NMR spectroscopy with language transformers

R. Andreev

MSc thesis
Computational Biology & Bioinformatics
D-BSSE, ETH Zurich

Supervised by E. Konukoglu (ETH Zurich)
Advised by N. Schmid (ZHAW)

2025-08-16 08:15

This MSc thesis is part of the Computational Biology and Bioinformatics (CBB) program at the Dept. of Biosystems Science and Engineering (D-BSSE), offered jointly by ETH Zurich, University of Basel, and University of Zurich.

This MSc thesis was hosted and supervised by E. Konukoglu at the Computational Vision and Learning group (D-ITET, ETHZ), advised by N. Schmid at the Institute of Applied Mathematics and Physics (Zurich University of Applied Sciences, ZHAW), in collaboration with M.-O. Ebert at the Laboratory of Organic Chemistry (D-CHAB, ETHZ).

Most of the compute was generously provided by Lambda GPU Cloud.

A big *thank you* to the people who have made this work viable.

NMR spectroscopy with language transformers

R. Andreev*

Abstract

NMR spectroscopy has extensive applications in (bio-)chemistry for characterizing molecular structures. We fine-tune the DistilGPT2 language transformer model to infer molecular structure from textual annotations of multimodal NMR spectra on the ~795k synthetic dataset by Alberts et al. [1]. The model is supervised on a multi-task objective that includes a vector embedding and functional group counts via auxiliary heads, as well as the next-token distribution w.r.t. *all* SMILES that serialize the target molecule. Inference with beam search shows ~80% top–10 accuracy on ~80% of the test set, which deteriorates with molecule size. To improve interpretability, we train data-informed confidence estimators with ~90% accuracy of whether a match is in the top–10. Selective heteromodal input fine-tuning suggests that the HSQC modality contributes more to the average accuracy than ^{13}C–NMR, despite being faster to acquire experimentally.

Abbreviations. **KL**: Kullback–Leibler (KL divergence is the mean extra surprise of a misspecified distribution); **NMR**: nuclear magnetic resonance (an analytical chemistry technique to probe molecules at atomic resolution); **HSQC**: heteronuclear single quantum coherence/correlation (a type of NMR sensing hydrogen–carbon bonds); **SMILES**: simplified molecular input line entry system (a non-unique atom-wise textual serialization of a molecule); **GRIMACE**: graph representation integrating multiple alternate chemical equivalents (a word-graph of SMILES)

1 Introduction

NMR spectroscopy is used in (bio-)chemistry for characterizing molecular structures from the magnetic properties of atomic nuclei such as ^{1}H (hydrogen) and ^{13}C (carbon-13), e.g., of synthetic starting materials, intermediates and products [2, §1.3]. This remains a laborious task for human experts, and it is therefore of substantial interest to automate it, especially in a self-driving lab. We address this problem using language transformers. More specifically, we fine-tune the autoregressive decoder-only language transformer DistilGPT2 to complete textual annotations of multimodal NMR spectra with the serialization of the molecule. Following the preliminaries of §1, we detail our approach in §2, present the results in §3, and conclude in §4. Most figures and tables are delegated to the backmatter.

*paper.mill.phd pm.me

2025-08-16

1.1 Nuclear magnetic resonance spectroscopy

NMR spectroscopy relies on the non-zero quantum spin of nuclei such as ^{1}H and ^{13}C. When placed in a strong magnetic field, their magnetic moment causes the nucleus to precess at the *Larmor* frequency proportional to the strength of the static magnetic field and the nucleus-specific magnetogyric ratio [2, §2.1]. Modern high-resolution NMR spectrometers operate with magnetic fields up to and sometimes above 23.5T, corresponding to proton resonant frequencies of 1000MHz [2, p. 13], but routinely at half of that.

From a range of NMR techniques [2, Table 1.2], we focus here on the basic 1-d techniques of ^{1}H–NMR and ^{13}C–NMR, as well as their 2-d correlation variant HSQC. In addition, we will assume that the chemical sum formula of the unknown molecule is known, for example, from a mass spectrometry experiment and other priors.

NMR occurs when a nucleus changes its spin state by absorbing supplied radiofrequency electromagnetic pulses. The perturbed spin system is allowed to relax towards the thermal equilibrium on the timescale of seconds, inducing a weak oscillating voltage in a receiver coil, known as *free induction decay* [2, §2.1–2.5]. This time-dependent signal is then Fourier–transformed into the frequency-domain spectrum, typically shown in parts per million (ppm) as deviation from a reference frequency (e.g., of tetramethylsilane) normalized by the same or by the operating frequency [2, §3.2.2, §3.3.2]. The spectral peaks provide key features:

- *chemical shift* – the position of a signal, which reflects the unique chemical environment of the nucleus;
- *scalar J-coupling* – fine splitting patterns of peaks, encoding connectivity between nearby nuclei, typically over two or three bonds for ^{1}H (but not for ^{13}C at low natural abundance);
- *integral* – proportional to the number of equivalent nuclei contributing to each peak, and lesser factors (applies mostly to ^{1}H–NMR).

This largely describes the 1-d NMR techniques. By contrast, HSQC correlates connected protons and carbons via one-bond scalar couplings that show up as crosspeaks in a 2-d chemical shift map [2, §7.3.1]. See §2.1 for further details.

Computing the spectrum from the molecule is termed the *forward problem*. Conversely, inferring the molecule from the free induction decay, the spectrum, or the annotations is the *inverse problem*, wherein each step could be considered an inverse problem in its own right.

Throughout, we assume a pure analyte in solvent, possibly with a reference compound.

1.2 Related machine-learning literature

We briefly review some of the machine learning literature pertinent to this work, with focus on exploiting alternate serializations of molecules and molecular structure inference with NMR spectroscopy. For an introduction to the SMILES serialization see §2.2. Recent reviews on inverse and forward NMR are: [3, 4, 5, 6, 7, 8, 9, 10, 11].

Vector representation. Almost any deep learning model operating on molecules could be repurposed for vector representation like smiles2vec [12] or mol2vec [13]. Ahmad et al. [14] pre-trained a RoBERTa [15] model from scratch, either with masked language modeling or multitask regression on 200 molecular descriptors on 77M unique canonical SMILES curated from PubChem [16]. They fine-tuned each on MoleculeNet [17], showing that scaling pre-training with multitask regression improves downstream performance. We use this model for molecular vector representation as one of our prediction targets because of its relatively large training set and availability on Hugging Face, see §2.3.

Alternate serializations. The fact that serialization of a molecule is non-unique has been widely exploited for dataset augmentation. Bjerrum [18] improved molecular property prediction by augmenting the training set ~130x using alternate SMILES. Bjerrum and Sattarov [19] trained LSTM–RNNs "heteroencoders" that translate one SMILES to another, observing enhanced chemical relevance of the latent representation in biological activity and physicochemical benchmarks but lower molecular fidelity than canonical SMILES autoencoders. Similarly, Winter et al. [20] proposed a model that maps a SMILES to the canonical one, and attached a head to the latent vector for simultaneous molecular property prediction. Zhang et al. [21] exploited large-scale availability of SMILES and 20x enumeration for self-supervised BERT [22] pretraining, then fine-tuned the outputs of task-specific padding tokens on classification and regression tasks (similarly to [14]). Skinnider [23] argued that generation and a posteriori filtering of invalid SMILES reduces structural bias and improves generalization, compared to constraining model output to valid SMILES only. Arús-Pous et al. [24] found that an alternate SMILES per molecule per epoch substantially enhances generalization and output diversity of RNNs trained to generate molecules, especially on small datasets. Alperstein et al. [25] built a fixed-length hierarchy of autoregressive Gaussian distributions as the molecular representation from $N = 5$ alternate SMILES that decodes via an LSTM "almost bijectively" to a different set of N SMILES, achieving state-of-the-art results in molecular property prediction and optimization.

NMR inverse problem. Huang et al. [26] trained a model on ~100k noised simulated spectra (#non-H ≤ 9) to predict 957 substructures from ^{1}H spectra, ^{13}C shifts and molecular formula, generating candidates bond-wise via beam search guided by substructure mismatch with ~90% top–10 accuracy on experimental spectra (#non-H ≤ 10). Hu et al. [27] trained an encoder–decoder transformer to assemble a SMILES from a fingerprint of the 957 substructures (top–15 accuracy of ~93%), and combined it into a model that reads ^{1}H and ^{13}C NMR spectra simulated with MestReNova (#non-H ≤ 19; top–15 accuracy of ~70%).

Jonas [28] trained a GNN [29] that takes an incomplete molecular structure with a known molecular formula, ^{13}C shifts/splits ("splitting reflects adjacent hydrogens" [28, §1.1]), and assigns probabilities to each new possible bond. Bonds are added until valences are met, choosing among candidates from multiple runs with the closest *predicted* spectrum. The network is trained to imitate an "oracle" that enumerates all bonds that can be added to a partial structure, which is similar to our next-token distribution supervision (§2.7). The training set of ~820k molecules was selected from PubChem [16] for max. 64 atoms, #C/O/N ≤ 32, rings only of 5–8 atoms, and no radicals, and the spectra are simulated with a GNN

forward model [30]. Top–5 inference accuracy of ~60% was reported [28, Table 1].

Yang et al. [31] combined a contrastive learning model aligning latents from a SMILES encoder and a ^{13}C–NMR encoder with candidate SMILES generation and library amplification, so that a new spectrum is matched to a candidate from the library.

Sridharan et al. [32] employed policy-guided Monte Carlo tree search [33] on molecular structures that adds bonds (subject to valency/ring constraints), given the molecular formula and ^{13}C–NMR spectrum. Trained on 2134 molecules (#C/O/N/F $\leq$ 9; no radicals/charge) from nmrshiftdb2 [34], and employing a pretrained forward model [30] and substructure match for reinforcement learning, the model achieves top–5 accuracy of ~86% [32, Fig. 6a].

Yao et al. [35] trained a transformer through a curriculum of tasks, including forward and inverse ^{13}C–NMR to SMILES prediction, progressing from 360M molecules from ZINC [36], to 45M from PubChem [16] simulated with "Gaussian", to experimental spectra "consistent with [31]" (appears to include nmrshiftdb2 [34], NAPROC-13 [37], and 370k spectra collected from literature). Top–10 accuracy of ~72% was reported [35, Table 1] on inference from ^{13}C–NMR and molecular formula.

Alberts et al. [38] trained an encoder–decoder transformer to translate ^{1}H- and ^{13}C-NMR spectra (H/F-decoupled) with molecular formula to SMILES. Molecules were sampled from precursors and products of ~1M reactions from "Pistachio" [39] ($6 \leq$ #non-H ≤ 35; only H/C/N/O/S/P/halogens; no charge/isotopes), and spectra were simulated and annotated with MestReNova (cf. Table 1). Presumably, ~1M molecules had both types of spectra [38, §4.1]. The model achieved a top–10 accuracy of ~85% [38, Table 1], comparable to our "F/Q/C/H $\rightarrow$ SMILES" model (see §3.1). Alberts et al. [1] mined USPTO reactions from Lowe [40] and simulated the spectra for ~795k molecules, see §2.1. Their "vanilla encoder–decoder transformer" achieved top–10 accuracy of ~90% on SMILES recovery from ^{1}H- and ^{13}C-NMR spectra with molecular formula [1, §4.1]. Alberts et al. [41] refined that approach by training on selectively heteromodal data (i.e., unpaired or partially missing spectral modalities), fine-tuning on infrared spectra from the NIST EPA gas-phase library [42], ^{13}C spectra from nmrshiftdb2 [34] and the small educational dataset by Van Bramer and Bastin [43], with a top–5 accuracy of ~98% on hold-out subsets of this educational dataset.

Priessner et al. [44] trained a transformer to read various spectra (^{1}H, ^{13}C, HSQC, infrared, etc.) and generate SMILES, with an additional adaptive dataset expansion strategy (improvement cycle), wherein molecules near a given novel set are sampled with a mol2mol model (cf. [45]), their spectra are simulated, and the transformer is fine-tuned on these new samples in order to bridge the distribution shift.

Yang et al. [46] pre-trained a diffusion molecular autoencoder and aligned the latents with those of an NMR spectrum encoder via contrastive learning. On molecules from the dataset by Alberts et al. [1] with #non-H ≤ 25, they reported top–10 inference accuracy of ~75% from ^{1}H- and ^{13}C-NMR spectra with molecular formula (see [46, Table 1] for qualifiers).

2 Design principles

In this section we describe how spectra and molecules are encoded, and the training setup. The multimodal NMR data are drawn from a synthetic dataset and compacted into tokenized text (§2.1). Molecules are represented as vector embeddings, as functional group counts, and as a graph over alternate SMILES serializations (§2.2, §2.3, §2.7). The dataset is split by clustering molecular embeddings to reduce leakage (§2.4). These targets are learned jointly by a transformer derived from pre-trained DistilGPT2 (§2.5). Tokenization builds on GPT-2 and includes task-specific trigger tokens (§2.6). Losses include adaptively whitened log-likelihoods (for embeddings and group counts), and KL divergence for next-token distributions over alternate SMILES; these are combined into a single multi-task objective (§2.8).

2.1 NMR data generation and representation

High-quality digital experimental NMR spectroscopy data, especially multimodal, are not currently available at scale. We use the synthetic dataset published by Alberts et al. [1] which contains spectra for ~795k molecules sourced from the "Chemical reactions from US patents" dataset by Lowe [40] and simulated in MestReNova. Of the ~1.6M unique molecules in [40], the dataset [1] includes only molecules with 5–35 non-hydrogens and common organic elements (H/C/N/O/S/P/B/Si/halogens), and for which all spectroscopic simulations (IR, NMR, MS) succeeded [1, §3]. The spectra have a resolution represented as vectors of size 10000 for both ^{1}H–NMR and ^{13}C–NMR (^{1}H-decoupled), and a 512^2 matrix for HSQC. In addition, the dataset includes annotations by MestReNova in a JSON format with peak range, peak type, centroid ppm and integrations (for ^{1}H–NMR), centroid ppm and intensity (for ^{13}C–NMR), ppm coordinates and intensities (for HSQC); see Table 1 (p. 7). Mestrelab does not publish details on the NMR simulation methodology [47] but it is a mixture of algorithms, including machine learning [7, §3]. Thus, the peak annotation is itself approximate, as it does not tap into the details of the simulation, but the same is true for *experimental* data.

Throughout, we use this peak annotation as the de facto NMR data, which we present to the language model as plain text in a compact format. An example is shown in Table 1. This representation appears somewhat cryptic because we need to balance readability vs fitting the samples into context length of 1024 tokens (cf. §2.5). While Alberts et al. [38, 41] split the text using a regex into short blocks such as `J`, `2H`, `dt`, `2.19`, etc., that become the vocabulary/tokens for the model, we use the tokenizer on the raw text (see §2.6). This has the advantage of high flexibility, for example, the language model can be fine-tuned on novel input, but the tokenization is far from parsimonious for the task, given that most tokens never appear in the NMR data.

We discuss the dataset in terms of functional groups in Fig. 5 (p. 23) and Fig. 6 (p. 24).

2.2 Molecular structure representation

Since our goal is to infer the identity of a molecule, it is natural to describe it as a graph of atoms/nodes and bonds/edges with attributes (rather than as a 3d cloud, say). This abstraction still needs to be instantiated, however. Here we can distinguish between representations that are in a sense permutation-invariant and those that necessitate a serialization order. In the first category we have the representation as a set of nodes and the adjacency matrix, but since the connectivity is not known in advance, many models fall back to generating or refining parts of the graph iteratively, see [48] for a review and [25, §4.7.2] for a related discussion. In the second category, we have, most notably, SMILES [49, 50] and SELFIES [51].

The Simplified Molecular Input Line Entry System (SMILES) is a textual molecular representation using atomic symbols, connectivity rules, and stereochemical descriptors, for example *ethanol* as `CCO` and *aspirin* as `CC(=O)Oc1ccccc1C(=O)O`, wherein

- atoms are denoted by chemical symbols, with hydrogen inferred by valency;
- double `C=C` and triple `C#C` bonds are explicit, while single bonds are normally implicit, and aromatic bonds are represented by lowercase atomic symbols ("unkekulized");
- parentheses define side chains;
- numbering can be used for ring closures (e.g., *benzene*, `c1ccccc1`);
- `@` and `@@` denote chiral centers, and slashes `\/` cis-trans isomerism;
- formal charges are indicated with square brackets, e.g., `[NH4+]` or `[O-]`.

Importantly, the SMILES representation of any molecule is in general not unique, as it can be serialized starting from any atom, with branches traversed in any order. Thus, all SMILES strings form equivalence classes, with each class corresponding to a unique molecule. We explore and exploit this fact in §2.7. The *canonical* SMILES is the unique representative of the given molecule that is reproducible within a given toolkit. We use RDKit's unkekulized canonicalization without explicit hydrogens, where needed (cf. [52]).

Self-referencing Embedded Strings (SELFIES) uses bracketed tokens whose grammar enforces valence and bonding rules. Unlike SMILES, any sequence of grammatically valid SELFIES tokens decodes to a molecule. We chose not to use SELFIES because invalid SMILES provide a measure of prediction uncertainty. A similar case for SMILES is made by Skinnider [23].

2.3 Molecular vector representation – "chembedding"

Ahmad et al. [14] trained a RoBERTa (cf. [15]) on 77M canonicalized PubChem SMILES to regress 200 molecular descriptors. We attached average–pooling of the last hidden states to define the "chembedding" as a 384-d vector representation of the molecule. Of the several similar models available on HuggingFace, we found ChemBERTa-77M-MTR had the best separation between embeddings of the non-canonical SMILES of the same molecule versus embeddings of SMILES of different molecules, see Fig. 1. This is the model we refer to throughout as the "chembedding".

Example molecule SMILES:

```
CCOC(=O)C(Cc1ccc(OC)c(CBr)c1)OC(C)C
```
[🖉] (1)

Annotated NMR data (excerpt, rounded):

```
...
{
  "category": "ddt",
  "centroid": 6.992,
  "delta": 6.993,
  "j_values": "0.88_2.02_8.79_",
  "nH": 1,
  "rangeMax": 7.021,
  "rangeMin": 6.965
},
...
```

```
...
{
  "delta (ppm)": 130.27
      ,
  "integral": 0.00096,
  "intensity": 0.0512,
  "width (ppm)": 0.0119
},
...
```

```
...
{
  "13C_centroid": 56.18,
  "13C_max": 56.91,
  "13C_min": 55.45,
  "1H_centroid": 3.813,
  "1H_max": 3.866,
  "1H_min": 3.759,
  "nH": 3.0
},
...
```

Compacted NMR text data (excerpt):

```
F|C16H23BrO4
Q|6.0:1.11-1.25@23|3.0:3.76-3.87@56.2|3.0:1.01-1.12@14.2|···|1.0:3.03-3.14@37.6
C|3.1@172|3.1@157|5.4@130|3.2@130|3.1@129|5.3@128|5.3@112|···|9.7@14.2
H|1:7.09-7.13:dp:0.86,1.8|1:6.96-7.02:ddt:0.88,2,8.8|···|3:1.04-1.09:t:6.4
```

Table 1: Example of annotated ^{1}H–NMR, ^{13}C–NMR and HSQC peaks for the molecule (1) in the multimodal NMR dataset by Alberts et al. [1], and the compacted text representation as presented to the language model, containing information such as intensity/integration, location or range in ppm units, multiplet type, and J-coupling values.

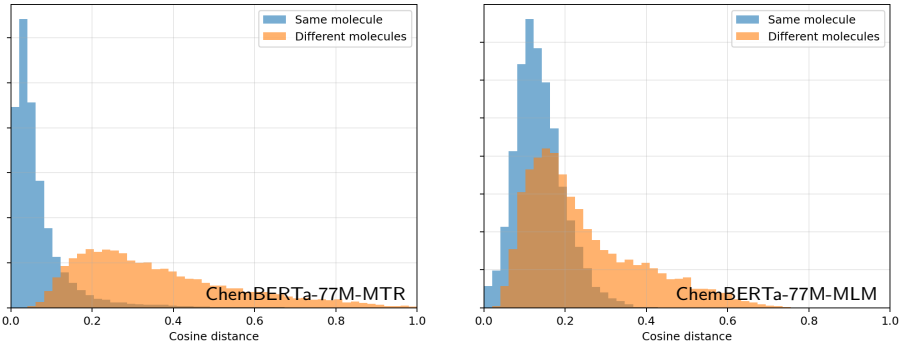


Figure 1: The cosine distance $(1 - \cos\angle)$ between the "chembeddings" of two random same-molecule SMILES or the SMILES of two different molecules. The "chembedding" is the average of the last hidden states of a ChemBERTa-2 model [14] from HuggingFace, cf. §2.3, specifically, (←) DeepChem/ChemBERTa-77M-**MTR** and (→) ...-**MLM**. Computed from 10000 pairs selected at random from the top 1M PubChem molecules.

2.4 PubChem and dataset “cheese” split

Recall that the multimodal NMR dataset by Alberts et al. [1] is based on the “Chemical reactions from US patents” by Lowe [40]. We found that the bipartite patent–molecule graph is both highly connected and imbalanced, so that a random training/validation/test split or a patent–aware split would risk information leakage. To mitigate this, we partition the chemical space into clusters, reserving the outer shell of each cluster for validation and test.

To define this “chemical space”, we obtained ~120M molecules with names and synonyms from PubChem [16]. We sorted by the number of synonyms and took the “top–1M” subset. We applied the “chembedding” to this subset and clustered the vectors into 1024 clusters with k–means by cosine–distance. We then sorted the NMR dataset into these clusters. The outer 10% of each cluster (“halo”), as well as the same number of molecules from the inner 90% of each cluster (“endo”), are randomly assigned to *validation* or *test* set. The remaining 80% of the molecules are used for *training*. We call this methodology the 80/10/10 “cheese” split (clustered halo-/endo- ensemble of sample embeddings).

2.5 Main model architecture

The GPT-2 family of decoder-only transformer models [53] was trained on 40GB of Reddit-curated WebText in an autoregressive, self-supervised manner. At the time, it achieved competitive zero-shot performance (i.e., without task-specific fine-tuning) in standard benchmarks, as well as on translation, summarization, question answering, and commonsense reasoning.

DistilGPT2 is a 6-layer, 82M° knowledge-distilled student (cf. [54]) of the smallest 12-layer, 124M° GPT-2 model with the same 768-d hidden size, 12 attention heads, embedding–output weight tying [55], and 1024 tokens context length.

Our main model is DistilGPT2 with auxiliary heads attached to the last hidden state (like the head for token logits):

- a $768 \to 384$ affine map for the “chembedding”, and (2.1)
- a $768 \to 289$ affine map followed by a softplus for the functional-group counts. (2.2)

See Table 5 (p. 22) for further specifics. The auxiliary heads are supervised only after the task-specific “trigger token”, as described below. The output of (2.2) is interpreted as log1p counts, as discussed in §2.8.

2.6 Tokenizer

We use the tokenizer that the pre-trained DistilGPT2 model was trained with and therefore depends on. The tokenizer was originally constructed for the GPT-2 series [53] by splitting the training corpus with a fixed regex, converting each chunk to UTF-8 bytes, and then greedily applying a byte-level BPE merge [56, 57], by fusing the most-frequent adjacent pairs 50000 times: starting from the 256 base bytes, plus the “end-of-text” token, this results in the deterministic vocabulary of 50257 tokens. For further nuances see Karpathy [58].

We resize the tokenizer and model token embedding to include new special tokens:

⋮ = trigger token for the "chembedding" estimation task, (3.1)
⬡ = trigger token for the functional group enumeration task, (3.2)
▒ = placeholder where an auxiliary head is supervised, (3.3)
⇄ = trigger token for the SMILES generation task, (3.4)
▮ = supervised end-of-sequence token, (3.5)
■ = non-supervised padding tokens in a batch. (3.6)

One might object that this tokenizer is not chemically informed, e.g., it does not distinguish between "`Cn` representing carbon bound to aromatic nitrogen and `Cn` for Copernicium" [59], and the majority of tokens have no chemical meaning and could be retrained to code for functional groups or numbers instead (cf. [60]). What is more, both token sequences `[46,4093]` and `[4503,34]` decode to the SMILES string `OCC`, but only the first one would be produced by the tokenizer and appear in the training data. Occasionally, the model learns SMILES continuations that are technically correct but that would be labelled/supervised as incorrect in training and token-based evaluation (such as the KL loss). We believe language models should have no trouble learning to disambiguate these cases and that fine-tuning *large* language models with their native tokenizer on SMILES data is worth exploring. Meanwhile, designing a "chemical" tokenizer is an active area of research [61, 62, 63, 64, 65, 66].

2.7 Next-token supervision with GRIMACE

Consider *aspirin* with the SMILES

`CC(=O)Oc1ccccc1C(=O)O` 🔗. (4)

In total, it has 304 distinct unkekulized SMILES variants, which can be collated in a directed acyclic word graph, where prefixes and suffixes are shared, i.e., a graph representation integrating multiple alternate chemical equivalents (GRIMACE), see Fig. 2. Note carefully that the SMILES strings are tokenized first. For example, there are two variants

`c1(ccccc1OC(=O)C)C(O)=O` and `c1(ccccc1OC(C)=O)C(O)=O` (5)

that differ the serialization of the acetyl group "–$COCH_3$" but share prefix and suffix. For a given molecule, the compact graph representation as in Fig. 2 (p. 11) is precisely the minimization of the deterministic finite automaton whose state is any prefix and transitions are the tokens on the way to a correct serialization. Algorithms for minimizing finite state automata are reviewed by Watson [67], and *online* algorithms are available that modify the graph as new words are added [68]. We construct the GRIMACE in two steps in linear time. *First*, each observed tokenized variant is streamed into a growing prefix tree, re-using existing edges or creating new ones while accumulating per-edge visit counts. *Then*, a bottom-up "Merkle" pass assigns each node a 64-bit digest $h(v) := \text{hash}(\text{tok}(v), \{h(c) : c \text{ is child of } v\})$, such that isomorphic suffixes share the same digest, and nodes with the same digest are merged into a single GRIMACE node with sum-aggregated edge counts.

Since we avoid any pre-processing such as regex-based splitting of SMILES character strings, the set of tokenized SMILES does not contain rare variants that would also decode to the same molecule, cf. the remark on `OCC` in §2.6.

To construct the GRIMACE for a given molecule in practice, we sample random SMILES variants using RDKit with the stopping criteria:

- at least 16 samples,
- at most 10000 samples,
- at most 200 consecutive iterations without new variants,
- at most 50% of the estimated total N of distinct variants.

Let U_n count distinct variants observed after n draws. We estimate the total N from

$$\mathbb{E}[U_n] = \sum_i \mathbb{P}[\text{variant } i \text{ observed within } n \text{ draws}] = N\left(1 - (1 - \tfrac{1}{N})^n\right), \tag{6}$$

assuming that all variants are equally likely, plugging in the observed number for the expectation, and solving for N iteratively using Newton–Raphson. At 50% of the estimated total, progress begins to slow down significantly (and we have $n \approx N$).

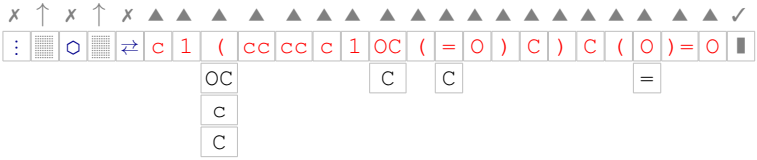
The GRIMACE is constructed from all the observed variants. The edges are first labeled with the number of observed variants that pass through them, later normalized to percentages for each source node (or prefix). For example, after the common prefix in (5), the GRIMACE in Fig. 2 records 107 and 113 outgoing samples. Thus, supervising the model on one of the variants from (5), the next-token distribution after the common prefix would be

$$\mathbb{P}(\texttt{=} \mid \text{prefix}) := \tfrac{107}{220} \approx 48.6\% \qquad \text{and} \qquad \mathbb{P}(\texttt{C} \mid \text{prefix}) := \tfrac{113}{220} \approx 51.4\%, \tag{7}$$

with all other tokens considered incorrect or *off-support* continuations. Our model will assign some probability to these off-support tokens, which we call the *off-support error*.

For practical reasons, for each molecule in our datasets, we sample *five* trajectories through its GRIMACE, recording the next-token distributions after each prefix, save them to a file and discard the rest of the tree for the sake of memory efficiency. At training time, one of those saved trajectories is selected at random at each training epoch.

In summary, we supervise the multi-task model using teacher forcing on targets (with task-specific trigger tokens from (3)) like this,



(8)

where "✗" indicates no supervision, "↑" supervision of the respective auxiliary head, "▲" KL supervision on the possible alternative tokens (cf. highlight in Fig. 2), "✓" the usual cross-entropy loss, and the losses are computed and combined as described in §2.8.

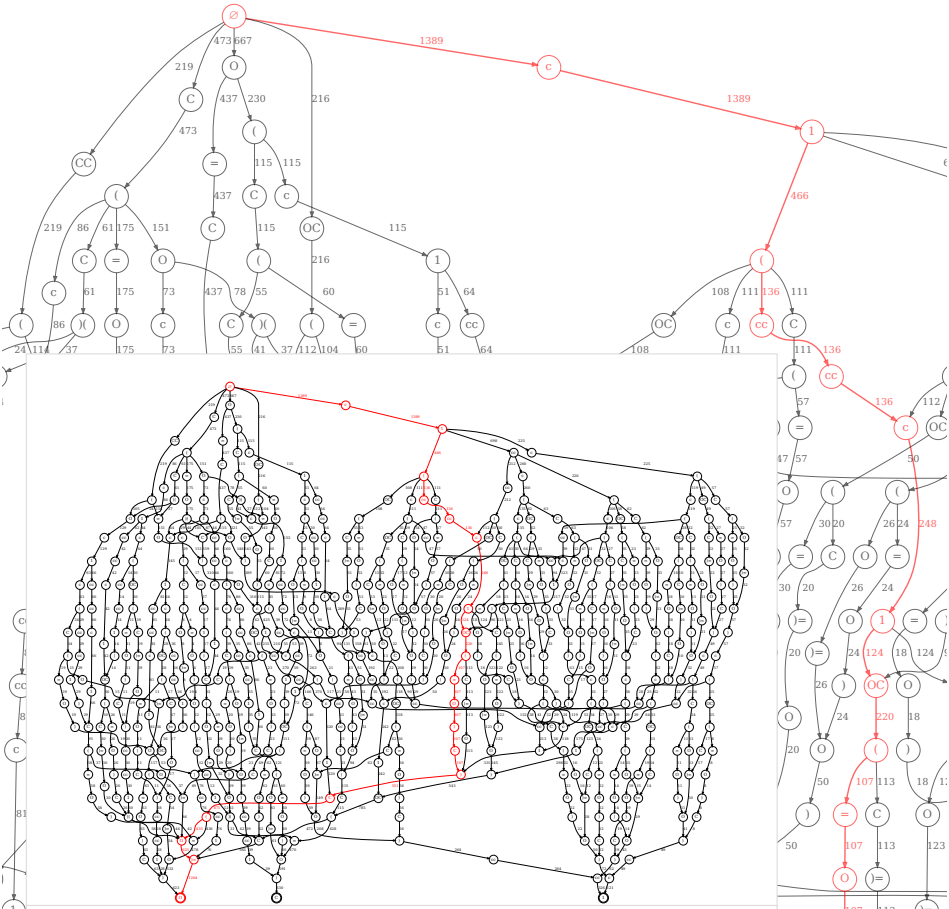


Figure 2: The "graph representation integrating multiple alternate chemical equivalents" (GRIMACE) consisting of all 304 tokenized serializations of *aspirin* (4) with one highlighted path for reference (cf. (8)). The numbers on the edges count the number of observed variants of the sampled total that pass through them (oversampled to make sure all variants are captured).

2.8 Adaptive whitening, losses, and the multi-task objective

The metrics on the auxiliary head outputs (2.1)/(2.2) require consideration. The structure of the learned "chembedding" space is unclear, so that a Mahalanobis metric seems appropriate. For the functional group counts, there are some correlations and very different prevalences. We propose to address both using *adaptive whitening* with trainable metrics (inspired by Cipolla et al. [69]), as follows. Let T be a real $n \times n$ matrix with $\det T > 0$. Then

$$p_2(\mathbf{e} \mid T) := (2\pi)^{-n/2} (\det T) \exp(-\tfrac{1}{2}\|T\mathbf{e}\|_2^2), \quad \mathbf{e} \in \mathbb{R}^n, \tag{9}$$

defines a multivariate Gaussian distribution on $\mathbb{R}^n$ with the negative log-likelihood

$$-\log p_2(\mathbf{e} \mid T) = \tfrac{1}{2}\|T\mathbf{e}\|_2^2 - \log\det T + \tfrac{n}{2}\log 2\pi. \tag{10}$$

Similarly, let S be a real $m \times m$ matrix with $\det S > 0$. Then

$$p_1(\mathbf{f} \mid S) := 2^{-m} (\det S) \exp(-\|S\mathbf{f}\|_1), \quad \mathbf{f} \in \mathbb{R}^m, \tag{11}$$

defines a Laplace-like distribution on $\mathbb{R}^m$, motivated by its negative log-likelihood

$$-\log p_1(\mathbf{f} \mid S) = \|S\mathbf{f}\|_1 - \log\det S + m\log 2. \tag{12}$$

We use the dimension–normalized negative log-likelihood as the loss:

- $\frac{1}{n}$(10) with $n = 384$ for the "chembedding" head, and (13.1)
- $\frac{1}{m}$(12) with $m = 289$ for the "log(1 + functional group count)" head. (13.2)

The transformations T and S are both *trainable*, and the $-\log\det$ term prevents collapse. T defines a Mahalanobis-style metric that attempts to minimize the variance of the error. S defines a metric less sensitive to outliers and promotes sparsity of the error, so we expect the model to capture a) certain functional groups over others, b) correlations, and c) relative abundances, some of which we confirmed in small-scale synthetic experiments. Since training runs in reduced precision, we parametrize T and S as a matrix-exponential (for $\det > 0$) of a component-wise tanh (for boundedness) of an unconstrained trainable square matrix.

Next-token distribution supervision on the GRIMACE (see §2.7) is achieved using the KL loss at each supervised position, averaged over the positions. In some cases, we multiply the position loss by a weight that increases geometrically with the position index, starting from the first one, in order to improve the prediction accuracy on longer SMILES.

The end-of-sequence token marks the end of the SMILES and is supervised with the usual cross-entropy loss (which equals the KL loss with a one-token reference distribution).

Unfortunately, both expressions (10) and (12) can become negative, which makes combining multiple losses nontrivial and reduces interpretability. We chose to combine the tasks/losses in the spirit of Cipolla et al. [69] into the main multi-task training objective

$$\text{minimize} \quad \textstyle\sum_{\text{task}} \left(\sigma_{\text{task}}^{-2}\, \text{softplus}(\text{loss}_{\text{task}}) + \log \sigma_{\text{task}}^2\right), \tag{14}$$

with a trainable σ_{task} for each of the four tasks, wherein "softplus" ensures non-negativity.

3 Training and results

In this section we discuss briefly the most pertinent baseline models and lead-up results of this work (§3.1). We proceed to define the training curriculum for our main multi-task transformer model, and evaluate its performance on the "chembedding" and functional group enumeration tasks, as well as on top–n structure beam-search inference accuracy, stratified by SMILES length and stereo-awareness (§3.2). We then introduce calibrated meta-classifiers for top–10 confidence estimation (§3.3), and quantify the relative informativeness of NMR modalities via input ablation and fine-tuning (§3.4).

To specify the active multi-modal combination we use here the abbreviations **F** = chemical sum formula, **Q** = HSQC, **C** = ^{13}C–NMR, and **H** = ^{1}H–NMR.

3.1 Baseline models and lead-up results

Besides introducing the multimodal spectroscopic dataset we use in this work, Alberts et al. [1] reported, among others, top–10 accuracy of ~90% on F/C/H → SMILES prediction (surpassing [38]), by means of an encoder–decoder transformer model [1, §A.6.3]. We trained the same model using their protocol, obtaining, for example, top–10 accuracy of ~81% on F/H → SMILES, whereas they reported ~84% in [1, Table 3]. We obtained similar accuracy by training a) on SMILES with explicit hydrogens, and b) on SMILES with both explicit hydrogens and stripped stereochemistry. For code, results and model checkpoints see ref. [70].

Next, we trained the DistilGPT2 decoder-only language model from its pre-trained state on F/Q/C/H → SMILES with the usual CausalLM cross-entropy token supervision loss, obtaining top–10 accuracy of ~86% on the multimodal spectroscopic dataset with a random 80/10/10 split after pre-filtering by correct sum formula and ignoring stereochemistry.

For a proof-of-concept of our main model, we then trained DistilGPT2 from its pre-trained state on

$$\text{SMILES} \longrightarrow \boxed{\vdots}\boxed{}\boxed{\bigcirc}\boxed{}\boxed{\rightleftarrows}\ \text{GRIMACE}\ \boxed{} \tag{15}$$

with the multi-task objective (14) on the top-1M PubChem SMILES of character length at most 70 with the 80/10/10 "cheese" split (cf. §2.4) for training and evaluation. Evaluating the model in 20-beam search mode, we obtained top–10 accuracy of ~93% on the full test set (where ~2% of SMILES are over 100 characters long) and top–10 accuracy of ~97% on SMILES up to 70 characters long. While this does not prove that the model is able to internalize the molecular structure (because it can "cheat" by looking at the given input SMILES), we observe convergence on all four tasks, in particular, the KL loss dips below 10% within ~30 training epochs and continues to decrease slowly after that.

In order to better understand the multimodal spectroscopy data we trained a GNN → C/H model that passes a molecule through a GNN and returns an estimate of ^{13}C–NMR and ^{1}H–NMR spectra with an important simplification: each C/H atom is represented by exactly one peak in its modality (but identical peaks are integrated). This is not the case for annotated spectra, where one might misinterpret noise as peaks or miss smaller/overlapping/broad

peaks, and effects like conformer exchange and quadrupolar coupling complicate peak readout [71, §8.5.1ff.], [2, §2.5f.]. Notably, we saw large ppm shift mismatches for hydrogens attached to oxygens, possibly due to "labile" proton exchange or hydrogen bonding within the molecule or with the simulated solvent (deuterated chloroform [1, §3.1]), cf. also [71, §6.4.3] and [2, §2.6.1.5]. In addition, our GNN looks *four* neighbors deep, capturing only local effects. Thus, this "quasi-NMR" model is inaccurate – but consistently so. We synthesized a quasi-NMR dataset from the top–1M PubChem SMILES with the intention to pre-train the main F/Q/C/H → GRIMACE model on it. This inference model learned faster on this quasi-NMR data than on the main NMR dataset, indicating that additional and more realistic NMR effects are rather confusing than helpful in structure inference. However, the resulting pre-trained model performed poorly on the main NMR dataset and additional training progressed slowly in the KL loss, so we leave this line of research to future work.

3.2 NMR annotation to multitask estimation and learning equivalent SMILES sequences

We trained our main multi-modal multi-task molecular structure inference model on the synthetic multimodal spectroscopic dataset "cheese" split (cf. §2.4) with the objective (14),

$$\text{Complete F/Q/C/H NMR annotation} \rightarrow \boxed{\vdots}\boxed{\blacksquare}\boxed{\hexagon}\boxed{\blacksquare}\boxed{\rightleftarrows}\ \text{GRIMACE}\ \boxed{\blacksquare} \tag{16}$$

through a curriculum of increasing character length of the canonical reference SMILES:

$$\#1\text{: 30 epochs on } |\text{SMILES}| \leq 30 \text{ from the pre-trained DistilGPT2 state,} \tag{17.1}$$
$$\#2\text{: 35 epochs on } |\text{SMILES}| \leq 35, \tag{17.2}$$
$$\#3\text{: 40 epochs on } |\text{SMILES}| \leq 40, \tag{17.3}$$
$$\#4\text{: 45 epochs on } |\text{SMILES}| \leq 45, \tag{17.4}$$
$$\#5\text{: 50 epochs on } |\text{SMILES}| \leq 50, \tag{17.5}$$
$$\#6\text{: 55 epochs on } |\text{SMILES}| \leq 55, \tag{17.6}$$
$$\#7\text{: 55 epochs on } |\text{SMILES}| \leq 55 \text{ with KL geometric weight } q_{\text{KL}} = 5\%. \tag{17.7}$$

We used the AdamW optimizer (fused [72]) with a learning rate $\text{lr} = 3 \times 10^{-4}$ or $\text{lr} = 5 \times 10^{-4}$, $(\beta_1, \beta_2) = (0.9, 0.95)$, $\varepsilon = 10^{-8}$, and weigth decay of 10^{-2} [73, 74].

Training and validation losses of all runs are documented in Fig. 7 (p. 25). Since the losses based on the log-likelihoods (10)/(12) become negative, we offset them by the 0.001 quantile in the plots for visibility. During run (17.6), the KL loss plateaus at ~14%. In run (17.7), we measure and show the geometrically weighted KL loss of ~30% but the off-support error on the training set drops about $8\% \rightarrow 6\%$ thanks to the geometric weight (not shown).

In Fig. 8 (p. 27) we compare the "chembedding" predicted by the model after the corresponding trigger token to that of the canonical SMILES of the target molecule in various norms. As expected, they tend to be closer when there is a top–10 match. In most cases, the predicted "chembedding" is still closer to the target than to most randomly selected molecules.

We observe analogous results for the vector of functional group counts, as shown in Fig. 9.

We analyze the functional group contributions to the loss and the adaptive metric S from (12) in Fig. 10 (p. 29). Some of the difficult-to-predict functional groups are: chiral center, trans double bond, halogen acetal-like, secondary/tertiary alcohol, secondary/tertiary amide, and bridged rings. Of note, we have two equivalent descriptions for a chiral center, and the application of the trained S almost eliminates one of them (i.e., sparsifies the error).

We ran predictions in 20-beam search mode on ~10k random test samples. We evaluated the top–10 on increasing subsets defined by the maximum reference SMILES character length, which is an important covariate of accuracy. We measured the top–n accuracy either as an exact structure match (Fig. 3a) or as a stereo-isomer match, i.e., stripping stereochemistry annotation (Fig. 3b). Selected datapoints are collected in Table 3, where we show the significant improvement during the last training run with the KL geometric weight.

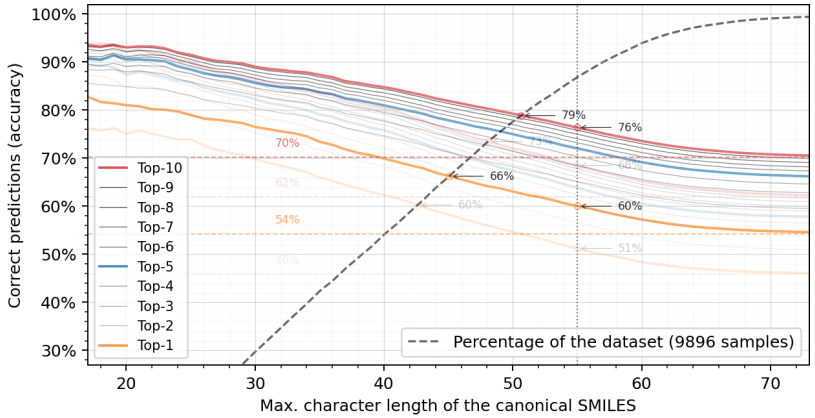
3.3 Data-informed confidence estimators

Initially, we only have the top–n accuracy as a measure of confidence about the set of candidates by the inference model from §3.2 for a given sample. To refine this confidence, we trained two separate binary meta-classifiers as data-informed confidence estimators on the top–10 beam search candidates, one to predict whether there is an exact match, and one for a stereo-isomeric match. As the input features we took (cf. Fig. 11b, p. 30)

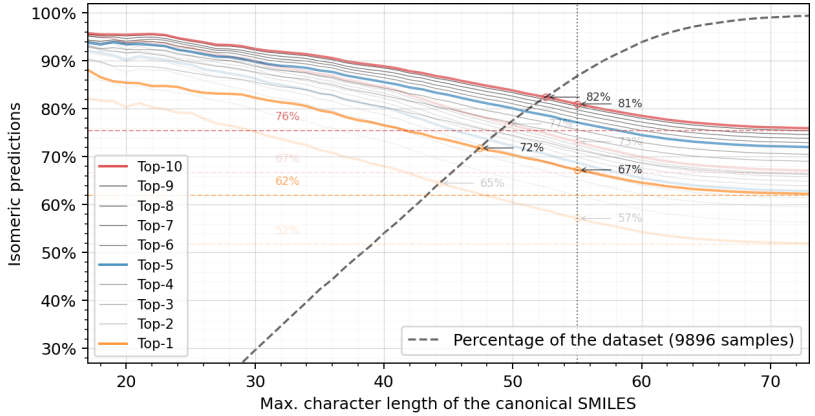
- `score`: the log-probability of the candidate according to the model,
- `is_valid`: whether the candidate is a valid SMILES,
- `freq`: the absolute frequency of the candidate molecule among the top–10,
- `is_majority`: whether the candidate molecule is the majority in the top–10,
- `is_sum_ok`: whether the sum formula of the candidate is correct,
- `char_len`: the character length of the candidate SMILES,
- `nmr_data_char_len`: the character length of the NMR input data.

Each of the ten candidates from beam search contributes one feature vector, with missing values imputed as zero. We apply a sample-aware 80/20 train/test split. Within the training set we precompute a 5-fold grouped cross-validation split for calibration purposes. The base meta-classifier is a `RandomForestClassifier` from scikit-learn [75] with 300 estimators, maximum depth 12, minimum 5 samples per leaf, balanced class weights, full parallelism, and a fixed seed. We wrap this in a `CalibratedClassifierCV` with isotonic regression and fit using the precomputed group folds.

We max-pool the meta-classifier's predictions from candidate to sample level, thus defining the confidence score of a top–10 match. We set $\hat{y} = 1$ if *score* $\geq$ *threshold*, and $\hat{y} = 0$ else, sweep the threshold, and compute precision, recall, and negative predictive value. These are shown in Fig. 4a for the exact match and in Fig. 4b for the stereo-isomer match. Selected datapoints are collected in Table 4 for the checkpoints (17.6) $\rightarrow$ (17.7). Of particular interest is the intersection of *precision* and *negative predictive value*, where we can define the accuracy of the meta-classifier as 85% for the exact match and as 90% for the stereo-isomer match, i.e., the meta-classifier is correct with this probability.



(a) Exact structure match.

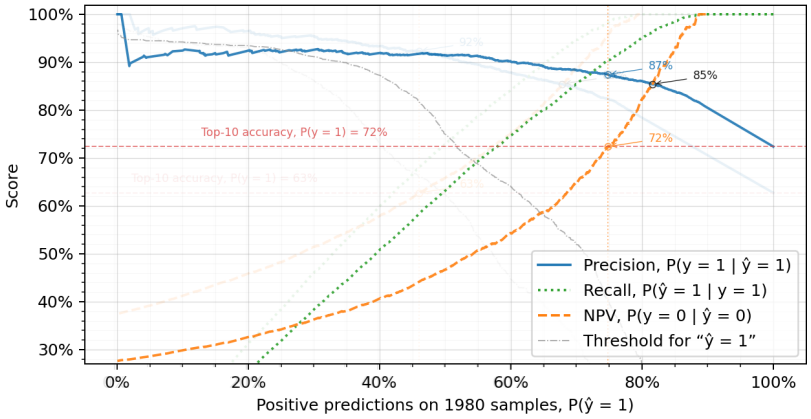


(b) Stereo-isomer match.

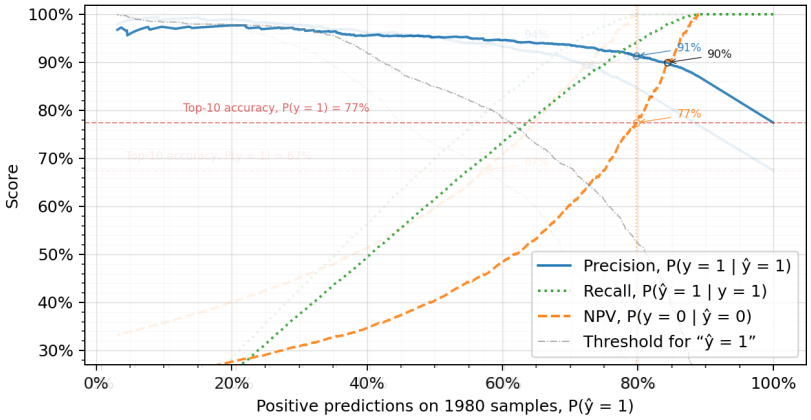
Figure 3: Top–n inference accuracy at checkpoint (17.7). Evaluated on increasing subsets of the test set parameterized by the max. character length of the canonical SMILES, see §3.2.

SMILES match	Subset	Top–1	Top–10
Exact Fig. 3a	$x\,\%$ of data	$60\% \to 66\%$	$73\% \to 79\%$
	$\lvert\text{SMILES}\rvert \leq 55$	$51\% \to 60\%$	$68\% \to 76\%$
	All	$46\% \to 54\%$	$62\% \to 70\%$
Stereo-isomer Fig. 3b	$x\,\%$ of data	$65\% \to 72\%$	$77\% \to 82\%$
	$\lvert\text{SMILES}\rvert \leq 55$	$57\% \to 67\%$	$73\% \to 81\%$
	All	$52\% \to 62\%$	$67\% \to 76\%$

Table 3: Selected datapoints from Fig. 3, showing the improvement with the KL geometric weight between checkpoints (17.6) $\to$ (17.7). The "$x\,\%$ of data" is the subset of shorter SMILES where we have $x\,\%$ accuracy on $x\,\%$ of data.



(a) Exact structure match.



(b) Stereo-isomer match.

Figure 4: Precision, recall and negative predictive value parameterized by threshold, but shown as functions of the number of positive predictions for the meta-classifier from §3.3.

SMILES match	Threshold	NPV	Precision
Exact Fig. 4a	NPV = accuracy	$63\% \to 72\%$	$92\% \to 87\%$
	NPV = precision	$85\% \to 85\%$	
Stereo-isomer Fig. 4b	NPV = accuracy	$67\% \to 77\%$	$94\% \to 91\%$
	NPV = precision	$89\% \to 90\%$	

Table 4: Selected datapoints from Fig. 4, showing the change with the KL geometric weight between checkpoints (17.6) → (17.7).

3.4 Selective heteromodal input fine-tuning

It is of interest to estimate the relative contributions of the NMR modalities to the overall inference accuracy because HSQC is much quicker to obtain than ^{13}C–NMR [2, §1.3]. To that end, we put the model at checkpoint (17.7) through selective heteromodal input fine-tuning on the NMR subset previously used for validation, wherein the HSQC and ^{13}C–NMR modalities were omitted (frozen per sample), each with probability $\frac{1}{2}$. Thus, we had samples with either F/H, F/Q/H, F/C/H, or F/Q/C/H modalities in roughly equal measure. We fine-tuned for 20 epochs on ~70k such samples with $|\text{SMILES}| \leq 55$ and KL geometric weight $q_{\text{KL}} = 1\%$, observing signs of overfitting on the test set in the weighted KL loss. Evaluating this model on the test set showed a drop of ~10% top–10 accuracy on complete-modality samples, for example, $81\% \searrow 71\%$ on $|\text{SMILES}| \leq 55$ for stereo-isomer matches (cf. Table 3). Interestingly, the same top–10 accuracy was better on F/Q/H (~62%) than on F/C/H (~59%), which suggests that the model paid more attention to HSQC than ^{13}C–NMR and/or found it more informative (in line with [44, §R&D]).

4 Discussion and outlook

We have developed a multi-task language transformer model based on DistilGPT2 that reads compact textual annotations of multimodal spectroscopic data and simultaneously predicts a chemical vector embedding, the functional group counts, and the molecular structure. Our main contribution is the representation of the molecule as a graph of *all* equivalent serializations (GRIMACE). In each pass over the training data, the model is trained on one particular serialization with teacher forcing, but is supervised on the next-token distribution of all possible continuations towards the same molecule.

A notable caveat in our results is that the size of the complete GRIMACE grows rapidly with the size of the molecule, and our approach to approximate it in reasonable time, described in §2.7, becomes inadequate. This may be the reason behind the accuracy breakdown we observed on molecules with SMILES beyond ~55 characters, see Fig. 3. It would be desirable to have a fast method to determine the next-token distribution given a SMILES prefix.

One indication that training on GRIMACE allows the model to internalize the chemical structure, rather than syntax or dataset-specific chunks of SMILES, is the diversity of the predicted SMILES. Indeed, in ~38% of test samples, our model at checkpoint (17.7) produces the correct structure in *all* of the top–10 beam-search predictions. By contrast, this number is less than 1% for the F/Q/C/H → SMILES model trained only on (mostly) canonical SMILES (see §3.1), even though it has higher top–10 accuracy. Our data-informed confidence estimator ingests this information via the majority-vote feature (§3.3). It would be interesting to understand how the model prioritizes a particular serialization as a function of the NMR input, for example, whether easier functional groups appear earlier.

We believe that, in order to be useful and drive decisions in the lab, reliable uncertainty quantification for model predictions is crucial. Supervision and inference with GRIMACE provides the space to express uncertainty via prediction diversity.

References

[1] M Alberts et al. “Unraveling molecular structure: A multimodal spectroscopic dataset for chemistry”. *NeurIPS 2024 Datasets and Benchmarks Track*. 2024.

[2] TDW Claridge. *High-Resolution NMR Techniques in Organic Chemistry*. Elsevier, 2016.

[3] P Klukowski and P Riek Roland Güntert. “Machine learning in NMR spectroscopy”. *Progress in Nuclear Magnetic Resonance Spectroscopy* 148–149 (2025), p. 101575.

[4] Y Luo et al. “Deep learning and its applications in nuclear magnetic resonance spectroscopy”. *Progress in Nuclear Magnetic Resonance Spectroscopy* 146–147 (2025), p. 101556.

[5] VK Shukla, GT Heller, and DF Hansen. “Biomolecular NMR spectroscopy in the era of artificial intelligence”. *Structure* 31.11 (2023), pp. 1360–1374.

[6] K Guo et al. “Artificial intelligence in spectroscopy: Advancing chemistry from prediction to generation and beyond”. *arXiv* 2502.09897 (2025).

[7] C Cobas. “NMR signal processing, prediction, and structure verification with machine learning techniques”. *Magnetic Resonance in Chemistry* 58.6 (2020), pp. 512–519.

[8] D Chen et al. “Review and prospect: Deep learning in nuclear magnetic resonance spectroscopy”. *Chemistry – A European Journal* 26.46 (2020), pp. 10391–10401.

[9] I Cortés et al. “Machine learning in computational NMR-aided structural elucidation”. *Frontiers in Natural Products* 2 (2023), p. 1122426.

[10] H Flores-Hernandez and E Martinez-Ledesma. “A systematic review of deep learning chemical language models in recent era”. *Journal of Cheminformatics* 16.129 (2024).

[11] E Jonas, S Kuhn, and N Schlörer. “Prediction of chemical shift in NMR: A review”. *Magnetic Resonance in Chemistry* 60.11 (2022), pp. 1021–1031.

[12] GB Goh et al. “SMILES2Vec: An interpretable general-purpose deep neural network for predicting chemical properties”. *ICLR 2018*. 2017.

[13] S Jaeger, S Fulle, and S Turk. “Mol2vec: Unsupervised machine learning approach with chemical intuition”. *Journal of Chemical Information and Modeling* 58.1 (2018), pp. 27–35.

[14] W Ahmad et al. “ChemBERTa-2: Towards chemical foundation models”. *arXiv* 2209.01712v1 (2022).

[15] Y Liu et al. “RoBERTa: a robustly optimized BERT pretraining approach”. *arXiv* 1907.11692 (2019).

[16] S Kim et al. “PubChem 2025 update”. *Nucleic Acids Research* 53.D1 (2025), pp. D1516–D1525.

[17] Z Wu et al. “MoleculeNet: a benchmark for molecular machine learning”. *Chemical Science* 9.2 (2018), pp. 513–530.

[18] EJ Bjerrum. “SMILES enumeration as data augmentation for neural network modeling of molecules”. *arXiv* 1703.07076 (2017).

[19] EJ Bjerrum and B Sattarov. “Improving chemical autoencoder latent space and molecular de novo generation diversity with heteroencoders”. *Biomolecules* 8.4 (2018), p. 131.

[20] R Winter et al. “Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations”. *Chemical Science* 10.6 (2019), pp. 1692–1701.

[21] X-C Zhang et al. “Pushing the boundaries of molecular property prediction for drug discovery with multitask learning BERT enhanced by SMILES enumeration”. *Research* 2022 (2022), Article ID: 0004.

[22] J Devlin et al. “BERT: pre-training of deep bidirectional transformers for language understanding”. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019, pp. 4171–4186.

[23] MA Skinnider. “Invalid SMILES are beneficial rather than detrimental to chemical language models”. *Nature Machine Intelligence* 6 (2024), pp. 437–448.

[24] J Arús-Pous et al. “Randomized SMILES strings improve the quality of molecular generative models”. *Journal of Cheminformatics* 11 (2019), Article number: 71.

[25] Z Alperstein, A Cherkasov, and JT Rolfe. “All SMILES variational autoencoder for molecular property prediction and optimization”. *QSPR/QSAR analysis using SMILES and quasi-SMILES*. Springer, Cham, 2023, pp. 85–115.

[26] Z Huang et al. “A framework for automated structure elucidation from routine NMR spectra”. *Chemical Science* 12.46 (2021), pp. 15329–15338.

[27] F Hu et al. “Accurate and efficient structure elucidation from routine one-dimensional NMR spectra using multitask machine learning”. *ACS Central Science* 10.11 (2024), pp. 2162–2170.

[28] E Jonas. “Deep imitation learning for molecular inverse problems”. *Advances in Neural Information Processing Systems*. Vol. 32. 2019.

[29] Z Wu et al. “A comprehensive survey on graph neural networks”. *IEEE transactions on neural networks and learning systems* 32.1 (2021), pp. 4–24.

[30] E Jonas and S Kuhn. “Rapid prediction of NMR spectral properties with quantified uncertainty”. *Journal of Cheminformatics* 11 (2019), Article number: 50.

[31] Z Yang et al. “Cross-modal retrieval between ^{13}C NMR spectra and structures for compound identification using deep contrastive learning”. *Analytical Chemistry* 93.50 (2021), pp. 16947–16955.

[32] B Sridharan et al. “Deep reinforcement learning for molecular inverse problem of NMR spectra to molecular structure”. *Journal of Physical Chemistry Letters* 13.22 (2022), pp. 4924–4933.

[33] D Silver et al. “Mastering the game of Go without human knowledge”. *Nature* 550 (2017), pp. 354–359.

[34] S Kuhn and NE Schlörer. “Facilitating quality control for spectra assignments of small organic molecules: nmrshiftdb2 – a free in-house NMR database with integrated LIMS for academic service laboratories”. *Magnetic Resonance in Chemistry* 53.8 (2015), pp. 582–589.

[35] L Yao et al. “Conditional molecular generation net enables automated structure elucidation based on ^{13}C NMR spectra and prior knowledge”. *Analytical Chemistry* 95.12 (2023), pp. 5393–5401.

[36] JJ Irwin et al. “ZINC: a free tool to discover chemistry for biology”. *Journal of Chemical Information and Modeling* 52.7 (2012), pp. 1757–1768.

[37] JL López-Pérez et al. “NAPROC-13: a database for the dereplication of natural product mixtures in bioassay guided protocols”. *Bioinformatics* 23.23 (2007), pp. 3256–3257.

[38] M Alberts, F Zipoli, and AC Vaucher. “Learning the language of NMR: Structure elucidation from NMR spectra using transformer models”. *ChemRxiv* (2023).

[39] NextMove Software. *Pistachio (reaction data, querying and analytics)*. 2025.

[40] D Lowe. *Chemical reactions from US patents (1976–Sep2016)*. figshare, 2017.

[41] M Alberts, N Hartrampf, and T Laino. “Automated structure elucidation at human-level accuracy via a multimodal multitask language model”. *ChemRxiv* (2025).

[42] SE Stein. *NIST/EPA gas-phase infrared database (SRD 35)*. 2008.

[43] SE Van Bramer and LD Bastin. “Spectroscopy data for undergraduate teaching”. *Journal of Chemical Education* 100.10 (2023), pp. 3897–3902.

[44] M Priessner et al. “Enhancing molecular structure elucidation: MultiModalTransformer for both simulated and experimental spectra”. *ChemRxiv* (2024).

[45] A Tibo et al. “Exhaustive local chemical space exploration using a transformer model”. *Nature Communications* 15 (2024), Article 7315.

[46] Q Yang et al. “DiffNMR: Diffusion models for NMR spectra elucidation”. *arXiv* 2507.08854 (2025).

[47] Mestrelab Research. *MestReNova 15.1 Manual*. Mestrelab Research, 2024.

[48] L Wang et al. “Diffusion models for molecules: A survey of methods and tasks”. *arXiv* 2502.09511 (2025).

[49] D Weininger. “SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules”. *Journal of Chemical Information and Computer Sciences* 28.1 (1988), pp. 31–36.

[50] D Weininger, A Weininger, and JL Weininger. “SMILES. 2. Algorithm for generation of unique SMILES notation”. *Journal of Chemical Information and Computer Sciences* 29.2 (1989), pp. 97–101.

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

[52] N Schneider, RA Sayle, and GA Landrum. “Get your atoms in order—An open-source implementation of a novel and robust molecular canonicalization algorithm”. *Journal of Chemical Information and Modeling* 55.10 (2015), pp. 2111–2120.

[53] A Radford et al. *Language models are unsupervised multitask learners*. Tech. rep. 2019.

[54] G Hinton, O Vinyals, and J Dean. “Distilling the knowledge in a neural network”. *NIPS 2014 Deep Learning Workshop* (2014).

[55] Hugging Face. *DistilGPT2 model card*. 2020.

[56] P Gage. “A new algorithm for data compression”. *The C Users Journal* 12.2 (1994), pp. 23–38.

[57] R Sennrich, B Haddow, and A Birch. “Neural machine translation of rare words with subword units”. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)* (2016), pp. 1715–1725.

[58] A Karpathy. *Let’s build the GPT tokenizer (YouTube video)*. 2024. https://youtu.be/zduSFxRajkE

[59] A Wadell, A Bhutani, and V Viswanathan. “Smirk: An atomically complete tokenizer for molecular foundation models”. *arXiv* 2409.15370 (2024).
[60] B Millidge. “Integer tokenization is insane”. *beren.io* (2023).
[61] UV Ucak, I Ashyrmamatov, and J Lee. “Improving the quality of chemical language model outcomes with atom-in-SMILES tokenization”. *Journal of Cheminformatics* 15 (2023), Article 55.
[62] M Leon et al. “Comparing SMILES and SELFIES tokenization for enhanced chemical language modeling”. *Scientific Reports* 14 (2024), Article 25016.
[63] F Mastrolorito et al. “fragSMILES as a chemical string notation for advanced fragment and chirality representation”. *Communications Chemistry* 8 (2025), Article 26.
[64] J Wang et al. “Token-Mol 1.0: tokenized drug design with large language models”. *Nature Communications* 16 (2025), Article 4416.
[65] K Zheng et al. “SMI-Editor: Edit-based SMILES language model with fragment-level supervision”. *ICLR (poster, OpenReview)* (2025).
[66] I Fender, JA Gut, and T Lemmin. “Beyond performance: How design choices shape chemical language models”. *bioRxiv* (2025).
[67] BW Watson. “A taxonomy of finite automata minimization algorithms”. *Computing Science Note 93/44* (1993), pp. 1–23.
[68] J Daciuk et al. “Incremental construction of minimal acyclic finite-state automata”. *Computational Linguistics* 26.1 (2000), pp. 3–16.
[69] R Cipolla, Y Gal, and A Kendall. “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics”. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018).
[70] R Andreev. *Lightweight repo to train/evaluate NMR-to-SMILES models by Alberts et al.* 2025. https://github.com/numpde/nmr-to-structure-lite
[71] RJ Abraham and M Mobli. *Modelling ^{1}H NMR spectra of organic compounds: Theory, applications and NMR prediction software*. Wiley, 2008.
[72] PyTorch Core Team. *PyTorch documentation: AdamW optimizer*. PyTorch Foundation, 2025.
[73] I Loshchilov and F Hutter. “Decoupled weight decay regularization”. *arXiv* 1711.05101 (2017).
[74] P Zhou et al. “Towards understanding convergence and generalization of AdamW”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.9 (2024), pp. 6486–6493.
[75] F Pedregosa et al. “Scikit-learn: Machine learning in Python”. *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
[76] NM O’Boyle et al. “Open Babel: An open chemical toolbox”. *Journal of Cheminformatics* 3 (2011), p. 33.
[77] G Landrum. *RDKit: Open-source cheminformatics*. 2013. https://www.rdkit.org

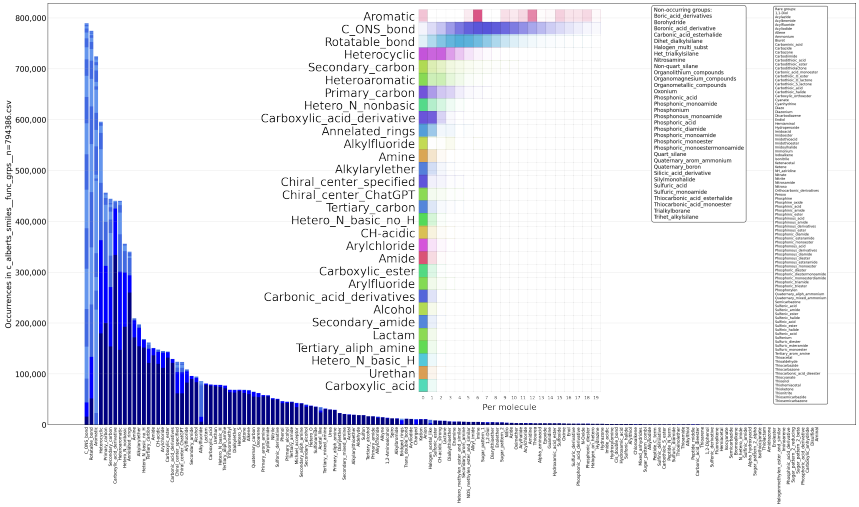
Many thanks to E. Konukoglu, N. Schmid, and M.-O. Ebert for a close reading of the manuscript, and to Lambda GPU Cloud for providing most of the compute.

For additional materials, visit https://numpde.github.io/shared/msc/

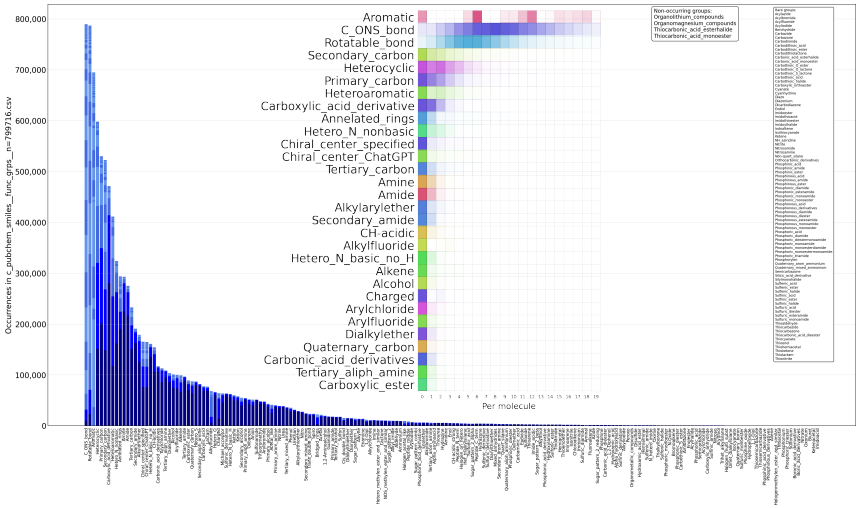
Additional figures and tables

```
# model_name = 'GPT2MultiHeadModel'
# model.num_parameters() = 82682607
# architecture:
GPT2MultiHeadModel(
  (transformer): GPT2Model(
    (wte): Embedding(50272, 768)
    (wpe): Embedding(1024, 768)
    (drop): Dropout(p=0.1, inplace=False)
    (h): ModuleList(
      (0-5): 6 x GPT2Block(
        (ln_1): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
        (attn): GPT2Attention(
          (c_attn): Conv1D(nf=2304, nx=768)
          (c_proj): Conv1D(nf=768, nx=768)
          (attn_dropout): Dropout(p=0.1, inplace=False)
          (resid_dropout): Dropout(p=0.1, inplace=False)
        )
        (ln_2): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
        (mlp): GPT2MLP(
          (c_fc): Conv1D(nf=3072, nx=768)
          (c_proj): Conv1D(nf=768, nx=3072)
          (act): NewGELUActivation()
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
    )
    (ln_f): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
  )
  (lm_head): Linear(in_features=768, out_features=50272, bias=False)
  (aux_heads): ModuleDict(
    (chemberta-v2): Sequential(
      (0): Linear(in_features=768, out_features=384, bias=True)
    )
    (func_group): Sequential(
      (0): Linear(in_features=768, out_features=289, bias=True)
      (1): Softplus(beta=1.0, threshold=20.0)
    )
    (tox21): Sequential(
      (0): Linear(in_features=768, out_features=12, bias=True)
      (1): Sigmoid()
    )
    (validity): Sequential(
      (0): Linear(in_features=768, out_features=1, bias=True)
      (1): Sigmoid()
    )
  )
  (aux_loss_fns): ModuleDict(
    (chemberta-v2): WhitenedNLL(n=384, mode='gauss', reduction='mean')
    (func_group): WhitenedNLL(n=289, mode='laplace', reduction='mean')
    (tox21): _FuncLoss()
    (validity): _FuncLoss()
  )
)
```

Table 5: Architecture of the main molecular structure inference model based on the Distil-GPT2 decoder-only transformer with auxiliary heads and losses for multi-task learning. The number of trainable parameters is ~83M with `wte`/`lm_head` weight tying (~121M without).

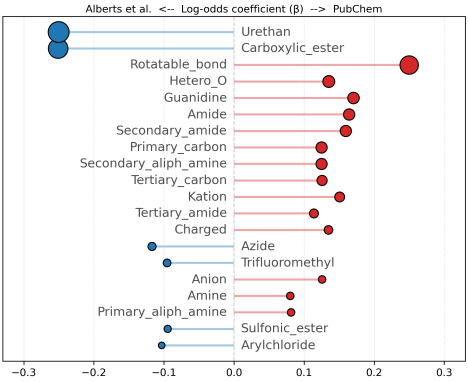


(a) The multimodal spectroscopic dataset of Alberts et al. [1], ~795k molecules.

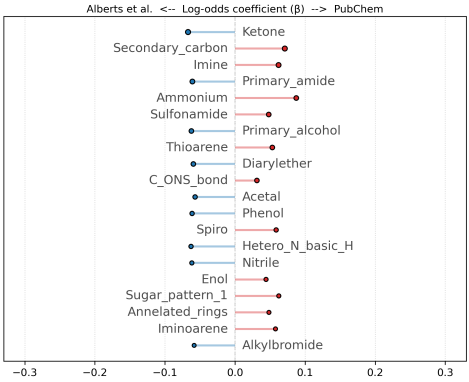


(b) The PubChem database [16], subsampled randomly to ~800k.

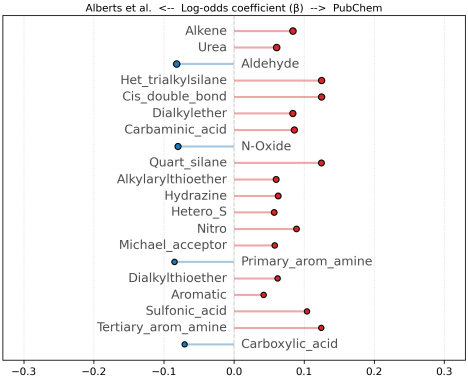
Figure 5: Functional group prevalence in two datasets. The histograms show the number of molecules containing a given functional group (the colors alternate with count). The inset shows the per-molecule distribution of the topmost functional groups (by total count). Functional group definitions were extracted as SMARTS patterns from the Open Babel [76] codebase and identified with RDKit [77]. We added `Chiral_center_ChatGPT`, which is equivalent to `Chiral_center_specified`.



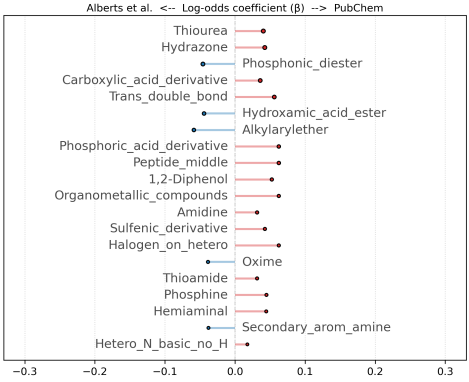
(a) 1 – 20



(b) 41 – 60

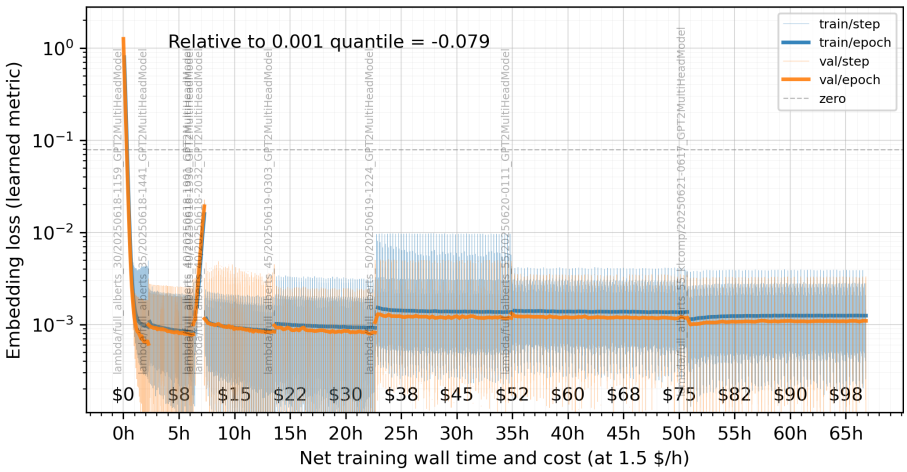


(c) 21 – 40

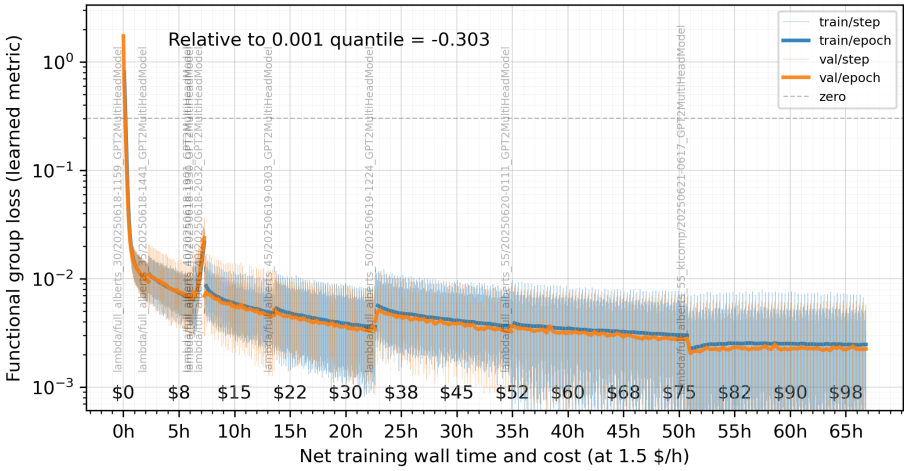


(d) 61 – 80

Figure 6: We wondered whether the multimodal spectroscopic dataset by Alberts et al. [1] is chemically "representative". To that end, we trained a sparse ElasticNet logistic regression classifier on the functional groups of a molecule to predict whether it was sampled from the multimodal spectroscopic dataset (*A*-like) or the PubChem database (*B*-like), cf. Fig. 5a/5b. The features, i.e., the functional group counts across the two datasets, were scaled to unit variance. The ROC–AUC of the model is ~75% (the likelihood that a random *B*-like molecule is deemed more *B*-like than a random *A*-like molecule). The log-odds coefficients β_j are represented in the above figure as bars, and only the top functional groups by $|\beta_j|$ are shown out of 281 non-zero coefficients. The circle area indicates the permutation importance of the functional group, i.e., the drop in the ROC–AUC when this functional group is randomly permuted, where the largest circle corresponds to ~2%. The topmost functional group, "uretan", a.k.a. *carbamate*, is found in protecting groups used during synthesis (as well as in polyurethanes and drugs/pesticides). This makes sense, as the molecules in multimodal spectroscopic dataset were sourced from the USPTO *reaction* dataset by Lowe [40].

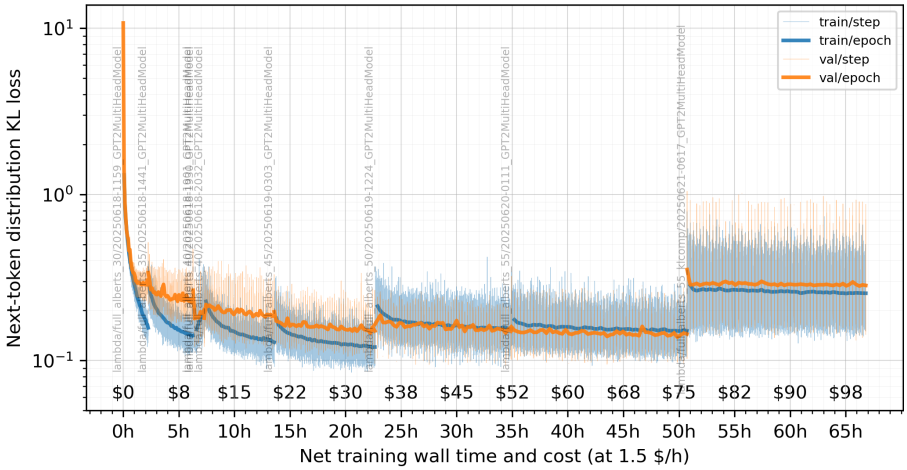


(a) The "chembedding" loss, offset by the 0.001 quantile for visibility.

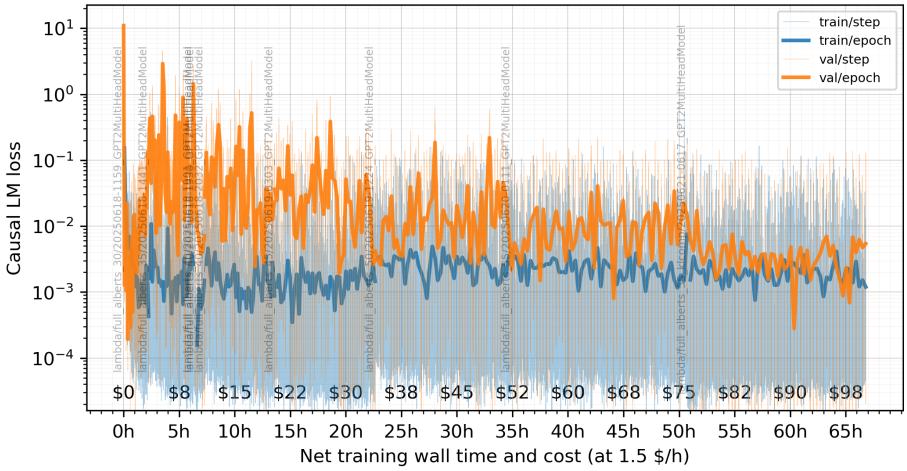


(b) The functional-group loss, offset by the 0.001 quantile for visibility.

Figure 7: Training losses for the training runs (17.1)–(17.7) as defined in §2.8.



(c) KL loss. The last run (17.7) shows the geometrically weighted KL loss.



(d) Cross-entropy (Causal-LM) loss for the end-of-sequence token.

Figure 7: Training losses (cont'd) for the training runs (17.1)–(17.7) as defined in §2.8.

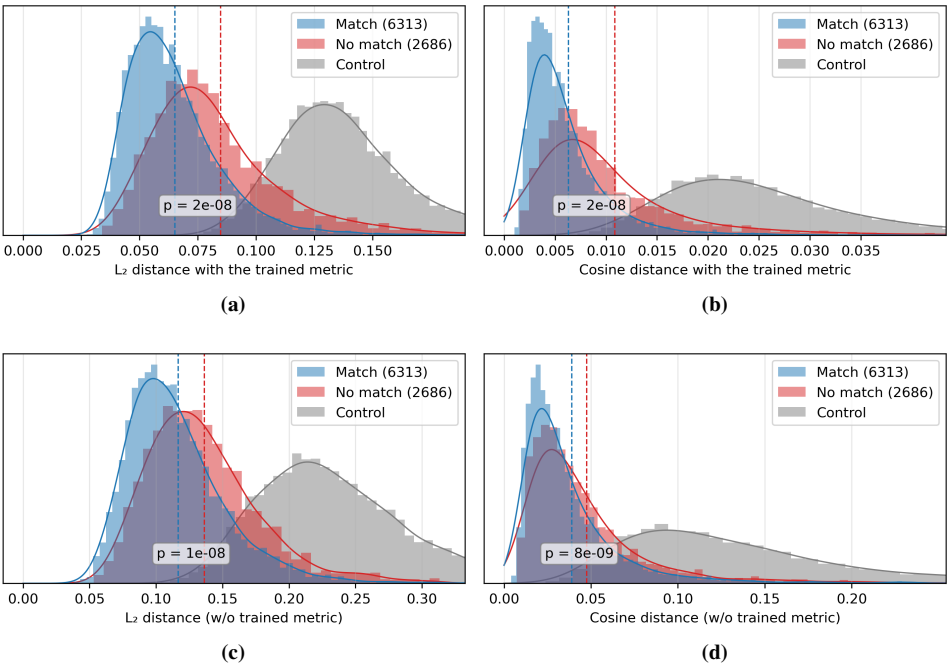


Figure 8: The distance between the "chembedding" predicted by the model at checkpoint (17.7) and the "chembedding" of the canonical reference SMILES, stratified by whether there is a top–10 exact structure match. As control, the distances between all the reference SMILES are also shown. Left *vs* right: measured in dimension-normalized L_2-norm $\frac{1}{\sqrt{n}}\|\cdot\|_2$ *vs* measured using cosine distance $(1-\cos\angle)$. Top *vs* bottom: the trainable metric-matrix from (10) is applied first *vs* omitted. The p-value shown is computed using the Cramér–von Mises test between the emperical data of "match" and "no match".

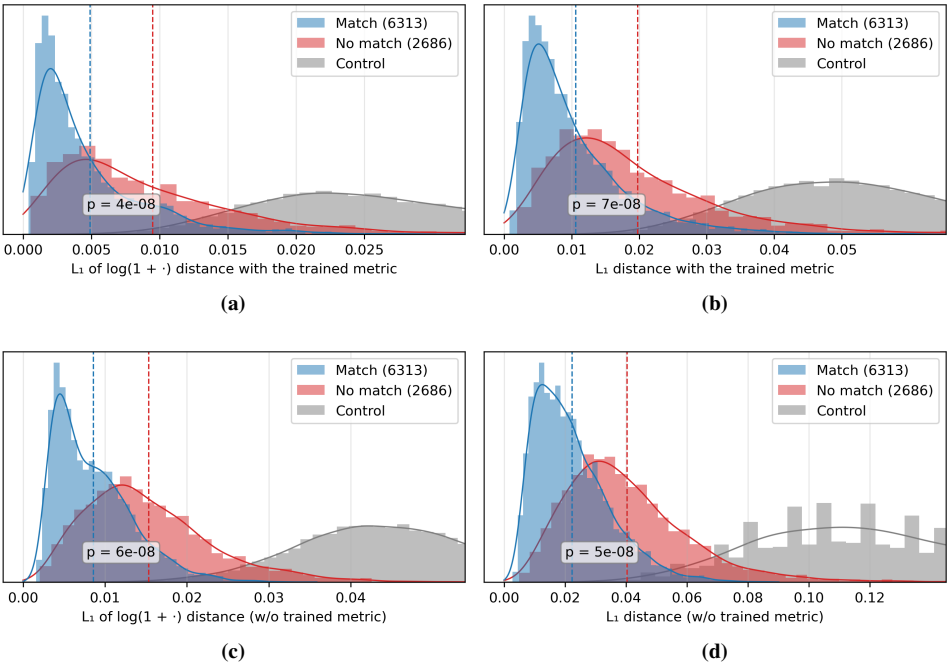


Figure 9: Similarly to Fig. 8, we evaluate functional group counts predicted by the model at checkpoint (17.7), measured in dimension-normalized L_1-norm $\frac{1}{m}\|\cdot\|_1$. Top *vs* bottom: the trainable metric-matrix from (12) is applied *vs* omitted. Left *vs* right: with $\log(1+\cdot)$ *vs* without. Computation of the p-value as in Fig. 8. The "comb" appears in (d) because of discrete values of the untransformed counts.

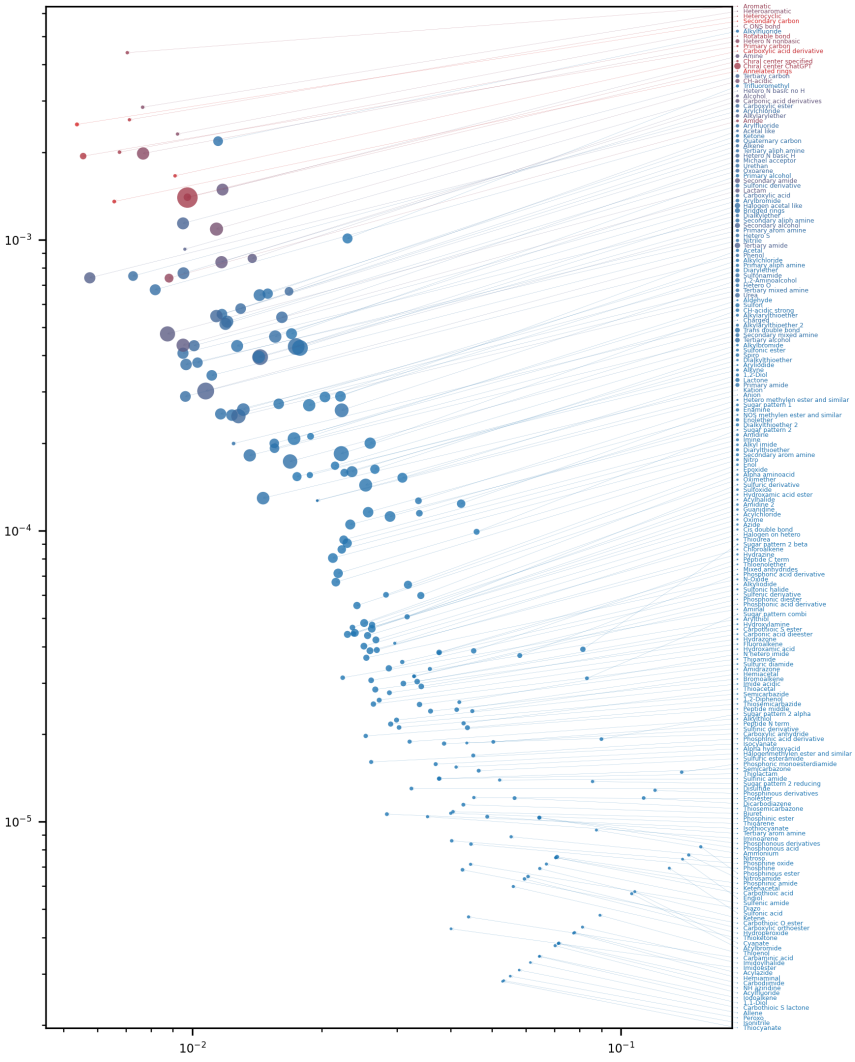
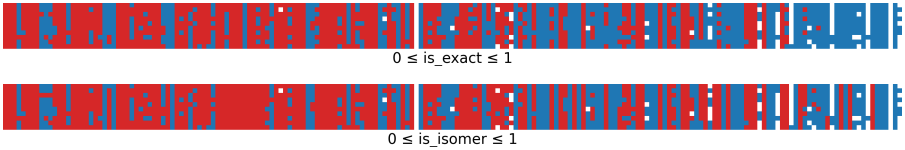
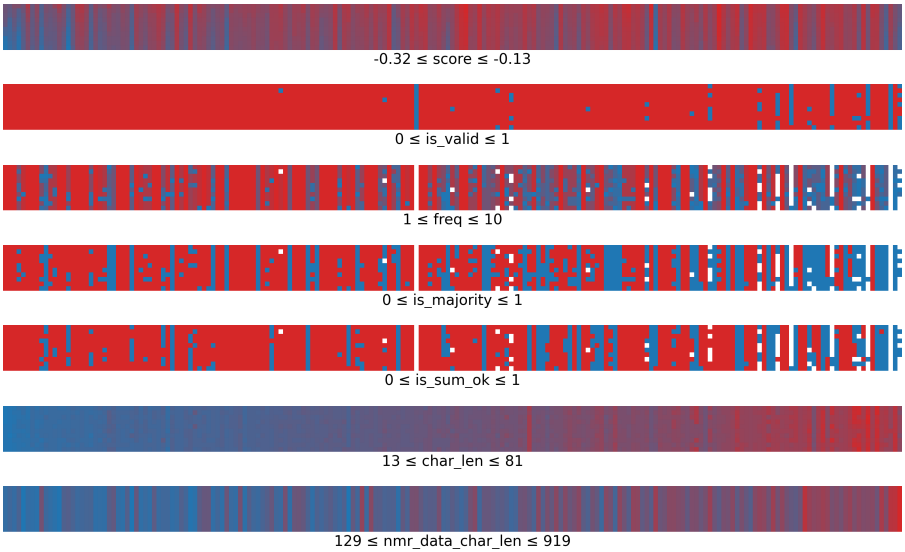


Figure 10: Errors per functional group g, evaluated on a random 9k subset of the test set, cf. §3.2. Set $f_{ig} := \log(1 + \text{count})$ of g in sample i, and $\hat{f}_{ig}$ for its predicted value (model checkpoint (17.7)). The area is proportional to $\sum_i |\sum_h S_{gh}(f_{ih} - \hat{f}_{ih})|$, i.e., the contribution of the *component* g in the S-weighted loss. The color (blue $\to$ red) represents $\sum_i |f_{ig} - \hat{f}_{ig}|$, i.e., the contribution of g to the vanilla L_1-error. Let c_g be the column of g in the data covariance matrix C. Let s_g be a column of the product matrix SC. The y–coordinate of g is $\frac{1}{m}\|s_g\|_1$, which estimates the contribution of g to the loss. The x–coordinate is $\frac{1}{m}\|s_g\|_1/c_{gg}$, which indicates how much g is discounted by the metric (lower means "squashed").

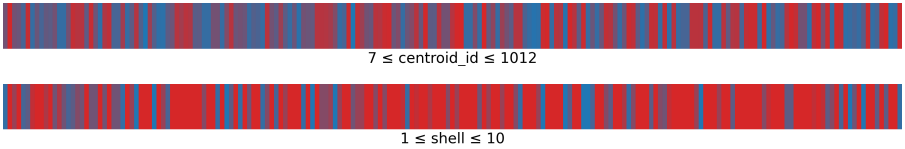


(a) Whether the candidate is an exact structural match (top); or a stereo-isomer (bottom).



(b) Input features to the meta-classifier as described in §3.3.

(c) Character length of the canonical SMILES of the target.



(d) Centroid and shell number of the reference molecule in the "cheese" dataset split of §2.4.

Figure 11: Features of the top–10 beam-search candidates (↕) of 200 random samples (↔) from the test set (cf. §2.4) by the model checkpoint (17.7). Sorted by character length of the reference SMILES. Red: high value; blue: low value; white: not a valid SMILES.