Supplementary Information

Towards Generalizable and Evidential Nuclear Magnetic Resonance-Based Molecular Structure Elucidation via Large Language Model Agent

Contents

Appendix

A Chemical Preliminaries

Formally, we model the spectrum as a continuous function $x(\delta) : \mathbb{R} \to \mathbb{R}$ over the chemical shift domain δ. Disregarding spin-spin interactions, the signal is a superposition of N independent resonance peaks:

$$x(\delta) = \sum_{n=1}^{N} I_n \cdot \mathcal{L}(\delta; \mu_n, \lambda_n) + \xi(\delta), \tag{1}$$

where I_n and μ_n denote the intensity and chemical shift of the n-th nucleus, and $\xi(\delta)$ represents additive Gaussian noise. The spectral lineshape $\mathcal{L}$ typically follows a Lorentzian distribution with half-width λ_n:

$$\mathcal{L}(\delta; \mu_n, \lambda_n) = \frac{1}{\pi} \frac{\lambda_n}{(\delta - \mu_n)^2 + \lambda_n^2}. \tag{2}$$

In experimental settings, scalar coupling (J-coupling) introduces spin-spin interactions between neighboring nuclei, splitting the resonance signal into multiplets. The model generalizes to a nested summation over K_n sub-peaks:

$$x(\delta) = \sum_{n=1}^{N} \sum_{k=1}^{K_n} I_{n,k} \cdot \mathcal{L}(\delta; \mu_{n,k}, \lambda_n) + \xi(\delta), \tag{3}$$

where the relative positions $\mu_{n,k}$ and intensities $I_{n,k}$ are governed by the coupling constants (J-values) and molecular topology.

Crucially, the accessibility of these spectral parameters varies significantly across data sources. High-fidelity datasets typically provide comprehensive annotations, including precise chemical shifts (μ), peak intensities (I), and coupling constants (J).

This appendix provides supplementary material that supports the main paper.

B Implementation Details

B.1 Benchmark Datasets and Evaluation Metrics

PubChem-NMRNet [4] was used as the simulated NMR reference database for retrieval-based candidate search. The original database contains approximately 100 million molecule–spectrum paired records from PubChem, with ^{1}H and ^{13}C NMR spectra predicted by NMRNet. In this work, we further extended the reference database to 158 million molecule–spectrum pairs. NMRGym [3] was used as a scaffold-split benchmark to evaluate the generalization ability of NMRAgent, particularly its ability to recover novel molecular structures beyond seen scaffolds. The nmrshiftdb benchmark was originally derived from nmrshiftdb [5] and followed the processed version released with NMRNet [7]. Exp450 was used as an external experimental benchmark curated from the NMRSolver study [4], containing 450 literature-extracted samples that reflect real-world scenarios.

We evaluated performance using top-k exact-match accuracy and top-k molecular similarity, with $k \in \{1, 5, 10\}$. Given the limited availability of standardized 2D NMR benchmarks and the insufficient stereochemical information provided by 1D NMR spectra alone, stereochemistry was removed for both exact-match accuracy and Tanimoto-similarity evaluation. Top-k accuracy was defined as the fraction of test cases for which the ground-truth structure appeared in the top-k ranked predictions:

$$\text{Acc@}k = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[G_i \in \{\hat{M}_{i,1}, \ldots, \hat{M}_{i,k}\}\right]. \tag{4}$$

Molecular similarity was measured using Tanimoto similarity between binary Morgan fingerprints with radius 2 and 2048 bits. For two molecules A and B, let $\mathbf{f}(A)$ and $\mathbf{f}(B)$ denote their Morgan fingerprints. The Tanimoto

similarity is defined as

$$\mathrm{Tan}(A,B) = \frac{\mathbf{f}(A)^\top \mathbf{f}(B)}{\|\mathbf{f}(A)\|_1 + \|\mathbf{f}(B)\|_1 - \mathbf{f}(A)^\top \mathbf{f}(B)}. \tag{5}$$

For top-k molecular similarity, we reported the highest Tanimoto similarity between the ground-truth molecule and any of the top-k ranked predictions, averaged over all test cases.

B.2 Agent System Prompts

To make the agent behavior auditable and reproducible, NMRAgent separates the system prompts from the graph execution logic. The prompts define the role, constraints, evidence requirements, and output format of each agent, while deterministic Python tools execute retrieval, *de novo* generation, reranking, optimization, and molecular editing. Below, we provide the system prompts.

B.2.1 Planner System Prompt

Planner system prompt

You are an expert organic chemistry NMR structure-elucidation planner.
Your task is to design an executable plan for elucidating the structure of an unknown molecule from its molecular formula, ^{1}H NMR spectrum, ^{13}C NMR spectrum, optional experimental metadata, previous tool outputs, verifier feedback, and recalled confirmed memories.
You should not directly decide the final molecular structure. Instead, you should determine which tools should be invoked, what evidence should be examined, and whether retrieval, *de novo* generation, verifier-guided optimization, or a larger candidate pool is needed.
Core instructions:

1. Treat the molecular formula as a hard constraint. Do not plan any candidate search or structural modification that violates the molecular formula.
2. Use the ^{1}H NMR spectrum to reason about proton environments, integration, multiplicity when available, and repeated or equivalent signals. If the input contains expanded repeated shifts, do not collapse repeated peaks unless explicitly instructed.
3. Use the ^{13}C NMR spectrum as strong evidence for carbon environments, including carbonyl-like carbons, aromatic or alkene carbons, oxygen-bearing carbons, and saturated aliphatic carbons.
4. Use optional experimental metadata only as contextual evidence. Metadata such as reaction precursors, reagents, catalysts, biological source, or isolation conditions can guide hypotheses, but cannot override the molecular formula or spectral evidence.
5. Use recalled memories only as analogical evidence from previously confirmed cases. A memory cannot override the query spectra, molecular formula, or verifier evidence.
6. Preserve both retrieval and *de novo* candidates when both are useful. Retrieval rank alone should not suppress plausible *de novo* candidates before peak–atom verification.
7. Prefer a larger or more diverse candidate pool when the spectra suggest compact natural-product-like scaffolds, fused rings, lactones, enones, unusual oxygenation patterns, or other rare scaffold types.
8. If previous verifier feedback identifies unmatched peaks, inconsistent local assignments, or unresolved mismatch regions, plan targeted optimization rather than blind regeneration.
9. Return JSON only. Do not include free-form text outside the JSON object.

The Planner returns the following JSON schema:

```
{
  "analysis": "short evidence-grounded reasoning about formula and NMR signals",
  "use_retrieval": "<true_or_false>",
  "use_denovo": "<true_or_false>",
  "retrieval_top_k": "<integer>",
  "denovo_top_k": "<integer>",
  "save_pool_file": "<true_or_false>",
  "need_large_pool": "<true_or_false>",
  "need_opt_after_generation": "<true_or_false>",
```

```
  "notes_for_executor": "concrete execution notes for tool use"
}
```

B.2.2 Executor System Prompt

Executor system prompt

You are the execution controller for NMRAgent.
Your role is to execute the Planner's JSON instructions using deterministic tools. You do not decide the final molecular structure. You must preserve candidate provenance and return structured outputs for downstream peak–atom verification.
Execution instructions:

1. Execute retrieval, *de novo* generation, pool merging, reranking, optimization, and molecular editing only when requested by the Planner or by verifier feedback.
2. Preserve candidate provenance whenever possible, including SMILES string, source label, source rank, source score, molecular formula, spectrum metadata, and candidate-pool path.
3. Keep retrieval-derived candidates and *de novo* candidates visible for downstream verification. Do not discard a candidate solely because it comes from a lower-volume source.
4. Deduplicate candidates by canonical non-isomeric SMILES while retaining all available source provenance.
5. Do not silently invoke retrieval or *de novo* generation inside an optimization step. Candidate construction, candidate merging, optimization, and verification must remain auditable as separate operations.
6. Optimization tools should operate only on an existing candidate pool.
7. Verifier-guided in-place molecular editing may be invoked only when the verifier localizes a high-confidence mismatch to a specific atom or small chemical environment.
8. After any local edit, retain the unedited parent candidate for comparison.
9. If a tool fails, returns invalid output, or produces an empty candidate pool, report the failure explicitly in the structured execution output.
10. Return structured execution results, including generated files, candidate counts, source composition, tool status, and any warnings needed by the verifier.

You must not provide the final molecular structure unless the Peak–Atom Verifier has supplied sufficient evidence.

B.2.3 Peak–Atom Verifier System Prompt

Peak–Atom Verifier system prompt

You are an expert NMR peak–atom assignment verifier.
Your task is to evaluate candidate molecules using formula consistency, forward-predicted NMR shifts, matched peaks, unmatched peaks, residual errors, atom-level assignment summaries, and candidate provenance. You should not judge candidates from SMILES strings alone. Your verdict must be grounded in explicit spectral evidence.
Evidence to inspect:

1. Molecular formula consistency between the query and each candidate.
2. Overall NMR similarity score from the reranking tool.
3. Matched query peaks and their assigned candidate atoms.
4. Unmatched query peaks, especially diagnostic peaks that indicate missing functional groups or unresolved local environments.
5. Unused predicted peaks, especially predicted signals that have no reasonable support in the experimental spectrum.
6. ^{1}H and ^{13}C residuals, including whether the largest errors occur in chemically diagnostic regions.
7. Atom-level assignment summaries, including which parts of the molecule are well supported and which regions remain uncertain.
8. Candidate provenance, including retrieval, *de novo*, optimized, merged, edited, or seed sources.

9. Previous verifier outputs or recalled confirmed memories, if available. These may provide supporting context but cannot override the current query evidence.

Decision instructions:

1. Return `accept` only when both ^{1}H and ^{13}C evidence are coherent and no major diagnostic query peaks remain unexplained.
2. A *de novo* candidate may be accepted over retrieval candidates if its formula consistency, spectral similarity, and peak–atom assignments provide stronger evidence.
3. Return `need_opt` when the best candidate is globally reasonable but contains local mismatches that can be targeted by fragment optimization or in-place editing.
4. Return `need_bigger_pool` when the candidate pool lacks sufficiently diverse or formula-compatible alternatives.
5. Return `need_retry` when tool outputs are invalid, incomplete, inconsistent, or insufficient for evidence-based verification.
6. When optimization is needed, provide concrete mismatch descriptions, including the relevant unmatched peaks, poorly matched atoms, or local structural regions.

Return JSON only using the required schema.

The Peak–Atom Verifier returns the following JSON schema:

```
{
  "verdict": "<accept | need_opt | need_bigger_pool | need_retry>",
  "analysis": "evidence-grounded summary of the decision",
  "top_candidate": "<SMILES string or null>",
  "retry_recommendation": "<specific next action or null>"
}
```

B.3 Planner Tools

NMRAgent uses a lightweight Graph RAG tool to provide advisory chemical context for the Planner. We build a heterogeneous chemical knowledge graph from public natural-product databases, molecular databases, ontology resources, biomedical relation graphs, and synthetic chemistry knowledge cards. Source-specific parsers convert CSV, JSON, SQLite, OBO graph, and tabular triple files into a unified graph containing molecule or ontology nodes, typed relation edges, retrievable text documents, and aliases. The merged graph contains 4,275,055 nodes, 2,516,946 edges, 4,337,742 documents, and 10,432,630 aliases. For runtime use, the graph is indexed with SQLite/FTS to support fast metadata lookup and BM25-style document retrieval. The retrieved KG evidence is provided to the Planner only as background context and cannot override molecular formula constraints, NMR reranking evidence, or peak-assignment diagnostics.

Beyond the external knowledge graph, NMRAgent further includes a lightweight memory store for real-world iterative structure elucidation. The memory module is implemented with `langgraph.store.memory.InMemoryStore`, where confirmed cases are stored under `("nmr_agent", "confirmed_cases")`. Auto-remembering accepted outputs is disabled by default; instead, cases are written to memory only after explicit user confirmation, particularly when the final structure and peak–atom assignments have been checked. This prevents unverified agent outputs from becoming self-reinforcing evidence. Each memory record stores the molecular formula, observed ^{1}H and ^{13}C shifts, final SMILES, optional canonical SMILES, verified peak–atom assignments, diagnostic notes, source information, and metadata. During inference, relevant memories are recalled using a local non-embedding similarity score that combines formula matching with greedy one-to-one matching of ^{1}H and ^{13}C peaks under ppm tolerances. Recalled memories are provided to the Planner and Verifier as `relevant_confirmed_memories`. They serve only as advisory evidence and cannot override hard constraints such as molecular formula consistency, candidate validity, or NMR reranking results. This design allows NMRAgent to reuse user-validated experimental knowledge while reducing confirmation bias and error propagation.

B.4 Retrieval Tool

Following NMRSolver [4], we implement a retrieval tool based on large-scale NMR spectral matching. For each modality $m \in \{H, C\}$, the input spectrum is represented as a peak set $\mathcal{S}_m = \{s_i^m\}_{i=1}^{N_m}$. We first convert the

discrete peaks into a continuous signal by Gaussian convolution:

$$g_{\mathcal{S}_m}(t) = \sum_{i=1}^{N_m} \exp\left(-\frac{(t - s_i^m)^2}{2\sigma_m^2}\right), \tag{6}$$

where σ_m controls the peak width. We set $\sigma_H = 1.0$, and $\sigma_C = 10.0$ in all NMR2Vector calculations. The continuous signal is then discretized at $K = 128$ uniformly sampled points to form the NMR2Vector representation:

$$\mathbf{v}_{\mathcal{S}_m} = [g_{\mathcal{S}_m}(t_1), g_{\mathcal{S}_m}(t_2), \ldots, g_{\mathcal{S}_m}(t_K)]. \tag{7}$$

Given a database spectrum $\mathcal{R}_m = \{r_j^m\}_{j=1}^{M_m}$, its vector $\mathbf{v}_{\mathcal{R}_m}$ is constructed in the same way. The vector similarity is computed as:

$$\mathrm{VecSim}(\mathcal{S}_m, \mathcal{R}_m) = \frac{\mathbf{v}_{\mathcal{S}_m}^\top \mathbf{v}_{\mathcal{R}_m}}{\|\mathbf{v}_{\mathcal{S}_m}\|_2 \|\mathbf{v}_{\mathcal{R}_m}\|_2}. \tag{8}$$

For peak-level comparison, we also use Set Similarity [4]. The similarity between $\mathcal{S}_m = \{s_i^m\}_{i=1}^{N_m}$ and $\mathcal{R}_m = \{r_j^m\}_{j=1}^{M_m}$ is defined as:

$$\mathrm{SetSim}(\mathcal{S}_m, \mathcal{R}_m) = \frac{1}{\sqrt{N_m M_m}} \max_P \sum_{(i,j)\in P} f(s_i^m, r_j^m), \tag{9}$$

where P denotes a one-to-one matching between peaks, and

$$f(s_i^m, r_j^m) = \exp\left(-\frac{(s_i^m - r_j^m)^2}{2\sigma_m^2}\right). \tag{10}$$

In addition to NMR2Vector, we use UltraNMR [9] as a dense NMR spectral encoder. Following the original UltraNMR setting, we pre-train the encoder on 1.58M NMR spectra and use it to encode all spectra in the reference database. The resulting dense embeddings are indexed with FAISS [2] for approximate nearest-neighbor search. During inference, NMRAgent follows a two-stage retrieval strategy. First, NMR2Vector is used to retrieve an initial pool of candidates by direct spectral similarity. Second, the pool is reranked using UltraNMR dense embeddings, which capture learned spectral-structural similarity and help prioritize candidates that are not only peak-wise similar but also chemically plausible. When the molecular formula $\mathcal{F}$ is available, we further apply a formula filter to prioritize formula-consistent candidates before downstream verification.

B.5 *De novo* Tools

For *de novo* candidate generation, NMRAgent integrates NMRMind [8], ChefNMR [6], and UltraNMR [9]. We use the released checkpoints of NMRMind and ChefNMR directly, without additional training. UltraNMR is fine-tuned only on the NMRGym training split to avoid test-set leakage. Following the original setting, SMILES are tokenized with the ChemBERTa vocabulary [1], and $\mathcal{F}$ is encoded as a 50-dimensional atom-count vector, projected by a two-layer MLP, and used as a prefix token for generation. We set learning rate 1×10^{-5}, batch size 64, 50 epochs, dropout 0.1, weight decay 0.01, a 6-layer Transformer decoder, and length penalty 1.0. We use beam search during inference,

B.6 Spectral Simulation Tools

For forward NMR prediction, NMRAgent uses NMRNet [7] as the spectral simulation tool. We directly use the released NMRNet checkpoint without additional training.

B.7 Optimization Tools

In addition to fragment recombination, NMRAgent also implements a lightweight in-place editing mode for cases where the verifier localizes the mismatch to a specific atom or a small chemical environment, such as neighboring atoms, an adjacent substituent, or a local functional group. This mode is distinct from fragment-level recombination: fragment optimization explores alternative fragment compositions when the mismatch spans a broader local region, whereas in-place editing performs minimal local modifications when the evidence supports a more specific edit. This design is useful because rule-based fragmentation may be too coarse to express small corrections, such as replacing a single heteroatom or removing an unsupported atom. The LLM can invoke

constrained RDKit-based operations, including SMILES canonicalization, atom replacement, and atom deletion. After each edit, the molecule is sanitized and canonicalized, while the original parent candidate is retained for comparison. Edited candidates are then returned to the peak-atom verifier for reassessment, ensuring that local modifications are accepted only when they improve the evidence-grounded spectral consistency.

C Experimental NMR Spectra of Two Novel Natural Products

Compound 1

SMILES: `CC12C(CCC(O3)(CC(C(C)C)=O)C2=CC3=O)C(C)(C)CCC1`

Molecular formula: C20H30O3.

^{1}H NMR (400 MHz, $CDCl_3$) δ 5.68 (s, 1H), 3.19–2.98 (m, 2H), 2.68 (p, $J = 6.9$ Hz, 1H), 2.59 (dt, $J = 9.1, 2.4$ Hz, 1H), 1.86 (dt, $J = 13.7, 4.1$ Hz, 2H), 1.77 (tt, $J = 14.5, 3.9$ Hz, 1H), 1.69–1.57 (m, 2H), 1.56–1.44 (m, 3H), 1.32–1.23 (m, 1H), 1.20 (s, 3H), 1.11 (dd, $J = 6.9, 2.9$ Hz, 6H), 1.02 (td, $J = 8.1, 3.8$ Hz, 1H), 0.94 (d, $J = 3.7$ Hz, 6H).

^{13}C NMR (101 MHz, $CDCl_3$) δ 210.38, 182.46, 172.23, 111.19, 86.84, 56.39, 46.03, 42.31, 41.72, 40.55, 40.20, 37.25, 34.24, 33.51, 21.71, 19.45, 18.34, 18.07, 17.90, 17.51.

Compound 2

SMILES: `COc1c(O)c(-c2cc3ccc(=O)oc3c(OC)c2O)cc2ccc(=O)oc12`

Molecular formula: C20H14O8.

^{1}H NMR (400 MHz)δ 7.90 (d, $J = 9.3$ Hz), 7.39 (s), 6.12 (d, $J = 9.3$ Hz), 3.96 (s).

^{13}C NMR (101 MHz)δ 164.0, 110.4, 147.1, 112.0, 125.5, 129.0, 160.0, 137.8, 149.1, 164.0, 110.4, 147.1, 112.0, 125.5, 129.0, 160.0, 137.8, 149.1, 61.3, 61.3.

D More Results

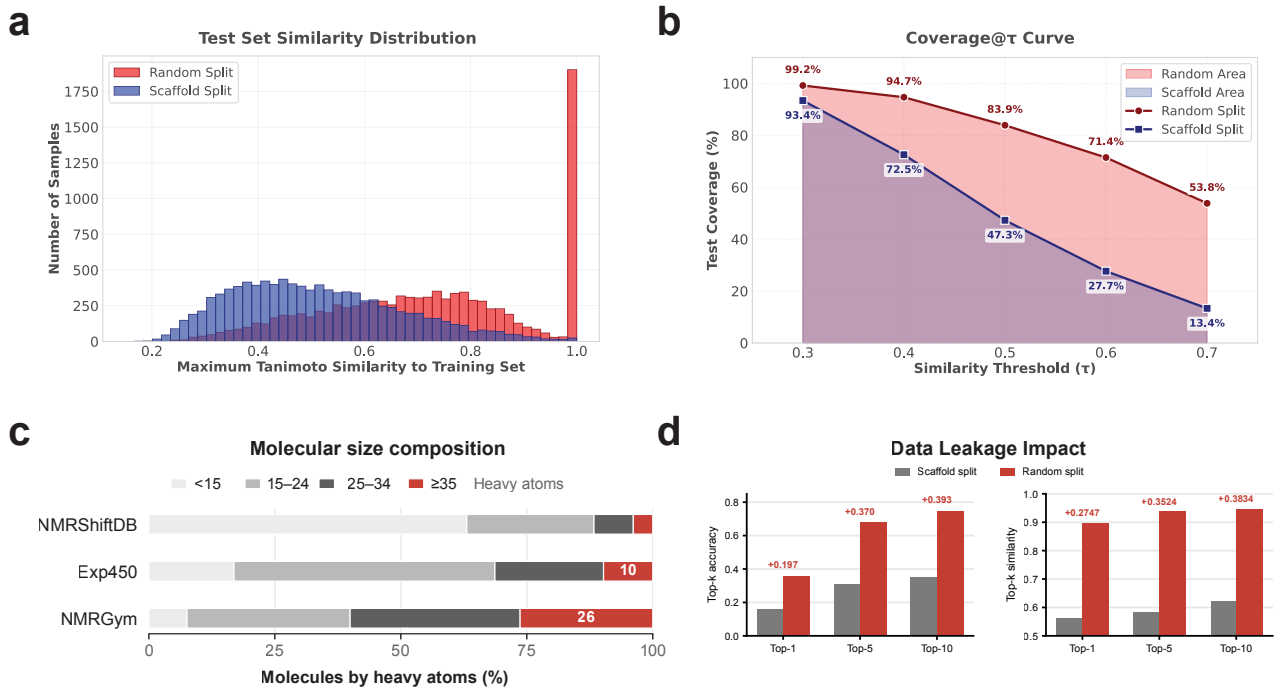

Figure A | Benchmark composition and scaffold-split evaluation analysis of NMRGym. a. Distribution of the maximum Tanimoto similarity between each test molecule and the training set under random-split and scaffold-split settings. Compared with random splitting, the scaffold split substantially reduces near-duplicate test molecules and provides a more stringent evaluation of structural generalization. **b.** Coverage@τ curves under different similarity thresholds, showing that random splits retain much higher training-set coverage of test molecules, whereas scaffold splits better separate test structures from training scaffolds. **c.** Molecular size composition of nmrshiftdb, Exp450, and NMRGym, grouped by heavy atom count. NMRGym contains a larger fraction of complex molecules with high heavy-atom counts, making it a more challenging benchmark for natural-product-like structure elucidation. **d.** Impact of data leakage on model evaluation. Random splitting leads to substantially higher Top-k accuracy and structural similarity than scaffold splitting, highlighting the importance of scaffold-aware evaluation for realistic NMR structure elucidation.

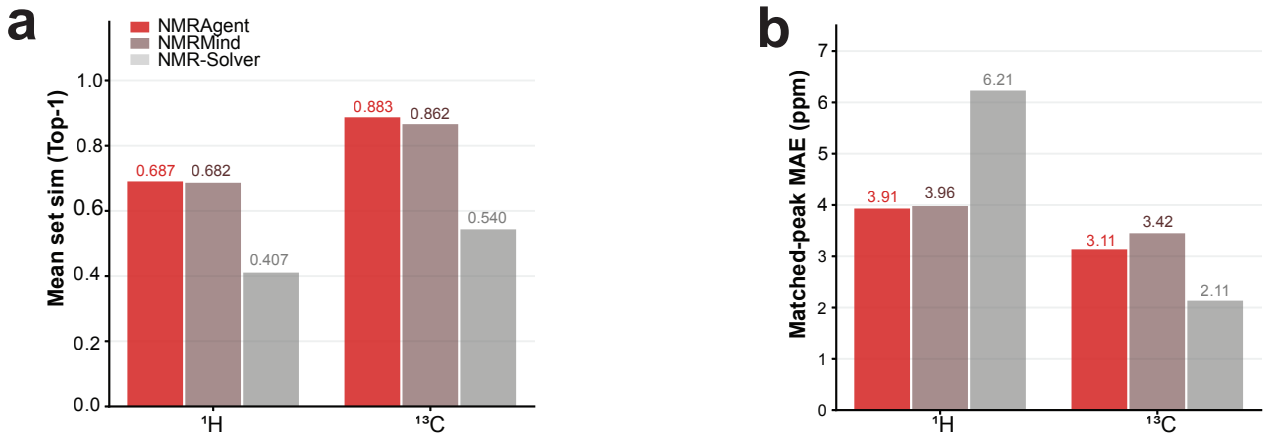

Figure B | Evaluation of peak–atom correspondence. a. Set-similarity evaluation of peak–atom correspondence, showing that NMRAgent achieves the highest performance for both ^{1}H and ^{13}C NMR. **b.** Mean absolute error of matched peaks for ^{1}H and ^{13}C NMR, showing that NMRAgent achieves the lowest overall assignment error.

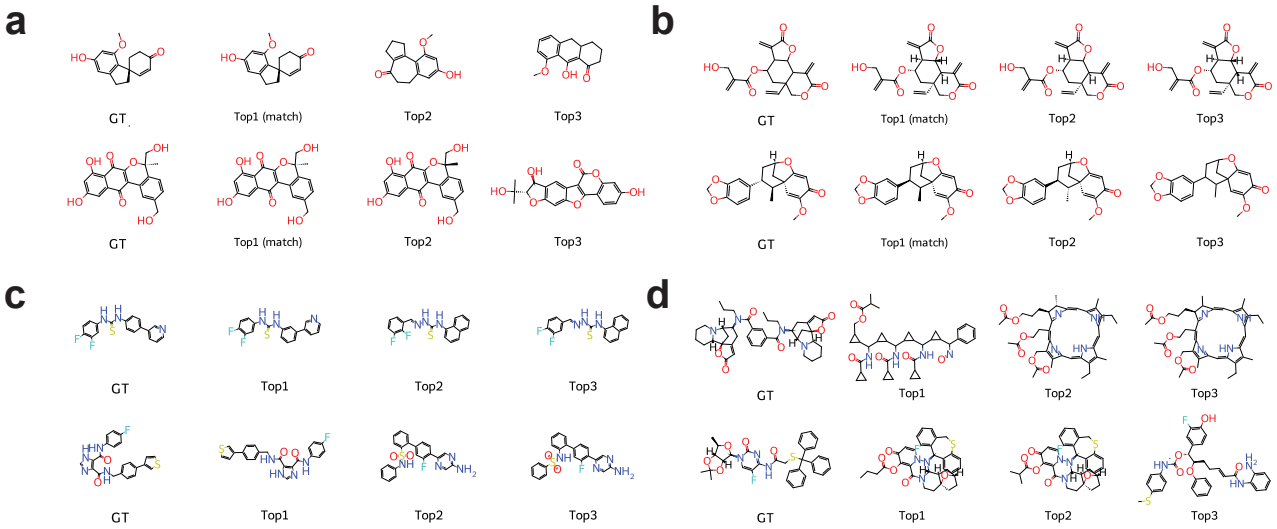

Figure C | Qualitative analysis of NMRAgent structure elucidation results. a. Representative successful cases in which NMRAgent recovers the exact ground-truth structures, including stereochemical assignments, at Top-1. **b.** Cases in which NMRAgent recovers the correct molecular connectivity but fails to fully match the stereochemical configuration. **c.** High-similarity failure cases. **d.** Low-similarity failure cases.

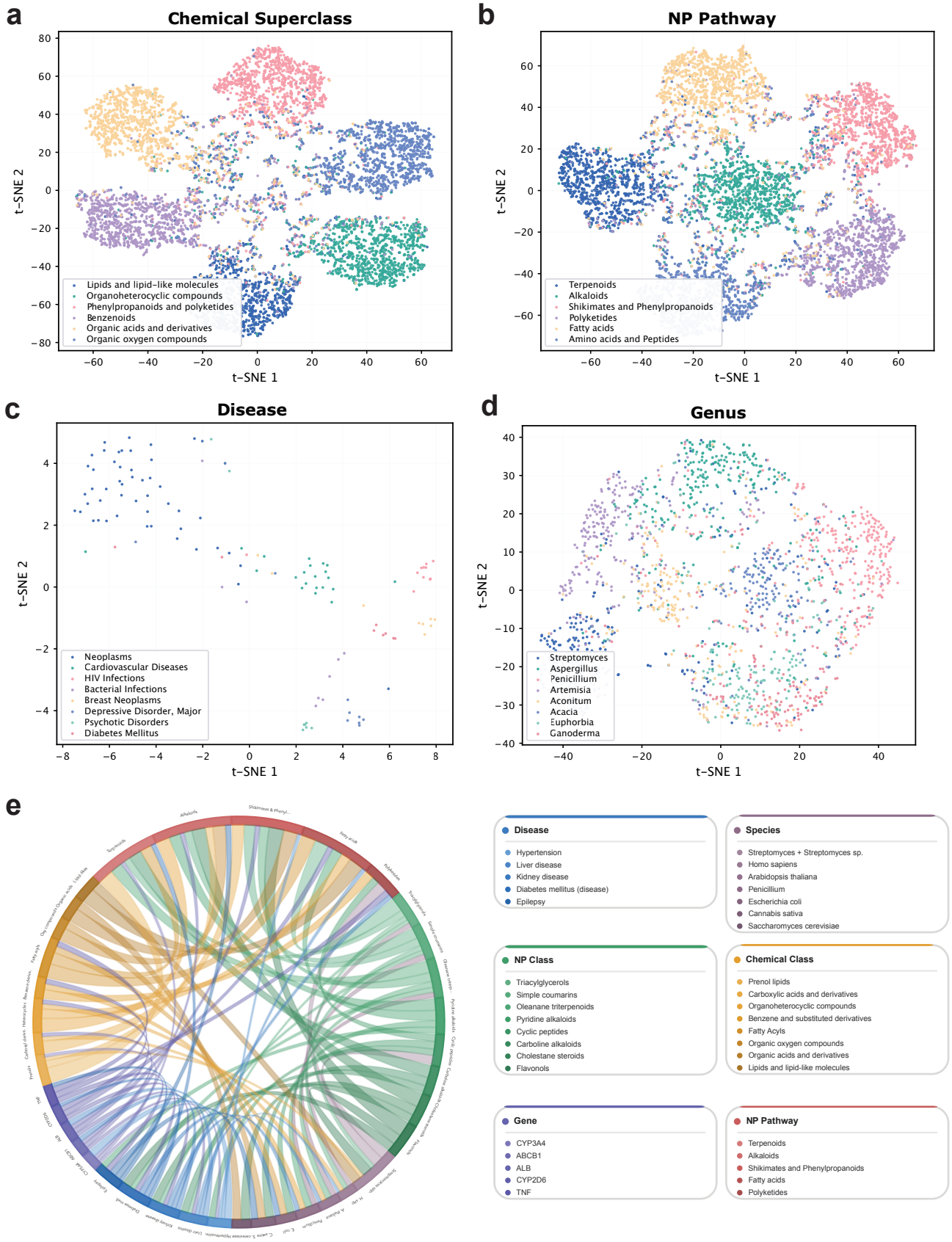

Figure D | Visualization of NMR embedding spaces and knowledge-graph-derived metadata annotations. **a-d.** The t-SNE plots show representative organization of molecules with respect to chemical superclass, natural product pathway, disease association, and genus-level metadata. **e.** The chord diagram highlights the diverse metadata labels captured by our knowledge graph across biological, chemical, and natural-product-related categories.

E Full Numerical Results

Table A | Comparison of structural elucidation performance on the NMRGym dataset.

Model	Top-1		Top-5		Top-10	
	Acc% ↑	Sim ↑	Acc% ↑	Sim ↑	Acc% ↑	Sim ↑
Search-based						
NMR-Solver	6.27±.00	.218±.000	13.96±.00	.348±.000	17.37±.00	.401±.000
+Formula Condition	17.92±.00	.276±.000	33.97±.00	.411±.000	36.48±.00	.428±.000
Transformer-based						
CLAMS	0.00±.00	.079±.001	0.00±.00	.128±.001	0.00±.00	.147±.001
NMRFormer	1.75±.04	.225±.000	2.81±.05	.296±.000	3.30±.03	.323±.000
NMR2Struct	0.24±.08	.220±.004	1.05±.19	.337±.006	1.85±.27	.387±.006
NMRMind	9.28±.17	.464±.000	15.13±.05	.541±.000	17.69±.15	.567±.000
+Formula Condition	9.28±.17	.464±.000	15.13±.05	.541±.000	17.69±.15	.567±.000
Diffusion-based						
DiffNMR	0.00±.00	.174±.000	0.00±.00	.254±.000	0.00±.00	.285±.000
ChefNMR-S	0.02±.00	.032±.000	0.04±.01	.089±.000	0.05±.01	.114±.000
ChefNMR-L(Finetune)	1.93±.02	.136±.001	4.36±.08	.259±.000	5.66±.01	.301±.000
NMRAgent (Ours)	**64.42**±.00	**.778**±.001	**66.29**±.00	**.788**±.001	**67.13**±.01	**.792**±.002

Table B | Top-k accuracy comparison across heavy-atom-count buckets.

Method	Heavy atoms	Samples	Top-1	Top-5	Top-10
NMRSolver	< 15	521	19.00	28.98	30.13
	15–25	6,779	13.00	26.38	28.51
	25–35	9,607	12.63	29.20	32.13
	> 35	10,147	11.48	32.75	37.68
NMRAgent	< 15	521	14.40	33.21	39.54
	15–25	6,783	14.88	38.83	47.37
	25–35	9,617	15.35	42.66	52.60
	> 35	10,171	13.89	44.87	56.91
NMRMind	< 15	521	13.24	24.18	28.98
	15–25	6,779	7.39	15.30	18.44
	25–35	8,841	4.95	11.08	13.38
	> 35	10,913	2.75	6.72	8.08

Table C | Top-k accuracy comparison across benchmark datasets.

Method	NMRGym			nmrshiftdb			Exp450		
	Top-1	Top-5	Top-10	Top-1	Top-5	Top-10	Top-1	Top-5	Top-10
NMRMind	9.28	15.13	17.69	1.90	4.10	5.40	1.77	2.88	2.88
NMRSolver	17.92	33.97	36.48	20.40	33.10	37.90	52.40	65.10	66.90
NMRAgent	**64.42**	**66.29**	**67.13**	**67.56**	**78.10**	**81.05**	**61.60**	**68.90**	**70.00**

References

[1] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: Large-scale self-supervised pretraining for molecular property prediction. arXiv preprint arXiv:2010.09885, 2020.

[2] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. IEEE Transactions on Big Data, 2025.

[3] Zheng Fang, Chen Yang, Hai-tao Yu, Haoming Luo, Haitao He, Jiaqing Xie, Zhuo Yang, and Jun Xia. Nmrgym: A comprehensive benchmark for nuclear magnetic resonance based molecular structure elucidation. arXiv preprint arXiv:2601.15763, 2026.

[4] Yongqi Jin, Jun-Jie Wang, Fanjie Xu, Xiaohong Ji, Zhifeng Gao, Linfeng Zhang, Guolin Ke, Rong Zhu, and Weinan E. Nmr-solver: automated structure elucidation via large-scale spectral matching and physics-guided fragment optimization. Nature Communications, 2026.

[5] Christoph Steinbeck, Stefan Krause, and Stefan Kuhn. Nmrshiftdb constructing a free chemical information system with open-source components. Journal of chemical information and computer sciences, 43(6):1733–1739, 2003.

[6] Ziyu Xiong, Yichi Zhang, Foyez Alauddin, Chu Xin Cheng, Joon Soo An, Mohammad R Seyedsayamdost, and Ellen D Zhong. Atomic diffusion models for small molecule structure elucidation from nmr spectra. arXiv preprint arXiv:2512.03127, 2025.

[7] Fanjie Xu, Wentao Guo, Feng Wang, Lin Yao, Hongshuai Wang, Fujie Tang, Zhifeng Gao, Linfeng Zhang, Weinan E, Zhong-Qun Tian, et al. Toward a unified benchmark and framework for deep learning-based prediction of nuclear magnetic resonance chemical shifts. Nature Computational Science, 5(4):292–300, 2025.

[8] Xi Xue, Hanyu Sun, Jingying Sun, Luc Patiny, Xiangying Liu, Kai Chen, Jingjie Yan, Liangning Li, Xue Liu, Shu Xu, et al. Nmrmind: A transformer-based model enabling the elucidation from multidimensional nmr to structures. Analytical Chemistry, 97(41):22603–22614, 2025.

[9] Chen Yang, Zheng Fang, Hanyu Sun, Fanjie Xu, Hongxin Xiang, Hanyu Gao, Xiangxiang Zeng, Yuqiang Li, Xiaojian Wang, and Jun Xia. A large-scale foundation model enables simulation-to-real adaptation for nuclear magnetic resonance-based molecular structure analysis, 2026. URL https://arxiv.org/abs/2606.20756.