Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Digital Discovery*, 2025, **4**, 2876

Inconsistency of LLMs in molecular representations

Bing Yan, [a] Angelica Chen [b] and Kyunghyun Cho *[ab]

Large language models (LLM) have demonstrated remarkable capabilities in chemistry, yet their ability to capture intrinsic chemistry remains uncertain. Within any familiar, chemically equivalent representation family, rigorous chemical reasoning should be representation-invariant, yielding consistent predictions across these representations. Here, we introduce the first systematic benchmark to evaluate the consistency of LLMs across key chemistry tasks. We curated the benchmark using paired representations of SMILES strings and IUPAC names. We find that the state-of-the-art general LLMs exhibit strikingly low consistency rates (≤1%). Even after finetuning on our dataset, the models still generate inconsistent predictions. To address this, we incorporate a sequence-level symmetric Kullback–Leibler (KL) divergence loss as a consistency regularizer. While this intervention improves surface-level consistency, it fails to enhance accuracy, suggesting that consistency and accuracy are orthogonal properties. These findings indicate that both consistency and accuracy must be considered to properly assess LLMs' capabilities in scientific reasoning.

Received 30th April 2025
Accepted 5th August 2025

DOI: 10.1039/d5dd00176e

rsc.li/digitaldiscovery

1 Introduction

Large language models (LLM) have rapidly become powerful tools across scientific domains, including chemistry. They have demonstrated impressive capabilities in tasks such as molecule design, property prediction, and synthesis planning.[1–6] In these applications, LLMs are typically trained on textual encodings of molecules, often as sequences such as SMILES, the simplified molecular input line entry system,[7] or IUPAC names, the standardized nomenclature for chemicals.[8] Despite their success, a fundamental question remains (Fig. 1): do LLMs truly understand the intrinsic chemistry of molecules (pink pathway), or do they merely exploit surface-level textual patterns (blue pathway)?

In principle, rigorous chemical reasoning should be independent of how a molecule is represented. A knowledgeable chemist, or an AI model with true chemical understanding, should draw the same conclusions about a molecule whether given its 2D graph, SMILES string, or IUPAC name. In other words, the representation should not influence the reasoning process or the outcome. This expectation aligns with the broader principle of self-consistency in AI models, which requires responses to remain invariant under semantics-preserving transformations of the input.[9]

However, if a model's reasoning depends on the chosen representation, logically equivalent inputs may yield different outcomes. This issue has been documented in natural language processing, where LLMs often produce contradictory responses when the same question is phrased in different ways or when the context is reworded. For instance, GPT-3 and GPT-4 exhibit poor self-consistency on multi-step reasoning tasks, giving different answers to re-framed but logically equivalent queries.[9]

A similar phenomenon has been observed in computer vision: image classifiers can learn superficial cues, such as texture rather than capturing the true shape of an object. As a result, a trivial change in surface pattern can lead to entirely different predictions for the same underlying object.[10] These examples from language and vision highlight a broader failure mode: when reasoning hinges on how information is presented instead of its intrinsic meaning, the model's reliability is compromised.

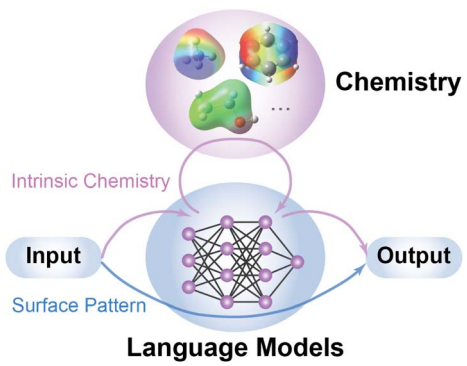


Fig. 1 Illustration of how language models approach predictions for chemistry tasks. It remains unclear whether their predictions rely on surface-level patterns in molecular representations (blue pathway) or on the intrinsic chemical properties (pink pathway) of the molecules.

[a]*Department of Computer Science, New York University, 60 5th Avenue, New York, NY 10011, USA. E-mail: kyunghyun.cho@nyu.edu*

[b]*Center for Data Science, New York University, 60 5th Avenue, New York, NY 10011, USA*

Despite the growing use of LLMs in chemistry, their consistency across different molecular representations has not been systematically evaluated. To address this gap, we introduce a benchmark to assess whether LLMs exhibit representation-invariant reasoning. We curated a paired dataset of molecules with both SMILES and IUPAC representations, spanning multiple chemistry tasks, including forward reaction prediction, retrosynthesis, and molecular property prediction. By evaluating LLMs on each task using both input formats, we can compute a consistency rate—the percentage of cases where the model produces identical predictions for SMILES and IUPAC representations. Our results show that state-of-the-art general-purpose LLMs exhibit a low consistency rate ($\leq$1%). Even after finetuning on our paired dataset, the models remain inconsistent, suggesting that they rely more on superficial text patterns than on the underlying chemistry.

Can this inconsistency be easily remedied? To explore this, we investigated whether a simple training intervention could enforce representation-invariant behavior. Specifically, we introduced a sequence-level symmetric Kullback–Leibler (KL) divergence loss as a consistency regularizer. This approach penalizes the model when its output distributions differ for the same molecule presented in different formats. While this regularization strategy led to mild improvements in consistency, the gains were limited – models still frequently produced diverging predictions depending on the input format. Furthermore, this intervention did not improve accuracy. The models became more likely to generate the same prediction for a given molecule, regardless of representation, but not necessarily the correct one. This suggests that consistency and accuracy are orthogonal properties, and that we must consider both to assess LLMs' capabilities in capturing intrinsic chemistry.

The persistence of inconsistency indicates a deeper, systematic issue in how LLMs learn chemistry that cannot be easily fixed with finetuning alone. Addressing this challenge will likely require fundamental advances. More broadly, our findings highlight a key requirement for AI-driven scientific reasoning: models should respect the domain's natural invariances to be reliable. By rigorously benchmarking this consistency gap, we take a step toward developing more trustworthy AI systems that reason based on substance rather than surface patterns.

2 Experiments

2.1 Problem setup

We study three chemistry tasks, forward reaction prediction, retrosynthesis, and property prediction, each formulated as a conditional generation problem: given an input sequence x, predict an output sequence y.

LLMs predict the output distribution $P_\theta(y|x)$, where θ denotes model parameters. The input molecules can be encoded in different formats (*e.g.*, SMILES, IUPAC names), leading to different output distributions, $P_\theta(y|x_S)$ for SMILES and $Q_\theta(y|x_I)$ for IUPAC. We evaluate consistency by comparing these distributions to assess whether models capture the intrinsic chemistry underlying symbolic representations.

2.2 Evaluation metrics

We evaluate model performance using two key metrics:

Consistency measures how often a model produces identical predictions for the same molecule when presented in different formats (SMILES *vs.* IUPAC). For forward reaction prediction and retrosynthesis: a prediction is considered consistent if the outputs match for both input representations. For binary property prediction: consistency is measured as the proportion of cases where classification outcome remains the same. For numeric property prediction: consistency is measured using the mean squared error (MSE) between predictions from SMILES and IUPAC inputs.

To distinguish cross-representation alignment from chance-level agreement, we report adjusted consistency, defined as the observed consistency minus a random-consistency baseline. For forward reaction prediction, retrosynthesis, and binary property prediction, the baseline is the expected match rate between two independent random predictions. For numeric property prediction, we subtract the expected MSE between two random predictions. Unless otherwise noted, all reported consistency values are adjusted.

Accuracy evaluates how closely model predictions align with the ground truth. For forward reaction prediction and retrosynthesis: accuracy is the percentage of exact matches between the predicted and target outputs in each format. For binary property prediction: accuracy is the percentage of correct classifications. For numeric property prediction: accuracy is measured as the MSE between predicted and ground truth.

Formal definitions and equations for both metrics are provided in Appendix A.1.

2.3 Evaluation of state-of-the-art LLMs

We evaluated the consistency and accuracy of state-of-the-art general LLMs for forward reaction prediction. The models include GPT-4,[11] GPT-4o,[12] o1-preview, o1-mini,[13] o3-mini,[14] Claude 3 Opus,[15] Llama 3.1 8B,[16] and the instruction-tuned $\text{LlaSMol}_{\text{Mistral}}$.[17] A test set of 300 chemical reactions was used.

We provided explicit instructions tailored to the input and output molecular representations. For instance, when both the input and output were in SMILES format, the instruction read: "Based on the SMILES strings of reactants and reagents, predict the SMILES string of the product. Please output the product directly."

2.4 Finetuning LLMs with mapped SMILES & IUPAC data

To mitigate biases in pretrained data, we finetuned GPT-2, Mistral 7B, and CodeT5 on carefully curated datasets where each input molecule had a one-to-one mapped SMILES and IUPAC representation. This setup isolates the impact of input format while preserving underlying chemistry. To further assess the effect of pretraining, we also finetuned a randomly initialized GPT-2 model.

For forward reaction prediction and retrosynthesis, models were trained to generate either SMILES or IUPAC outputs with

equal probability, indicated by a flag ("S" for SMILES, "I" for IUPAC). All models were optimized using cross-entropy loss.

We further examined the effect of model size by training four GPT-2 variants (124M, 355M, 774M, and 1.5B parameters). To estimate variability, we ran experiments with different random seeds. The training hyperparameters and implementation details are provided in Appendix B.1.1 and B.1.2.

2.5 Sequence-level KL divergence loss

To improve consistency across molecular representations, we introduce a sequence-level KL divergence loss to minimize divergence between the probabilistic distributions generated from SMILES and IUPAC inputs, $P_\theta(y|x_S)$ and $Q_\theta(y|x_I)$.

We consider both directions of the KL divergence, $D_{KL}(P\|Q)$ and $D_{KL}(Q\|P)$:

$$D_{KL}(P \parallel Q) = \sum_{y\in Y} P_\theta(y|x_S)\log\frac{P_\theta(y|x_S)}{Q_\theta(y|x_I)}$$
$$D_{KL}(Q \parallel P) = \sum_{y\in Y} Q_\theta(y|x_I)\log\frac{Q_\theta(y|x_I)}{P_\theta(y|x_S)} \quad (1)$$

where Y is the set of all possible output sequences.

However, the sequence-level KL divergence is computationally intractable. Therefore, we estimate the KL divergence using Monte Carlo sampling method. Details of KL divergence loss can be found in Appendix C.1.

2.6 SMILES ↔ IUPAC translation

To study whether LLMs learn an internal mapping between SMILES and IUPAC representations, we evaluated models on the SMILES ↔ IUPAC translation task. We used o3-mini as a representative commercial LLM and GPT-2 small finetuned on forward reaction prediction as a representative open-source baseline.

We also examined whether translation pretraining improves downstream performance. Specifically, we first trained a GPT-2 small model on a SMILES ↔ IUPAC translation dataset, then finetuned it on the forward reaction prediction task, with and without the addition of KL divergence loss.

2.7 Data

We base our work on the SMolInstruct dataset, which is a large-scale instruction-tuning dataset for chemistry.[17] We used the "property prediction", "chemical reaction", and "name conversion: IUPAC to SMILES and SMILES to IUPAC" subsets. We used the official training, validation, and test splits provided by the SMolInstruct dataset. For evaluation, we uniformly sampled 300 examples when the test set contains more than 300 examples.

The original "property prediction" and "chemical reaction" subsets use SMILES representation. We translated SMILES into IUPAC to construct one-to-one mapped input datasets. For each molecule, we first used `PubChemPy`,[18] a Python wrapper for the PubChem PUG REST API, to retrieve its IUPAC name. If no IUPAC name was found, we used `Chemical-Converters`,[19] an open-source model to translate SMILES into IUPAC. We validated the translation using `pyopsin`, a Python wrapper for OPSIN.[20]

The training datasets for the forward reaction prediction and retrosynthesis both consist of 1M entries. For most models, we used an 80k subset for finetuning. To evaluate the impact of dataset size, we trained a GPT-2 model on the full dataset. We filtered the "name conversion" dataset by removing examples with more than one molecule. The statistics of all datasets are listed in Appendix Table 4.

3 Results and discussion

3.1 Evaluation of state-of-the-art LLMs

We evaluated the consistency and accuracy of forward reaction prediction across seven state-of-the-art LLMs, focusing on their performance when using SMILES *versus* IUPAC input representations. The results revealed four key insights (Fig. 2).

First, across all models, the adjusted consistency scores ranged from 0% to 1%, revealing a poor alignment between SMILES and IUPAC representations. The result indicates that LLMs struggle to maintain consistent outputs when given different input representations.

Second, LLMs without instruction tuning achieved higher accuracy for IUPAC inputs. This discrepancy is likely due to the training data distribution, which tends to include more examples using IUPAC,[21–23] providing the models with a familiarity advantage for this representation.

Third, models optimized for reasoning, such as o1-preview, demonstrated improved accuracy, but the increase in accuracy did not lead to a comparable increase in consistency. This observation suggests that accuracy and consistency are orthogonal metrics. We explored the orthogonality further in the discussion.

Finally, the instruction-tuned model, $LlaSMol_{Mistral}$, achieved significantly higher accuracy with SMILES inputs, reflecting the impact of its SMILES-specific training. However, this tuning did not improve accuracy with IUPAC inputs, indicating a limited generalization between the two representations. This result highlights a key limitation of current LLMs—

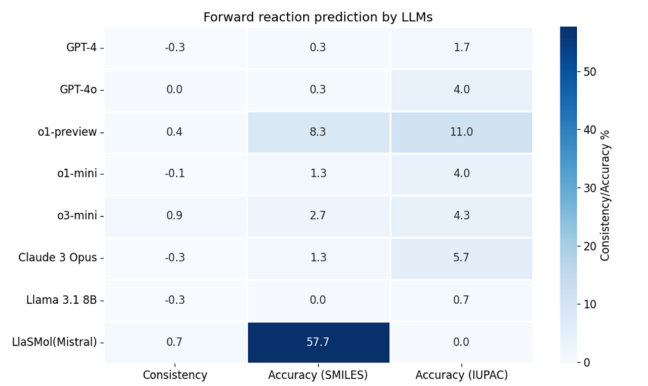


Fig. 2 Consistency (adjusted) and accuracy of forward reaction predictions by state-of-the-art LLMs. Across all models, consistency remains low. Most models exhibit higher accuracy for IUPAC inputs, except for $LlaSMol_{Mistral}$, which is instruction-tuned on a SMILES dataset. Darker colors represent higher values, while lighter colors indicate lower values.

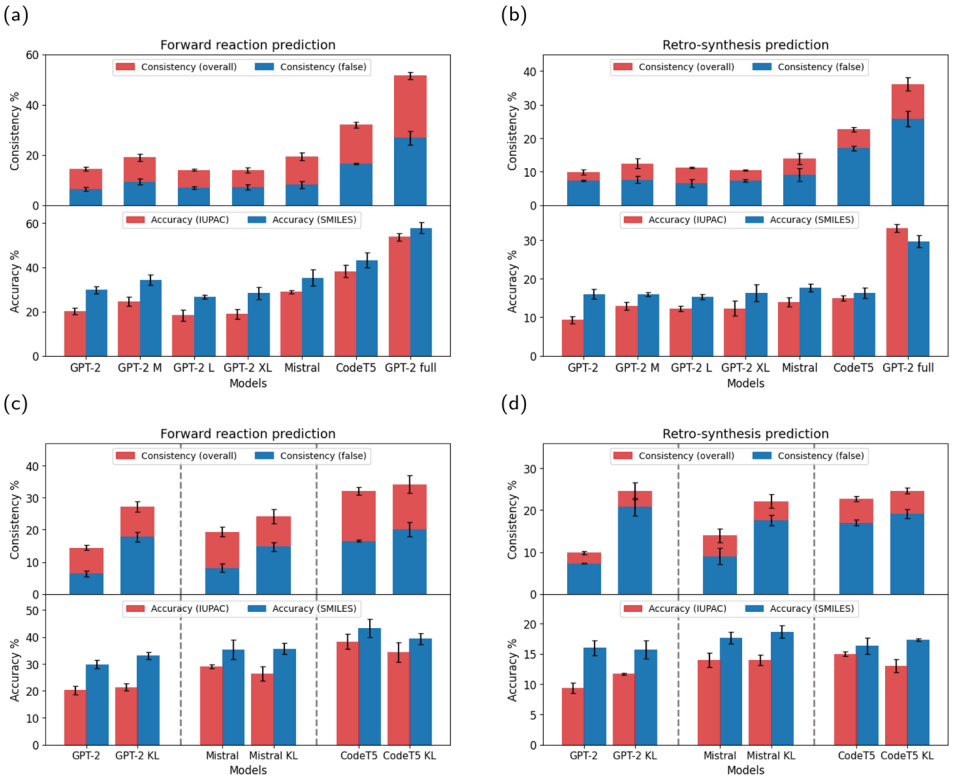


Fig. 3 Consistency and accuracy of LLMs in (a) (c) forward reaction prediction and (b) (d) retrosynthesis after finetuning on one-to-one mapped data. The finetuning of (c) and (d) has added a KL divergence loss. The overall consistency (red) and false consistency (blue) are overlaid. Most models are finetuned on an 80k dataset subset, except for "GPT-2 full" – a GPT-2 small model trained on the full 1M dataset. Error bars represent the standard deviation across training runs with varying random seeds.

Table 1 Consistency (raw and adjusted) and accuracy of LLMs in binary property prediction after finetuning (columns 3–6) and with KL divergence loss (columns 7–10). Entries that improve with the addition of KL divergence loss are highlighted in bold. Error bars represent the standard deviation across training runs with varying random seeds. An upward arrow (↑) indicates that higher values correspond to better performance

		Performance (%)↑				Performance w/KL (%)↑			
Properties	Models	Consist	Adj. consist	Acc. (S)	Acc. (I)	Consist	Adj. consist	Acc. (S)	Acc. (I)
BBBP	GPT-2	83.6 ± 1.1	26.9 ± 1.1	83.6 ± 1.7	81.0 ± 2.1	**91.5 ± 1.8**	**34.8 ± 1.8**	86.2 ± 0.9	**82.0 ± 1.1**
	Mistral	85.2 ± 6.8	28.5 ± 6.8	68.3 ± 5.8	76.7 ± 1.3	**90.5 ± 1.1**	**33.8 ± 1.1**	**84.1 ± 4.3**	**78.8 ± 5.3**
	CodeT5	85.7 ± 2.0	29.0 ± 2.0	85.7 ± 0.3	85.2 ± 2.9	**88.9 ± 2.4**	**32.2 ± 2.4**	**86.2 ± 1.5**	82.5 ± 0.3
ClinTox	GPT-2	95.4 ± 1.9	9.5 ± 1.9	93.1 ± 0.4	91.6 ± 1.5	**96.2 ± 2.0**	**10.3 ± 2.0**	93.1 ± 1.2	**92.4 ± 0.0**
	Mistral	100 ± 4.8	14.1 ± 4.8	92.4 ± 0.0	92.4 ± 4.0	99.2 ± 0.4	13.3 ± 0.4	92.4 ± 0.0	91.6 ± 0.4
	CodeT5	87.0 ± 2.0	1.1 ± 2.0	89.3 ± 1.2	85.5 ± 3.1	**94.7 ± 0.4**	**8.8 ± 0.4**	**91.6 ± 0.9**	**90.8 ± 1.2**
HIV	GPT-2	97.3 ± 0.7	6.2 ± 0.7	95.3 ± 0.4	95.3 ± 0.3	**98.3 ± 0.0**	**7.2 ± 0.0**	**96.3 ± 0.3**	95.3 ± 0.2
	Mistral	99.7 ± 0.2	8.6 ± 0.2	95.7 ± 0.2	95.3 ± 0.0	99.7 ± 0.2	8.6 ± 0.2	95.3 ± 0.0	95.0 ± 0.2
	CodeT5	96.7 ± 0.5	5.6 ± 0.5	96.0 ± 0.5	96.0 ± 0.2	**97.3 ± 1.1**	**6.2 ± 1.1**	95.7 ± 0.2	**96.3 ± 0.2**
SIDER	GPT-2	61.3 ± 1.2	6.2 ± 1.2	55.7 ± 1.2	62.0 ± 2.5	**77.7 ± 3.8**	**22.6 ± 3.8**	55.7 ± 0.3	**65.7 ± 0.3**
	Mistral	98.3 ± 0.8	43.2 ± 0.8	65.0 ± 3.5	66.0 ± 0.2	96.7 ± 1.3	41.6 ± 1.3	64.7 ± 3.6	63.3 ± 1.5
	CodeT5	71.3 ± 4.3	16.2 ± 4.3	60.7 ± 2.8	60.7 ± 1.0	**76.7 ± 5.9**	**21.6 ± 5.9**	**62.3 ± 1.3**	**61.7 ± 1.2**

they fail to develop an intrinsic understanding of the chemical equivalence between different molecular representations.

3.2 Finetuning LLMs with mapped SMILES & IUPAC data

The state-of-the-art LLMs discussed earlier are not trained on one-to-one mapped data, which may favor either IUPAC or SMILES representation. To mitigate bias, we performed finetuning using a one-to-one mapped dataset of SMILES and IUPAC representations, ensuring that the representation format was the only variable.

We evaluated three architectures – GPT-2, Mistral 7B,[24] and CodeT5 small[25] – on three tasks: forward reaction prediction, retrosynthesis, and property prediction. For GPT-2, we further varied the model size (small, medium (M), large (L), and extra-

Table 2 Consistency (raw and adjusted) and accuracy of LLMs in numeric property prediction after finetuning (columns 3–6) and with KL divergence loss (columns 7–10). Entries that improve with the addition of KL divergence loss are highlighted in bold. Error bars denote the standard deviation across training runs with varying random seeds. A downward arrow (↓) indicates that lower values correspond to better performance, and an upward arrow (↑) indicates that higher values correspond to better performance

		Performance (MSE)				Performance w/KL (MSE)			
Properties	Models	Consist↓	Adj. consist↑	Acc. (S)↓	Acc. (I)↓	Consist↓	Adj. consist↑	Acc. (S)↓	Acc. (I)↓
ESOL	GPT-2	4.3 ± 0.5	5.1 ± 0.5	1.5 ± 0.1	3.3 ± 0.6	**2.7 ± 0.3**	**6.7 ± 0.3**	1.6 ± 0.3	**3.1 ± 0.1**
	Mistral	4.9 ± 0.5	4.5 ± 0.5	1.7 ± 0.8	4.5 ± 0.6	**2.1 ± 0.2**	**7.3 ± 0.2**	**1.3 ± 0.3**	**2.9 ± 0.4**
	CodeT5	5.9 ± 0.5	3.5 ± 0.5	0.9 ± 0.2	5.4 ± 0.4	**3.1 ± 0.7**	**6.3 ± 0.7**	1.8 ± 0.3	**3.6 ± 0.2**
LIPO	GPT-2	1.1 ± 0.1	1.5 ± 0.1	1.2 ± 0.0	1.2 ± 0.0	**0.7 ± 0.0**	**1.9 ± 0.0**	**1.0 ± 0.1**	**1.0 ± 0.0**
	Mistral	0.9 ± 0.2	1.7 ± 0.2	1.5 ± 0.2	1.2 ± 0.0	**0.5 ± 0.1**	**2.1 ± 0.1**	**1.2 ± 0.0**	**1.1 ± 0.0**
	CodeT5	1.0 ± 0.2	1.6 ± 0.2	1.0 ± 0.0	0.9 ± 0.1	1.0 ± 0.0	1.6 ± 0.0	1.1 ± 0.0	1.0 ± 0.1

large (XL)) to examine the impact of scaling. Additionally, we compared performance using two training data sizes: 80k and 1M examples. To assess the effects of pretraining, we also trained a GPT-2 model from random initialization.

We evaluated performance using two metrics: consistency and accuracy. We used both overall and false consistency (cases where SMILES and IUPAC inputs produce the same incorrect predictions), which is critical for disentangling consistency from accuracy. Accuracy was measured separately for SMILES and IUPAC inputs. The results are presented in Fig. 3a, b, Tables 1 and 2. To provide context for our results, we compare the performance of our models with state-of-the-art LLMs (Table 5). Our finetuned GPT-2 model achieves accuracy comparable to existing benchmarks.

3.2.1 Impact of model architectures. For forward reaction prediction and retrosynthesis tasks, CodeT5 consistently outperformed Mistral and GPT-2. Its encoder–decoder architecture likely contributes to this by constructing a structured latent representation of the input, enabling better transformation into the output space.[25] In contrast, the decoder-only architectures of GPT-2 and Mistral, designed for autoregressive generation, may be less suited for structured prediction tasks. Additionally, CodeT5's Unicode-based tokenizer may better preserve meaningful substrings in symbolic domains like SMILES or IUPAC, compared to the byte-level tokenizers used by GPT-2 and Mistral.

For property prediction, however, the results vary across models and tasks. The mixed results indicate that while certain architectures, such as the encoder–decoder framework of CodeT5, may excel at capturing structural patterns, decoder-only models, such as GPT-2 and Mistral, may generalize better for less complex tasks.[26]

3.2.2 Impact of model size. Scaling up the GPT-2 model from small to XL showed no significant improvements in consistency or accuracy, suggesting that simply increasing model size does not improve performance or generalization ability.

3.2.3 Impact of data size. For GPT-2, increasing the training dataset size from 80k to 1M led to substantial improvements in both consistency and accuracy for forward reaction prediction and retrosynthesis. The increase in overall consistency aligns with the improvement in accuracy, indicating that the larger dataset enhances the model's ability to make correct predictions for both SMILES and IUPAC inputs. However, the gap between overall consistency and false consistency widened, suggesting that the additional data results in limited improvement in false consistency.

3.2.4 Effects of pretraining. Models trained from randomly initialized weights showed a slight decrease in consistency and accuracy compared to their pretrained counterparts (Fig. 6, Tables 6 and 7). This suggests that pretraining data contains useful chemistry-related information, which contributes to the model's performance.

3.3 Adding sequence-level KL divergence loss

In this section, we examined the impact of adding sequence-level KL divergence loss during training on three models: GPT-2, Mistral 7B, and CodeT5, for forward reaction prediction, retrosynthesis, and property prediction. The results are summarized in Fig. 3c, d, 6, Tables 1, 2, 6 and 7.

3.3.1 Consistency improvements. Adding KL divergence loss led to notable improvements in consistency across all models and tasks, including the randomly initialized GPT-2 model. For forward reaction prediction and retrosynthesis, false consistency increased, and the gap between overall and false consistency narrowed, contrasting with the trends observed with increasing dataset size. These results confirm that KL divergence loss enhances consistency by aligning predictions across input representations.

3.3.2 Accuracy unchanged. Despite improvements in consistency, accuracy remained largely unchanged across models and tasks. This suggests that gains in consistency do not compromise accuracy but also highlights the orthogonality of these two metrics: improving one does not inherently lead to improvement in the other.

3.4 SMILES ↔ IUPAC translation

We used SMILES ↔ IUPAC translation as an evaluation tool and a pretraining strategy to study whether models develop internal mappings across representations.

3.4.1 Translation for evaluation. We evaluated the translation ability of o3-mini and GPT-2. The accuracy of o3-mini is near random chance (0.3%), suggesting no learned alignment between representations. GPT-2 finetuned on forward reaction prediction achieves low translation accuracy (2–8%). KL

divergence regularization improves translation accuracy to 4–15%, indicating that KL helps enforce cross-representation alignment.

3.4.2 Translation for pretraining. We pretrained a GPT-2 model on SMILES ↔ IUPAC translation with an accuracy of 45.3% for IUPAC → SMILES and 12.7% for SMILES → IUPAC. The pretraining improves consistency of forward reaction prediction from 14.7% to 23.0%. The consistency gains diminish when KL regularization is applied. However, the translation pretraining does not improve the accuracy of forward reaction prediction (Fig. 7).

The results show that both KL regularization and translation pretraining enhance surface-level consistency across representations, but do not improve the model's intrinsic chemical reasoning.

4 Analysis

4.1 Consistency transition with KL divergence loss

To explore how KL divergence loss improves consistency, we analyzed forward reaction prediction as a representative task, focusing on reactions with consistency transitions. Out of 300 reactions in the test set, 46 reactions transitioned from inconsistent to consistent predictions after adding KL divergence loss. These reactions were categorized into five groups (Fig. 4, Scheme 1, and Appendix Schemes 2–10):

(1) Complicated reactions: we group reactions that require a good understanding of chemistry and substantial manipulation of symbolic representations as "complicated reactions". For instance, hydroquinone oxidation by cerium(IV) ammonium nitrate requires recognizing the hydroquinone structure and the oxidant. In addition, the product's SMILES string differs from the reactant's SMILES string in multiple positions (Scheme 1, entry 1). More than half of the reactions (24/46) fall into this category.

These reactions span five types: redox, coupling, cyclization, addition, and condensation. The distribution is shown in Fig. 4. Additional examples are listed in Schemes 1–6.

(2) Position inconsistency: the second-largest group consists of reactions whose predicted products are inconsistent in reaction sites or the positions of functional groups between SMILES and IUPAC inputs (Schemes 1 and 7).

(3) Reaction type inconsistency: SMILES and IUPAC inputs lead to predicted products from different reaction types (Schemes 1 and 8).

(4) Reaction step inconsistency: SMILES and IUPAC inputs result in predicted products involving different numbers of reaction steps (Schemes 1 and 9).

(5) Minor inconsistency: reactions with minor errors in either SMILES or IUPAC representations, such as mislabeling a nitrogen atom as carbon (Schemes 1 and 10).

The reverse transition – from consistent to inconsistent predictions – follows a similar pattern. Out of 300 reactions, 6 reactions became inconsistent with KL divergence loss: three complicated reactions and three position inconsistencies (Schemes 11 and 12).

For complicated reactions, models often make inconsistent and incorrect predictions without KL divergence loss. With KL divergence loss, the predictions become consistent but still incorrect. In contrast, for reactions where the model makes correct predictions in one representation but minor mistakes in the other, KL divergence loss aligns predictions and enables correct outputs for both representations.

The results suggest that KL divergence loss effectively addresses surface-level inconsistencies, but it falls short of achieving both accuracy and consistency. Advanced techniques will be required to capture the deeper intrinsic chemistry and achieve the ultimate goal of accurate and consistent predictions across representations.

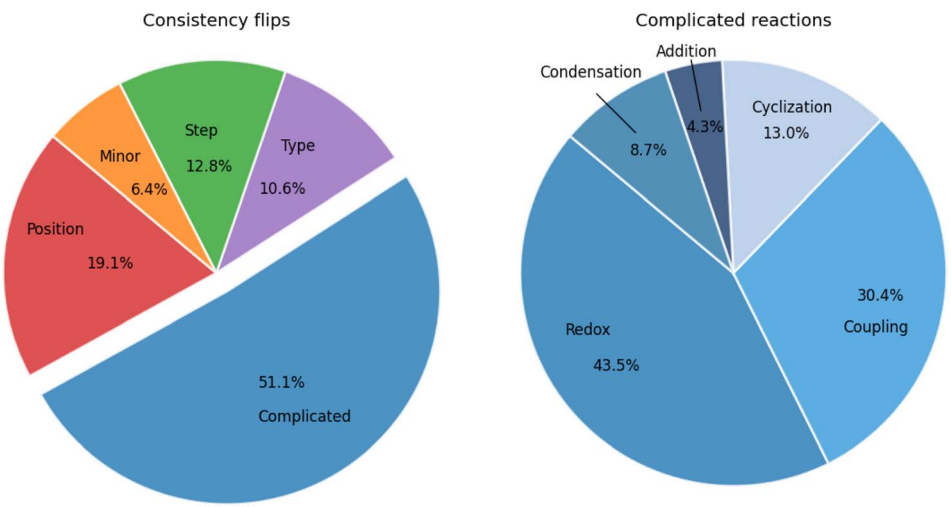
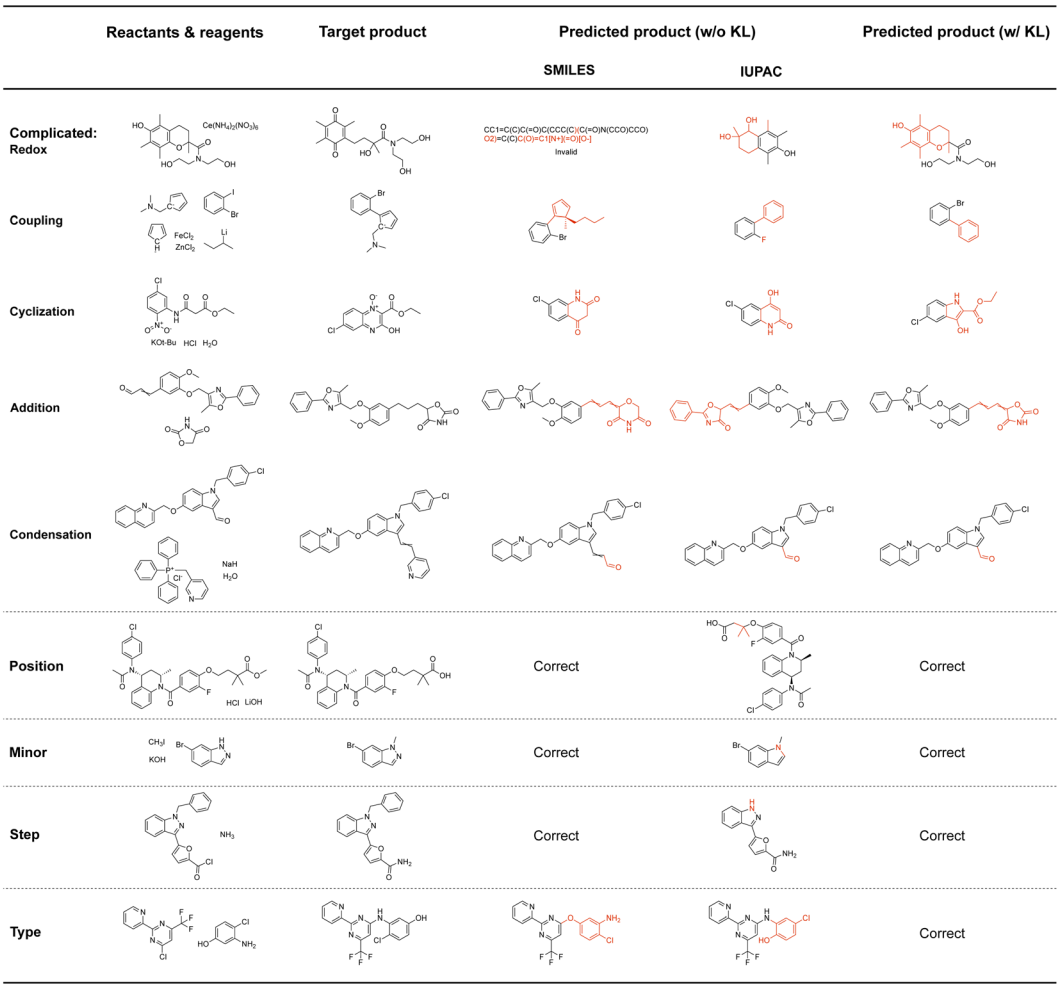


Fig. 4 Summary of reactions that transition from inconsistent without KL divergence loss to consistent with KL divergence loss. (Left) Reactions are categorized into five groups: complicated reactions, position inconsistencies, minor mistakes, reaction-step inconsistencies, and reaction-type inconsistencies. (Right) Complicated reactions are further subdivided into six types: redox reactions, coupling reactions, cyclization reactions, addition reactions, and condensation reactions.



Scheme 1 Examples of reactions transitioning from inconsistent to consistent predictions after adding KL divergence loss. Incorrect fragments are highlighted in red. For correct predictions, only the label "correct" is written without drawing the chemical structure.

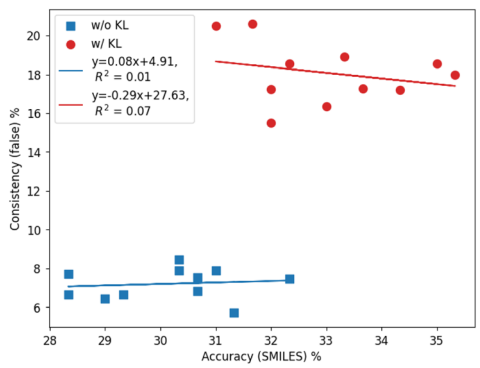


Fig. 5 Consistency (false) *versus* accuracy of the GPT-2 model in forward reaction prediction, without KL divergence loss (blue) and with KL divergence loss (red) across different random seeds in training. A linear fit of the data demonstrates minimal correlation between consistency and accuracy.

4.2 Orthogonality between consistency and accuracy

To explicitly analyze the relationship between consistency and accuracy, we studied the forward reaction prediction using GPT-2 small models with various random seeds. We used false consistency instead of overall consistency to exclude cases where both representations produce correct predictions to provide a clear measure of consistency.

We plotted consistency *versus* accuracy for models finetuned with and without KL divergence loss (Fig. 5). In both cases, there was minimal correlation between false consistency and accuracy, suggesting their orthogonality. Linear regression of the data yielded slopes of −0.29 and 0.08 for the results with and without KL divergence loss, respectively, further demonstrating that improvements in accuracy do not directly lead to better consistency. These findings highlight the need for strategies that enhance both metrics independently.

5 Conclusion

This work explores whether LLMs truly understand the intrinsic chemistry of molecules. We evaluated the consistency of LLMs across chemistry tasks using different molecular representations, including SMILES strings and IUPAC names. Our findings reveal that LLMs exhibit low consistency between the

representations, even when trained on carefully curated one-to-one mapped data. Incorporating sequence-level KL divergence loss improved surface-level consistency by aligning predictions, but did not enable the models to capture or exploit deeper intrinsic chemical properties. Further analysis suggested orthogonality between consistency and accuracy, suggesting that improvements in one do not inherently lead to enhancements in the other.

These findings underscore the limitations of current LLM architectures and the pressing need for more advanced models capable of scientific understanding and reasoning. In particular, we find it necessary for such an advanced model to readily incorporate prior knowledge of target domains, such as chemistry in this case, similarly to graph neural networks and other geometric deep learning approaches.[27] Such advances are crucial for achieving both accurate and consistent predictions in chemistry tasks.

Author contributions

B. Y., A. C. and K. C. designed the experiments, wrote the paper, and interpreted the results. B. Y. ran the experiments, performed analysis and illustrated the results. K. C. supervised the project.

Conflicts of interest

There are no conflicts to declare.

Data availability

The code is available at **https://github.com/bingyan4science/consistency**. The data are available at **https://doi.org/10.5281/zenodo.14430369**. The finetuned GPT-2 models for forward reaction prediction, with and without KL divergence loss, are available on the Hugging Face Hub at **https://huggingface.co/bing-yan/consistency**.

A Appendices

A.1 Formal definitions of evaluation metrics

Consistency measures how often the model generates identical outputs when provided with different molecular representations as input.

(1) Forward reaction prediction and retrosynthesis: for a given input format, the model is tested to generate outputs in either SMILES and IUPAC representations. For SMILES input (x_S), the model generates SMILES ($\hat{y}_S^{x_S}$) or IUPAC outputs ($\hat{y}_I^{x_S}$); for IUPAC input (x_I), the model generates SMILES ($\hat{y}_S^{x_I}$) or IUPAC output (($\hat{y}_I^{x_I}$)).

The outputs from different input representations "match" if identical:

$$\begin{aligned} \mathrm{MATCH_S} &= 1[\hat{y}_S^{x_S} = \hat{y}_S^{x_I}] \\ \mathrm{MATCH_I} &= 1[\hat{y}_I^{x_S} = \hat{y}_I^{x_I}] \end{aligned} \tag{2}$$

$1[\cdot]$ is the indicator function which returns 1 if the condition inside is true and 0 otherwise. The consistency score for a single entry is the average of SMILES and IUPAC matches. For a dataset of N entries, the overall consistency is calculated as:

$$\begin{aligned} \mathrm{Consist(overall)} &= \frac{1}{2N}\sum_{i=1}^{N}(\mathrm{MATCH_{S,}}_i + \mathrm{MATCH_{I,}}_i) \\ &= \frac{1}{2N}\sum_{i=1}^{N}\left(1\left[\hat{y}_{S,i}^{x_S} = \hat{y}_{S,i}^{x_I}\right] + 1\left[\hat{y}_{I,i}^{x_S} = \hat{y}_{I,i}^{x_I}\right]\right) \end{aligned} \tag{3}$$

We also compute the false consistency, defined as the consistency of entries that produce incorrect predictions from both SMILES and IUPAC inputs. For M entries:

$$\mathrm{Consist(false)} = \frac{1}{2M}\sum_{i=1}^{M}\left(1\left[\hat{y}_{S,i}^{x_S} = \hat{y}_{S,i}^{x_I}\right] + 1\left[\hat{y}_{I,i}^{x_S} = \hat{y}_{I,i}^{x_I}\right]\right) \tag{4}$$

where $\hat{y}_{S,i}^{x_S} \neq y_{S,i}, \hat{y}_{S,i}^{x_I} \neq y_{S,i}, \hat{y}_{I,i}^{x_S} \neq y_{I,i}, \hat{y}_{I,i}^{x_I} \neq y_{I,i}$, and $y_{S,i}, y_{I,i}$ are target outputs.

We compute adjusted consistency to measure consistency beyond chance. Let $p(y)$ be the empirical label distribution. Then the expected chance-level consistency is:

$$\mathrm{Consist(rand)} = \sum_y p(y)^2 \tag{5}$$

The adjusted consistency is then:

$$\mathrm{Consist(adj)} = \mathrm{consist(overall)} - \mathrm{consist(rand)} \tag{6}$$

(2) Binary property prediction: the predictions are denoted as $\hat{y}^{x_S}$ and $\hat{y}^{x_I}$ for SMILES and IUPAC inputs, respectively. The consistency for a dataset with N entries is:

$$\mathrm{Consist(binary)} = \frac{1}{N}\sum_{i=1}^{N}(1[\hat{y}_i^{x_S} = \hat{y}_i^{x_I}]) \tag{7}$$

The expected random agreement baseline is:

$$\mathrm{Consist(rand)} = p(0)^2 + p(1)^2 \tag{8}$$

where $p(0)$ and $p(1)$ are the empirical probabilities of predicting 0 or 1. The adjusted consistency is:

$$\mathrm{Consist(adj)} = \mathrm{consist(binary)} - \mathrm{consist(rand)} \tag{9}$$

(3) Numeric property prediction: consistency is measured as the mean squared error (MSE) between the predictions from SMILES and IUPAC inputs:

$$\mathrm{Consist(numeric)} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i^{x_S} - \hat{y}_i^{x_I})^2 \tag{10}$$

We define the random consistency baseline as:

$$\text{Consist(rand)} = 2 \cdot \text{Var}(\hat{y}) \tag{11}$$

where $\hat{y}$ denotes the set of all predictions from both input representations. The adjusted consistency is the improvement over this random baseline:

$$\text{Consist(adj)} = \text{consist(rand)} - \text{consist(numeric)} \tag{12}$$

A.1.1 Accuracy. Accuracy evaluates how closely the model's predictions align with the ground truth.

(1) Forward reaction prediction and retrosynthesis: For SMILES input, accuracy is calculated as the percentage of exact matches between the predicted SMILES output ($\hat{y}_S^{x_S}$) and the target SMILES output (y_S); for IUPAC input, accuracy is calculated between the predicted IUPAC output ($\hat{y}_I^{x_I}$) and the target IUPAC output (y_I).

$$\begin{aligned}\text{Accuracy(SMILES)} &= \frac{1}{N}\sum_{i}^{N}\left(1\left[\hat{y}_{S,i}^{x_S} = y_{S,i}\right]\right)\\ \text{Accuracy(IUPAC)} &= \frac{1}{N}\sum_{i}^{N}\left(1\left[\hat{y}_{I,i}^{x_I} = y_{I,i}\right]\right)\end{aligned} \tag{13}$$

(2) Binary property prediction: accuracy is calculated as the percentage of predictions same to the ground-truth y.

$$\begin{aligned}\text{Accuracy(SMILES)} &= \frac{1}{N}\sum_{i}^{N}\left(1[\hat{y}_i^{x_S} = y_i]\right)\\ \text{Accuracy(IUPAC)} &= \frac{1}{N}\sum_{i}^{N}\left(1[\hat{y}_i^{x_I} = y_i]\right)\end{aligned} \tag{14}$$

(3) Numeric property prediction: accuracy is measured as the MSE between the predicted outputs and the ground truth values.

$$\begin{aligned}\text{Accuracy(SMILES)} &= \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i^{x_S} - y_i\right)^2\\ \text{Accuracy(IUPAC)} &= \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i^{x_I} - y_i\right)^2\end{aligned} \tag{15}$$

B.1 Implementation details

B.1.1 Software and hardware. In this work, we use Python 3.10. The major Python packages we used are Transformers 4.43.4, PyTorch 2.1.0, RDKit 2023.3.3.

We train models using Nvidia A100 or H100 GPUs. We use one GPU for GPT-2 small, GPT-2 medium, GPT-2 large, and CodeT5 small models, and two GPUs for GPT-2 XL and Mistral 7B models.

B.1.2. Hyperparameters. We train all models using the AdamW optimizer.[28,29] We use random seeds of 42, 123, 999, 1234, 2024, 2718, 4321, 5678, 8080, 31 415, and 98 765. The other hyperparameters for each model are summarized in Table 3.

Table 3 Hyperparameters used to finetune LLMs: learning rate (LR), batch size (BSZ), accumulation (Acc.), number of epochs, and training time on one H100 GPU

Model	LR	BSZ	Acc.	Epochs	Time (h)
GPT-2 small	1×10^{-4}	32	1	20	2.28
GPT-2 medium	1×10^{-4}	16	1	20	6.24
GPT-2 large	1×10^{-4}	8	1	20	15.57
GPT-2 XL	1×10^{-4}	8	2	20	24.91
CodeT5 small	1×10^{-4}	32	1	20	2.57
Mistral 7B	1×10^{-5}	8	2	10	25.25

B.1.3 Input and output examples. We provide examples of input and output sequences for finetuning and evaluation.

(1) Evaluation of state-of-the-art LLMs: we provide a simple instruction specifying the input and output representation in the inquiry. The molecules are separated by comma (".") For example:

Input in SMILES: "Based on the SMILES strings of reactants and reagents, predict the SMILES string of the product. Please output the product directly.

⟨SMILES⟩ COc1ccc2c(c1)C(=O)c1ccccc1CC2.[BH4-].[OH-].[Na+].CCO ⟨SMILES⟩"

Target output in SMILES: "COc1ccc2c(c1)C(O)c1ccccc1CC2"

Input in IUPAC: "Based on the IUPAC names of reactants and reagents, predict the IUPAC name of the product. Please output the product directly.

⟨IUPAC⟩ 5-methoxytricyclo[9.4.0.03,8]pentadeca-1(15),3(8),4,6,11,13-hexaen-2-one.boranuide.hydroxide.sodium(1+). ethanol ⟨IUPAC⟩"

Target output in IUPAC:

"5-methoxytricyclo[9.4.0.03,8]pentadeca-1(15),3(8),4,6,11,13-hexaen-2-ol"

(2) Finetuning of LLMs: we append a flag at the end of the input sequence to specify the output representation, "S" for SMILES and "I" for IUPAC. For example:

Input in SMILES expecting output in SMILES:

"COc1ccc2c(c1)C(=O)c1ccccc1CC2.[BH4-].[OH-].[Na+].CCO.**S**"

Target in SMILES: "COc1ccc2c(c1)C(O)c1ccccc1CC2"

Input in SMILES expecting output in IUPAC:

"COc1ccc2c(c1)C(=O)c1ccccc1CC2.[BH4-].[OH-].[Na+].CCO.**I**"

Target in IUPAC: "5-methoxytricyclo[9.4.0.03,8]pentadeca-1(15),3(8),4,6,11,13-hexaen-2-ol"

C.1 KL divergence loss

Here we show the loss function for the sequence-level KL divergence: $D_{KL}(P\|Q)$ and $D_{KL}(Q\|P)$. We use $D_{KL}(P\|Q)$ as an example to demonstrate the calculation.

The gradient of $D_{KL}(P\|Q)$ is (we simplify $P_\theta(y|x_S)$ as $P_\theta(y)$, and $Q_\theta(y|x_I)$ as $Q_\theta(y)$):

Table 4 Statistics of the datasets used to finetune LLMs

Task	#Train	#Valid	#Test
Forward prediction (full)	963 567	1956	300
Forward prediction (subset)	76 379	1956	300
Retrosynthesis (full)	932 616	2004	300
Retrosynthesis (subset)	76 471	2004	300
Property – BBBP	1521	188	189
Property – ClinTox	1063	127	131
Property – HIV	32 864	4104	300
Property – SIDER	21 800	2540	300
Property – ESOL	888	111	112
Property – LIPO	3358	385	300
SMILES ↔ IUPAC	274 053	1397	300

$$\nabla_\theta D_{\mathrm{KL}}(P \parallel Q) = \sum_{y\in Y}\nabla_\theta\left(P_\theta(y)\log\frac{P_\theta(y)}{Q_\theta(y)}\right) = \sum_{y\in Y}\nabla_\theta(P_\theta(y))\log\frac{P_\theta(y)}{Q_\theta(y)} + P_\theta(y)\nabla_\theta\left(\frac{P_\theta(y)}{Q_\theta(y)}\right) \tag{16}$$

Using the trick $\nabla_\theta(P_\theta(y)) = P_\theta(y)\nabla_\theta(\log(P_\theta(y)))$:

$$\nabla_\theta D_{\mathrm{KL}}(P \parallel Q) = \sum_{y\in Y}P_\theta(y)\nabla_\theta(\log(P_\theta(y)))\log\frac{P_\theta(y)}{Q_\theta(y)} + P_\theta(y)\nabla_\theta\left(\frac{P_\theta(y)}{Q_\theta(y)}\right) = \mathbb{E}_{y\sim P_\theta(y)}\left[\nabla_\theta(\log P_\theta(y))\log\frac{P_\theta(y)}{Q_\theta(y)} + \nabla_\theta\left(\log\frac{P_\theta(y)}{Q_\theta(y)}\right)\right] \tag{17}$$

Therefore, we can define the KL loss corresponding to the KL divergence $D_{\mathrm{KL}}(P\|Q)$:

$$\mathrm{KL\ loss} \equiv \mathbb{E}_{y\sim P_\theta(y)}\left[\log P_\theta(y)\log\frac{P_\theta(y)}{Q_\theta(y)}\cdot\mathrm{detach} + \log\frac{P_\theta(y)}{Q_\theta(y)}\right] \tag{18}$$

However, the expectation is untractable, so we use a Monte Carlo to estimate it by sampling M sequences $\{y^1, \ldots, y^m\}$ from $P_\theta(y)$ and pass them through the models $P_\theta(y)$ and $Q_\theta(y)$:

$$\mathrm{KL\ loss}(PQ) \approx \frac{1}{M}\sum_{m=1}^{M}\left[\log P_\theta(y^m)\log\frac{P_\theta(y^m)}{Q_\theta(y^m)}\cdot\mathrm{detach} + \log\frac{P_\theta(y^m)}{Q_\theta(y^m)}\right] \tag{19}$$

Similarly, we can calculate the loss for the KL divergence of $Q_\theta(y)$ from $P_\theta(y)$ ($D_{\mathrm{KL}}(Q\|P)$) and the Monte Carlo estimation by sampling N sequences $\{y^1, \ldots, y^n\}$ from $Q_\theta(y)$:

$$\mathrm{KL\ loss}(QP) \equiv \mathbb{E}_{y\sim Q_\theta(y)}\left[\log Q_\theta(y)\log\frac{Q_\theta(y)}{P_\theta(y)}\cdot\mathrm{detach} + \log\frac{Q_\theta(y)}{P_\theta(y)}\right] \approx \frac{1}{N}\sum_{n=1}^{N}\left[\log Q_\theta(y^n)\log\frac{Q_\theta(y^n)}{P_\theta(y^n)}\cdot\mathrm{detach} + \log\frac{Q_\theta(y^n)}{P_\theta(y^n)}\right] \tag{20}$$

During training, we added a weight to the KL divergence loss. We screened values ranging from 0.001 to 10.0 and found that a weight of 1.0 gave the best consistency for all tasks and models.

D.1 Dataset

Here we list the statistics of the datasets used in this work in Table 4. There are three finetuning tasks: forward reaction prediction, retrosynthesis, and property prediction. These datasets are all one-to-one mapped between SMILES and IUPAC inputs. Furthermore, we have included the SMILES ↔ IUPAC translation dataset to evaluate and pretrain the LLMs.

E.1 Comparison with existing models

To contextualize our results, we present a comparison with state-of-the-art LLMs on chemistry tasks (Table 5). The table includes performance from our GPT-2 Small model finetuned on the full datasets, the best-performing model ($\mathrm{LlaSMol_{Mistral}}$), and the average performance of the top four models. Full results can be found in.[17] We use the accuracy of SMILES inputs for our GPT-2 model as used in the benchmarks.

Table 5 Comparison of our results to state-of-the-art LLMs on chemistry tasks. We report the performance of the finetuned GPT-2 small model and the best-performing model, $\mathrm{LlaSMol_{Mistral}}$. Additionally, we provide the average performance of the top four models for a broader comparison. Complete results are available in ref. 17

	Accuracy (%↑ or RMSE↓)		
Task	Ours (GPT-2)	Best ($\mathrm{LlaSMol_{Mistral}}$)	Top 4 models averaged
Forward reaction prediction (%)	57.7	63.3	53.9
Retrosynthesis (%)	29.7	32.9	26.7
Property – BBBP (%)	86.2	74.6	70.4
Property – ClinTox (%)	93.1	93.1	92.9
Property – HIV (%)	96.3	96.7	96.7
Property – Sider (%)	55.7	70.7	69.9
Property – ESOL (RMSE)	1.150	1.036	2.215
Property – LIPO (RMSE)	0.995	1.010	1.191

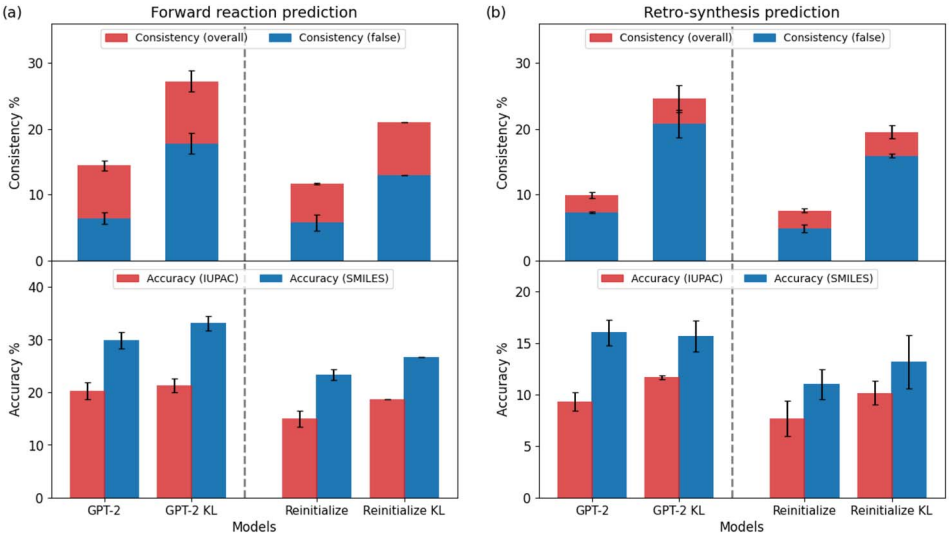


Fig. 6 Consistency and accuracy of pretrained *vs.* reinitialized GPT-2 in (a) forward reaction prediction and (b) retrosynthesis prediction with the addition of KL divergence loss. Overall consistency (red) and false consistency (blue) are overlaid. All models are finetuned on an 80k dataset subset. Error bars represent the standard deviation across training runs with varying random seeds.

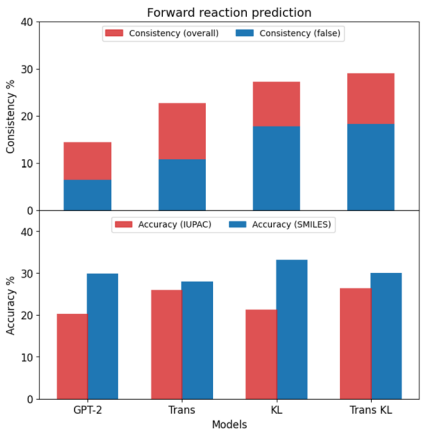


Fig. 7 Consistency and accuracy of GPT-2 in forward reaction prediction. All models are finetuned on an 80k dataset subset. "Trans" denotes a pretraining on SMILES ↔ IUPAC. "KL" refers to the addition of KL divergence loss during finetuning.

F.1 Reinitialized model

We trained a randomly initialized GPT-2 model using the same finetuning setup as its pretrained counterpart. This allows us to isolate the contribution of pretraining data. The results are presented in Fig. 6, Tables 6 and 7.

G.1 Consistency transition

Here we list all of the reactions that transit either from inconsistent to consistent predictions, or from consistent to inconsistent predictions.

G.1.1 Consistent-to-inconsistent transitions. Here we list 46 reactions that transition from inconsistent to consistent predictions between SMILES and IUPAC inputs after adding KL divergence loss in Schemes 2–10.

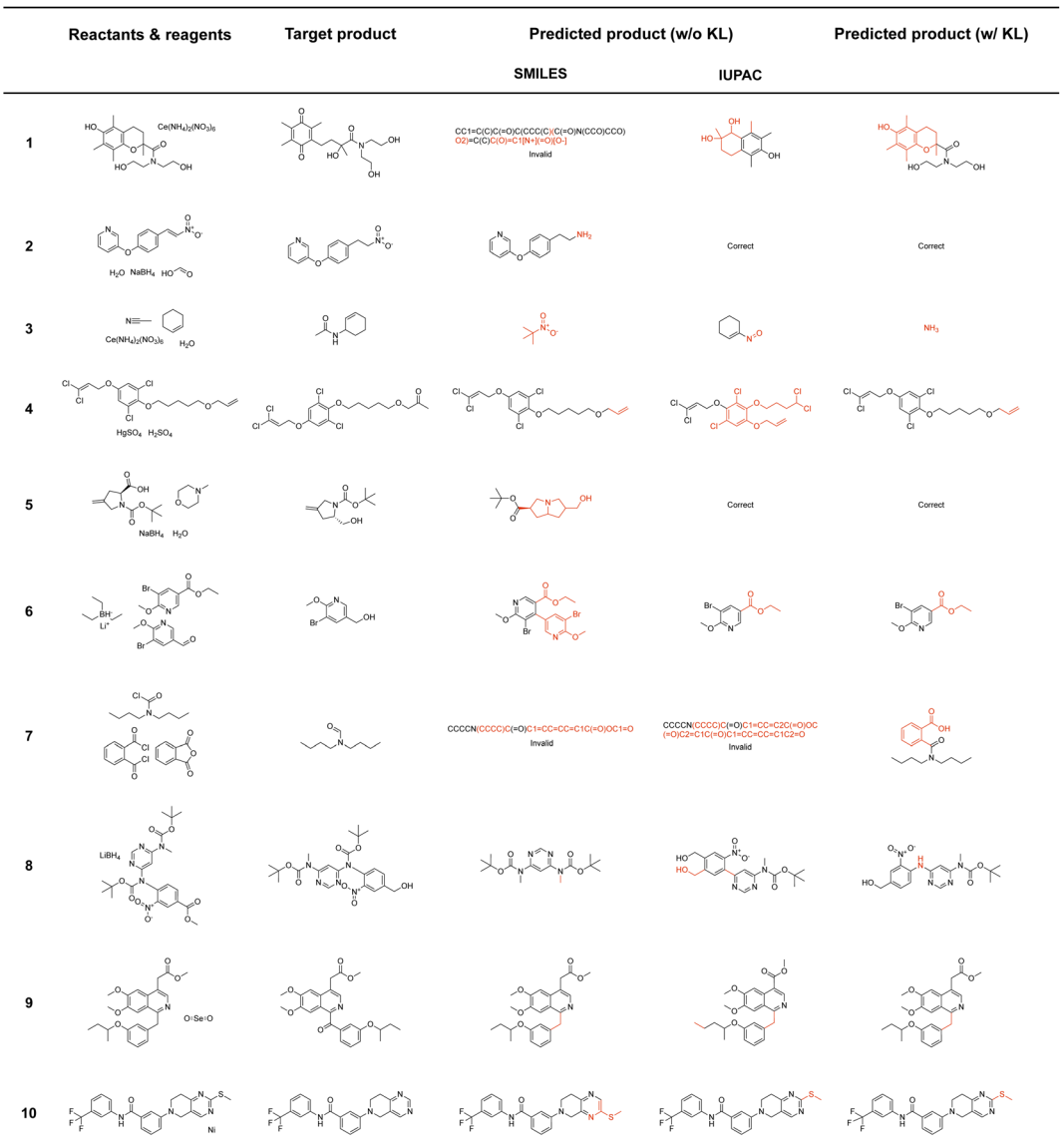
G.1.2 Inconsistent-to-consistent transitions. Here we list 6 reactions that transition from consistent to inconsistent predictions between SMILES and IUPAC inputs after adding KL divergence loss in Schemes 11 and 12.

Table 6 Consistency (raw and adjusted) and accuracy of reinitialized GPT-2 in binary property prediction after finetuning (columns 3–6) and with KL divergence loss (columns 7–10). Entries with improvements following the addition of KL divergence loss are highlighted in bold. Error bars represent the standard deviation across training runs with varying random seeds. An upward arrow (↑) indicates that higher values correspond to better performance

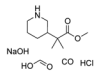
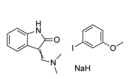
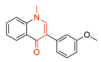
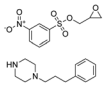
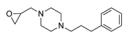
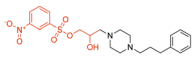
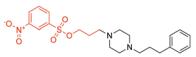
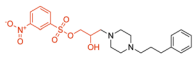
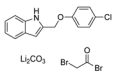
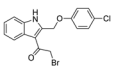
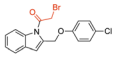
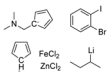
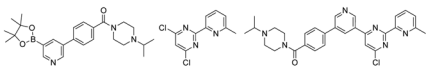
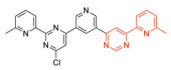
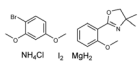
		Performance (%)↑				Performance w/KL (%)↑			
Properties	Models	Consist	Adj. consist	Acc. (S)	Acc. (I)	Consist	Adj. consist	Acc. (S)	Acc. (I)
BBBP	GPT-2	83.1 ± 0.8	26.4 ± 0.8	81.5 ± 0.6	78.9 ± 1.3	**92.1 ± 1.5**	**35.4 ± 1.5**	**82.5 ± 0.5**	**85.2 ± 1.4**
ClinTox	GPT-2	99.2 ± 2.2	13.3 ± 2.2	92.4 ± 0.2	93.2 ± 1.3	**100.0 ± 2.3**	**14.1 ± 2.3**	92.4 ± 0.9	92.4 ± 0.1
HIV	GPT-2	97.7 ± 0.8	6.6 ± 0.8	94.3 ± 0.3	95.7 ± 0.3	**99.3 ± 0.1**	**8.2 ± 0.1**	**94.7 ± 0.4**	95.3 ± 0.1
SIDER	GPT-2	77.3 ± 1.5	22.2 ± 1.5	64.3 ± 1.1	57.7 ± 1.9	**84.3 ± 3.2**	**29.2 ± 3.2**	**65.3 ± 0.5**	**62.0 ± 0.2**

Table 7 Consistency (raw and adjusted) and accuracy of reinitialized GPT-2 in numeric property prediction after finetuning (columns 3–6) and with KL divergence loss (columns 7–10). Entries with improvements after the addition of KL divergence loss are highlighted in bold. Error bars denote the standard deviation across training runs with varying random seeds. A downward arrow (↓) indicates that lower values correspond to better performance, and an upward arrow (↑) indicates that higher values correspond to better performance

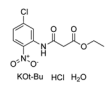
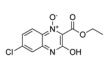
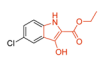
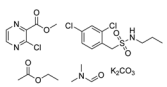
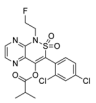
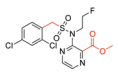
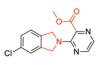
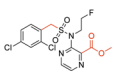
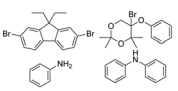
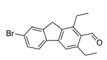
		Performance (MSE)				Performance w/KL (MSE)			
Properties	Model	Consist↓	Adj. consist↑	Acc. (S)↓	Acc. (I)↓	Consist↓	Adj. consist↑	Acc. (S)↓	Acc. (I)↓
ESOL	GPT-2	3.4 ± 0.1	6.0 ± 0.1	1.8 ± 0.1	2.8 ± 0.4	**2.9 ± 0.3**	**6.5 ± 0.3**	**1.1 ± 0.1**	3.6 ± 0.2
LIPO	GPT-2	1.6 ± 0.2	1.0 ± 0.2	1.3 ± 0.1	1.3 ± 0.1	**0.7 ± 0.0**	**1.9 ± 0.0**	1.4 ± 0.1	**1.1 ± 0.1**



Scheme 2 Complicated redox reactions that transition from inconsistent to consistent predictions after adding KL divergence loss.

	Reactants & reagents	Target product	Predicted product (w/o KL)		Predicted product (w/ KL)
			SMILES	IUPAC	
1				Correct	Correct
2				Correct	Correct
3					
4					
5					
6		Correct			Correct
7					

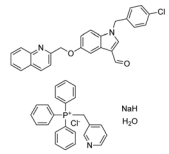
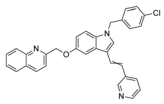
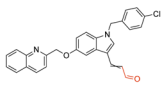
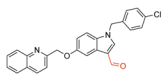
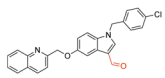
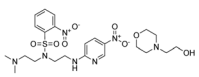
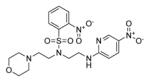
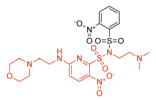
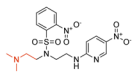
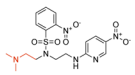
Scheme 3 Complicated coupling reactions that transition from inconsistent to consistent predictions after adding KL divergence loss.

	Reactants & reagents	Target product	Predicted product (w/o KL)		Predicted product (w/ KL)
			SMILES	IUPAC	
1					
2					
3					

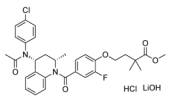
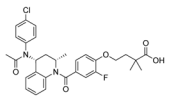
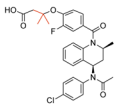
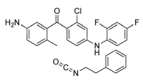
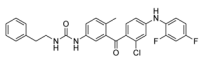
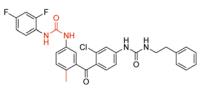
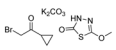
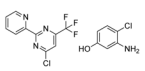
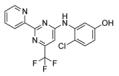
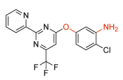
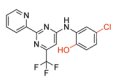
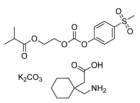
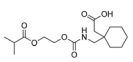
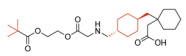
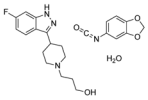
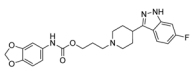
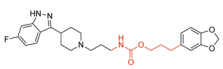
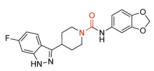
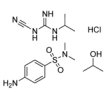
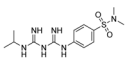
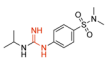
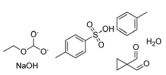
Scheme 4 Complicated cyclization reactions that transition from inconsistent to consistent predictions after adding KL divergence loss.

	Reactants & reagents	Target product	Predicted product (w/o KL)		Predicted product (w/ KL)
			SMILES	IUPAC	
1					

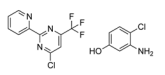
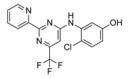
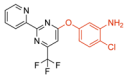
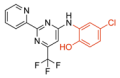
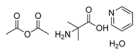
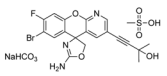
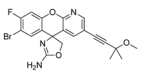
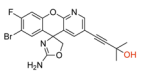
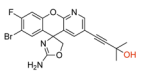
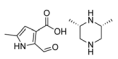
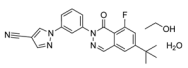
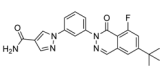
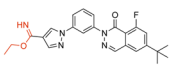
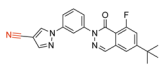
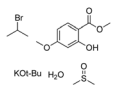
Scheme 5 Complicated addition reactions that transition from inconsistent to consistent predictions after adding KL divergence loss.

	Reactants & reagents	Target product	Predicted product (w/o KL)		Predicted product (w/ KL)
			SMILES	IUPAC	
1					
2					

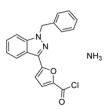
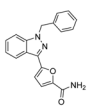
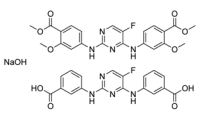
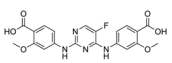
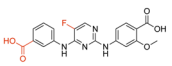
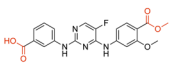
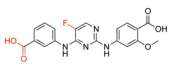
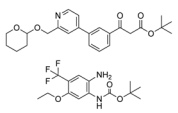
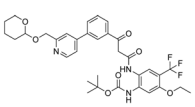
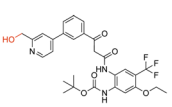
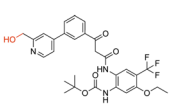
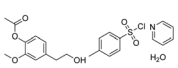
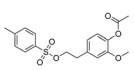
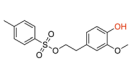
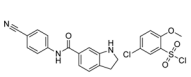
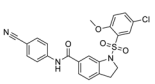
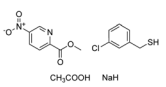
Scheme 6 Complicated condensation reactions that transition from inconsistent to consistent predictions after adding KL divergence loss.

	Reactants & reagents	Target product	Predicted product (w/o KL)		Predicted product (w/ KL)
			SMILES	IUPAC	
1			Correct		Correct
2			Correct		Correct
3			Correct		Correct
4					Correct
5			Correct		Correct
6			Correct		
7			Correct		
8			Correct		Correct

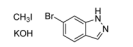
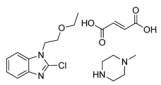
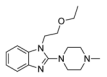
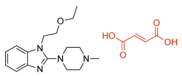
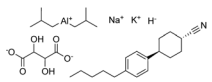
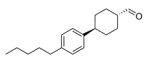
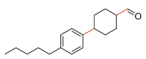
Scheme 7 Position-inconsistent reactions that transition from inconsistent to consistent predictions after adding KL divergence loss.

	Reactants & reagents	Target product	Predicted product (w/o KL)		Predicted product (w/ KL)
			SMILES	IUPAC	
1					Correct
2			Correct		Correct
3					
4			Correct		Correct
5					Correct
6				Correct	Correct

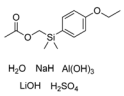
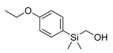
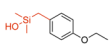
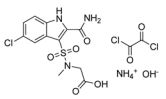
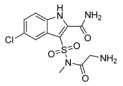
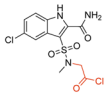
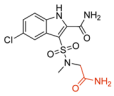
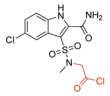
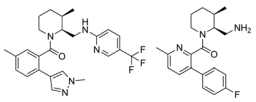
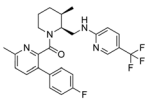
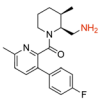
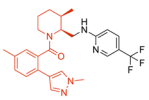
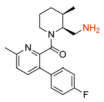
Scheme 8 Reaction type-inconsistent reactions that transition from inconsistent to consistent predictions after adding KL divergence loss.

	Reactants & reagents	Target product	Predicted product (w/o KL)		Predicted product (w/ KL)
			SMILES	IUPAC	
1			Correct		Correct
2					
3				Correct	
4				Correct	Correct
5			Correct	COC1=CC=C(Cl)C=C1S(=O)(=O)NC1=CC=C(C **Invalid**	Correct
6					

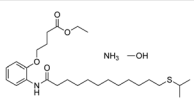
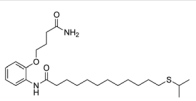
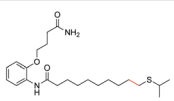
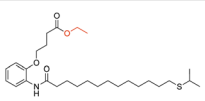
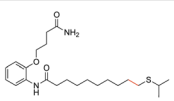
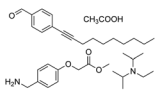
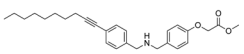
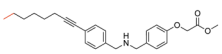
Scheme 9 Reaction step-inconsistent reactions that transition from inconsistent to consistent predictions after adding KL divergence loss.

	Reactants & reagents	Target product	Predicted product (w/o KL)		Predicted product (w/ KL)
			SMILES	IUPAC	
1			Correct		Correct
2			Correct		Correct
3			Correct		Correct

Scheme 10 Minor inconsistent reactions that transition from inconsistent to consistent predictions after adding KL divergence loss.

	Reactants & reagents	Target product	Predicted product (w/o KL)		Predicted product (w/ KL)
			SMILES	IUPAC	
1					
2					
3					

Scheme 11 Complicated reactions that transition from consistent to inconsistent predictions after adding KL divergence loss.

	Reactants & reagents	Target product	Predicted product (w/o KL)		Predicted product (w/ KL)
			SMILES	IUPAC	
1					
2	Correct		Correct		Correct
3		Correct		Correct		Correct

Scheme 12 Position inconsistent reactions that transition from consistent to inconsistent predictions after adding KL divergence loss.

Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) with a grant funded by the Ministry of Science and ICT (MSIT) of the Republic of Korea in connection with the Global AI Frontier Lab International Collaborative Research. This work was also supported by the Samsung Advanced Institute of Technology

(under the project Next Generation Deep Learning: From Pattern Recognition to AI) and the National Science Foundation (under NSF Award 1922658).

Notes and references

1 D. M. Anstine and O. Isayev, *J. Am. Chem. Soc.*, 2023, **145**, 8736–8750.
2 D. Tran, L. Pascazio, J. Akroyd, S. Mosbach and M. Kraft, *ACS Omega*, 2024, **9**, 13883–13896.
3 K. M. Jablonka, P. Schwaller, A. Ortega-Guerrero and B. Smit, *Nat. Mach. Intell.*, 2024, **6**, 161–169.
4 A. M. Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White and P. Schwaller, *Nat. Mach. Intell.*, 2024, 1–11.
5 P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. A. Hunter, C. Bekas and A. A. Lee, *ACS Cent. Sci.*, 2019, **5**, 1572–1583.
6 T. Guo, B. Nan, Z. Liang, Z. Guo, N. Chawla, O. Wiest, X. Zhang, *et al.*, *Adv. Neural Inf. Process. Syst.*, 2023, **36**, 59662–59688.
7 D. Weininger, *J. Chem. Inf. Comput. Sci.*, 1988, **28**, 31–36.
8 *Nomenclature of Organic Chemistry: IUPAC Recommendations and Preferred Names 2013*, ed. H. A. Favre and W. H. Powell, Royal Society of Chemistry, Cambridge, UK, 2014.
9 A. Chen, J. Phang, A. Parrish, V. Padmakumar, C. Zhao, S. Bowman and K. Cho, *Transactions on Machine Learning Research*, 2024.
10 R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann and W. Brendel, *International Conference on Learning Representations*.
11 J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, *arXiv*, 2023, preprint, arXiv:2303.08774, DOI: **10.48550/arXiv.2303.08774**.
12 A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, *et al.*, *arXiv*, 2024, preprint, arXiv:2410.21276, DOI: **10.48550/2410.21276**.
13 OpenAI, Introducing OpenAI o1-preview and o1-mini, 2024, **https://openai.com/index/introducing-openai-o1-preview**.
14 OpenAI, OpenAI o3-mini, 2025, **https://openai.com/index/openai-o3-mini**.
15 Anthropic, Introducing the Next Generation of Claude, 2024, **https://www.anthropic.com/news/claude-3-family**.
16 M AI, Introducing Meta Llama 3.1: The Most Capable Openly Available LLM to Date, 2024, **https://ai.meta.com/blog/meta-llama-3-1/**.
17 B. Yu, F. N. Baker, Z. Chen, X. Ning and H. Sun, *First Conference on Language Modeling*, 2024.
18 M. Swain, A simple Python wrapper around the PubChem PUG REST API, Version 1.0.4, **https://github.com/mcs07/PubChemPy**.
19 Knowledgator, Chemical-Converters is collection of tools for converting one chemical format into another, Version 0.1.1, **https://github.com/Knowledgator/chemical-converters?tab=readme-ov-file**.
20 D. M. Lowe, P. T. Corbett, P. Murray-Rust and R. C. Glen, *Chemical Name to Structure: OPSIN, an Open Source Solution*, 2011.
21 National Center for Biotechnology Information (NCBI), PubMed Central (PMC), 2024, **https://pmc.ncbi.nlm.nih.gov/**.
22 U. S. Patent and T. O. (USPTO), USPTO Patent Database, 2024, **https://ppubs.uspto.gov/pubwebapp/**.
23 W Foundation, Wikimedia Downloads, **https://dumps.wikimedia.org**.
24 A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, *et al.*, *arXiv*, 2023, preprint, arXiv:2310.06825, DOI: **10.48550/arXiv.2310.06825**.
25 Y. Wang, W. Wang, S. Joty and S. C. Hoi, *arXiv*, 2021, preprint, arXiv:2109.00859, DOI: **10.48550/arXiv.2109.00859**.
26 Y. Chen, G. Ou, M. Liu, Y. Wang and Z. Zheng, *arXiv*, 2024, preprint, arXiv:2410.22240, DOI: **10.48550/arXiv.2410.22240**.
27 M. M. Bronstein, J. Bruna, T. Cohen and P. Veličković, *arXiv*, 2021, preprint, arXiv:2104.13478, DOI: **10.48550/arXiv.2104.13478**.
28 D. P. Kingma and J. Ba, *arXiv*, 2014, preprint, arXiv:1412.6980, DOI: **10.48550/arXiv.1412.6980**.
29 I. Loshchilov and F. Hutter, *arXiv*, 2017 preprint, arXiv:1711.05101, DOI: **10.48550/arXiv.1711.05101**.