MiST: Understanding the Role of Mid-Stage Scientific Training in Developing Chemical Reasoning Models

Andres M Bran[*] [1 2] **Tong Xie**[*] [3 4] **Shai Pranesh**[*] [1 2 5] **Jeremy Goumaz** [1] **Xuan Vu Nguyen** [1] **David Ming Segura** [1]
Ruizhi Xu [3] **Jeffrey Meng** [3] **Dongzhan Zhou** [6] **Wenjie Zhang** [3] **Philippe Schwaller** [1 2]

Abstract

Large Language Models (LLMs) can acquire emergent reasoning via online fine-tuning with simple rule-based rewards, when tasks are already latent-solvable by the base model. We study chemical reasoning and identify two prerequisites for RL-based training: (1) symbolic competence and (2) latent domain knowledge. We introduce MiST, a mid-stage training protocol using SMILES-aware data mix and continued pretraining on 2.9 B tokens, and supervised finetuning on chain-of-thoughts reasoning data. MiST doubles the latent-solvability score of IUPAC-to-SMILES translation and increases the precision of reaction prediction from 4.1% to 25.2%, while producing faithful reasoning traces. Our work defines clear prerequisites for chemical reasoning and underscores the value of mid-stage pretraining.

1. Introduction

Reasoning tasks in chemistry are fundamental yet notoriously challenging, requiring models to integrate chemical knowledge and logical deduction (Coley et al., 2019; Alampara et al., 2024). Traditional cheminformatics methods rely heavily on supervised architectures optimized for specific tasks, lacking generalization and human-like reasoning (Schwaller et al., 2019; Mirza et al., 2024a). Recent reinforcement learning (RL) frameworks (Guo et al., 2025b) have shown emergent reasoning capabilities in domains like math and coding. However, independent studies indicate these capabilities arise from amplified patterns already present, albeit with low likelihood, in the base model (Guo et al., 2023; Flam-Shepherd & Aspuru-Guzik, 2023), suggesting RL's effectiveness depends critically on latent task solvability within the model.

Chemistry presents a stringent test for this hypothesis due to specialized symbol systems (e.g., SMILES, IUPAC) and domain-specific constraints (Weininger, 1988). Standard LLMs often fail at generating syntactically valid chemical strings, limiting RL success (Bran et al., 2025). Empirically, direct RL application fails as correct outputs rarely appear in candidate distributions.

This raises a fundamental question: What pre-training prerequisites must an LLM satisfy for RL to reliably enhance chemical reasoning? We address this by: 1) proposing quantitative diagnostics to measure latent solvability, 2) introducing and ablating two domain-specific prerequisites, and 3) demonstrating RL improvements when diagnostics surpass defined thresholds.

We propose symbolic competence and latent chemical knowledge as essential prerequisites. Symbolic competence requires the model to handle valid chemical strings (SMILES, IUPAC, CIF). Latent chemical knowledge ensures correct answers exist in the model's prior distribution, enabling RL exploitation. We develop a diagnostic benchmark for latent solvability, confirming that enhancing these prerequisites boosts RL performance by up to 20

Additionally, we propose representative chemistry reasoning tasks solvable by expert humans (Section C). Ablation and generalization tests validate that removing any prerequisite collapses RL gains, confirming their necessity. We release our diagnostic benchmark and pre-training corpus, offering a foundation for developing robust chemical reasoning AI.

2. Related Work

Post-training methods for reasoning Standard alignment methods combine supervised fine-tuning (SFT) with reinforcement learning from human or synthetic feedback (RLHF/RLAIF), which improves helpfulness but often struggles with multi-step reasoning. Recent strategies include

---

[*] Equal contribution [1] École Polytechnique Fédérale de Lausanne [2] National Centre of Competence in Research (NCCR) Catalysis [3] University of New South Wales [4] Green Dynamics [5] BITS Pilani - Pilani Campus [6] Shanghai Artificial Intelligence Laboratory. Correspondence to: Tong Xie <tong.xie@unsw.edu.au>, Philippe Schwaller <philippe.schwaller@epfl.ch>.

Workshop on Multi-modal Foundation Models and Large Language Models for Life Sciences at ICML 2025. Copyright 2025 by the author(s).

chain-of-thought distillation (Wei et al., 2022; Li et al., 2023), step-aware reward models (Weng et al., 2025), and tree search with self-consistency (Xie et al., 2024b). Crucially, Guo et al. (2025b) demonstrated that rule-based rewards enhance mathematical and coding capabilities if the base model already places non-negligible probability on correct solutions. Independent studies confirm RL amplifies latent solutions in the base model (Yue et al., 2025), and weaker bases benefit more from SFT using larger-model-generated traces (Guo et al., 2025b). Our work adopts this ”RL as amplifier” perspective to explore pre-training conditions enabling latent chemical problem-solving.

Chemical language modeling Language models adapted to chemistry tasks often use linear molecular strings (SMILES (Weininger, 1988), SELFIES (Krenn et al., 2020), IUPAC). Masked pre-training approaches like ChemBERTa (Chithrananda et al., 2020) and MolBERT (Fabian et al., 2020) enhance QSAR performance, while Molecular Transformers target synthesis prediction (Schwaller et al., 2019; 2020). Recent works extend general LLMs to molecule generation, property prediction, and Q&A (Frey et al., 2023; Zhang et al., 2024; Jablonka et al., 2024; Xie et al., 2023b), as well as integrating robotic labs and hypothesis generation (Bran et al., 2023; Boiko et al., 2023; Yang et al., 2025). New workflows address molecular design and synthesis planning (Wang et al., 2024a; Bran et al., 2025).

Mid-stage domain adaptation Domain-adaptive pre-training (DAPT) or continued pre-training (CPT) effectively specializes general LLMs. Early successes include BioMegatron (Shin et al., 2020), Legal-BERT (Chalkidis et al., 2020), and Code-Llama (Rozière et al., 2023). Recent scientific domain studies confirm substantial gains: AdaptLLM in finance (Cheng et al., 2023), Tag-LLM and Efficient-CPT with adapters (Shen et al., 2024; Xie et al., 2024a), SciLitLLM in scientific literature (Li et al., 2024a), and adaptations in materials science, radiation oncology, finance, and cybersecurity (Lu et al., 2025; Holmes et al., 2023; Hirano & Imajo, 2024; Bayer et al., 2024). Caveats include potential degradation of zero-shot prompting (Cheng et al., 2023) and limited capability gains for models ¡$2B parameters (Lu et al., 2025; Hsieh, 2025). Crucially, prior studies did not evaluate whether CPT induces latent solvability exploitable by RL. Chemical domain CPT studies like ChemBERTa-2 (Maziarka et al., 2023), ChemLLM (Brand et al., 2023), DARWIN-Chem (Xie et al., 2023a), and SciDFM (Sun et al., 2024) remain limited to single-shot recognition tasks.

LLM capability diagnostics Traditional benchmarks like accuracy and perplexity omit detailed insights from conditional probability distributions. Holistic evaluations (HELM (Liang et al., 2022), LiveBench (White et al., 2024)) still aggregate probabilities into singular metrics. Intrinsic probes provide deeper insights: minimal pairs measure grammatical preference gaps (Meister & Cotterell, 2021), brittleness of in-context learning (Zhao et al., 2024), and out-of-domain intent detection (Wang et al., 2024b). Calibration studies reveal token-level probabilities reflect model certainty (Jiang et al., 2021; Kadavath et al., 2022), now underpinning OOD detection, self-correction, and medical reasoning (Liu et al., 2024a;b; Li et al., 2024b). Diagnostics increasingly use distribution-matching (KL divergence, Wasserstein distance (Pezeshkpour, 2023; Wang et al., 2024c)) and dispersion metrics linked to robustness (Ye et al., 2024).

3. MiST: Mid-stage Scientific Training

The purpose of this mid-stage training is to enhance the model’s ability to generate valid SMILES, accurately follow chemistry-focused instructions, and strengthen its general chemical knowledge. We do this by continuing pretraining (next token prediction objective) on chemical and SMILES-related data, and then by performing SFT to better follow instructions and increase the thinking context window.

3.1. Datasets

The FineWeb chemistry dataset was filtered from FineWeb-Edu ((Penedo et al., 2024)) using a custom non-ML classifier built using word frequency. The entire FineWeb-Edu dataset was fetched, and about 10,000 texts were manually labeled as chemistry and 50,000 as non-chemistry (based on the text source). These texts were lemmatized before building word frequency vectors for the two classes. The frequencies of the lemma k in chemistry texts and non-chemistry texts are denoted f_k^c and f_k^n, respectively. The text chemistry score (TCS) is computed using the following formula:

$$\text{TCS(text)} := \frac{1}{N_{\text{lemmas}}} \sum_{k \in \text{lemmas in text}} w_k, \quad w_k = \begin{cases} \frac{f_k^c}{f_k^n}, & \text{if } \frac{f_k^c}{f_k^n} > 1 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

This labeling strategy was applied to the entire FineWeb-Edu corpus, and the texts with $TCS > 4$ were retained, yielding a pretraining set of 1.4 billion tokens of high-quality chemistry-labeled texts.

The first three million compounds from the PubChem database ((Kim et al., 2025)) were dumped and filtered using the following pipeline: the compounds with ambiguous SMILES (different RDKit canonical SMILES from the IUPAC, InChI (Heller et al., 2015), or PubChem SMILES) were discarded, and the duplicates (for SMILES and IUPAC) were filtered out. Four SMILES variants (non-canonical SMILES) were generated from the canonical SMILES for each valid compound. Based on this strategy, the first million compounds from PubChem were filtered to around 600,000 compounds. The same approach was applied to the rest of the compounds, and the dataset was split in the following manner: the first million compounds (CID from 1 to 1,000,000) were used for pretraining, the second million compounds (CID from 1,000,001 to 2,000,000) were used for GRPO training, and the third million compounds (CID from 2,000,001 to 3,000,000) were used as the test split. Multiple derived datasets were also generated for the different chemical tasks used with GRPO training.

To construct the pretraining data, we used the data mixture as described in Table 1. All the data underwent the same preprocessing pipeline to interleave SMILES with text whenever a molecule name appeared (e.g. IUPAC, common name, short form, etc), this type of interleaved data was also used in (Taylor et al., 2022). We additionally generated a synthetic dataset using RDkit (RDKit, online) extracted properties of molecules (like QED, TPSA, etc) and filled it in a template. Furthermore, we include a ”replay” dataset aiming to preserve the model’s natural language abilities while furthering it’s learning about chemical knowledge. We chose the Qwen2.5-3B base model to perform the pretraining for 3 epochs.

Table 1. Dataset composition and token distribution used for the pretraining step.

Dataset Source	Tokens	Percentage
ChemRxiv + S2ORC	1.2B	41.38%
FineWeb chemistry filtered data	1.4B	48.28%
PubChem synthetic data (600k compounds)	220M	7.59%
CommonCrawl Replay dataset	80M	2.75%
Total	**2.9B**	**100%**

For SFT, we utilized question-answering (QA) training examples derived from SmolInstruct (Yu et al., 2024a), specifically employing only the SMILES↔IUPAC and molecule captioning subsets. We also collect examples from MPtrj dataset(Deng et al., 2023). Additionally, we incorporated MMLU and chain-of-thought (CoT) reasoning traces from DeepSeek-R1, which were preprocessed to maintain coherence with our pretraining data. In this phase, we also expanded the model’s context window from 4,096 to 8,192 tokens to accommodate longer reasoning sequences. The pretrained model underwent SFT for approximately 8 epochs, continuing until the previously observed loss spikes were fully mitigated. During fine-tuning, two distinct question types were used:

∗ Questions requiring explicit reasoning traces, with solutions prefixed by the tag ”<think>”.
∗ Questions directly presenting the final answers, prefixed by the tag ”<answer>”.

Table 2. Instruction tuning dataset composition used for the SFT step.

Dataset Source	Notes / Sample Count
DeepSeek reaction traces	∼7K samples
DeepSeek relaxation traces	∼2K samples
MPtrj dataset	∼20K samples
SmolInstruct dataset	I2S, S2I, Molecule captioning and generation tasks
MMLU	Train: ∼350 samples, Chemistry: ∼300 samples
CoT Chain	∼27K samples
Total Tokens	**1B**

The model, despite using about 3B tokens for continued pretraining and 1B tokens for SFT, performs better on some tasks in comparison with models like NatureLM (Xia et al., 2025), which has used hundreds of billions of tokens for pretraining and SFT. This was made possible by the high-quality interleaved text produced by our preprocessing pipeline.

More details about our SMILES-text preprocessing pipeline can be found in Appendix B.

4. Post-training Experiments

This section quantifies how much of the potential unlocked by Mid-stage Scientific Training (MiST) can actually be surfaced with standard post-training recipes. We therefore keep the mid-training configuration fixed (Section 3) and vary the post-training stack:

1. BASE: original Qwen2.5-3B
2. +MIST: after MiST continued pre-training (checkpoint `v6-1`)
3. +MIST+SFT: MiST backbone after SFT on 120k DeepSeek-R1 traces (see Section A.3).
4. +MIST+SFT+RL(TASK i): previous model further optimized with RLVR (see Section A.3). (TASK i) specifies the single task the model is trained on with RLVR.

As an initial downstream test of our pipeline’s performance, we use ChemBench (Mirza et al., 2024b) to evaluate the general chemistry knowledge of LLMs; the results are shown in Table 3.

The results shown in Table 3 indicate that the proposed MiST combined with reasoning supervised fine-tuning (SFT) significantly enhances downstream performance on general chemistry knowledge across most chemistry subdomains. Particularly notable improvements (up to 6-7%) were observed in Organic, Inorganic, and General Chemistry compared to the Qwen+SFT baseline, and improvements exceeding 11% over the instruction-tuned base model, highlighting the efficacy of both post-training stages. These results serve as diagnostic indicators for assessing the success of mid-training methods and inform model selection for

Table 3. ChemBench sub-domain Accuracy (%)

	Models		
Sub-domain	Qwen-2.5-3B Instruct	Qwen+SFT	MiST+SFT (ours)
Organic Chem	44.99	46.15	**50.12**
Inorganic Chem	46.70	51.08	**57.60**
Toxicity/Safety	21.33	**26.52**	26.37
Material Sci	35.84	42.50	**48.75**
General Chem	33.56	38.25	**44.30**
Chem Preference	45.40	50.00	**52.10**
Analytical Chem	25.00	34.20	**40.70**
Technical Chem	42.11	44.74	**50.00**
Physical Chem	20.60	35.10	**38.78**
Total	35.06	40.95	**45.41**

Table 4. Effect of MiST and each post-training stage on downstream reasoning tasks. SCS = symbolic-competence score, CCS = chemical-competence score; both are unitless effect-size measures ranging from 0 (no separation) to 2 (near-perfect separation); higher is better. I2S = IUPAC→SMILES translation, RxP = forward reaction prediction, RxN = reaction-naming, CMG = conditional material generation. For the three downstream tasks we report top-1 accuracy; values in parentheses are with CoT prompting.

Model	SCS↑	CCS↑	I2S↑	RxP↑	RxN↑	CMG↑
Qwen-2.5 3B	0.95	0.352	0.03	0.6	10.33 (10.87)	58.6
+MiST	1.639	0.443	49.12	4.1	12.8 (11.30)	1.2
+SFT	1.906	0.771	68.2 (34.5)	8.20 (21.2)	11.9 (11.0)	34.8
OrgChem Tasks						
+RL(I2S)	1.825	0.759	**68.39 (67.5)**	—	—	—
+RL(RxP)	1.880	0.782	—	**8.80 (25.2)**	—	—
+RL(RxN)	1.906	0.789	—	—	**22.87 (35.17)**	—
MatSci Tasks						
+RL(CMG)	0.893	0.777	—	—	—	**73.8**
Ablations						
no MiST + SFT	1.853	0.788	22.00	5.10	2.6 (4.80)	—

subsequent RLVR training aimed at strengthening reasoning and problem-solving skills.

Subsequently, the model's capability to learn online through Reinforcement Learning with Verifiable Rewards (RLVR) is evaluated. As detailed in Section C, various chemistry tasks amenable to reasoning and reward verification were implemented. Several models trained in this RLVR framework were evaluated based on their base model origin. Two distinct inference strategies—System-1 (direct answer) and System-2 (reasoning-based), as characterized by (McGlynn, 2014)—were tested by appending ”<answer>” or ”¡reasoning¿” tags, respectively, to guide model responses. Table 22 summarizes the outcomes on the defined chemical tasks (Section C) and the diagnostic metrics described in Section A.1.

The results highlight the significant impact of the MiST proposed here on symbolic competence, as evidenced by the SCS column. Pretrained models like Qwen2.5-3B clearly lack symbolic capabilities essential for tasks involving SMILES comprehension and generation; however, this limitation is effectively overcome through MiST. Additionally, results demonstrate that reinforcement learning (RL) generally enhances the performance of LLMs on chemical tasks, especially those involving SMILES synthesis, such as *Reaction Prediction* and *IUPAC2SMILES*.

An important observation is that activating reasoning usually yields better performance in RL-trained LLMs; yet, this trend reverses in specific cases, notably the *IUPAC2SMILES* task. In this scenario, better results are achieved without activating reasoning, although the performance gap is narrower for RL-trained models. This phenomenon is attributed to the symbolic ability already being strongly developed during supervised fine-tuning (SFT), potentially hindering RL-driven learning of alternative strategies. Further research in this direction is recommended.

5. Discussion

This paper aims to answer the practical question of what conditions a general-purpose large language model (LLM) must fulfill to reliably perform chemical reasoning through lightweight, rule-based post-training methods (Supervised Fine-Tuning (SFT) + Reinforcement Learning via Rewards (RLVR)). Experiments were conducted using the Qwen2.5-3B model, demonstrating that the proposed Mid-stage Scientific Training (MiST) is necessary to unlock chemical reasoning capabilities in LLMs.

Results indicate that MiST significantly enhances both symbolic competence and latent chemical knowledge, leading to substantial improvements in downstream tasks involving SFT and RLVR. Reinforcement Learning (RL), consistent with prior findings, acts primarily as an amplifier of existing knowledge and behaviors in LLMs rather than introducing entirely new capabilities.

Critically, symbolic competence emerges as the main bottleneck for small-scale LLMs, particularly evident in tasks heavily reliant on symbolic systems like SMILES notation. For example, performance on *Reaction Prediction* and *IUPAC to SMILES* tasks is near 0% accuracy in base models. While SFT marginally improves *Reaction Prediction* accuracy to 5.10%, incorporating MiST significantly boosts accuracy to 25.2% when reasoning processes are activated. Moreover, improvements in untrained materials science tasks (e.g., the CMG task) underline the broader utility of MiST for scientific tasks.

6. Limitations

While MiST shows targeted mid-stage pre-training can unlock chemical reasoning in a 3B-parameter model, several caveats remain. First, we only probed one backbone size; other sizes might differ in symbolic–competence thresholds, limiting extrapolation. Second, RLVR rewards emphasize syntactic agreement (e.g. exact SMILES or high

Tanimoto similarity), potentially allowing chemically implausible or unsafe outputs via reward hacking. Third, our evaluation suite—reaction prediction, IUPAC to SMILES, and conditional material generation—is narrow, omitting tasks involving stereochemistry, kinetics, spectroscopy, or 3D conformations. Finally, our pre-training corpus, dominated by small-molecule organic literature and patents, may bias the model against inorganic, macromolecular, or biochemical domains. Resolving these scale, reward, coverage, and data-bias issues is essential for MiST-style models to serve as reliable scientific assistants.

Acknowledgments

AMB, SP, and PS acknowledge support from the NCCR Catalysis (grant number 225147), a National Centre of Competence in Research funded by the Swiss National Science Foundation. This work was supported as part of the Swiss AI Initiative by a grant from the Swiss National Supercomputing Centre (CSCS) under project ID a05 on Alps.TX acknowledges the support provided by the Pawsey Supercomputing Research Centre,funded by the Australian Government and the Government of Western Australia. Additional support was received from Microsoft's Accelerate Foundation Model Research Project and the Katana computational cluster at UNSW Sydney. This work was also supported by the Australian Renewable Energy Agency (ARENA) through the Australian Centre for Advanced Photovoltaics (ACAP).Responsibility for the views, information or advice expressed herein is not accepted by the Australian Government.

References

Alampara, N., Schilling-Wilhelmi, M., Ríos-García, M., Mandal, I., Khetarpal, P., Grover, H. S., Krishnan, N., and Jablonka, K. M. Probing the limitations of multimodal language models for chemistry and materials research. *arXiv preprint arXiv:2411.16955*, 2024.

Bayer, M., Kuehn, P. D., Shanehsaz, R., and Reuter, C. A. CySecBERT: A domain-adapted language model for the cybersecurity domain. *ACM Transactions on Privacy and Security*, 2024.

Blecher, L., Cucurull, G., Scialom, T., and Stojnic, R. Nougat: Neural optical understanding for academic documents, 2023. URL `https://arxiv.org/abs/2308.13418`.

Boiko, D. A., MacKnight, R., Kline, B., and Gomes, G. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Bran, A. M., Cox, S., Schilter, O., Baldassari, C., White, A. D., and Schwaller, P. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6:525 – 535, 2023. URL `https://api.semanticscholar.org/CorpusID:258059792`.

Bran, A. M., Neukomm, T. A., Armstrong, D. P., Jončev, Z., and Schwaller, P. Chemical reasoning in llms unlocks steerable synthesis planning and reaction mechanism elucidation, 2025. URL `https://arxiv.org/abs/2503.08537`.

Brand, U. N., Du, Z., Taheri, A., and Schwaller, P. ChemLLM: A large language model for chemistry. arXiv preprint arXiv:2310.01890, 2023.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., and Androutsopoulos, I. LEGAL-BERT: The Muppets straight out of law school. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Cheng, D., Huang, S., and Wei, F. Adapting large language models via reading comprehension. *arXiv preprint arXiv:2309.09117*, 2023.

Chithrananda, S., Grand, G., and Ramsundar, B. Chemberta: Large-scale self-supervised pretraining for molecular property prediction. *Preprint at https://arxiv.org/abs/2010.09885*, 2020.

Coley, C. W., Barzilay, R., Jaakkola, T. S., Green, W. H., and Jensen, K. F. Prediction of Organic Reaction Outcomes Using Machine Learning. *ACS Central Science*, 3 (5):434–443, May 2017. ISSN 2374-7943. doi: 10.1021/acscentsci.7b00064. URL `https://doi.org/10.1021/acscentsci.7b00064`. Publisher: American Chemical Society.

Coley, C. W., Jin, W., Rogers, L., Jamison, T. F., Jaakkola, T. S., Green, W. H., Barzilay, R., and Jensen, K. F. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem. Sci.*, 10:370–377, 2019.

Davies, D., Butler, K., Jackson, A., Morris, A., Frost, J., Skelton, J., and Walsh, A. Computational screening of all stoichiometric inorganic materials. *Chem*, 1(4):617–627, 2016. ISSN 2451-9294. doi: https://doi.org/10.1016/j.chempr.2016.09.010. URL `https://www.sciencedirect.com/science/article/pii/S2451929416301553`.

Deng, B. Materials Project Trajectory (MPtrj) Dataset. *arXiv preprint arXiv:2302.14231*, 7 2023. doi: 10.6084/m9.figshare.23713842.v2. URL `https://figshare.com/articles/dataset/Materials_Project_Trjectory_MPtrj_Dataset/23713842`.

Deng, B., Zhong, P., Jun, K., Riebesell, J., Han, K., Bartel, C. J., and Ceder, G. Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, 5(9):1031–1041, 2023.

Fabian, B., Edlich, T., Gaspar, H., Segler, M. H. S., Meyers, J., Fiscato, M., and Ahmed, M. Molecular representation learning with language models and domain-relevant auxiliary tasks. *ArXiv*, abs/2011.13230, 2020. URL `https://api.semanticscholar.org/CorpusID:227209142`.

Flam-Shepherd, D. and Aspuru-Guzik, A. Language models can generate molecules, materials, and protein binding sites directly in three dimensions as xyz, cif, and pdb files, 2023.

Frey, N. C., Soklaski, R., Axelrod, S., Samsi, S., G´omez-Bombarelli, R., Coley, C. W., and Gadepally, V. Neural scaling of deep chemical models. *Nature Machine Intelligence*, 5:1297 – 1305, 2023. URL `https://api.semanticscholar.org/CorpusID:262152780`.

Ganose, A. M. and Jain, A. Robocrystallographer: automated crystal structure text descriptions and analysis. *MRS Communications*, 9(3):874–881, 2019.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., Liu, A., Xue, B., Wang, B., Wu, B., Feng, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Ding, H., Xin, H., Gao, H., Qu, H., Li, H., Guo, J., Li, J., Wang, J., Chen, J., Yuan, J., Qiu, J., Li, J., Cai, J. L., Ni, J., Liang, J., Chen, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Zhao, L., Wang, L., Zhang, L., Xu, L., Xia, L., Zhang, M., Zhang, M., Tang, M., Li, M., Wang, M., Li, M., Tian, N., Huang, P., Zhang, P., Wang, Q., Chen, Q., Du, Q., Ge, R., Zhang, R., Pan, R., Wang, R., Chen, R. J., Jin, R. L., Chen, R., Lu, S., Zhou, S., Chen, S., Ye, S., Wang, S., Yu, S., Zhou, S., Pan, S., Li, S. S., Zhou, S., Wu, S., Ye, S., Yun, T., Pei, T., Sun, T., Wang, T., Zeng, W., Zhao, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Xiao, W. L., An, W., Liu, X., Wang, X., Chen, X., Nie, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yang, X., Li, X., Su, X., Lin, X., Li, X. Q., Jin, X., Shen, X., Chen, X., Sun, X., Wang, X., Song, X., Zhou, X., Wang, X., Shan, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhang, Y., Xu, Y., Li, Y., Zhao, Y., Sun, Y., Wang, Y., Yu, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Ou, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Xiong, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Zhu, Y. X., Xu, Y., Huang, Y., Li, Y., Zheng, Y., Zhu, Y., Ma, Y., Tang, Y., Zha, Y., Yan, Y., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Xie, Z., Zhang, Z., Hao, Z., Ma, Z., Yan, Z., Wu, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Pan, Z., Huang, Z., Xu, Z., Zhang, Z., and Zhang, Z. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025a. URL `https://arxiv.org/abs/2501.12948`.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025b.

Guo, T., Nan, B., Liang, Z., Guo, Z., Chawla, N., Wiest, O., Zhang, X., et al. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in Neural Information Processing Systems*, 36: 59662–59688, 2023.

Heller, S. R., McNaught, A., Pletnev, I., Stein, S., and Tchekhovskoi, D. Inchi, the iupac international chemical identifier. *J. Cheminf.*, 7(1):1–34, 2015.

Hirano, M. and Imajo, K. Construction of domain-specified japanese large language model for finance through continual pre-training. In *16th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, 2024.

Holmes, J., Liu, Z., Zhang, L., Ding, Y., Sio, T. T., McGee, L. A., Ashman, J. B., et al. Evaluating large language models on a highly-specialized topic, radiation oncology physics. *Frontiers in Oncology*, 2023.

Hsieh, S.-K. Continual pre-training is (not) what you need in domain adaption. *arXiv preprint arXiv:2501.01234*, 2025.

Jablonka, K. M., Schwaller, P., Ortega-Guerrero, A., and Smit, B. Leveraging large language models for predictive chemistry. *Nat. Mac. Intell.*, 6:161–169, 2024. URL `https://api.semanticscholar.org/CorpusID:267538205`.

Jacobsson, T. J., Hultqvist, A., García-Fernández, A., Anand, A., Al-Ashouri, A., Hagfeldt, A., Crovetto, A., Abate, A., Ricciardulli, A. G., Vijayan, A., Kulkarni, A., Anderson, A. Y., Darwich, B. P., Yang, B., Coles, B. L., Perini, C. A. R., Rehermann, C., Ramirez, D., Fairen-Jimenez, D., Di Girolamo, D., Jia, D., Avila, E., Juarez-Perez, E. J., Baumann, F., Mathies, F., Anaya González, G. S., Boschloo, G., Nasti, G., Paramasivam, G., Martínez-Denegri, G., Näsström, H., Michaels, H., Köbler, H., Wu, H., Benesperi, I., Ibrahim Dar, M., Bayrak Pehlivan, I., Gould, I. E., Vagott, J. N., Dagar, J., Kettle, J., Yang, J., Li, J., Smith, J. A., Pascual, J., Jerónimo-Rendón, J. J., Montoya, J. F., Correa-Baena, J.-P., Qiu, J., Wang, J., Sveinbjörnsson, K., Hirselandt, K., Dey, K., Frohna, K., Mathies, L., Castriotta, L. A., Aldamasy, M. H., Vasquez-Montoya, M., Ruiz-Preciado, M. A., Flatken, M. A., Khenkin, M. V., Grischek, M., Kedia, M., Saliba, M., Anaya, M., Veldhoen, M., Arora, N., Shargaieva, O., Maus, O., Game, O. S., Yudilevich, O., Fassl, P., Zhou, Q., Betancur, R., Munir, R., Patidar, R., Stranks, S. D., Alam, S., Kar, S., Unold, T., Abzieher, T., Edvinsson, T., David, T. W., Paetzold, U. W., Zia, W., Fu, W., Zuo, W., Schröder, V. R. F., Tress, W., Zhang, X., Chiang, Y.-H., Iqbal, Z., Xie, Z., and Unger, E. An open-access database and analysis tool for perovskite solar cells based on the fair data principles. *Nature Energy*, 7:107–115, 2022. doi: 10.1038/s41560-021-00941-3. URL `https://doi.org/10.1038/s41560-021-00941-3`.

Jain, A., Ong, S. P., Hautier, G., Chen, W., Richards, W. D., Dacek, S., Cholia, S., Gunter, D., Skinner, D., Ceder, G., and Persson, K. A. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1):011002, 07 2013. ISSN 2166-532X. doi: 10.1063/1.4812323. URL `https://doi.org/10.1063/1.4812323`.

Jiang, Z., Araki, J., Ding, H., and Neubig, G. How can we know when language models know? on the calibration of language models for question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4661–4673, 2021.

Kadavath, S., Conerly, T., Askell, A., Bai, Y., Ganguli, D., Hernandez, D., Schiefer, N., et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B., Thiessen, P., Yu, B., Zaslavsky, L., Zhang, J., and Bolton, E. PubChem 2025 update. *Nucleic Acids Research*, 53(D1):D1516–D1525, January 2025. ISSN 1362-4962. doi: 10.1093/nar/gkae1059. URL `https://doi.org/10.1093/nar/gkae1059`.

Krenn, M., Häse, F., Nigam, A., Friederich, P., and Aspuru-Guzik, A. Self-Referencing Embedded Strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn.: Sci. Technol.*, 1:045024, 2020.

Li, L. H., Hessel, J., Yu, Y., Ren, X., Chang, K.-W., and Choi, Y. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In *Annual Meeting of the Association for Computational Linguistics*, 2023. URL `https://api.semanticscholar.org/CorpusID:259251773`.

Li, S., Huang, J., Zhuang, J., Shi, Y., Cai, X., Xu, M., Wang, X., Zhang, L., and Ke, G. SciLitLLM: How to adapt LLMs for scientific literature understanding. *arXiv preprint arXiv:2408.04567*, 2024a.

Li, S. S., Balachandran, V., Feng, S., Ilgen, J. S., Pierson, E., Koh, P. W., and Tsvetkov, Y. Mediq: Question-asking llms and a benchmark for reliable interactive clinical reasoning. In *Neural Information Processing Systems*, 2024b. URL `https://api.semanticscholar.org/CorpusID:270219405`.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Liu, B., Zhan, L., Lu, Z., Feng, Y., Xue, L., and Wu, X.-M. How good are llms at out-of-distribution detection?, 2024a. URL `https://arxiv.org/abs/2308.10261`.

Liu, D., Nassereldine, A., Yang, Z., Xu, C., Hu, Y., Li, J., Kumar, U., Lee, C., and Xiong, J. Large language models have intrinsic self-correction ability. *ArXiv*, abs/2406.15673, 2024b. URL `https://api.semanticscholar.org/CorpusID:270703467`.

Lowe, D. M., Corbett, P. T., Murray-Rust, P., and Glen, R. C. Chemical name to structure: Opsin, an open source solution. *J. Chem. Inf. Model.*, 51(3):739–753, 2011. doi: 10.1021/ci100384d.

Lu, W., Luu, R. K., and Buehler, M. J. Fine-tuning large language models for domain adaptation: Exploration of training strategies, scaling, model merging and synergistic capabilities. *npj Computational Materials*, 2025.

Maziarka, , Rataj, K., Danel, T., Warchoł, P., and Jastrzebski, S. ChemBERTa-2: Large-scale self-supervised pretraining for molecules. *arXiv preprint arXiv:2309.12948*, 2023.

McGlynn, N. F. Thinking fast and slow. *Australian veterinary journal*, 92 12:N21, 2014. URL `https://api.semanticscholar.org/CorpusID:36031679`.

Meister, C. and Cotterell, R. Language model evaluation beyond perplexity. *arXiv preprint arXiv:2106.04638*, 2021.

Meuschke, N., Jagdale, A., Spinde, T., Mitrović, J., and Gipp, B. *A Benchmark of PDF Information Extraction Tools Using a Multi-task and Multi-domain Evaluation Framework for Academic Documents*, pp. 383–405. Springer Nature Switzerland, 2023. ISBN 9783031280320. doi: 10.

1007/978-3-031-28032-0_31. URL `http://dx.doi.org/10.1007/978-3-031-28032-0_31`.

Mirza, A., Alampara, N., Kunchapu, S., Ríos-García, M., Emoekabu, B., Krishnan, A., Gupta, T., Schilling-Wilhelmi, M., Okereke, M., Aneesh, A., et al. Are large language models superhuman chemists? *arXiv preprint arXiv:2404.01475*, 2024a.

Mirza, A., Alampara, N., Kunchapu, S., Ríos-García, M., Emoekabu, B., Krishnan, A., Gupta, T., Schilling-Wilhelmi, M., Okereke, M., Aneesh, A., Elahi, A. M., Asgari, M., Eberhardt, J., Elbeheiry, H. M., Gil, M. V., Greiner, M., Holick, C. T., Glaubitz, C., Hoffmann, T., Ibrahim, A., Klepsch, L. C., Köster, Y., Kreth, F. A., Meyer, J., Miret, S., Peschel, J. M., Ringleb, M., Roesner, N., Schreiber, J., Schubert, U. S., Stafast, L. M., Wonanke, D., Pieler, M., Schwaller, P., and Jablonka, K. M. Are large language models superhuman chemists?, November 2024b. URL `http://arxiv.org/abs/2404.01475`. arXiv:2404.01475 [cs].

Penedo, G., Kydlíček, H., allal, L. B., Lozhkov, A., Mitchell, M., Raffel, C., Werra, L. V., and Wolf, T. The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale, October 2024. URL `http://arxiv.org/abs/2406.17557`. arXiv:2406.17557.

Pezeshkpour, P. Measuring and modifying factual knowledge in large language models. In *Proceedings of the 22nd International Conference on Machine Learning and Applications*, pp. 992–999, 2023.

RDKit, online. RDKit: Open-source cheminformatics. `http://www.rdkit.org`, 2023.

Rozière, B., Lample, G., Izacard, G., Simón, J., Palmer, A., Ruan, S., Nguyen, M. O., Scales, N., et al. Code Llama: Open foundation models for code. Technical report, Meta AI, 2023.

Schwaller, P., Laino, T., Gaudin, T., Bolgar, P., Hunter, C. A., Bekas, C., and Lee, A. A. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS Cent. Sci.*, 5(9):1572–1583, 2019.

Schwaller, P., Petraglia, R., Zullo, V., Nair, V. H., Haeuselmann, R. A., Pisoni, R., Bekas, C., Iuliano, A., and Laino, T. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chem. Sci.*, 11:3316–3325, 2020.

Shen, J., Tenenholtz, N., Hall, J., Alvarez-Melis, D., and Fusi, N. Tag-LLM: Repurposing general-purpose LLMs for specialized domains. *arXiv preprint arXiv:2402.07927*, 2024.

Shin, J., Wang, C., Yu, Z., Ho, M., Smith, J. R., Pugh, C., Hajishirzi, H., Ostendorf, M., Farhadi, A., and Yih, W.-t. Biomegatron: Larger biomedical domain language model. *arXiv preprint arXiv:2010.09889*, 2020.

Sun, L., Luo, D., Ma, D., Zhao, Z., Shen, Z.-W., Zhu, S., Chen, L., Chen, X., and Yu, K. SciDFM: A large language model with mixture-of-experts for science. *arXiv preprint arXiv:2409.01234*, 2024.

Swain, M. C. and Cole, J. M. Chemdataextractor: A toolkit for automated extraction of chemical information from the scientific literature. *Journal of Chemical Information and Modeling*, 56(10):1894–1904, 2016. doi: 10.1021/acs.jcim.6b00207. URL `https://doi.org/10.1021/acs.jcim.6b00207`. PMID: 27669338.

Taylor, R., Kardas, M., Cucurull, G., Scialom, T., Hartshorn, A., Saravia, E., Poulton, A., Kerkez, V., and Stojnic, R. Galactica: A large language model for science, 2022. URL `https://arxiv.org/abs/2211.09085`.

Wang, H., Skreta, M., Ser, C.-T., Gao, W., Kong, L., Strieth-Kalthoff, F., Duan, C., Zhuang, Y., Yu, Y., Zhu, Y., et al. Efficient evolutionary search over chemical space with large language models. *arXiv preprint arXiv:2406.16976*, 2024a.

Wang, P., He, K., Wang, Y., Song, X., Mou, Y., Wang, J., Xian, Y., Cai, X., and Xu, W. Beyond the known: Investigating llms performance on out-of-domain intent detection. In *International Conference on Language Resources and Evaluation*, 2024b. URL `https://api.semanticscholar.org/CorpusID:268032564`.

Wang, W., Haddow, B., Birch, A., and Peng, W. Assessing the reliability of large language model knowledge. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1234–1249, 2024c.

Wang, Y., Yang, Q., Zeng, Z., Ren, L., Liu, L., Peng, B., Cheng, H., He, X., Wang, K., Gao, J., Chen, W., Wang, S., Du, S. S., and Shen, Y. Reinforcement Learning for Reasoning in Large Language Models with One Training Example, April 2025. URL `http://arxiv.org/abs/2504.20571`. arXiv:2504.20571 [cs].

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Weininger, D. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28:31–36, 1988.

Weng, W., Tan, X., Wang, H., and Zhou, P. Realign: Bilingual text-to-motion generation via step-aware reward-guided alignment. 2025. URL `https://api.semanticscholar.org/CorpusID:278394818`.

White, C., Dooley, S., Roberts, M., Pal, A., Feuer, B., Jain, S., Shwartz-Ziv, R., Jain, N., Saifullah, K., Naidu, S., et al. Livebench: A challenging, contamination-free benchmark for large language models. *arXiv preprint arXiv:2403.12345*, 2024.

Xia, Y., Jin, P., Xie, S., He, L., Cao, C., Luo, R., Liu, G., Wang, Y., Liu, Z., Chen, Y.-J., Guo, Z., Bai, Y., Deng, P., Min, Y., Lu, Z., Hao, H., Yang, H., Li, J., Liu, C., Zhang, J., Zhu, J., Bi, R., Wu, K., Zhang, W., Gao, K., Pei, Q., Wang, Q., Liu, X., Li, Y., Zhu, H., Lu, Y., Ma, M., Wang, Z., Xie, T., Maziarz, K., Segler, M., Yang, Z., Chen, Z., Shi, Y., Zheng, S., Wu, L., Hu, C., Dai, P., Liu, T.-Y., Liu, H., and Qin, T. Nature language model: Deciphering the language of nature for scientific discovery, 2025. URL `https://arxiv.org/abs/2502.07527`.

Xie, T., Wan, Y., Huang, W., Yin, Z., Liu, Y., Wang, S., Linghu, Q., Kit, C., Grazian, C., Zhang, W., and Hoex, B. Darwin series: Domain specific large language models for natural science. *arXiv preprint arXiv:2308.09913*, 2023a.

Xie, T., Wan, Y., Huang, W., Yin, Z., Liu, Y., Wang, S., Linghu, Q., Kit, C., Grazian, C., Zhang, W., Razzak, I., and Hoex, B. Darwin series: Domain specific large language models for natural science. *ArXiv*, abs/2308.13565, 2023b. URL `https://api.semanticscholar.org/CorpusID:274142505`.

Xie, Y., Aggarwal, K., and Ahmad, A. Efficient continual pre-training for building domain specific large language models. In *Findings of the Association for Computational Linguistics (ACL)*, 2024a.

Xie, Y., Goyal, A., Zheng, W., Kan, M.-Y., Lillicrap, T. P., Kawaguchi, K., and Shieh, M. Monte carlo tree search boosts reasoning via iterative preference learning. *ArXiv*, abs/2405.00451, 2024b. URL `https://api.semanticscholar.org/CorpusID:269484186`.

Yang, Z., Liu, W., Gao, B., Xie, T., Li, Y., Ouyang, W., Poria, S., Cambria, E., and Zhou, D. Moose-chem: Large language models for rediscovering unseen chemistry scientific hypotheses, 2025. URL `https://arxiv.org/abs/2410.07076`.

Ye, F., Yang, M., Pang, J., Wang, L., Wong, D. F., Yilmaz, E., Shi, S., and Tu, Z. Benchmarking large language models via uncertainty quantification. *arXiv preprint arXiv:2401.12321*, 2024.

Yu, B., Baker, F. N., Chen, Z., Ning, X., and Sun, H. Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset, 2024a. URL `https://arxiv.org/abs/2402.09391`.

Yu, B., Baker, F. N., Chen, Z., Ning, X., and Sun, H. Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset. *arXiv preprint arXiv:2402.09391*, 2024b.

Yue, Y., Chen, Z., Lu, R., Zhao, A., Wang, Z., Yue, Y., Song, S., and Huang, G. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?, 2025. URL `https://arxiv.org/abs/2504.13837`.

Zhang, D., Liu, W., Tan, Q., Chen, J., Yan, H., Yan, Y., Li, J., Huang, W., Yue, X., Ouyang, W., Zhou, D., Zhang, S., Su, M., Zhong, H.-S., and Li, Y. Chemllm: A chemical large language model, 2024. URL `https://arxiv.org/abs/2402.06852`.

Zhao, S., Nguyen, T., and Grover, A. Probing the decision boundaries of in-context learning in large language models. *arXiv preprint arXiv:2406.01234*, 2024.

A. Preliminaries

The notions and the metrics used throughout the paper that constitute our diagnostic suite is as follows:

A.1. Prerequisite 1: Symbolic Competence

To assess the symbolic competence of models, we compute the likelihood of generating a given sequence, in our case, a set of SMILES strings. We use a dataset of 10,000 molecules obtained from PubChem (Kim et al., 2025), and use the following definitions to compute a symbolic competence score.

Token log-likelihood extraction Given a model p_θ and a SMILES string $s = (t_1, ..., t_L)$.

At position i we compute the log-likelihood $r_{i,p_\theta}(s)$ of ground-truth token s_i within p_θ's next-token distribution $r_{i,p_\theta}(s) := p_\theta(t_i|t_1...t_{i-1})$. The mean of the whole string is taken as:

$$r_{p_\theta}(s) = \frac{1}{L}\sum_{i=1}^{L} r_{i,p_\theta}(s) \tag{2}$$

Symbolic competence score We define the symbolic competence score (SCS) on the assumption that a symbolically competent model should assign better likelihoods to chemically correct strings than to corrupted or invalid ones. We therefore measure the separation in the distributions of mean ranks between valid (canonical) SMILES and corrupted ones:

$$SCS := \frac{\bar{r}(corrupt(m)) - \bar{r}(canon(m))}{\sigma_{pool}}, \tag{3}$$

$$\sigma_{\text{pool}} = \sqrt{\frac{(n_1-1)\sigma_1^2 + (n_2-1)\sigma_2^2}{n_1+n_2-2}} \tag{4}$$

where σ_1 and σ_2 are each set's standard deviations, and σ_{pool} is the pooled standard deviation of the two sets. SCS is the Cohen's d effect size, where higher values indicate a cleaner separation and therefore stronger symbolic competence. A score of 0 means the model cannot distinguish canonical from corrupted strings, while SCS $\approx$ 2 corresponds to ¿ 95 % separation. $corrupt$ is a SMILES corruption operator that randomly deletes grammar characters with a probability of 0.2, effectively yielding invalid but similar SMILES. For the material science (MatSci) task, instead of corrupting and calculating the SCS on SMILES, the calculations are performed on compositions, which specify their elements and space group in the format: A B A B ¡sg$X$¿, where A and B are elements, and X represents the space group number.

A.2. Prerequisite 2: Latent Chemical Knowledge

As has recently been shown, the role of RL in training reasoning LLMs seems to be that of an amplifier, i.e., correct answers already exist in the base model's prior distribution with non-negligible probability.

With this in mind, we aim to assess the latent chemical knowledge of a given base model. As a proxy to this, we adopt the same strategy as that we use with the symbolic data, by measuring the Chemical-Competence Score (CCS), defined as the difference in the distributions of mean ranks between factually correct chemical statements and wrong ones. Given a list of chemical statements, such as the SMolInstruct Molecule Description subset (Yu et al., 2024b), we generate corrupted data by randomly swapping one sentence from each original statement with that from another randomly chosen statement in the pool.

A.3. Post-training methods

Large-scale pre-training furnishes the *prerequisites* discussed in Sections A.1–A.2. We now describe the two post-training methods that we use throughout this work to surface and amplify these capabilities.

Supervised fine-tuning on reasoning traces Recent research (Guo et al., 2025a) has revealed that small base models can be trained with SFT on reasoning traces, resulting in small reasoning models that mimic the behavior demonstrated in the SFT training data, even if such data does not directly target the specific downstream task the models are evaluated on. The reason is that SFT transfers the response style and not only the task-specific capabilities, thus serving as an *amplifier* of latent knowledge. Following this, some reasoning traces were distilled from DeepSeek-R1 and used to perform SFT on our pretrained models. We generated $\sim$ 600,000 solutions for two canonical tasks: IUPAC → SMILES and SMILES → IUPAC, based on PubChem compounds.

Reinforcement learning with verifiable rewards Following recent works (Wang et al., 2025), we adopt Reinforcement Learning with Verifiable Rewards (RLVR) as a post-training method for our models. In this context, models are trained online with rule-based rewards that depend entirely on the final outcome. The goal of this type of training, as exemplified in previous works (Wang et al., 2025) is to encourage the model to achieve good results on the training tasks, while developing intermediate strategies to achieve this, that might involve reasoning.

We designed and used different types of reward functions for our GRPO experiments: (1) formatting rewards to ensure separation between the model reasoning and answer, (2) accuracy rewards to verify the correctness of the model answer, (3) helper rewards to penalize the model if the completions are ill-formed (such as very short completions, repetitive behaviors etc.). For the accuracy rewards, we employed different approaches to compare the answer and the solution, such as exact matches, Tanimoto similarity between SMILES, or Levenshtein distance.

Downstream reasoning tasks To train and evaluate the reasoning capabilities of our models, we implemented a suite of challenging tasks relevant to chemistry. The tasks have been selected with the following criteria in mind: (1) Difficulty: the task must be challenging enough to be unsolvable by base models alone, (2) Reasoning-suitable: tasks must be suitable for reasoning, i.e. solving an instance of the task would require more System-2 thinking from human experts than System-1 (see 4), and (3) Dataset availability: Datasets must be readily available such that, upon adaptation, an input-outcome dataset can be built that is representative of the task. The final list of tasks is listed in Table 22, and implementation details are provided in the Appendix C.

B. preprocessing pipeline

An overview of our preprocessing pipeline is depicted as follows. Initially, we leveraged Nougat (Blecher et al., 2023) and GROBID (Meuschke et al., 2023) libraries for converting PDF documents into textual formats. Nougat demonstrated superior performance in accurately transforming complex structures such as tables, formulae, bibliographic references, and figure captions into LaTeX-formatted text. Conversely, GROBID excelled at extracting plain textual content from PDFs. The output of the authors were merged with explicit tags assigned to each structural element: tables were encapsulated with [START_TABLE] and [END_TABLE], formulas marked by [START_FORMULA] and [END_FORMULA], bibliographic references enclosed within [START_BIBREF] and [END_BIBREF], and figure descriptions bracketed by [START_FIGURE] and [END_FIGURE]. Subsequently, this structured text was processed through the Chemical Data Extractor 2 (Swain & Cole, 2016), identifying candidate molecule entities along with their positional context within the text. To ensure high precision in entity identification, candidates were further validated using a custom-trained sentence transformer model, designed specifically to discern genuine molecular entities from contextual information. Validated molecular entities were then translated from their IUPAC nomenclature to SMILES notation using py2opsin, a Python interface for OPSIN (Lowe et al., 2011). In cases where OPSIN failed to yield a definitive conversion, entities were cross-referenced against PubChem (Kim et al., 2025). Ultimately, during the pretraining phase alone, our model encountered approximately 800,000 unique chemical compounds along with their corresponding SMILES representations.

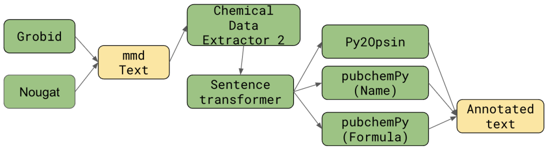


Figure 1. Overview of the preprocessing pipeline

C. Chemical Tasks

This section compiles all of the chemical tasks we have implemented for training models with RLVR, including details of the dataset compilation, rewards used, along with additional experimental results not shown in the main manuscript. The details for each task are specified with a table containing a summary of the rewards used, short explanations of what the input and expected outputs are, and finally, some sample outputs from the trained models. Following the table, we report additional experimental results, implementation details, and other task-specific details.

C.1. Reaction Prediction (RxP)

Table 5. Example of reaction prediction task.

Task: Reaction prediction	
Dataset Source	USPTO-480K (Coley et al., 2019)
Input	Reason and predict the correct product in SMILES notation from the following reaction: `CCN(CC)CC.COc1ccc(O)cc1.ClC(Cl)Cl.ClCCl.O.O=C(Cl)c1ccccc1`.
Target Molecule	`COc1ccc(OC(=O)c2ccccc2)cc1`
Output	<think>The reaction involves the acylation of a phenolic hydroxyl group using benzoyl chloride `O=C(Cl)c1ccccc1` in the presence of triethylamine `CCN(CC)CC` as a base. Here's the step-by-step mechanism: . . . 1. Reactants Identification: . . . 2. Mechanism: . . . 3. Product: . . . Final Answer: The product SMILES is `COc1ccc(OC(=O)c2ccccc2)cc1`. </think> <answer>`COc1ccc(OC(=O)c2ccccc2)cc1` </answer>
Answer	`COc1ccc(OC(=O)c2ccccc2)cc1` ✓

The reaction prediction task requires the model to reason and predict the correct product molecule given a list of reactants and reagents (Table 5). Solving this task usually requires expert chemists to think about the reactivity of the reactants involved, and propose and evaluate different reaction mechanism hypotheses. These serve as arguments and causal explanations that support the decisions.

The dataset for the RLVR training of this task was derived from the USPTO-480K (Coley et al., 2019) after removing the samples used in the SFT phase. 50K reactions were randomly chosen for the training set, and 500 reactions for the test set.

Given a model output o, from which a final answer a can be extracted, the reward function is the sum of format correctness ($R_{\text{format}} : o \mapsto [-1, 1]$, see Appendix E) and accuracy of the predicted product ($R_{\text{acc}} : a \mapsto \{-1, -0.5, 1\}$). The accuracy reward is determined by an exact match check against the ground truth:

$$R_{\text{acc}}(a) = \begin{cases} -1 & a \quad \text{is invalid/ not SMILES} \\ -0.5 & a \neq \text{true molecule} \\ +1 & a = \text{true molecule} \end{cases}$$

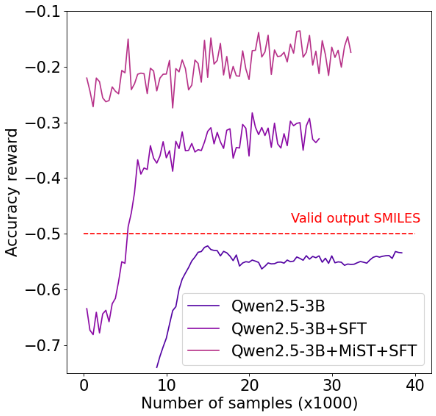


Figure 2. Accuracy reward evolution.

Figure 2 illustrates the evolution of the accuracy reward throughout training. The base Qwen2.5-3B model plateaus early at a reward below the -0.5 threshold, indicating that while it frequently generates syntactically valid SMILES strings, it fails to predict the correct product molecules. In contrast, both fine-tuned variants (Qwen2.5-3B+SFT and Qwen2.5-3B+MiST+SFT) maintain accuracy rewards above -0.5 in the majority of the training process. The SFT-only model shows a sharp increase in reward during the first 8,000 training samples, reflecting rapid format alignment. This is followed by a more gradual rise, suggesting improved chemical reasoning over time. Notably, the model pretrained with MiST before SFT follows a similar trajectory but consistently achieves a higher reward ceiling. This highlights the effectiveness of MiST and SFT in enhancing the model's learning capability during RLVF.

C.2. IUPAC to SMILES (I2S)

Table 6. Example of IUPAC→SMILES conversion.

Task: IUPAC to SMILES Conversion	
Dataset Source	PubChem (CID 2,000,001 to 3,000,000) (Kim et al., 2025)
Input	Question: You are an expert in Cheminformatics, who is very familiar with Simplified Molecular Input Line Entry System (SMILES) notation, and here's a task for you: Given a molecule with the IUPAC name as below, please provide the corresponding SMILES notation. Here is the IUPAC name: "4-(azepan-1-ylsulfonyl) -N-(pyridin-3-ylmethylideneamino) benzamide".
Target Molecule	O=C(NN=CC1=CC=CN=C1)C1=CC=C(S(=O)(=O)N2CCCCCC2)C=C1
Output	<think>The SMILES representation of 4-(azepan-1-ylsulfonyl) -N-(pyridin-3-ylmethylideneamino) benzamide is O=C(NN=CC1=CC=CN=C1)C1=CC=C(S(=O) (=O)N2CCCCCC2)C=C1 ...</think>
Answer	<answer=C(NN=CC1=CC=CN=C1)C1=CC=C(S(=O)(=O)N2CCCCCC2)C=C1</answer> ✓

The IUPAC-to-SMILES conversion task presented requires the model to translate a systematic chemical name into its corresponding SMILES string notation. We source our examples from the USPTO-480K dataset and prompt the model with a clear generative instruction "Given this IUPAC name, please provide the corresponding SMILES notation." The ground truth is established by the Tanimoto similarity score. The model's chain-of-thought reasoning is captured inside <think>…</think>tags and the final SMILES inside <answer>…</answer>. This setup tests the model's ability to parse complex nomenclature and generate SMILES structural representations.

For GRPO reward pipeline, we use accuracy reward function that only considers the final '<answer>' SMILES s_{ans}. After converting both the reference s^* and prediction s_{ans} to Morgan fingerprints, it computes the raw Tanimoto similarity τ and then applies a piecewise linear scaling:

$$r_{\mathrm{tf}}(\tau) = \begin{cases} +1.0, & \tau = 1.0, \\ \tau - 0.3, & 0.3 \le \tau < 1.0, \\ -0.5, & \tau < 0.3. \end{cases}$$

A perfect match yields $r_{\mathrm{tf}} = 1.0$, moderate similarity is rewarded proportionally above -0.5, and very poor matches incur a negative penalty.

C.3. Reaction Naming

Table 7. Reaction Naming experiment description.

Task: Reaction Naming (RxN)	
Dataset Source	USPTO Reaction 1M; (Coley et al., 2017)
Input	`What is the name of this chemical reaction? The reagents are CC(C)(C)[Si](C)(C)Cl and Nc1ccc(-c2cccc(O)c2)c2c1C(=O)NC2 and the product CC(C)(C)[Si](C)(C)Oc1cccc(-c2ccc(N)c3c2CNC3=O)c1. Choose ONLY from the following options and write your response choice inside <answer>...</answer>: "Acylation", "Aromatic Heterocycle Formation", "C-C Coupling", "Deprotection", "Functional Group Addition", "Functional Group Interconversion", "Heteroatom Alkylation and Arylation", "Miscellaneous", "Protection", "Reduction". Do not provide a final answer different than what is provided in this list.`
Target Reaction Name	Protection
Output	<think>First, let's break down the reagents. The first reagent is `CC(C)(C)[Si](C)(C)Cl` which is `tert-butyldimethylsilyl chloride`...
Answer	<answer>Protection </answer>

The reaction naming task is a classic example of a structured classification problem in cheminformatics, where the goal is to categories the nature of a reaction given reactants, conditions and products. This approach aim to test the ability of the LLM to conduct chemical reasoning and instruction following for discrete level answering. In addition to that, this setup also tests the model's ability to interpret chemical structures from linear notation and enables us to reveal how chain-of-thought guidance and prompt design impact classification accuracy. To stimulate reasoning, the model is tasked to output his thinking process inside <think>…</think>tags before emitting the final choice in <answer>…</answer>tags. The ground-truth class labels are evenly drawn from ten commonly found reaction type in chemistry: ”Acylation”, ”Aromatic Heterocycle Formation”, ”C-C”, ”Coupling”, ”Deprotection”, ”Functional Group Addition”, ”Functional Group Interconversion”, ”Heteroatom”. ”Alkylation and Arylation”, ”Miscellaneous”, ”Protection” and ”Reduction” derived from curated USPTO reactions dataset.

Reward Functions:

- **Continuous Format Reward:**
  - This reward is described in Section E.2.1 in the Algorithm 3.
- **Accuracy Reward:**
  - 0 if no answer is given
  - 0.1 if a single answer is given (but wrong)
  - 1 if the answer is entirely correct
  - -0.2 penalty if the model always output the same wrong class
- **Accuracy Percentage Reward:** discrete reward to foster perfect answers
  - 0 if the answer is wrong
  - 1 if the answer is entirely correct

C.4. Reaction Replacement

Table 8. Reaction Replacement experiment description.

Task: Reaction Replacement (RxR)	
Dataset Source	USPTO Reaction 1M; (Coley et al., 2017)
Input	Question: Which chemical reaction is correct? Choose from the following options: A. In the following reaction, the reagents are: `Cc1ncc(C=O)n1C1CC1`, `CC(C)OC=C(Br)C=O`, `Cl`, `O=C(c1cc(N2CCNC2=O)ccc1F)N1CCCN(c2nccs2)CC1` and the product is: `O=Cc1cnc2n1CCCC2`. B. In the following reaction, the reagents are: `Cc1ncc(C=O)n1C1CC1`, `CC(C)OC=C(Br)C=O`, `Cl`, `N=C1CCCCN1` and the product is: `CNC(=O)CC1(O)CCCN(C(=O)c2cncc(F)c2)C1`. C. In the following reaction, the reagents are: `Cc1ncc(C=O)n1C1CC1`, `CC(C)OC=C(Br)C=O`, `Cl`, `N=C1CCCCN1` and the product is: `O=Cc1cnc2n1CCCC2`. D. In the following reaction, the reagents are: `Cc1ccccc1OCCC(=O)N1CCCC(c2ccn[nH]2)C1`, `CC(C)OC=C(Br)C=O`, `Cl`, `N=C1CCCCN1` and the product is: `O=Cc1cnc2n1CCCC2`. Make sure to give your choice A, B, C, or D inside the <answer>…</answer>tags.
Target Molecular Reaction (Choice)	C
Output	<think>Let’s evaluate each option step by step to determine which one is correct. Option A: The reagent: `Cc1ncc(C=O)n1C1CC1` matches with the molecule `Cc1ncc(C=O)n1C1CC1. ...`
Answer	<answer>C </answer>

The reaction replacement tasks challenges the model to understand chemical reaction concepts, validity and ability to detect subtle structural inconsistencies. By providing the model with four nearly identical choices, chemical reaction notation coherence understanding is required. Each dummy reaction has one reagent randomly swapped, where starting from a correct USPTO reaction, we generate three “corrupted” variants by replacing a single reactant or product with the most Tanimoto-similar molecule drawn from a random batch of 50 Enamine50k compounds. In the prompt we provide the lists options A–D, each specifying reagent SMILES, conditions SMILES, and product SMILES, and the model is then instructed to answer one of the four choices as the correct one. The model is also instructed to think through each option step by step inside <think>…</think>and the answer is emitted inside <answer>…</answer>tags.

Reward Functions:

- **Continuous Format Reward:**
  - This reward is described in Section E.2.1 in the Algorithm 3.
- **Accuracy Reward:**
  - 0 if the answer is wrong
  - 1 if the answer is entirely correct

C.5. Reaction Inversion

Table 9. Reaction Inversion experiment description.

Task: Reaction Inversion (RxI)	
Dataset Source	USPTO Reaction 1M; (Coley et al., 2017)
Input	Question: Which chemical reaction is correct? Choose from the following options: A. In the following reaction, the reagents are: `BrCc1ccccc1`, `[K+]`, `[OH-]`, `O=C(O)c1ccc(OCc2ccccc2)cc1` and the product is: `CCOC(=O)c1ccc(O)cc1`. B. In the following reaction, the reagents are: `C=O`, `O=Cc1ccccc1`, `[B-]C#N`, `[Na+]`, `CN[C@H]1[C@@H](C)C[C@@H](c2ccncc2NC(=O)OC(C)(C)C)C[C@H]1NC(=O)OC(C)(C)C`, the conditions are: `CO`, `[OH-]`, `[OH-]`, `[Pd+2]`, and the product is: `C[C@H]1C[C@@H](c2ccncc2NC(=O)OC(C)(C)C)C[C@@H](NC(=O)OC(C)(C)C)[C@H]1N`. C. In the following reaction, the reagents are: `CCOC(=O)C#N`, `CCOC(=O)Cl`, `Cc1ccoc1C=Nc1ccccc1`, the condition is: `C1(C)C(C)=CC=CC=1`, and the product is: `CCOC(=O)c1cc2ccoc2cn1`. D. In the following reaction, the reagents are: `CC1(C)OB(c2cn[nH]c2)OC1(C)C`, `Nc1nc(-c2cc3c(s2)-c2ccc(-c4cn[nH]c4)cc2OCC3)c(-c2ccccc2Cl)s1` and the product is: `Nc1nc(-c2cc3c(s2)-c2ccc(Br)cc2OCC3)c(-c2ccccc2Cl)s1`. Make sure to give your choice A, B, C, or D inside the <answer>…</answer>tags.
Target Molecular Reaction (Choice)	C
Output	<think>Starting with option A: The reaction uses benzyl bromide `BrCc1ccccc1` …
Answer	<answer>C </answer>

The reaction inversion task challenges the model to understand chemical reaction concepts, validity and ability to detect subtle structural inconsistencies. By providing the model with four completely different choices, strong chemical reaction notation coherence understanding is required. Each dummy reaction has one reagent randomly swapped with the longest string SMILES among the products, enabling us to obtain 4 different reaction choices. In the prompt we provide the lists options A–D, each specifying reagent SMILES, conditions SMILES, and product SMILES, and the model is then instructed to answer one of the four choices as the correct one. The model is also instructed to think through each option step by step inside <think>. . . </think>and the answer is emitted inside <answer>. . . </answer>tags.

Reward Functions:

- **Continuous Format Reward:**
  - This reward is described in Section E.2.1 in the Algorithm 3.
- **Accuracy Reward:**
  - 0 if the answer is wrong
  - 1 if the answer is entirely correct

C.6. Reaction True/False

Table 10. Reaction True/False experiment description.

Task: Reaction True/False (RxTF)	
Dataset Source	USPTO Reaction 1M; (Coley et al., 2017)
Input	Question: Is this chemical reaction correct? In the following reaction, the reagent is: `COC(=O)c1ccc(OC)c(OCCc2cccc(C#N)c2)c1`, the conditions are: `C1COCCO1`, `[Li+]`, `[OH-]`, and the product is: `COc1ccc(C(=O)O)cc1OCCc1cccc(C#N)c1`.
Target Molecular Reaction Validity	True
Output	<think>First, I remember that LiOH, `[Li+][OH-]` is a strong base, so it's likely an acid-base reaction. The ester group in the starting material . . .
Answer	<answer>True </answer>

The Reaction True/False task is a binary derivative of the Reaction Replacement task. In this case, the model is asked to analyze and judge based on one single reaction, whether the reaction is correct or wrong. Each prompt presents one reaction—listing the reagent SMILES, the reaction conditions SMILES, and the product SMILES—and then asks "Is this chemical reaction correct?". The examples are drawn from the Reaction Replacement set, where some of the reactions have been corrupted by swapping one random molecule in the reaction string by a new candidate. The model is instructed to reason step by step inside <think>. . . </think>, then has to emit <answer</answer>or <answer</answer>accordingly. This format was designed to simplify the reaction replacement task by providing only a binary label choice, allowing us to not only reduce the task complexity but also diminish the hallucination effects emanating from providing many examples in the prompt. In this scenario, the model only ever sees one reaction, and its ability to detect subtle mismatches in chemical transformations and to follow a true/false classification protocol with transparent chain-of-thought is tested.

Reward Functions:

- **Continuous Format Reward:**
  - This reward is described in Section E.2.1 in the Algorithm 3.
- **Accuracy Reward:**
  - 0 if the answer is wrong
  - 1 if the answer is entirely correct

C.7. Chemical Formula Balancing Task (CeB)

Table 11. Template for reporting RL experiments.

Task: Chemical Formula Balancing Task (CeB)	
Dataset Source	The Perovskite Dataset (Jacobsson et al., 2022)
Input	You are now acting as a skilled chemistry expert. Your goal is to identify and complete the missing compound(s) represented by [MASK] in the given chemical reaction. Reaction: 6 BaCO3 + 1 Na2CO3 + 4 SiO2 == [MASK] Na2Ba[MASK] (Si[MASK]O[MASK])(SiO[MASK])2 + 7 CO2. Analyze the equation carefully, considering the law of conservation of mass and stoichiometry. Ensure the total number of atoms of each element on both sides of the equation remains balanced.
Target Molecule (Material)	6 BaCO3 + 1 Na2CO3 + 4 SiO2 = 1 Na2Ba6(Si2O9)(SiO3)2 + 7 CO2
Output	¡thinking¿1. Element Count Analysis: - Reactants: - Ba: 6 (from 6 BaCO3) - Na: 2 (from Na2CO3) - Si: 4 (from 4 SiO2) - C: 7 (6 from BaCO3 + 1 from Na2CO3) - O: 6×3 (BaCO3) + 1×3 (Na2CO3) + 4×2 (SiO2) = 29 - Products: - CO2: 7 CO2 (7 C, 14 O) - Missing compound must account for remaining elements: 6 Ba, 2 Na, 4 Si, and 15 O (29 total O - 14 from CO2). 2. Structure of Missing Compound: - The formula template [MASK] Na2Ba[MASK](Si[MASK]O[MASK]) (SiO[MASK])2 suggests: - 2 Na, 6 Ba (to balance Ba from reactants), 4 Si (from SiO2), and 15 O. - Silicate groups: (Si2O9) and (SiO3)2 (to total 4 Si and 15 O). 3. Charge Balance: - Cations: Na2 (2+) + Ba6 (12+) = +14 charge. - Anion: Si4O15 (4 Si × +4 + 15 O × -2 = +16 -30 = -14) balances the +14 charge. 4. Final Formula: - Na2Ba6(Si2O9)(SiO3)2 consolidates all elements and charges correctly.¡/thinking¿
Answer	<answer>6 BaCO3 + 1 Na2CO3 + 4 SiO2 = 1 Na2Ba6(Si2O9)(SiO3)2 + 7 CO2</answer>

The goal of this task is to evaluate the performance of large language models (LLM) in predicting balanced chemical equations, specifically leveraging a reward mechanism to enhance the accuracy of chemical formula generation. The GRPO reinforcement learning data set was derived from a data set of chemical formulas focusing on perovskite compounds. The dataset was prepared by selectively masking certain products or their stoichiometric coefficients within balanced chemical equations.

Reward Function Design: The reward function combines two distinct components: format correctness and prediction accuracy.

- **Format Reward:**
  - Initial reward is set to 0 if the generated chemical equation does not follow the correct format.
  - An additional reward of 0.3 is given if the left-hand side (reactants) of the generated equation matches exactly with the ground truth.
- **Accuracy Reward:** This reward component assesses the similarity between the generated right-hand side (products) (y_{LLM}) and the ground truth (y_{GT}) using a predefined similarity metric, formulated as:

$$\text{reward} = \text{format\_reward} + \text{sim}(y_{\text{LLM}}, y_{\text{GT}})$$

C.8. Conditional Material Generation (CMG)

Table 12. Conditional Material Generation experiment description.

Task: Conditional Material Generation (CMG)	
Dataset Source	Materials Project (Jain et al., 2013)
Input	Build a material that has O, Te, Tm. Please keep your reasoning as concise as possible. For example ¡material¿ A A B B B ¡sg12¿¡/material¿ where A, B refer to elements and ¡sg12¿ denotes the space group; for example: ¡material¿ Pa In Tc Tc ¡sg225¿¡/material¿.
Target Material Composition	Any valid composition made up of the elements O, Te, and Tm with a valid space group number, e.g.: ¡material¿ O O Te Tm Tm Te ¡sg127¿¡/material¿.
Output	To combine O, Te, and Tm, a space group like sg127 (for a 1:1:1 ratio) could work. A possible structure could be: ¡material¿ O O Te Tm Tm Te ¡sg127¿¡/material¿. This fits the space group 127, which is common for space-filling structures. (Note: space groups can vary, but this is a reasonable choice for a simple ternary compound.)
Answer	¡material¿ O O Te Tm Tm Te ¡sg127¿¡/material¿

This task aims to leverage the scientific knowledge embedded in MiST-trained LLMs to generate novel materials from a specified set of elements. The experiment focuses on the model's ability to understand three-dimensional atomic relationships within crystal structures and, based on that understanding, produce valid compositions. If the model can perform this task with high accuracy, it could significantly enhance the efficiency and cost-effectiveness of the material generation phase in the materials discovery process.

Reward Function Design: The quality of the generated composition is measured by the metrics: validity, precision and novelty. Validity is assessed using SMACT (Davies et al., 2016) validity, which checks whether the generated composition adheres to fundamental chemical rules, such as charge neutrality. Precision measures the model's ability to follow instructions and correctly include the specified elements. It is computed using the following equation:

$$\text{Precision} = \frac{|E_{pi} \cap E_{qi}|}{E_{pi}},$$

where E_{pi} is the set of elements specified in the *i*-th prompt and E_{qi} is the corresponding generated element (Xia et al., 2025). The novelty of the generated composition was determined based on whether the composition was present within the materials project dataset or was previously generated by the model. Furthermore, to ensure the model provided its generated solution in a valid format, the reward function also checked that the generated composition was enclosed within the ¡material¿...¡/material¿ tags and that the assigned space group number lies within the valid range of 1 to 230.

Therefore, the reward function used to train the LLM for the conditional material generation task was:

$$R = \alpha_1\text{Validity} + \alpha_2\text{Precision} + \alpha_3\text{Novelty} + \alpha_4\text{Format},$$

where the parameters α_i for $i = 1, ..., 5$ are scaling factors for each portion of the reward.

C.9. Binary Compound Structure Relaxation Task (CrR)

Table 13. Template for reporting RL experiments.

Task: Binary Compound Structure Relaxation Task (CrR)

Dataset Source	Materials Project Trajectory (MPtrj) Dataset (Deng, 2023), Materials Project (Jain et al., 2013)
Input	Given a perturbed binary compound: formula Ti 1_int Ni 1_int, space_group_symbol ’P1’ lattice_parameters a 2.74603000 b 2.94419691 c 2.94419691 alpha 119.99997545 beta 90.00000000 gamma 90.00000000 Zr 1_int 0.00000000 0.00000000 0.00000000 N 1_int 0.50000000 0.66670000 0.33330000 perform multiple steps Structural Relaxation on the given perturbed binary compound and reduce the internal energy. Please document your thought process within <think>…</think>tags, and provide the final corrected structure in <answer>…</answer>tags.
Target Material	A more stable structure of the same material with lower inner energy.
Output	¡thinking¿The input structure describes ZrN in a hexagonal P-6m2 space group with significant initial distortions. The relaxation process improves stability through three key mechanisms: 1. Symmetry Restoration:… ¡/thinking¿
Answer	<answerZr 1_int N 1_int space_group_symbol ’P1’ lattice_parameters a 3.15089295 b 3.15093266 c 2.93526192 alpha 90.00000000 beta 90.00000000 gamma 119.99958321 Zr 1_int 0.99999066 0.99998132 0.00000000 N 1_int 0.33330934 0.66661868 0.50000000</answer>

This task aims to utilize LLMs to perform preliminary relaxation (pre-relaxation) of unstable crystal structures, aiming to yield more stable structural configurations. This preliminary step is intended to substantially decrease computational costs and improve efficiency in subsequent high-accuracy Density Functional Theory (DFT) calculations. DFT calculations, while accurate, are computationally intensive. By leveraging LLM-generated pre-relaxation adjustments, the experiment seeks to effectively reduce the quantity of computationally unfavorable structures, thereby streamlining and accelerating the DFT computational pipeline.

Format Reward:

$$R_{\text{format}}(S_{\text{gen}},\, S_{\text{in}}) = \begin{cases} -1, & \text{if } S_{\text{gen}} \text{ is valid} \\ & \text{Mat2Seq format and} \\ & \text{has lower inner energy} \\ & \text{than } S_{\text{in}} \\ -5, & \text{if } S_{\text{gen}} \text{ is valid} \\ & \text{Mat2Seq format} \\ -10, & otherwise \end{cases}$$

D. Benchmarking procedure

In this section we elaborate on the methods used to evaluate the models in the multiple ways displayed in Table 22. Here we give details of how diagnostic metrics have been computed (SCS, CCS), which evaluate the capabilities in LLMs that are necessary for success on chemical tasks in an RL setting. Additionally, performances on downstream tasks have been computed using benchmarks derived from each task (see Appendix above), along with different prompting techniques, that mark the difference between direct answer, or reasoning answer.

D.1. Latent Symbolic and Chemical Knowledge

D.1.1. SYMBOLIC COMPETENCE SCORE BENCHMARK

The Symbolic Competence Score benchmark measures the model's latent capability to read and write correct chemical symbols. In this benchmark we focus particularly on SMILES, as organic chemistry spans a majority of our tasks. For this we collected 10000 valid SMILES from PubChem (**?**), such that no overlap exists with the MiST data. A second dataset is created with corrupted smiles based on these smiles, where corruptions are minimal, however render the smiles unvalid. The corruption procedure is specified in Algorithm 1. The algorithm removes a random subset of key structural grammar elements (ring/branch brackets and digits) from the SMILES string, producing broken or ambiguous strings. Corruption rate ρ controls the proportion of removed elements, which for all our experiments has been set to 0.2.

Algorithm 1 SMILES Grammar Element Corruption

```
SMILES string $s$, corruption rate $\rho$
Corrupted SMILES string $s_{\text{corrupt}}$
Let $\mathcal{G} = \{ (, ), [, ], 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ (grammar elements)
$L \leftarrow$ length of $s$
$I \leftarrow$ indices of $s$ where $s_i \in \mathcal{G}$
$|I| = 0$
    $s$
$N_{\text{remove}} \leftarrow \max(1, \lfloor \rho \cdot |I| \rfloor)$
Randomly select $R \subseteq I$ with $|R| = N_{\text{remove}}$
$s_{\text{corrupt}} \leftarrow$ empty string
$i \leftarrow 1$ to $L$
    $i \notin R$
        Append $s_i$ to $s_{\text{corrupt}}$
$s_{\text{corrupt}}$
```

Finally, evaluation happens in two stages. First, the log-likelihoods are computed using the model for the following string, that provides context for the string to look more natural:

```
The molecule represented with
the SMILES [BEGIN_SMILES] smiles
[END_SMILES]
```

Where `smiles` is replaced by both the correct, and the incorrect SMILES string. The log-likelihoods corresponding to the smiles tokens are isolated by dropping the computed likelihoods associated with the context shown above. The two corresponding strings are thus

Original SMILES:

```
The molecule represented with
the SMILES [BEGIN_SMILES] O=C(O)
C[C@H](O)C[C@H](O)CCn2c(c(c(c2c1
ccc(F)cc1)c3ccccc3)C(=O)Nc4ccccc
4)C(C)C [END_SMILES]
```

Corrupted SMILES:

```
The molecule represented with
the SMILES [BEGIN_SMILES]
O=C(O)C[C@H](O)C[C@H](O)CCn2c(c(c(c2
c1ccc(F)cc1)c3ccccc3)
C(=O)Nc4ccccc4)C(C)C
[END_SMILES]
```

Average loglikelihoods are computed for the whole sample of 10000 SMILES in this manner, and SCS score is computed as the Cohen's d effect size between the distributions of loglikelihoods of correct smiles, vs that of corrupted smiles.

Note that although the structure of material compositions is different from that of SMILES, the corruption method is similar, as key structural elements such as the space group number tag (`<sg12>`) and elemental symbols are replaced with special characters.

D.1.2. CHEMICAL COMPETENCE SCORE BENCHMARK

The Chemical Competence Score (CCS) evaluates a model's latent ability to distinguish between chemically accurate and inaccurate factual statements. To construct this benchmark, we selected 1,000 samples from the test split of the SMolInstruct Molecule Description dataset (Yu et al., 2024b), which was never used in all post-training stages. Each sample in the dataset consists of a brief description of an organic molecule. For example, one entry describes an acetamide as:

```
N-[4-(1,3-thiazol-2-ylsulfamoyl)
phenyl] acetamide is
a sulfonamide that is
benzenesulfonamide substituted
by an acetylamino group
at position 4 and a
1,3-thiazol-2-yl group at
the nitrogen atom.  It is a
metabolite of sulfathiazole.
It has a role as a marine
xenobiotic metabolite.  It is
```

```
a sulfonamide, a member of
acetamides, and a member of
1,3-thiazoles.
```

For material data, we utilized `Robocrystallographer` (Ganose & Jain, 2019) to generate 600 natural text descriptions for crystal structures from the Material Project (Jain et al., 2013). Here is an example entry:

```
AlN is Wurtzite structured and
crystallizes in the hexagonal
P6_3mc space group.  Al(1) is
bonded to four equivalent N(1)
atoms to form corner-sharing
AlN4 tetrahedra.  There are
three shorter (1.90 Å) and one
longer (1.91 Å) Al(1)-N(1) bond
length.  N(1) is bonded to four
equivalent Al(1) atoms to form
corner-sharing NAl4 tetrahedra.
```

To create a contrastive benchmark, we generated an incorrect version for each entry by replacing one sentence in the original description with a sentence from a different one, while keeping the target molecule/crystal unchanged. Here is an example of an incorrect version of the above acetamide example with the edited section highlighted:

```
N-[4-(1,3-thiazol-2-ylsulfamoyl)
phenyl]acetamide is a
tricyclic triterpenoid of the
isomalabaricane group.  It is
a metabolite of sulfathiazole.
It has a role as a marine
xenobiotic metabolite.  It
is a sulfonamide, a member
of acetamides and a member of
1,3-thiazoles.
```

D.2. Task Benchmarks

The benchmarks have been obtained by selecting a subset of the datasets defined in Appendix C, for each of the tasks.

D.3. Inference techniques

We observed that models' full text generation often overflows the available context window, without providing any final answer within <answer>tags, thus preventing its correct evaluation. To overcome this, upon failure to generate an <answer>tag, we directly append the <answer>tag and retry the generation, biasing the model towards generating an answer at that point. Pseudo-code for this is provided in Algorithm 2.

Table 14. Template for reporting RL experiments.

Task: Binary Compound Structure Relaxation Task (CrR)	
Dataset Source	Materials Project Trajectory (MPtrj) Dataset (Deng, 2023), Materials Project (Jain et al., 2013)
Input	Given a perturbed binary compound: formula Ti 1_int Ni 1_int, space_group_symbol 'P1' lattice_parameters a 2.74603000 b 2.94419691 c 2.94419691 alpha 119.99997545 beta 90.00000000 gamma 90.00000000 Zr 1_int 0.00000000 0.00000000 0.00000000 N 1_int 0.50000000 0.66670000 0.33330000 perform multiple steps Structural Relaxation on the given perturbed binary compound and reduce the internal energy. Please document your thought process within <think>. . . </think>tags, and provide the final corrected structure in <answer>. . . </answer>tags.
Target Material	A more stable structure of the same material with lower inner energy.
Output	¡thinking¿The input structure describes ZrN in a hexagonal P-6m2 space group with significant initial distortions. The relaxation process improves stability through three key mechanisms: 1. Symmetry Restoration:. . . ¡/thinking¿
Answer	<answerZr 1_int N 1_int space_group_symbol 'P1' lattice_parameters a 3.15089295 b 3.15093266 c 2.93526192 alpha 90.00000000 beta 90.00000000 gamma 119.99958321 Zr 1_int 0.99999066 0.99998132 0.00000000 N 1_int 0.33330934 0.66661868 0.50000000</answer>

An extension of such an injection technique is that models can be biased from the beginning of the completion towards directly providing an answer, thereby allowing us to evaluate the effect of the intermediate text inside <think>tags. In Table 22 in the main manuscript, direct answer results are reported outside of the parentheses, while reasoning results are in parentheses.

E. Experimental settings

E.1. MiST: Mid-stage Scientific Training

Our MiST model is based on the Qwen-2.5-3B model. We continue the pre-training and perform SFT thereafter on a chemically enriched corpus spanning a diversity of sources, targeting the two prerequisites we proposed in the main manuscript.

Algorithm 2 Answer tag injection `<answer>` - Think and answer procedure

```
InputInput OutputOutput
prompt, model_sampling_params, model, nbr_max_retries A completion containing <answer>...</answer>
result ← llm.generate(prompt, sampling_params) completion ← result.outputs[0].text
i ← 1 max_retries Append the '<answer>' token to coax a proper tag new_prompt ← prompt ++ competition ++ "<answer>" result ← llm.generate(new_prompt, sampling_params) complete_completion ← result.outputs[0].text HasAnswer(complete_completion ) complete_completion complete_completion *fallback if still no tag
```

The following configuration of hyperparameters was used for training:

Table 15. MiST Pretraining Hyperparameters

Parameter	**Value**
Model Architecture	Qwen-2.5-3B
Epochs	4 (~90,000 steps)
Batch Size	32
Max/Min Learning Rate	1×10^{-5} / 1×10^{-6}
LR Warmup Steps	1,000
LR Decay Steps	1,000
Optimizer	AdamW
Loss Function	Cross-Entropy
Hardware	32 × H100 GPUs
Total GPU Hours	640

After this stage, the model is further trained with SFT on instruction and Q&A data, as well as reasoning traces obtained from a stronger reasoning LLM, on more chemistry-relevant tasks; see the following section for more details. The following configuration was used:

Table 16. MiST SFT Hyperparameters

Parameter	**Value**
Model Architecture	Qwen-3B
Epochs	3 (~32,000 steps)
Batch Size	32
Learning Rate	1×10^{-6}
Optimizer	AdamW
Loss Function	Cross-Entropy
Hardware	32 × H100 GPUs

E.2. Reinforcement Learning experiments

The Open-R1 repository from Hugging Face (`https://github.com/huggingface/open-r1`) was forked and modified with additional features/optimizations for the GRPO experiments. Each training was run for 12 hours on four nodes (with four NVIDIA GH200 120GB GPUs), summing to 16 GPUs and 192 GPU-hours per training. The best hyperparameters are summarized in Table 17. A completion length of 8192 was used to let the model output long reasoning thoughts. The best hyperparameters and rewards were optimized using a total of 30k GPU-hours with variations in the experimental setups. The list of used rewards is described in Section E.2.1.

parameter	value
per_device_train_batch_size	1
gradient_accumulation_steps	8
learning_rate	2e-6
lr_scheduler_type	cosine
warmup_ratio	0.03
beta	0.04
max_prompt_length	384
max_completion_length	8192
num_generations	8
use_vllm	true
vllm_max_model_len	8192

Table 17. Optimized hyperparameters used for the GRPO training experiments.

E.2.1. REWARDS

The rewards designed for our GRPO experiments are grouped into two main categories:

- Format reward: the goal is to ensure that the trained model uses the appropriate format with reasoning (between <think>tags) and answer (between <answer>tags).
- Accuracy reward: the goal is to verify the answer of the model for the given task.

Accuracy reward: For the different tasks, different accuracy rewards are implemented in a continuous manner if possible. For SMILES-based tasks, the Tanimoto similarity score is generally used. However, for MCQA-based tasks, the rewards are usually discrete since the answers are correct or wrong. These rewards typically range from 0 to 1 (perfect answer).

Accuracy percentage reward: For each task, we also implement a discrete accuracy percentage reward to foster perfect answers and to log the training accuracy of the models. This reward is 0 if the answer is wrong and 1 if the answer is entirely correct.

Continuous format reward: A continuous format reward has been implemented with the structure described in Algorithm 3. The idea behind this reward is to output a score between -1 (very bad format) and 1 (perfect format) with continuous incremental small steps to guide the model to the expected format.

Algorithm 3 Continuous Formatting Reward Calculation

```
InputInput OutputOutput Raw model output $o \in$ String
Formatting reward $r \in [-1, 1]$
$r \leftarrow 0.0$ *[r]Initialize reward
$T \leftarrow \{$<think>, </think>, <answer>, </answer>$\}$
Check each tag appears exactly once tag $\in T$
COUNT($o$, tag) $= 1$ $r \leftarrow r + 0.05$  $r \leftarrow r - 0.05$
Check correct start and end tags
STARTS_WITH($o$, <think>) $r \leftarrow r + 0.05$
$r \leftarrow r - 0.05$
ENDS_WITH($o$, </answer>) $r \leftarrow r + 0.05$  $r \leftarrow r - 0.05$
Check think-answer boundary
COUNT($o$, </think>\n<answer>) $= 1$
$r \leftarrow r + 0.1$  $r \leftarrow r - 0.1$
Check answer block extraction $m_1 \leftarrow$
REGEX_MATCH(<answer>(.*)</answer>, $o$)
$m_1 =$ None $r \leftarrow r - 0.2$ NUM_GROUPS($m_1$) $\neq 1$
$r \leftarrow r - 0.05$  $r \leftarrow r + 0.2$
Check whole think \n answer pattern $m_2 \leftarrow$
REGEX_MATCH(<think>(.*)</think>\n
<answer>(.*)</answer>, $o$) $m_2 =$ None
$r \leftarrow r - 0.4$ NUM_GROUPS($m_2$) $\neq 2$ $r \leftarrow r - 0.1$
$r \leftarrow r + 0.4$
$r$
```

F. Data

F.1. Data sources and processing

F.1.1. FINEWEB-EDU

The FineWeb-Edu can be found on Hugging Face (`https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu`) (Penedo et al., 2024). The subsets "CC-MAIN-2013-20" to "CC-MAIN-2024-10" were downloaded for a total of ∼6 TB, which represents roughly 1.3T tokens and 1.26B individual texts. Based on the representative subset "sample-10BT" (also downloaded), the text sources were computed by taking the base URL (from the dataset column "url"), then these sources were sorted from the most prevalent to the least. We manually labeled the most prevalent sources as "chemistry", "non-chemistry", or "undetermined". The goal was to label a source as "chemistry" only if nearly all the texts from that source are about chemistry. On the other hand, a source is classified as "non-chemistry" only if there is no mention of chemistry in all the texts from that source. When a source contains a mix, like a school website with chemistry texts and texts for other fields, the label used is "undetermined", and the source is not used. After this manual labeling, the texts from "sample-10BT" were classified based on the labeled sources. It led to a ground truth of approximately 10,000 "chemistry" texts and 50,000 "non-chemistry" texts (out of the ∼10M texts found in "sample-10BT"). Based on this ground truth, a custom non-ML classifier was built using the word frequencies in "chemistry" and "non-chemistry" texts. The texts were lemmatized before building word frequency vectors for the two classes using a simple processing script that replaces any non-standard character with a space, before splitting the strings by the spaces. A custom vocabulary was also built to store these lemmatized texts in a tokenized manner. Other lemmatization methods (such as Spacy or NLTK) were also tried, but did not lead to better results and were extremely expensive to use on the full FineWeb dataset (>6 TB). After building the vocabulary and the word frequency vectors for the two classes, the formula below was applied to each FineWeb text to create an associated "chemistry score" (ranging from 0 to "infinity"). The frequencies of the lemma k in chemistry texts and non-chemistry texts are denoted f_k^c and f_k^n, respectively. The text chemistry score (TCS) is computed using the following equation:

$$\mathrm{TCS(text)} := \frac{1}{N_{\mathrm{lemmas}}} \sum_{\substack{k=\mathrm{lemma}\\ \mathrm{in\ text}}} w_k \quad \text{with} \quad w_k = \begin{cases} \frac{f_k^c}{f_k^n}, & \text{if } \frac{f_k^c}{f_k^n} > 1, \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

This labeling strategy was applied to the entire FineWeb-Edu corpus, and the texts with $TCS > 4$ were retained, yielding a pretraining set of 1.4 billion tokens of high-quality chemistry-labeled texts. The threshold $TCS > 4$ was decided based on the PR curve plot shown in Figure 3. This threshold allows for high precision, and the quantity of texts retrieved was sufficient for our pretraining pipeline. Additional plots with the percentage of chemistry texts by threshold and the cumulative number of chemistry token counts by threshold can be observed in Figures 4 and 5, respectively. Some chemistry text examples (with their associated TCS scores) are shown in Figure 6.

F.1.2. PUBCHEM

The first three million compounds from the PubChem database (Kim et al., 2025) (CID from 1 to 3,000,000) were dumped using the PUG REST API with batched requests in October 2024. Each record contains these columns (among others): CanonicalSMILES, IsomericSMILES, IUPACName, and InChI. Since the molecule canonicalization algorithm used in the PubChem database is not the

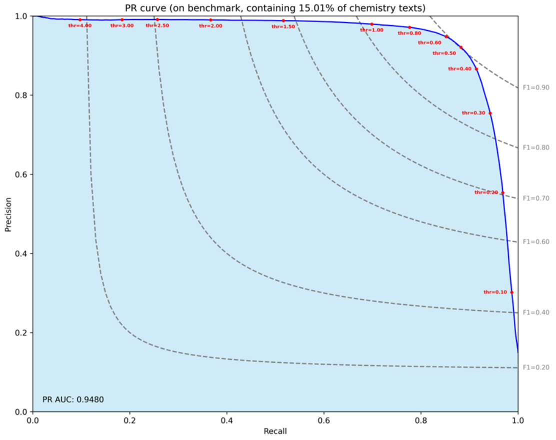


Figure 3. Precision-recall curve of the estimated retrieved chemistry texts based on the manually labeled ground truth. The different TCS thresholds are shown in red dots on the PR curve.

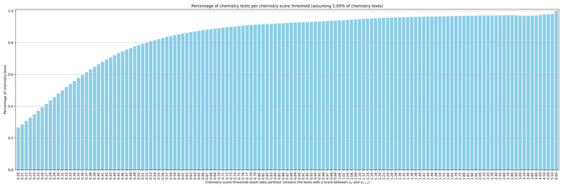


Figure 4. Estimated percentage of chemistry texts by TCS threshold.

same as the one used by RDKit, all the compounds were re-canonicalized. The canonical SMILES consistency was also ensured for each compound by computing four canonical SMILES for each molecule:

- CanonicalSMILES → canonicalized using RDKit.
- IsomericSMILES → canonicalized using RDKit.
- IUPACName → SMILES using py2opsin and then canonicalized using RDKit.
- InChI → canonical SMILES using RDKit.

Then the four newly generated canonical SMILES were compared, and if a mismatch is found, the compound is discarded. This method filtered out approximately 40% of the compounds, and the duplicated canonical SMILES were also discarded. For the remaining compounds, four ”SMILES variants” were computed using RDKit based on the canonical SMILES to have four non-canonical SMILES in each record. At the end of this processing script, an approximate of 1,800,000 compounds were kept and ready to be used. The dataset was then split in the following manner: the first million compounds (CID from 1 to 1,000,000) were used for pretraining, the second million compounds (CID from 1,000,001 to 2,000,000) were used for GRPO training, and the third million compounds (CID from 2,000,001 to 3,000,000) were used as the test split for benchmarking. Each split contains ∼600,000 valid compounds. Multiple derived datasets were also generated for the different chemical tasks used with GRPO training (explained in Section F.2 below).

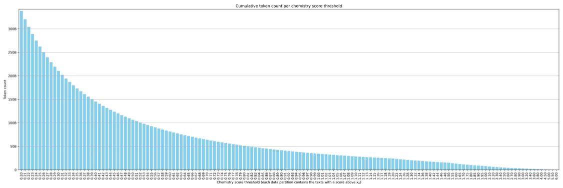


Figure 5. Estimated cumulative chemistry token count by TCS threshold.

F.2. Chemical Tasks Data sources

All MCQA-derived tasks for GRPO training are built on the USPTO Reaction 1M dataset, and the I2S dataset was built using the PubChem dataset from Section F.1.2:

Reaction Prediction (RxP)

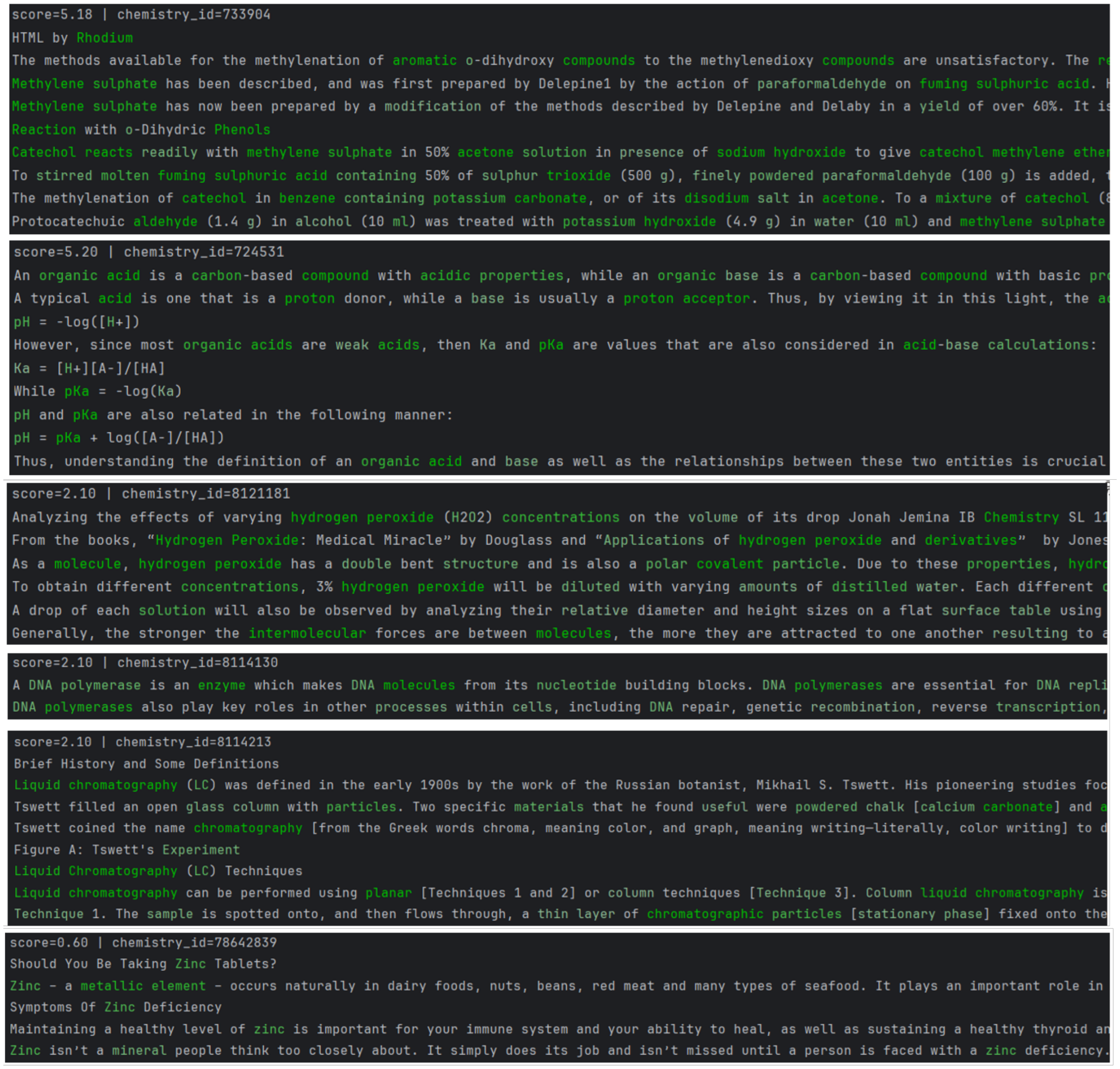
- The USPTO-480K dataset (Coley et al., 2019) consists of approximately 480K organic reactions, divided into training and test splits.
- We retained only reactions with a single product, resulting in roughly 400K training samples and 38K test samples.
- The first 10K reactions from the training set are used to generate reasoning traces.
- An additional 50K reactions, randomly selected from the remaining training data, are used for RLVF.
- A set of 500 reactions, randomly sampled from the test set, is used for benchmarking.

IUPAC to SMILES (I2S)

- The processed PubChem compounds (CID from 1,000,001 to 2,000,000) from the Section F.1.2 are used as the base data.
- The canonical SMILES and the IUPAC were directly used from the dataset.

Reaction Naming (RxN)

- Start from the full USPTO 1M reaction set.
- Use Rxn-Insight’s class generation to detect the reaction name.
- Filter to 600000 samples, evenly distributing across the 10 classes.



```
score=5.18 | chemistry_id=733904
HTML by Rhodium
The methods available for the methylenation of aromatic o-dihydroxy compounds to the methylenedioxy compounds are unsatisfactory. The r
Methylene sulphate has been described, and was first prepared by Delepine1 by the action of paraformaldehyde on fuming sulphuric acid.
Methylene sulphate has now been prepared by a modification of the methods described by Delepine and Delaby in a yield of over 60%. It i
Reaction with o-Dihydric Phenols
Catechol reacts readily with methylene sulphate in 50% acetone solution in presence of sodium hydroxide to give catechol methylene ether
To stirred molten fuming sulphuric acid containing 50% of sulphur trioxide (500 g), finely powdered paraformaldehyde (100 g) is added,
The methylenation of catechol in benzene containing potassium carbonate, or of its disodium salt in acetone. To a mixture of catechol (
Protocatechuic aldehyde (1.4 g) in alcohol (10 ml) was treated with potassium hydroxide (4.9 g) in water (10 ml) and methylene sulphate
```

```
score=5.20 | chemistry_id=724531
An organic acid is a carbon-based compound with acidic properties, while an organic base is a carbon-based compound with basic pr
A typical acid is one that is a proton donor, while a base is usually a proton acceptor. Thus, by viewing it in this light, the a
pH = -log([H+])
However, since most organic acids are weak acids, then Ka and pKa are values that are also considered in acid-base calculations:
Ka = [H+][A-]/[HA]
While pKa = -log(Ka)
pH and pKa are also related in the following manner:
pH = pKa + log([A-]/[HA])
Thus, understanding the definition of an organic acid and base as well as the relationships between these two entities is crucial
```

```
score=2.10 | chemistry_id=8121181
Analyzing the effects of varying hydrogen peroxide (H2O2) concentrations on the volume of its drop Jonah Jemina IB Chemistry SL 1
From the books, “Hydrogen Peroxide: Medical Miracle” by Douglass and “Applications of hydrogen peroxide and derivatives”  by Jone
As a molecule, hydrogen peroxide has a double bent structure and is also a polar covalent particle. Due to these properties, hydr
To obtain different concentrations, 3% hydrogen peroxide will be diluted with varying amounts of distilled water. Each different
A drop of each solution will also be observed by analyzing their relative diameter and height sizes on a flat surface table using
Generally, the stronger the intermolecular forces are between molecules, the more they are attracted to one another resulting to
```

```
score=2.10 | chemistry_id=8114130
A DNA polymerase is an enzyme which makes DNA molecules from its nucleotide building blocks. DNA polymerases are essential for DNA repl
DNA polymerases also play key roles in other processes within cells, including DNA repair, genetic recombination, reverse transcription,
```

```
score=2.10 | chemistry_id=8114213
Brief History and Some Definitions
Liquid chromatography (LC) was defined in the early 1900s by the work of the Russian botanist, Mikhail S. Tswett. His pioneering studies fo
Tswett filled an open glass column with particles. Two specific materials that he found useful were powdered chalk [calcium carbonate] and
Tswett coined the name chromatography [from the Greek words chroma, meaning color, and graph, meaning writing—literally, color writing] to
Figure A: Tswett's Experiment
Liquid Chromatography (LC) Techniques
Liquid chromatography can be performed using planar [Techniques 1 and 2] or column techniques [Technique 3]. Column liquid chromatography is
Technique 1. The sample is spotted onto, and then flows through, a thin layer of chromatographic particles [stationary phase] fixed onto the
```

```
score=0.60 | chemistry_id=78642839
Should You Be Taking Zinc Tablets?
Zinc – a metallic element – occurs naturally in dairy foods, nuts, beans, red meat and many types of seafood. It plays an important role in
Symptoms Of Zinc Deficiency
Maintaining a healthy level of zinc is important for your immune system and your ability to heal, as well as sustaining a healthy thyroid a
Zinc isn’t a mineral people think too closely about. It simply does its job and isn’t missed until a person is faced with a zinc deficiency.
```

Figure 6. Examples of labeled chemistry texts with the associated TCS scores.

Reaction Replacement (RxR)

- Duplicate each USPTO 1M reaction four times.
- For three copies, randomly select one molecule (reactant or reagent) to replace.
- Draw a batch of 50 candidate molecules from Enamine50k and compute Tanimoto similarity.
- Swap in the most similar molecule as the replacement.

Reaction Inversion (RxI)

- Take four instances of reactions in USPTO 1M, and invert one reagent with a product for 3 of them.
- The LLM is required to predict which one of the four reactions is still correct.

Reaction True/False (RxTF)

- Derived from the Reaction Replacement dataset.
- Present a single reaction (original or corrupted) and ask the model to judge its chemical correctness.

F.3. Material Tasks Data sources

Chemical Formula Balancing Task (CeB)

- A total of 1500 chemical formulas were selected from the Perovskite Dataset (Jacobsson et al., 2022) to form the data set, and the data set was then enhanced by randomly masking individual stoichiometric coefficients within products or entire product compounds using [MASK].

Conditional Material Generation (CMG)

- We selected 1000 samples from Materials Project (Jain et al., 2013) and extracted the constituent elements from each sample to create our dataset. For example, the compound TeO_2 was decomposed into its constituent elements Te and O to form our training set.,

Binary Compound Structure Relaxation Task (CrR)

- We selected 2,000 binary compound crystal structures from the Materials Project (Jain et al., 2013) across the following categories: Intermetallics, Semiconductors, Oxides, Sulfides, Nitrides, Carbides, Hydrides, Halides, Borides, Silicides, Phosphides, Arsenides, Tellurides, and Selenides. And we applied perturbations to alter the positions of certain atoms and modify the cell parameters of these structures to form our training dataset.

F.4. Resulting data mixture

The pretraining dataset was post-processed using an annotation pipeline to detect each molecule in the texts. For each molecule, the tags ”[START_MOL]” and ”[END_MOL]” were added to enclose it. Similarly, the SMILES were computed for each molecule and added between ”[START_SMILES]” and ”[END_SMILES]” tags after the molecule.

Table 18. MiST Pretraining Dataset Composition

Data Source	Tokens	Proportion
ChemRxiv + S2ORC	1.2B	41.37%
FineWeb (Q4–6)	1.4B	48.27%
PubChem Synthetic	120M	4.14%
Synthetic Reactions	100M	3.44%
CommonCrawl Replay	80M	2.75%
Total	**2.9B**	100%

Supervised fine-tuning was performed on the MiST - Qwen-3B model, primarily using chemistry-specific reasoning and instruction datasets, as follows:

Table 19. MiST SFT Dataset Composition

Data Source	Contents/Size
DeepSeek Rxn Traces	$\sim$7,000 samples
SmolInstruct	I2S, S2I, captioning, gen.
MMLU	350 general + 300 chemistry samples
Chain-of-Thought (CoT)	$\sim$27,000 samples

G. Compute resources

As described in Section E.2 for the GRPO experiments, each training was run for 12 hours on four nodes (with 4 NVIDIA GH200 120GB GPUs or 8 AMD MI250x 128GB GPUs), summing to 16 GPUs and 192 GPU-hours per training. The best hyperparameters and rewards were optimized using a total of 30k GPU-hours with variations in the experimental setups. An additional 10k GPU-hours were used for the final runs, summing to a total of 40k GPU-hours.

H. Additional experimental results

H.1. MiST

We conducted other experiments to evaluate our MiST model's performance on other tasks and in comparison with strong baselines from the literature. In particular, we compare against NatureLM (Xia et al., 2025) and other general-purpose LLMS, on the task of SMILES to IUPAC and IUPAC to SMILES conversion. The results shown below put our MiST model (3B) on par with NatureLM 8B, while approaching the 8x7B MoE variant on IUPAC-to-SMILES conversion.

Table 20. Accuracy for IUPAC-to-SMILES and SMILES-to-IUPAC on benchmark datasets. The best value in each column is shown in bold.

Model	**IUPAC-to-SMILES**	**SMILES-to-IUPAC**
STOUT	0.735	0.565
GPT-4	0.033	0.000
Claude 3 Opus	0.177	0.000
LlaSMol_Mistral	0.701	0.290
NatureLM (1B)	0.476	0.284
NatureLM (8B)	0.679	0.517
Qwen+MiST+SFT	0.682	0.445

H.2. RL

From Table 21, it can be observed that the base model, Qwen-2.5 3B, possesses a degree of domain knowledge in materials science sufficient to generate some valid compositions. However, the relatively low scores suggest that the model is primarily retrieving compositions seen during training or generating valid combinations through rough heuristics. This is further supported by its low SCS, which indicates a limited understanding of compositions at the symbolic level.

The introduction of MiST leads to a significant improvement in SCS, as MiST specifically targets symbolic competence during training. However, since the model was not trained directly on materials science data and has a relatively small parameter size, it likely replaced some of its prior knowledge with representations more aligned with SMILES syntax. This shift contributes to the lower validity and precision scores, reflecting a reduced ability to follow instructions in non-SMILES-based tasks. As a result, the model often fails to generate outputs in the required format, especially when it encounters ambiguous prompts or reaches its maximum output length.

Fine-tuning the MiST model using SFT yields improvements in both SCS and instruction-following ability, as evidenced by higher validity and precision scores. These gains suggest that the model is able to recover some materials science knowledge while refining its symbolic understanding. However, the low novelty score indicates limited generalization, implying that the model is overfitting to training data and struggles to produce truly new compositions.

In comparison, SFT applied directly to the base Qwen-2.5 3B model results in high validity and precision but retains a poor SCS score. This contrast highlights that symbolic competence is primarily achieved through MiST, not SFT. Additionally, the low novelty score again suggests overfitting, as the model continues to rely on memorized examples rather than generating original compositions.

When combining MiST, SFT, and RL, there is a substantial improvement in novelty, indicating that the model is better able to utilize its symbolic understanding and domain knowledge to generate rather than recall compositions. This suggests that while base models have weak symbolic competence, MiST significantly enhances this capability. Though MiST initially reduces instruction-following ability due to longer and more complex outputs, SFT helps regain this ability for specific tasks. Ultimately, RL fine-tuning balances symbolic competence with domain-specific generation, enabling the model to produce valid, precise, and novel compositions using the specified elements.

Table 21. CMG = Conditional Material Generation.

Model	SCS ↑	CCS ↑	Validity ↑	Precision ↑	Novelty ↑
Qwen-2.5 3B	0.122	0.828	58.6	68.0	74.8
+MiST	0.989	0.795	1.2	0.67	84.6
+SFT	1.142	0.785	34.8	38.5	49.2
+RL	0.893	0.777	73.8	97.1	91.3
Ablations					
no MiST + SFT	0.199	0.824	87.4	93.9	60.2

Table 22. CrR = Binary crystal stucture relaxation, CeB = Chemical formula balancing.

	Metrics		Reasoning tasks	
Model	SCS ↑	CCS ↑	CrR ↑	CeB ↑
Qwen-2.5 3B	0.346	0.834	0	1.2
+MiST	0.355	0.795	0	26
+SFT	0.528	2.361	16.2	29.2
MatSci Tasks				
+RL(CrR)	0.447	2.599	65	—
+RL(CeB)	1.653	0.666	—	47
Ablations				
no MiST + SFT(CrR)	0.573	2.652	12.6	–
no MiST + SFT(CeB)	1.494	0.849	—	45

In contrast to the findings observed in the Conditional Material Generation task, we did not detect any notable improvement in CCS after introducing MiST to the Binary Crystal Structure Relaxation task. This discrepancy arises because the Binary Crystal Structure Relaxation task specifically emphasizes structural relaxation, a domain not directly targeted by MiST training. Consequently, MiST did not enhance the model's chemical competence related to structural relaxation.

However, subsequent fine-tuning via SFT successfully incorporated relevant domain knowledge into the model, resulting in substantial performance improvements on the task. This step notably increased the model's capability to accurately execute structural relaxations, which was previously limited. Moreover, further refinement through reinforcement learning (RL) effectively enhanced the model's success rate, demonstrating that the integration of RL optimally balances domain-specific expertise with task-oriented performance improvements.

We further conducted an additional analysis across all 200 test set datapoints, and observed that the model performed comparably across the five crystal systems included in the test set.

Table 23. Summary of Crystal Systems for the MiST + SFT + RL (CrR) Model. This table presents a detailed breakdown of the performance (accuracy) of the MiST + SFT + RL (CrR) task, as shown in the Table, evaluated separately across different crystal systems.

Crystal System	Average Accuracy	Total Samples
Tetragonal system	0.6383	47
Orthorhombic system	0.6897	29
Hexagonal system	0.6250	72
Trigonal system	0.6572	35
Monoclinic system	0.7143	7
Cubic system	N/A	N/A
Triclinic system	N/A	N/A

We illustrate the capability of our Mist + SFT + RL model to reduce the inner energy of a perturbed, unstable ZnSe-P4_nmm crystal structure within 10 steps, where the stable state of the ZnSe-P4_nmm crystal has an inner energy of -2.94069766998291.

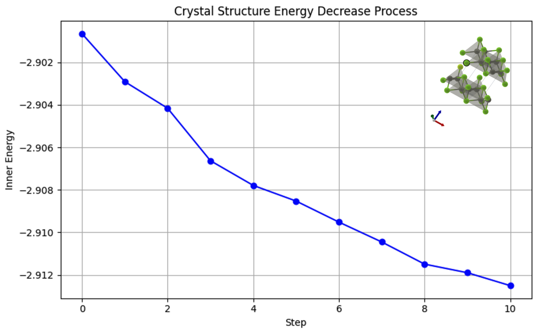
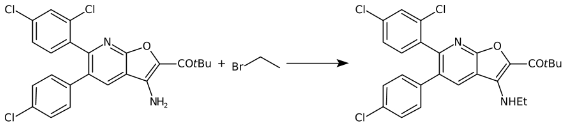


Figure 7. Graph demonstrating the relaxation of the ZnSe-P4_nmm crystal structure with the Mist + SFT + RL model

I. Model analysis

Here, we provide some analysis of the models' behavior. For the following example reaction:



Question

```
Reason and predict the correct
product in SMILES notation
from the following reaction
CC(C)(C)C(=O)c1oc2nc
(-c3ccc(Cl)cc3Cl)c(-c3ccc(Cl)cc3)
cc2c1N.CCBr
```

Correct answer

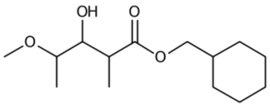
```
CCNc1c(C(=O)C(C)(C)C)oc2nc(-c3ccc(Cl)
cc3Cl)c(-c3ccc(Cl)cc3)cc12
```

After RL training, although the output format is correct, the base model Qwen2.5-3B (+RL) fails to predict the correct product of this reaction, with simple reasoning and incorrect reaction type (esterification reaction). On the other hand, the variant with SFT exhibits a more detailed reasoning that delves into the functional groups of the precursors. However, it hallucinates the presence of sodium hydride and the hydroxyl group, which are not present in the input reagents, leading to an incorrect result. This could be attributed to the lack of SMILES symbolic understanding. Lastly, the model that underwent the full pipeline (MiST + SFT + RL) successfully reasons the correct product with reasonable description of the reactants' structure and mechanism, showing better symbolic understanding and Chemical knowledge.

The full responses of these models along with the depiction of the predicted product molecules are given below:

Table 24. Qwen2.5-3B + RL [Wrong].

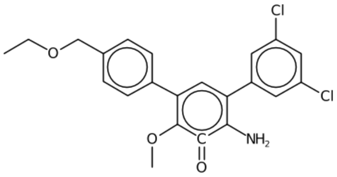
Qwen2.5-3B + RL [Wrong]



<think>The given reactants are a carboxylic acid and a bromoalkane. The reaction between these two compounds is an esterification reaction, where the carboxylic acid reacts with the alkyl halide to form an ester and a halide. The product of this reaction is an ester with the general formula R-COOR'. In this case, the product is a carboxylic acid ester with the general formula R-COOR', where R is the alkyl group and R' is the alkyl group from the bromoalkane.
</think>
<answer>COC(C)C(O)C(C)C(=O)OCC1CCCCC1</answer>

Table 25. Qwen2.5-3B + SFT + RL [Wrong].

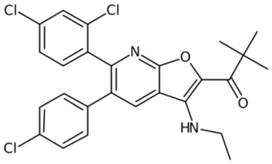
Qwen2.5-3B + SFT + RL [Wrong]



<think>The reaction involves the alkylation of a phenolic oxygen in the first compound using ethyl bromide CCBr under basic conditions. Here's the step-by-step mechanism:
1. **Deprotonation:** Sodium hydride [H-].[Na+] deprotonates the hydroxyl group (-OH) on the benzene ring, generating a phenoxide [O-] ion, a strong nucleophile.
2. **SN2 Attack:** The phenoxide [O-] attacks the electrophilic carbon in ethyl bromide CCBr, displacing Br- and forming an ether C{O{C bond.
3. **Product Formation:** The final product retains the fused aromatic core, with the hydroxyl replaced by an ether-linked ethyl group. Other functional groups remain unchanged.
Product SMILES:
CCOCc1ccc(-c2cc(-c3cc(Cl)cc(Cl)c3)c(N)c(=O)c2OC)cc1
This is a Williamson ether synthesis: a phenoxide displacing a halide to form an ether.
</think>
<answer1ccc(-c2cc(-c3cc(Cl)cc(Cl)c3)c(N)c(=O)c2OC)cc1</answer>

Table 26. Qwen2.5-3B + MiST + SFT + RL [Correct].

Qwen2.5-3B + MiST + SFT + RL [Correct]



<think>The reaction involves the alkylation of a primary amine group in the first molecule by bromoethane CCBr. Here's the step-by-step mechanism:
1. **Reactant Analysis**: - First molecule: a bicyclic compound with a tert-butyl group (CC(C)(C)C(=O)), a furan-like ring, and two chlorobenzene substituents. - Second molecule: bromoethane CCBr, where Br is a good leaving group.
2. **Mechanism**: - The lone pair on the primary amine nitrogen acts as a nucleophile. - An SN2 reaction with bromoethane displaces Br−, forming a secondary amine bonded to an ethyl group (CC).
3. **Product Structure**: - The amine nitrogen in the bicyclic core is now NCC. - All other substituents remain unchanged.
Final SMILES:
CC(C)(C)C(=O)c1oc2nc(-c3ccc(Cl)cc3Cl)c(-c3ccc(Cl)cc3)cc2c1NCC This is a selective alkylation of the amine via Williamson-like conditions.
</think>
<answer>CC(C)(C)C(=O)c1oc2nc(-c3ccc(Cl)cc3Cl)c(-c3ccc(Cl)cc3)cc2c1NCC </answer>

J. Reproducibility

All the code used to produce the results presented in this work can be found under `https://figshare.com/account/articles/29132657`. The continued pre-training and supervised fine-tuning, as described in Section 3 and Appendix E, have been conducted using the megatron library (see `https://github.com/huggingface/megatron`). The configuration files and datasets used are released at `https://figshare.com/account/articles/29132657`.

Table of Released Assets

Table 27. List of digital assets released with this work, including usage instructions and licensing/citation information. Note: All assets are hosted anonymously on Figshare for double-blind review.

Asset	**Usage Instructions**	**License/Citation Info**	**Location/URL**
Source code	Download and unzip. See `README.md` for installation and experiment scripts (`run_train.py`).	MIT License. Please cite this paper.	`https://figshare.com/account/articles/29132657`
Model checkpoints	Download the archive. Full instructions in `README.md`.	MIT License. Please cite this paper.	`https://figshare.com/account/articles/29132657`
Datasets (pretraining/fine-tuning splits)	Download files; load as a HuggingFace Dataset.	For research use only. Cite the original dataset and this paper.	`https://figshare.com/account/articles/29132657`
Training configs	Config YAML files for megatron available as `.yaml`; pass as argument to megatron CLI.	MIT License.	`https://figshare.com/account/articles/29132657`