD Papers

### Article Selection and Overview

- The dataset should be composed of high-quality research papers focused on polymer biodegradation, with a preference for those published in reputable journals.
- Make sure that the selection includes studies primarily investigating biodegradable materials. Review articles summarizing data from various studies are also acceptable. Ensure access to all selected articles through institutional or open-access resources.
- Organize each paper as a separate folder. Name each folder using a key.

### Attribute Identification

- Find attributes that are critical in defining material compositions and are consistently reported across articles.

### Annotation Process for Samples

- For papers containing multiple samples, create a separate JSON file for each sample within the folder corresponding to the paper. Ensure that each JSON file contains only one value per field (e.g., do not combine multiple DS values in a single file).
- Use the structure defined in the provided JSON format (see Appendix C).

### Composition and Property Extraction

- Carefully read the text, tables, and figures in each paper to extract information about the composition and properties of the samples. Ensure that common components like polymer type, substitution type, and test conditions are consistently annotated.
- For properties appearing as figures (e.g., biodegradation plots), use the PlotDigitizer tool:
  - Upload the plot image from the article into PlotDigitizer.
  - Calibrate the axes by marking reference points (e.g., labeled ticks or values).
  - Trace the curve to extract numerical data points as (x, y) pairs.
  - Convert these extracted values into the structured JSON format specified.

### Validation and Quality Control

- The first annotator should extract the data, while a second annotator reviews the annotations for accuracy.
- If discrepancies arise, the team should discuss and resolve them.

### Notes on Special Cases

- If images are too complex or data is not clearly labeled, extract a few representative data points to provide a manageable subset for analysis.

### Final Checks and Updates

- Review annotations periodically to confirm that all attributes align with the defined guidelines.
- If new important attributes or inconsistencies are identified, update the guidelines accordingly.

Figure 4: Annotation guidelines for identifying PBD sample compositions and properties.

## F Prompts

### F.1 Text-only Input

#### PROMPT

You extract structured data from scientific articles.

Please read the following paragraphs, find all the nano-composite samples, and fill out the given JSON template for each one of those nanocomposite samples. Do not merge samples of different compositions. If an attribute is not mentioned, fill that section with "null". Mass and Volume Composition should be followed by a %.

#### JSON Template:

```
{
  "Matrix Chemical Name": "matrix_chemical_name",
  "Matrix Chemical Abbreviation": "matrix_abbreviation",
  "Filler Chemical Name": "filler_chemical_name",
  "Filler Chemical Abbreviation": "filler_abbreviation",
  "Filler Composition Mass": "filler_mass_value",
  "Filler Composition Volume": "filler_volume_value",
  "Filler Particle Surface Treatment Chemical Name": "surface_treatment_chemical_name"
  "Properties": [
    {
      "property name": "property_name",
      "headers": ["header1", "header2"],
      "data": []
    }
  ]
}
```

Properties is a list of dictionaries where each dictionary represents a property of the nanocomposite. The property name should be filled out with the name of the property where the choices are: electrical, mechanical, viscoelastic, thermal, volumetric, rheological. The headers should be filled out with the x and y labels which are the names of the conditions or the labels of the data (e.g. time, temperature, frequency, strain, conductivity, dielectric strength, etc.). The data should be a list of (x, y) tuples. For example, if the property is 24 MPa at temperature 25°C and 30 MPa at temperature 50°C, the data should be [(25, 24), (50, 30)]. If no data is mentioned, please fill it with null.

#### Article:

```
\documentclass[10pt]{article}
\graphicspath{ {./images/} }

\title{Study on microstructure and dielectric property of the  $\text{BaTiO}_3$  / epoxy resin
composites }

\author{Zhi-Min Dang {a,*}, Yan-Fei {b}, Hai-Ping Xu {c},
{a} Jinbo Bai {b}
{a} Key Laboratory of Beijing City on Preparation and Processing of Novel Polymer Materials
and Key Laboratory of the Ministry of
Education on Nanomaterials, Beijing University of Chemical Technology, Beijing 100029, PR China\
{b} Lab. MSS-MAT, CNRS UMR 8579, Ecole Centrale de Paris, Grande Voie des Vignes,
Châtenay Malabry Cedex 92295, France}
\date{}

\begin{document}
\maketitle
Received 26 December 2006; received in revised form 23 April 2007; accepted 8 May 2007

Available online 18 May 2007

\begin{abstract}
High dielectric permittivity barium titanate/epoxy resin  $\left(\text{BaTiO}_3 / \text{EPR}\right)$ 
composites with different size  $\text{BaTiO}_3$  particles were prepared and their
dielectric properties were studied via a wide range of temperature and frequency. The results show
that an appropriate silane coupling agent can be used in order to improve the interaction between  $\text{BaTiO}_3$ 
and  $\text{EPR}$ , and subsequently induces a high dielectric permittivity and
a low loss tangent...
\end{abstract}
```

Figure 5: Sample prompt to GPT-4o for extracting nanocomposite samples. The provided article has been truncated due to space constraints. The input is given without including the figures parsed by DePlot and without providing any images—only textual input was given to the model.



## F.2 LLM Output

### GPT-4o RESPONSE

Based on the document, one nanocomposite sample was identified:

```
{
  "Matrix Chemical Name": "Epoxy Resin",
  "Matrix Chemical Abbreviation": "EPR",
  "Filler Chemical Name": "Barium Titanate",
  "Filler Chemical Abbreviation": "BaTiO3",
  "Filler Composition Mass": "null",
  "Filler Composition Volume": "null",
  "Filler Particle Surface Treatment Chemical Name": "Silane Coupling Agent KH550",
  "Properties": [
    {
      "property name": "thermal",
      "headers": [
        "Temperature [\u00b0C]",
        "Dielectric Permittivity"
      ],
      "data": null
    },
    {
      "property name": "volumetric",
      "headers": [
        "Volume Fraction [%]",
        "Dielectric Permittivity"
      ],
      "data": null
    },
    {
      "property name": "mechanical",
      "headers": [
        "Temperature [\u00b0C]",
        "Glass Transition Temperature [Tg]"
      ],
      "data": [[0, 69.4], [30, 53], [40, 65.4], [60, 57.4], [70, 35.3]]
    }
  ]
}
```

Note: The document provides detailed thermal and mechanical properties, but specific mass and volume compositions are not mentioned, hence "null" in the corresponding fields.

Figure 6: GPT-4o response to the nanocomposite data extraction task described in Figure 5.

### F.3 Text+Image Input

#### PROMPT

Given the image and the following polymer nanocomposite sample compositions, first identify which sample composition is present in the image, then extract its properties.

#### Sample Compositions:

```
{
  "Matrix Chemical Name": "Epoxy Resin",
  "Matrix Chemical Abbreviation": "EPR",
  "Filler Chemical Name": "Barium Titanate",
  "Filler Chemical Abbreviation": "BaTiO3",
  "Filler Composition Mass": "null",
  "Filler Composition Volume": "null",
  "Filler Particle Surface Treatment Chemical Name": "Silane Coupling Agent KH550",
  "Properties": [
    {
      "property name": "thermal",
      "headers": [
        "Temperature [\u00b0C]",
        "Dielectric Permittivity"
      ],
      "data": null
    },
    {
      "property name": "volumetric",
      "headers": [
        "Volume Fraction [%]",
        "Dielectric Permittivity"
      ],
      "data": null
    },
    {
      "property name": "mechanical",
      "headers": [
        "Temperature [\u00b0C]",
        "Glass Transition Temperature [Tg]"
      ],
      "data": [[0, 69.4], [30, 53], [40, 65.4], [60, 57.4], [70, 35.3]]
    }
  ]
}
```

For those sample compositions that are present in the image, extract the information about the property.

Properties is a list of dictionaries where each dictionary represents a property of the nanocomposite. The property name should be filled out with the name of the property where the choices are: electrical, mechanical, viscoelastic, thermal, volumetric, rheological. The headers should be filled out with the x and y labels which are the names of the conditions or the labels of the data (e.g. time, temperature, frequency, strain, conductivity, dielectric strength, etc.). The data should be a list of (x, y) tuples. For example, if the property is 24 MPa at temperature 25°C and 30 MPa at temperature 50°C, the data should be [(25, 24), (50, 30)]. If no data is mentioned, please fill it with null.

Expand the sample composition JSONs to include the property information and return all the expanded JSONs.

Figure 7: Sample prompt to GPT-4o for identifying polymer nanocomposite sample compositions from an image and extracting their properties.

## F.4 VLM Output

### GPT-4o RESPONSE

The image shows dielectric permittivity and loss tangent as a function of temperature for two different loadings of BaTiO<sub>3</sub> in an epoxy resin matrix. However, the sample composition provided does not specify the particle size or volume fraction, so we cannot directly match it to the image.

Given the sample composition provided, we can expand the JSON to include the property information as follows:

```
[
  {
    "Matrix Chemical Name": "Epoxy Resin",
    "Matrix Chemical Abbreviation": "EPR",
    "Filler Chemical Name": "Barium Titanate",
    "Filler Chemical Abbreviation": "BaTiO3",
    "Filler Composition Mass": "null",
    "Filler Composition Volume": "null",
    "Filler Particle Surface Treatment Chemical Name": "Silane Coupling Agent KH550",
    "Properties": [
      {
        "property name": "thermal",
        "headers": ["Temperature [°C]", "Dielectric Permittivity"],
        "data": null
      },
      {
        "property name": "volumetric",
        "headers": ["Volume Fraction [%]", "Dielectric Permittivity"],
        "data": null
      },
      {
        "property name": "mechanical",
        "headers": ["Temperature [°C]", "Glass Transition Temperature [Tg]"],
        "data": [
          [0, 69.4],
          [30, 53],
          [40, 65.4],
          [60, 57.4],
          [70, 35.3]
        ]
      }
    ]
  }
]
```

Figure 8: GPT-4o response to the nanocomposite data extraction task described in Figure 7, given one of the images from the paper. Note that if there are  $n$  images in the document, there will be  $n$  separate responses for each image, which will later be merged together.

## G Human Evaluation Guideline

### Instructions

For each plot pair, please provide a score from 1 (poor) to 5 (excellent) for the following two aspects:

- Axis Label Accuracy:
  - Check if the labels on the x-axis and y-axis of the predicted plot match those of the ground-truth plot.
  - 1: Labels are entirely incorrect or missing.
  - 2: Labels are mostly incorrect, with one or two minor matches.
  - 3: Labels are partially correct (e.g., one axis matches, the other is incorrect).
  - 4: Labels are mostly correct, with only minor errors (e.g., small formatting differences).
  - 5: Labels are completely correct and match perfectly with the ground-truth plot.
- Curve Trend Consistency:
  - 1: The trend is completely different from the ground truth.
  - 2: The trend shows some alignment but is mostly inconsistent.
  - 3: The trend matches in some sections.
  - 4: The trend is mostly consistent, with only minor deviations.
  - 5: The trend is fully consistent and matches the ground truth perfectly.

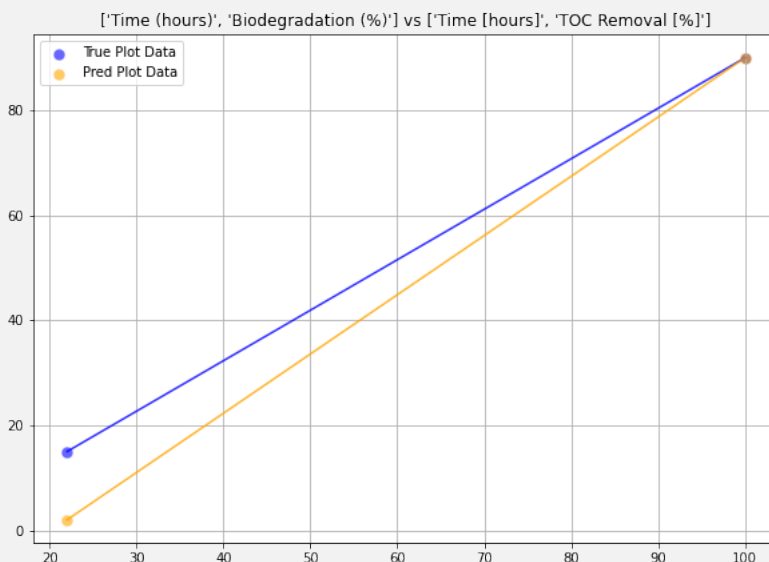


Figure 9: Scoring guidelines for human evaluation.