Teaching Language Models Mechanistic Explainability Through Arrow-Pushing

Théo A. Neukomm
LIAC - EPFL
Lausanne, Switzerland
theo.neukomm@epfl.ch

Zlatko Jončev
LIAC - EPFL
Lausanne, Switzerland

Philippe Schwaller
LIAC - EPFL
Lausanne, Switzerland
philippe.schwaller@epfl.ch

Abstract

Chemical reaction mechanisms provide crucial insight into synthesizability, yet current Computer-Assisted Synthesis Planning (CASP) systems lack mechanistic grounding. We introduce a computational framework for teaching language models to predict chemical reaction mechanisms through arrow pushing formalism, a century-old notation that tracks electron flow while respecting conservation laws. We developed MechSMILES, a compact textual format encoding molecular structure and electron flow, and trained language models on four mechanism prediction tasks of increasing complexity using mechanistic reaction datasets, such as mech-USPTO-31k and FlowER. Our models achieve more than 95% top-3 accuracy on elementary step prediction and scores that surpass 73% on mech-USPTO-31k, and 93% on FlowER dataset for the retrieval of complete reaction mechanisms on our hardest task. This mechanistic understanding enables three key applications. First, our models serve as post-hoc validators for CASP systems, filtering chemically implausible transformations. Second, they enable holistic atom-to-atom mapping that tracks all atoms, including hydrogens. Third, they extract catalyst-aware reaction templates that distinguish recycled catalysts from spectator species. By grounding predictions in physically meaningful electron moves that ensure conservation of mass and charge, this work provides a pathway toward more explainable and chemically valid computational synthesis planning, while providing an architecture-agnostic framework for the benchmarking of mechanism prediction.

1 Introduction

Recent advances in Computer-Assisted Synthesis Planning (CASP) have produced both commercial and open-source systems capable of proposing synthetic routes.[1–9]. These systems predominantly employ a retrosynthetic approach, starting from the target molecule and working backwards through careful sequences of reactions, until reaching a set of starting materials. However, most models suffer from two key limitations. First, they lack explainability, as their mechanistic reasoning behind proposed transformations remains opaque. Second, they often suggest reactions that are formally valid as graph transformations but chemically implausible due to high-energy intermediates or forbidden electron movements. While various approaches have attempted to address feasibility through scores[10–12], or via the crafting of elaborated expert-curated templates[1,13,14], these methods do not capture the underlying mechanistic logic that experimental chemists use to evaluate reactions. We address these limitations by teaching language models to reason about reactions through arrow pushing mechanisms, a century-old formalism that experimental chemists use to predict reaction feasibility. Arrow pushing abstracts complex quantum phenomena into intuitive electron movements, providing a physically grounded representation that enforces conservation of mass and charge. We developed a computational environment (available as a Python package) that encodes the rules of

2025 ELLIS ML4Molecules Workshop

the arrow pushing formalism and trained models on four mechanism prediction tasks of increasing complexity. Our most challenging task mirrors the problem human chemists face: predicting the complete mechanism given only the available reactant species and main product, without knowing stoichiometry or by-products.

To enable this approach, we developed MechSMILES, a compact and unambiguous textual format for representing mechanistic steps. MechSMILES concatenates a mapped standard Simplified Molecular Input Line Entry System (SMILES)[15] string with an addendum encoding electron movements through three arrow types: attacks, ionizations, and bond attacks (details in Section 2.1). This representation is both human-readable and suitable for training sequence-to-sequence models, bridging the gap between chemical notation and machine learning.

While mechanism prediction from reactions using arrow pushing has been addressed by Bradshaw et al.[16] in 2019, and more recently by Miller et al.[17] in 2025. Our work extends these works by overcoming some limitations of the former, and by standardizing the task and comparing the multiple datasets available in the literature, while open-sourcing our data and models for better reproducibility across the field.

Moreover, our work equally complements recent advances in mechanistic modeling by Chen et al.[18] and Joung et al.[19], who trained models to predict multi-step forward synthesis one elementary step at a time. While their approaches function as "reasoning traces" that generate mechanisms before products, our work operates as a "post-hoc rationalizer" that validates proposed reactions by reconstructing their mechanisms. This distinction is crucial for integration with existing CASP workflows: state-of-the-art systems propose single-step transformations from reactants to products without mechanistic decomposition. Our models can validate these proposals by attempting to find physically plausible mechanistic pathways, effectively serving as a chemical feasibility filter for any CASP system, whether template-based or template-free, commercial or open-source.

Beyond validation, mechanism prediction unlocks two additional capabilities that are difficult or impossible with current methods. First, by tracking electron flow, we enable holistic atom-to-atom mapping that includes all hydrogen atoms, critical for reactions where hydrogen transfer determines the outcome. Second, we can extract catalyst-aware reaction templates that distinguish catalysts recycled during the mechanism from inert spectator species, a distinction invisible to traditional template extraction methods that only compare initial and final states.

2 Methods

2.1 MechSMILES format

Previous works have tackled the encoding of chemical transformations in textual format. For example SMIRKS[20] is a format that was developed to encode transformations. In terms of amount of information, such a format is equivalent to a fully mapped reactant-product couple as is presented in Joung et al.[19]. However, both of these can be used to convey any arbitrary number of steps, and not mandatorily elementary steps. Works such as Tavakoli et al.[21] represent a middle ground, as the PMechDB dataset is using a format in which the reactant, the product, as well as an *arrow-code* are reported. These three pieces of information are redundant, as the reactant and arrows only are sufficient to infer the product of the elementary step. Finally, Chen et al.[22] shared their dataset mech-USPTO-31k using a format that is almost equivalent to the MechSMILES format. However, this format distinguishes arrow-types that we did not think deserved to be separated, and mech-USPTO-31k, although reporting the reactant side and all arrows, does not explicitly mention the stable intermediate structures that are reached in between the reactant and the product.

For all of these reasons, we developed MechSMILES, a compact, unambiguous and simple format for arrow pushing that is composed of only three types of arrows: (1) The attack, describing the attack of a lone pair of atom a on atom b, increasing their bond-degree by one, is written `(a, b)`; (2) the ionization, describing a bond losing one degree to ionize both of its ends, is written `((a, b), b)`, where a is positively ionized, and b is negatively ionized; (3) The bond attack, describing the attack of a bond (a, b) on a third atom c through one of its ends, is written `((a, b), c)` when attacking through b. This move decreases the degree of the bond (a, b) by one, and increases the degree of the bond (b, c) by one. To form a MechSMILES, one simply needs to concatenate a SMILES at least

Table 1: Summary of advantages of MechSMILES for the encoding of elementary steps.
flow: FlowER. m-us: mech-USPTO-31k, pmdb: PMechDB, msmi: MechSMILES (this work).
*Results reported for the training split of PMechDB split-0, a visual representation is visible on Figure S2.

	flow [19]	m-us [22]	pmdb [17]	msmi (this work)
Arrows reported	✗	✓	✓	✓
Reports stable intermediates	✓	✗	✓	✓
Product implicit in the format	✗	✓	✗	✓
Characters per elem. step *	302 ± 223	176 ± 113	139 ± 65	**77** ± **33**

minimally mapped, and the arrows that will act on the structure. Two example MechSMILES are depicted below as examples.

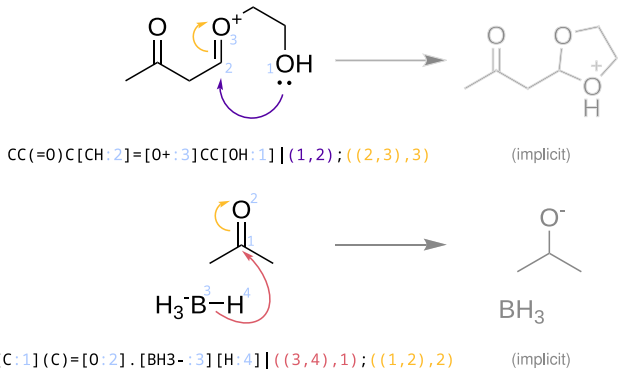

Figure 1: Two examples of MechSMILES, with the colors illustrating the purpose of each part of the string. On this figure, attacks are violet, bond-attacks pink and ionizations yellow.

Compared to existing formats in literature, MechSMILES offers three key advantages for conveying a mechanistic action (Table 1, Figure S2). First, its character efficiency per elementary step (44.6% more efficient than the second-densest format) makes it ideal for text-based models, effectively reducing the cost at both training and inference, without sacrificing human readability. This efficiency emerges from its complementarity with our environment: by encoding only reactants and arrows, the product becomes redundant information that can be computed. Second, its step-wise flexibility allows the model to include only the molecules interacting at each step, implicitly capturing the sequential order of reagent addition. Third, explicit hydrogen representation enables autonomous selection of specific atoms, including hydrogens. This feature becomes particularly important for stereospecific mechanisms, such as E2 eliminations that deviate from Zaitsev's rule due to steric constraints, where the identity of the eliminated hydrogen determines the stereochemical outcome.

Alternative formats reported in the literature have made different design trade-offs. Some provide the reactant set mapped for every heavy atom, and then encode all arrows with respect to this fixed mapping [22]. This method removes expressiveness to the model, forcing it to refer to a particular hydrogen atom with respect to the heavy atom it is attached to. Other formats have chosen to explicit the partially mapped reactants, partially mapped products and a code for arrows [21], or have chosen to give the complete mapped reactants and products of every elementary step [19]. These formats, while not directly designed for language-based tasks, require substantially more characters to express information that becomes redundant with our environment. Given the growing adoption of language models for chemistry applications [23–26], MechSMILES represents a promising text-based representation that balances a high expressiveness at inference time, a preservation of human readability and a high character efficiency, critical considerations for text-based architectures.

3 Results

We present results in two parts. First, we benchmark model performance across tasks of increasing difficulty, from elementary step prediction (where all chemical information is provided) to complete mechanism reconstruction (where only reactant species and main product are given). Second, we demonstrate three applications enabled by mechanism prediction: post-hoc validation of CASP proposals, holistic atom-to-atom mapping including hydrogens, and extraction of catalyst-aware reaction templates. Details on the datasets and training procedures including model hyperparameters are available in the SI.

3.1 Evaluation on tasks with increasing difficulty

We evaluated model performance across four tasks with increasing complexity as shown in Figure 2 to assess the capabilities and limitations.

- **Task 1 – Elementary Step Prediction:** Given current molecules and next intermediate, predict the electron movements. This is essentially a transcription and annotation task, as all chemical information is provided.
- **Task 2 – Equilibrated Reaction:** Given current molecules and all final products (including by-products, with stoichiometry), predict the next step of the mechanism. This requires chemical planning to identify the sequence of elementary steps.
- **Task 3 – Reaction without By-products:** Given current molecules with stoichiometry and main product only, predict the mechanism. The model must infer which species are consumed and ignore spectators.
- **Task 4 – Reaction without Stoichiometry:** Given available species (without stoichiometry) and main product only, predict the mechanism. This mirrors real human-level mechanism prediction scenarios where only reactants, conditions and desired product are known.

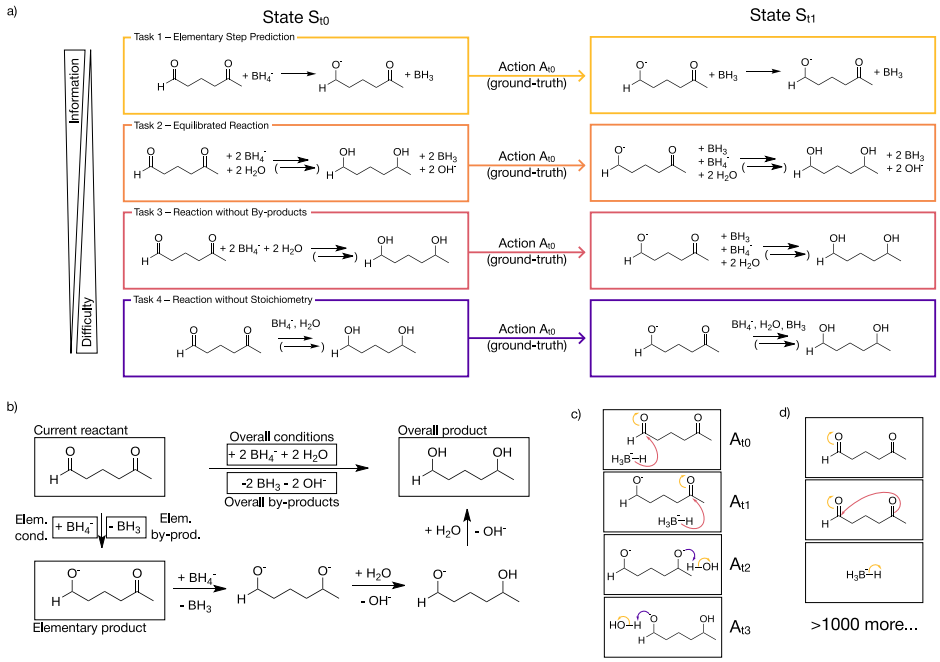

Figure 2: Reaction mechanism prediction framework. (a) Progressive task difficulty showing elementary step prediction with decreasing input information from Task 1 (full information) to Task 4 (no stoichiometry). Model receives state information (left boxes) and predicts actions as MechSMILES. (b) Complete reduction mechanism decomposed into elementary steps with reactants/conditions and products/by-products (distinction for visualization only). (c) Ground-truth action sequence from initial to goal state. (d) Sample legal arrow pushing moves on state S_{t0}, where >1000 moves are possible but only a small subset are chemically meaningful and productive.

We trained T5-based models[27] with a custom MechSMILES tokenizer adapted from Schwaller et al.[28] to handle both SMILES tokens and mechanism-specific notation. We evaluated our approach on two datasets: mech-USPTO-31k[22] (using a random split at reaction level) and FlowER[19] (using their original splits), as well as on PMechDB[21](using their original splits) for the elementary step task. While the random split employed for mech-USPTO-31k enables efficient benchmarking, reaction-template based splits could equally provide additional insight into the generalization capabilities of a model trained on this dataset. Table 2 summarizes performance across all tasks and datasets for the T5-based models, compared to some LLaMa[29] models. Our training framework is language model architecture-agnostic and other models[30,31] could be added in subsequent versions.

Table 2: Performance of models on mechanism prediction tasks (*Results of FlowER models are reported for a related but different task: forward prediction via mechanisms). Elementary step accuracy measures correctness at each step; pathway accuracy measures the recovery of the complete mechanism for a given beam width Bw. n/a indicates "not applicable", because the task has only one step (no complete mechanism to retrieve).
flow: FlowER[19], m-us: mech-USPTO-31k[22], pmdb: PMechDB split-0[17]

Training			**Evaluation**	
Task	Dataset	Model	Elem. step acc. [%] Top-1 (Top-3)	Pathway accuracy [%] Bw = 1 (Bw = 3)
Elementary step	flow	T5	99.70 (99.89)	n/a
		LLaMa	97.51 (98.65)	
	m-us	T5	96.63 (97.11)	
	pmdb	T5	80.86 (85.15)	
Equilibrated reaction	flow	FlowER*[19]	88.48 (97.96)	88.97 (96.48)
		FlowER-large*[19]	89.74 (98.66)	92.50 (98.15)
		T5	84.27 (98.14)	96.02 (98.69)
		LLaMa	78.22 (92.63)	76.92 (84.62)
	m-us	T5	96.47 (97.35)	75.22 (87.37)
Reaction (w/o by-prod.)	flow	T5	84.21 (98.11)	95.86 (98.56)
		LLaMa	79.75 (93.69)	74.26 (86.23)
	m-us	T5	96.48 (97.38)	76.03 (88.81)
Reaction w/o stoichio. (w/o by-prod.)	flow	T5	83.33 (97.61)	93.16 (97.58)
		LLaMa	77.38 (91.83)	68.24 (81.85)
	m-us	T5	95.72 (96.56)	73.33 (86.54)

As expected, the elementary step prediction task achieves near-perfect performance for two biggest datasets. These results serve as a control experiment: since all chemical information is contained in the input, the model only needs to rewrite the reactants with arrow annotations. For the three remaining tasks, which require chemical *planning* in addition to annotation, we observe that top-1 accuracy per step decreases modestly as the task difficulty increases. Interestingly, this trend does not consistently hold for top-3 accuracy, suggesting that beam search produces high-quality alternatives for the more challenging tasks. At inference, beam search of width n will maintain the n most probable sequences, with sequence scores computed as a heuristic score based on the sum of the log probabilities of the model at each decoding step. Comparing the two datasets, we observe higher top-1 elementary step accuracy on mech-USPTO-31k[22] dataset compared to FlowER dataset[19], likely due to the random split performed on mech-USPTO-31k[22], enabling the model to train on the same distribution of reaction templates than its test set. However, this advantage does not translate to complete mechanism retrieval, where the trend is actually reversed. Retrieval of the complete ground-truth mechanism with a beam-width of 1 resulted in a performance of 73.33% on mech-USPTO-31k[22], and 93.16% on FlowER dataset[19] for our most challenging task. The fact for the full mechanism retrieval to have a better score than its elementary step accuracy on the FlowER dataset[19] can be explained by the fact that multiple elementary steps that are part of the same mechanism are aggregated on the ground-truth retrieval metric, potentially grouping multiple unsuccessful step predictions under a single ground-truth not found. Overall, both datasets demonstrate sufficient size and quality to enable effective learning of the chemical transformation spaces they represent.

3.2 Transfer learning to new classes (ozonolysis and Suzuki cross-coupling)

Once the models were trained, we decided to investigate the amount of data necessary for the model to learn a new type of mechanism. We chose ozonolysis and Suzuki cross-coupling reactions to showcase this transfer for several reasons. These reactions have both a non-trivial mechanism of multiple steps, represent known and robust chemical transformations, and were absent from training (ozonolysis) or severely underrepresented (11 examples for Suzuki), resulting in poor performance of the model trained on the "reaction without by-products" task on the FlowER dataset for these two reaction classes (Figure 3).

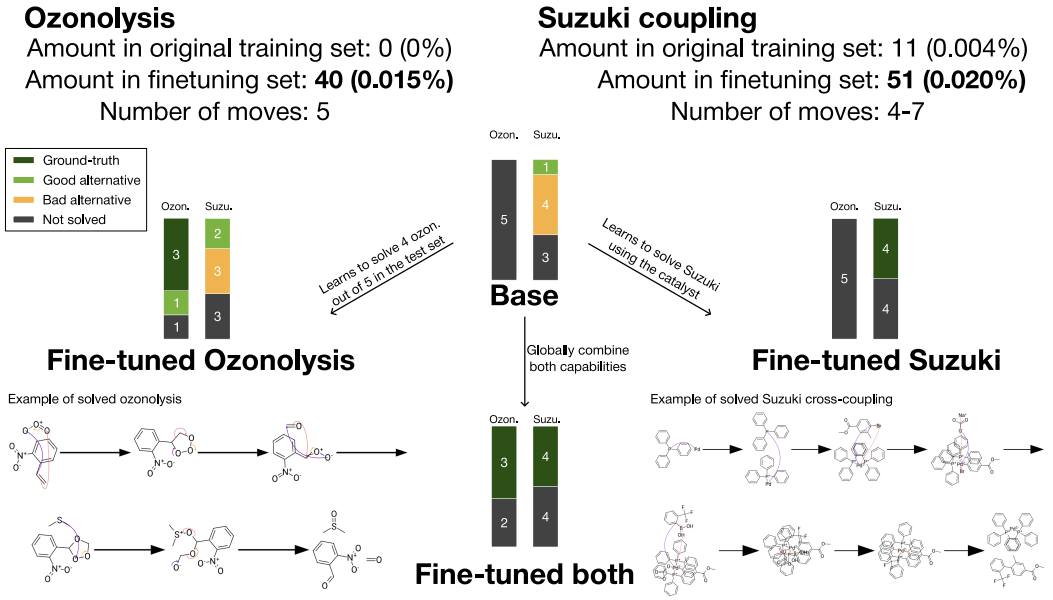

Figure 3: Transfer learning results showing important improvement after fine-tuning on small curated datasets. The base model (trained on FlowER without by-products task) achieves 0 out of 5 and 1 out of 8 accuracy on ozonolysis and Suzuki test sets respectively, while fine-tuned models trained on only 40 additional manually annotated examples for each class achieve 3 out of 5 and 4 out of 8 reactions in the test set.

To enable this transfer learning, we curated small datasets manually, using a custom-made GUI for drag-and-drop mechanism annotations (Details in SI). We selected reactions from USPTO-MIT 480k[32] that matched our criteria: ozonolysis reactions with reductive workup and Suzuki couplings with carbonate salt. This selection resulted in datasets of 47 and 50 reactions mechanistically annotated, respectively. These were split into train/validation/test sets (40/2/5 for ozonolysis, 40/2/8 for Suzuki). The base model (trained only on FlowER without by-products task) achieved 0% (0/5) and 12.5% (1/8) accuracy on these test sets. After fine-tuning on the training sets, performance improved to 60% (3/5) for ozonolysis and 50% (4/8) for Suzuki reactions, demonstrating effective few-shot transfer learning for novel mechanism classes. Since our curated dataset represents only a subset of qualifying Suzuki reactions in USPTO-MIT 480k, the fine-tuned model could now be leveraged to generate additional training data through automatic annotation of the remaining examples.

3.3 Application 1: Explainability of a CASP model

Once trained on the mechanism prediction task, a model can serve as a policy for an agent acting on the environment, interacting with molecule sets through arrow pushing. This agent can operate in a free search setting to validate individual reaction steps proposed by other models. Existing state-of-the-art CASP methods, whether template-based or template-free, generally do not decompose single step transformation further. Without mechanistic or physics based grounding, chemically implausible transformation undergoing highly energetic intermediate states could be accepted. By

restraining our agent to the arrow pushing formalism, we ensure a physically grounded path exists, decreasing the chance of an unlikely transformation being accepted. Concrete examples of this application can be seen in Figures 4 and S1. In case an even higher confidence is needed, the explicit path proposed by the agent can be assessed energetically to verify its validity.

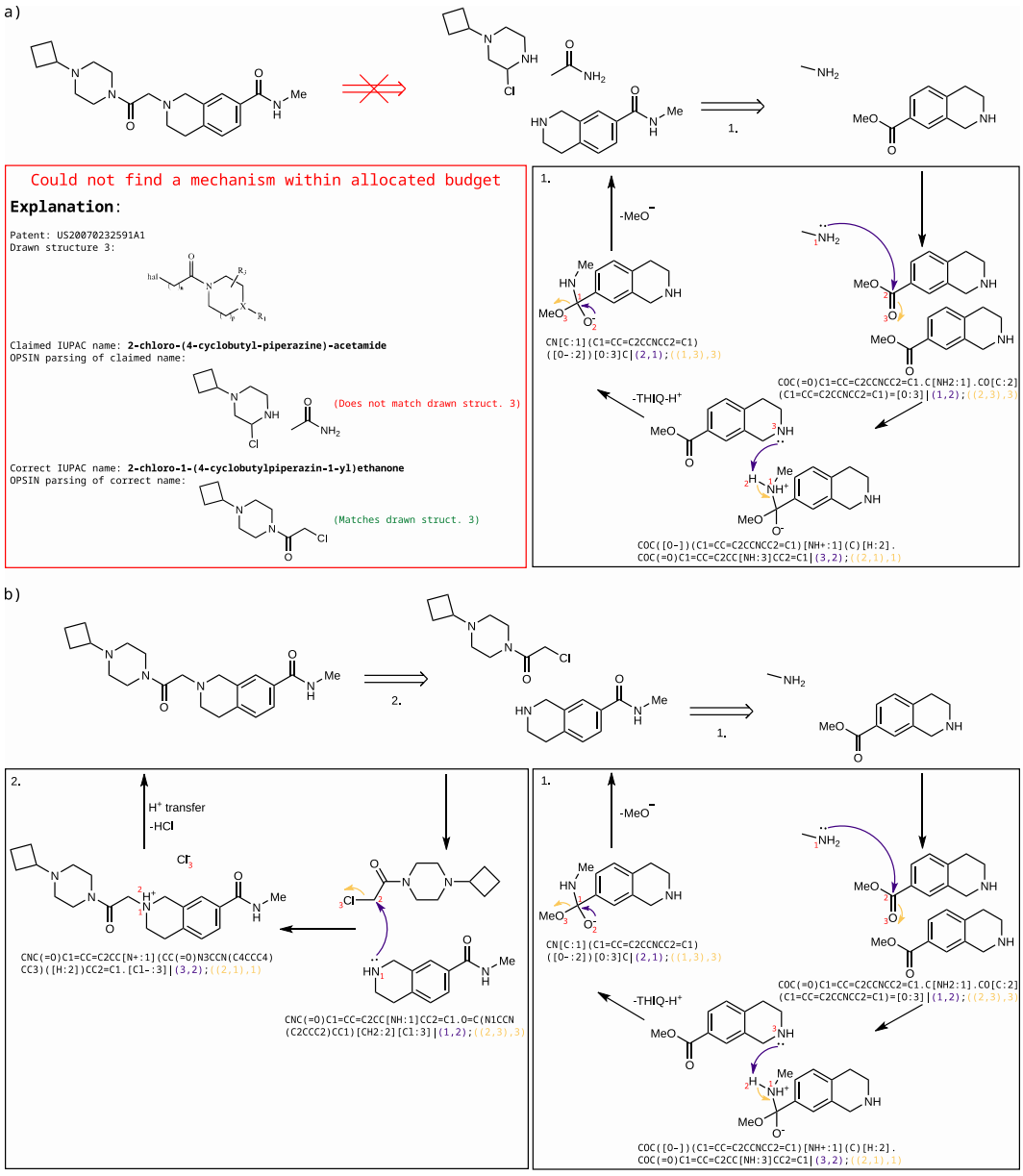

Figure 4: a) Example of a CASP validation of the multistep reaction visible in figure S2 of the PaRoutes paper[33]. No mechanism can be found for the last reaction, with similar search settings as figure S1 (same model, search algorithm and budget), hinting that this reaction might be wrong. After investigation, we found out this error might come from the name *2-chloro-(4-cyclobutyl-piperazine)-acetamide* not respecting IUPAC rules in the original patent, potentially confusing tools such as OPSIN[34] for the creation of the molecules. b) The multistep reaction found in the original patent. After our correction, the now corrected last step finds a simple mechanism, hinting at the fact that this is indeed the correct transformation.

3.4 Application 2: Holistic Atom-to-Atom Mapping Including Hydrogens

Furthermore, once the mechanism of a reaction is known, this information unlocks the understanding of the role of each atom in the reaction. In particular, this level of understanding allows for a mapping that can track every atom, hydrogens included. In the simple case of a ketone reduction by BH_4^-, hydrogen atoms play a crucial role, and two species (water and borohydride) lose hydrogen atoms that are *transferred* to the ketone. While typical state-of-the-art atom-to-atom mapping tools[35,36] would not allow us to follow the path of these atoms, mostly because they are based on implicit-hydrogens SMILES, the examination of the mechanism makes the role of each species straightforward. Even though this example is chosen to be chemically simple, the same concept applies to more complex reactions.

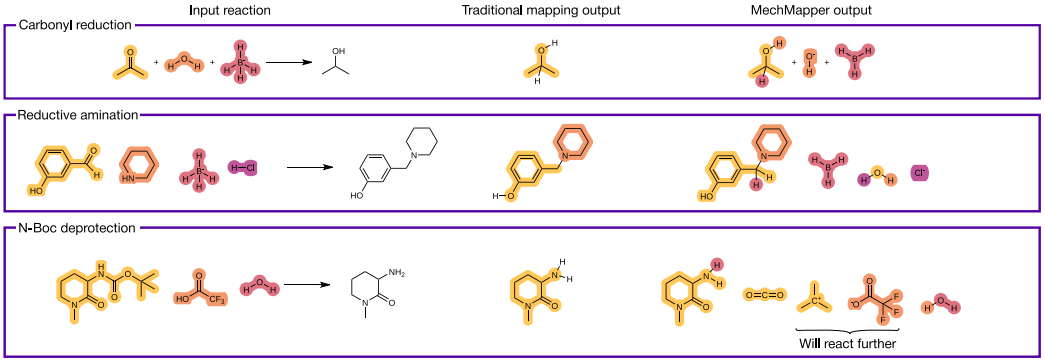

Figure 5: Few example reactions mapped both with SOTA tools, and with mechanistic mapping using our model as the mechanism predictor. Even though the mapping on heavy atoms is similar, the mechanism insight brings more information to the user, mapping hydrogens as well as by-products.

3.5 Application 3: Catalyst-Aware Reaction Template Extraction

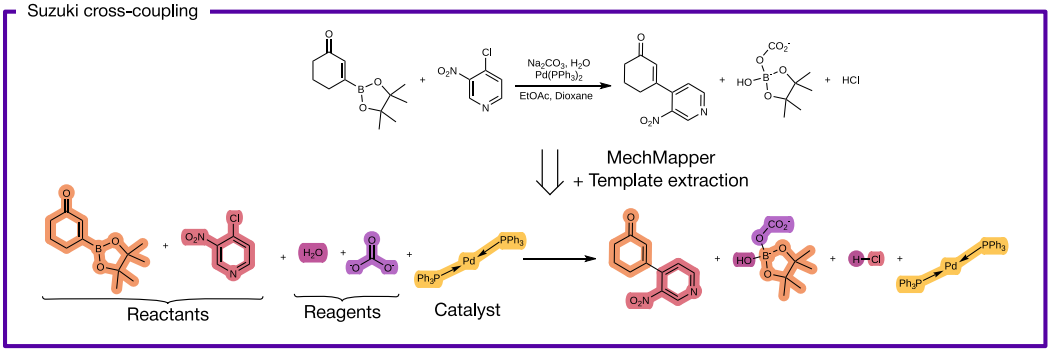

Figure 6: Suzuki coupling reaction taken from the test set of the FlowER dataset[19]. $Pd(PPh_3)_2$ that can be seen over the arrow is a recycled catalyst. Visual representation of the mechanistic template of radius $= \infty$ of this reaction. The mechanistic template distinguishes a recycled catalyst from other pure observer species, and includes it in the template.

In addition to that, once every elementary step is extracted, and a reaction is mapped with respect to its mechanism, it is equally straightforward to extract a catalyst-aware template. What is meant by that is that typically, template creation is either done by experts[1,19] or via an automatic SMARTS transformation creation, using mapping as a guide to understand the environment change of each atom[37]. However, by design, this latter method is blind to the effect of a species interacting with the substrates during intermediate steps, but recycled before the final step of a reaction. Such species, that we will call catalysts (but also encompass some acido-basic species), are usually crucial or highly favor the formation of the product. The mechanistic understanding distinguishes naturally purely observer species, e.g. solvents, from species that have a role in the intermediate stages of the

mechanism, e.g. catalysts. As can be seen in Figure 6, during a Suzuki reaction, as the catalyst is recycled, the inclusion of the palladium species in the template is non-obvious without knowing the mechanism. The same principle could be transferred to other reactions with recycled catalysts.

4 Discussion

Overall, we demonstrate that mechanism prediction agents can interact with chemical species through arrow pushing, with very high accuracy score, surpassing 95.72% and 83.33% top-1 on our most challenging task for both mech-USPTO-31k[22] and FlowER[19] datasets respectively. For complete reaction mechanisms retrieval, models reach performances of 73.33% and 93.16% accuracy on the same datasets. Top-3 scores reach values > 96% on the step accuracy for both datasets, and the complete reaction mechanisms retrieval with a beam of size 3 reaches 86.54% on mech-USPTO-31k[22] and 97.58% on FlowER[19] dataset. This capacity offers an alternative approach to synthesizability, framed in a chemically-grounded formalism that respects the laws of conservation of mass and charge. Moreover, search-based mechanism prediction can act as a post-hoc rationalizer. In that capacity, it is able to enhance already established CASP models, whether template-based or template-free, by automatically filtering reactions that are valid as graph transformations but physically implausible. Practical integration might involve pairing our model with a condition generation model when working with CASP systems that do not explicitly propose such information.

Beyond validation, we demonstrate that the mechanism prediction task on its own can serve multiple purposes. First, as a holistic atom-to-atom mapper, capable of tracking every atom, including hydrogens. Second, as template extractor that understands the broader context of the whole transformation and differentiates a regenerated catalyst from a purely spectator species. This mechanistic understanding captures information invisible to methods traditionally used to automatically perform both of these tasks. The current framework focuses on polar mechanisms involving closed-shell electron movements, representing the majority of synthetic organic transformations; extension to radical pathways is envisioned to broaden its applicability to additional reaction classes in future work. From a methodological perspective, the mechanism environment represents an architecture-agnostic framework that allows for a fair comparison of different works related to mechanisms. Written in Python, it is accessible to the broader computational chemistry community.

We see this work as a step towards more explainable and chemically grounded CASP. The mechanistic formalism presented here provides a foundation for integrating different levels of abstraction, leading to a more holistic view of the synthetic process, and ultimately enabling more robust and interpretable tools for automated synthetic chemistry.

Acknowledgments

T.A.N. acknowledges support from Intel and Merck KGaA via the AWASES programme. Z.J. acknowledges support by the Swiss National Science Foundation (SNSF) (grant number 214915). P.S. acknowledges support from the NCCR Catalysis (grant number 225147), a National Centre of Competence in Research funded by the Swiss National Science Foundation.

Data and code availability

All datasets, models, and their tokenizers are available through the HuggingFace Hub, with all mechanistic steps being conveyed in the MechSMILES format (`https://huggingface.co/SchwallerGroup`). Moreover, code is available at `https://github.com/schwallergroup/ChRIMP`

References

[1] Bartosz A. Grzybowski, Tomasz Badowski, Karol Molga, and Sara Szymkuć. Network search algorithms and scoring functions for advanced-level computerized synthesis planning. *WIREs Comput. Mol. Sci.*, 13(1):e1630, 2023.

[2] 1. Reaxys database, 2024. URL `https://www.reaxys.com`. (Accessed Jul 29, 2021).

[3] Marwin HS Segler, Mike Preuss, and Mark P Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610, 2018.

[4] Philippe Schwaller, Riccardo Petraglia, Valerio Zullo, Vishnu H Nair, Rico Andreas Haeuselmann, Riccardo Pisoni, Costas Bekas, Anna Iuliano, and Teodoro Laino. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chemical science*, 11(12):3316–3325, 2020.

[5] Samuel Genheden, Amol Thakkar, Veronika Chadimová, Jean-Louis Reymond, Ola Engkvist, and Esben Bjerrum. AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic planning. *J. Cheminf.*, 12:1–9, 2020.

[6] Philippe Schwaller, Alain C Vaucher, Ruben Laplaza, Charlotte Bunne, Andreas Krause, Clemence Corminboeuf, and Teodoro Laino. Machine intelligence for chemical reaction space. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(5):e1604, 2022.

[7] Lakshidaa Saigiridharan, Alan Kai Hassen, Helen Lai, Paula Torren-Peraire, Ola Engkvist, and Samuel Genheden. Aizynthfinder 4.0: developments based on learnings from 3 years of industrial application. *Journal of cheminformatics*, 16(1):57, 2024.

[8] Zhengkai Tu, Sourabh J Choure, Mun Hong Fong, Jihye Roh, Itai Levin, Kevin Yu, Joonyoung F Joung, Nathan Morgan, Shih-Cheng Li, Xiaoqi Sun, et al. Askcos: Open-source, data-driven synthesis planning. *Accounts of Chemical Research*, 58(11):1764–1775, 2025.

[9] Andres M Bran, Theo A Neukomm, Daniel P Armstrong, Zlatko Jončev, and Philippe Schwaller. Chemical reasoning in llms unlocks steerable synthesis planning and reaction mechanism elucidation. *arXiv preprint arXiv:2503.08537*, 2025.

[10] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):8, 2009.

[11] Connor W Coley, Luke Rogers, William H Green, and Klavs F Jensen. Scscore: synthetic complexity learned from a reaction corpus. *Journal of chemical information and modeling*, 58 (2):252–261, 2018.

[12] Rebecca M Neeser, Bruno Correia, and Philippe Schwaller. Fsscore: A personalized machine learning-based synthetic feasibility score. *Chemistry-Methods*, 4(11):e202400024, 2024.

[13] Sara Szymkuć, Ewa P Gajewska, Tomasz Klucznik, Karol Molga, Piotr Dittwald, Michał Startek, Michał Bajczyk, and Bartosz A Grzybowski. Computer-assisted synthetic planning: the end of the beginning. *Angewandte Chemie International Edition*, 55(20):5904–5937, 2016.

[14] Hitesh Patel, Wolf-Dietrich Ihlenfeldt, Philip N Judson, Yurii S Moroz, Yuri Pevzner, Megan L Peach, Victorien Delannée, Nadya I Tarasova, and Marc C Nicklaus. Savi, in silico generation of billions of easily synthesizable compounds through expert-system type rules. *Scientific data*, 7(1):384, 2020.

[15] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28 (1):31–36, 1988.

[16] John Bradshaw, Matt J. Kusner, Brooks Paige, Marwin H. S. Segler, and José Miguel Hernández-Lobato. A generative model for electron paths. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=r1x4BnCqKX`.

[17] Ryan J Miller, Alexander E Dashuta, Brayden Rudisill, David Van Vranken, and Pierre Baldi. Mechanism-aware deep learning for polar reaction prediction. *Journal of the American Chemical Society*, 2025.

[18] Shuan Chen, Kye Sung Park, Taewan Kim, Sunkyu Han, and Yousung Jung. Predicting chemical reaction outcomes based on electron movements using machine learning. *arXiv preprint arXiv:2503.10197*, 2025.

[19] Joonyoung F Joung, Mun Hong Fong, Nicholas Casetti, Jordan P Liles, Ne S Dassanayake, and Connor W Coley. Electron flow matching for generative reaction mechanism prediction. *Nature*, pages 1–9, 2025.

[20] Daylight Theory: SMIRKS. URL `https://www.daylight.com/dayhtml/doc/theory/theory.smirks.html`. (Accessed Nov 15, 2021).

[21] Mohammadamin Tavakoli, Ryan J Miller, Mirana Claire Angel, Michael A Pfeiffer, Eugene S Gutman, Aaron D Mood, David Van Vranken, and Pierre Baldi. Pmechdb: A public database of elementary polar reaction steps. *Journal of Chemical Information and Modeling*, 64(6): 1975–1983, 2024.

[22] Shuan Chen, Ramil Babazade, Taewan Kim, Sunkyu Han, and Yousung Jung. A large-scale reaction dataset of mechanistic pathways of organic reactions. *Scientific Data*, 11(1):863, 2024.

[23] Andrew D White. The future of chemistry is language. *Nature Reviews Chemistry*, 7(7): 457–458, 2023.

[24] Andres M Bran and Philippe Schwaller. Transformers and large language models for chemistry and drug discovery. In *Drug Development Supported by Informatics*, pages 143–163. Springer, 2024.

[25] Mayk Caldas Ramos, Christopher J Collison, and Andrew D White. A review of large language models and autonomous agents in chemistry. *Chemical science*, 2025.

[26] Robert MacKnight, Daniil A Boiko, Jose Emilio Regio, Liliana C Gallegos, Théo A Neukomm, and Gabe Gomes. Rethinking chemical research in the age of large language models. *Nature Computational Science*, pages 1–12, 2025.

[27] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[28] Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A Hunter, Costas Bekas, and Alpha A Lee. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS central science*, 5(9):1572–1583, 2019.

[29] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[30] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[31] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[32] Wengong Jin, Connor Coley, Regina Barzilay, and Tommi Jaakkola. Predicting organic reaction outcomes with weisfeiler-lehman network. *Advances in neural information processing systems*, 30, 2017.

[33] Samuel Genheden and Esben Bjerrum. Paroutes: towards a framework for benchmarking retrosynthesis route predictions. *Digital Discovery*, 1(4):527–539, 2022.

[34] Daniel M Lowe, Peter T Corbett, Peter Murray-Rust, and Robert C Glen. Chemical name to structure: Opsin, an open source solution, 2011.

[35] Philippe Schwaller, Benjamin Hoover, Jean-Louis Reymond, Hendrik Strobelt, and Teodoro Laino. Extraction of organic chemistry grammar from unsupervised learning of chemical reactions. *Science Advances*, 7(15):eabe4166, 2021.

[36] Shuan Chen, Sunggi An, Ramil Babazade, and Yousung Jung. Precise atom-to-atom mapping for organic reactions via human-in-the-loop machine learning. *Nature Communications*, 15(1): 2250, 2024.

[37] Christos Kannas, Amol Thakkar, Esben Bjerrum, and Samuel Genheden. rxnutils–a cheminformatics python library for manipulating chemical reaction data. 2022.

[38] Nextmove software namerxn. URL `https://www.nextmovesoftware.com/namerxn.html`. (Accessed Nov 30, 2025).

Supplementary Information

A Additional concrete example of CASP validation via mechanism prediction

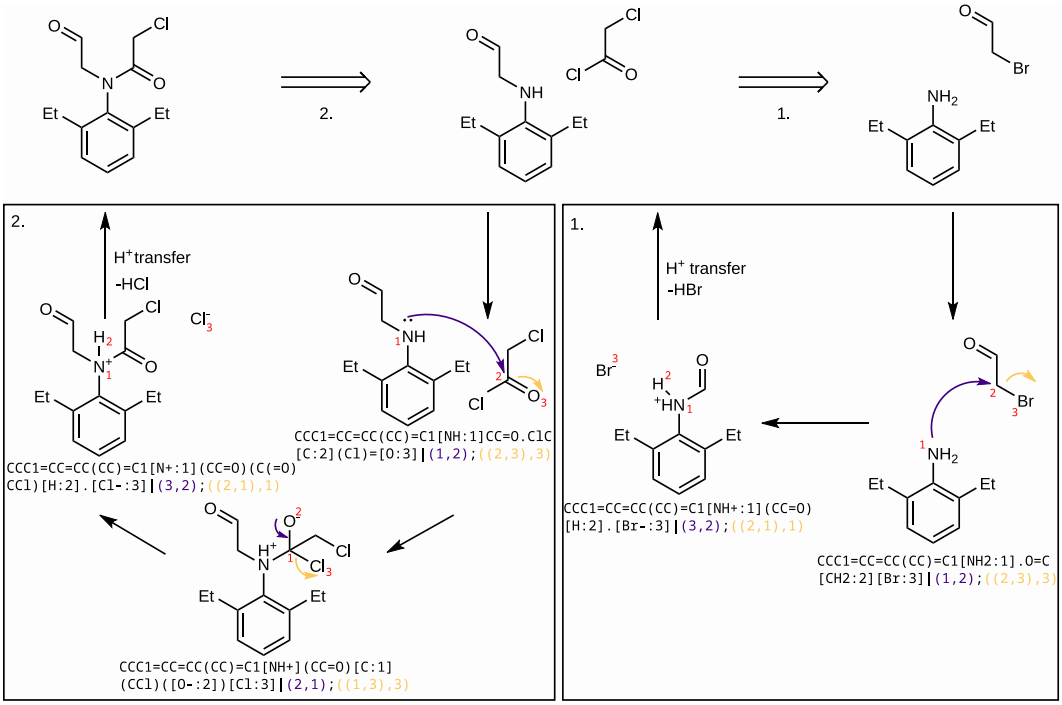

Figure S1: Example of a CASP validation of the multistep reaction visible in figure 1 of the PaRoutes paper[33]. Each step of this retrosynthesis (numbered arrows) is solved mechanistically in the frame below it, and with similar number. The mechanism found for each step comforts us in accepting these transformations as valid. All MechSMILES of this figure have been found by our model trained on the FlowER dataset, on the task of reaction without by-products. For the search algorithm, we used a best-first search algorithm with a maximum budget of 50 nodes to expand. In this particular case, the model performed perfectly, solving the arrow 1 by expanding 2 nodes, and arrow 2 by expanding 3 nodes.

B Training format

As mentioned in the main text, the performances exhibited in this article come uniquely from language-based models. Therefore we decided to train these models with special tokens telling the model which part of the input represents the reactants `[reac]`, and which part represents the product `[prod]`. Because we wanted to be able to train decoder-only models as well, we equally ended every input with the token `[mech]`, marking the end of the input and the beginning of the mechanism prediction. In order to let the model focus on the chemistry and not on positional artifacts, each training input has been shown in the forward direction (reactant before product) and in the retro direction (the opposite). Equally, to try to infuse the reactivity of species, training samples have been shown to our models without the product. Despite the high performances observed with this methodology, an ablation study would be necessary to assess the utility of each of these data augmentations.

The inputs for the example task, shown in figure 2, as well as the expected output for all of them, are the following:

Elementary steps:
`[prod]B.CC(=O)CCCC[O-][reac]CC(=O)CCCC=O.[BH4-][mech]`
`[reac]CC(=O)CCCC=O.[BH4-][prod]B.CC(=O)CCCC[O-][mech]`
`[reac]CC(=O)CCCC=O.[BH4-][mech]`

Equilibrated reactions:
`[prod]B.B.CC(=O)CCCC[O-].[OH-].[OH-][reac]CC(=O)CCCC=O.O.O.[BH4-].[BH4-][mech]`
`[reac]CC(=O)CCCC=O.O.O.[BH4-].[BH4-][prod]B.B.CC(=O)CCCC[O-].[OH-].[OH-][mech]`
`[reac]CC(=O)CCCC=O.O.O.[BH4-].[BH4-][mech]`

Stoichio. reactants, no by-prod:
`[prod]CC(=O)CCCC[O-][reac]CC(=O)CCCC=O.O.O.[BH4-].[BH4-][mech]`
`[reac]CC(=O)CCCC=O.O.O.[BH4-].[BH4-][prod]CC(=O)CCCC[O-][mech]`
`[reac]CC(=O)CCCC=O.O.O.[BH4-].[BH4-][mech]`

No stoichio., no by-prod.:
`[prod]CC(=O)CCCC[O-][reac]CC(=O)CCCC=O.O.[BH4-][mech]`
`[reac]CC(=O)CCCC=O.O.[BH4-][prod]CC(=O)CCCC[O-][mech]`
`[reac]CC(=O)CCCC=O.O.[BH4-][mech]`

Ground-truth output (minimal MechSMILES of the next arrow pushing move):
`[BH3-:2][H:3].[CH:4](CCCC(C)=O)=[O:1]|((2, 3), 4);((4, 1), 1)`

C Data

C.1 Statistics

The work done in this article was enabled in parts thanks to the two large mechanistic datasets released by Joung et al.[19] and Chen et al.[22]. The first has been released as a large dataset of mapped reactants and products of elementary steps, along with an index indicating which elementary steps are part of the same reaction. An algorithm was written to convert these mapped elementary reactions to MechSMILES, and the same splits as the original work were used for the training, validation and test sets respectively. The second has been released as multi-elementary steps directly, associated with all arrows necessary to pass from the initial reactants to final products. As no indication of the stable intermediates were given, we designed an algorithm checking for compliance with our environment for each arrow. Each intermediate that could be created in our environment has been deemed a stable intermediate of the mechanism. As no splits were explicitly given for this dataset, we performed a random split at the reaction level (keeping all elementary steps of the same reaction in the same split). Moreover, we also performed elementary step predictions on one split of the PMechDB dataset[21] released in a recent work[17], this dataset is shared as a list of independent elementary steps, shared as a SMIRKS accompanied by an *arrow-code*, effectively encoding for the transformation, and the arrows respectively. All the data that was used for training and evaluation are available publicly through HuggingFace datasets.

Table S1: Additional metrics on datasets.* Extracted successfully the with the snippet mentioned in section below. ** Trivial elementary steps where no transformation happens and duplicates are not accounted for.

Dataset	mech-USPTO-31k[22]	FlowER[19]	PMechDB split-0[17]
Reactions reported	31,199	289,024	not reported
Reactions extracted*	31,199 (100%)	259,834 (89.9%)	n/a
Elem. steps reported**	not reported	1,748,823	13,104
Elem. steps extracted*	114,826	1,435,192 (82.1%)	12,951 (98.8%)
Train/val/test splits	80/10/10	89/1/10	80/10/10

C.2 Curation

Snippets of code have been developed to translate the different formats of arrow pushing move to our MechSMILES format. In particular, we give access, along with our environment, to snippets of code capable of curating and translating the following formats:

1. Fully mapped elementary steps (similar to the Flower[19] format)
2. From a reactant and the set of all arrows (similar to the mech-USPTO-31k[22] format)
3. SMIRKS accompanied by an arrow-code (similar to PMechDB[21] format)

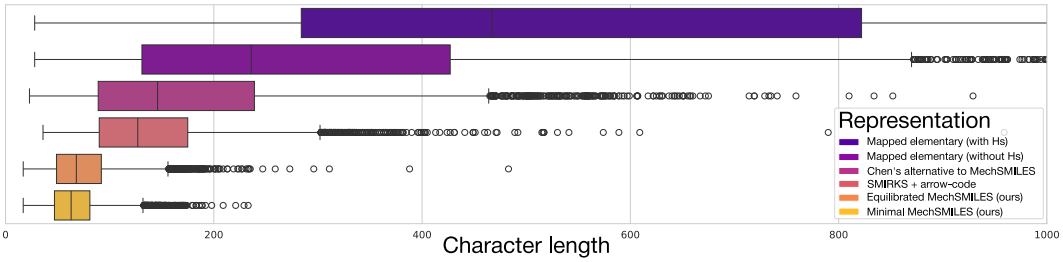

Figure S2: Character length distribution to encode the same mechanistic data using the different formats mentioned in this work. The main difference between equilibrated and minimal MechSMILES is that the latter will not explicitely rewrite species that do not interact in the specific elementary step. The data used for this example is the training set of PMechDB split-0[17]

D Data curation for transfer learning

To train a model on the transfer learning task, we first analyzed the USPTO-MIT 480K dataset[32] and tagged reactions using NameRXN software[38]. For all classes (down to 2nd hierarchy level) containing at least 200 reactions (65 classes total), we randomly selected 20 representatives and attempted to predict a complete mechanism using our best T5 model on the "reaction without by-products" task, conducting 1,300 searches total. The parameters of this search were: budget of 50 nodes to expand, using a best-first search algorithm (using a heuristic score based of the sum of log-probabilities given by our model during inference), with maximum expansion of 5 nodes from the top-10 model predictions.

Following this screening, we performed manual inspection of the reaction classes and then selected two reaction classes that were absent or severely underrepresented in training and showed poor search performance: ozonolysis (0 training examples, 0/20 mechanisms solved) and Suzuki cross-coupling (11 training examples, 4/20 mechanisms solved). Note that these success rates represent a pessimistic lower bound on model capabilities, as many reactions in USPTO-MIT 480K lack reagents or conditions necessary for complete mechanistic prediction via arrow pushing.

To create high-quality training data for these classes, we manually curated two datasets using a custom web-based Graphical User Interface (GUI) with drag-and-drop arrow pushing capabilities (front-end) connected to our computational environment (back-end). Screenshots of this GUI are shown in Figure S3. The final curated datasets consisted of 47 ozonolysis reactions with reductive workup and 50 Suzuki cross-coupling reactions with carbonate salt as base.

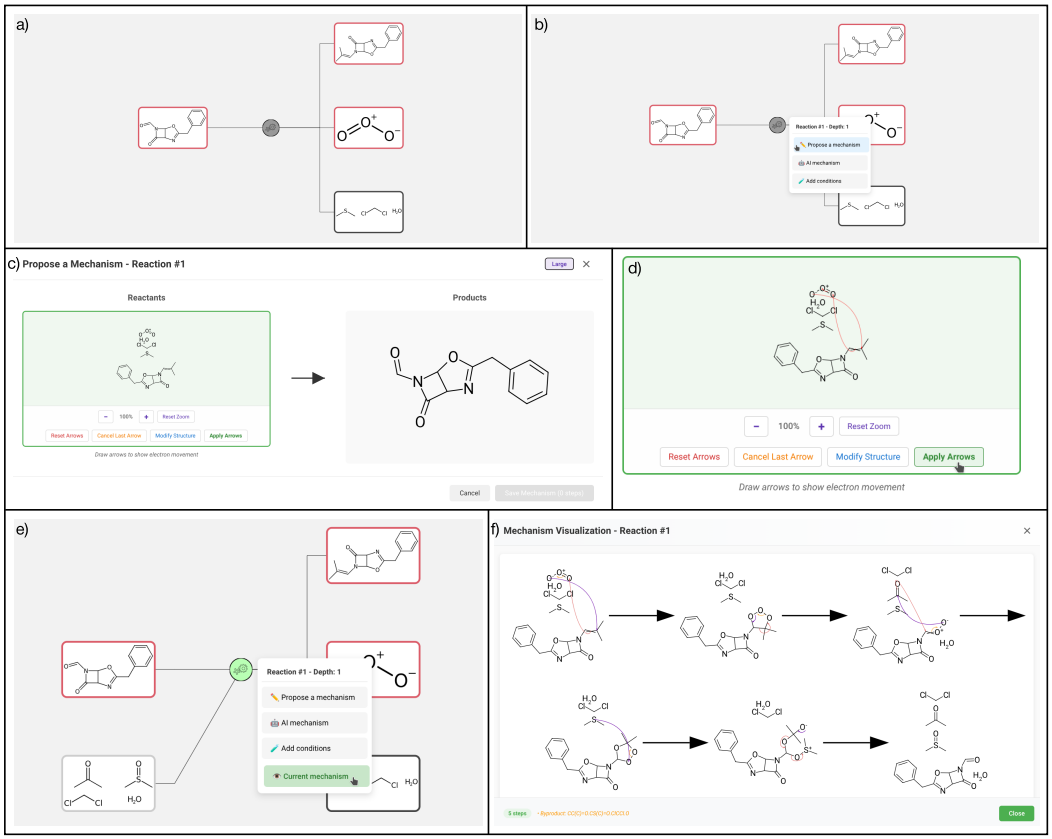

Figure S3: Screenshot of the GUI used to create mechanistic database. a) Global view of the different nodes are mixed to form the product (right to left). b) The user can propose a mechanism manually. c-d) The manual mechanism panel opens, and the user can draw arrows by drag-and-drop on the structure, clicking "Apply arrows" to get the next intermediate, computed from the drawn arrows. e-f) Once a mechanism is informed for a reaction, the by-products are automatically computed and the user can get a global view to verify no mistake has been made.

E Other architectures

To display the capability of this format to be learnt by decoder-only models as well, we trained 4 models based on the LLaMa architecture on our 4 main tasks. These trainings have been performed on the FlowER dataset, being the biggest available, as a proof of concept that the environment could be used to train decoder-only architecture. However, seeing their performances were lower than the T5-based models, as can be seen in Table 2, the results of the main paper are reported for the encoder-decoder models.

F Hyperparameters of model

T5-model has been created with the default hyperparameters of HuggingFace's transformers package, except for the following list:

Table S2: Hyperparameters of T5-based model reported in the article

Key	Value
`d_model`	512
`d_ff`	1024
`num_heads`	6
`num_layers`	4
`vocab_size`	tokenizer's vocabulary size (260)

Table S3: Hyperparameters of LLaMa-based model reported in the article

Key	Value
`hidden_size`	512
`intermediate_size`	512
`num_attention_heads`	4
`num_hidden_layers`	6
`rope_theta`	8192
`vocab_size`	tokenizer's vocabulary size (260)