

Improving the Efficiency and the Validity of Molecular Transformers

Leone Bacciu

*Department of Environmental Sciences,
Informatics and Statistics
Ca' Foscari University of Venice
Venice, Italy
leone.bacciu@unive.it*

Matteo Grazioso

*Department of Environmental Sciences,
Informatics and Statistics
Ca' Foscari University of Venice
Venice, Italy
matteo.grazioso@unive.it*

Silvia Multari

*Department of Molecular Sciences
and Nanosystems
Ca' Foscari University of Venice
Venice, Italy
silvia.multari@unive.it*

Francesca Grisoni

*Department of Biomedical Engineering
Eindhoven University of Technology
Eindhoven, Netherlands
f.grisoni@tue.nl*

Angelica Mazzolari

*Department of Pharmaceutical Sciences
University of Milan
Milan, Italy
angelica.mazzolari@unimi.it*

Marco S. Nobile

*Department of Environmental Sciences,
Informatics and Statistics
Ca' Foscari University of Venice
Venice, Italy
marco.nobile@unive.it*

Abstract—Since their advent, Transformer models have been applied across a wide range of fields, including cheminformatics. In this context, drug discovery has benefited from using Molecular Transformers by leveraging diverse string representations of molecules, such as the Simplified Molecular Input Line Entry Systems (SMILES), for a variety of tasks. In this study, we present a model focused on the optimization of a formerly developed Molecular Transformer specifically dedicated to metabolism prediction. Metabolism refers to all the biotransformations a drug undergoes once inside the human body, directly influencing its therapeutic effect and potential toxicity, and therefore represents a key topic in medicinal chemistry. Framing molecular transformation prediction as a sequence-to-sequence translation task has shown promise, but suffers from limitations such as low validity of generated molecules and high computational cost. To address this limitation, we here propose an optimized model that integrates pre-training, transfer learning, and fine-tuning techniques, already improving validity and reducing computation time. Finally, by separating the metabolism prediction task from the SMILES syntax learning, we ensure broader applicability of the proposed model across diverse datasets and a variety of SMILES-based tasks beyond metabolic transformations, expanding its potential utility.

Index Terms—Metabolism, Biochemical reactions, Drug discovery, Transformers

I. INTRODUCTION

Drug discovery and development is the process of identifying, designing, and bringing a new therapeutic molecule to the market. It encompasses several stages, *i.e.* from target identification to clinical trials, and despite the recent advancements in both understanding biological systems and developing innovative technologies, it remains a complex and expensive endeavor [1].

In this context, metabolism plays a key role, as it refers to the set of chemical reactions that xenobiotics, such as drugs, undergo once inside the human body. Drug metabolism is mediated mainly by liver enzymes and occurs in two main phases: (1) phase I metabolism involves oxidation, reduction,

and hydrolysis reactions and is primarily catalyzed by the cytochrome P450 (CYP450) enzyme family [2], while (2) phase II metabolism includes conjugation reactions mediated by transferases, which increase metabolite hydrophilicity and promote their excretion. Both phases involve a structural modification of the parent compounds and therefore influence the pharmacokinetics and pharmacodynamics of drugs. [3] Hence, the metabolic profile of a drug candidate strictly affects its bioavailability – defined as the rate and extent a substance becomes completely available to its intended biological destinations [4] – and can be responsible for (1) drug-drug interactions, (2) the therapeutic effect of a molecule and its efficacy profile, and (3) the potential toxicity of a drug and its safety profile [5] [6].

Understanding metabolism becomes crucial in distinguishing promising drug candidates from molecules with a non-suitable metabolic profile, and in turn speed up the first stages of the drug discovery process. [7]

Computational methods for metabolism prediction are already available [8], but they are mainly rule-based models designed to predict the sets of atoms where the reaction would occur, known as sites of metabolism (SoMs) [9] [10]. Although these models provide valuable insights into where a reaction might take place and therefore what transformations could occur, they suffer from significant limitations. Specifically, they have a poor ability to generalize, and they are often biased toward CYP450-mediated reactions because they are more abundant in real-life scenarios [11]. As a result, these models often fail to predict outcomes beyond their predefined rules, not encompassing the full spectrum of metabolic processes, and this underscores the need for approaches capable of broader and more accurate metabolism predictions [12].

Deep Learning (DL)-based solutions can learn directly from data and build a rule-free end-to-end model. In particular, the Simplified Molecular Input Line Entry Systems (SMILES)

[13] notation can be used to represent 2D molecules with 1D strings, in order to frame the metabolism prediction task as a machine translation problem. Specifically, a Transformer architecture [14], known for its powerful sequence-to-sequence learning capabilities, can be used to model metabolic transformations directly from SMILES representations. Figure 1 represents the basic architecture of a molecular transformer applied to metabolism prediction.

One key innovation in this line of research was a dual-input strategy: the structural information coming from the SMILES string representing the substrate was integrated with the information about the reaction the molecule was undergoing, referred to as the Reaction Class (RC) [15]. As a matter of fact, providing contextual data about the reaction type enhanced the ability of the model to accurately predict metabolic transformations. Regardless of its remarkable performance, such a model still has room for improvement. Specifically, the main challenges are two-fold: on the one hand, the high computational cost and time required for training limit the scalability of the approach; on the other hand, the Transformer might generate outputs that do not correspond to chemically valid molecules (*i.e.*, not following a correct SMILES syntax), hindering the robustness of the model.

In this work we propose an optimized Molecular Transformer intended to overcome the aforementioned challenges. To enhance the efficiency of the model and increase the validity of the predicted metabolites, we divide the learning process of the model in two tasks: (1) a pre-training phase aimed at learning chemical rules and the SMILES syntax, and (2) a main training phase focused on teaching the model how to predict the metabolic transformations of substrates.

The paper is structured as follows. In Section II, we present an overview of prior work on xenobiotic metabolism prediction, focusing on both ligand-based and structure-based computational strategies. In Section III, we introduce the methods used in this work: the SMILES notation, along with the datasets used for this study; the Molecular Transformer neural architecture; the transfer learning approach that we adopted. In Section IV we detail the training approach, which exploits a model pre-trained on a different task, fine-tuned on metabolism prediction. The pre-trained model is described in Section V. We tested three different configurations, which are listed in Section VI, together with the hyperparameters used for each one. The results and comments of these tests are presented in Section VII. Finally, we conclude the manuscript with some future developments in Section VIII.

II. RELATED WORKS

A wide variety of computational strategies for the prediction of xenobiotic metabolism has already been explored in the literature [8]. Among these methods, a primary distinction lies in the perspective they adopt: ligand-based, focusing on the analysis of the molecular structures of compounds with known metabolic profiles, versus structure-based, typically simulating how and why metabolism occurs at specific sites. Both paradigms encompass a several kinds of models, ranging

from traditional rule-based systems to more advanced machine learning (ML) and deep learning (DL) techniques. The vast majority of these solutions primarily aim to predict SoMs to guide compound optimization, or focus on ligand relation to the metabolizing enzymes, as estimating the binding affinity or the likelihood of the enzymes induction or inhibition [16]. They all give valuable insights on where and how the metabolic transformations occur, but they often take into consideration only a single aspect of the process. For example, reactivity-based approaches, such as CypScore [17], predict sites of metabolism by computing local physico-chemical descriptors and assigning a reactivity score to atoms or functional groups [18]. Other approaches include fingerprint-based data mining, as Metaprint2D and Bioclipse [19], that consider structural similarity to known substrates, and classical docking approaches (*e.g.*, AutoDock VINA [20]).

Since metabolism is driven by a complex interplay of multiple factors, this in turn limits the real-world applicability of the models. Combined approaches represent a first step forward in the comprehensive understanding of metabolism, as they integrate multiple sources of information to offer a more realistic and accurate representation of biotransformations. For example, MetaSite [21] and GLORYx [10] both perform the prediction of the SoM and the generation of likely metabolites, a complex task addressed by relatively fewer models, despite being equally relevant. A fundamental challenge lies in the complexity of the underlying biological processes: rule-based approaches fail in grasping all the rules governing metabolic transformations, or effectively combining the multitude of contributing factors. Therefore growing interest in data-driven, end-to-end learning approaches has risen, as they can automatically learn intricate patterns directly from raw data [22].

Among these models, particular interest has been gained by Transformer-based architectures [14], for their ability to capture long-range dependencies within molecular representations. MetaTrans [12], BioTransformer [23], Metapredictor [24] are notable examples. They leverage SMILES-based molecular representations and Transformer architectures to predict the outcomes of metabolic reactions. They frame the problem as a sequence-to-sequence translation task to directly predict metabolic products from the substrate SMILES, and by integrating rule-based knowledge with machine learning, they improve coverage and interpretability.

Despite their promising performance, current models still face challenges related to chemical validity of outputs, generalizability across different enzymatic classes, and computational efficiency, motivating the need for further optimization and innovation in this space. The goal of this work is to propose a new approach which can, in principle, be applied to similar transformer-based methodologies. Since our method and existing approaches are designed to address different tasks, comparing their performances is not straightforward, and may not offer a meaningful point of comparison. As a result, we decided to focus our evaluation on internal metrics, such as validity and correctness, which are better suited to capture the specific characteristics of our model's outputs.

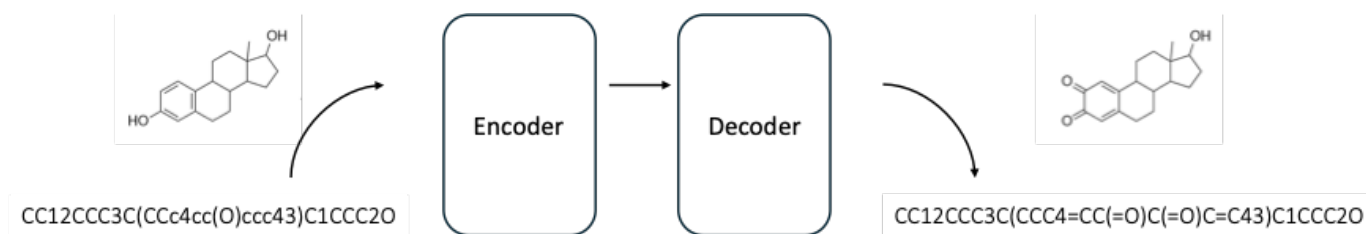


Fig. 1. Basic architecture of a molecular transformer applied to metabolism prediction.

III. METHODS

A. SMILES

The Simplified Molecular Input Line Entry System (SMILES) is a computer-friendly textual formalism used to represent molecules and chemical structures [13]. Technically, SMILES are strings composed of a sequence of characters that represent atoms and bonds, with specific rules and syntax to define branches and ring closures. In a SMILES sequence, each non-hydrogen atom is denoted by its chemical symbol. Bonds are represented by specific characters such as “-” (single bonds), “=” (double bonds), and “#” (triple bonds). Branching is denoted by a pair of parentheses, while ring closures are denoted by numbers.

SMILES are quite effective and widespread for the encoding and manipulation of small molecules due to their simplicity and compactness. In particular, SMILES are widely adopted in drug discovery and cheminformatics for database searches, molecular modeling, and structure-activity relationship studies, providing a versatile and efficient tool for handling molecular data.

An interesting characteristic of SMILES strings is that, depending on the atom chosen as the starting “reading” point, the same molecule can be represented by different strings. These different representations can be mapped back to a unique standard form, known as the canonical SMILES.

The similarity (or in this case, equality) of the two molecules can be determined by fingerprint-based calculations. Specifically, we chose to use the Tanimoto coefficient [25] computed on circular topological fingerprints called Extended Connectivity Fingerprint (ECFPs) [26]. ECFPs are binary strings where 1 bit represents the presence of a substructure, and 0 its absence. The Tanimoto similarity index is computed by taking the intersection of the two sets divided by the sum of their sizes. It goes from 0 to 1, where 0 indicates no similarity and 1 indicates perfect similarity.

Since SMILES strings can encode small molecules, in this work they are used to represent both substrates to be processed and the produced metabolites, turning the problem of predicting metabolism into a sequence-to-sequence translation problem. The translation is tackled using a Machine Learning approach, as described in the next section.

B. Dataset

For the purpose of the pre-training process, focused on learning molecular representations rather than performing

metabolism-related tasks, it was not necessary to use a database containing metabolic information. We, therefore, leveraged the ZINC20 database [27], a collection of more than 880 million small molecules represented as SMILES strings. More specifically, we used 400 000 random-sampled molecules derived from the drug-like subset.

The model was then trained for metabolism prediction, for which only a limited number of datasets are publicly available. To address this, we used the combination of MetaQSAR [28] and MetaTree [29] described in [15]. The two manually curated, experts-checked databases are specifically dedicated to collecting information on small molecule metabolism, leveraging the SMILES representation, and are focused on both phase I and phase II metabolism. More in detail, MetaQSAR is a wide source of reaction annotations including also incomplete metabolic studies focused on single classes of metabolic reactions, and does not explore the entire metabolic tree of complete studies, whereas MetaTree is a collection of complete metabolic trees only, helping the extrapolation of suitable datasets of true negative data, meaning substrates/sites of metabolites not undergoing the query reaction. This can help overcome the high rate of false negatives that typically affects model performance in this field.

The combined dataset was still relatively small, so it was expanded using data augmentation techniques leveraging non-canonical SMILES, and resulted in approximately 100 000 instances, each consisting of a substrate paired with an ID corresponding to a reaction class, and the resulting metabolite when the reaction occurs, or the substrate itself for reactions not taking place on that substrate.

C. Molecular Transformers

The Transformer model, as introduced by Vaswani *et al.* [14], leverages an encoder-decoder architecture to process sequential data more efficiently and effectively, addressing key limitations of Recurrent Neural Networks (RNNs), such as sequential processing, vanishing gradients, and difficulty in capturing long-range dependencies. Both the Encoder and Decoder are composed of multiple layers. Each Encoder layer includes a multi-head self-attention mechanism and a position-wise feed-forward network with two linear transformations separated by a ReLU activation. The Decoder closely resembles that of the Encoder but adds an additional multi-head attention layer that focuses on the Encoder’s output, facilitating the integration of encoded information during decoding, and

outputs a vector that is transformed by a Softmax function into a distribution of the most probable tokens.

In the case of Molecular Transformers (MT), tokens are represented by all the characters composing the SMILES strings: the model predicts the next token choosing among the most probable ones, based on the Temperature parameter. A higher sampling temperature makes the process more random, while a lower sampling temperature makes the process more conservative, enabling the model to predict multiple metabolites for a single substrate. Unlike conventional Transformer architectures, the MT employed in this study incorporates the Reaction Class as an additional input feature [15].

D. Transfer Learning

Transfer learning enables the knowledge learned from one task to be used to improve performance in related tasks. It is different from the traditional methods that require training models from scratch for a particular task. The pre-trained models are used in order to decrease the reliance on large labeled datasets and speed up the training process [30]. The most important advantage of transfer learning is the capability to exploit representations that were learned from diverse and extensive source. Pre-trained representations contain more fundamental patterns and relationships in data that can be easily adapted to new, unseen target tasks [31]. With the transferred features that were acquired by models, higher performance can be achieved together with improved generalization of the target task after minimal supplementary training. For Transformer architectures, freezing the decoder has been shown to improve performance and generalization capabilities [32]. Moreover, using multiple (*e.g.*, adaptive) learning rates can improve overall performance by maximizing the learning potential of each component of a Transformer model [33].

IV. TRAINING METHODOLOGIES

The MT proposed in this work is the improved version of the model described in [15], which retains the original updated Transformer architecture while exploiting an optimized training procedure, introducing key innovations thoroughly described in the next section. The training strategy adopted in this study is detailed as follows.

- 1) *Training Pre-trained Model*: We train an MT model to apply a syntactic transformation to SMILES strings, which requires an understanding of the SMILES syntax and the underlying chemical structure of the represented molecule. Although this pre-training task operates within the same broad domain of molecular processing, it can be considered orthogonal to the specific task of metabolic reactions. The details of this pre-training phase will be elaborated in Section V.
- 2) *Transfer Learning with Specialized Pre-training*: Instead of training the MT from scratch, we implement transfer learning by initializing it using the pre-trained model weights [30]. Carefully applied pre-training can have a significant impact on both the training effectiveness

and the performance of a given model by leveraging previously acquired knowledge.

- 3) *Fine-tuning on Metabolic Transformation Tasks*: We fine-tune the transferred MT model, by training it on the primary task of metabolic transformation. To facilitate effective adaptation, reduce training time, and improve performance, during this phase the decoder component of the MT is frozen; that is, its weights are not updated during back-propagation.
- 4) *Final Fine-tuning without Freezing*: To further optimize performance, we conduct additional fine-tuning over a few epochs using a smaller learning rate to avoid losing the patterns learned with the frozen decoder. In this stage, the decoder is unfrozen, allowing all network weights to be updated. The decreased learning rate ensures that the updates occur more gradually, promoting stable convergence and fine-grained adjustments to the model parameters.

By freezing the layers of the MT, only 4.5 million out of a total 13.5 million parameters are updated in the main training phase, speeding up the training epochs and reducing computational overhead and training time. Layer freezing followed by a second fine-tuning without freezing allows us to take advantage of the patterns learned during the pre-training phase, avoiding catastrophic forgetting [34] and overfitting [35]. In second the fine-tuning phase, all model weights are unfrozen and updated; this phase takes up a limited fraction of the training time and allows most parts of the process to be performed with the quickest epoch duration. To evaluate the effectiveness of this training strategy, we conducted benchmark assessments. Detailed results and comparative analyses are discussed in Section VII.

Our pre-training methodology is designed to operate independently from the metabolic transformation task, ensuring that the decoder remains capable of generating high-quality SMILES strings with only minimal adjustments. Besides this, it yields an encoder that is highly effective in laying a strong foundation for modeling different metabolic pathways by leveraging its deep understanding of SMILES representations.

V. PRE-TRAINED MODEL

In this section, we detail the pre-trained model and the training process. The task of the model is the “canonicalization” of the generic SMILES string. Specifically, given a generic SMILES string – where the syntax is correct but the atom order is arbitrary – the model generates the canonical SMILES string of the corresponding molecule, ensuring uniqueness for each molecular structure. This task forces the model to learn a latent representation that encapsulates the molecule’s overall geometry, independent of its sequential SMILES notation. This allows the model to capture much more intrinsic geometric properties rather than any specific ordering of atoms in the SMILES string, which increases the preciseness and consistency of canonical forms. In contrast, simpler pre-training tasks such as auto-encoding tend to favor memorization – especially when model capacity greatly exceeds the amount

of training data – leading the network to reproduce training examples rather than learn broadly applicable, generalizable features, required for this task [36].

To ensure both diversity and comprehensiveness in the pre-training process, we exploited the ZINC20 database [27]. The ZINC datasets contain only SMILES strings, to convert it to a labeled dataset we used the library RDKit [37]. Each molecule was first parsed from its SMILES string into a molecular graph representation. Subsequently, we employed the function `MolToSmiles`¹ to regenerate SMILES strings in two different ways:

- 1) *Input*: the flag `doRandom=True` is set, which randomizes the traversal of the molecule graph, so we can generate a random SMILES string.
- 2) *Target*: the flag `canonical=True` is set, to get the unique canonical SMILES representation of the molecule.

This dataset was then used to train an MT model with two additional enhancements:

- [i] First, to improve generalization, we implemented data augmentation by generating three randomized SMILES inputs for each molecule. This expanded the dataset to 1.2 million entries, thereby, increasing the diversity and robustness of the data.
- [ii] Second, we addressed an architectural feature inherited from [15]: the inclusion of an additional input parameter, “Reaction Class.” As a matter of fact, this parameter is irrelevant to the SMILES canonicalization pre-training task. To mitigate its impact, we assigned a random value, effectively signaling that this input should be disregarded. Specifically, the random values were sampled uniformly from the range of valid reaction classes to maintain consistency with the model’s expectations. We avoided setting the “Reaction Class” to zero to prevent potential issues such as exploding gradients.

Most of the training configuration is similar to [15], except for early stopping patience which was set to 4 and minimum delta 10^{-4} .

VI. EXPERIMENTAL SETUP

To evaluate the effectiveness of our pre-training strategy, we performed several experiments comparing the performance of three training configurations of the MT in different aspects, *e.g.*, the impact of pre-training and fine-tuning on model performances and generalization capabilities. The configurations are carefully designed to separate the effects: weight initialization, freezing the decoder, and learning rate adjustment. The hyperparameters not specified are taken from [15]. Specific configuration details are as follows:

- (a) *Baseline*: an MT with randomly initialized weights is trained with early stopping, as described in [15].
- (b) *Pre-Trained (Frozen Fine-tune)*: an MT whose weights are initially taken from the pre-trained model. The de-

coder part of the network is frozen. The model is trained with early stopping.

- (c) *Pre-Trained (Full Fine-tune)*: the model trained in (b), further trained for some extra epochs, with early stopping. The decoder is not frozen. The learning rate is set to an order of magnitude smaller than the previous phases.

TABLE I
TRAINING CONFIGURATIONS OF THE MOLECULAR TRANSFORMER.

	Baseline	Pre-Trained Frozen Fine-Tune	Pre-Trained Full Fine-Tune
	(a)	(b)	(c)
Weight Initialization	Random	Pre-trained	From (b)
Decoder Status	Unfrozen	Frozen	Unfrozen
Learning Rate	10^{-4}	10^{-4}	10^{-5}

As illustrated in Table I, each configuration varies in weight initialization, decoder status, and learning rate settings. This comparison allows for an evaluation of how pre-training and fine-tuning strategies influence the MT’s performance and generalization capabilities.

Early stopping is employed in order to reduce overfitting, by stopping the training process if the model’s performance on the validation set ceased to improve. Specifically, training was stopped if the validation loss did not decrease by at least 10^{-3} over 3 epochs. Upon triggering early stopping, the best model weights, from 3 epochs prior, were restored. The learning rate is chosen to be 10^{-4} for (a) and (b) and 10^{-5} for (c). Adam was used as the optimizer, with its default configuration. The dataset was randomly split into training (80%) and validation (20%), ensuring that no pair from the training set appeared in the validation set, and the same split was used for all configurations. The training metrics and prediction performance of these models are described in the following section.

Two configurations were excluded from this comparison: the baseline model with a frozen decoder and the pre-trained model without initial parameter freezing. The frozen baseline model was omitted because it failed to achieve convergence within 50 epochs (exceeding 3600 seconds), as indicated by the non-increasing training accuracy and a training loss that did not decrease below 1.2. The pre-trained model without freezing, instead, rapidly converged to a suboptimal local minimum, consistently outputting the canonicalized input molecule. This behavior indicates that the model was merely replicating the pre-trained task rather than learning the metabolization process. As a result, the model exhibited zero correctness for molecules that required metabolization to different structures.

VII. RESULTS

In order to investigate the effectiveness of both pre-training and fine-tuning, we need to analyze both the training history and the quality of MT predictions. All experiments that follow were performed on a workstation equipped with an Intel(R)

¹<https://www.rdkit.org/docs/source/rdkit.Chem.rdmolfiles.html#rdkit.Chem.rdmolfiles.MolToSmiles>

Xeon(R) Gold 6348 CPU @ 2.60GHz processor, 200GB of RAM, and an NVIDIA A40 GPU. The main software packages used are: Python (3.11), TensorFlow (2.14.0), RD-Kit (2023.09.1), numpy (1.26.4), pandas (2.1.2), scikit-learn (1.6.0).

For each input molecule, a top-6 set of candidate metabolized molecules is generated: one is computed deterministically by always selecting the next token with the highest output probability, while the other five are obtained using temperature sampling with a temperature parameter of 0.8.

As an initial evaluation, we compared the performance of the three models by calculating two quality metrics. Table II summarizes the statistics related to validity and correctness for the three configurations.

- *Validity* refers to the percentage of inputs for which at least one SMILES string generated by the MT follows the correct SMILES syntax rules, and can thus be interpreted as an actual molecule. This metric indicates the proportion of syntactically valid strings produced by the neural network.
- *Correctness* measures the extent to which the generated SMILES strings accurately predict the intended molecular structures. For a given input, the output is considered correct if at least one of the predicted SMILES has a Tanimoto Similarity [25] greater than or equal to 0.8. A correctly generated SMILES string should map onto the same chemical structure as the target molecule, thus assuring chemical integrity in the outputs of the model. Invalid predictions are included in the assessment: if none of the generated strings for a specific input row are syntactically valid (*i.e.*, validity is zero), the correctness for the input is zero.

The values of these metrics are summarized in Table II, while the training statistics can be found in Table III. For each configuration, we comment on the results.

a) Baseline configuration: Starting from randomly initialized weights, this configuration achieved a Validity of 82% and a Correctness of 32%. These figures present the basic level of performance, showing the model to be capable of generating a significant proportion of valid SMILES strings.

b) Pre-Trained (Frozen Fine-Tune) configuration: the MT was initialized with pre-trained weights and the decoder was frozen during fine-tuning. Validity slightly decreased to 79%, with Correctness marginally declining to 31%. This decrement suggests that whereas pre-training is a good starting point, freezing the decoder may keep the architecture from reaching its full expressiveness for our main task. Despite this, considering the significantly lower training time (see Table III), the performance achieved by this model is remarkable. It is also interesting to note that, as visible in Figure 2, the initial loss of (b) is lower than the one of (a), and the accuracy is higher. This is a symptom of the success of the pre-training strategy, which gives the model a better starting point to learn.

c) The Pre-Trained (Full Fine-Tune) configuration: this case improves configuration (b), by running an extra fine-tuning without freezing the decoder, but using a lower learning

rate. This presented the largest improvements: Validity increased to 90% while Correctness increased to 37%. Allowing the fine-tuning of the decoder along with the rest of the model manages to exploit the architecture to a much higher extent while starting from an already successful configuration (b). Figure 2 and Table III show that the time it takes to fine-tune with configuration (c) added to (b) is less than the time it takes to train with configuration (a), while achieving much better performance. The jump in validation performance can mostly be attributed to the unfreezing of the decoder, combined with a lower learning rate, which allows the model to quickly learn more complex patterns without destroying the acquired knowledge. It is important to note that – given the similar performance of (a) and (b) – what in (a) was learned by both the encoder and the decoder in an end-to-end fashion has to be learned in (b) by only updating the encoder. Since (b) reaches a comparable performance to (a) despite updating only a fraction of the weights, unfreezing the decoder in (c) allows for a much larger performance gain. This is because introducing many newly trainable parameters enables the model to capture more information. Additionally, since training with fewer parameters tends to reduce overfitting, (b) benefits from an implicit regularization effect [38]. As a result, when the second fine-tuning phase begins, the model is already predisposed to better generalization compared to (a). The behavior of maximizing the performance of all parts of a Transformer network separately is an open research topic, and is often tackled with adaptive learning rates [33].

TABLE II
PERFORMANCE METRICS FOR DIFFERENT CONFIGURATIONS AND FINE-TUNING APPROACHES.

	Baseline (a)	Pre-Trained Frozen Fine-Tune (b)	Pre-Trained Full Fine-Tune (c)
Validity	82%	79%	90%
Correctness	32%	31%	37%

TABLE III
EPOCHS AND TRAINING TIME FOR DIFFERENT CONFIGURATIONS AND FINE-TUNING APPROACHES.

	Baseline (a)	Pre-Trained Frozen Fine-Tune (b)	Pre-Trained Full Fine-Tune (c)
Number of Epochs	22	15	15 + 6 = 21
Training Time (s)	1810	1095	1095 + 500 = 1595

The presented comparison evidences that pre-training enhances the Transformer’s capability of generating valid and correct SMILES strings with full fine-tuning. It is clearly seen by the improvement in Validity from 82% Baseline to 90% Full Fine-Tune, with just the use of pre-trained weights, thus allowing the overall model adaptation. Similarly, the improvement in Correctness from 32% to 37% highlights

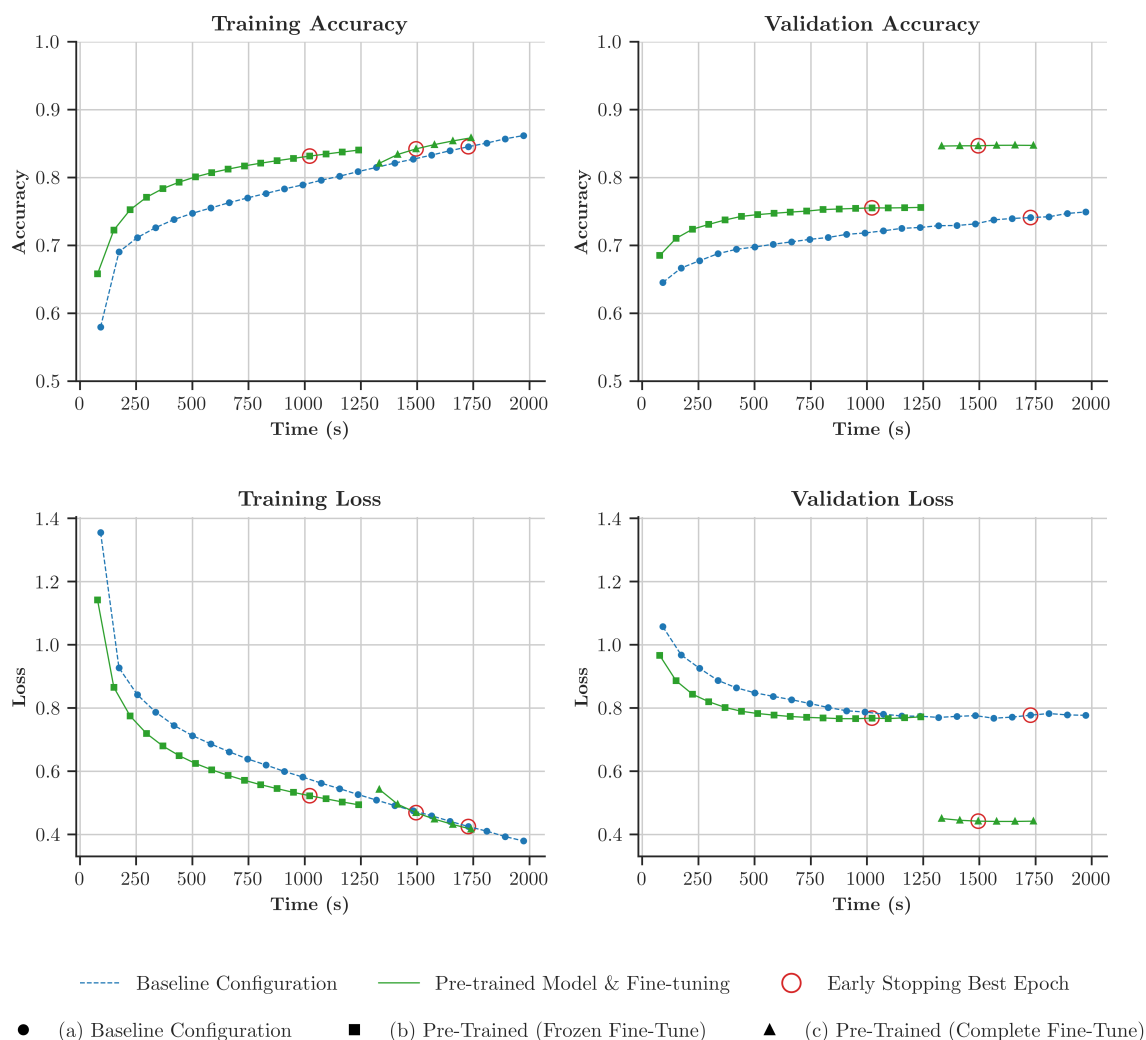


Fig. 2. Training and validation performance comparison of baseline and pre-trained models with fine-tuning strategies, including early stopping epochs. The top row presents Training Accuracy and Validation Accuracy, while the bottom row displays Training Loss and Validation Loss. The Baseline Configuration is represented by a blue dashed line, while the Pre-trained Model with Fine-tuning is shown as a green solid line. Different fine-tuning strategies are indicated by markers: squares for Pre-trained (Frozen Fine-Tune) and triangles for Pre-trained (Complete Fine-Tune). Red circles denote Early Stopping Best Epochs. The time axis represents the duration from the start of training to the end of each corresponding epoch, thus the curves do not begin at zero.

an enhanced capability of the model to accurately represent molecular structures post-fine-tuning. Configuration (c) also provides a $4.5\times$ increase in the accuracy of metabolized molecules (molecules that change during the process). In Configuration (b) the frozen decoder during fine-tuning seems to limit the potential gain of pre-training since the Validity and Correctness scores slightly went down compared to Baseline, although these scores were reached in significantly less time. This suggests that while transfer learning provides a valuable initialization, the flexibility to adjust all model parameters during fine-tuning is important for maximizing performance

gains.

A. Clinical Relevance

To illustrate the model's ability to capture complex and clinically relevant biotransformations, we present a case study involving a glutathione addition–elimination reaction (ID 104). The SMILES and 2D structures of both the substrate and metabolite are shown in Figure 3. The substrate is a halo-genated aromatic compound bearing an iodine substituent on an electron-rich ring. This structural motif is known to be susceptible to nucleophilic substitution by the thiol group of glutathione (GSH). The model correctly predicted the

formation of the thioether-linked metabolite, which results from the nucleophilic attack of GSH on the aromatic carbon, followed by elimination of the iodide group. This glutathione-mediated addition-elimination plays a key role in detoxifying electrophilic xenobiotics and therefore represents a crucial step for accurate metabolite profiling and early-stage drug safety assessment. However, being a complex and detoxification-driven transformation, it is often challenging to predict. The correct prediction of this transformation highlights the model's capacity to infer chemical reactivity beyond common metabolic transformations and to highlight potential safety-relevant pathways that are often neglected.

Substrate	Reaction	Metabolite
	104: Reactions of glutathione addition-elimination (at C-X groups, acyl halides, halogenated olefins, etc)	
<chem>FC12C(=CC=C(C1O2)N)I</chem>		<chem>Nc1ccc(I)c(O)c1SCC(NC(=O)CCC(N)(C(=O)O)C(=O)NCC(=O)O</chem>

Fig. 3. Example of clinically relevant metabolic reaction, that was correctly predicted by our model.

VIII. CONCLUSION AND FUTURE WORKS

The Transformer architecture can be effectively used to implement Molecular Transformers able to generate and manipulate small molecules encoded as SMILES strings. In this paper, we investigated the possibility of pre-training a MT to produce high quality SMILES, in order to maximize the validity of generated strings. Moreover, we also coupled the approach with a fine-tuning for a specific task (*i.e.*, drugs metabolism), in order to also improve the correctness of predictions.

Our results show that using a pre-trained model as a starting point can provide multiple benefits, that is, to reduce the overall training time while improving both the validity of SMILES and the correctness of metabolism prediction tasks. Moreover, a significant advantage is represented by the possibility of exploiting our pre-training approach for other tasks, not related to metabolism prediction, in the domain of SMILES-based molecules translation using Transformers.

Given the promise held by the proposed approach, we could think of extending Transformer-based machine translation beyond small molecules, broadening its applicability to other biological entities. Peptides, for example, are increasingly recognized as promising alternatives to antibodies in targeted therapies, making their accurate modeling and transformation an important direction for further research.

Another interesting direction is to improve model interpretability, such as using attention visualizations, to understand which atoms or regions of the SMILES strings contributed most heavily to our predictions.

The pre-trained model created for this paper is available upon request.

ACKNOWLEDGMENT

This work has been partially supported by the National Recovery and Resilience Plan and the European Union—Next Generation EU, Mission 4, Component 2, CUP B93D21010860004, as part of the research activities of the “National Centre for Gene Therapy and Drugs based on RNA Technology”, Spoke 8 “Platforms for RNA/DNA delivery”, under the project ALLIANCE (CUP H73C24000120005). This work was also partially supported by the high-performance computing infrastructure developed under the project “CONVECS”, funded by the PR Veneto FESR 2021-2027 program, Priority 1 - Specific Objective 1.1 - Action 1.1.2.

REFERENCES

- [1] N. Singh, P. Vayer, S. Tanwar, J.-L. Poyet, K. Tsaïoun, and B. O. Villoutreix, “Drug discovery and development: introduction to the general public and patient groups,” *Frontiers in Drug Discovery*, vol. 3, p. 1201419, 2023.
- [2] B. H. Abdelmonem, N. M. Abdelaal, E. K. Anwer, A. A. Rashwan, M. A. Hussein, Y. F. Ahmed, R. Khashana, M. M. Hanna, and A. Abdelnaser, “Decoding the role of cyp450 enzymes in metabolism and disease: A comprehensive review,” *Biomedicine*, vol. 12, no. 7, p. 1467, 2024.
- [3] F. Esteves, J. Rueff, and M. Kranendonk, “The central role of cytochrome p450 in xenobiotic metabolism—a brief review on a fascinating enzyme family,” *Journal of xenobiotics*, vol. 11, no. 3, pp. 94–114, 2021.
- [4] G. Price and D. A. Patel, *Drug bioavailability*. 2020.
- [5] K. Ball, G. Bruin, E. Escandón, C. Funk, J. N. Pereira, T.-Y. Yang, and H. Yu, “Characterizing the pharmacokinetics and biodistribution of therapeutic proteins: an industry white paper,” *Drug metabolism and disposition*, vol. 50, no. 6, pp. 858–866, 2022.
- [6] Z. Zhang and W. Tang, “Drug metabolism in drug discovery and development,” *Acta Pharmaceutica Sinica B*, vol. 8, no. 5, pp. 721–732, 2018.
- [7] Y. Lai, X. Chu, L. Di, W. Gao, Y. Guo, X. Liu, C. Lu, J. Mao, H. Shen, H. Tang, *et al.*, “Recent advances in the translation of drug metabolism and pharmacokinetics science for drug discovery and development,” *Acta Pharmaceutica Sinica B*, vol. 12, no. 6, pp. 2751–2777, 2022.
- [8] J. Kirchmair, M. J. Williamson, J. D. Tyzack, L. Tan, P. J. Bond, A. Bender, and R. C. Glen, “Computational prediction of metabolism: sites, products, sar, p450 enzyme dynamics, and mechanisms,” *Journal of chemical information and modeling*, vol. 52, no. 3, pp. 617–648, 2012.
- [9] A. Mazzolari, P. Perazzoni, E. Sabato, F. Lunghini, A. R. Beccari, G. Vistoli, and A. Pedretti, “Metaspot: a general approach for recognizing the reactive atoms undergoing metabolic reactions based on the metasqr database,” *International Journal of Molecular Sciences*, vol. 24, no. 13, p. 11064, 2023.
- [10] C. de Bruyn Kops, M. Šícho, A. Mazzolari, and J. Kirchmair, “Gloryx: prediction of the metabolites resulting from phase 1 and phase 2 biotransformations of xenobiotics,” *Chemical research in toxicology*, vol. 34, no. 2, pp. 286–299, 2020.
- [11] J. D. Tyzack and J. Kirchmair, “Computational methods and tools to predict cytochrome p450 metabolism for drug discovery,” *Chemical biology & drug design*, vol. 93, no. 4, pp. 377–386, 2019.
- [12] E. E. Litsa, P. Das, and L. E. Kavrakli, “Prediction of drug metabolites using neural machine translation,” *Chemical science*, vol. 11, no. 47, pp. 12777–12788, 2020.
- [13] D. Weininger, “Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules,” *Journal of chemical information and computer sciences*, vol. 28, no. 1, pp. 31–36, 1988.
- [14] A. Vaswani, “Attention is all you need,” *Advances in Neural Information Processing Systems*, 2017.

- [15] S. Multari, R. Özçelik, A. Mazzolari, M. S. Nobile, and F. Grisoni, "Predicting metabolic reactions with a molecular transformer for drug design optimization," in *2024 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pp. 1–8, IEEE, 2024.
- [16] H. Raunio, M. Kuusisto, R. O. Juvonen, and O. T. Pentikäinen, "Modeling of interactions between xenobiotics and cytochrome p450 (cyp) enzymes," *Frontiers in pharmacology*, vol. 6, p. 123, 2015.
- [17] M. Hennemann, A. Friedl, M. Lobell, J. Keldenich, A. Hillisch, T. Clark, and A. H. Göller, "Cypscore: Quantitative prediction of reactivity toward cytochromes p450 based on semiempirical molecular orbital theory," *ChemMedChem: Chemistry Enabling Drug Discovery*, vol. 4, no. 4, pp. 657–669, 2009.
- [18] P. Rydberg, "Reactivity-based approaches and machine learning methods for predicting the sites of cytochrome p450-mediated metabolism," *Drug Metabolism Prediction*, pp. 265–292, 2014.
- [19] L. Carlsson, O. Spjuth, S. Adams, R. C. Glen, and S. Boyer, "Use of historic metabolic biotransformation data as a means of anticipating metabolic sites using metaprint2d and bioclipse," *BMC bioinformatics*, vol. 11, pp. 1–7, 2010.
- [20] J. Eberhardt, D. Santos-Martins, A. F. Tillack, and S. Forli, "Autodock vina 1.2.0: New docking methods, expanded force field, and python bindings," *Journal of chemical information and modeling*, vol. 61, no. 8, pp. 3891–3898, 2021.
- [21] G. Cruciani, E. Carosati, B. De Boeck, K. Ethirajulu, C. Mackie, T. Howe, and R. Vianello, "Metasite: understanding metabolism in human cytochromes from the perspective of the chemist," *Journal of medicinal chemistry*, vol. 48, no. 22, pp. 6970–6979, 2005.
- [22] R. e. a. Gupta, "Artificial intelligence to deep learning: machine intelligence approach for drug discovery," *Mol. Divers*, 2021.
- [23] Y. Djoumbou-Feunang, J. Fiamoncini, A. Gil-de-la Fuente, R. Greiner, C. Manach, and D. S. Wishart, "Biotransformer: a comprehensive computational tool for small molecule metabolism prediction and metabolite identification," *Journal of cheminformatics*, vol. 11, pp. 1–25, 2019.
- [24] K. Zhu, M. Huang, Y. Wang, Y. Gu, W. Li, G. Liu, and Y. Tang, "Metapredictor: in silico prediction of drug metabolites based on deep language models with prompt engineering," *Briefings in Bioinformatics*, vol. 25, no. 5, p. bbae374, 2024.
- [25] P. Todeschini, V. Consonni, H. Xiang, J. Holliday, M. Buscema, and P. Willett, "Similarity coefficients for binary chemoinformatics data: overview and extended comparison using simulated and real data sets," *Journal of chemical information and modeling*, vol. 52, no. 11, pp. 2884–2901, 2012.
- [26] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *Journal of chemical information and modeling*, vol. 50, no. 5, pp. 742–754, 2010.
- [27] J. J. Irwin, K. G. Tang, J. Young, C. Dandarchuluun, B. R. Wong, M. Khurelbaatar, Y. S. Moroz, J. Mayfield, and R. A. Sayle, "Zinc20—a free ultralarge-scale chemical database for ligand discovery," *Journal of Chemical Information and Modeling*, vol. 60, p. 6065–6073, Oct. 2020.
- [28] A. Pedretti, A. Mazzolari, G. Vistoli, and B. Testa, "Metaqsar: an integrated database engine to manage and analyze metabolic data," *Journal of medicinal chemistry*, vol. 61, no. 3, pp. 1019–1030, 2018.
- [29] A. Mazzolari, L. Sommaruga, A. Pedretti, and G. Vistoli, "Metatree, a novel database focused on metabolic trees, predicts an important detoxification mechanism: The glutathione conjugation," *Molecules*, vol. 26, no. 7, p. 2098, 2021.
- [30] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [31] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," *Advances in neural information processing systems*, vol. 27, 2014.
- [32] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't stop pretraining: Adapt language models to domains and tasks," *arXiv preprint arXiv:2004.10964*, 2020.
- [33] C. Liang, H. Jiang, S. Zuo, P. He, X. Liu, J. Gao, W. Chen, and T. Zhao, "No parameters left behind: Sensitivity guided adaptive learning rate for training large transformer models," 2022.
- [34] K. D. Dhole, "A multi-encoder frozen-decoder approach for fine-tuning large language models," 2025.
- [35] A. A. Falaki and R. Gras, "A robust approach to fine-tune pre-trained transformer-based models for text summarization through latent space compression," in *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 160–167, 2022.
- [36] A. Radhakrishnan, K. Yang, M. Belkin, and C. Uhler, "Memorization in overparameterized autoencoders," 2018.
- [37] "RDKit: Open-source cheminformatics," <http://www.rdkit.org>. [Online; accessed 2024].
- [38] K. Goutam, S. Balasubramanian, D. Gera, and R. R. Sarma, "Layerout: Freezing layers in deep neural networks," *SN Computer Science*, vol. 1, Sept. 2020.