Exploring Neural Scaling Laws in Molecular Pretraining with Synthetic Tasks

Rodrigo S. Hormazabal [*1] **Seung Woo Ko** [1] **Inwan Yoo** [1] **Sehui Han** [1] **Paul Bertens** [1]

Abstract

Acquiring large-scale annotated molecular datasets has been one of the main bottlenecks in scaling foundational models for chemistry applications. Proprietary issues, cost, and complex experimental setups have led to restrictively small labeled datasets for some predictive tasks. Unlike in language models, where pre-training has shown significant improvements in downstream tasks, molecular pre-training has yet to demonstrate a similar impact. Inspired by the success of neural scaling laws, we explore the impact of synthetic pre-training for molecular machine learning models on their downstream performance. We pre-train models of three sizes (12M, 45M, 230M parameters) using 12 synthetic tasks in an encoder-decoder setup and evaluate their performance across 10 downstream tasks. Our findings reveal a general correlation between pre-training perplexity and downstream task performance, with this relationship varying across different tasks. These insights suggest that pre-training metrics could provide valuable estimates of model performance post-fine-tuning. We pre-train models with over 10B tokens and observe that models saturate, indicating potential for further parameter scaling. This study represents a preliminary exploration of synthetic task pre-training for molecular models, which can be complementary to other pre-training methods such as multi-task learning with labeled data and multimodal learning.

1. Introduction

Recent advances in artificial intelligence are significantly impacting molecular design and drug discovery research. However, acquiring large-scale annotated molecular datasets remains a bottleneck due to being time-consuming to run high-quality experiments and the related costly lab work. This results in relatively small labeled datasets, where machine learning approaches tend to struggle with out-of-distribution (OOD) structures and tasks.

Inspired by the success of language-based foundation models, which can be fine-tuned to solve diverse downstream tasks even in low-data regimes, this paper explores similar approaches for molecular tasks, specifically using *"synthetic"* tasks. Several literature exploring pre-training of molecular models exists; self-supervised setups(Chithrananda et al., 2020; Wu et al., 2023a; Rong et al., 2020a; Wang et al., 2022), pre-training with large labeled datasets (Jiao et al., 2023; Ying et al., 2021; Zaidi et al., 2022), and multi-modal approaches (Zeng et al., 2022; Edwards et al., 2022; Fang et al., 2022; Li et al., 2022a). For a more comprehensive survey on chemical pre-trained refer to the work of (Xia et al., 2022)

Unlike language models, which benefit from abundant online text data allowing for a next-token prediction objective that naturally aligns with language understanding, obtaining this learning signal in the chemical domain is less straightforward. For self-supervised approaches, predicting masking objectives for stable structures might not provide a strong and scalable learning signal to build general foundation models. In this context, we study the impact of pre-training on low-cost synthetic tasks on downstream applications, focusing on which tasks can endow models with a useful set of priors and operations that can effectively help solve tasks native to graph-space.

Following existing work in neural scaling laws for language models (Kaplan et al., 2020; Hernandez et al., 2021), vision models (Klug & Heckel, 2022; Cherti et al., 2023), Reinforcement Learning (Neumann & Gros, 2022), and pre-training from labeled chemical data (Frey et al., 2023), we pre-train a set of T5 models (Raffel et al., 2020) with a diverse set of synthetic tasks. These tasks range from general tasks, such as finding the shortest paths between arbitrary atoms in a molecule and synthetic atom-mapping, to low-fidelity software-generated tasks, such as 2D positions or RDKit functions (Landrum, 2016). This approach allows us to create large-scale data without the need for expensive human-curated annotations, effectively extracting

[1]LG AI Research. Correspondence to: Rodrigo S. Hormazabal <rodrigo@lgresearch.ai>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

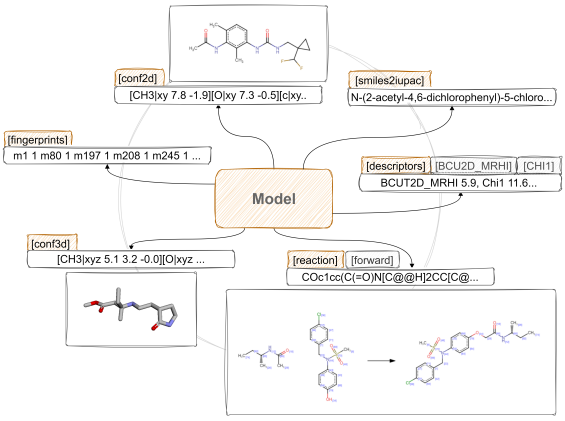

Figure 1: Pre-training on synthetic task setup, where all tasks are tokenized into text space to pre-train a LLM on large-scale synthetic tasks.

information from individual molecules and selecting what information to embed into the model.Also, synthetic pre-training can be seen as complementary to pre-training with labeled chemical data or multi-modal datasets.

Previous approaches in pre-training for molecular tasks, such as (Li et al., 2024) and (Gao et al., 2022), highlight the potential of large-scale pre-trained models in molecular discovery. These models, leveraging vast amounts of unlabeled data, generate informative representations crucial for downstream tasks. Additionally, studies like (Choung et al., 2023) and (Fang et al., 2023) emphasize the integration of human knowledge and contrastive learning to enhance model performance.

The motivation behind this approach is supported by findings from various studies, such as SELFormer (Yüksel et al., 2023) and (Born & Manica, 2023), which demonstrate the effectiveness of Transformer-based models in molecular tasks. Furthermore, research like (Born et al., 2023) and (Wang et al., 2023) underscores the importance of pre-training on diverse molecular representations.

1.1. Motivation

We opted to create a wide variety of tasks to avoid potential negative transfer that can occur in some pre-training methods (Wang et al., 2019; Hu et al., 2019). If we only pre-train on one particular task a high risk we only learn embeddings to solve that one task. By diversifying the tasks we can ensure we are more likely to learn generalizable embeddings that cover the full space of potential downstream molecular tasks.

Even if the tasks are synthetic, they can still be useful to create a more reasonable prior for the model representations than a completely random one. The main idea is to equip the model with mechanisms and symmetries normally introduced through inductive, through a more flexible mechanism instead of requiring the manual calculation of these features as is done in standard Graph Neural Networks (GNNs) approaches (Schütt et al., 2017; Gasteiger et al., 2020; 2021). By having a set of synthetic pre-training objectives integrated together through next-token prediction, we can gain secondary benefits as well, where other features might emerge just from this single objective. For example in the case of predicting 3D conformations, we could pre-calculate an approximate 3D conformer configuration and use it in a graph encoder through relative position bias. However, we now need the ability to calculate or predict the conformations at inference time (which is non-trivial (Zhou et al., 2023; Xu et al.; Jing et al., 2022)), and it would only this resulting characteristics and no lower-level interactions that led to it. If on the other hand during training we learn to predict these conformations, the model has to learn to encode other related features that might be helpful in the conformation prediction task (such as the degree of the atoms, relation between node-types). We will be less dependant on a surrogate model or simulation at inference time.

SMILES encoder vs Graph encoder There is a close relation between sequence-based Transformer models (Vaswani et al., 2017) and attention-based Graph Neural Networks (GNNs) (Veličković et al., 2017). Besides the local or global attention mechanism, one of the main differences is in their positional encoding, where generally GNNs use either relative positional encodings in the attention bias or more topological/structural encodings, while the vanilla Transformers use absolute positional encodings.

Specifically for molecular representation learning, GNNs can better represent natural inductive biases for graphs structures, which leads them to be more sample-efficient than sequence-based approaches, since they are permutation invariant to node ordering, unlike SMILES, not relying on node ordering when learning representations. GNNs, however, can be more complex from an computational perspective, requiring specialized sparse gather-scatter operations and special batching techniques. In the case of sequence-based models, due to the rise of generative pre-trained transformers (GPT) (Brown et al., 2020), significant innovations are constantly being developed to make transformer-based pipelines more efficient for distributed training and inference, even at the hardware level (Wang et al., 2020). This presents a compelling argument to keep exploring for opportunities on how new advances in other AI-related fields can impact research in molecular representation learning, specifically how string representations for molecules can better exploit optimizations in the NLP domain. Although the

methods described in this paper can also be generalized to the pre-training of foundation models using synthetic tasks for graph-based encoding models, base on these reasons, we focus our experiments on SMILES based representations. In this context, one of our main motivation is to overcome some of the limitations of SMILES representations (like the lack of permutation invariance), through large scale pre-training, eliminating the need for manual feature engineering of a graph-based encoder.

1.2. Contributions

Our preliminary findings show that models pre-trained on the synthetic tasks explored here benefit downstream tasks and overall improve as the number of pre-training tokens increases for all model sizes. These trends align and complement with results from studies like (Zhang et al., 2023) and (Seidl et al., 2023), which show significant improvements in model performance through advanced pre-training techniques. To measure the impact of the learned molecular representations, we adopt a simplistic fine-tuning setup where we freeze all encoder layers and pool the token representations at the end of the transformer. We then train a shallow MLP on top of this pooled representation and evaluate downstream performance.

It's important to note that our focus is not on achieving State-of-the-Art (SOTA) pre-training performance under this setup, which would likely require a more flexible fine-tuning approach. Instead, we aim to explore the impact of pre-training with synthetic tasks compared to from-scratch baselines. Nevertheless, even with this simplistic setup, many of our downstream metrics are competitive with SOTA models, whether pre-trained or specialized architectures. Analysis of more powerful fine-tuning techniques (e.g., adding a learnable pooling token analogous to `[CLS]` tokens or full fine-tuning) and comparison with all available SOTA models for our downstream tasks is work-in-progress.

To summarize our contributions:

- **Creation of a Large-Scale Synthetic Dataset:** We construct a comprehensive dataset of synthetic molecular tasks covering both singular molecules and multi-molecule reactions, comprising approximately 1.1 billion molecules and totaling around 1 trillion tokens. We plan to open-source our data, pre-training scripts, and multi-GPU parallel downstream evaluation.
- **Initial Scaling Law Observations:** Our initial results align with scaling laws observed in large language models. The observation that the pre-training loss saturates long before running out of data suggests that scaling to even larger models could further enhance performance. Models continue to improve while perplexity in the pre-training task decreases, and this trend persists with more pre-training tokens and increased model capacity.
- **Downstream Benchmarks:** Despite a simplistic fine-tuning setup, our models achieve competitive performance metrics on several downstream tasks, underscoring the effectiveness of the learned molecular representations. More detailed comparisons and benchmarks against SOTA models are work in progress.
- **Ablation Study of Pre-Training Tasks:** We conducted an ablation study to determine which pre-training tasks have the most significant impact. This analysis helps identify the most effective tasks for improving model performance and build new synthetic tasks.

These contributions highlight the potential our synthetic pre-training setup to improve chemistry models, with possibilities to extend synthetic tasks and scale models further. Our future focus is to provide an initial blueprint to design, test and expand synthetic tasks pre-training in an open-source effort.

2. Datasets

The main requirement for pre-training datasets is scale; we focus on datasets large enough to allow for large-scale pre-training (¿100M+ molecules), such as Pubchem (Kim et al., 2023) and ZINC (Irwin & Shoichet, 2005; Irwin et al., 2020; Tingle et al., 2023). Although it is possible to combine labeled datasets during pre-training, we would like to avoid this as much as possible in this work, since it would also reduce the set of possible downstream tasks we can evaluate on, limiting the scalability of our pre-training approach.

As mentioned before, some of the largest publicly available datasets with structures are currently ZINC (1B molecules) and PubChem ($\tilde{1}$50M molecules). ZINC is partially synthetic but mostly consists of synthesizable compounds that are commercially available. Several versions of ZINC exist (ZINC15, ZINC17, ZINC22), and it is continuously being updated to extend the set of viable compounds. In addition, other dataset that could potentially be used is the GDB-17 177B (Ruddigkeit et al., 2012), which enumerate all 'reasonable' permutation of up to 17 atoms that result in relatively stable molecules. However, these are even more synthetic, and possibly not synthesizable in reality, so we chose not not include these in our preliminary experiments.

In total our dataset then is comprised of $\tilde{1}$.1B molecules, to which we can apply synthetic tasks; as a reference, with the ten tasks presented in this preliminary work and an average of 100 tokens per molecule, we can obtain a dataset of about 1T tokens for pre-training experiments.

3. Synthetic Tasks

We perform our experiments on a suite of mostly synthetic or highly available tasks, 9 of which capture single molecular patterns (IUPAC Name Prediction, Stereochemistry, Sanitization, SMILES2IUPAC, Shorted Paths, 2D Conformation, 3D Conformation, Attributes, Descriptors) and 3 of which focus to learn inter-molecular interactions emulating mechanisms useful for chemical reactions/retrosynthesis (Synth. Atom mapping, Synth. forward reaction, Synth. Retrosynthesis). We briefly describe how we create each task and the reasoning their design . We create input/output pairs for each of the tasks, and ensure all of them can be completely reason over text-space through special conversion rules depending on the task. We also randomly mask input tokens during training, similar to masked language modeling (MLM) (Devlin et al., 2018), forcing the decoder to look at neighbouring tokens and atoms when attempting to generate learn these tasks.

3.1. RDKit Functions, Descriptors, and Fingerprints

The primary goal of pre-training on RDKit outputs is to enable the model to learn chemical knowledge that is indirectly encoded in the hand-crafted rules of RDKit implementations (Landrum, 2016).

Basic chemical knowledge can be extracted by counting the types of atoms and bonds in a molecule, identifying common substructures, or applying standard sanitization rules. We convert such knowledge into tasks by training the model to predict the number of atoms and bonds in a molecule, perform sanitization on corrupted SMILES, and estimate RDKit's fingerprints and descriptors from the SMILES input. Additionally, we compute the synthetic accessibility score in RDKit, which serves as another task.

By learning these tasks, the model can encode chemical knowledge embedded in RDKit's rules, such as properly counting atoms and bonds (e.g., three carbons, one nitrogen, three single bonds, one double bond) and encoding common substructures (e.g., Morgan fingerprints (Morgan, 1965)).

3.2. Graph features and shortest paths

When encoding SMILES in a Transformer, we lose the relative positional information between atoms, especially since SMILES can be permuted without changing the molecule. This loss of positional encoding means we also lose permutation invariance, where the permutation of SMILES can affect the encoding and resulting embedding. To address this, we utilize shortest path prediction between atoms in the molecule.

Shortest paths between atoms are a common method of including relative positional encodings while maintaining equivariance. We randomly sample two atoms in the molecule and calculate both the length of the shortest path and the sequence of atom mapping numbers in that shortest path. We train the model to predict these two values, implicitly learning the relative positions of the atoms in SMILES.

The model should learn to predict the shortest path length and the sequence of the path traversal directly. Our task tokens thus become the task and a query between two atoms within the atom-mapped SMILES input (e.g., `[sp][query][71][67]`), which should output the length and sequence (e.g., *len: 10*, *seq: 71 96 16 93 94 35 63 42 69 52 67*). This way, we can indirectly learn shortest path encodings between atoms. Since the decoder has to be capable of performing shortest path traversal, embeddings should also learn to align with this task.

The main goal is to learn the graph inductive bias itself, instead of manually constructing a graph-based inductive bias. By learning the shortest path, other useful features might also be learned, such as atom types and atom degrees, to ensure embeddings are unique along the path.

3.3. Synthetic reactions and atom mappings

Learning the role of a molecule in a reaction is crucial not only for reaction-based downstream tasks but also for property prediction. Instead of relying on limited reaction data, we create synthetic reactions by fragmenting molecules. To ensure these fragmentations yield more reaction-like results, we calculate the probability distribution for each bond in the molecule. This calculation is based on the normalized max-flow in the molecular graph (analogous to spectral clustering), bond type (giving less weight to aromatic rings), and the proximity to leaf atoms, which are more likely to break. This method reduces the chances of unlikely reactions, such as the breaking of rings. See Figure 2 for an example of a generated reaction using this method.

Additionally, we train the model to predict the mapping between atoms in synthetic reactions. This forces the model to learn how to distinguish atoms in distinct local neighborhoods, which is closely related to the atom-mapping task in reaction prediction (Schwaller et al., 2019).

By creating synthetic reactions and enforcing atom mapping predictions, the model learns to encode multiple molecules and match graphs. Graph-matching requires learning embeddings for atoms that are unique and accurately describe their local neighborhood, leading to better overall embeddings.

3.4. SMILES to IUPAC

IUPAC (International Union of Pure and Applied Chemistry) (Bergmeier, 1998) names of molecules contain rich

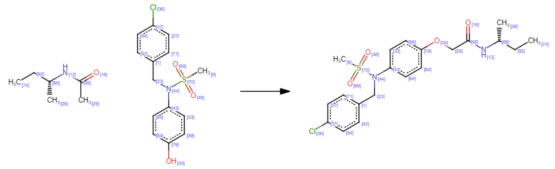

Figure 2: Example of a synthetic reaction that was created by fragmenting the product molecule into two reactants.

Figure 3: Example of how we can tokenize the 2D positions of atoms by rounding the absolute positions to create a hierarchical tokenization. A similar method can also be applied in 3D space.

information about their functional groups. By training a model to predict the IUPAC names from SMILES representations, we aim to align the SMILES embeddings in latent space, allowing them to represent similar functional groups more effectively. This alignment helps in capturing the structural and functional similarities between molecules. Additionally, predicting IUPAC names is challenging due to the one-to-many mapping nature, where a single molecule can have multiple valid IUPAC names. This complexity enhances the model's ability to learn robust and generalized embeddings. To create this task, we sampled approximately 100 million molecules from the PubChem database that have corresponding IUPAC names. Even though this task is not strictly synthetic, there is enough available information for our pre-training purposes.

3.5. Conformations and stereochemistry

The structures of molecules significantly impact their properties and reactivity. Therefore, it is crucial for the model to understand structural information to perform molecular downstream tasks. However, SMILES representations often fail to retain this information. To augment the model with structural data, we obtain approximate 2D and 3D conformations from the ZINC dataset (3D, calculated with OpenEye's Omega (Hawkins et al., 2010)) and standard RDKit functions (2D). In our experiments, we focus on an autoregressive Transformer model and need a method to tokenize the coordinates of each atom in a molecule. By assigning coordinates to each atom in the SMILES sequence, similar to assigning atom-map numbers, we achieve a custom conversion from position to text space. By rounding the positions to a certain decimal place, the sequential prediction of the coordinates in autoregressive decoding works as if we are first approximating the position of the atom in a larger grid and then sequentially determining the more precise positions, as shown in Figure 3.

We thus get a text-based representation of the 2D and 3D conformations, e.g., *[C—xyz 3.1 9.0 1.3][N—xyz 3.1 9.0 1.3]*, where *—xy* and *—xyz* let the model decode either 2D or 3D positions purely in text space. Although these are absolute positions and not SE(3)-invariant, given that we are using them as targets and not features, and with a large enough number of training examples, we believe this can constitute an extremely powerful learning signal. By tokenizing and learning to predict these positional SMILES representations, representations of the 3D and 2D geometric space of molecules should emerge within the latent space of the encoder model. Since our goal is purely for pretraining, we do not necessarily need high accuracy; we only need a sufficient self-supervised learning signal for the model to create better representations.

In addition, the stereochemistry of molecules can also have a significant impact on their properties. We add a task of predicting a molecule's stereochemistry by removing the stereochemical information from the SMILES and recovering it in the decoder.

4. Method

4.1. Model

We use a T5-based model with an encoder-decoder architecture (Raffel et al., 2020). This architecture allows us to separate the encoder and decoder parts of the model, enabling us to use only the encoder for fine-tuning on downstream tasks. The choice of model sizes was made to align with model sizes commonly used in previous literature (Irwin et al., 2022). We opted for the following model sizes:

- **12M / 5.5M (encoder-only)**
- **45M / 19.5M (encoder-only)**
- **230M / 101M (encoder-only)** (in-progress)

All models are initially pre-trained on synthetic tasks using the full T5 model. Subsequently, the encoder part is used with a linear layer for fine-tuning on downstream tasks. The pre-training process was conducted on 8xA40 GPUs, with the 12M model taking approximately 12 hours, the 45M model taking approximately 48 hours, and the 230M parameter model expected to train in about 5 days.

4.2. Tokenizer

For each task, we include a special task token that is provided only to the decoder. This decision ensures that the

learned encoding remains as general as possible for any potential downstream task. By restricting the task token to the decoder, the encoder and embedding layers must be larger, as they are required to encode all features regardless of the downstream or decoding task. If the task token were included in the encoder, the model could dynamically adjust its attention behavior, but we would not know the task token at inference time. Our goal is to develop a generalist encoding model.

Although we generally expect SMILES inputs during the fine-tuning of downstream tasks, we also added extra tokens for all additional characters or words that might be used in some of the pre-training tasks (e.g., numbers for conformation predictions or IUPAC generation). In total, the vocabulary size is around 1100 tokens.

5. Results

5.1. Downstream property prediction finetuning

To evaluate the downstream task performance of the pre-trained models, we fine-tuned them on MoleculeNet (Wu et al., 2018) property prediction tasks. A prediction head with two fully connected layers was fed with mean-pooled hidden states from the encoder of the pre-trained model to predict the desired property values. The model was trained for 5000 steps with a batch size of 64 and early stopped when the validation metric did not improve for 5 validation checks. Each dataset was split using scaffold splitting following previous work (Chithrananda et al., 2020; Li et al., 2022b; Luong & Singh, 2024; Ross et al., 2022; Rong et al., 2020b; Wu et al., 2023b; Zhu et al., 2022; Bai et al., 2023), and the test set performance was recorded.

5.2. Downstream tasks evaluation

Figures 4 and 5 display the results on the downstream tasks for the 12M and 45M models, as well as initial results for the 230M model (still in progress). Multiple checkpoints of the pre-trained models were fine-tuned and evaluated to investigate the effect of the amount of data seen for each model size. Generally, the larger models performed better on the downstream tasks as the number of pre-training tokens increased.

For certain tasks (ClinTox, Tox21, QM8, QM9), the smaller 12M (5.5M encoder) model showed a tendency to perform worse on downstream tasks with more pre-training tokens, possibly indicating that its capacity is too limited to learn proper embeddings for all tasks. This could mean it sacrifices performance on harder pre-training tasks over time, specializing in tasks it can solve, potentially leading to an embedding space that does not generalize well to downstream tasks. Further experiments are needed to fully investigate this behavior.

Table 1: Model performance on various classification tasks for different model sizes and checkpoints. Higher is better (↑).

Model Size	Checkpoint	BACE ↑	BBBP ↑	ClinTox ↑	Tox21 ↑
stp-t5-12m	Baseline	0.6841	0.6684	0.9860	0.7621
	6M Tokens	0.7120	0.6803	0.9883	0.7554
	24M Tokens	0.7726	0.6624	0.9878	0.7574
	42M Tokens	0.7890	0.6604	0.9871	0.7602
	60M Tokens	0.7775	0.6568	0.9871	0.7613
	300M Tokens	0.8051	0.7040	0.9867	0.7685
	2.1B Tokens	0.8010	0.6964	0.9927	0.7583
	3.6B Tokens	0.8022	0.7167	0.9753	0.7545
	5.7B Tokens	0.7936	0.7081	0.9713	0.7493
	7.8B Tokens	0.7972	0.7065	0.9500	0.7428
	11.4B Tokens	0.8059	0.7069	0.9574	0.7344
stp-t5-45m	Baseline	0.6992	0.6401	0.9887	0.7624
	6M Tokens	0.7332	0.6697	0.9884	0.7713
	24M Tokens	0.7231	0.6596	0.9888	0.7686
	42M Tokens	0.7396	0.6606	0.9882	0.7674
	60M Tokens	0.7468	0.6691	0.9882	0.7660
	300M Tokens	0.7707	0.6750	0.9876	0.7810
	2.1B Tokens	**0.8116**	0.7123	0.9889	**0.7915**
	3.6B Tokens	0.7950	0.7115	0.9925	0.7887
	5.7B Tokens	0.8091	0.7113	0.9960	0.7892
	7.8B Tokens	0.8054	**0.7394**	0.9980	0.7873
	11.4B Tokens	0.8082	0.7336	**0.9984**	0.7877
stp-t5-230m	Baseline	0.6995	0.6580	0.9879	0.7650
	6M Tokens	0.6998	0.6302	0.9879	0.7629
	24M Tokens	0.7350	0.6364	0.9890	0.7638
	42M Tokens	0.7441	0.6414	0.9877	0.7655
	60M Tokens	0.7503	0.6442	0.9885	0.7733
	300M Tokens	0.7654	0.6685	0.9975	0.7773

Figure 7 shows the perplexity over next-token prediction for all the synthetic pre-training tasks. The tables demonstrate that as more tokens are shown to the pre-trained models, they improve at their pre-training tasks. Additionally, larger models exhibit better performance, which is expected. The pre-training perplexity generally aligns with downstream task performance. However, models tend to saturate in their pre-training objectives due to constrained size and sometimes show worse performance. While further experiments are needed to fully investigate the relationship between model size, the number of pre-training tokens, the loss of pre-training tasks, and downstream performance, initial results are promising.

Given its increased capacity, initial results for fine-tuning the 230M model on pre-training tasks show perplexity curves seem to converge to far lower values but at slower rate, which is promising but, due to the training speed of this bigger version, these improvements cannot be seen in our early checkpoints (until 300B tokens). We will report final results for pre-training and downstream performance for 230M and a 700M models trained on 200B billion tokens in future work (in-progress)

5.3. Ablation study

We conducted initial ablation studies on each of the individual pre-training tasks to assess their importance (see Table 2). Overall, pre-training on the synthetic accessibility score and learning to count the number of atoms and bonds appear to be among the most crucial tasks. This may be because the

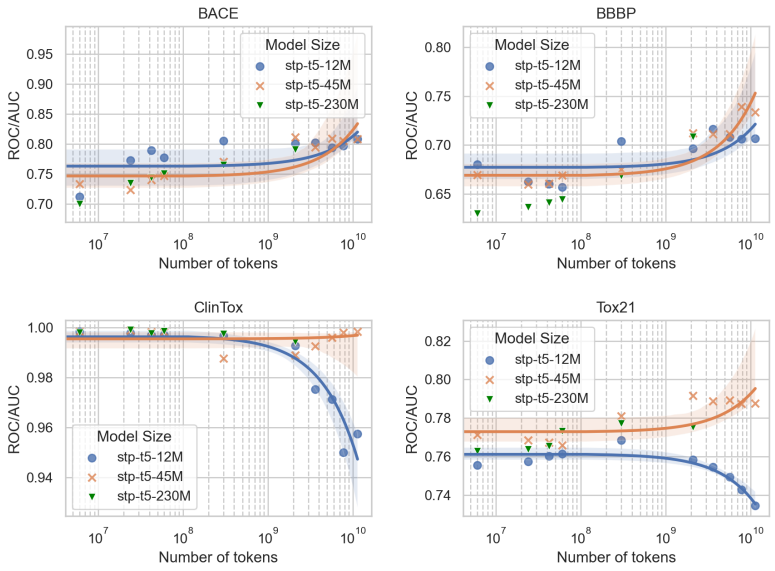

Figure 4: Plots of downstream classification performance on downstream tasks. The baseline model trained from scratch is plotted as dotted line for each model size.

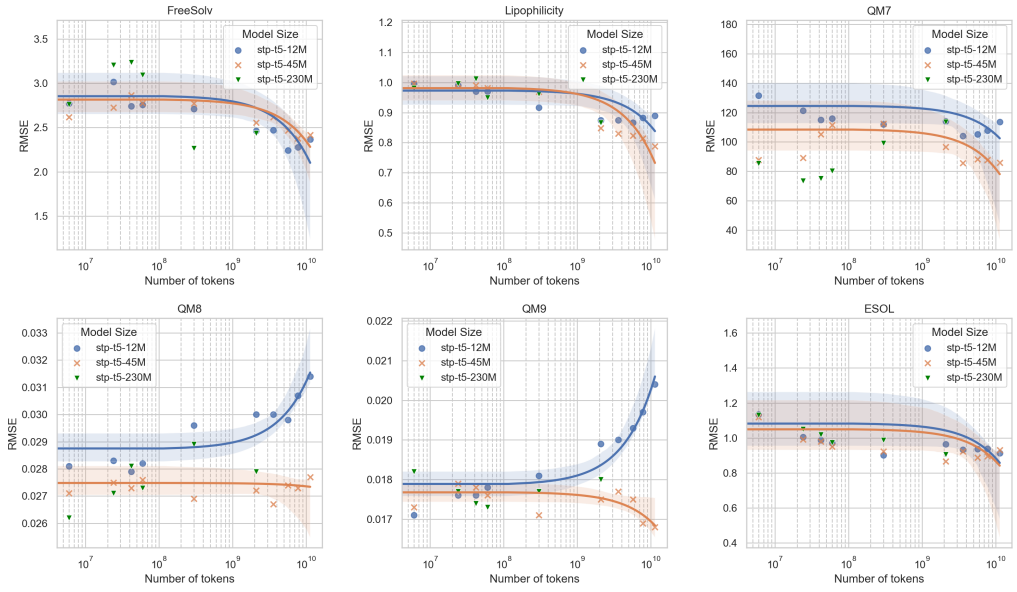

Figure 5: Plots of downstream regression performance on downstream tasks. The baseline model trained from scratch is plotted as dotted line for each model size.

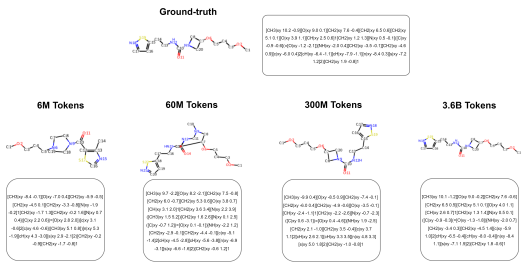

Figure 6: An example of how 2D conformation predictions evolve during pre-training, becoming increasingly accurate as more pre-training tokens are processed. This indicates that the model improves its understanding of the geometric structure of molecules over time.

models, at 12M and 45M capacities, focus on solving more manageable tasks. Conversely, pre-training solely on the 3D conformation or descriptors tasks performs significantly worse, likely due to their complexity. A more extensive analysis is needed to fully understand the reasons behind this, including examining larger model sizes and the dynamics of pre-training task losses.

Table 2: Ablation study on the impact of each synthetic pre-training task on downstream performance. The pre-training setup is kept consistent, but for each pre-training run, only tokens/samples from a single task are used. For clarity, we present the overall rank of each pre-training task.

Model Size	Task	BACE	BBBP	ESOL	QM7	Tox21	Avg. Rank
	All tasks	1	1	1	1	1	1.0
	Formula & SAScore	4	2	2	2	2	2.4
	Stereochemistry	3	7	4	5	7	5.2
	Sanitization	8	6	3	7	3	5.4
	Forward Reaction	7	3	7	11	5	6.6
	SMILES2IUPAC	6	4	9	4	11	6.8
stp-t5-12m	Mapping	2	12	8	6	9	7.4
	Retro Reaction	13	5	5	10	6	7.8
	Attributes	5	8	10	12	8	8.6
	2D Conformation	11	9	6	9	12	9.4
	Shortest Path	12	10	13	8	4	9.4
	Descriptors	10	13	12	3	14	10.4
	3D Conformation	9	11	11	13	10	10.8
	All tasks	1	1	1	1	1	1.0
	Formula & SAScore	8	3	4	5	2	4.6
	Stereochemistry	7	6	2	3	5	4.8
	Sanitization	10	11	7	2	3	6.8
	Forward Reaction	2	7	3	4	4	4.0
	SMILES2IUPAC	12	4	10	12	10	10.0
stp-t5-45m	Mapping	6	13	6	11	8	9.4
	Retro Reaction	11	10	9	8	7	9.4
	Attributes	3	12	5	6	9	7.2
	2D Conformation	4	9	12	9	12	9.8
	Shortest Path	5	2	11	10	6	7.2
	3D Conformation	9	5	8	13	11	9.6
	Descriptors	13	8	13	7	13	11.6

6. Discussion

Limitations While the data generation process is highly scalable for most tasks since they are calculated and derived directly from the SMILES representations, there are notable limitations. Specifically, the Smiles2IUPAC task is constrained by the availability of labels from PubChem, as these labels are challenging to derive or generate. One potential solution is to train a model that can generate IUPAC names from SMILES, but a more scalable alternative might be to use a byte pair encoding tokenizer on the SMILES strings and use those tokens as outputs. This type of tokenization might encode functional groups similarly to IUPAC names. Additionally, fingerprints serve a similar function, suggesting that the IUPAC task may not be as critical. Further experiments are necessary to investigate the importance and impact of the IUPAC task on overall model performance.

Improvements One important improvement to our setup would be the addition of chemical reaction benchmarks to better understand the impact of our synthetic tasks. There are also several improvements possible for fine-tuning the model on the downstream tasks. We used a simple pooling layer on top of the encoder that pools the embeddings of the entire sequence, but this might not be optimal, since it cannot learn to weigh the importance of different tokens in the sequence when pooling. As mentioned before, more specialized fine-tuning techniques, such as adding a learnable pooling token analogous to `[CLS]` tokens or full fine-tuning, need to be explored. These methods could potentially enhance the model's performance on downstream tasks by better leveraging the pre-trained embeddings.

Future directions One of the main observations from pre-training is that we saturate on the pre-training loss long before we run out of data. This indicates that we could potentially train even larger models. Currently, we have 230M and 700M models in progress, but with further developments, we could aim for 3B or 7B parameters, similar to current Pre-trained Large Language Models (LLMs). We only trained on about 10% of the molecules and only 1% of the possible tokens we could have generated. Larger models might get expensive at inference time for downstream tasks, but we can potentially leverage optimizations being developed for LLMs, such as LoRa-based fine-tuning (Hu et al., 2021), or use distillation techniques to create a smaller model.

References

Bai, P., Liu, X., and Lu, H. Geometry-aware line graph transformer pre-training for molecular property prediction. *arXiv preprint arXiv:2309.00483*, 2023.

Bergmeier, S. C. *Principles of Chemical Nomenclature: A Guide to IUPAC Recommendations*, volume 42. Blackwell Science, Oxford, U.K., May 1998. ISBN 0-86542-685-6. doi: 10.1021/jm990206e. URL https://doi.org/10.1021/jm990206e.

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

Born, J., Markert, G., Janakarajan, N., Kimber, T. B., Volkamer, A., Martínez, M. R., and Manica, M. Chemical representation learning for toxicity prediction. *Digital Discovery*, 2(3):674–691, 2023.

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

Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., and Jitsev, J. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 2023.

Chithrananda, S., Grand, G., and Ramsundar, B. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

Choung, O.-H., Vianello, R., Segler, M., Stiefl, N., and Jiménez-Luna, J. Learning chemical intuition from humans in the loop. 2023.

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

Edwards, C., Lai, T., Ros, K., Honke, G., Cho, K., and Ji, H. Translation between molecules and natural language. *arXiv preprint arXiv:2204.11817*, 2022.

Fang, Y., Zhang, Q., Yang, H., Zhuang, X., Deng, S., Zhang, W., Qin, M., Chen, Z., Fan, X., and Chen, H. Molecular contrastive learning with chemical element knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 3968–3976, 2022.

Fang, Y., Zhang, Q., Zhang, N., Chen, Z., Zhuang, X., Shao, X., Fan, X., and Chen, H. Knowledge graph-enhanced molecular contrastive learning with functional prompt. *Nature Machine Intelligence*, 5(5):542–553, 2023.

Frey, N. C., Soklaski, R., Axelrod, S., Samsi, S., Gomez-Bombarelli, R., Coley, C. W., and Gadepally, V. Neural scaling of deep chemical models. *Nature Machine Intelligence*, 5(11):1297–1305, 2023.

Gao, Z., Tan, C., Wu, L., and Li, S. Z. Cosp: Co-supervised pretraining of pocket and ligand. *arXiv preprint arXiv:2206.12241*, 2022.

Gasteiger, J., Groß, J., and Günnemann, S. Directional message passing for molecular graphs. arxiv. 2020 doi: 10.48550. *arxiv*, 2020.

Gasteiger, J., Becker, F., and Günnemann, S. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34: 6790–6802, 2021.

Hawkins, P. C., Skillman, A. G., Warren, G. L., Ellingson, B. A., and Stahl, M. T. Conformer generation with omega: algorithm and validation using high quality structures from the protein databank and cambridge structural database. *Journal of chemical information and modeling*, 50(4):572–584, 2010.

Hernandez, D., Kaplan, J., Henighan, T., and McCandlish, S. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., and Leskovec, J. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

Irwin, J. J. and Shoichet, B. K. Zinc- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, 45(1): 177–182, 2005.

Irwin, J. J., Tang, K. G., Young, J., Dandarchuluun, C., Wong, B. R., Khurelbaatar, M., Moroz, Y. S., Mayfield, J., and Sayle, R. A. Zinc20—a free ultralarge-scale chemical database for ligand discovery. *Journal of chemical information and modeling*, 60(12):6065–6073, 2020.

Irwin, R., Dimitriadis, S., He, J., and Bjerrum, E. J. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022, 2022.

Jiao, R., Han, J., Huang, W., Rong, Y., and Liu, Y. Energy-motivated equivariant pretraining for 3d molecular graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8096–8104, 2023.

Jing, B., Corso, G., Chang, J., Barzilay, R., and Jaakkola, T. Torsional diffusion for molecular conformer generation. *Advances in Neural Information Processing Systems*, 35: 24240–24253, 2022.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B., et al. Pubchem 2023 update. *Nucleic acids research*, 51(D1): D1373–D1380, 2023.

Klug, T. and Heckel, R. Scaling laws for deep learning based image reconstruction. *arXiv preprint arXiv:2209.13435*, 2022.

Landrum, G. Rdkit: open-source cheminformatics http://www. rdkit. org. *Google Scholar There is no corresponding record for this reference*, 3(8), 2016.

Li, H., Zhao, D., and Zeng, J. Kpgt: knowledge-guided pre-training of graph transformer for molecular property prediction. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 857–867, 2022a.

Li, J., Liu, Y., Fan, W., Wei, X.-Y., Liu, H., Tang, J., and Li, Q. Empowering molecule discovery for molecule-caption translation with large language models: A chatgpt perspective. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Li, S., Zhou, J., Xu, T., Dou, D., and Xiong, H. Geomgcl: Geometric graph contrastive learning for molecular property prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 4541–4549, 2022b.

Luong, K.-D. and Singh, A. K. Fragment-based pretraining and finetuning on molecular graphs. *Advances in Neural Information Processing Systems*, 36, 2024.

Morgan, H. L. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of chemical documentation*, 5(2):107–113, 1965.

Neumann, O. and Gros, C. Scaling laws for a multi-agent reinforcement learning model. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.

Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., and Huang, J. Grover: Self-supervised message passing transformer on large-scale molecular data. *arXiv preprint arXiv:2007.02835*, 2(3):17, 2020a.

Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., and Huang, J. Self-supervised graph transformer on large-scale molecular data. *Advances in neural information processing systems*, 33:12559–12571, 2020b.

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

Ruddigkeit, L., Van Deursen, R., Blum, L. C., and Reymond, J.-L. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of chemical information and modeling*, 52(11):2864–2875, 2012.

Schütt, K., Kindermans, P.-J., Sauceda Felix, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

Schwaller, P., Laino, T., Gaudin, T., Bolgar, P., Hunter, C. A., Bekas, C., and Lee, A. A. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS central science*, 5(9):1572–1583, 2019.

Seidl, P., Vall, A., Hochreiter, S., and Klambauer, G. Enhancing activity prediction models in drug discovery with the ability to understand human language. In *International Conference on Machine Learning*, pp. 30458–30490. PMLR, 2023.

Tingle, B. I., Tang, K. G., Castanon, M., Gutierrez, J. J., Khurelbaatar, M., Dandarchuluun, C., Moroz, Y. S., and Irwin, J. J. Zinc-22 a free multi-billion-scale database of tangible compounds for ligand discovery. *Journal of chemical information and modeling*, 63(4):1166–1176, 2023.

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

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Wang, H., Wu, Z., Liu, Z., Cai, H., Zhu, L., Gan, C., and Han, S. Hat: Hardware-aware transformers for efficient natural language processing. *arXiv preprint arXiv:2005.14187*, 2020.

Wang, Y., Wang, J., Cao, Z., and Barati Farimani, A. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022.

Wang, Y., Xu, C., Li, Z., and Barati Farimani, A. Denoise pretraining on nonequilibrium molecules for accurate and transferable neural potentials. *Journal of Chemical Theory and Computation*, 19(15):5077–5087, 2023.

Wang, Z., Dai, Z., Póczos, B., and Carbonell, J. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11293–11302, 2019.

Wu, F., Radev, D., and Li, S. Z. Molformer: Motif-based transformer on 3d heterogeneous molecular graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 5312–5320, 2023a.

Wu, Y., Ni, X., Wang, Z., and Feng, W. Enhancing drug property prediction with dual-channel transfer learning based on molecular fragment. *BMC bioinformatics*, 24 (1):293, 2023b.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Xia, J., Zhu, Y., Du, Y., and Li, S. Z. A systematic survey of chemical pre-trained models. *arXiv preprint arXiv:2210.16484*, 2022.

Xu, Z., Luo, Y., Zhang, X., Xu, X., Xie, Y., Liu, M., Dickerson, K. A., Deng, C., Nakata, M., and Ji, S. Molecule3d: A benchmark for predicting 3d geometries from molecular graphs. 2021c. In *URL https://openreview. net/forum*.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

Yüksel, A., Ulusoy, E., Ünlü, A., and Doğan, T. Selformer: molecular representation learning via selfies language models. *Machine Learning: Science and Technology*, 4 (2):025035, 2023.

Zaidi, S., Schaarschmidt, M., Martens, J., Kim, H., Teh, Y. W., Sanchez-Gonzalez, A., Battaglia, P., Pascanu, R., and Godwin, J. Pre-training via denoising for molecular property prediction. *arXiv preprint arXiv:2206.00133*, 2022.

Zeng, Z., Yao, Y., Liu, Z., and Sun, M. A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. *Nature communications*, 13(1):862, 2022.

Zhang, Z., Xie, A., Guan, J., and Zhou, S. Molecular property prediction by semantic-invariant contrastive learning. *Bioinformatics*, 39(8):btad462, 2023.

Zhou, G., Gao, Z., Wei, Z., Zheng, H., and Ke, G. Do deep learning methods really perform better in molecular conformation generation? *arXiv preprint arXiv:2302.07061*, 2023.

Zhu, Y., Chen, D., Du, Y., Wang, Y., Liu, Q., and Wu, S. Improving molecular pretraining with complementary featurizations. *arXiv preprint arXiv:2209.15101*, 2022.

A. Supplemental Materials

This section provides supplementing tables and plots for the main paper's findings.

A.1. Pre-training Perplexity Results

Figure 7 illustrates the perplexity over pre-training tasks as the number of tokens increases. It shows that larger models generally achieve lower perplexity as more tokens are observed, indicating better performance in downstream tasks with lower pre-training loss. Initial results for fine-tuning the 230M model on pre-training tasks show promising trends, though improvements cannot be fully observed in early checkpoints. Final results for pre-training and downstream performance for 230M and 700M models trained on 200B tokens will be reported in future work.

A.2. Smiles2IUPAC Pre-training Task Results

In figure 8 we show how the accuracy of IUPAC generation improves as more data is processed during training. Although initial IUPAC names are valid and can be converted to SMILES, they do not match the ground truth until much later in the training process.

A.3. Performance on Downstream Regression Tasks

In table 3 we display the model performance on various regression tasks for different model sizes and checkpoints. Lower RMSE values indicate better performance. The results highlight the impact of different model sizes and the amount of pre-training on the regression tasks.

Table 3: Model performance on various regression tasks for different model sizes and checkpoints. Lower is better (↓).

Model Size	Checkpoint	FreeSolv ↓	Lipo ↓	QM7 ↓	QM8 ↓	QM9 ↓	ESOL ↓
stp-t5-12m	Baseline	3.593	1.114	170.4	0.02962	0.01843	1.624
	6M Tokens	2.7720	0.9955	131.6422	0.0281	0.0171	1.1353
	24M Tokens	3.0146	0.9904	121.2261	0.0283	0.0176	1.0062
	42M Tokens	2.7413	0.9704	114.9021	0.0279	0.0176	0.9882
	60M Tokens	2.7565	0.9699	115.8992	0.0282	0.0178	0.9693
	300M Tokens	2.7097	0.9161	112.0384	0.0296	0.0181	0.9017
	2.1B Tokens	2.4624	0.8749	114.1018	0.0300	0.0189	0.9644
	3.6B Tokens	2.4694	0.8749	103.9951	0.0300	0.0190	0.9334
	5.7B Tokens	**2.2439**	0.8669	105.1660	0.0298	0.0193	0.9369
	7.8B Tokens	2.2764	0.8826	107.7765	0.0307	0.0197	0.9387
	11.4B Tokens	2.3650	0.8897	113.6905	0.0314	0.0204	0.9137
stp-t5-45m	Baseline	3.366	1.099	162.1	0.02922	0.01815	1.529
	6M Tokens	2.6180	0.9959	87.8506	0.0271	0.0173	1.1184
	24M Tokens	2.7270	0.9956	89.1413	0.0275	0.0179	0.9909
	42M Tokens	2.8634	0.9909	105.1391	0.0273	0.0178	0.9803
	60M Tokens	2.7761	0.9810	111.4210	0.0276	0.0176	0.9499
	300M Tokens	2.7699	0.9690	112.3831	0.0269	**0.0171**	0.9235
	2.1B Tokens	2.5550	0.8484	96.5613	0.0272	0.0175	**0.8673**
	3.6B Tokens	2.6181	0.8299	85.5782	0.0267	0.0177	0.9228
	5.7B Tokens	2.4683	0.8231	88.2237	0.0274	0.0175	0.8879
	7.8B Tokens	2.3794	0.8146	87.8312	0.0273	0.0169	0.8950
	11.4B Tokens	2.4145	**0.7882**	85.9396	0.0277	0.0168	0.9308
stp-t5-230m	Baseline	2.659	1.172	175.7	0.02915	0.01831	1.615
	6M Tokens	2.7598	0.9813	85.3427	**0.0262**	0.0182	1.1301
	24M Tokens	3.2061	0.9959	**73.5555**	0.0271	0.0177	1.0505
	42M Tokens	3.2342	1.0123	75.0612	0.0281	0.0174	1.0194
	60M Tokens	3.0928	0.9495	80.1895	0.0273	0.0173	0.9734
	300M Tokens	2.6273	0.9622	99.2142	0.0289	0.0177	0.9881

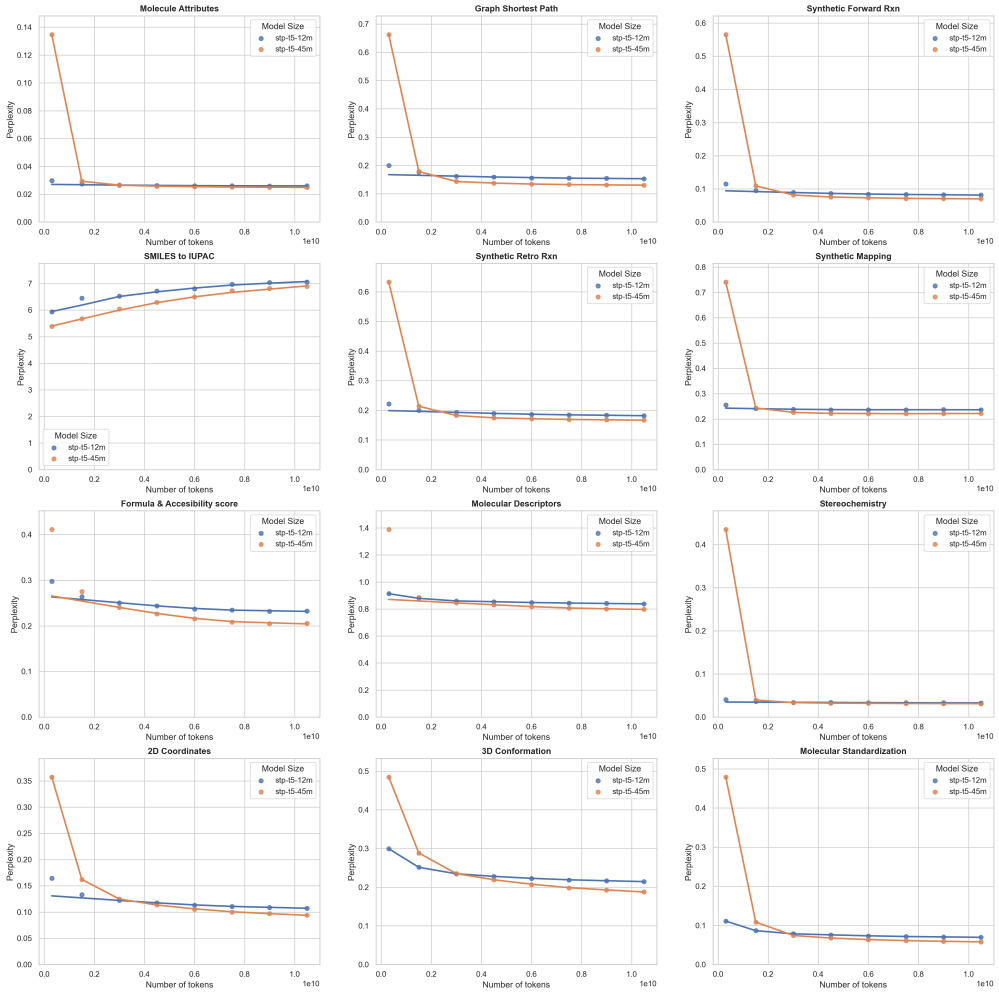

Figure 7: Perplexity over pre-training tasks as the number of tokens increases. Values are reported for 12M and 45M models, while results for the 230M parameters model are still in progress (Trained for 300M tokens, downstream results shown in table 1 & 3).

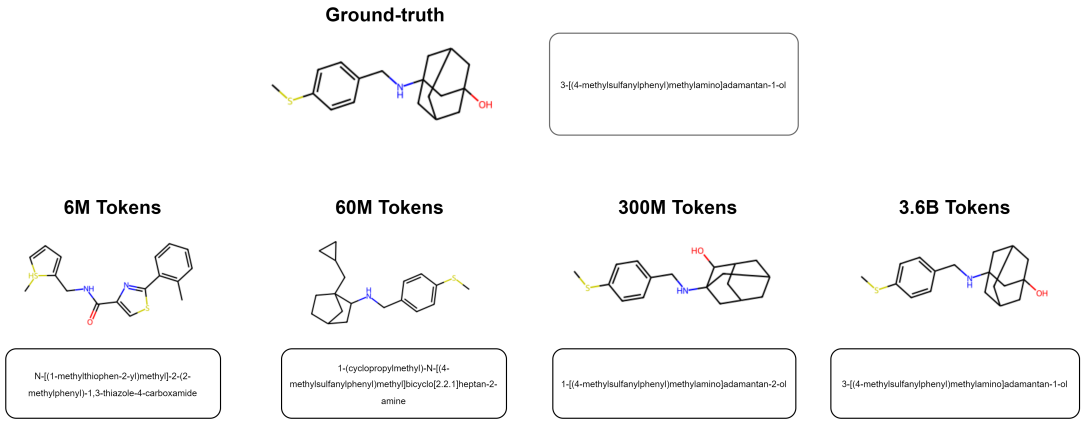

Figure 8: An example of how IUPAC generation becomes increasingly accurate as more data is processed during training. Although initial IUPAC names are valid and can be converted to SMILES, they do not match the ground truth until much later in the training process.