ChemPile: A 250 GB Diverse and Curated Dataset for Chemical Foundation Models

Adrian Mirza ✉
HIPOLE Jena & HZB

Nawaf Alampara
FSU Jena

Martiño Ríos-García
FSU Jena

Mohamed Abdelalim
Independent researcher

Jack Butler
Faculty

Bethany Connolly
Faculty

Tunca Dogan
Hacettepe University

Marianna Nezhurina
JSC, LAION

Bünyamin Şen
Hacettepe University

Santosh Tirunagari
EMBL-EBI

Mark Worrall
Faculty

Adamo Young
University of Toronto

Philippe Schwaller
LIAC and NCCR Catalysis

Michael Pieler ✉
Independent researcher

Kevin Maik Jablonka ✉
HIPOLE Jena, FSU Jena, CEEC Jena, JCSM Jena

✉ mail@adrianmirza.com, michael.pieler@gmail.com, mail@kjablonka.com.

Full affiliations are listed in the appendix.

Abstract

Foundation models have shown remarkable success across scientific domains, yet their impact in chemistry remains limited due to the absence of diverse, large-scale, high-quality datasets that reflect the field's multifaceted nature. We present the ChemPile, an open dataset containing over 77 billion tokens of curated chemical data, specifically built for training and evaluating general-purpose models in the chemical sciences. The dataset mirrors the human learning journey through chemistry—from educational foundations to specialized expertise—spanning multiple modalities and content types including structured data in diverse chemical representations (SMILES, SELFIES, IUPAC names, InChI, molecular renderings), scientific and educational text, executable code, and chemical images. ChemPile integrates foundational knowledge (textbooks, lecture notes), specialized expertise (scientific articles and language-interfaced data), visual understanding (molecular structures, diagrams), and advanced reasoning (problem-solving traces and code)—mirroring how human chemists develop expertise through diverse learning materials and experiences. Constructed through hundreds of hours of expert curation, the ChemPile captures both foundational concepts and domain-specific complexity. We provide standardized training, validation, and test splits, enabling robust benchmarking. ChemPile is openly released via HuggingFace with a consistent API, permissive license, and detailed documentation. We hope the ChemPile will serve as a catalyst for chemical AI, enabling the development of the next generation of chemical foundation models.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks.

https://doi.org/10.52202/085713-1861

1 Introduction

Foundation models are transforming science, with particularly promising applications in the chemical sciences [1, 2, 3]. Progress in this field could fundamentally advance drug discovery, accelerate materials development for energy transition, and provide new solutions for climate change mitigation [4]. The potential societal impact is immense. Recent developments demonstrate that large language models (LLMs) can already answer chemical queries [5, 6, 7], predict molecular properties [8, 9, 10, 11, 12] or crystal structures [13, 14, 15], and direct experiments [16, 17, 18, 19]. Yet, their performance is often brittle, with shallow reasoning and poor generalization beyond narrow domains [20, 21, 22]. These limitations likely stem not (only) from architectural constraints, but from the data on which these models are trained. Current chemical datasets are fragmented and narrowly focused. Most are confined to a single modality—such as SMILES strings [23]—and few capture the underlying reasoning or contextual knowledge that defines chemical understanding. Moreover, they are seldom curated with machine learning in mind, leading to issues with inconsistency, data leakage, and poor coverage of fundamental principles [24]. As a result, existing foundation models in chemistry struggle to learn generalizable patterns, reason across domains, or provide interpretable outputs.

To address this, we introduce ChemPile, a large, multimodal open dataset designed to support the training and evaluation of foundation models in chemistry. The ChemPile is the result of an extensive, community-driven effort, involving hundreds of hours of expert curation, cleaning, and annotation. It provides a unified interface for diverse, multimodal data and is built to serve as a foundational resource for chemical foundation models. The ChemPile mirrors the journey of chemical expertise development in humans—from foundational concepts to specialized knowledge to advanced reasoning—through its diverse content types collected in different subsets:

- ChemPile-**Education:** Captures foundational core knowledge through curated educational content—similar to how students build conceptual understanding through textbooks and lectures.
- ChemPile-**Paper:** Incorporates curated scientific literature filtered for chemical content—allowing the models to learn from the frontiers of science.
- ChemPile-**(m)LIFT:** Provides structured factual knowledge through language-interfaced [25] tabular datasets with chemical information in multiple representations (IUPAC, SELFIES, InChI, images) — allowing the model to learn nuanced structure-property-function relationships.
- ChemPile-**Reasoning:** Compiles explicit reasoning traces for chemical problems — to allow models to learn reasoning which is needed to solve advanced chemical problems.
- ChemPile-**Code:** Includes chemical code —reflecting that code has often been shown to increase model capabilities [26, 27].
- ChemPile-**Caption:** Compiles pairs of chemical images with the corresponding descriptive text—reflecting that chemical information is typically multimodal and requires joint reasoning over different modalities such as images or text.
- ChemPile-**Instruction:** Includes multi-turn chemistry conversations —since instruction data is crucial in the training of LLMs [28].

Just as human chemists learn through diverse materials and experiences—textbooks for foundations, laboratory work for hands-on skills, research papers for specialized knowledge, and problem-solving for developing reasoning—ChemPile's varied content types aim to provide a comprehensive learning environment for chemical AI.

The core features of the ChemPile are:

- **Scale:** To our knowledge, ChemPile is the largest open curated chemical corpus, providing sufficient data volume for foundation model training and scaling studies.
- **Expert curation:** The ChemPile has been rigorously cleaned, annotated, and reviewed by domain experts through an extensive collaborative effort.

https://doi.org/10.52202/085713-1861

- **Content type diversity:** The ChemPile combines different kinds of content on a spectrum from conceptual understanding (ChemPile-Education), over detailed knowledge (ChemPile-(M)LIFT), to advanced multimodal reasoning (ChemPile-Reasoning, ChemPile-Code, ChemPile-Caption), covering materials that mirror the human chemist's educational journey.
- **Chemical diversity:** The ChemPile spans the entire spectrum from biochemistry to materials science, enabling research on domain adaptation and knowledge transfer across chemical subfields that were previously siloed.
- **Multimodality:** The ChemPile integrates images with captions, molecular and crystal representations in various formats, chemical drawings, and other visual elements essential to chemical communication, providing a foundation for multimodal models.
- **Ease of use:** The ChemPile is hosted on HuggingFace under a consistent API for public access with a permissive license. We provide recommended training/validation/test splits based on analysis of chemical compounds and extensive documentation (chempile.lamalab.org) to facilitate immediate research use.

By centralizing high-quality chemical data in a machine learning-ready format that reflects the multifaceted nature of chemical expertise, we hope that the ChemPile will catalyze innovation at the intersection of AI and chemistry. The ChemPile aims to not just be a dataset, but a bridge between disciplines that will enable a new generation of researchers to contribute to chemical AI and accelerate scientific discovery.

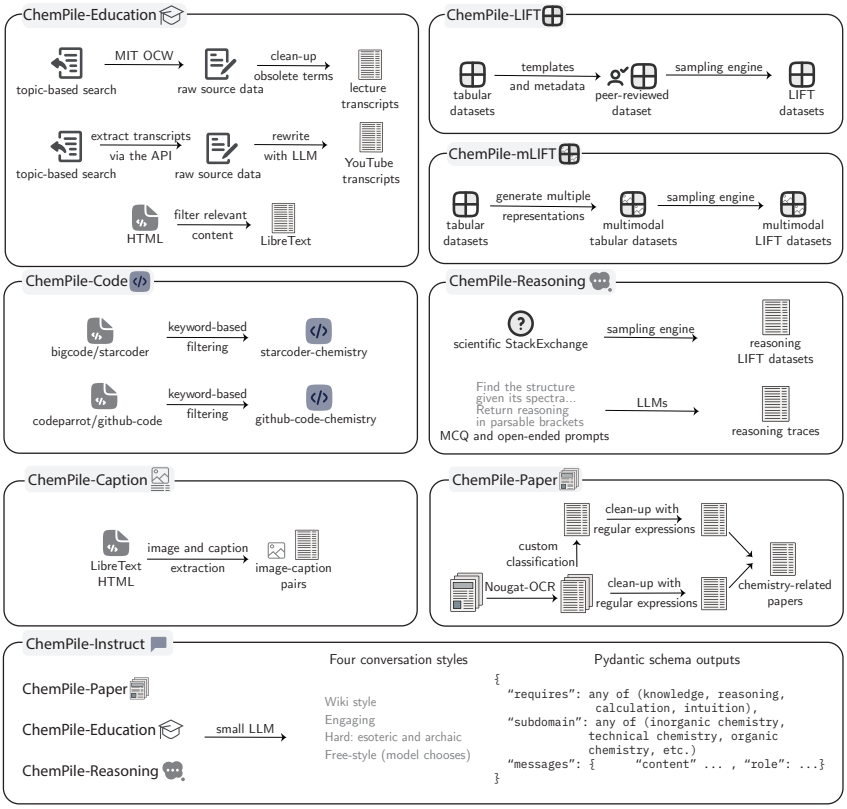


Figure 1: **Overview of the ChemPile and its curation process.** The figure illustrates the dataset creation process. Education and Caption consist of gathering resources from online resources. Code and (m)LIFT are based on dataset content, for the first filtering from general datasets, while for the second, by filling templates with the data. For ChemPile-Paper, the content is collected by filtering and processing published open-source papers. ChemPile-Reasoning is based on distilling knowledge from LLMs and processing data from Stack Exchange. Finally, ChemPile-Instruct has been compiled using LLM rephrasing on three other subsets of ChemPile. The resulting datasets are published in a format that is very easy to use on HuggingFace.

https://doi.org/10.52202/085713-1861

2 Related work

The ChemPile is the first dataset that combines diverse content types and chemical subdisciplines under a consistent interface, addressing several key limitations in existing resources. To achieve this, ChemPile builds on a foundation of prior work.

2.1 Datasets for training of foundation models

Web-scale data has become the standard approach for pre-training foundation models [29, 30], with empirical scaling laws suggesting that performance improves with dataset size [31, 32, 31]. However, recent studies challenge the "more data is always better" paradigm, exploring data-effective learning approaches that focus on quality and representativeness rather than sheer volume [33, 34, 35, 36]. Current state-of-the-art training pipelines use carefully constructed mixtures of different data types including natural text, code, textbooks, and reasoning traces to improve model capabilities [37, 27, 38]. While this trend toward high-quality, diverse data mixtures has transformed general-purpose AI, the chemical domain has not yet benefited from similar approaches.

For multimodal foundation models, image-text pairs represent the primary training data format, typically sourced from web pages containing images with associated alt-text, captions, or surrounding text [39]. However, equivalent multimodal resources have been largely absent in the chemical domain until the ChemPile.

2.2 Chemical datasets

Traditional chemistry datasets have largely relied on tabular formats as compiled in MoleculeNet [40] or Therapeutic Data Commons [41]. Resources like PubChem [42] and UniProt [43] provide large collections of molecular structures or protein sequences for tasks in (bio)molecular property prediction. A critical limitation is that these resources cannot be directly used for training LLMs as they require conversion into natural language through templates that demand significant domain knowledge to set up properly [44].

While experimentally derived datasets are typically small or medium-sized, larger resources such as QM9 [45] have been compiled based on computational screenings. However, these datasets may not fully capture real-world variations and experimental noise. Other resources such as the USPTO database [46] are, for example, patent-derived and come with corresponding biases [47, 48, 49].

A fundamental challenge remains the integration of information from different sources, chemical subfields, and modalities. Scientific information is frequently distributed across multiple datasets, making it difficult to assemble comprehensive resources that reflect the true complexity of chemical phenomena [50]. The ChemPile explicitly addresses this fragmentation by unifying diverse chemical information under a consistent interface.

In addition, it is important to realize that molecules can be represented in various string formats, including IUPAC names, SMILES, DeepSMILES, SELFIES, and InChI [51]. Currently, there is no consensus on which representation is optimal for training chemical foundation models. To enable the systematic comparison of their effectiveness, the ChemPile includes multiple representations for the same molecules.

2.3 Chemical text and multimodal datasets

Recent efforts to create specialized chemical datasets include the Mol-instructions dataset [52], which provides around 2 million biomolecular and protein-related instructions. In the multimodal space, several specialized resources such as MoMu [53], PubChemSTM [54], Llamole [55], and MultiMat [56] have emerged.

While these specialized datasets represent important advances, they remain limited in scope and typically focus on a single chemical subdomain or modality pairing.

The ChemPile builds upon these efforts by providing a unified resource that spans multiple chemical subfields and integrates all relevant modalities under a consistent framework, addressing the fragmentation, narrow focus, modality restrictions, and inconsistent formats of existing chemical datasets.

3 Overview of the ChemPile

The ChemPile is distinct in scale, breadth, curation quality, and ease-of-use.

Scale One of the most essential characteristics of a dataset for training foundation models is its scale [32]. For reference, we compare ChemPile to other domain-specific foundation models (Figure 2). `ChemDFM` [57] is the largest chemical foundation model that has been reported. It has been trained on a dataset of 34B tokens which, however, has not been released. Even though it contains general-purpose data (such as Wikipedia and the WuDao Corpora [58]), it is still more than 50% smaller than the ChemPile. Other notable chemistry datasets, such as LlaSMol [59] and ChemDual [60], are orders of magnitude smaller.

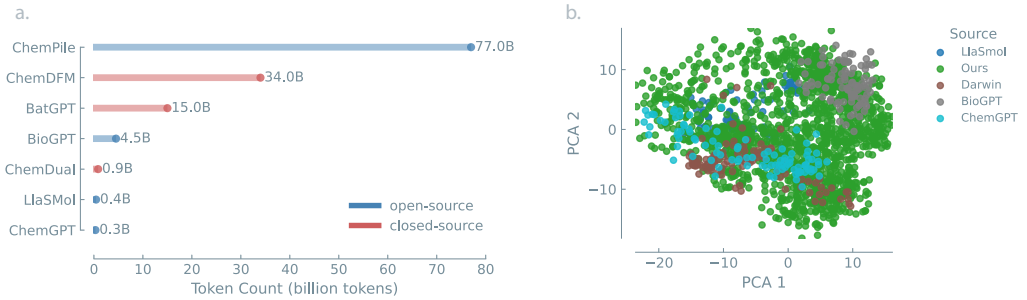


Figure 2: **(a): Token count comparison between the ChemPile dataset and other domain-specific large datasets used to train foundation models.** `ChemDFM` [57] is a foundation model for chemistry trained on 34B tokens in chemistry-related papers and textbooks augmented with general text (49M tokens), BatGPT [61] is a foundation model for chemical engineering, BioGPT [62] for biology, and ChemGPT [63] is a foundation model trained only on SMILES string. LlaSMol [59] is an instruction-tuning dataset for chemistry. ChemDual [60] is a 4.4 million instruction dataset for chemical reactions. The value for BioGPT is an estimate based on: 15 M abstracts × 250 words × 1.2 tokens ≈ 4.5 B tokens. We compute the estimate for LlaSMol based on the published HuggingFace dataset. The scale of our dataset exceeds any of the corpora used to pre-train or fine-tune existing chemistry LLMs. **(b): Embedded datapoints sampled from various subsets of ChemPile vs other public datasets**. Note, only the instruction tuning data made public by the authors of Darwin [64] is used. We embed only the first 512 tokens of each sampled document using the `specter2-base` model provided by Singh et al. [65]. Along PCA, we provide UMAP and TSNE plots in Appendix Q.

As Figure 2a) illustrates, ChemPile is the largest open chemical dataset we are aware of and the only one that reaches a scale that is meaningful for training foundation models.

Diversity The ChemPile is not only large but also diverse. Data mixing for training LLMs is still not fully understood and is an active field of research. Different mixes typically yield different generalization performance [37, 66, 67]. To enable such research, the ChemPile was designed to be maximally diverse. Figure 2b) illustrates this. In this figure, we showcase that the embeddings of data from the ChemPile span a larger space than data from many other chemical datasets combined.

We achieve this in multiple ways: First, sampling and curating data from very different sources and, second, by representing chemical entities in various modalities and text forms.

In contrast to other large chemical datasets, ChemPile is a systematic collection of multiple subsets that were curated to encompass specific knowledge or to potentially convey specific abilities to models trained on those subsets. These subsets, which we describe in detail in Section 4, contain data sampled from very different sources such as structured chemical datasets, recordings of lectures, or data we created from scratch.

In addition, ChemPile considers the fact that chemical entities, such as molecules, can be represented in diverse forms. This includes diverse string representations, such as IUPAC names, SELFIES [68], SMILES [23], and InChI [69], but also molecular drawings [70] in addition to images from chemical textbooks.

https://doi.org/10.52202/085713-1861

This feature of the ChemPile is relevant because while SMILES are widely used in cheminformatics, it is not obvious that they are also the best choice for building foundation models. First, SMILES and other chemical representations are not handled in optimally in conventional pretrained tokenizers [71, 72]. This is particularly interesting for finetuning and continued pretraining studies and can be seen in a correlation analysis (see Figure 3 for details). If different chemical representations are embedded with existing embedding models, similarity between embeddings of IUPAC names correlates much strongly with established similarity measures—such as the Tanimoto similarity between molecular fingerprints [73]—than the embeddings of other molecular representations.

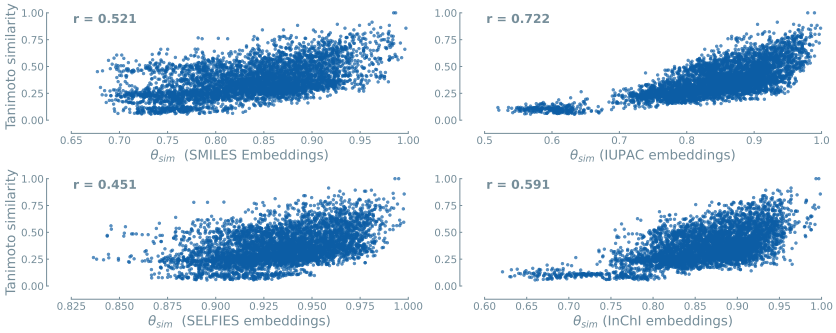


Figure 3: **Correlation between the Tanimoto similarity and the cosine similarity (θ_{sim}) of the embeddings for most common chemical representations**. The correlation is shown for four representation embeddings: SMILES (top left), IUPAC name (top right), SELFIES (bottom left), and InChI (bottom right). For the four subplots, we show the Pearson correlation **r** in the top left corner of all subplots.

In addition, one might expect benefits from the inclusion of IUPAC names as they are not only closer to common English text but, in particular, the text seen in chemical papers and hence might improve training dynamics.

Quality Curation quality distinguishes ChemPile from previous chemical datasets. Domain specialists manually reviewed each subset, ensuring scientific accuracy and relevance. For ChemPile-(m)LIFT, we implemented a systematic verification protocol where chemical experts checked template design, property assignments, and molecular representations. All datasets underwent multiple validation passes to eliminate inconsistencies, incorrect nomenclature, and formatting errors. This curation process, representing hundreds of expert hours, delivers a dataset that captures both foundational concepts and specialized knowledge with high fidelity.

Ease of use The ChemPile prioritizes accessibility for researchers from different domains through consistent interfaces across all datasets. We host the entire data collection on HuggingFace with uniform formatting and comprehensive documentation (chempile.lamalab.org). Each subset includes detailed metadata, usage examples, and explicit training/validation/test splits designed to prevent chemical structure leakage between partitions. The modular architecture allows researchers to use specific subsets independently (Appendix S.1), or combine them as needed (Appendix S.2). This accessibility reduces barriers to entry for researchers from both machine learning and chemistry backgrounds, enabling immediate application to foundation model training, specialized fine-tuning, or directed research on particular chemical domains.

3.1 Modeling with ChemPile

To demonstrate the quality of our curated data, we sample up to 100M tokens from our datasets (more detailed description in Appendix K. We then do LoRA finetuning using different subsets of ChemPile. The evaluation of the results is done on ChemBench[5].

To best utilize our data, we explored two approaches for combining the different LoRA adapters (each trained on a subset of ChemPile). The first approach is to treat the adapters independently and select the generation with the lowest perplexity, while the second implies the direct linear merging of the adapters (described in ref. [74]). As illustrated in Figure 4, our approaches outperform the

base model, `Qwen-2.5-7B-Instruct` by 9 %, which demonstrates on one hand the quality of our data, and on the other the effectiveness of the chosen approach, which are both novel for chemistry foundation models. Both of these approaches represent the state-of-the-art for 7 B parameter models.

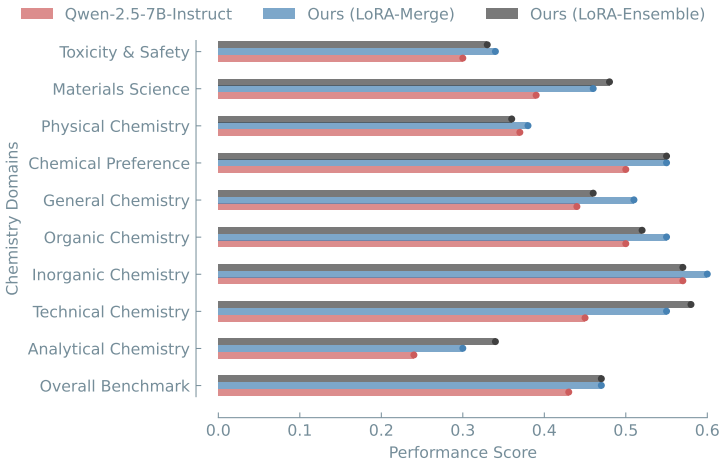


Figure 4: **Performance of our models compared to the base Qwen-2.5-7B-Instruct model on ChemBench.** We show the results on the overall benchmark, but also at individual chemistry fields, to assess whether the performance boost is consistent.

4 Diving into the ChemPile

The ChemPile can be used for many tasks, but the focus is on the training of general-purpose foundation models for the chemical sciences. This section provides detailed information on the seven datasets, detailed in Table 1, making up the ChemPile data and their curation process.

Table 1: **Token count and size in GB of the datasets**. The number of tokens was estimated using `tiktoken` [75] with the model `gpt2`. The dataset size indicated in this table corresponds to the compressed file size following conversion to the Parquet format. Note that the number of figures for the multimodal datasets and the number of images is equal to the number of documents—one image per entry.

Dataset	Size (GB)	Number of text tokens	Number of documents	HuggingFace dataset
ChemPile-Education	0,25	130M	66,9K	🤗
ChemPile-Paper	31,6	14,1B	11,7M	🤗
ChemPile-LIFT	49.1	29,4B	185M	🤗
ChemPile-mLIFT	155	15,0B	61.6M	🤗
ChemPile-Code	15,6	18,0B	2,27M	🤗
ChemPile-Reasoning	0,10	20,0M	72,9K	🤗
ChemPile-Caption	3,23	10,3M	100K	🤗
ChemPile-Instruction	2,85	396M	410K	🤗
ChemPile	257	77,0B	260M	🤗

4.1 ChemPile-Education

ChemPile-Education contains (foundational) knowledge exposition from lectures and textbooks as well as worked practice problems (see Figure 6).

The data collection involved four distinct methodologies tailored to source-specific characteristics. **LibreTexts Chemistry** contains open-source chemistry textbooks, which we mined using a pipeline that systematically parses HTML documents, stripping non-content elements to compile a chemically

https://doi.org/10.52202/085713-1861

focused corpus of 114 million tokens. **MIT OpenCourseWare** lecture materials were programmatically retrieved through topic-specific searches (biology, chemistry, chemical engineering, physics), with course names and download links preserved. **YouTube course transcripts** were sourced via LLM-generated keyword queries, restricted to Creative Commons-licensed videos, and refined using `GPT-4.1` to correct transcription errors and enhance coherence. **US Olympiad problems** from 2003 to 2024 were manually processed using the `Gemini 2.0 Flash` model, which aligned PDF-based questions and solutions into JSON-structured metadata and selected solutions exceeding 250 characters.

A detailed explanation of the workflow for each of the sources can be found in Appendix P.1

4.2 ChemPile-Paper

As a resource for cutting-edge applications of chemical knowledge and reasoning, we also curated a dataset of papers from diverse repositories in ChemPile-Paper. The **EuroPMC** dataset [76], comprising 27 million abstracts and 5 million full-text articles, was filtered using a BERT-based multilabel classifier trained on the CAMEL datasets (20,000 examples per discipline) [77] and validated against FineWebMath [78] annotations (F_1-score ≈ 0.77 on 150 entries we manually annotated). Chemistry-related content was identified by analyzing the first five 512-token chunks per document with 50-token overlaps, yielding 3.3 billion tokens. Preprints from **ChemRxiv**, **BioRxiv**, and **MedRxiv** were collected via PaperScraper [79], processed with Nougat [80] for text extraction, and enriched with metadata (license, publication date, authors, title). **ArXiv** submissions were filtered by materials science and physical chemistry keywords (e.g., `cond-mat.mtrl-sci`), with PDFs retrieved via PaperScraper. For all scientific articles, we employed a postprocessing pipeline that removed text that is not directly linked to chemical information (e.g., authors, acknowledgments, page numbers) as explained in Appendix P.3.2. Additionally, we included **Materials Safety Data Sheets (MSDS)** as structured tabular data (H/P statements) and natural text, ensuring comprehensive coverage of safety information. This multi-source approach balances breadth and domain specificity across literature, preprints, and regulatory documents. A more concise explanation about the sources and in the data-curation procedure is in Appendix P.3.

4.3 ChemPile-(m)LIFT

In ChemPile-(m)LIFT, we compile language-interfaced tabular data about properties of molecules, materials, and reactions to allow models to learn intricate structure-property-function relationships.

Curation process We manually collected and annotated structured chemical datasets. In the annotation process, domain experts not only annotated the meaning (in many cases including links to ontologies) and possible namings of columns but also created multiple templates that use the tabular data in different language-interfaced settings such as (multiobjective) property prediction or inverse design. The entire curation process was organized via Pull Requests on GitHub which were reviewed by at least one other domain expert. We provide examples of some of those templates in Table 12 (a total of 1636 templates have been manually curated). All curation scripts and metadata files are available on GitHub.

Sampling engine The language-interfaced tabular data has been generated with a sampling engine that includes several functionalities: flexible multiple-choice question generation (including permutation of enumeration symbols), synonym sampling (e.g., diverse sampling of property names or molecular representations), as well as conditional formatting (e.g., for negations). An illustrated example and a more detailed explanation can be found in Appendix N. In the sampled datasets, we distinguish between completion and instruction type templates and allow the user to select data formatted in specific templates to allow systematic ablation studies [44].

ChemPile-mLIFT Since our annotation process clearly identified columns containing molecular, material, or reaction information, we could systematically compute alternative representations such as SMILES, InChI, SELFIES, IUPAC names, and images for all entries in ChemPile-LIFT using cheminformatics tools. In particular, images were created in various styles using a pipeline based on RanDepict [70]. The details on the SMILES-to-IUPAC conversion are described in Appendix R.

4.4 ChemPile-Code

Programming is a crucial part of chemistry research, for example, as part of data analysis or computational chemistry. Hence, it is important to cover chemistry-related code knowledge during training. Moreover, it has been shown that including code datasets during pretraining can improve reasoning [26, 81].

To create the ChemPile-Code subset, we filter some of the biggest and widely used datasets. We use regular-expression-based filters to relevant code snippets pertaining to chemistry, materials science, and biology, as well as specific scientific software packages. The majority of the code after filtering is related to simulations, see Figure 7 (also see the keywords used for filtering Table 13).

The collection primarily comprises a filtered version of the StarCoder and CodeParrot datasets. **StarCoder** [82] is a filtered version of the Stack dataset [83]. **Codeparrot** is a subset of GitHub-code. The datasets were deduplicated based on exact hash matching. Furthermore, entries from CodeParrot were removed from Starcoder again based on hash string matching to avoid any overlaps between the two datasets.

4.5 ChemPile-Reasoning

As training on worked examples and reasoning chains is known to improve the performance of foundation models, we specifically created such datasets.

ChemPile-Reasoning combines data from two primary sources. For the first sources, we gathered and filtered content from the **Chemistry**, **Matter Modeling**, and **Physics Stack Exchange** forums. The collected data was processed using templates incorporating questions and answers in distinct templates and linguistic styles to enhance diversity. This approach yielded datasets of 12 million, 7 million, and 1.7 million tokens for physics, chemistry, and matter modeling, respectively.

The second source involves **synthetic reasoning traces** generated by the `Claude-3.5-Sonnet` and `Deepseek-R1` models [84]. These models were prompted to perform spectral elucidation tasks, analyzing molecular spectra to identify corresponding molecules. Over 2 million tokens of distilled synthetic reasoning data were collected through this process. We provide additional methodological details, including data parsing and curation steps, in Appendix P.4.

4.6 ChemPile-Caption

The ChemPile-Caption dataset contains over 100,000 text-image pairs focused on foundational chemistry concepts. We sourced images and their corresponding captions and alt texts from **LibreTexts Chemistry** using HTML parsing. To ensure data quality, we excluded images lacking descriptive text or with fewer than 200 combined characters in their captions and alt texts. This curation process resulted in a high-quality multimodal dataset, as LibreTexts Chemistry content originates from peer-reviewed college courses and textbooks, ensuring reliability and academic relevance.

4.7 ChemPile-Instruct

The Chempile-Instruction dataset comprises over 395 million tokens organized into multi-turn conversations averaging eight turns per conversation. This resource was generated by rephrasing the ChemPile-Reasoning, ChemPile-Education, and a 100-million-token subset of ChemPile-Paper using the `gpt-4o-mini-2024-07-18` model. Mirroring the methodology of Pieler et al. [85], we instructed the model to employ diverse stylistic approaches during rephrasing, including Wikipedia, hard/technical, engaging, and unstyled variants. ChemPile-Instruction consequently represents the largest available multi-turn chemistry instruction-tuning dataset. Further information about the dataset creation process is detailed in Appendix P.5.

4.8 Splits

Depending on the representation of molecules and macromolecular structures (e.g., proteins or polymers), we split the datasets differently. All SMILES across the various tabular datasets are combined into a single list. Then, we apply scaffold splitting (based on the RDKit Murcko scaffolds) [40, 86]. At the same time, we ensure that for all datasets, the validation and test sets are not empty. This

 https://doi.org/10.52202/085713-1861

ensures the usability of individual language-interfaced tabular datasets for other downstream tasks, such as fine-tuning.

For amino-acid sequences (i.e., proteins), we follow the same procedure for deduplication, but apply random splitting on all sequences across datasets. For datasets without SMILES or amino-acid sequences, we apply random splitting for individual datasets. More details on the splitting procedure are shown in Appendix M.

5 Future work

ChemPile establishes the essential foundation for the next generation of chemical AI, creating a pathway for numerous exciting developments now within reach. The infrastructure we've created enables seamless integration of organometallic chemistry datasets, which are currently underrepresented, as specialized representations evolve [68]. Our multimodal datasets provide the perfect scaffold for incorporating spectroscopic data, reaction dynamics visualizations, and materials-specific representations. The robust splitting methodology in ChemPile-(m)LIFT and the ChemPile-Paper dataset opens the door to sophisticated chemical entity recognition across papers, a capability that will further enhance model performance through improved deduplication and knowledge integration. Our extensible sampling engine can be extended to support data generation for foundation model architectures beyond language models, including GNNs and contrastive models, broadening ChemPile's utility.

6 Conclusions

The chemical sciences stand at the forefront of AI's potential societal impact, with applications ranging from drug discovery to climate change mitigation. Until now, progress — for example, in the development of chemical foundation models — has been constrained by the absence of data resources that reflect chemistry's multifaceted nature. ChemPile transforms this landscape by providing the first dataset with meaningful scale and diversity for chemistry. By mirroring the human learning journey—from educational foundations to specialized knowledge to multimodal understanding—ChemPile creates a comprehensive learning ecosystem for chemical AI. ChemPile serves as a bridge between disciplines that will enable a new generation of researchers to contribute to chemical AI and accelerate scientific discovery.

7 Acknowledgments

This work was supported by the Carl Zeiss Foundation and by Intel and Merck via the AWASES programme.

Parts of A.M.'s work were supported by the Helmholtz Association within the framework of the Helmholtz Foundation Model Initiative (project SOL-AI).

K.M.J. is part of the NFDI consortium FAIRmat funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project 460197019. K.M.J. also acknowledges support from OpenPhilanthropy.

P.S. acknowledges support from the NCCR Catalysis (grant number 225147), a National Centre of Competence in Research funded by the Swiss National Science Foundation.

M.N. acknowledges funding by the Federal Ministry of Education and Research of Germany (BMBF) under grant no. 01IS22094B (WestAI - AI Service Center West), under grant no. 01IS24085C (OPENHAFM) and under the grant 16HPC117K (MINERVA), as well as co-funding by EU from EuroHPC Joint Undertaking programm under grant no. 101182737 (MINERVA) and from Digital Europe Programme under grant no. 101195233 (openEuroLLM).

In addition, we thank the OpenBioML.org community and their ChemNLP project team as well as Prof. Andrew White (FutureHouse and University of Rochester, US) and Prof. David Windridge (Middlesex University, UK) for valuable discussions. We also thank Stability.AI for the access to its HPC cluster.

We thank Anagha Aneesh, Mara Schilling-Wilhelmi, and Meiling Sun for feedback on the manuscript.

References

[1] Andrew D White. "The future of chemistry is language". In: *Nature Reviews Chemistry* 7.7 (2023), pp. 457–458.

[2] Mayk Caldas Ramos, Christopher J. Collison, and Andrew D. White. "A review of large language models and autonomous agents in chemistry". In: *Chemical Science* 16.6 (2025), pp. 2514–2572. ISSN: 2041-6539. DOI: 10.1039/d4sc03921a. URL: http://dx.doi.org/10.1039/D4SC03921A.

[3] Kevin Maik Jablonka et al. "14 examples of how LLMs can transform materials science and chemistry: a reflection on a large language model hackathon". In: *Digital Discovery* 2.5 (2023), pp. 1233–1250. ISSN: 2635-098X. DOI: 10.1039/d3dd00113j. URL: http://dx.doi.org/10.1039/D3DD00113J.

[4] Zhenpeng Yao et al. "Machine learning for a sustainable energy future". In: *Nature Reviews Materials* 8.3 (Oct. 2022), pp. 202–215. ISSN: 2058-8437. DOI: 10.1038/s41578-022-00490-5. URL: http://dx.doi.org/10.1038/s41578-022-00490-5.

[5] Adrian Mirza et al. "Are large language models superhuman chemists?" In: *arXiv preprint arXiv: 2404.01475* (2024).

[6] Michael D. Skarlinski et al. "Language agents achieve superhuman synthesis of scientific knowledge". In: *arXiv preprint arXiv: 2409.13740* (2024).

[7] Markus J. Buehler. "MechGPT, a Language-Based Strategy for Mechanics and Materials Modeling That Connects Knowledge Across Scales, Disciplines, and Modalities". In: *Applied Mechanics Reviews* 76.2 (Jan. 2024). ISSN: 2379-0407. DOI: 10.1115/1.4063843. URL: http://dx.doi.org/10.1115/1.4063843.

[8] Chen Qian et al. "Can Large Language Models Empower Molecular Property Prediction?" In: *arXiv preprint arXiv: 2307.07443* (2023).

[9] Kevin Maik Jablonka et al. "Leveraging large language models for predictive chemistry". In: *Nature Machine Intelligence* 6.2 (2024), pp. 161–169.

[10] Zhiqiang Zhong, Kuangyu Zhou, and Davide Mottin. "Benchmarking Large Language Models for Molecule Prediction Tasks". In: *arXiv preprint arXiv: 2403.05075* (2024).

[11] Andre Niyongabo Rubungo et al. "LLM-Prop: Predicting Physical And Electronic Properties Of Crystalline Solids From Their Text Descriptions". In: *arXiv preprint arXiv: 2310.14029* (2023).

[12] Yuyan Liu et al. "MolecularGPT: Open Large Language Model (LLM) for Few-Shot Molecular Property Prediction". In: *arXiv preprint arXiv: 2406.12950* (2024).

[13] Luis M. Antunes, Keith T. Butler, and Ricardo Grau-Crespo. "Crystal structure generation with autoregressive large language modeling". In: *Nature Communications* 15.1 (Dec. 2024). ISSN: 2041-1723. DOI: 10.1038/s41467-024-54639-7. URL: http://dx.doi.org/10.1038/s41467-024-54639-7.

[14] Jingru Gan et al. "Large Language Models Are Innate Crystal Structure Generators". In: *arXiv preprint arXiv: 2502.20933* (2025).

[15] Nate Gruver et al. "Fine-Tuned Language Models Generate Stable Inorganic Materials as Text". In: *arXiv preprint arXiv: 2402.04379* (2024).

[16] Ziming Wei et al. "Fleming: An AI Agent for Antibiotic Discovery in Mycobacterium tuberculosis". In: (Apr. 2025). DOI: 10.1101/2025.04.01.646719. URL: http://dx.doi.org/10.1101/2025.04.01.646719.

[17] Andres M. Bran et al. "Augmenting large language models with chemistry tools". In: *Nature Machine Intelligence* 6.5 (May 2024), pp. 525–535. ISSN: 2522-5839. DOI: 10.1038/s42256-024-00832-8. URL: http://dx.doi.org/10.1038/s42256-024-00832-8.

[18] Kourosh Darvish et al. "ORGANA: A robotic assistant for automated chemistry experimentation and characterization". In: *Matter* 8.2 (Feb. 2025), p. 101897. ISSN: 2590-2385. DOI: 10.1016/j.matt.2024.10.015. URL: http://dx.doi.org/10.1016/j.matt.2024.10.015.

[19] Daniil A Boiko et al. "Autonomous chemical research with large language models". In: *Nature* 624.7992 (2023), pp. 570–578.

[20] Marcel Binz et al. “How should the advancement of large language models affect the practice of science?” In: *Proceedings of the National Academy of Sciences* 122.5 (Jan. 2025). ISSN: 1091-6490. DOI: 10.1073/pnas.2401227121. URL: http://dx.doi.org/10.1073/pnas.2401227121.

[21] Markus J. Buehler. “Generative Retrieval-Augmented Ontologic Graph and Multiagent Strategies for Interpretive Large Language Model-Based Materials Design”. In: *ACS Engineering Au* 4.2 (Jan. 2024), pp. 241–277. ISSN: 2694-2488. DOI: 10.1021/acsengineeringau.3c00058. URL: http://dx.doi.org/10.1021/acsengineeringau.3c00058.

[22] James Boyko et al. “An Interdisciplinary Outlook on Large Language Models for Scientific Research”. In: *arXiv preprint arXiv: 2311.04929* (2023).

[23] David Weininger. “SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules”. In: *Journal of chemical information and computer sciences* 28.1 (1988), pp. 31–36.

[24] Nawaf Alampara et al. “Probing the limitations of multimodal language models for chemistry and materials research”. In: *arXiv preprint arXiv: 2411.16955* (2024).

[25] Tuan Dinh et al. “LIFT: Language-Interfaced Fine-Tuning for Non-Language Machine Learning Tasks”. In: *arXiv preprint arXiv: 2206.06565* (2022).

[26] Yingwei Ma et al. “At Which Training Stage Does Code Data Help LLMs Reasoning?” In: *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL: https://openreview.net/forum?id=KIPJKST4gw.

[27] Alon Albalak et al. “A Survey on Data Selection for Language Models”. In: *arXiv preprint arXiv: 2402.16827* (2024).

[28] Long Ouyang et al. “Training language models to follow instructions with human feedback”. In: *arXiv preprint arXiv: 2203.02155* (2022).

[29] Leo Gao et al. “The pile: An 800gb dataset of diverse text for language modeling”. In: *arXiv preprint arXiv:2101.00027* (2020).

[30] Colin Raffel et al. “Exploring the limits of transfer learning with a unified text-to-text transformer”. In: *Journal of machine learning research* 21.140 (2020), pp. 1–67.

[31] Jordan Hoffmann et al. “Training Compute-Optimal Large Language Models”. In: *arXiv preprint arXiv: 2203.15556* (2022).

[32] Jared Kaplan et al. “Scaling laws for neural language models”. In: *arXiv preprint arXiv:2001.08361* (2020).

[33] Max Marion et al. “When less is more: Investigating data pruning for pretraining llms at scale”. In: *arXiv preprint arXiv:2309.04564* (2023).

[34] Suriya Gunasekar et al. “Textbooks are all you need”. In: *arXiv preprint arXiv:2306.11644* (2023).

[35] Samir Yitzhak Gadre et al. “Datacomp: In search of the next generation of multimodal datasets”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 27092–27112.

[36] Guilherme Penedo et al. “The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale”. In: *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*. Ed. by Amir Globersons et al. 2024. URL: http://papers.nips.cc/paper%5C_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets%5C_and%5C_Benchmarks%5C_Track.html.

[37] Luca Soldaini et al. “Dolma: An open corpus of three trillion tokens for language model pretraining research”. In: *arXiv preprint arXiv:2402.00159* (2024).

[38] Steven Feng et al. “Maximize Your Data’s Potential: Enhancing LLM Accuracy with Two-Phase Pretraining”. In: *arXiv preprint arXiv: 2412.15285* (2024).

[39] Christoph Schuhmann et al. “LAION-5B: An Open Large-Scale Dataset for Training next Generation Image-Text Models”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 25278–25294.

[40] Zhenqin Wu et al. “MoleculeNet: a benchmark for molecular machine learning”. In: *Chemical science* 9.2 (2018), pp. 513–530.

[41] Kexin Huang et al. “Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development”. In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*. Ed. by Joaquin Vanschoren and Sai-Kit Yeung. 2021. URL: https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/4c56ff4ce4aaf9573aa5dff913df997a-Abstract-round1.html.

[42] Sunghwan Kim et al. “PubChem Substance and Compound databases”. In: *Nucleic Acids Research* 44.D1 (Sept. 2015), pp. D1202–D1213. ISSN: 1362-4962. DOI: 10.1093/nar/gkv951. URL: http://dx.doi.org/10.1093/nar/gkv951.

[43] UniProt Consortium. “UniProt: a worldwide hub of protein knowledge”. In: *Nucleic acids research* 47.D1 (2019), pp. D506–D515.

[44] Carmelo Gonzales et al. “Evaluating Chemistry Prompts for Large-Language Model Fine-Tuning”. In: *AI for Accelerated Materials Design-NeurIPS*. 2024.

[45] Raghunathan Ramakrishnan et al. “Quantum chemistry structures and properties of 134 kilo molecules”. In: *Scientific Data* 1.1 (Aug. 2014). ISSN: 2052-4463. DOI: 10.1038/sdata.2014.22. URL: http://dx.doi.org/10.1038/sdata.2014.22.

[46] Daniel Lowe. “Chemical reactions from US patents (1976-Sep2016)”. In: (June 2017). DOI: 10.6084/m9.figshare.5104873.v1. URL: https://figshare.com/articles/dataset/Chemical_reactions_from_US_patents_1976-Sep2016_/5104873.

[47] Nadine Schneider et al. “Big Data from Pharmaceutical Patents: A Computational Analysis of Medicinal Chemists’ Bread and Butter”. In: *Journal of Medicinal Chemistry* 59.9 (Apr. 2016), pp. 4385–4402. ISSN: 1520-4804. DOI: 10.1021/acs.jmedchem.6b00153. URL: http://dx.doi.org/10.1021/acs.jmedchem.6b00153.

[48] Xiwen Jia et al. “Anthropogenic biases in chemical reaction data hinder exploratory inorganic synthesis”. In: *Nature* 573.7773 (Sept. 2019), pp. 251–255. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1540-5. URL: http://dx.doi.org/10.1038/s41586-019-1540-5.

[49] Paul Raccuglia et al. “Machine-learning-assisted materials discovery using failed experiments”. In: *Nature* 533.7601 (May 2016), pp. 73–76. ISSN: 1476-4687. DOI: 10.1038/nature17439. URL: http://dx.doi.org/10.1038/nature17439.

[50] Daniele Ongari et al. “Data-Driven Matching of Experimental Crystal Structures and Gas Adsorption Isotherms of Metal–Organic Frameworks”. In: *Journal of Chemical & Engineering Data* 67.7 (Feb. 2022), pp. 1743–1756. ISSN: 1520-5134. DOI: 10.1021/acs.jced.1c00958. URL: http://dx.doi.org/10.1021/acs.jced.1c00958.

[51] Mario Krenn et al. “SELFIES and the future of molecular string representations”. In: *Patterns* 3.10 (Oct. 2022), p. 100588. ISSN: 2666-3899. DOI: 10.1016/j.patter.2022.100588. URL: http://dx.doi.org/10.1016/j.patter.2022.100588.

[52] Yin Fang et al. “Mol-Instructions: A Large-Scale Biomolecular Instruction Dataset for Large Language Models”. In: *International Conference on Learning Representations* (2023). DOI: 10.48550/arXiv.2306.08018.

[53] Bing Su et al. “A Molecular Multimodal Foundation Model Associating Molecule Graphs with Natural Language”. In: *arXiv preprint arXiv: 2209.05481* (2022).

[54] Shengchao Liu et al. “Multi-modal molecule structure–text model for text-based retrieval and editing”. In: *Nature Machine Intelligence* 5.12 (Dec. 2023), pp. 1447–1457. ISSN: 2522-5839. DOI: 10.1038/s42256-023-00759-6. URL: http://dx.doi.org/10.1038/s42256-023-00759-6.

[55] Gang Liu et al. “Multimodal Large Language Models for Inverse Molecular Design with Retrosynthetic Planning”. In: *arXiv preprint arXiv: 2410.04223* (2024).

[56] Viggo Moro et al. “Multimodal Learning for Materials”. In: *arXiv preprint arXiv:2312.00111* (2023).

[57] Zihan Zhao et al. “ChemDFM: A Large Language Foundation Model for Chemistry”. In: *arXiv preprint arXiv:2401.14818* (2024).

[58] Sha Yuan et al. “Wudaocorpora: A super large-scale chinese corpora for pre-training language models”. In: *AI Open* 2 (2021), pp. 65–68.

[59] Botao Yu et al. “LlaSMol: Advancing Large Language Models for Chemistry with a Large-Scale, Comprehensive, High-Quality Instruction Tuning Dataset”. In: *arXiv preprint arXiv: 2402.09391* (2024).

[60] Xuan Lin et al. “Enhancing Chemical Reaction and Retrosynthesis Prediction with Large Language Model and Dual-task Learning”. In: *arXiv preprint arXiv: 2505.02639* (2025).

[61] Yifei Yang et al. “BatGPT-Chem: A Foundation Large Model For Retrosynthesis Prediction”. In: *arXiv preprint arXiv: 2408.10285* (2024).

[62] Renqian Luo et al. “BioGPT: generative pre-trained transformer for biomedical text generation and mining”. In: *Briefings in bioinformatics* 23.6 (2022), bbac409.

[63] Nathan C Frey et al. “Neural scaling of deep chemical models”. In: *Nature Machine Intelligence* 5.11 (2023), pp. 1297–1305.

[64] Tong Xie et al. “Darwin 1.5: Large language models as materials science adapted learners”. In: *arXiv preprint arXiv:2412.11970* (2024).

[65] Amanpreet Singh et al. “SciRepEval: A Multi-Format Benchmark for Scientific Document Representations”. In: *Conference on Empirical Methods in Natural Language Processing*. 2022. URL: https://api.semanticscholar.org/CorpusID:254018137.

[66] Jack W. Rae et al. “Scaling Language Models: Methods, Analysis & Insights from Training Gopher”. In: *arXiv preprint arXiv: 2112.11446* (2021).

[67] Jiasheng Ye et al. “Data mixing laws: Optimizing data mixtures by predicting language modeling performance”. In: *arXiv preprint arXiv:2403.16952* (2024).

[68] Mario Krenn et al. “Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation”. In: *Machine Learning: Science and Technology* 1.4 (2020), p. 045024.

[69] Stephen R Heller et al. “InChI, the IUPAC international chemical identifier”. In: *Journal of cheminformatics* 7 (2015), pp. 1–34.

[70] Henning Otto Brinkhaus et al. “RanDepict: Random chemical structure depiction generator”. In: *Journal of cheminformatics* 14.1 (2022), p. 31.

[71] Nawaf Alampara, Santiago Miret, and Kevin Maik Jablonka. “MatText: Do language models need more than text & scale for materials modeling?” In: *arXiv preprint arXiv:2406.17295* (2024).

[72] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. “ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction”. In: *arXiv preprint arXiv: 2010.09885* (2020).

[73] Dávid Bajusz, Anita Rácz, and Károly Héberger. “Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations?” In: *Journal of cheminformatics* 7 (2015), pp. 1–13.

[74] Hugging Face. *Model merging*. https://huggingface.co/docs/peft/en/developer_guides/model_merging. PEFT documentation. 2024. (Visited on 09/15/2025).

[75] OpenAI. *tiktoken: A fast BPE tokenizer for use with OpenAI’s models*. https://github.com/openai/tiktoken. Accessed: 2025-05-14. 2022.

[76] Summer Rosonovski et al. “Europe PMC in 2023”. In: *Nucleic Acids Research* 52.D1 (2024), pp. D1668–D1676.

[77] Guohao Li et al. *CAMEL: Communicative Agents for "Mind" Exploration of Large Scale Language Model Society*. 2023. arXiv: 2303.17760 [cs.AI].

[78] Loubna Ben Allal et al. *SmolLM2: When Smol Goes Big – Data-Centric Training of a Small Language Model*. 2025. arXiv: 2502.02737 [cs.CL]. URL: https://arxiv.org/abs/2502.02737.

[79] Jannis Born and Matteo Manica. “Trends in Deep Learning for Property-driven Drug Design”. In: *Current Medicinal Chemistry* 28.38 (2021), pp. 7862–7886.

[80] Lukas Blecher et al. “Nougat: Neural Optical Understanding for Academic Documents”. In: *arXiv preprint arXiv: 2308.13418* (2023).

[81] Viraat Aryabumi et al. “To Code, or Not To Code? Exploring Impact of Code in Pre-training”. In: *arXiv preprint arXiv: 2408.10914* (2024).

[82] Raymond Li et al. “StarCoder: may the source be with you!” In: *Trans. Mach. Learn. Res.* 2023 (2023). URL: https://openreview.net/forum?id=KoFOg41haE.

[83] Denis Kocetkov et al. “The Stack: 3 TB of permissively licensed source code”. In: *Trans. Mach. Learn. Res.* (2022). DOI: 10.48550/arXiv.2211.15533.

[84] DeepSeek-AI et al. “DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning”. In: *arXiv preprint arXiv: 2501.12948* (2025).

[85] Michael Pieler et al. “Rephrasing natural text data with different languages and quality levels for Large Language Model pre-training”. In: *arXiv preprint arXiv: 2410.20796* (2024).

[86] Guy W Bemis and Mark A Murcko. “The properties of known drugs. 1. Molecular frameworks”. In: *Journal of medicinal chemistry* 39.15 (1996), pp. 2887–2893.

[87] Lukas Blecher et al. “Nougat: Neural optical understanding for academic documents”. In: *arXiv preprint arXiv:2308.13418* (2023).

[88] Daniel M Lowe et al. *Chemical name to structure: OPSIN, an open source solution*. 2011.

[89] Rico Sennrich, Barry Haddow, and Alexandra Birch. “Neural Machine Translation of Rare Words with Subword Units”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Katrin Erk and Noah A. Smith. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. DOI: 10.18653/v1/P16-1162. URL: https://aclanthology.org/P16-1162/.

[90] Taku Kudo. “Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. Ed. by Iryna Gurevych and Yusuke Miyao. Association for Computational Linguistics, 2018, pp. 66–75. DOI: 10.18653/V1/P18-1007. URL: https://aclanthology.org/P18-1007/.

 https://doi.org/10.52202/085713-1861

Appendix

A Full affiliations

HZB Helmholtz-Zentrum Berlin für Materialien und Energie GmbH, Hahn-Meitner-Platz 1, 14109

- Adrian Mirza

HIPOLE Jena Helmholtz Institute for Polymers in Energy Applications Jena (HIPOLE Jena), Lessingstrasse 12-14, 07743 Jena, Germany

- Adrian Mirza
- Kevin Maik Jablonka

FSU Jena Laboratory of Organic and Macromolecular Chemistry (IOMC), Friedrich Schiller University Jena, Humboldtstrasse 10, 07743 Jena, Germany

- Nawaf Alampara
- Martiño Ríos-García
- Kevin Maik Jablonka

Independent researcher Mohamed Abdelalim and Michael Pieler

Faculty Faculty, 160 Old Street, London, UK

- Jack Butler
- Bethany Connolly
- Mark Worrall

Hacettepe University Biological Data Science Lab, Dept. of Computer Engineering, Hacettepe University, 06800, Ankara, Türkiye and Dept. of Health Informatics, Institute of Informatics, Hacettepe University, 06800, Ankara, Türkiye

- Tunca Dogan
- Bünyamin Şen

JSC Juelich Supercomputing Center (JSC), Research Center Juelich (FZJ), Germany

- Marianna Nezhurina

LAION

- Marianna Nezhurina

EMBL-EBI Literature Services Team, European Bioinformatics Institute, European Molecular Biology Laboratory (EMBL-EBI), Wellcome Trust Genome Campus, Hinxton, CB10 1SD, Cambridge, United Kingdom.

- Santosh Tirunagari

University of Toronto Department of Computer Science, University of Toronto

- Adamo Young

LIAC Laboratory of Artificial Chemical Intelligence (LIAC), Institut des Sciences et Ingénierie Chimiques, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

- Philippe Schwaller

NCCR Catalysis National Centre of Competence in Research (NCCR) Catalysis, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

- Philippe Schwaller

CEEC Jena Center for Energy and Environmental Chemistry Jena (CEEC Jena), Friedrich Schiller University Jena, Philosophenweg 7a, 07743 Jena, Germany

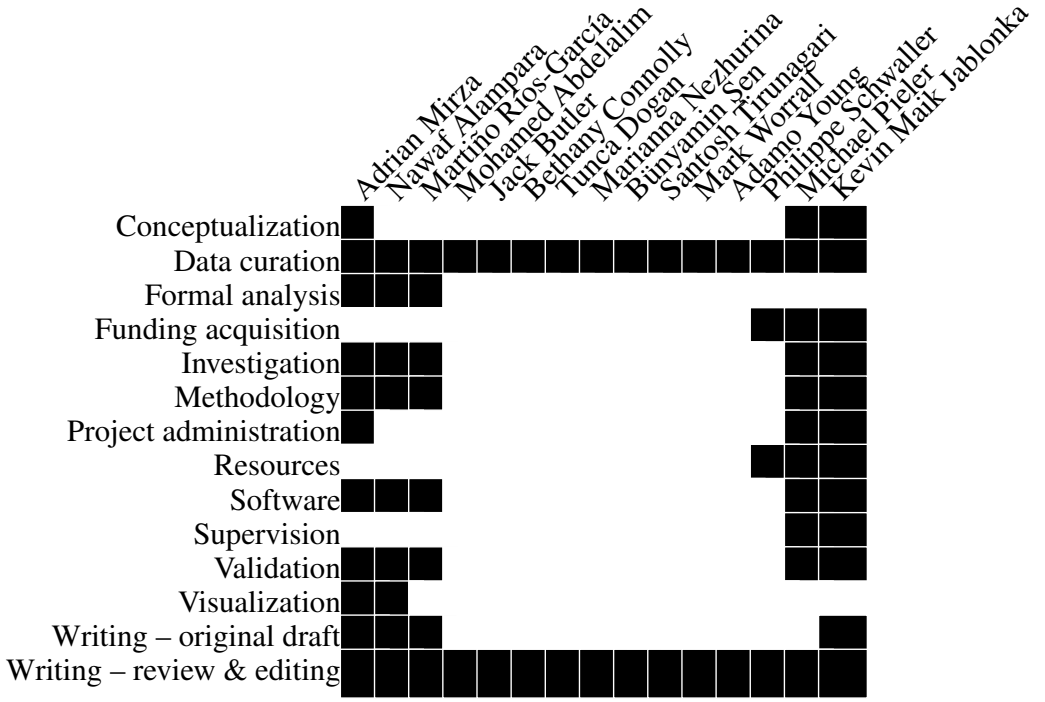
- Kevin Maik Jablonka

JCSM Jena Jena Center for Soft Matter (JCSM), Friedrich Schiller University Jena, Philosophenweg 7, 07743 Jena, Germany

- Kevin Maik Jablonka

B Credits

The project was conceptualized as part of the ChemNLP project, led by Michael Pieler and Kevin Maik Jablonka. Michael Pieler led the development of the sampling engine, which was refactored by Kevin Maik Jablonka and Adrian Mirza. Postprocessing code for natural text data was developed by Michael Pieler and Kevin Maik Jablonka. Dataset filtering code and models were developed by Nawaf Alampara, and Adrian Mirza led the final curation of ChemPile-Paper, and Nawaf Alampara led the curation of ChemPile-Code. Martiño Ríos-García created parts of the ChemPile-Education and the ChemPile-Instruct corpus and led the development of the ChemPile-website. Adrian Mirza revised the ChemPile-(m)LIFT corpora and led the creation of the HuggingFace collections. The final version was compiled by Adrian Mirza, Nawaf Alampara, and Martiño Ríos-García in the research group led by Kevin Maik Jablonka. All authors contributed to the data curation.



 https://doi.org/10.52202/085713-1861

C ChemPile Education Datasheet

Dataset Details	
Purpose of the dataset	We released ChemPile Education to make Large Language Model training in undergraduate-level chemistry more accessible for the ML community.
Curated by	The dataset was curated by the ChemNLP consortium.
Funded by	Members of the ChemNLP consortium were supported by different funding sources, which are detailed in the article describing the dataset. A main funding source is the Carl Zeiss Foundation.
Language(s)	English
License	The dataset is released under the Creative Commons (CC) BY-NC-SA 4.0 license.
Dataset Structure	
Data Instances	The following is an example from the dataset. It is part of the `LibreText_Chemistry` snapshot and was parsed on `2025-04-22T23:12:56Z`: `{` `"url":` `↪ "Bookshelves/Introductory_Chemistry/Beginning_Chemistry_(Ball)/03%3A_Atoms_Molecules_and_Ions/\n3.01%3A_Prelude_to_Atoms_Molecules_and_Ions"` `"text": "Although not an SI unit, the angstrom (A) is a useful unit of length. It is one ten-billionth of a meter, or 10 -10 m. Why is it a useful unit? The ultimate particles that compose all matter are about 10 -10 m in size, or about 1 Å. This makes the angstrom a natural---though not approved---unit for describing these particles. The angstrom unit is named after Anders Jonas Ångström, a nineteenth-century Swedish physicist. Ångström's research dealt with light being emitted by glowing objects, including the sun. Ångström studied the brightness of the different colors of light that the sun emitted and was able to deduce that the sun is composed of the same kinds of matter that are present on the earth. By extension, we now know that all matter throughout the universe is similar to the matter that exists on our own planet. Anders Jonas Ångstrom, a Swedish physicist, studied the light coming from the sun. His contributions to science were sufficient to have a tiny unit of length named after him, the angstrom, which is one ten-billionth of a meter. Source: Photo of the sun courtesy of NASA's Solar Dynamics Observatory."` `}`
Data Fields (LibreText Chemistry)	- `url (string)`: Source URL for transparency and verification - `text (string)`: Educational content about chemistry concepts
Data Fields (MIT OCW Lecture Transcripts)	- `course (string)`: Course name and identifier - `url (string)`: Original source URL for reference - `topic (string)`: Specific lecture topic - `text (string)`: Lecture transcript content - `index (int)`: Document identifier

Data Fields (US Olympiad Problems)	- `metadata (dict)`: Problem details (year, number, topic) - `problem_statement (string)`: Original olympiad problem - `options (list)`: Multiple choice options - `solution (string)`: Detailed solution explanation - `correct_answer (string)`: Correct option identifier - `text (string)`: Combined problem and solution content - `filter (bool)`: Quality control flag - `index (int)`: Document identifier
Data Fields (YouTube Transcripts as Lectures)	- `id (string)`: Unique YouTube video identifier - `title (string)`: Video title - `text (string)`: Cleaned and structured transcript content - `index (int)`: Document identifier
Data Splits	The set is divided into Training, Validation, and Test sets in a ratio of 0.9, 0.1, and 0.1, respectively.
Dataset Creation	
Curation Rationale	With ChemPile Education, we aim to provide the open-source ML community with a clean dataset about chemistry educational resources for pretraining LLMs.
Source Data	The source data consists of books, course transcripts, and US Olympiad data crawled by the ChemNLP consortium over the 2024-2025 period.

Data processing steps	The data processing pipeline consists of:
	`- URL filtering`
	`- Text extraction and parsing`
	`- Text filtering and cleaning`

Annotations	The dataset does not cover the broad field of chemistry on an undergraduate level.
Personal and Sensitive Information	Certain author names may persist in the dataset despite text-parsing and cleaning processes.
Considerations for Using the Data	
Known Limitations	Due to the crawling, some elements might not be correctly filtered, including decorators from the HTML pages or information about the educational contents.

D ChemPile LIFT Datasheet

Dataset Details	
Purpose of the dataset	We released ChemPile LIFT to make Large Language Model training in language-interfaced chemical properties, different nomenclatures, and a diverse set of templates more accessible for the ML community.
Curated by	The dataset was curated by the ChemNLP consortium.
Funded by	Members of the ChemNLP consortium were supported by different funding sources, which are detailed in the article describing the dataset. A main funding source is the Carl Zeiss Foundation.
Language(s)	English
License	The dataset is released under the Creative Commons (CC) BY-NC-SA 4.0 license.
Dataset Structure	

https://doi.org/10.52202/085713-1861

Data Instances	The following is an example from the dataset. It is part of the qm8 snapshot and was parsed on 2025-04-28T12:24:43Z:
	`{`
	`'text': 'The S0 -> S1 transition energy computed using RI-CC2/def2TZVP of the molecule with the SMILES C is 0.433 a. u.',`
	`'input': 'The S0 -> S1 transition energy computed using RI-CC2/def2TZVP of the molecule with the SMILES C is 0.433 a. u.',`
	`'output': None,`
	`'answer_choices': [],`
	`'correct_output_index': Null`
	`}`

Data Fields	- `text (string)`: the text content or question. It includes the input question or prompt related to the chemical property, often formatted as a natural language query, and the correct answer. - `input (string)`: The input text for the model, often a question or prompt related to the chemical property. - `output (string)`: The expected output or answer to the input question. - `answer_choices (list)`: A list of possible answer choices for the input question, if applicable. - `correct_output_index (float)`: The index of the correct answer in the answer_choices list.
Data Splits	The set is divided into Training, Validation, and Test sets in a ratio of 0.9, 0.1, and 0.1, respectively.
Dataset Creation	
Curation Rationale	With ChemPile LIFT, we aim to make accessible a broad range of language-interfaced chemical properties, different nomenclatures, and a diverse set of templates.
Source Data	The source data consists of transforming into text a large amount of the content of several of the most used chemical datasets.

Data processing steps	The data processing pipeline consists of:
	`- Datasets identification`
	`- Datasets cleaning and pre-processing`
	`- Template gathering`
	`- Template filling`

Annotations	The dataset does not cover the broad field of chemistry.
Personal and Sensitive Information	NA
Considerations for Using the Data	
Known Limitations	The templates used to contain the data from the datasets are probably not diverse enough.

E ChemPile Paper Datasheet

Dataset Details

Purpose of the dataset	The objective of ChemPile Paper is to make the open-source articles about chemistry more easily accessible for the ML community.
Curated by	The dataset was curated by the ChemNLP consortium.
Funded by	Members of the ChemNLP consortium were supported by different funding sources, which are detailed in the article describing the dataset. A main funding source is the Carl Zeiss Foundation.
Language(s)	English
License	The dataset is released under the Creative Commons (CC) BY-NC-ND 4.0 license.
Dataset Structure	

https://doi.org/10.52202/085713-1861

Data Instances	The following is an example from the dataset. It is part of the `euro_pmc_chemistry_papers` snapshot and was parsed on `2025-05-08T14:08:25Z`: `{` `'pmcid': None,` `'pmid': 15460519,` `'topic': chemistry,` `'confidence': 0.998524,` `'class_distribution': [` `0.00002563928137533367,` `0.0001321481540799141,` `0.0012931367382407188,` `0.9985242486000061,` `0.00002489764847268816` `],` `'text': 'Defining the great debate. Defining the great debate. Treatment guidelines offer a credible approach to MCO management of specialty pharmaceuticals by basing decisions on the best available scientific evidence. Drawbacks to using guidelines include lack of available evidence, bias on the part of the expert team, a lack of currency, and difficulties in implementation. Treatment guidelines can and often do serve, along with product labeling, as the basis for reimbursement and usage rules as specified in MCO medical policies. The relationship between provider-oriented guidelines and payment criteria varies by payer, with implications for care quality, cost, and access. Increased awareness of the advantages and disadvantages of using treatment guidelines to shape prescribing policies for specialty pharmaceuticals may lead to worthwhile non-product-specific discussions within MCOs about sources and methods used in medical policy development.'` `}`

Data Fields (arxiv-cond-mat.mtrl-sci_processed-default)	- `fn (string)`: ArXiv identifier (e.g., 10.48550_arXiv.0708.1447) - `text (string)`: Parsed text of the article - `doi (string)`: DOI of the article (if available) - `title (string)`: Article title - `authors (string)`: Article authors - `index (string)`: Document identifier
Data Fields (arxiv-physics.chem-ph_processed-default)	- `fn (string)`: ArXiv identifier (e.g., 10.48550_arXiv.0708.1447) - `text (string)`: Parsed text of the article - `doi (string)`: DOI of the article (if available) - `title (string)`: Article title - `authors (string)`: Article authors - `index (string)`: Document identifier
Data Fields (bioRxiv)	- `fn (string)`: Unique identifier (e.g., 014597_file10) - `text (string)`: Full text content extracted via Nougat
Data Fields (medRxiv)	- `fn (string)`: Unique identifier (e.g., 014597_file10) - `text (string)`: Full text content extracted via Nougat
Data Fields (chemrxiv)	- `fn (string)`: Unique identifier (e.g., 10.26434_chemrxiv-2022-cgnf5) - `text (string)`: Full text content extracted via Nougat - `doi (string)`: DOI of the article (if available) - `title (string)`: Article title - `authors (string)`: Article authors - `license (string)`: Preprint license (e.g., CC BY-NC 4.0) - `published_url (string)`: Publication URL - `index (string)`: Document identifier
Data Fields (EuroPMC)	- `pmcid (string)`: PubMed Central identifier - `pmid (string)`: PubMed identifier - `topic (string)`: Main classification topic (e.g., "Chemistry", "Physics", "Biology") - `confidence (float)`: Classification confidence score - `class_distribution (string)`: Multi-label classification distribution - `text (string)`: Full article text content
Data Splits	The set is divided into Training, Validation, and Test sets in a ratio of 0.9, 0.1, and 0.1, respectively.
Dataset Creation	
Curation Rationale	With ChemPile Paper, we aim to provide the open-source ML community with a focused dataset about chemistry research articles and resources for pretraining LLMs.
Source Data	The source data consists of articles collected by the ChemNLP consortium over the 2022-2025 period.

Data processing steps	The data processing pipeline consists of:
	`- Training and evaluating a classifier model`
	`- Classifying and filtering articles using a classifier`
	`- Cleaning of the text`

https://doi.org/10.52202/085713-1861

Annotations	The dataset does not cover the broad field of chemical research. The dataset is incomplete because it does not contain all the information referring to the broad field of chemical research.
Personal and Sensitive Information	Certain author names may persist in the dataset despite text-parsing and cleaning processes.
Considerations for Using the Data	
Known Limitations	Certain author names may persist in the dataset despite text-parsing and cleaning processes. Some of the articles in the dataset might not include chemical research and only be related to chemistry.

F ChemPile Code Datasheet

Dataset Details	
Purpose of the dataset	We release the ChemPile Code dataset to make code related to chemistry accessible for the ML community.
Curated by	The dataset was curated by the ChemNLP consortium.
Funded by	Members of the ChemNLP consortium were supported by different funding sources, which are detailed in the article describing the dataset. A main funding source is the Carl Zeiss Foundation.
Language(s)	English
License	The dataset is released under the AGPL 3.0 license.
Dataset Structure	

Data Instances	The following is an example from the dataset. It is part of the codeparrot_github-code-chemistry-python snapshot and was parsed on 2025-05-08T16:57:40Z:
	{
	'text': '####################################
	####################################\n# This program is
	↪ copyright (c) Upinder S. Bhalla, NCBS, 2015.\n# It
	↪ is licenced under the GPL 2.1 or higher.\n# There is
	↪ no warranty of any kind. You are welcome to make
	↪ copies under \n# the provisions of the GPL.\n# This
	↪ programme illustrates building a panel of multiscale
	↪ models to\n# test neuronal plasticity in different
	↪ contexts.\n##################################
	##\ntry:\n
	↪ import moogli\nexcept Exception as e:\n print(
	↪ "[INFO] Could not import moogli. Quitting...")\n
	↪ quit()\n\nimport numpy\nimport time\nimport
	↪ pylab\nimport moose\nfrom moose import neuroml\nfrom
	↪ PyQt4 import Qt, QtCore, QtGui\nimport
	↪ matplotlib.pyplot as plt\nimport sys\nimport
	↪ os\nfrom moose.neuroml.ChannelML import
	↪ ChannelML\nsys.path.append(\'../../../Demos/util\')\n
	import rdesigneur as rd\n\nPI = 3.14159265359\nuseGssa =
	↪ True\ncombineSegments = True\n# Pick your favourite
	↪ cell here.\n#elecFileName = "ca1_minimal.p"\n## Cell
	↪ morphology from Bannister and Larkman J Neurophys
	↪ 2015/NeuroMorpho\nelecFileName =
	↪ "h10.CNG.swc"\n#elecFileName =
	↪ "CA1.morph.xml"\n#elecFileName =
	↪ "VHC-neuron.CNG.swc"\nsynSpineList = []\nsynDendList
	↪ = []\nprobeInterval = 0.1\nprobeAmplitude =
	↪ 1.0\ntetanusFrequency = 100.0\ntetanusAmplitude =
	↪ 1000\ntetanusAmplitudeForSpines = 1000\nframeRunTime
	↪ = 1e-3 # 1 ms\nbaselineTime = 0.05\ntetTime =
	↪ 0.01\npostTetTime = 0.01\nruntime = baselineTime +
	↪ tetTime + postTetTime\n\ndef buildRdesigneur():\n
	↪ \'\'\'\n ############################
	##\n # Here we
	↪ define which prototypes are to be loaded in to the
	↪ system.\n # Each specification has the format\n
	↪ # source [localName]\n # source can be any of\n
	↪ # filename.extension, # Identify type of file by
	↪ extension, load it.\n # function(), #
	↪ func(name) builds object of specified name\n #
	↪ file.py:function() , # load Python file, run
	↪ function(name) in it.\n # moose.Classname #
	↪ Make obj moose.Classname, assign to name.\n #
	↪ path...'
	}

https://doi.org/10.52202/085713-1861

Data Fields (CodeParrot)	- `text (string)`: The code snippet - `repo_name (string)`: The name of the repository where the code snippet was found - `path (string)`: The path to the file within the repository - `language (string)`: The programming language of the code snippet - `license (string)`: The license of the repository - `size (integer)`: The size of the code snippet in bytes - `keyword (list)`: A list of keywords that were used to filter the code snippet - `text_hash (string)`: A hash of the code snippet to avoid duplicates
Data Fields (StarCoder)	- `text (string)`: The code snippet - `repo_name (string)`: The name of the repository where the code snippet was found - `keyword (list)`: A list of keywords that were used to filter the code snippet - `text_hash (string)`: A hash of the code snippet to avoid duplicates
Data Splits	The set is divided into Training, Validation, and Test sets in a ratio of 0.9, 0.1, and 0.1, respectively.
Dataset Creation	
Curation Rationale	The objective is to curate a subset from big code datasets, and filter the data related to chemistry libraries.
Source Data	The source data is StarCoder and Codeparrot-Github-code

Data processing steps	The data processing pipeline consists of:
	`- Curating a big code dataset`
	`- Filter them based on keywords`
	`- Simple deduplication based on hashing`

Annotations	The dataset does not cover all the information and data relative to the broad field of chemistry and all its coding variables and possibilities.
Personal and Sensitive Information	Due to incomplete filtering, some code snippets contain information relative to the author/s of the code.
Considerations for Using the Data	
Known Limitations	The accuracy of the classifier used to select the code relative to chemistry is not perfect. Therefore, the dataset might contain some code that is only slightly related to chemistry.

G ChemPile Reasoning Datasheet

Dataset Details	
Purpose of the dataset	The purpose of ChemPile Reasoning is to provide the community with a clean, well-curated, open-source, and high-quality resource to enhance the reasoning capabilities in chemistry of LLMs.
Curated by	The dataset was curated by the ChemNLP consortium.
Funded by	Members of the ChemNLP consortium were supported by different funding sources, which are detailed in the article describing the dataset. A main funding source is the Carl Zeiss Foundation.
Language(s)	English
License	The dataset is released under the Creative Commons (CC) BY-SA license.
Dataset Structure	

Data Instances	The following is an example from the dataset. It is part of the `claude-3.5-distilled-spectral-reasoning-default` snapshot and was parsed on `2025-04-22T23:12:56Z`: `{` `'text': "\nWhich of the following is the correct molecule that corresponds to the given spectra?\n\nA. CC(N)=C(N)C(C)(C)C B. CC(C)=C(N)C(C)(C)N C. CC(C)=C(N)C(C)(C)C D. CSC(C)(C)C(N)=C(C)C\nThe 1H NMR spectrum is as follows: Deltas 1.19 (s, 9H), 1.52 (s, 6H).\nPlease PLACE YOUR ENTIRE REASONING PATHWAY within [START_REASONING][END_REASONING] tags and the correct SMILES within [START_SMILES][END_SMILES] tags.\\n\\nLet's analyze the 1H NMR spectrum step by step:\\n\\n1. We see two signals in the spectrum:\\n - delta 1.19 (s, 9H)\\n - delta 1.52 (s, 6H)\\n\\n2. Both signals are singlets (s), which means they don't couple with any other protons.\\n\\n3. The signal at delta 1.19 integrates for 9 protons. This is likely a tert-butyl group (-C(CH3)3).\\n\\n4. The signal at delta 1.52 integrates for 6 protons. This could be two equivalent methyl groups.\\n\\n5. We don't see any signals for NH2 protons, which would typically appear at higher chemical shifts (around 1-5 ppm, broad singlet). This suggests that if there are any nitrogen atoms, they are likely not primary amines.\\n\\n6. We also don't see any signals for alkene protons, which would typically appear at higher chemical shifts (around 4-6 ppm).\\n\\nNow, let's look at the given options:\\n\\nA. CC(N)=C(N)C(C)(C)C - This has two NH groups and would show signals for these protons.\\nB. CC(C)=C(N)C(C)(C)N - This also has two NH groups.\\nC. CC(C)=C(N)C(C)(C)C - This matches our spectrum. It has a tert-butyl group (9H) and two equivalent methyl groups (6H).\\nD. CSC(C)(C)C(N)=C(C)C - This has a sulfur atom and would likely show different chemical shifts.\\n\\nBased on this analysis, the molecule that best matches the given 1H NMR spectrum is option C: CC(C)=C(N)C(C)(C)C\\n\\nThis molecule has:\\n- A tert-butyl group (-C(CH3)3) accounting for the 9H singlet at delta 1.19\\n- Two equivalent methyl groups accounting for the 6H singlet at delta 1.52\\n- No NH protons\\n- No alkene protons (the double bond is tetrasubstituted)\\n\\n",` `}`

 https://doi.org/10.52202/085713-1861

Data Fields (DeepSeek Configurations)	- `smiles (string)`: The SMILES representation of the molecule associated with the spectral data - `reasoning (string)`: The reasoning trace or explanation provided by the model for the spectral analysis - `response (string)`: The model's response to the spectral reasoning task - `response_smiles (string)`: The SMILES representation of the molecule parsed from the model's response - `correct (boolean)`: If the model's response is correct or not, based on the spectral data - `question (string)`: The question or task related to the spectral data that the model is addressing - `text (string)`: The joined text of the question, reasoning, and response for the model's output
Data Fields (Claude-3.5-Sonnet Configuration)	- `prompt (string)`: The prompt or question related to the spectral data - `extracted_reasoning (string)`: The reasoning trace or explanation with the final answer provided by the model for the spectral analysis - `text (string)`: The joined text of the prompt and extracted reasoning for the model's output - `index (int)`: The index of the example in the dataset
Data Fields (StackExchange Completion and Instruction Format)	- `text (string)`: The original text from the Stack Exchange post - `input (string)`: The input text for the model, which may include the question or context - `output (string)`: The expected output or answer to the question - `answer_choices (list)`: A list of possible answer choices for the question - `correct_output_index (int)`: The index of the correct answer in the answer_choices list
Data Fields (StackExchange Raw Data)	- `title (string)`: The title of the Stack Exchange post - `q (string)`: The question text from the Stack Exchange post - `a (string)`: The answer text from the Stack Exchange post - `split (string)`: The split of the dataset (train, test, or validation) - `index (int)`: The index of the post in the dataset - `text (string)`: The joined text of the title, question, and answer for the post
Data Splits	The set is divided into Training, Validation, and Test sets in a ratio of 0.9, 0.1, and 0.1, respectively.
Dataset Creation	
Curation Rationale	With ChemPile Reasoning, we aim to enrich the chemical reasoning data in the open-source community.
Source Data	The source data consists of reasoning traces distilled from the leading models by the ChemNLP consortium over the 2024-2025 period. Additionally, it contains Stack Exchange discussions in the field of Materials, Physics, and Chemistry collected over the 2022-2025 period.

Data processing steps	The data processing pipeline consists of:
	`- URL filtering`
	`- Text extraction`

Annotations	The dataset does not cover the broad field of chemical reasoning, and all the chemistry-related tasks.
Personal and Sensitive Information	Certain user names may persist in the dataset despite text-parsing and cleaning processes.
Considerations for Using the Data	
Known Limitations	Due to the crawling, some elements might not be correctly filtered, including decorators from the HTML pages or information about the educational contents.

H ChemPile MLIFT Datasheet

Dataset Details	
Purpose of the dataset	The purpose of the ChemPile MLIFT dataset is to provide the ML community with a comprehensive dataset with language-interfaced chemical property text, accompanied by an image of the molecule involved.
Curated by	The dataset was curated by the ChemNLP consortium.
Funded by	Members of the ChemNLP consortium were supported by different funding sources, which are detailed in the article describing the dataset. A main funding source is the Carl Zeiss Foundation.
Language(s)	English
License	The dataset is released under the Creative Commons (CC) BY-NC-SA 4.0 license.
Dataset Structure	

Data Instances	The following is an example from the dataset. It is part of the `BACE-multimodal` snapshot and was parsed on `2025-04-22T08:22:19Z`:
	`{`
	`'SMILES':`
	`↪ 'Cc1ccccc1-c1ccc2nc(N)c(C[C@@H](C)C(=O)N[C@@H]3CCOC(C)(C)C3)cc2c1',`
	`'pIC50': 9.1549015,`
	`'BACE_inhibition': 1,`
	`'IMAGE': <PIL.PngImagePlugin.PngImageFile image mode=RGB size=300x300 at 0x15481C082A50>,`
	`'SELFIES':`
	`↪ '[C][C][=C][C][=C][C][=C][Ring1][=Branch1][C][=C][C][=C][N][=C][Branch1][C][N][C][Branch2][Ring1][=Branch2][C][C@@H1][Branch1][C][C][C][=Branch1][C][=O][N][C@@H1][C][C][O][C][Branch1][C][C][Branch1][C][C][C][Ring1][Branch2][=C][C][Ring2][Ring1][Branch1][=C][Ring2][Ring1][=Branch2]',`
	`'InChIKey': 'QMSHBBGXSXAGOO-XMSQKQJNSA-N',`
	`'IUPAC':`
	`↪ '(2R)-3-[2-azanyl-6-(2-methylphenyl)quinolin-3-yl]-N-[(4R)-2,2-dimethyloxan-4-yl]-2-methyl-propanamide',`
	`'template_original': 'The {#compound|chemical!} with the {SMILES__description} of {SMILES#} {#shows|exhibits|displays!} {BACE_inhibition#no &NULL}{BACE_inhibition__names__noun}.',`
	`'template': 'The compound with the SMILES of Cc1ccccc1-c1ccc2nc(N)c(C[C@@H](C)C(=O)N[C@@H]3CCOC(C)(C)C3)cc2c1 exhibits inhibition of the human beta-secretase 1 (BACE-1).'}`

Data Fields	- `SMILES (string)`: SMILES representation of the molecule - `property (float)`: the value of the property relative to the molecule - `IMAGE (PIL object)`: image of the molecule involved - `SELFIES (string)`: SELFIES representation of the molecule - `InChIKey (string)`: InChIKey representation of the molecule - `IUPAC (string)`: IUPAC name of the molecule involved - `template_original (string)`: template to adopt with the different representations - `template (string)`: template to adopt with the different representations
Data Splits	The set is divided into Training, Validation, and Test sets in a ratio of 0.9, 0.1, and 0.1, respectively.
Dataset Creation	

https://doi.org/10.52202/085713-1861

Curation Rationale	With ChemPile Education, we aim to provide one of the first Multimodal Language-Interfaced datasets relative to chemistry.
Source Data	The source data consists of transforming a large amount of text content from several of the most used chemical datasets into text, providing an image of the involved molecule for each of the rows.
Data processing steps	The data processing pipeline consists of: - Datasets identification - Datasets cleaning and pre-processing - Template gathering - Image generation - Representation generation
Annotations	The dataset does not cover all the information related to the broad field of chemistry.
Personal and Sensitive Information	NA
Considerations for Using the Data	
Known Limitations	The templates used to contain the data from the datasets are probably not diverse enough.

I ChemPile Caption Datasheet

Dataset Details	
Purpose of the dataset	We released ChemPile Caption to make multimodal Large Language Model training in undergraduate-level chemistry more accessible for the ML community.
Curated by	The dataset was curated by the ChemNLP consortium.
Funded by	Members of the ChemNLP consortium were supported by different funding sources, which are detailed in the article describing the dataset. A main funding source is the Carl Zeiss Foundation.
Language(s)	English
License	The dataset is released under the Creative Commons (CC) BY-NC-SA 4.0 license.
Dataset Structure	
Data Instances	The following is an example from the dataset. It was parsed on `2025-05-06T18:33:36Z`: `{` `'text': 'Figure \\(\\PageIndex{5}\\): Mild cognitive impairment (MCI) is a state between normal ageing and dementia, where someone's mind is functioning less well than would be expected for their age. This image is for illustrative purposes only. (Public Domain; Center For Functional Imaging, Lawrence Berkeley National Laboratory. Alzheimer's Disease Neuroimaging Initiative (ADNI).',` `'image': <PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=804x400 at 0x14F34EFF7EC0>` `}`

Data Fields	- `text (string)`: the text content - `image (PIL object)`: the image containing chemical knowledge
Data Splits	The set is divided into Training, Validation, and Test sets in a ratio of 0.9, 0.1, and 0.1, respectively.
Dataset Creation	
Curation Rationale	With ChemPile Caption, we aim to release an open-source image dataset with images and text at an undergraduate chemistry level.
Source Data	The source data consists of pairs of image-caption data crawled from LibreTexts Chemistry by the ChemNLP consortium over the 2024-2025 period.

Data processing steps	The data processing pipeline consists of:
	`- URL filtering`
	`- Text extraction and parsing`
	`- Image extraction and parsing`
	`- Dataset cleaning`

Annotations	The dataset does not cover the broad field of chemistry and is biased towards open-source textbook data.
Personal and Sensitive Information	NA
Considerations for Using the Data	
Known Limitations	Due to the crawling, some elements might not be correctly filtered, including decorators from the HTML pages or information about the educational contents.

J ChemPile Instruction Datasheet

Dataset Details	
Purpose of the dataset	ChemPile Instruction is built to push forward the instruction-following training of LLMs in the field of chemistry.
Curated by	The dataset was built by the ChemNLP consortium.
Funded by	Members of the ChemNLP consortium were supported by different funding sources, which are detailed in the article describing the dataset. A main funding source is the Carl Zeiss Foundation.
Language(s)	English
License	The dataset is released under the Creative Commons (CC) BY 4.0 license.
Dataset Structure	

https://doi.org/10.52202/085713-1861

<table>
<tr><td>Data Instances</td><td>The following is an example from the dataset. It is part of the <code>chempile-education</code> snapshot and was parsed on <code>2025-05-06T18:33:36Z</code>:
<pre>{
  "first_tag": [],
  "second_tag": [
    "General Chemistry",
    "Organic Chemistry"
  ],
  "origin": {
    "config": "LibreText_Chemistry-default",
    "dataset": "jablonkagroup/chempile-education",
    "prompt_type": "engaging",
    "split": "train"
  },
  "messages": [
    {
      "content": "Can you explain what molecular compounds
      ↪ are and how to name them?",
      "role": "user"
    },
    {
      "content": "Sure! Molecular compounds are a type of
      ↪ inorganic compound that consist of discrete
      ↪ molecules. They're made up of nonmetals, and you
      ↪ most likely know some typical examples like water,
      ↪ which has the formula \\(\\ce{H_2O} \\), and
      ↪ carbon dioxide, which is \\(\\ce{CO_2} \\).",
      "role": "assistant"
    }
  ]
}</pre></td></tr>
<tr><td>Data Fields</td><td>- <code>first_tag (list[str])</code>: list with the “requires” keywords, e.g., “["requires-knowledge", "requires-calculation"]” - <code>second_tag (list[str])</code>: list with the chemical domains covered, e.g., “["Analytical Chemistry", "Organic Chemistry"]” - <code>origin (dict[str, str])</code>: dict specifying the original document origin, including dataset, config, split, and prompt used for the rephrasing. - <code>messages (list[dict[str, str]]</code>: list of messages using the LiteLLM format.</td></tr>
<tr><td>Data Splits</td><td>The set is divided into Training, Validation, and Test sets in a ratio of 0.9, 0.1, and 0.1, respectively.</td></tr>
<tr><td colspan="2">Dataset Creation</td></tr>
<tr><td>Curation Rationale</td><td>With ChemPile Instruction, we aim to release the largest open-source multi-turn chemical dataset to advance the post-training of LLM for the chemical domains.</td></tr>
<tr><td>Source Data</td><td>The source of data is other ChemPile subsets, such as ChemPile Paper, Education, or Reasoning. The source data was then rephrased into multi-turn conversations using the proprietary LLM <code>gpt-4o-mini-2024-07-18</code>.</td></tr>
<tr><td>Data processing steps</td><td>The data processing pipeline consists of:
<pre>- Selection of the data to rephrase
- Curation and optimization of the prompts to use
- Rephrasing using the LLM.
- Manual review of random samples of the new data to ensure
↪ validity.</pre></td></tr>
</table>

Annotations	The dataset does not cover the broad field of chemistry and is biased towards open-source textbook data.
Personal and Sensitive Information	NA
Considerations for Using the Data	
Known Limitations	The quality of the dataset is highly dependent on the LLM used for rephrasing.

https://doi.org/10.52202/085713-1861

K Modeling using ChemPile

To assess the quality of our dataset, we conducted a series of experiments. As a first step, we randomly sampled 100M tokens from four of our subsets: ChemPile-LIFT, ChemPile-Education, ChemPile-Paper and ChemPile-Instruct. We then employed several training approaches to evaluate the different subsets and a combination thereof. One of our approaches is also novel in the field of the chemical foundation models: combining LoRA adapters trained on our subsets.

In Appendix K, we refer to two distinct approaches for adapter merging: LoRA-Ensemble and LoRA-Merge. The former involves the selection of the generation with the lowest perplexity across a set of LoRA adapters, while the latter describes a linear merge (we give equivalent weights to all the adapters).

Table 10: **Model performance on the different topics sources the ChemBench benchmark**[5]

Subset	Token Count	Model	Overall Benchmark	Analytical Chemistry	Technical Chemistry	Inorganic Chemistry	Organic Chemistry	General Chemistry	Chemical Preference	Physical Chemistry	Materials Science	Toxicity & Safety
–	–	Mixtral 8x7B	0.42	0.27	0.32	0.55	0.48	0.42	0.54	0.33	0.42	0.27
–	–	Qwen2.5-7B-Instruct	0.43	0.24	0.45	0.57	0.50	0.44	0.50	0.37	0.39	0.30
ChemPile-Paper	100M	LoRA	0.45	0.29	**0.60**	0.59	0.49	0.44	0.54	0.32	0.41	0.31
ChemPile-Education	100M	LoRA	0.44	0.26	0.58	0.57	0.49	0.46	0.51	0.39	0.43	0.32
ChemPile-LIFT	100M	LoRA	0.43	0.28	0.55	0.37	0.44	0.35	0.50	0.27	0.42	0.26
LIFT, Education, Paper, Reasoning, Mix	395M	LoRA-Ensemble	**0.47**	**0.34**	0.58	0.57	0.52	0.46	**0.55**	0.36	**0.48**	0.33
LIFT, Education, Paper, Reasoning (Mix)	315M	LoRA-Merge	**0.47**	0.30	0.55	**0.60**	**0.55**	**0.51**	**0.55**	**0.38**	0.46	**0.34**
ChemPile-Instruction	100M	SFT	0.43	0.25	0.48	0.50	0.49	0.41	0.53	0.36	0.38	0.32
LIFT, Education, Paper, Reasoning + Instruction	415M	LoRA Mix + SFT	0.46	**0.34**	0.58	0.57	0.52	0.46	0.55	0.36	**0.48**	0.33

L Licenses of the datasets

The datasets comprising ChemPile operate under heterogeneous licensing agreements reflecting their diverse origins, detailed in Table 11. Specifically, the mLIFT, Education, and Caption datasets are distributed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0). The Paper dataset employs the more restrictive CC BY-NC-ND 4.0 license, while the Code repository utilizes the software AGPL 3.0 license. Notably, the Reasoning and Instruction datasets feature the most permissive terms through their CC BY-SA 4.0 and CC BY 4.0 licenses. This licensing framework preserves the original terms associated with each constituent data source while facilitating transparent reuse guidelines.

Table 11: **Different licenses of the datasets forming the ChemPile**. While ChemPile-Instruction has very permissive licenses such as CC BY 4.0, allowing broad reuse including for commercial purposes, the other datasets of the ChemPile adopt more restrictive Creative Commons variants, such as CC BY-NC-SA 4.0, which permits adaptation and sharing but only for non-commercial use and requires derivatives to be licensed under identical terms, or CC BY-NC-ND 4.0 which further prohibits the creation of derivative works. CC BY-SA 4.0 allows both commercial and non-commercial use but mandates attribution and sharing under the same license.

Dataset	License
ChemPile-Education	CC BY-NC-SA 4.0
ChemPile-Paper	CC BY-NC-ND 4.0
ChemPile-LIFT	CC BY-NC-SA 4.0
ChemPile-mLIFT	CC BY-NC-SA 4.0
ChemPile-Code	AGPL 3.0
ChemPile-Reasoning	CC BY-SA 4.0
ChemPile-Caption	CC BY-NC-SA 4.0
ChemPile-Instruction	CC BY 4.0

M Data splitting for tabular datasets

We concatenated all molecules for the datasets containing the SMILES representation. The challenge lies in achieving a consistent train–test–validation split across all such tabular datasets, which are distinct both in terms of the number of molecules, and molecular diversity. We demonstrate the full algorithm to obtain non-empty scaffold splits across tabular datasets as pseudo-code. The full Python implementation can be found on GitHub.

Pseudo-code for scaffold splitting across tabular datasets

```
# STEP 1: Create global split assignments for all molecules
function CreateGlobalMoleculeSplits():
    all_molecules = empty set

    # Collect all unique molecules across designated datasets
    for each dataset in scaffold_split_datasets:
        molecules = extract_smiles_from(dataset)
        add molecules to all_molecules

    # Convert to list for indexing
    all_molecules_list = convert_to_list(all_molecules)

    # Shuffle list
    shuffle(all_molecules_list)
```


https://doi.org/10.52202/085713-1861

```
    # Assign to splits based on fractions
    train_size = floor(length(all_molecules_list) * train_fraction)
    val_size = floor(length(all_molecules_list) * val_fraction)

    train_molecules = all_molecules_list[0 : train_size]
    val_molecules = all_molecules_list[train_size : train_size + val_size]
    test_molecules = all_molecules_list[train_size + val_size : end]

    # Save for future reference
    save_to_file("val_molecules.txt", val_molecules)
    save_to_file("test_molecules.txt", test_molecules)

    return train_molecules, val_molecules, test_molecules

# STEP 2: Apply consistent splits to all datasets with SMILES
function ApplyConsistentSplitsToAllDatasets(val_molecules, test_molecules):
    # Load predefined splits
    val_molecules = read_from_file("val_molecules.txt")
    test_molecules = read_from_file("test_molecules.txt")

    for each dataset in all_datasets_with_smiles:
        smiles_columns = identify_smiles_columns(dataset)

        # Process each row
        for each row in dataset:
            molecules_in_row = extract_molecules_from_columns(row,
            ↪  smiles_columns)

            # Apply split priority logic
            if any molecule in molecules_in_row is in test_molecules:
                row.split = "test"
            else if any molecule in molecules_in_row is in val_molecules:
                row.split = "valid"
            else:
                row.split = "train"

        save_dataset_with_splits(dataset)


# STEP 3: Main execution flow
function main():
    # First perform scaffold split to establish global molecule assignments
    train_molecules, val_molecules, test_molecules =
    ↪  CreateGlobalMoleculeSplits()

    # Handle amino acid sequences similarly (not shown)
    # ...

    # Apply consistent splits across all remaining datasets with SMILES
    ApplyConsistentSplitsToAllDatasets(val_molecules, test_molecules)

    # Handle remaining datasets with random splits
    # ...
```

The same concatenation approach has been implemented for amino-acid sequences, but in this case most datasets are relatively large (at least 200k). After concatenating all sequences, we apply a random train–test–validation split, based on the general idea presented above.

N Sampling engine

Our template sampler consists of more than 800 lines of Python code meant to cover many functionalities. In Figure 5 we show an example of how the sampling engine operates. For the engine to work as intended two files are needed: `meta.yaml` and `data_clean.csv`. The former contains the information about the column names (with specific metadata about semantic types), the text templates, semantic variations of how a property or a representation can be named. The `meta.yaml` file also contains other metadata such as the URL sources, the citation, the number of points, and a short description of the dataset.

The `data_clean.csv` file contains the raw data, with one or more columns for both representations and properties. When sampling, the pipeline extracts the information from this file by pointing to a specific column with the `#` (e.g. `SMILES#`, BACE_inhibition#) symbol. For sampling multiple choice questions the `%` is used (e.g. `SMILES%` to sample SMILES as options). To indicate the number of MCQ questions, and the type of symbols indicating the different options the following syntax is used: `%multiple_choice_enum%2-5%aA1`. In this example we randomly sample 2 to 5 options. The `__` component in, for example, `{BACE_inhibition__names__adjective}` points towards one of the adjectives in the `names` subfield of the column with the identifier `BACE_inhibition`.

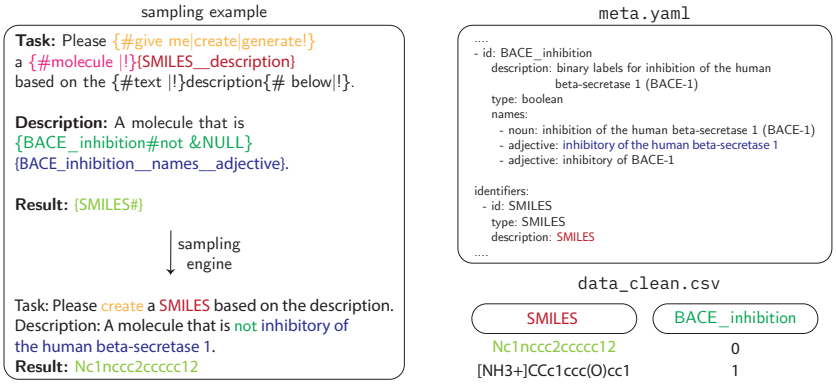


Figure 5: **Example of how our sampling engine operates.** The sampling depends on two a metadata file, and a raw data file containing all the correct columns as described in the metadata. The colors match what elements of the text templates is replaced in the final text with natural language.

In Table 12 we present the five different template types used to generate the ChemPile-(M)LIFT datasets. Each example uses special grammar, aforementioned. The pipeline is robust with regards to the representation type, and can include valuable additional information such as units for properties.

O Embeddings for correlation analysis

The embeddings in Figure 3 were generated using OpenAI's `text-embedding-3-large` model. Analysis of 5,000 molecular pairs revealed that the cosine similarity of IUPAC name embeddings demonstrates a stronger correlation with molecular graph chemical similarity (Pearson correlation coefficient r = 0.722) compared to SMILES embeddings (r = 0.521). The difference of 0.201 was evaluated using Fisher's r-to-z transformation, yielding a z-statistic of 16.7 ($p < 10^{-9}$), which describes a statistically significant difference between the two Pearson correlations.

https://doi.org/10.52202/085713-1861

Table 12: **Template types and examples for each template.** We represent here the five template types used to create the LIFT and (M)LIFT datasets.

Template type	Template
Completion (generative)	The {#CIF|CIF file|CIF card!} of the material with {#chemical formula|composition|reduced formula!} {formula#}, {spacegroup_number__names__noun} {spacegroup_number#} and {density__names__noun} {density#} {density__units} is {cif#}.
Completion (predictive)	The {spacegroup__names__noun} of the symmetrized version of the {#material|compound|solid!} with the {#CIF|CIF file|CIF card!} {cif#} is {spacegroup#}.
Instruction (generative)	Task: {#Please design|Design!} a {#crystal structure|material|compound|material structure|structure!} based on the {cifstr__names__noun}. CIF: {cifstr#} {#Description|Answer!}: {description#}
Instruction (predictive)	Task: Please {#determine|predict|estimate!} if the {#molecule|compound!} with the {SMILES__description} {SMILES#} is {MUV-713__names__noun}. Result: {MUV-713#no&yes}
Instruction (multiple-choice)	{#Task|Problem statement!}: Answer the {#multiple choice|multiple-choice|MCQ!} question. {#Question|Query!}: What is the {herg_central_at_1uM__names__noun} of a {#compound|drug!} with the {SMILES__description} {SMILES#}? Constraint: You must return none, one or more options from {%multiple_choice_enum%2-5%aA1} without using any {#other|additional!} words. Options: {herg_central_at_1uM%SMILES%} Answer: {%multiple_choice_result}. {herg_central_at_1uM#}{herg_central_at_1uM__units}.

P Dataset details

P.1 ChemPile-Education

The general idea pursued by the ChemPile-Education dataset is visually presented in Figure 6. LibreTexts constitute the exposition of the model to diverse background chemistry knowledge, and worked examples. This is further reinforced by lectures from MIT-OCW and YouTube, where examples are often explained and taught step-by-step. Further, we also collected US Olympiad data that can be used in training modes like finetuning or reinforcement learning.

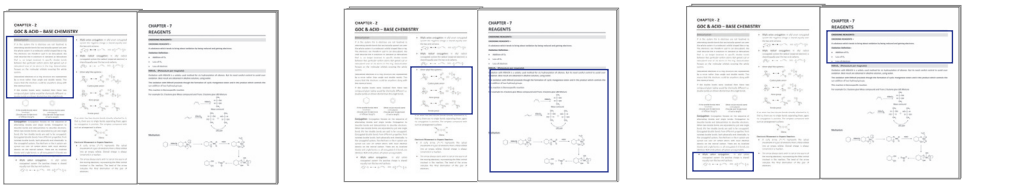


Figure 6: **ChemPile-Education covers different kinds of educational data.** Textbook data contains foundational knowledge, but also worked examples.

P.1.1 Sources

LibreTexts Chemistry We systematically extracted and processed the primary textual content from LibreTexts Chemistry HTML documents using a custom Python pipeline—utilizing the Beautiful Soup library for DOM (Document Object Model) parsing. Non-content elements, including navigation menus, scripts, embedded media, acknowledgments, and references, were programmatically removed to isolate chemically relevant educational material. This automated extraction process covered all HTML files in the LibreTexts Chemistry repository (accessed 2025-04-21), generating a structured corpus for subsequent natural language processing analysis. The final dataset contains 114,288,417 tokens across partitioned subsets: training (102,922,903 tokens), validation (5,694,288 tokens), and test sets (5,671,224 tokens).

US Olympiad data We manually extracted US Olympiad papers as PDFs from 2003 to 2024 as provided by the American Chemical Society. We used the `Gemini 2.0 Flash Thinking Experimental 01-21` model and its large context window. Two PDF files were provided to the model: the problem file and the solution file for each year. Based on the problem index, a JSON file was generated with the necessary metadata, question-answer pairs, and answer options. The dataset was then filtered to include only problem solutions that exceeded 250 characters. The rate of success has been evaluated manually on 50 extracted examples. While we do not observe any mismatch between the question and answer pairs, a few minor mismatches are present (e.g. Δ replaced by A).

MIT OpenCourseWare transcripts To download the data from the MIT OCW we made use of the platform's topic-based search (selecting biology, chemistry, chemical engineering and physics), which allowed us to identify the relevant URL structure for downloading the relevant document. We also provide the course name and the links used to download the course contents.

YouTube transcripts We find a list of YouTube videos by querying YouTube on a list of LLM-generated keywords. Then, the list of videos is filtered by their license (only the videos labeled as *Creative Commons reuse allowed* were selected). This criterion was achieved by filtering for videos containing EgIwAQ%3D%3D in the HTML code of respective pages. We then use `gpt-4.1` to rewrite the raw transcripts into lecture-like content. The advantage of rewriting lies in the inherent gaps in scientific transcripts (i.e., sometimes scientific terms are incorrectly transcribed), which the LLM can fill. All transcripts in foreign languages (e.g. Hindi) have also been translated into English. The prompt used to achieve this is given in the snippet below:

LLM Prompt

```
The following is a transcript of a YouTube video.
Your task is to rewrite the transcript into a lecture format.

Return only the lecture, without any additional text or explanation.
Use the tags [LECTURE] and [/LECTURE] to indicate the start and end of
the lecture.
The lecture should be structured and easy to follow, feel free to fill
knowledge gaps.
If the discussion is mathematical, include the equations in LaTeX format.
Same for chemical equations, use the appropriate format.
The lecture should be in English and should not contain any other
language.

The lecture should be in a single paragraph, without any line breaks.

The transcript is as follows:

{transcript}
```


https://doi.org/10.52202/085713-1861

P.2 ChemPile-Code dataset

Distribution of keywords Simulation tools dominate the landscape with the highest number of entries matching simulation tool keywords, indicating their common use in scientific computation across domains. Visualization and Analysis tools follow. Notice that the visualization here is very domain-specific visualization codes (for example, PyMol, VMD, and not matplotlib or plotly). The keywords used for filtering are provided in Table 13, and the distribution of the five categories of keywords is shown in Figure 7.

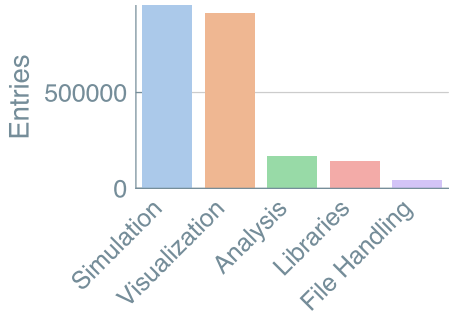


Figure 7: **Keyword distribution by category**. The plot shows the dataset distribution based on keywords identified in the entry. Here we have considered all the keywords from different categories irrespective of the domain (chemistry, materials science, and biology).

Keywords for filtering code Table 13 shows the keywords used to filter and create ChemPile-Code dataset.

P.3 ChemPile-Paper

P.3.1 Sources

EuroPMC The Europe PMC dataset is a comprehensive, open-access repository of life sciences literature, which includes peer-reviewed full-text research articles, abstracts, and preprints. Europe PMC houses around 27 million abstracts and 5 million full-text articles. We classified this articles and then filtered out only the abstract and full text articles which are related to chemistry or close to chemistry.

To create a chemistry-specific subset of the extensive Europe PMC dataset, we employed a custom-trained BERT multilabel classifier. This classifier was developed from the ground up using the CAMEL dataset, which provided 20,000 examples for each of the following topics: chemistry, code, math, biology, and physics. The classifier's performance in identifying chemistry-related articles was evaluated on approximately 150 manually annotated entries from the FineWebMath dataset, achieving an F_1-score of approximately 0.77 for the chemistry label. We split the document into chunks of 512 tokens, with an overlap of 50 tokens between adjacent chunks, and then took a weighted average of the predictions to determine the topic. We only considered the first five chunks for classifying a document. The abstract-only filtered dataset is over 3 billion tokens.

Specialized preprint servers To collect chemistry-related articles from specialized preprint servers such as ChemRxiv, BioRxiv and MedRxiv we used the PaperScraper package from Born and Manica [79]. All the articles were processed using the Nougat OCR base model from Blecher et al. [87]. We also collect and distribute metadata such as the license, the publication date, the author list and the title of each preprint.

Arxiv The Arxiv is a pre-print server initially created for the rapid distribution of physics papers. However, with time, its scope grew to include many other quantitative fields such as materials science, and quantitative biology. Thus, based on the topic keywords `cond-mat.mtrl-sci` and `physics.phys-chem`, we extracted the DOIs of chemistry-related articles. We then further used the PaperScraper [79] to download the PDF of the respective DOIs.

Table 13: **Overview of Keyword Categories Used for Filtering the Code Dataset.** The table includes all keywords within each predefined list. Note: The "Keyword Count" reflects the number of distinct terms in each specific list before they are aggregated into a single unique set for regex matching

Domain	Category	Count	All Keywords
Chemistry	Simulation	10	GROMACS, LAMMPS, OpenMM, CP2K, Quantum ESPRESSO, NWChem, Psi4, PySCF, ABINIT, Octopus
	Analysis	7	MDAnalysis, MDTraj, ChemPy, RDKit, ASE, PySCeS, Open Babel
	Visualization	6	VMD, PyMOL, Jmol, Avogadro, Gabedit, RasMol
	File Handling	5	Open Babel, Pybel, cclib, Chemfiles, ASE
	Libraries	8	RDKit, ChemPy, PySCeS, ASE, OpenFF Toolkit, Chemfiles, Open Babel, CDK
Materials Science	Simulation	10	LAMMPS, Quantum ESPRESSO, ABINIT, SIESTA, Octopus, GPAW, OpenMX, Elk, Elmer FEM, MOOSE
	Analysis	6	pymatgen, matminer, phonopy, Matscipy, ASE, MDAnalysis
	Visualization	6	OVITO, VMD, ParaView, VTK, VisIt, Mayavi
	File Handling	5	ASE, pymatgen, MDAnalysis, MDTraj, Chemfiles
	Libraries	5	pymatgen, Matscipy, OpenKIM, pycalphad, ASE
Biology	Simulation	6	NEURON, Brian, COPASI, OpenCOR, Smoldyn, MCell
	Analysis	9	BLAST, Bowtie, BWA, Biopython, BioPerl, BioJava, Bioconductor, Galaxy, OpenMS
	Visualization	4	Cytoscape, PyMOL, ChimeraX, Napari
	File Handling	4	Biopython, pysam, NetCDF, HTSeq
	Libraries	8	Biopython, BioPerl, BioJava, scikit-bio, pysam, Bioconductor, Bioconda, Cytoscape
Other common Quantum Simulations	Software Names	62	Gaussian, VASP, ORCA, CASTEP, Amber, Desmond, WIEN2k, NAMD, xTB, MOE, Discovery Studio, BoltzTrap, CHARMM, Wannier90, MOPAC, DMol3, ATK/QuantumATK, Molpro, GROMOS, GAMESS, ADF, TURBOMOLE, Q-Chem, YASARA, Dalton, MacroModel, TINKER, CRYSTAL, FoldX, Jaguar, EPW, RASPA, FHI-aims, FEFF, Hyperchem, GULP, HOOMD-blue, CPMD, CFOUR, FPLO, OpenMolcas, DIRAC, MOLCAS, Yambo, DL_POLY, PWmat, BerkeleyGW, GPUMD, ESPResSo, Firefly, TeraChem, DFTB+, JDFTx, ACEMD, exciting, FLEUR, QMCPACK, COLUMBUS, deMon2k, TB-LMTO-ASA, ONETEP, CASINO

Materials Safety Data Sheets Materials Safety Data Sheets (MSDS) are important resources that disclose the molecular and material safety. We included the tabular form of the MSDS in the language-interfaced tabular data, distinguishing between hazard statements (H) and precautionary statements (P). However, it is often important to describe safety in a more verbose manner. Hence, we converted the PDF version of the MSDS into natural text using the Nougat OCR model from Blecher et al. [87]

https://doi.org/10.52202/085713-1861

P.3.2 Post-processing of papers

We use a series of regular expression-based filters to remove references (both parenthetical and bracketed citations), figure and schema captions, email addresses. The core function uses year number patterns to identify and truncate citation sections, detecting where reference lists likely begin by finding clusters of publication years, and then cuts the text at the last complete sentence before this section begins. This cleaning process helps to extract the meaningful scientific content from the papers while removing formatting artifacts and reference materials.

P.4 ChemPile-Reasoning

P.4.1 Sources

Single-spectra to molecule reasoning traces We employed a multiple-choice question framework to generate synthetic reasoning paths for spectral interpretation using `Claude 3.5 Sonnet` with temperature set to one. Each question presented four candidate molecules alongside a unique spectrum, requiring the model to identify the correct molecular match through structural analysis.

We implemented structured output formatting to facilitate parsing and dataset construction for the single-spectra analysis. Specifically, we instructed the model to encapsulate its reasoning traces between the dedicated tokens [REASONING] and [\REASONING], which were subsequently extracted using regular expression pattern matching. Following the approach of Mirza et al. [5], we similarly prompted the model to enclose final answers between [SMILES] and [\SMILES] tokens for unambiguous identification.

The validity of predicted SMILES strings was rigorously verified through computational comparison with ground truth structures using `RDKit`'s molecular object representation. This validation ensured chemical equivalence by comparing molecular graph topologies rather than relying on string matching alone.

Multi-spectra to molecule reasoning traces We generated synthetic reasoning paths from spectral data employing a question-based prompting strategy with the `Deepseek-R1` model, with the temperature set to 0.6. Each prompt included carbon/proton NMR and IR spectra, supplemented with atomic counts, molecular formula, and molecular mass to compensate for the absence of mass spectrometry (MS) data. We evaluated two prompting formats:

1. Open-ended questions, requiring free-form generation of the correct SMILES string.
2. Multiple-choice questions (MCQs), where the model selected the correct answer from structural isomers of the target compound.

For the multi-spectra dataset, we maintained consistency by employing the same SMILES formatting protocol and evaluation methodology as in the single-spectra case. All outputs were parsed using identical regex patterns, with structural validity assessed through RDKit-based comparison against reference structures.

It is important to note that our evaluation criteria were primarily centered on answer accuracy, rather than compliance with formatting requirements. No assessment was conducted regarding the model's adherence to instructed output formatting guidelines.

The resulting multi-spectra datasets capture questions, step-by-step reasoning traces, final answers, and a boolean correctness label. The open-ended dataset comprises 358,000 tokens, while the MCQ variant contains 946,930 tokens.

P.5 ChemPile-Instruction

The ChemPile-Instruction dataset was generated through systematic rephrasing of the ChemPile-Education, ChemPile-Reasoning, and a 100-million-token subset of ChemPile-Paper datasets into multi-turn conversational formats. This transformation was executed using the `gpt-4o-mini-2024-07-18` large language model. The resulting dataset constitutes a substantial resource of over 200 million tokens of instruction-following data, structured into three distinct subsets corresponding to their respective source datasets.

Rephrasing was constrained through a predefined `Pydantic` schema that specified two mandatory elements: metadata tags and conversation structure. The schema enforced strict adherence to the `LiteLLM` format, where each conversational turn contains defined *role* and *content* fields. Valid roles are exclusively "user" or "assistant", while the content field contains the corresponding message text.

For the rephrasing, the used prompts are detailed below for each of the styles:

wiki

```
Use formal, encyclopedic English resembling Wikipedia.

Original text:

{text}

Now, rephrase this text into a multi-turn conversation about chemistry.
```

hard

```
Use esoteric vocabulary and complex syntax suitable for academics.

Original text:

{text}

Now, rephrase this text into a multi-turn conversation about chemistry.
```

engaging

```
Present information in a clear yet engaging manner, suitable for a broad
audience.

Original text:

{text}

Now, rephrase this text into a multi-turn conversation about chemistry.
```

no style defined

```
Original text:

{text}

Now, rephrase this text into a multi-turn conversation about chemistry.
```

Q Additional embedding visualizations

In Figure 8 we provide additional visualization for the embeddings in Figure 2b. These reinforce the main idea of Figure 2b, the ChemPile is by far the most diverse chemical dataset, capturing a large semantic space.

R SMILES-IUPAC translation

Interestingly, the generation of IUPAC names at scale is challenging due to the lack of open-source tools that can create IUPAC names based on SMILES. However, the validation can be robustly

https://doi.org/10.52202/085713-1861

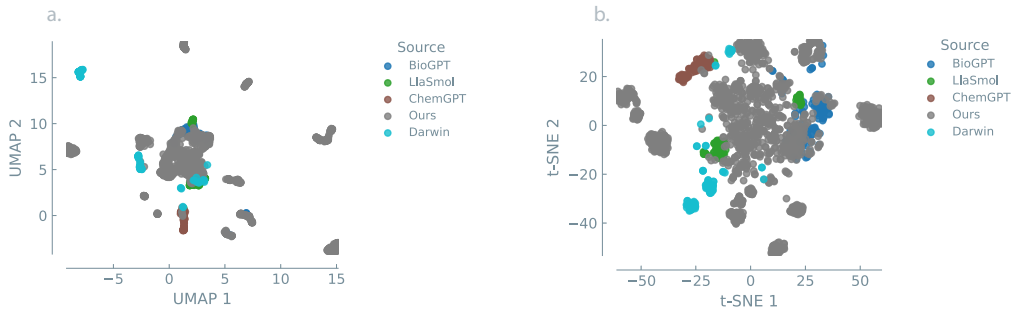


Figure 8: **Additional embedding dimensionality reduction visualizations for Figure 2b** We use the `umap-learn` package for UMAP and the `scikit-learn` package for TSNE. Default settings are used.

performed using the open-source IUPAC-to-SMILES converter OPSIN [88]. Thus, we trained a SMILES-to-IUPAC model based on an encoder-decoder architecture and automatically verified the validity of the outputs of the model using OPSIN.

Source SMILES sequences are tokenized using Byte-Level Byte Pair Encoding [89], while target IUPAC sequences utilize a Unigram tokenizer [90] with whitespace, punctuation, and digit-level pre-tokenization. The model architecture consists of 8 encoder and 8 decoder layers, an embedding dimension of 1536, 8 attention heads, a 4096-dimensional feed-forward network, and 0.1 dropout. Standard sinusoidal positional encodings are added to scaled token embeddings. The model has been trained for two epochs, with a final training loss of 0,006 and a validation loss of 0,0089. The model has an accuracy of approximately 91%. We provide an interface and script for the model on HuggingFace🤗.

S Code snipets

The following snippets show usage examples of the ChemPile.

S.1 Using subsets

ChemPile-Caption

```python
from datasets import load_dataset

dataset = load_dataset("jablonkagroup/chempile-caption")
print(dataset)
# DatasetDict({
#     train: Dataset({
#         features: ['text', 'image'],
#         num_rows: 90350
#     })
#     validation: Dataset({
#         features: ['text', 'image'],
#         num_rows: 5019
#     })
#     test: Dataset({
#         features: ['text', 'image'],
#         num_rows: 5020
#     })
# })

sample = dataset['train'][0]
```

```
print(f"Sample caption: {sample}")
# Sample caption: {'text': '2 drawings and a photograph, as described...',
  'image': <PIL...}
```

S.2 Mixing data

Obtaining pretraining data-mixes with ChemPile

```
from datasets import load_dataset, get_dataset_config_names,
  concatenate_datasets, Dataset
from typing import List

# --- Function to mix data in specified ratios ---
def mix_data_in_ratios(
    grouped_datasets_with_text: List[List[Dataset]],
    ratios: List[float],
    seed: int = 42
) -> Dataset:

    subsampled_data_for_final_mix = []
    for i, group_list in enumerate(grouped_datasets_with_text):
        category_ds = concatenate_datasets(group_list)

        num_samples_to_take = int(len(category_ds) * ratios[i])

        if num_samples_to_take > 0:
            selected_subset =
            category_ds.shuffle(seed=seed).select(range(num_samples_to_take))
            subsampled_data_for_final_mix.append(selected_subset)

    final_mixed_dataset = concatenate_datasets(subsampled_data_for_final_mix)
    return final_mixed_dataset

# --- Main script logic for loading, preparing, and mixing ---
def create_mixed_dataset(category_sources_with_ratios, split):

    dataset_groups_for_mixing = []
    final_ratios_for_mixing = []

    for _, path, ratio in category_sources_with_ratios:
        configs = get_dataset_config_names(path)
        raw_sub_datasets = [load_dataset(path, config,
        trust_remote_code=True)[split] for config in configs]
        sub_datasets_for_category = []
        for ds in raw_sub_datasets:
            if "text" in ds.column_names and len(ds) > 0:
                sub_datasets_for_category.append(ds.select_columns(["text"]))

        if sub_datasets_for_category:
            dataset_groups_for_mixing.append(sub_datasets_for_category)
            final_ratios_for_mixing.append(ratio)

    # Call the mixing function if data is available
    mixed_dataset = mix_data_in_ratios(
        dataset_groups_for_mixing,
        final_ratios_for_mixing,
        seed=42, # for reproducibility
    )
    return mixed_dataset


```


https://doi.org/10.52202/085713-1861

```python
category_sources_with_ratios = [
    ("education", "jablonkagroup/chempile-education", 1.0),
    ("paper", "jablonkagroup/chempile-paper", 1.0),
    ("code", "jablonkagroup/chempile-code", 0.1)
]
resulting_mixed_dataset = create_mixed_dataset(category_sources_with_ratios,
    split="train")
```

T Data availability

The scripts used for the review app, the script to compute token counts, but also the training and evaluation scripts are available at `https://github.com/lamalab-org/chempile-scripts`. Here we also include the scripts used to generate the ChemPile-Instruct dataset.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We believe this manuscript accurately reflects the described contributions, and introduces solid arguments about the novelty of the used approach for creating foundation model scale datasets for the chemical sciences. We show quantified and/or qualitative evidence for the scale, diversity, and curation claims we make.

   Guidelines: [NA]

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We describe the limitations of our work in the "Future work" section, where we list potential improvements.

   Guidelines: [NA]

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: [NA]

   Guidelines: [NA]

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper describes all the methods that have been used to generate / create the data.

   Guidelines: [NA]

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide full access to the dataset on HuggingFace as a collection of the subsets described fully in this paper, as the main artifact of this paper.

   Guidelines: [NA]

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [NA]

   Justification: [NA]

   Guidelines: [NA]

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

https://doi.org/10.52202/085713-1861

Answer: [Yes]

Justification: For Figure 3 we provide a significance test in Appendix O. Otherwise, no experiments were conducted.

Guidelines: [NA]

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The paper's focus is a new dataset and not empirical experiments.

   Guidelines:

   -

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The datasets do not involve human subjects or obviously ethically sensitive applications. While chemical data and models can have broad impacts we assume the dataset we report conforms to the NeurIPS Code of Ethics.

   Guidelines:

   -

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: While advances in chemical research can have broad implications we do not expect immediate societal impacts from the release of our dataset.

    Guidelines:

    -

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: Our dataset is publicly available under a permissive, non-commercial license. While there is potential of dual use for chemistry data (but also for any major dataset), the current risk associated with the ChemPile is minimal as most dual risk pathways are still constrained by other safety measures.

    Guidelines:

    - We suggest careful use of the dataset, and the avoidance of training any future model based on this data on additional harmful data.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The ChemPile contains metainformation and original licenses are respected.

    Guidelines: [NA]

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide code-snippets in the Appendix section of this paper and a web-page describing the data (alongside with the snippets). All LIFT datasets contain metadata about the datasets.

    Guidelines:

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: [NA]

    Guidelines: [NA]

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: [NA]

    -

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We use LLMs for generating / modifying two of the datasets: used for generating reasoning traces for spectra elucidation, and for rewriting YouTube transcripts in lecture-like format.

https://doi.org/10.52202/085713-1861