Improving Precision in Language Models Learning from Invalid Samples

Niels Jakob Larsen[*]
Technical University of Denmark
s213428@student.dtu.dk

Giorgio Giannone[*†]
Technical University of Denmark
gigi@dtu.dk

Ole Winther
Technical University of Denmark
University of Copenhagen
olwi@dtu.dk

Kai Blin[†]
Technical University of Denmark
kblin@biosustain.dtu.dk

Abstract

Language Models are powerful generative tools capable of learning intricate patterns from vast amounts of unstructured data. Nevertheless, in domains that demand precision, such as science and engineering, the primary objective is to obtain an exact and accurate answer. Precision takes precedence in these contexts. In specialized tasks like chemical compound generation, the emphasis is on output accuracy rather than response diversity. Traditional self-refinement methods are ineffective for such domain-specific input/output pairs, unlike general language tasks. In this study, we introduce `invalid2valid`, a powerful and general post-processing mechanism that can significantly enhance precision in language models for input/output tasks spanning different domains and specialized applications.

1 Introduction

In recent years large language models (LLMs [6, 25, 3]) have emerged as a transformative technology, exhibiting exceptional performance across a broad spectrum of tasks, providing a step towards a generalist generative model. Following such success, LLMs have started to be used in science and engineering, with the aim to accelerate and augment scientific discovery [17, 33].

LLMs excel in many scenarios, but for fields where precision is important, like chemistry, genomics, and engineering design, they pose challenges in controllability, deployment, and factual accuracy [3]. LLM sampling introduces errors, from inaccuracies to inconsistencies [19], potentially undermining their reliability in critical situations, potentially undermining their utility and reliability in high-stakes scenarios. Fine-tuning [20, 1], prompting [34] and in-context learning [3], self-refining mechanisms [36, 10, 27] aim to mitigate these limitations by implementing various strategies to improve and control the model's output. These mechanisms are essential for real-time applications where manual correction is infeasible. However, it is difficult to control for precision at the character or symbol level. If input/output mappings are from different domains, i.e. text to molecule [4, 7] or gene to molecule then self-reflection and self-critiques cannot be easily used. Prompting works only for large models. So there is a need for general methods that can be used with relatively small and specialized models to improve the precision of their predictions.

We propose `invalid2valid`, a method designed to enhance the precision of language models in situations where input and output domains differ. Our method employs a three-phase pipeline: a (i) *Generative Phase*, where we fine-tune a language model using a small dataset containing input/output

[*]Equal contribution.
[†]Equal advising.

NeurIPS 2023 Generative AI and Biology Workshop.

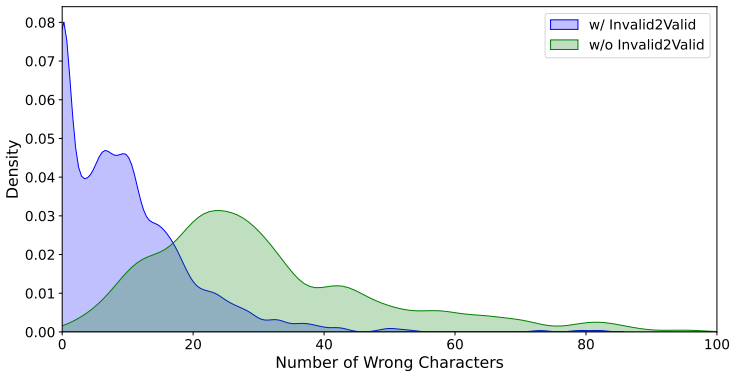


Figure 1: Distribution of wrong characters (i.e. SMILES symbols) on the test set for small molecule prediction given a gene cluster using a specialized fine-tuned model (green density) and using the same model plus invalid2valid refinement (blue density). Invalid2Valid greatly reduces the number of errors, changing the distribution and adding a large amount of mass toward a small number of mistakes. This result shows that our refinement mechanism is effective at improving generation precision for specialized domains.

pairs; a (ii) *Sampling Phase*, where, after fine-tuning, we freeze the model and generate multiple predictions for a given input; a (iii) *Constraining Phase*, where we fine-tune a second model using the collected mappings. The underlying idea is rooted in the fact that LLMs excel as reasoning engines and generative models but often struggle with precision and control. However, when LLMs make errors, these errors are often minor, affecting only a small portion of the output. Consequently, it becomes possible to establish a secondary, lightweight mapping to rectify these errors made during the generative phase. The objective of this secondary mapping is to correct and enhance the precision of the predicted output, encouraging the initial model's output to align with the correct target using a constraining mapping.

Contribution. Our contributions are the following: **(i)** We propose `invalid2valid`, a two-phase pipeline to improve language model precision. **(ii)** Our method is domain and model-agnostic and can be applied to any input/output pairs. **(iii)** We provide experimental validation on challenging constrained generative problems involving, text, genes, and molecules.

2 Background

Domain-specialized Language Models. General Language Models [32, 6, 3] have been widely successful for a large variety of tasks. However, for specialized domains, like chemistry, genomics, engineering design, and protein design, it is still valuable to build smaller, highly specialized generative models that are fast to sample, easy to deploy and iterate, and less prone to inaccurate output thanks to explicit supervision for a task or domain. In the life sciences, specialized language models have been developed for protein design [16], protein folding [9], protein generation [13] molecular generation [7, 4] and property prediction [2, 26], biology [14] and medicine [31].

Refinement Techniques for Language Models. Improving the output of LLMs is an active area of research. Finetuning techniques are used to achieve alignment to human preferences [20, 29] and reduce harm [1]. Refinement is leveraged to improve the quality and correctness of sampling: learnable prompting [12], Chain-of-thoughts prompting [34], self-refinement [15] and self-critique [27] aim to improve the effectiveness of LLMs for tasks involving accuracy. Overall, LLMs can be self-corrected using three main techniques: pre-processing, in-model-processing, and post-processing. Pre-processing involves manipulating input data to reduce ambiguity. In-model-processing integrates correction techniques into the model architecture itself. Post-processing applies corrections to the model's output. For more self-correction strategies see [21]. Leveraging invalid samples to improve generative models' precision has also been explored in [8] in the field of engineering design.

Gene Clusters and Metabolite Molecules. Biosynthetic gene clusters (BGCs [37, 5]) are defined as a group of genes that are located in close genomic proximity to each other and are involved in a specific metabolic pathway. These clusters consist of genes that work together to produce secondary metabolite molecules. Secondary metabolite molecules (SMM [37, 5]) perform various

tasks for bacteria with functional domains, including antifungal, cytotoxic, and antibiotics. The MIBiG Database [30], is a well-established database for BGCs and SMMs, which at the moment has information on more than 2500 BGCs. Pfam [28] is a comprehensive database for protein domain families and is frequently used for identifying conserved domains within proteins coded by BGCs [18]. It is possible to represent the genes in a BGC as the Pfam domains present in the gene cluster. Utilizing Pfam representation for a BGC significantly reduces the memory size required, compared to using raw genes and gene products.

3 Method

The upper part of Fig. 2) illustrates a conventional method of using Pfam annotations of the genes in BGC as input to a generative LLM to predict the resulting SMM. The lower part introduces an additional step, called `invalid2valid`, where the model uses the predictions from the conventional approach as new input data for a generative LLM. Keeping the same train-test splits as for the conventional method to avoid clear biases. This extra mapping step allows the second LLM to focus on refining the predictions made by the first model, thereby improving the accuracy and reliability of the resulting SMM estimations. Specifically, the focus of the second LLM varies from task to task. In the case illustrated in Fig. 2, the second LLM focuses on enhancing chemical structure validity. Generally, by leveraging the initial predictions as a starting point, the `invalid2valid` approach aims to rectify any inaccuracies or inconsistencies, leading to enhanced performance. Our strategy for

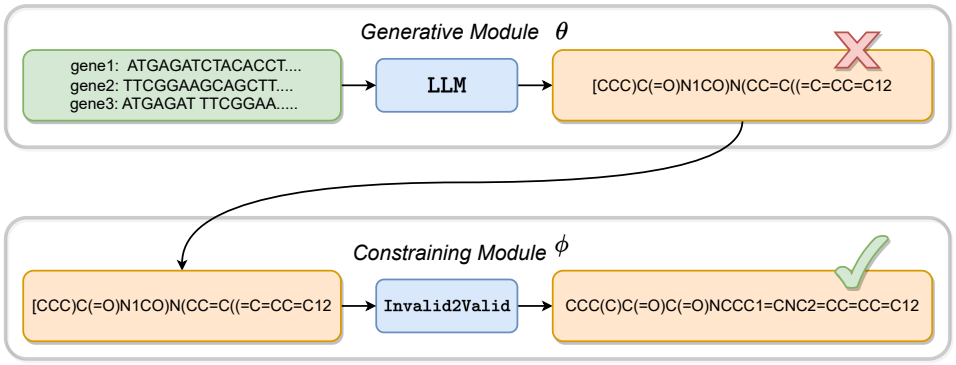


Figure 2: Pipeline Inference. On the upper part of the image, the raw gene sequences or gene representations (Pfam, ESM) from the BGC are used as input for a generic LLM. The target sequences are the corresponding SMMs for each BGC. After fine-tuning, the LLM will output some combination of symbols resembling a molecule, but without respecting the domain constraints and being invalid. We then leverage a second language model (bottom), performing `invalid2valid` to fix the mistakes of the first model. Doing so we greatly improve the precision of the generated output without reducing the diversity and generative capacities of the first model.

fine-tuning tasks involves a self-correction/refining post-hoc approach that is easy to understand and follow. This approach includes two LLMs, namely model A and model B, connected in a sequence. The output of model A is used as the input for model B, which then maps the prediction from model A to the final target y. Both models A and B have the same mapping target, y. This sequential mapping M can be described as:

$$M_{\mathrm{B}} : (M_{\mathrm{A}} : \mathbf{x} \rightarrow \mathbf{y}) \rightarrow \mathbf{y}, \tag{1}$$

where $\mathbf{x}$ is the original input feature vector; f_θ as the function representing the mapping from $\mathbf{x}$ to $\mathbf{y}$ in model A. f_ϕ as the function representing the mapping from $f_\theta(\mathbf{x})$ to $\mathbf{y}$ in model B. The objective functions to minimize the losses for each model would be:

$$\mathcal{L}_A = \mathtt{CE}(f_\theta(\mathbf{x}), \mathbf{y}), \quad \mathcal{L}_B = \mathtt{CE}(f_\phi(f_{\bar{\theta}}(\mathbf{x})), \mathbf{y}). \tag{2}$$

Both $\mathcal{L}_A$ and $\mathcal{L}_B$ are optimized during training to make $f_\theta(\mathbf{x})$ and $f_\phi(f_{\bar{\theta}}(\mathbf{x}))$ to minimize cross-entropy (CE) wrt to $\mathbf{y}$. Notice that the objectives are minimized sequentially, i.e. we train A, freeze θ, and then train B. In practice is, model A trained until the loss function plateaus; first, the outputs of model A saved and used to create a dataset for model B. The test and train splits are conserved to avoid biases in the learning. The practical approach is illustrated in Fig 2.

4 Experiments

Setup. We investigate three NLP tasks related to natural products, each framed as a string-to-string problem: predicting an SMM's molecular class (SMM2Class), its molecular activity (SMM2Activity), and generating an SMM from a BGC (BGC2SMM). In particular, SMM2Class is a standard classification task (with 8 unique classes) and can be solved with standard methods. SMM2Activity is, in principle, a classification task as well; however, the number of classes grows exponentially with the number of activities, making such a task difficult to model as a classification. Finally, BGC2SMM is the most complex and interesting task, where given some representation for a BGC, we want to generate the corresponding small molecule connected with the cluster.

Flan T5 [25] and Text+Chem T5 [4] serve as the baseline models for comparison with `invalid2valid` (denoted as w/ `i2v`) approach. For details on evaluation metrics see Appendix C.

The datasets used in this project pertain to the field of natural products and are sourced or derived from the MIBiG DB. Details on the general structure of these datasets can be found in the appendix in Table 9, while a summary is provided in Table 8.

Results. Qualitative results for predicting bio-synthetic classes from SMILES-formatted molecules are in Table 5. A language model maps molecules to classes, and a second model refines these predictions. The table shows baseline errors like misclassifying `Saccharide` as `Polycharide`. The `invalid2valid` approach corrects these, proving its utility in standard classification.

Table 1 presents a comprehensive overview of the accuracy, measured by the ROUGE-1 F1 score, achieved at different epochs for two main tasks: SMM to Activity prediction and BGC to SMM prediction. Three distinct models are evaluated for each task: Flan T5, Text+Chem T5, and `w/ i2v`. Notably, the `invalid2valid` model consistently outperforms the other models in both tasks at all epochs, demonstrating remarkable accuracy improvements with minimal finetuning. For the SMM to Activity task, the ROUGE-1 F1 score of the `i2v` model increases from 0.85 at epoch 1 to 0.94 at epoch 10. Similarly, for the BGC to SMM task, the `i2v` model achieves a score of 0.77 at epoch 1, which significantly rises to 0.89 at epoch 10. These findings highlight the effectiveness of incorporating the invalid2valid refinement in enhancing the accuracy of the models for both tasks.

Table 1: Accuracy as measured by ROUGE-1 F1 score on the test set at different epochs for the main tasks.

Epochs	SMM → Activity			BGC → SMM		
	Flan T5	Text+Chem T5	w/ `i2v`	Flan T5	Text+Chem T5	w/ `i2v`
1	0.04	0.44	**0.85**	0.53	0.68	**0.77**
2	0.37	0.44	**0.92**	0.58	0.67	**0.81**
10	0.41	0.43	**0.94**	0.61	0.68	**0.89**

BGC to SMM. Table 2 offers a comprehensive comparison of various encoding strategies and their impact on key metrics. Specifically, it assesses the performance of baselines encoding for the gene cluster, encoding with the Pfam approach both without and with the inclusion of the `invalid2valid` refinement. The metrics evaluated include ROUGE-1 F1 score, Character Error Rate (CER), and the BLEU score leveraging the references proposed in [24]. Notably, the introduction of the `invalid2valid` refinement yields substantial improvements across all metrics, underscoring its effectiveness in enhancing the encoding process. As we progress through different epochs, we observe a consistent trend of improvement, with the most significant gains occurring when `invalid2valid` is integrated. These findings highlight the invaluable role of `invalid2valid` in elevating the performance of encoding approaches, ultimately contributing to the overall success of the BGC2SMM task.

Table 2: Metrics for the task BGC2SMM. We compare baselines encoding for the gene cluster (gene product - DESCRIBE), encoding with the Pfam approach without and with the invalid to valid refinement. We see how adding the invalid to valid refinement greatly improves all the metrics. CER: Character Error Rate.

Epochs	Baseline			w/ Pfam			w/ Pfam w/ `invalid2valid`		
	ROUGE-1 F1 ↑	CER ↓	BLEU ↑	ROUGE-1 F1 ↑	CER ↓	BLEU ↑	ROUGE-1 F1 ↑	CER ↓	BLEU ↑
1	0.57	0.42	0.31	0.620	0.35	0.37	0.77	0.17	0.67
5	0.65	0.33	0.43	0.65	0.30	0.45	0.87	0.08	0.81
15	0.67	0.30	0.45	0.67	0.30	0.48	**0.90**	**0.05**	**0.87**

5 Conclusion

While Language Models have showcased their generative prowess, their adaptability to specialized tasks often falls short when accuracy is the primary goal. In tasks like chemical compound generation, precision takes precedence over response diversity, necessitating tailored approaches. To address this challenge, we have introduced `invalid2valid`, a versatile post-processing mechanism that demonstrates its efficacy across a range of domains and specialized applications. This mechanism represents a significant step forward in enhancing the precision of Language Models in input/output tasks. By bridging the gap between the adaptability of Language Models and the precision requirements of specialized tasks, `invalid2valid` opens up new possibilities for the seamless integration of language models into critical domains.

References

[1] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[2] J. Born and M. Manica. Regression transformer enables concurrent sequence regression and generation for molecular language modelling. *Nature Machine Intelligence*, 5(4):432–444, 2023.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] D. Christofidellis, G. Giannone, J. Born, O. Winther, T. Laino, and M. Manica. Unifying molecular and textual representations via multi-task language modelling. *arXiv preprint arXiv:2301.12586*, 2023.

[5] P. Cimermancic, M. H. Medema, J. Claesen, K. Kurita, L. C. Wieland Brown, K. Mavrommatis, ..., and P. R. Jensen. Insights into secondary metabolism from a global analysis of prokaryotic biosynthetic gene clusters. *Cell*, 158(2):412–421, 2014.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] C. Edwards, C. Zhai, and H. Ji. Text2mol: Cross-modal molecule retrieval with natural language queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 595–607, 2021.

[8] G. Giannone, L. Regenwetter, A. Srivastava, D. Gutfreund, and F. Ahmed. Learning from invalid data: On constraint satisfaction in generative models. *arXiv preprint arXiv:2306.15166*, 2023.

[9] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[10] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020.

[11] G. Landrum, P. Tosco, B. Kelley, Ric, D. Cosgrove, sriniker, gedeck, R. Vianello, N. Schneider, E. Kawashima, D. N, G. Jones, A. Dalke, B. Cole, M. Swain, S. Turk, A. Savelyev, A. Vaucher, M. Wójcikowski, ..., and strets123. rdkit/rdkit: 2023_03_3 (q1 2023) release (release_2023_03_3), 2023.

[12] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[13] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[14] R. Luo, L. Sun, Y. Xia, T. Qin, S. Zhang, H. Poon, and T.-Y. Liu. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6):bbac409, 2022.

[15] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, et al. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023.

[16] A. Madani, B. McCann, N. Naik, N. S. Keskar, N. Anand, R. R. Eguchi, P.-S. Huang, and R. Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

[17] M. Manica, J. Born, J. Cadow, D. Christofidellis, A. Dave, D. Clarke, Y. G. N. Teukam, G. Giannone, S. C. Hoffman, M. Buchan, V. Chenthamarakshan, T. Donovan, H. H. Hsu, F. Zipoli, O. Schilter, A. Kishimoto, L. Hamada, I. Padhi, K. Wehden, L. McHugh, A. Khrabrov, P. Das, S. Takeda, and J. R. Smith. Accelerating material design with the generative toolkit for scientific discovery. *npj Computational Materials*, 9(1):69, 2023.

[18] M. Mortazavi, M. Zarenezhad, S. Gholamzadeh, S. M. Alavian, M. Ghorbani, R. Dehghani, A. Malekpour, M. Meshkibaf, and A. Fakhrzad. Bioinformatic identification of rare codon clusters (rccs) in hbv genome and evaluation of rccs in proteins structure of hepatitis b virus. *Hepatitis Monthly*, 2016.

[19] N. Mündler, J. He, S. Jenko, and M. Vechev. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *arXiv preprint arXiv:2305.15852*, 2023.

[20] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

[21] L. Pan, M. Saxon, W. Xu, D. Nathani, X. Wang, and W. Y. Wang. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. 2023.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[24] M. Post. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*, 2018.

[25] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[26] J. Ross, B. Belgodere, V. Chenthamarakshan, I. Padhi, Y. Mroueh, and P. Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, 2022.

[27] W. Saunders, C. Yeh, J. Wu, S. Bills, L. Ouyang, J. Ward, and J. Leike. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*, 2022.

[28] E. L. Sonnhammer, S. R. Eddy, and R. Durbin. Pfam: A comprehensive database of protein domain families based on seed alignments. *Proteins: Structure, Function, and Bioinformatics*, 28(3):405–420, 1997.

[29] Z. Sun, Y. Shen, Q. Zhou, H. Zhang, Z. Chen, D. Cox, Y. Yang, and C. Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *arXiv preprint arXiv:2305.03047*, 2023.

[30] B. R. Terlouw, K. Blin, J. C. Navarro-Muñoz, N. E. Avalon, M. G. Chevrette, S. Egbert, S. Lee, D. Meijer, M. J. Recchia, Z. Reitz, J. van Santen, N. Selem-Mojica, T. Tørring, L. Zaroubi, M. Alanjary, G. Aleti, C. Aguilar, S. A. Al-Salihi, H. Augustijn, J. Avelar-Rivas, L. Avitia-Domínguez, F. Barona-Gómez, J. Bernaldo-Agüero, V. A. Bielinski, F. Biermann, T. Booth, V. Carrion Bravo, R. Castelo-Branco, F. Chagas, P. Cruz-Morales, C. Du, K. Duncan, A. Gavriilidou, D. Gayrard, K. Gutiérrez-García, K. Haslinger, E. N. Helfrich, J. J. van der Hooft, A. Jati, E. Kalkreuter, N. Kalyvas, K. Kang, S. Kautsar, W. Kim, A. Kunjapur, Y.-X. Li, G.-M. Lin, C. Loureiro, J. R. Louwen, N. L. Louwen, G. Lund, J. Parra, B. Philmus, B. Pourmohsenin, L. U. Pronk, A. Rego, D. Rex, S. Robinson, L. Rosas-Becerra, E. Roxborough, M. Schorn, D. Scobie, K. Singh, N. Sokolova, X. Tang, D. Udwary, A. Vigneshwari, K. Vind, S. J. M. Vromans, V. Waschulin, S. Williams, J. Winter, T. Witte, H. Xie, D. Yang, J. Yu, M. Zdouc, Z. Zhong, J. Collemare, R. Linington, T. Weber, and M. Medema. MIBiG 3.0: a community-driven effort to annotate experimentally validated biosynthetic gene clusters. *Nucleic Acids Research*, 51(D1):D603–D610, 2023.

[31] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting. Large language models in medicine. *Nature medicine*, pages 1–11, 2023.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[33] H. Wang, T. Fu, Y. Du, W. Gao, K. Huang, Z. Liu, P. Chandak, S. Liu, P. Van Katwyk, A. Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

[34] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, and D. Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

[35] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019.

[36] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[37] N. Ziemert, M. Alanjary, and T. Weber. The evolution of genome mining in microbes–a review. *Natural product reports*, 33(8):988–1005, 2016.

A Additional Experiments

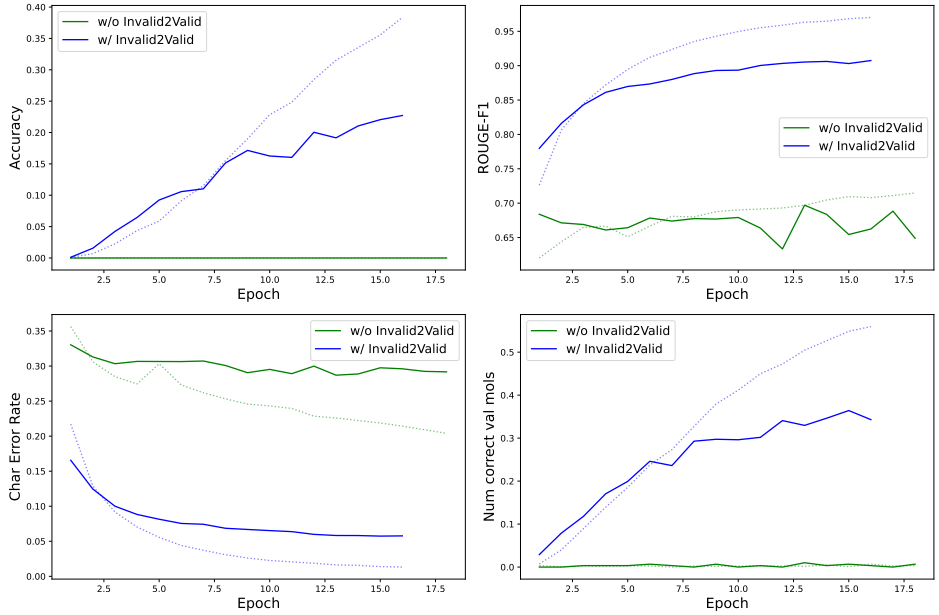


Figure 3: The figure compares the performance of using Pfam (Green lines) and 'invalid2valid' (Blue lines) in the task of predicting SMMs. Dotted lines represent training data, and solid lines represent testing data. Four evaluation metrics are shown: Accuracy (exact matching), Rouge F1, Char error rate, and rate of valid molecules. A noticeable performance improvement across all four metrics is evident when using the 'invalid2valid' model as opposed to relying solely on Pfam annotations.

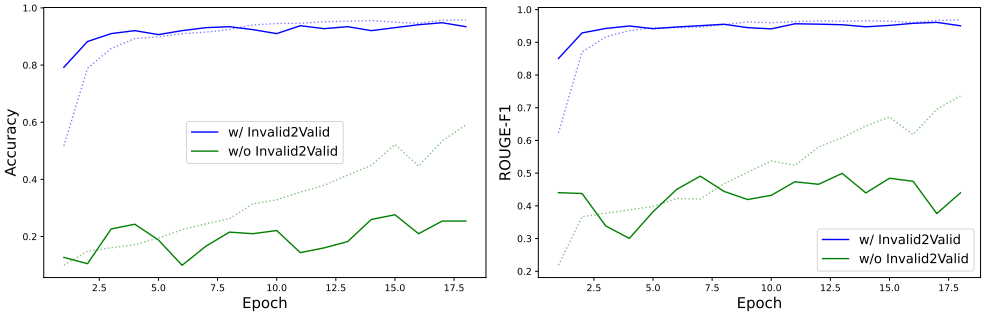


Figure 4: The figure displays a performance comparison between using SMILES (Green lines) and 'invalid2valid' (Blue lines) for predicting the molecular activity of SMMs. Dotted lines represent the training data, while solid lines represent testing data. The evaluation is based on two metrics: Accuracy (exact matching) and Rouge F1. A significant performance improvement is observed when applying the 'invalid2valid' approach.

B Qualitative Results

Table 3: Qualitative examples of the task of predicting SMM given BGC for baselines and w/ and w/o Invalid2Valid.

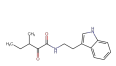
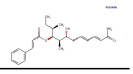
Method	SMILES	Chem. struct.
Baseline	CCC(C)C(=O)N1=O)N((=CCNC2=C=CC=C12	
w/ Pfam	[CCC)C(=O)N1CO)N(CC=C((=C=CC=C12	
w/ Pfam w/ I2V (ours)	CCC(C)C(=O)C(=O)NCCC1=CNC2=CC=CC=C12	
Target	CCC(C)C(=O)C(=O)NCCC1=CNC2=CC=CC=C12	
Baseline	C[C==CC)1ccccccccc(ccccO2cnH](cO-])C)CC((C(C)C)	
w/ Pfam	[C==C/C)1C2ccccccc2ccccO2cCH]([O-])C)CC(C()C)	
w/ Pfam w/ I2V (ours)	C/C(=CCO)1cccc2cccc(cccccc13[C+]2[O-])O)/Cc=C(C)C	
Target	C/C(=CCOc1cccc2c1nc1cccc(c1[n+]2[O-])O)/CCC=C(C)C	
Baseline	CCC1C@HH]1C)[C@H]1C(=O)[C=CC==CC=CC=C1)CC@@](C)CC@H](C)C(C=CC=CC==)=O	
w/ Pfam	CC(C@HH]1C)[C@H]1O(=O)[C=CC==C=CC=C1)CC@H](O)[C@H](O)[(C=CC=CC==)=O	
w/ Pfam w/ I2V (ours)	CC[C@@H](C)[C@H](OC(=O)/=CC1=CC=CC=C1)[C@H](C)[C@H](O)C=CC=C/C=N)=O	
Target	CC[C@@H](C)[C@H](OC(=O)C=CC1=CC=CC=C1)[C@H](C)[C@H](O)CC=CC=C/C(N)=O	

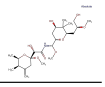
Table 4: Qualitative examples from the test set of the task of predicting molecular activities given SMM.

Label	Input to baseline	Baseline predictions	w/ I2V (ours)
antibacterial;antifungal;inhibitor	CCC(C)CC(C)C(=O)C1=C ...	antibacterial;antifungal;cytohibitor	antibacterial;antifungal;inhibitor
siderophore	CCC(C)C1=CN=C(C(=O) ...	siderophore	siderophore
antibacterial;signalling	NC(CC1=CNC2=C1C=CC=C2)C ...	antibacterial;signalling	antibacterial;signalling
surfactant	CC(C)CCCCCC(=O)N[C@H] ...	antifactant	surfactant

Table 5: Qualitative examples from the test set of the task of predicting molecular classes given SMM.

Label	Inputs to baseline	Baseline predictions	w/ I2V (ours)
Saccharide	OC[C@H]1O[C@@H](N2C=NC3=C2NC=NC[C@H]3O)[C@H](O)[C@@H]1O	Polycharide	Saccharide
Polyketide	CO/[N+](=O)=C/C(=O)N/C=C/C(=O)O	Nketide	Polyketide
Polyketide	CC(=O)N[C@@H](CC1=CC=CC=C1)C2=CC(=CC(=O)O2)OC	Riketide	Polyketide
RiPP	CCC1CCCCC(=O)C2=C(O1)C=C(C=C2CC(=O)OCC)O	OtherPP	RiPP
NRP	COC1=CC(/C=N/O)=NC(=C1)C1=NC=CC=C1	PolyRP	NRP

Table 6: Qualitative examples of the task of predicting SMM given BGC for baselines and w/ and w/o Invalid2Valid.

Method	SMILES	Chem. struct.
Input to baseline	nonribosomal peptide synthetase, BudA_nonribosomal peptide synthetase, ...	
Baseline	CCC(C)C(=O)N1=O)N((=CCNC2=C=CC=C12	
Input to Pfam	PF05222.17_PF01262.23_PF12769.9_PF022 ...	
w/ Pfam	[CCC)C(=O)N1CO)N(CC=C((=C=CC=C12	
w/ Pfam w/ I2V (ours)	CCC(C)C(=O)C(=O)NCCC1=CNC2=CC=CC=C12	
Target	CCC(C)C(=O)C(=O)NCCC1=CNC2=CC=CC=C12	
Input to baseline	bacteriophage integrase_hypothetical protein_putative thioesterase_putative ...	
Baseline	CCC(CCCCCCCCCCC=)C@HH](CC(N)=O)C(N[=O)CO	
Input to Pfam	PF13356.8_PF00589.24_PF08020.13_PF00975.22_PF00144.26 ...	
w/ Pfam	CCC1CCCCCCCCCCC)C@HH](O)=)NO)=(O)=O)=O)	
w/ Pfam w/ I2V (ours)	CCCCCCCCCCC((N[C@@H](C=O)=O)CC(N)=O)=O	
Target	CCCCCCCCCCCCC(N[C@@H](C(O)=O)CC(N)=O)=O	
Input to baseline	PilN domain-containing protein ...	
Baseline	C[C@H]1[C@@]([[C@H2[(=C(([C@@H](C)=O)O[C@H](CC@@H]([[C@@]([(=C@@](C2)[[C@@H](C)O)OC)C)O)O)O)OO	
Input to Pfam	PF11288.10_PF07015.13_PF05128.14_PF05128.14_PF01145.27 ...	
w/ Pfam	[C@H]1[C@H](C]C@H2C)=O)(CC@HH](O)=O)O[C@H](CC@HH]([[C@H]([[=C@H](O2)C)C@HH](C)O)OC)C)O)C)OC)OC)OC)	
w/ Pfam w/ I2V (ours)	C[C@H]1[C@H](O[C@](CC1=C)([C@@H](C(=O)N[C@H]([C@@H]2C[C@H](C([C@H](O2)C[C@@H](CO)OC)(C)C)O)OC)O)OC)C	
Target	C[C@H]1[C@H](O[C@](CC1=C)([C@@H](C(=O)N[C@H]([C@@H]2C[C@H](C([C@H](O2)C[C@@H](CO)OC)(C)C)O)OC)O)OC)C	

C Evaluation metrics and software

For the evaluation matrices was, the following used: For accuracy was Scikit-learn used [23]. For char error rate, Rouge and sacreBleu were PyTorch torchmetrics used [22]. For chemical structure validity, was rdkit used [11] PyTorch was used to build models in [22]. Pre-trained models was downloaded from HuggingFace [35].

C.1 Metrics

Accuracy. Refers to exact matching.

Char Error Rate. Refers to the percentage of characters that were predicted incorrectly. A lower value signifies better performance, with a CharErrorRate of 0 indicating flawless predictions.

ROUGE-1. Refers to the overlap of unigrams between the predicted molecule and target molecule. We compute the F1 score of such metric to balance precision and recall.

sacreBLEU. Computes the BLEU score by averaging the accuracy of phrases of different lengths in the machine translation, adjusting for over-counting and sentence length. It ensures consistent scoring by using fixed settings and standardized reference data [24].

Chem.MolFromSmiles. Checks the validity of a molecule represented in SMILES notation. This function attempts to create a molecular object based on the input SMILES string. Common SMILES symbols are illustrated in table 7

C.2 SMILES

Table 7: Most common SMILES symbols and meanings

Symbol	Description
C / c	Carbon atom / aromatic carbon atom
O / o	Oxygen atom / aromatic oxygen atom
N / n	Nitrogen atom / aromatic nitrogen atom
()	Encloses branches
=	Double bond
#	Triple bond
\/	Single or double bond (cis or trans)
@	Chirality (clockwise)
@@	Chirality (anticlockwise)
[ ]	Atomic properties (charge, isotopic mass, etc.)

D Datasets

Table 8: Summary of Datasets. In the row corresponding to the Pfam dataset, the numbers in parentheses indicate the subset used for training and evaluation.

Dataset	Train Set	Test Set	Input	Target
SMILES-biosyn	1964	492	SMM	Class
SMILES-activity	713	179	SMM	Function
Pfam	1191 (845)	298 (211)	BGC	SMM
invalid2valid	2535	633	SMM	SMM

Table 9: The general structure of all the datasets

Symbol	Description
`<Task: >`	The task or problem to be solved.
`<Input_text>`	Input text for the task.
`<target_string>`	The target string or expected output.

E How to train `invalid2valid`

F Hyper-parameters

For all the tasks was, the same hyper-parameters used; Learning rate: 7e-4, batch size: 6, and weight decay: 0.05.

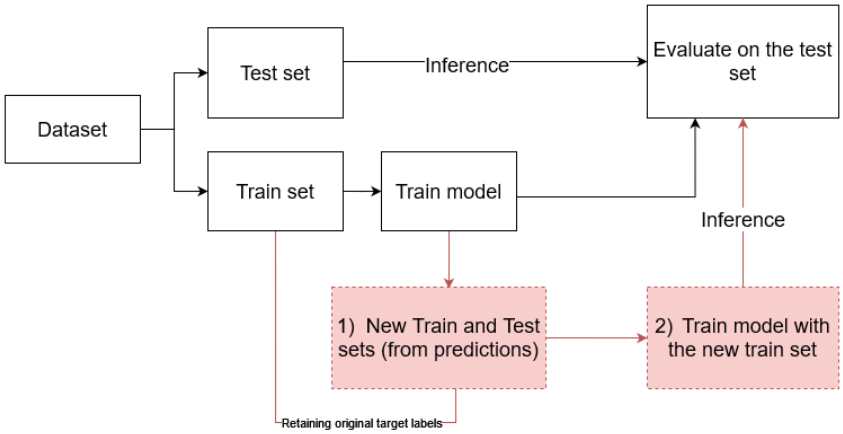


Figure 5: The figure distinguishes the invalid2valid method, represented by red squares, from a standard machine learning approach, shown in white squares. 'Invalid2valid' augments the standard machine learning approach with two steps. 1): saving post-convergence predictions for both train and test sets and then using these to form new train and test sets while keeping original target labels intact. 2): Fine-tuning/training a new LLM on the new train set and assessing its performance on a (semi-)original test set.

Table 10: Relevant Hyperparameters for baselines and 'Invalid2Valid

	SMM2Class	SMM2Activity	Pfam	invalid2valid
Batch size	6	6	6	6
Architecture	T5	T5	T5	T5
Iterations/epochs	18	18	18	18
Learning rate	$7e^{-4}$	$7e^{-4}$	$7e^{-4}$	$7e^{-4}$
Optimizer	AdamW	AdamW	AdamW	AdamW