Enhancing Low-Cost Molecular Property Prediction with Contrastive Learning on SMILES Representations

Marcos G. Quiles[1(✉)], Piero A. L. Ribeiro[1], Gabriel A. Pinheiro[1], Ronaldo C. Prati[2], and Juarez L. F. da Silva[3]

[1] Institute of Science and Technology, Federal University of São Paulo (UNIFESP), São José Dos Campos, SP, Brazil
{quiles,piero.andreeto,gabriel.pinheiro}@unifesp.br

[2] Center for Mathematics, Computing and Cognition, Federal University of ABC (UFABC), Santo André, São Paulo, Brazil
ronaldo.prati@ufabc.edu.br

[3] São Carlos Institute of Chemistry, University of São Paulo, P.O. Box 780, 13560-970 São Carlos, SP, Brazil
juarez_dasilva@iqsc.usp.br

Abstract. This paper explores self-supervised contrastive learning techniques in the Simplified Molecular Input Line Entry System (SMILES) representations. The contrastive model is responsible for pre-training the model before its downstream application on molecular property prediction. Central to our approach is the use of contrastive learning paradigms, in which pairs of distinct SMILES corresponding to the same molecule are treated as positive samples. In contrast, negative samples comprise pairs of SMILES that represent distinct molecules. This methodology aims to enrich the model's understanding of molecular structures by emphasizing the invariance of molecular properties despite permutation in the SMILES representation, thereby enhancing the robustness and predictive accuracy of the model. We conducted a series of experiments on publicly available and proprietary datasets to evaluate the efficacy of our approach. Empirical results underscore the merits of the contrastive learning model over traditional methods, demonstrating improvements in molecular property prediction tasks. This study not only validates the potential of contrastive learning in the realm of chemoinformatics, but also sets a promising direction for future research in leveraging SMILES permutations for molecular analysis.

Keywords: Contrastive Learning · SMILES · Molecular Property Prediction

1 Introduction

Accurate prediction of molecular properties is the cornerstone of numerous scientific endeavors, ranging from the design of new materials to the discovery of

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
O. Gervasi et al. (Eds.): ICCSA 2024 Workshops, LNCS 14823, pp. 387–401, 2024.
https://doi.org/10.1007/978-3-031-65329-2_26

groundbreaking drugs [1]. These predictions allow researchers and developers to traverse the vast chemical space by identifying compounds with desirable properties for specific technological applications [2–4]. However, traditional methods for predicting molecular properties, such as density functional theory (DFT) and wave function methods, often face limitations due to the high computational cost and high dimensionality of the chemical space [5].

Traditionally, the determination of molecular properties has been based on experimental techniques and computational approaches such as the modern computational implementation of DFT [6]. Although experimental techniques provide a route for the characterization of the physical-chemical properties, they are often labor intensive, time consuming, and expensive. Furthermore, experimental techniques face challenges to yield an atom-level understanding of the interrelation between properties and structure. Similarly, despite providing a high degree of accuracy, DFT and other quantum mechanical calculations demand substantial computational resources, making them impractical for large-scale screenings, particularly assuming that the chemical space is practically infinite [7,8]. Consequently, while these traditional approaches remain valuable, their high costs and resource requirements present significant barriers to the rapid discovery and development of novel materials.

Machine learning (ML) has emerged as a powerful tool to bridge this gap, offering the potential to predict molecular properties quickly and accurately [5]. Several methods have been proposed in this scenario, particularly in the supervised paradigm, where molecules with target properties are used to train ML models [9–14]. For this purpose, different molecular descriptors are employed to represent chemical structures numerically, allowing ML algorithms to learn patterns and relationships between molecular features and properties. These descriptors range from simple fingerprints and text representation, such as the ECFPs (Extended-Connectivity Fingerprints) and the SMILES (Simplified Molecular Input Line Entry System), to more complex descriptors based on graph theory or quantum chemistry calculations, such as SOAP (Smooth Overlap of Atomic Positions) and MBTR (Many-body Tensor Representation) [11,15,16].

Despite the molecular representation, the properties are commonly obtained via DFT calculations, which can still be computationally expensive. For example, in [12] the authors have used a molecular descriptor extracted from SMILES using the Mordred toolkit [17] to build neural network predictors. Advanced descriptors, such as SOAP and MBTR, have been considered in other studies [13]. In both studies, the labeled data were generated by DFT calculations.

However, the effectiveness of these models is highly dependent on the quality and quantity of training data, which are often scarce or expensive to obtain in this domain [18]. Unsupervised and self-supervised learning approaches have gained attention as a means to overcome the limitations of supervised learning in the prediction of molecular properties and other downstream tasks [18–20]. In this case, an ML model is pre-trained on large datasets of unlabeled molecules and learns to capture underlying patterns and representations without explicit property labels. This pre-training step allows the model to develop a rich

understanding of molecular structures and their relationships, which can then be fine-tuned using smaller labeled datasets to make specific property predictions.

Contrastive learning, which belongs to the self-supervised learning paradigm, has shown remarkable success in domains such as Natural Language Processing (NLP) and Computer Vision [21–24]. By learning to differentiate between similar and dissimilar instances, contrastive learning facilitates the development of models that can capture complex patterns and relationships in the data, even without direct supervision. Taking into account the downstream property prediction task, in [25], the authors have proposed MolCLR, a framework based on GNNs (Graph Neural Networks). In [19], instead of using only molecular graphs, the authors proposed SmiCLR, which incorporates SMILES representations together with graphs. The SmiCLR relies on a contrastive model with two branches, one modeled by a graph neural network and the other by an LSTM (Long Short-Term Memory, to jointly learn representations from molecular graphs and SMILES strings. Overall, the use of contrastive learning techniques, such as MolCLR and SmiCLR, shows promise in enhancing molecular property prediction models by leveraging large amounts of unlabeled data to learn rich representations of molecular structures and relationships.

Here, we propose a more straightforward approach that takes only the SMILES of the molecule in the pre-training phase, thus reducing the cost of obtaining the molecular graph, atom, and bond descriptors and the geometry of the molecules. Our model learns a robust representation of molecular structures by treating pairs of distinct SMILES (permuted SMILES) that represent the same molecule as positive samples and pairs of SMILES from different molecules as negative samples. Subsequently, we deploy this pre-trained model in a downstream task to predict molecular properties, demonstrating the efficacy of our approach through computer experiments on both public and private datasets.

Our findings reveal that, compared to traditional approaches, the contrastive learning method with permuted SMILES enhances the model's performance in predicting molecular properties. Furthermore, another advantage is the elimination of the SMILES canonization process, as our model is designed to inherently recognize and interpret permuted SMILES as equivalent representations, streamlining the process and reducing preprocessing requirements. Thus, our contrastive model paves the way for more efficient and cost-effective methods for screening large chemical datasets and allows for faster exploration of the chemical space.

Our main contributions are:

- Proposing a contrastive learning approach using only the SMILES representations for pre-training a molecular property prediction model;
- This study shows the potential of contrastive learning with permuted SMILES to improve the accuracy and efficiency of screening large chemical datasets;
- We demonstrated the effectiveness of our approach through experiments on public and private datasets.

The structure of this manuscript is delineated as follows: Sect 2 elucidates the core principles that underlie this study. The methodological framework is

described in Sect. 3. Computational experiments and their corresponding discussions are presented in Sect. 4. Ultimately, Sect. 5 synthesizes the terminal insights and articulates various outstanding questions that warrant further academic exploration.

2 Background

2.1 SMILES Representation

SMILES is a notation for encoding molecular structures in a string [15]. Developed to facilitate the digital storage, retrieval, and processing of chemical information, SMILES notation represents the structure of a molecule using a series of characters including letters, numbers, and symbols. Each character or combination thereof corresponds to atoms, bonds, or specific configurations within the molecule, making SMILES a compact and versatile language for chemoinformatics [26].

The SMILES representation captures the essence of a molecule's connectivity, allowing for the depiction of complex organic structures in a straightforward and readable format. This includes the representation of cyclic structures, branching, and stereochemistry. One of the key advantages and also a disadvantage of SMILES is its ability to represent a molecule in multiple valid ways, due to permutations of chemical information within its string representation [15]. Therefore, distinct SMILES can represent the same molecule and the SMILES is not invariant to permutations. However, this property is particularly relevant for computational analyses, where the diversity of representations can be leveraged to enhance model robustness and generalizability. This process can be understood as a data augmentation method [27], in which multiple samples are generated from an available pattern, an approach analogous to rotations and translations in image classification. Figure 1 shows an illustration of the Caffeine molecule using a graph representation and also two possible SMILES that represent this compound.

The use of SMILES notation extends beyond mere structural depiction, serving as a foundation for numerous computational chemistry and molecular modeling tasks. It facilitates the application of machine learning algorithms by converting intricate molecular geometries into a form that can be easily processed by neural networks and other computational natural language processing models [28,29]. In particular, the linear nature of SMILES strings makes them amenable to processing by sequential models, such as Long Short-Term Memory (LSTM) networks, which can capture short and long dependencies and relationships within the molecular structure for various predictive and generative tasks [19].

In summary, SMILES is a powerful representation in the field of chemoinformatics, offering a concise and flexible method for encoding molecular information with low computational cost, because it does not require the geometry optimization of molecules via quantum chemistry simulations. Furthermore, its compatibility with computational and machine learning techniques has made it

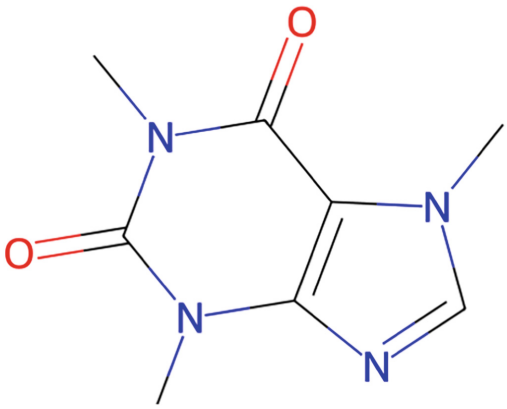


Fig. 1. Representation of the Caffeine molecule. The molecular graph representation and two possible SMILES of the caffeine are depicted on the top and bottom, respectively.

an indispensable component of modern approaches to molecular property prediction, drug discovery, and materials science.

2.2 Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are a special class of Recurrent Neural Networks (RNNs) designed to address the limitations of traditional RNNs in capturing long-range dependencies within sequential data. Introduced by Hochreiter and Schmidhuber in [30], and extended by Gers et al. in [31], LSTMs have emerged as a pivotal technology in the field of deep learning, particularly for tasks involving sequences such as time series prediction, natural language processing, and more recently chemoinformatics.

The core of LSTM networks lies in their architecture, which incorporates memory cells and multiple gates (input, output, and forget gates). These components work in concert to regulate the flow of information, allowing the network to remember or forget information over long periods. This capability enables LSTMs to learn and retain complex dependencies within the data, a feature especially beneficial for processing sequential information such as the SMILES and other string-based representations.

In the context of chemoinformatics, LSTM networks have demonstrated significant potential in modeling and predicting molecular properties [19,28]. By treating SMILES strings as sequences, LSTMs can capture the intricate patterns and relationships encoded within the molecular structure, facilitating

accurate predictions of various chemical and physical properties. The adaptability of LSTM networks to handle variable-length sequences makes them particularly suited for dealing with the diverse and complex nature of chemical compounds represented in the SMILES format.

Moreover, LSTMs, as long as their competitors, such as GRU (Gated Recurrent Unit), have been instrumental in advancing the field of molecule generation and property prediction [14]. By learning the underlying rules and patterns in molecular structures, LSTM-based models can generate novel compounds with desired properties, a task of great importance in drug discovery and materials science [14].

2.3 Contrastive Learning

Contrastive learning is a form of self-supervised learning that teaches a model to understand which data points are similar or different without relying on explicit labels [32]. This approach is particularly useful for learning meaningful representations from unlabeled data, which can then be fine-tuned for a wide range of specific tasks [19].

In contrastive learning, the model is trained to distinguish between pairs of similar ("positive") and dissimilar ("negative") inputs. For example, in image processing, a positive pair could consist of two different augmentations (e.g., cropping, resizing, and color jittering) of the same image, while a negative pair would consist of augmentations from two different images [33]. The goal is to learn to project similar items closer together and dissimilar items farther apart in the embedding space [21]. This is typically achieved by minimizing a contrastive loss function, such as triplet loss [34] or the Noise Contrastive Estimation (NCE) loss [32].

A contrastive model can simultaneously use different types of input within a single model framework [35]. For example, one branch of the contrastive model could process the SMILES representation of a molecule, while another branch could handle the corresponding molecular graph [19]. Similarly, in the domain of image processing, one branch could receive an image, while another could process a textual description of that image. The choice of neural network architecture is crucial to effectively manage and learn from these varied data types. This study focuses on the Long Short-Term Memory (LSTM) architecture as it can straightforwardly handle sequential data.

3 Methodology

3.1 Datasets

In this study, three datasets were considered, two publicly available datasets (ZINC and QM9), and an in-house dataset (ionic liquid molecules). Their descriptions are depicted below:

ZINC : This dataset is a collection of commercially available compounds. It includes SMILES and XYZ representations, along with a range of calculated properties. The dataset contains 230 million molecules [36]. From the ZINC dataset, we select a subset of approximately 1.3 million molecules with atom types C, H, O, N, and F, limiting the number of heavy atoms to 14.

QM9 : This is a subset of the GDB-17 database [37], consisting of 133,385 organic molecules with up to nine heavy atoms from four chemical species: C, O, N, and F. This dataset has previously been used as a benchmark for several machine learning studies in quantum chemistry applications. It provides two textual molecular representations (SMILES and InChI) and the XYZ files, which represents the molecule's optimized geometry via DFT (B3LYP/6-31G(2df,p)). Additionally, a set of 15 molecular properties (including geometric, energetic, electronic, and thermodynamic properties) is also presented [38].

Ionic Liquid Molecules This dataset, developed by the QTNano group [39], comprises approximately 20,000 molecules (ions) and was generated through the combination of various cations (138) and anions (40) extracted from known literature. These molecules were generated by combining different cores and backbones derived from the selected cations and anions. The generated compounds were optimized using DFT (FHI-AIMS), and their HOMO and LUMO properties were calculated. These properties are crucial for estimating the electrochemical window of the ionic liquid, which is of paramount importance for the development of batteries and supercapacitors.

The ZINC dataset were mainly used as a pre-training dataset. The QM9 and the Ionic Liquids dataset were also considered in the molecular property prediction tasks.

3.2 Positive and Negative Pairs

Contrastive learning is based on the effective differentiation between positive and negative pairs of examples to learn meaningful representations. Here, we detail the methodology for generating these pairs.

Positive Pair Generation: Positive pairs are formed by using the inherent property of SMILES strings, where a single molecule can be represented by multiple valid SMILES notations because of the different sequences in which atoms can be traversed and written down. These variations, while representing the same molecular structure, provide an adequate opportunity to create positive samples for our contrastive learning model.

Given a molecule M, let $S(M)$ represent the set of all possible valid SMILES representations for M. A positive pair (s_i, s_j) is then defined as two distinct SMILES strings $s_i, s_j \in S(M)$, where $i \neq j$. Formally, this can be represented as:

$$(s_i, s_j) \in P \Leftrightarrow s_i, s_j \in S(M) \wedge s_i \neq s_j, \tag{1}$$

where P denotes the set of all positive pairs. Generation of $S(M)$ for a given molecule involves permuting the SMILES string to create different valid representations, ensuring that s_i and s_j are distinct, yet describe the same molecular entity. The RDKit package [40] was used for the permutation of SMILES.

Negative Pair Generation. Negative pairs are crucial for teaching the model to distinguish between different molecular structures. A negative pair consists of two SMILES strings that represent different molecules. Given two molecules M_1 and M_2 with their respective SMILES representations s_{M_1} and s_{M_2}, a negative pair is formed if and only if $M_1 \neq M_2$. Formally, a negative pair (s_{M_1}, s_{M_2}) is defined as:

$$(s_{M_1}, s_{M_2}) \in N \Leftrightarrow M_1 \neq M_2, \tag{2}$$

where N denotes the set of all negative pairs. This definition ensures that the model is exposed to a wide variety of molecular structures during training.

Pair Selection Strategy. The selection of pairs, both positive and negative, is critical for the learning process. For positive pairs, we generate two random permutations s_j from s_i of the SMILES of the same molecule, ensuring a diverse set of representations for the same molecule. For negative pairs, we randomly select SMILES strings from two distinct molecules. The balance between positive and negative pairs in the training set is crucial for effective contrastive learning. An imbalance can lead to a model that is biased towards identifying either similarities or differences, thus affecting its generalization capability. Therefore, we maintain a similar number of positive and negative pairs in our training dataset to ensure a balanced learning environment.

3.3 SMILES Tokenization

Neural networks require numerical input to operate, thus necessitating a tokenization process to convert SMILES strings into a form these models can process. This tokenization involves breaking down each SMILES string into its constituent tokens, typically representing atoms, bonds, or structural motifs. Our tokenization approach is meticulously designed to capture the details of the molecular structure, thereby facilitating the model's ability to learn from and recognize variations in molecular representations.

The tokenization process begins by identifying all valid tokens within the SMILES syntax, including single-character elements (e.g., 'C' for carbon, 'O' for oxygen), multi-character elements (e.g., 'Cl' for chlorine, 'Br' for bromine), and special characters denoting bond types or branching (e.g., '=', '#', '(',')'). In addition, ring closures are marked by digits that follow the atoms involved in the ring. Each token is then mapped to a unique identifier in our model's embedding space, enabling efficient numerical processing during the learning phase. For this project, we used the BasicSmilesTokenizer routine available in the DeepChem toolbox [41], with a modified regular expression (regex) designed to include all species present in our datasets.

3.4 Self-supervised Contrastive Pre-training

The pre-training phase of our contrastive model is designed to learn robust representations of molecular structures encoded as SMILES strings. This phase is critical for capturing the patterns and relationships inherent in the chemical space. Our model architecture employs two LSTM networks, which are particularly suited for processing the sequential nature of SMILES strings. This section outlines the architecture, the embedding process, and the training methodology of our contrastive learning model.

Model Architecture. Given a pair of SMILES strings (s_i, s_j), whether it forms a positive or negative pair, each string in the pair is independently processed by two LSTM networks. The two LSTM networks have identical architectures, but with their own set of weights. The output of each LSTM, for a given SMILES string s, is a latent representation $h_s \in \mathbb{R}^d$, where d is the dimensionality of the LSTM output.

Following extraction of latent representations h_{s_i} and h_{s_j} by the LSTMs, both are passed through a shared projection head. This projection head is an MLP (Multilayer Perceptron) neural network layer that maps the latent space h to an embedding space z, where contrastive loss is applied. The projection head is defined by a function $f : \mathbb{R}^d \to \mathbb{R}^p$, $p > d$, resulting in embeddings $z_{s_i} = f(h_{s_i})$ and $z_{s_j} = f(h_{s_j})$. The use of a shared projection head for both LSTMs ensures that the embeddings are comparable and lie in the same space, facilitating the effective application of contrastive loss.

To train our model, we employ the Normalized Temperature-Scaled Cross-Entropy Loss (NT-Xent Loss), which is widely used in contrastive learning for its effectiveness in learning discriminative features. The NT-Xent Loss for a pair of embeddings (z_{s_i}, z_{s_j}) is defined as:

$$\ell(z_{s_i}, z_{s_j}) = -\log \frac{\exp(\text{sim}(z_{s_i}, z_{s_j})/\tau)}{\sum_{k=1}^{2N} 1_{[k \neq i]} \exp(\text{sim}(z_{s_i}, z_{s_k})/\tau)}, \tag{3}$$

where $\text{sim}(u, v) = u^\top v / \|u\| \|v\|$ denotes the cosine similarity between two vectors, τ represents the temperature parameter, $2N$ is the total number of examples in the batch, half of them are samples generated from enumerated smiles from the dataset, and $1_{[k \neq i]}$ is an indicator function that equals 1 iff $k \neq i$. This loss function forces the model to produce similar embeddings for positive pairs and dissimilar embeddings for negative pairs.

Optimization and Training. For the optimization of our model, we chose the Adam optimizer. The Adam optimizer dynamically adjusts the learning rate for each parameter, based on estimates of the first and second moments of the gradients, improving the convergence rate and stability of the training process. The pre-training process involves iteratively sampling pairs of SMILES strings, processing them through the LSTMs and the projection head, and updating the

model weights based on the computed NT-Xent Loss. This iterative process continues until the model converges or a predefined number of epochs is reached. By the end of pre-training, the model is expected to have learned a rich and discriminative embedding space z, which can be leveraged for downstream molecular property prediction tasks.

3.5 Molecular Property Prediction

Upon completing the pre-training phase, the learned embeddings serve as input to a predictor neural network. For this purpose, we also use an MLP neural network, which is designed to map the embedding representation z to the target property y.

Given a labeled dataset where each molecule is associated with a specific property, the MLP is trained to minimize the prediction error between the estimated property values and the true values. In the learning phase, the predictor takes into account the mean squared error loss function. Then, the errors are backpropagated with the Adam optimizer.

The integration of the predictor neural network with the pre-trained contrastive model is straightforward. The output embedding z from the pre-trained model serves as the input feature vector for the MLP. This approach allows the predictor to utilize the knowledge distilled from the contrastive learning phase, enhancing its predictive performance on the target molecular property.

4 Experiments and Discussion

This section offers an overview of our neural models and the computer experiments conducted for the property prediction task.

4.1 Experimental Setup

To train, evaluate and report our results, we divided the datasets into training, validation, and test sets. In the supervised phase, we randomly selected a subset of 5k molecules from QM9 as the training set and two additional subsets of 10k each for validation and testing. For the cation (anion) datasets from the ionic liquid database, we defined test and validation sets of 900 (750) samples, respectively, with the remainder being used as the training set. During the self-supervised phase, we used 100k molecules from ZINC data as the validation set and assigned the rest for training. Moreover, during the self-supervised phase, we maintained the same validation and test sets as in the supervised phase and used the remaining data for training for both the QM9 and ionic liquid datasets.

The hyperparameters were empirically set to optimize the performance of the models. Specifically, the following parameters and computational infrastructure were considered in the experiments:

Computational Infrastructure: The learning phase and experiments were conducted on a high-performance workstation equipped with Dual 12-core Intel Xeon Silver 4214R processors (16.5 MB cache, 12 cores, 24 threads, up to 3.50 GHz Turbo), 256GB of RAM, and 2 NVidia RTX A6000 GPUs, each with 48GB of memory;

Optimization Parameters: The models were trained for a maximum of 100 epochs with a learning rate of $1e-3$ and no learning rate decay. The temperature was set to 0.1, and the batch size was set to 512 for the self-supervised phase. During the supervised phase, we set the number of epochs to 300 and the batch size to 256, with learning rate decay enabled;

Baseline Model Hyperparameters: The baseline model, a three-layers bidirectional Long Short-Term Memory (LSTM) network, was configured with 128 neurons. This setup was chosen to establish a foundational comparison for the performance of more complex models;

Contrastive Model Hyperparameters: After empirical experiments, we selected embedding dimensions of 64, and three LSTM layers with 128 neurons each;

Prediction Model Hyperparameters: The prediction model was designed with a 2-layer Multilayer Perceptron (MLP) neural network, with 64 neurons in the hidden layer using the hyperbolic tangent activation function and a single linear neuron for the output. This predictive head was used by both the baseline and the contrastive models.

4.2 Property Prediction

We designed four distinct tests to evaluate the performance of our models under varying conditions. Initially, a baseline LSTM model served as our benchmark, establishing a performance metric for comparison. Subsequently, we explored the impact of pre-training our contrastive model exclusively with the downstream datasets, namely QM9, to assess the direct benefits of contrastive learning in a simpler context with no external dataset. The third test, which was applied for both the QM9 and Ionic Liquids datasets, we expanded the pre-training scope by incorporating a substantial subset of the ZINC dataset, approximately 1.3 million molecules with atom types C, H, O, N, and F, limiting the number of heavy atoms to 14. This test aimed to understand the effect of a larger chemical dataset on model performance.

Table 1 shows the results with the QM9 dataset. Table 2 shows the results for the Ionic Liquids dataset. It can be observed that the contrastive model delivers results superior to those achieved with the baseline model when solely using the QM9 for both the self-supervised and supervised phase. However, the results of the model pre-trained with the ZINC dataset did not perform as expected. For instance, even in delivering results superior to the baseline model, the contrastive trained with the ZINC+QM9 was inferior than the model trained with QM9 alone.

Table 1. Test error (MAE) of 8 molecular properties from the QM9. (*i*) A baseline LSTM model; (*ii*) Contrastive model pre-trained with the QM9 dataset; (*iii*) Contrastive model pre-trained with the QM9 and a subset of the ZINC dataset (Aprox. 1.3M molecules).

Property	*i*	*ii*	*iii*
U_0 (eV)	0.3053 ± 0.0049	0.2439 ± 0.0033	0.2787 ± 0.0016
U (eV)	0.3056 ± 0.0056	0.2436 ± 0.0058	0.2858 ± 0.0019
H (eV)	0.3298 ± 0.0020	0.2475 ± 0.0075	0.2885 ± 0.0078
G (eV)	0.2987 ± 0.0046	0.2380 ± 0.0055	0.2779 ± 0.0040
Cv (cal (molK)$^{-1}$)	0.2384 ± 0.0027	0.2032 ± 0.0032	0.2271 ± 0.0062
α (a_0^3)	0.6089 ± 0.0053	0.5203 ± 0.0136	0.5860 ± 0.0140
ϵ_{HOMO} (eV)	0.1592 ± 0.0019	0.1562 ± 0.0021	0.1603 ± 0.0024
Δ_ϵ (eV)	0.2349 ± 0.0020	0.2198 ± 0.0020	0.2298 ± 0.0037

Table 2. Test error (MAE) of 2 molecular properties fromthe Ionic Liquids dataset. (*i*) A baseline LSTM model; (*ii*) Contrastive model pre-trained with the ZINC subset

Property	(*i*)	(*ii*)
Anions		
ϵ_{HOMO} (eV)	0.1361 ± 0.0008	0.1330 ± 0.0035
ϵ_{LUMO} (eV)	0.1313 ± 0.0042	0.1337 ± 0.0034
Cations		
ϵ_{HOMO} (eV)	0.2086 ± 0.0055	0.2013 ± 0.0048
ϵ_{LUMO} (eV)	0.2019 ± 0.0036	0.2011 ± 0.0041

5 Conclusion

This study proposed a contrastive learning model that takes only the SMILES representation into account for the pre-training phase. The core of our approach lies in treating pairs of SMILES strings, representing the same molecule in its original form and a permuted variant, as positive samples and contrasting them with negative samples drawn from different molecules. This methodology was designed to exploit the inherent structural and property invariance of molecular representations, facilitating the learning of robust and discriminative features critical for accurate property prediction. Furthermore, as previously mentioned, by allowing the model to learn from permuted SMILES, we eliminate the need for a canonization process.

Our experiments, conducted using publicly available and proprietary datasets, have shown superior performance of our contrastive learning model over traditional approaches, albeit contrastive models that take the molecular graph into account still deliver superior results. These findings reinforce the value of

contrastive learning in chemoinformatics and also highlight the power of SMILES permutations to capture the nuanced complexities of molecular structures.

Interestingly, the model pre-trained with a combination of the QM9 and ZINC datasets performed worse than the model pre-trained with only the QM9 dataset. Moreover, the ionic liquids dataset results were not superior to those of the baseline models. These findings were contrary to expectations and warranted further scrutiny. Owning to the promising outcomes, this study opens several avenues for future research; for instance, more powerful neural network models, such as the Transformers, together with larger unlabeled datasets, could be used to improve the learned representation in the pre-training phase.

Acknowledgements. The authors gratefully acknowledge support from FAPESP (São Paulo Research Foundation), projects No. 2017/11631-2 and 2022/09285-7, and the CNPq (National Council for Scientific and Technological Development), project No. 313680/2021-3.

References

1. Shen, J., Nicolaou, C.A.: Molecular property prediction: recent trends in the era of artificial intelligence. Drug Discov. Today Technol. **32–33**, 29–36 (2019). https://doi.org/10.1016/j.ddtec.2020.05.001, https://www.sciencedirect.com/science/article/pii/S1740674920300032
2. Gómez-Bombarelli, R., et al.: Automatic chemical design using a data-driven continuous representation of molecules. ACS Central Sci. **4**(2), 268–276 (2018). https://doi.org/10.1021/acscentsci.7b00572, pMID: 29532027
3. Shen, C., Krenn, M., Eppel, S., Aspuru-Guzik, A.: Deep molecular dreaming: inverse machine learning for de-novo molecular design and interpretability with surjective representations. Mach. Learn. Sci. Technol. **2**(3), 03LT02 (2021). https://doi.org/10.1088/2632-2153/ac09d6
4. Back, S., et al.: Accelerated chemical science with AI. Digit. Disc. **3**, 23–33 (2024). https://doi.org/10.1039/D3DD00213F
5. Butler, K.T., Davies, D.W., Cartwright, H., Isayev, O., Walsh, A.: Machine learning for molecular and materials science. Nature **559**(7715), 547–555 (2018). https://doi.org/10.1038/s41586-018-0337-2
6. Wei, J., et al.: Machine learning in materials science. InfoMat **1**(3), 338–358 (2019). https://doi.org/10.1002/inf2.12028, https://onlinelibrary.wiley.com/doi/abs/10.1002/inf2.12028
7. Kirkpatrick, P., Ellis, C.: Chemical space. Nature **432**(7019), 823–823 (2004). https://doi.org/10.1038/432823a
8. Coley, C.W.: Defining and exploring chemical spaces. Trends Chem. **3**(2), 133–145 (2021). https://doi.org/10.1016/j.trechm.2020.11.004
9. Rupp, M., Tkatchenko, A., Müller, K.R., von Lilienfeld, O.A.: Fast and accurate modeling of molecular atomization energies with machine learning. Phys. Rev. Lett. **108**, 058301 (2012). https://doi.org/10.1103/PhysRevLett.108.058301
10. Nyshadham, C., et al.: Machine-learned multi-system surrogate models for materials prediction. NPJ Comput. Materials **5**(1), 51 (2019). https://doi.org/10.1038/s41524-019-0189-9

11. von Lilienfeld, O.A., Müller, K.R., Tkatchenko, A.: Exploring chemical compound space with quantum-based machine learning. Nat. Rev. Chem. **4**(7), 347–358 (2020). https://doi.org/10.1038/s41570-020-0189-9
12. Pinheiro, G.A., Mucelini, J., Soares, M.D., Prati, R.C., Da Silva, J.L.F., Quiles, M.G.: Machine learning prediction of nine molecular properties based on the SMILES representation of the QM9 quantum-chemistry dataset. J. Phys. Chem. A **124**(47), 9854–9866 (2020). https://doi.org/10.1021/acs.jpca.0c05969, pMID: 33174750
13. Cesar de Azevedo, L., Pinheiro, G.A., Quiles, M.G., Da Silva, J.L.F., Prati, R.C.: Systematic investigation of error distribution in machine learning algorithms applied to the quantum-chemistry qm9 data set using the bias and variance decomposition. J. Chem. Inf. Model. **61**(9), 4210–4223 (2021). https://doi.org/10.1021/acs.jcim.1c00503
14. Oliveira, A.F., Da Silva, J.L.F., Quiles, M.G.: Molecular property prediction and molecular design using a supervised grammar variational autoencoder. J. Chem. Inform. Model. **62**(4), 817–828 (2022). https://doi.org/10.1021/acs.jcim.1c01573, pMID: 35174705
15. Weininger, D.: Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. J. Chem. Inform. Comput. Sci. **28**(1), 31–36 (1988). https://doi.org/10.1021/ci00057a005
16. Langer, M.F., Goeßmann, A., Rupp, M.: Representations of molecules and materials for interpolation of quantum-mechanical simulations via machine learning. NPJ Comput. Mater. **8**(1), 41 (2022). https://doi.org/10.1038/s41524-022-00721-x
17. Moriwaki, H., Tian, Y.S., Kawashita, N., Takagi, T.: Mordred: a molecular descriptor calculator. J. Cheminformatics **10**(1), 4 (2018). https://doi.org/10.1186/s13321-018-0258-y
18. Liu, X., et al.: Self-supervised learning: generative or contrastive. IEEE Trans. Knowl. Data Eng. **35**(1), 857–876 (2021). https://doi.org/10.1109/TKDE.2021.3090866
19. Pinheiro, G.A., Da Silva, J.L.F., Quiles, M.G.: SMICLR: contrastive learning on multiple molecular representations for semi-supervised and unsupervised representation learning. J. Chem. Inform. Model. **62**, 3948–3960 (2022). Under Review
20. Xie, J., Wang, Y., Rao, J., Zheng, S., Yang, Y.: Self-supervised contrastive molecular representation learning with a chemical synthesis knowledge graph. J. Chem. Inf. Model. (2024). https://doi.org/10.1021/acs.jcim.4c00157
21. Wang, X., Zhang, R., Shen, C., Kong, T., Li, L.: Dense contrastive learning for self-supervised visual pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3024–3033 (2021)
22. Yang, J., et al.: Vision-language pre-training with triple contrastive learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15671–15680 (June 2022)
23. Le-Khac, P.H., Healy, G., Smeaton, A.F.: Contrastive representation learning: a framework and review. IEEE Access **8**, 193907–193934 (2020). https://doi.org/10.1109/ACCESS.2020.3031549
24. Rethmeier, N., Augenstein, I.: A primer on contrastive pretraining in language processing: methods, lessons learned, and perspectives. ACM Comput. Surv. **55**(10) (2023). https://doi.org/10.1145/3561970
25. Wang, Y., Wang, J., Cao, Z., Barati Farimani, A.: Molecular contrastive learning of representations via graph neural networks. Nature Mach. Intell. **4**(3), 279–287 (2022). https://doi.org/10.1038/s42256-022-00447-x

26. Elton, D.C., Boukouvalas, Z., Fuge, M.D., Chung, P.W.: Deep learning for molecular design-a review of the state of the art. Mol. Syst. Des. Eng. **4**, 828–849 (2019). https://doi.org/10.1039/C9ME00039A
27. Bjerrum, E.J.: Smiles enumeration as data augmentation for neural network modeling of molecules (2017)
28. Skinnider, M.A., Stacey, R.G., Wishart, D.S., Foster, L.J.: Chemical language models enable navigation in sparsely populated chemical space. Nature Mach. Intell. **3**(9), 759–770 (2021). https://doi.org/10.1038/s42256-021-00368-1
29. Flam-Shepherd, D., Zhu, K., Aspuru-Guzik, A.: Language models can learn complex molecular distributions. Nature Commun. **13**(1), 3293 (2022). https://doi.org/10.1038/s41467-022-30839-x
30. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
31. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM. Neural Comput. **12**(10), 2451–2471 (2000). https://doi.org/10.1162/089976600300015015
32. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607. PMLR (2020). https://doi.org/10.48550/ARXIV.2002.05709
33. Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D., Makedon, F.: A survey on contrastive self-supervised learning. Technologies **9**(1), 2 (2021)
34. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015)
35. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
36. Irwin, J.J., Shoichet, B.K.: Zinc - a free database of commercially available compounds for virtual screening. J. Chem. Inf. Model. **45**(1), 177–182 (2005). https://doi.org/10.1021/ci049714
37. Ruddigkeit, L., van Deursen, R., Blum, L.C., Reymond, J.L.: Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. J. Chem. Inform. Model. **52**(11), 2864–2875 (2012). https://doi.org/10.1021/ci300415d, pMID: 23088335
38. Ramakrishnan, R., Dral, P.O., Rupp, M., von Lilienfeld, O.A.: Quantum chemistry structures and properties of 134 kilo molecules. Sci. Data **1**(1), 140022 (2014). https://doi.org/10.1038/sdata.2014.22
39. Moraes, A.S., et al.: Screening of the role of the chemical structure in the electrochemical stability window of ionic liquids: DFT calculations combined with data mining. J. Chem. Inf. Model. **62**(19), 4702–4712 (2022). https://doi.org/10.1021/acs.jcim.2c00748
40. Landrum, G.: RDKit: open-source cheminformatics software (2016). https://github.com/rdkit/rdkit/
41. Ramsundar, B., Eastman, P., Walters, P., Pande, V., Leswing, K., Wu, Z.: Deep Learning for the Life Sciences. O'Reilly Media (2019). https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837