Comparison of unsupervised, supervised and semi-supervised chemical reaction embeddings for reaction classification

submitted in partial fulfillment for the degree of master of science

Yunlong Cheng
14949423

Master Information Studies
information systems
Faculty of Science
University of Amsterdam

Submitted on 2.06.2023

	UvA Supervisor	External Supervisor
Title, Name	Paul Groth	Kinga Szarkowska, Timur Madzhidov, Markus Schwörer
Affiliation	UvA Supervisor	External Supervisor
Email	p.groth@uva.nl	k.szarkowska@elsevier.com t.madzhidov@elsevier.com m.schwoerer@elsevier.com

Amsterdam Data Science

ELSEVIER

ABSTRACT

Classifying chemical reactions is crucial for various chemistry applications, as it helps with the organization and retrieval of chemical data. Traditional methods rely on static, expert-curated transformation rules, which can lead to misclassifications in noisy datasets. State-of-the-art data-driven approaches involve converting chemical reactions into vectors, commonly known as fingerprinting. In this study, we introduce a new BERT-based semi-supervised embedding method for reaction classification, known as Contrastive Fingerprint (CFP). This method uses contrastive learning based on the distance between reactions to fine-tune a pre-trained BERT model, originally trained only on the masked language modeling (MLM) task. This enhances the model's ability to distinguish between similar and dissimilar chemical reactions, improving the accuracy of reaction classification tasks.

Our evaluations on the Reaxys and USPTO 1k TPL datasets demonstrate the effectiveness of our approach. On the Reaxys dataset, using 1-nearest neighbour as the classification benchmark, CFP achieved an accuracy of 96.18%, surpassing the supervised BERT-based transformer model (RXNFP) at 95.79% and the unsupervised Differential Reaction Fingerprint (DRFP) at 92.69%. Additionally, CFP showed a 13% improvement in accuracy over embeddings generated using the pre-trained BERT model. Further analysis revealed that CFP significantly enhances the performance of pre-trained BERT at both the superclass and class levels, resulting in more accurate classification outcomes. On the USPTO 1k TPL dataset, the fine-tuned model improved classification accuracy by 15% compared to pre-trained BERT embeddings, further demonstrating the broad applicability of our distance-based contrastive learning fine-tuning method.

KEYWORDS

Chemical Computational Science, Contrastive Learning, BERT, Reaction Classification

GITHUB REPOSITORY

https://github.com/Yunlong-Elsevier/Reaction-Classification.git

1 INTRODUCTION

Reaxys [5], developed and maintained by Elsevier, is a comprehensive chemistry database encompassing a wide range of chemical information, including details on compounds, chemical reactions, and their properties. Classifying chemical reactions facilitates numerous applications for chemists, such as efficient database searches. However, reaction classifiers in Reaxys are based on static, expert-curated transformation rules, which makes them prone to misclassifications in noisy datasets, such as reactions mined from the literature [19]. In this project, we aim to explore methods for automatically assigning reaction classes to chemical reactions in Reaxys.

Automatic classification of common chemical reactions typically involves two steps [19]. The first step is embedding reactions as vectors (often referred to as 'fingerprints') in a metric space and then training a machine learning classifier on these reaction fingerprints, based on the assumption of class clustering [19].

In Schwaller et al.'s 2021 study [22], the embedding step was shown to be of great importance. Hence, this study proposes a novel BERT-based, semi-supervised embedding method for reaction classification tasks, fine-tuned using contrastive learning. We refer to our method as the Contrastive Fingerprint (CFP) for convenience in subsequent discussions. Specifically, we will compare this method with the Differential Reaction Fingerprint (DRFP) designed by Daniel et al. [19] and a BERT model pre-trained on the MLM (masked language modelling) task on the Reaxys database. Both of these approaches represent unsupervised embedding techniques. We will also compare our method with a supervised embedding approach tailored for chemical reaction tasks, the BERT-based transformer model (RXNFP), proposed by Schwaller et al. [23]. These comparisons will be conducted using the Reaxys database and public USPTO 1k TPL dataset to evaluate our proposed method's effectiveness and broad applicability.

This leads to the following research question:

RQ: To what extent can the CFP improve classification performance on the chemistry databases compared to other methods?

In addressing this question, we assess several subquestions:

- To what extent contrastive fine-tuning can improve reaction classification accuracy compared to embeddings pre-trained only on the MLM task?
- What are the benefits of our approach compared to other embedding generation methods like DRFP and RXNFP?
- Does the CFP exhibit broad applicability, and how does it perform when applied to the public USPTO 1k TPL dataset?

2 RELATED WORK

2.1 Background

Reaction classification serves multiple chemistry applications, efficiently organizing reactions within databases[15][21]. This classification enhances structure-based retrieval systems, promoting browsing strategies[11]. In reaction retrieval systems, it proves instrumental for managing extensive post-search results, streamlining query formation, integrating reaction data from various sources, and facilitating access to broad categories of information. It offers a foundational knowledge base for predicting reactions and designing synthetic strategies, enabling the prediction of novel reactions[11]. Additionally, it supports the establishment of automated processes for analysis, correlation, quality control, and overlap assessments[11].

Recent reviews[2] have categorized automatic reaction classification techniques into two main types: "model-driven" and "data-driven" approaches. The model-driven approach, requiring scientists to formulate classification principles and propose a predetermined classification model, limits the optimization of classification principles and is susceptible to misclassification in noisy datasets[19].

As increasingly larger relevant datasets become available, data-driven approaches have demonstrated progressively stronger performance in the field of chemical reaction classification[19]. This method involves first converting chemical reaction equations into computer-understandable expressions. The most commonly used format for this conversion is the Simplified Molecular Input Line Entry System (SMILES)[27]. SMILES converts the three-dimensional structure of a chemical substance into a string of symbols, making it easily interpretable by computer software for model input. By leveraging machine learning models, computers can automatically analyze sets of reactions and generate classification results. Unlike traditional methods that derive physical laws from theoretical principles, this data-driven approach establishes numerical connections between accessible variables and attributes related to the dataset's creation[24]. This method enhances robustness against noise in reaction equations and eliminates the need for explicit rule formulation.

In the following sections, we will analyze traditional and state-of-the-art fingerprints, explore their applicability to this project, and draw inspiration from them to design our fingerprints.

2.2 Embeddings

2.2.1 Traditional Fingerprints. Traditionally, reaction fingerprints were manually crafted based on the reaction centre or by combining fingerprints of the reactant, reagent, and product[23]. ChemAxon[1], as an example, offers eight distinct types of reaction fingerprints[23]. Among the most commonly used manual fingerprints is the difference fingerprint, introduced by Schneider et al. [21]. This particular fingerprint requires knowledge of the reactant-reagent division since it assigns different weights to reactants and reagents[23]. Following the concept of differentiable molecule fingerprints by Duvenaud et

al.[3], Wei et al.[26] introduced the first learned reaction fingerprint to predict chemical reactions. Unfortunately, its design limited this fingerprint to accommodate only reactions involving two reactants and one reagent, thus limiting its applicability to reactions that fit this specific configuration[23].

2.2.2 DRFP. Recently, these limitations have been overcome by Differential Reaction Fingerprint (DRFP) designed by Daniel et al.[19]. The DRFP method draws on principles from chemical fingerprints such as Extended Connectivity Fingerprints (ECFP)[20] and MinHash Fingerprints (MHFP)[18][19], creating ring substructures from molecules and then hashing their SMILES representations. This hashing and folding process converts different SMILES representations into binary vectors of a predetermined dimension, independent of the input's complexity and requiring minimal memory space, thus facilitating the processing by most machine learning algorithms[19]. Figure 1 illustrates converting the Favorskii rearrangement reaction into its corresponding DRFP.

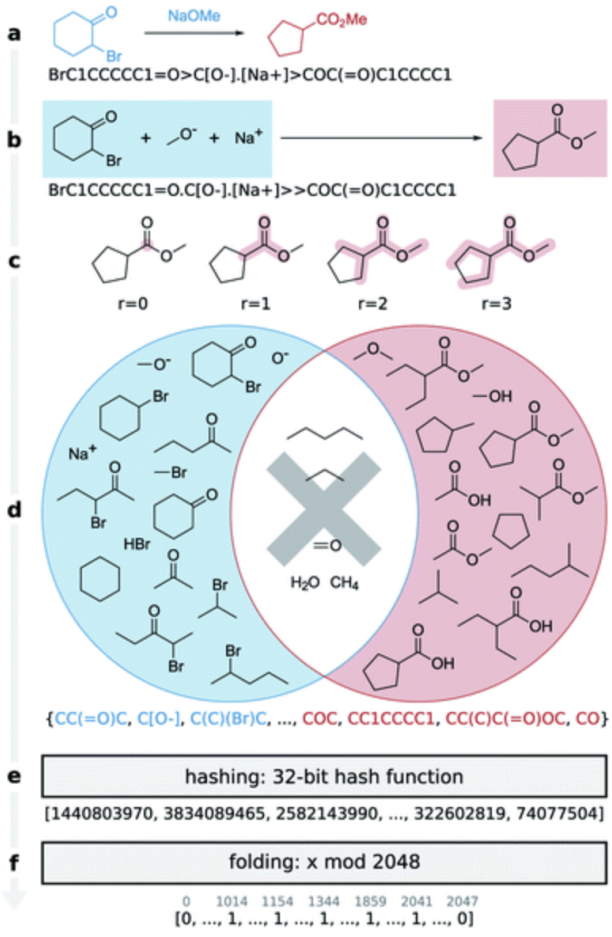

Figure 1: Encoding a Favorskii rearrangement as a DRFP fingerprint[19]

In tests conducted by Daniel et al., classifiers trained using DRFP outperformed those taught with traditional, unlearned fingerprints, surpassing the performance of learned fingerprints without the need for supervised pre-classification learning[19].

2.2.3 RXNFP. These limitations have also been overcome by applying natural language processing-inspired transformer architectures for learning vector embeddings of reactions[19]. The BERT-based transformer model, designed by Schwaller et al.[22], when used as a reaction fingerprint, not only eliminates the need for partitioning reactants and reagents but also accommodates an arbitrary number of molecules on either side of the chemical equation. During the pre-training of Schwaller et al.'s BERT model, individual tokens in the reaction SMILES are masked and then predicted by the model[22]. Since the preceding [CLS] token is never masked, the model can always use its representation to recover the masked tokens. The model embeds a global reaction description using the [CLS] token. Before fine-tuning, the [CLS] token embedding is learned entirely through self-supervised learning[22]. In the subsequent supervised classification tasks, embedding the [CLS] token is used as input to a single-layer classification head. In their example, the [CLS] token embedding is a vector of size 256, corresponding to the hidden size of the BERT model. The model must focus on the reaction centre and certain precursors specific to individual-named reactions.

Their dataset labels came from the highly imbalanced Pistachio dataset[16], which uses NameRXN for reaction classification. The test set contains 132k reactions from 792 different categories in Pistachio. Their best model achieved a classification accuracy of 98.2%[22] on this test set. Additionally, when they used this RXNFP as inputs for the 5-NN classifier, the accuracy reached 98.9%[22]. This proved that learned representations can be used as reaction fingerprints, capturing fine-grained differences between reaction classes more effectively than traditional reaction fingerprints[22].

2.3 Contrastive Representation Learning based Fingerprint

Schwaller et al. demonstrated that their pre-trained BERT model on the Pistachio dataset can already capture reaction differences. When using only the pre-trained BERT as inputs for the 5-NN classifier, the accuracy reached 81.9% on the Pistachio test set[22]. This gave us the idea that if we further train the pre-trained model to embed similar reactions closer together and different reactions further apart, the model's performance would improve when using the nearest neighbour classifier. Fine-tuning the model using contrastive representation learning is the perfect realization of this idea.

2.3.1 Contrastive representation learning. The goal of contrastive representation learning is to learn an embedding space where similar samples are close to each other, while dissimilar samples are far apart[13]. This approach involves training the model with pairs of samples, each consisting of similar or dissimilar items. By optimizing a contrastive loss function, the model learns to minimize the distance between embeddings of similar pairs and maximize the distance between embeddings of dissimilar pairs[10]. This training method ensures that the learned representations are more discriminative and effective for classification, clustering, and retrieval tasks. Contrastive representation learning has been successfully applied in various domains, including computer vision[8], natural language processing[30], and, as we propose, chemical reaction fingerprinting[28].

2.3.2 Loss function. In contrastive representation learning, the loss function is crucial in optimizing the embedding space. The most commonly used is contrastive loss[25], which ensures that the distance between similar samples is minimized and the distance between dissimilar samples is maximized.

For a given pair of samples (x_i, x_j), the result of the similarity function, y_{ij} , is 1 if the samples are similar and 0 if they are dissimilar. The contrastive loss function $\mathcal{L}$ can be defined as:

$$\mathcal{L}(x_i, x_j, y_{ij}) = y_{ij} \cdot \frac{1}{2}\|f(x_i) - f(x_j)\|^2 + (1 - y_{ij}) \cdot \frac{1}{2}\max(0, m - \|f(x_i) - f(x_j)\|)^2 \quad (1)$$

Here, $f(x)$ represents the embedding function that maps input samples into the embedding space, $|\cdot|$ denotes the Euclidean distance, and m is a margin parameter that defines the minimum distance between dissimilar samples.

The loss function consists of two parts:

(1) The first part, $y_{ij} \cdot \frac{1}{2}|f(x_i) - f(x_j)|^2$, minimizes the distance between similar samples.
(2) The second part, $(1 - y_{ij}) \cdot \frac{1}{2}\max(0, m - |f(x_i) - f(x_j)|)^2$, penalizes dissimilar samples that are closer than the margin m.

By minimizing this loss function, the model learns an embedding space where similar reactions are embedded closer together, and different reactions are pushed further apart, improving the overall classification performance.

2.3.3 Contrastive Learning in Reaction Classification. A study by Wen et al.[28] demonstrated the application of contrastive self-supervised learning for improving machine learning performance on small chemical reaction datasets. Using unsupervised contrastive learning, they proposed a strategy that first trains a graph neural network (GNN) model on unlabelled reaction data. Then, they fine-tuned it on a small number of labelled reactions. This method involves creating augmented versions of the input reactions and maximizing the agreement between their representations while distinguishing them from other reactions.

The pre-trained model showed significant improvements over traditional methods. For instance, when using only eight labelled reactions per class in a training set, the pre-trained model achieved an F1 score of 0.86, compared to 0.64 and 0.63 for supervised and traditional fingerprint-based models, respectively. This highlights the effectiveness of contrastive pre-training in leveraging unlabelled data to enhance reaction classification tasks.

Their study proved contrastive representation learning with a well-defined loss function can effectively enhance the model's ability to distinguish between different reaction classes, making it a powerful technique for generating robust and discriminative chemical reaction fingerprints. This approach leverages unlabelled data to improve model performance on small labelled datasets, providing a significant advantage in chemical reaction classification tasks[28].

2.3.4 Summary. Having reviewed the work of preceding researchers, we propose a novel method for this project. Our approach begins with leveraging BERT for pre-training on unlabeled reaction data from the Reaxys database. Following this pre-training phase, we utilize known reaction labels to distinguish between similar and dissimilar samples. Specifically, our method involves fine-tuning the BERT model through contrastive learning, enhancing its ability to discern nuanced similarities and differences between chemical reactions.

In contrast to Wen et al. [28], who applied regular classification techniques, we focus on metric-based prediction. This approach aims to minimize the distance between embeddings of similar reactions while maximizing the distance between those of different classes. This process is expected to yield more refined spatial embeddings, ultimately contributing to more accurate and insightful analyses of chemical reactions.

2.4 Evaluation metrics

In classification tasks, the performance of a model is typically evaluated using various performance metrics, such as accuracy, recall, precision, and F-measures. These metrics can be averaged using Macro-averaging (MaA) and Micro-averaging (MiA). MaA evaluates the model for individual classes and then averages the scores across all classes. An advantage of MaA is that it treats all classes equally, giving a balanced view of performance, especially useful when dealing with imbalanced datasets[6]. In contrast, MiA assesses the model's performance by aggregating the contributions of all classes before calculating the overall metric. The advantage of MiA is that it considers the overall effectiveness of the model across all instances, making it more sensitive to the performance of classes with a more significant number of samples[17]. Below, we discuss the metrics utilized in this research.

2.5 MaA

MaA calculates metrics for each label individually and then takes the mean. The metrics in question can be accuracy, precision, recall, or an F-measure, all calculated with combinations of true positives (TP_i), false positives (FP_i), true negatives (TN_i), and false negatives (FN_i). Setting B to be one of these metrics, we can present MaA with the following formula:

$$\text{MaA} = \frac{1}{L}\sum_{i=1}^{L} B(TP_i, FP_i, TN_i, FN_i), \tag{2}$$

where L is the total set of different labels.

2.6 MiA

MiA first sums the TP, FP, TN, and FN for each label and then applies a function B. This is captured in:

$$\text{MiA} = B\left(\sum_{i=1}^{L} TP_i, \sum_{i=1}^{L} FP_i, \sum_{i=1}^{L} TN_i, \sum_{i=1}^{L} FN_i\right), \tag{3}$$

3 METHODOLOGY

In this section, the methodology will be discussed. We first discuss the data preparation, after which we will review the implementation and experiments.

3.1 Dataset preparation

3.1.1 Start point. The Reaxys database comprises over 14 million reactions[4]. A total of 4.04 million reactions were retrieved in RXN format from Reaxys using the Reaxys API based on specific transformation queries corresponding to reaction classes. These reactions included embedded MOL v3000 files and atom-atom mapping.

3.1.2 Data preprocessing. Of the 4.04 million reactions, 3.07 million (76%) were retrieved with only one class (the remaining data was retrieved with more than one class assigned). We agreed to use only these reactions to simplify the initial problem, discarding the remaining 1 million reactions. The atom-to-atom mapping was transferred into the SMILES notation during the conversion, increasing its complexity. Therefore, we removed this mapping from the MOL files using Regex scripting. In the standard RXN-to-SMILES conversion, multiple reactants or products are stored within one SMILES field, separated by a period ("."). This separation creates ambiguity when salts are involved because the ions are separated by periods.

We converted the MOL files to SMILES using the "Unique SMILES" option, which canonicalises the SMILES, reducing the variety of potential notations for the same substance. The available RXN data source may include multiple instances of the same reactant or product within a reaction due to the atom-to-atom mapping feature. This redundancy is undesirable for the generated SMILES data, so we de-duplicated the reactant and product SMILES per reaction. We tracked the conversion process to ensure all reactants and products were successfully converted to SMILES. If not all substances were converted successfully, the reaction was discarded. Consequently, 10,122 out of 3.07 million reactions (0.33%) were affected, resulting in 3.06 million reactions under 1674 different class labels.

We visualised the distribution of reactants and products for each reaction as shown in Figure 2 and found that 51.5% of reactions had 2 reactants and 1 product, the most common configuration in this dataset. Additionally, 38.3% of reactions had 1 reactant and 1 product, the second most common configuration. Only 2.2% of reactions had more than 3 reactants or products. We also analysed the population of class labels and found that many classes had a low population of reactions, which may be insufficient for training.

Based on this analysis, we removed specific reactions to establish specialised reaction test sets. These included reactions where the number of reactants or products exceeded two (totalling 104,467 reactions)

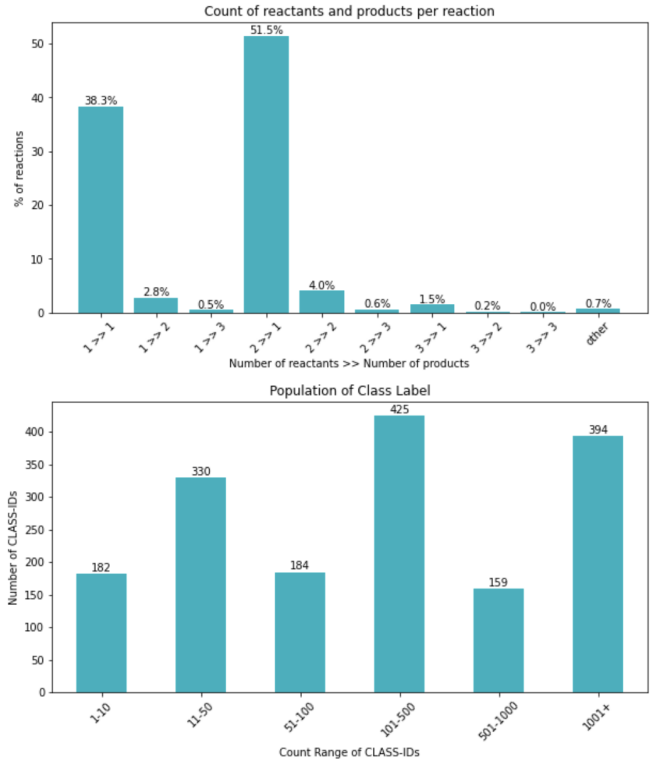

Figure 2: Reaxys data analysis

and reactions categorised under rare classes (fewer than ten reactions per class).

The remaining reactions are represented in the following form: R1.R2 » P1.P2, where R represents reactants, and P represents products. If R2 or P2 is absent, the period connectors between reactants or products are not required. The reaction fingerprints used in this project do not need the partitioning of reactants and reagents and can accommodate any number of molecules on both sides of the chemical equation.

3.1.3 Data Split. Finally, we extracted 2.95 million reactions assigned under a single class. The SMILES represents chemical reactions, and the dataset is categorised under 1,299 distinct class labels and 34 superclass labels. This dataset demonstrates a substantial imbalance. These reactions were divided into a training set, a validation set, and a test set, with proportions of 80%, 10%, and 10%, respectively. Reactions with identical products were kept within the same set to avoid potential overfitting.

3.1.4 Public Dataset. To evaluate the broad applicability of our proposed method, we identified suitable public datasets for further testing. One such dataset is the USPTO 1k TPL, which consists of 445k reactions divided into 1000 template labels. These labels were derived through a multi-step process: first, the USPTO dataset was atom-mapped using RXNMapper. Then, the template extraction workflow by Thakkar et al. was applied to identify the reaction templates[7]. Finally, reactions corresponding to the 1,000 most frequent template hashes were selected and designated as class labels. The USPTO 1k TPL dataset is characterised by a significant imbalance in class distribution.

3.2 Model Implementation

The model pipeline is visualised in Figure 3 and features the process from input to model output, including the baselines and our proposed method.

3.2.1 Baselines. To assess the effectiveness of our method, we compare its performance against three baselines. The first baseline is the DRFP introduced by Daniel et al. [19]. The second baseline, developed by Schwaller et al. [23], is a BERT-based transformer model known as RXNfP. The third baseline is a BERT model pre-trained on the Reaxys database.

DRFP is considered an unsupervised embedding method because it creates fingerprints based on the symmetric difference of circular molecular n-grams from reaction SMILES without relying on labelled data or supervised learning algorithms.

The pre-trained Bert model is also an unsupervised embedding method that learns to represent chemical reactions through masked language modelling. This involves predicting masked tokens in reaction SMILES strings without requiring labelled data for the specific task of reaction classification.

For RXNFP, which employs a supervised approach, we acknowledge that there may be some discrepancies between the reaction classification methods used in the Reaxys database and those employed during the supervised fine-tuning with the Pistachio dataset.

3.2.2 Contrastive Representation Learning based Fingerprint. We propose an approach based on contrastive learning, utilising labelled pairs of similar and dissimilar reactions. This method trains a BERT model, pre-trained on the Reaxys database, to learn the similarities between reactions. Consequently, it generates more meaningful spatial embeddings. The fine-tuning process can be divided into three main steps: systematically generating pairs of chemical reactions that are either similar or dissimilar based on existing label information, designing a loss function, and optimising the pre-trained model through the loss function and learning rate adjustments.

Data pairs setting The data pairs setting is divided into positive pairs (similar reactions) and negative pairs (dissimilar reactions).

Positive pairs are formed by grouping reactions within the same distinct class, implying higher similarity. Each reaction is paired with the following sequential reaction within its group, cycling back to the first reaction at the end of the group to ensure all reactions are paired.

For negative pairs, we designed two different methods: hard negative pairs and soft negative pairs.

In hard negative pairs, reactions across different superclasses are considered, ensuring the reactions belong to different superclasses and thus have a marked difference in their transformative characteristics. An exception is made for reactions classified under the 'AVNAMEDR' superclass, where the negative pairs are selected from within the same superclass but not from the same class. This is because the 'AVNAMEDR' superclass corresponds to named reactions—reactions that have historical names. Since named reactions encompass all possible reactions with historical names and are not assigned to this mechanistic hierarchy, there is some novelty in each named reaction. Therefore, we consider them all different. This pairing method ensures that the model can learn the differences between completely different reactions, ultimately improving classification accuracy.

Soft negative pairs involve pairing reactions within the same superclass but belonging to distinct classes. This method aims to capture the subtle differences between similar reactions that do not belong to the same category, thereby enhancing classification accuracy.

Loss function We employed the commonly used contrastive loss in contrastive learning for the loss function. The relevant formula and definitions are in 2. The critical parameter in this function is the margin, which ensures that the embeddings of different classes are sufficiently separated in the embedding space. By penalising negative pairs closer than the margin, the model learns to push different classes apart, thereby enhancing class separability. An appropriate margin allows the model to account for intra-class variability. If the margin is too small, slight variations within the same class could be incorrectly penapenalised. The margin might lead to positive pairs being treated as negatives, which can confuse the model[14].

We visualised the distance distributions of positive and negative pairs by selecting 100,000 pairs of each type for observation, as shown in Figure 4. Based on this distribution, we determined the margin to be 2.5. This value lies near the tail end of the distribution for positive pairs,

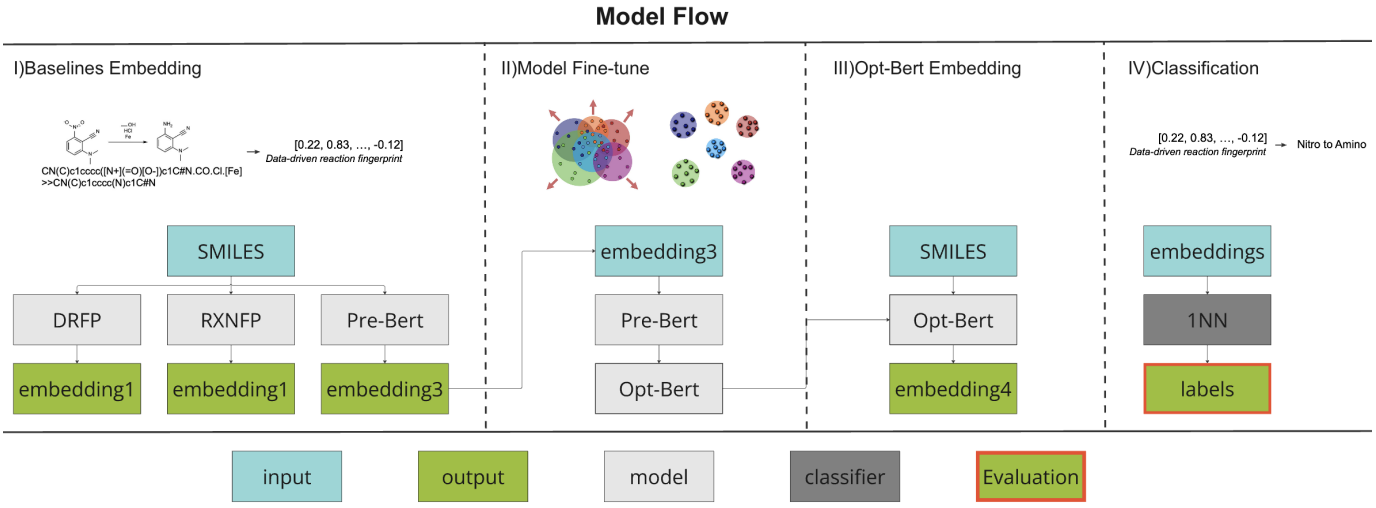

Figure 3: Model Flow

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)	Precision (Micro)	Recall (Micro)	F1-Score (Micro)
CFP (full data)	**0.9618**	0.8889	0.8761	0.8759	**0.9619**	**0.9619**	**0.9616**
CFP (50% data)	**0.9618**	0.8882	0.8757	0.8757	0.9618	0.9618	0.9616
CFP (30% data)	0.9616	**0.8957**	**0.8800**	**0.8813**	0.9614	0.9614	0.9612
RXNFP	0.9579	0.8747	0.8642	0.8625	0.9581	0.9580	0.9578
CFP (10% data)	0.9563	0.9020	0.8747	0.8812	0.9564	0.9564	0.9561
CFP (10% data, soft negative pairs)	0.9446	0.8913	0.8595	0.8679	0.9447	0.9447	0.9444
DRFP	0.9269	0.8903	0.8603	0.8675	0.9279	0.9270	0.9266
Pre-trained Bert	0.8328	0.7646	0.7221	0.7330	0.8325	0.8328	0.8319

Table 1: Performance of Different Models

indicating that most positive pairs will have distances less than this margin, thereby incurring minimal to no penalty. It is also around the peak for negative pairs, ensuring that a significant portion of negative pairs have distances greater than the margin, thus correctly incurring little to no penalty. However, there is a clear section of negative pairs below this margin, which would correctly contribute to the learning signal by penalising the model for not sufficiently separating these pairs.

Figure 4: Distance distribution plot (Pre-trained model)

By setting the margin at 2.5, we aim to achieve a balance that maximises the model's ability to distinguish between similar and dissimilar reactions, thereby improving the overall quality of the embeddings.

Optimization and Training To optimise the pre-trained BERT model, we employed the Adam optimiser and implemented gradient accumulation to handle larger batch sizes. This approach efficiently uses computational resources while ensuring stable training dynamics. Additionally, mixed precision training was utilised using torch.cuda.amp. This technique enhances computational efficiency by performing operations with lower precision when possible, thus speeding up training without sacrificing model performance.

During training, the model undergoes five epochs, each comprising several iterations over the training dataset. In each iteration, pairs of reactions are processed, and their embeddings are generated using the BERT model. The contrastive loss is then computed based on the embeddings and their respective labels (similar or dissimilar). The loss function penalises the model if similar pairs are not close enough or dissimilar pairs are too close within the embedding space.

Gradient accumulation is performed to manage memory usage effectively. Instead of updating the model parameters after each batch, gradients are accumulated over multiple batches and the model is updated less frequently. This approach simulates a larger batch size, which is beneficial for training stability and convergence.

After each epoch, the model's performance is evaluated on a validation set to monitor its generalisation ability. The validation loss is calculated similarly to the training loss, indicating how well the model is expected to perform on unseen data. A consistent decrease in training and validation loss indicates successful learning and convergence of the model.

Through this structured optimisation and training process, we ensure that the fine-tuned BERT model can effectively learn to differentiate between similar and dissimilar reaction pairs, thereby improving its overall classification performance.

3.2.3 Classification. Inspired by the literature from Schwaller et al.[22], we used the k-nearest neighbour classifier based on the FAISS framework developed by the Facebook research team[9] for reaction classification. This is because FAISS provides an efficient implementation of brute-force k-nearest neighbour search that can be applied to relatively large datasets, thereby avoiding the biases that approximate methods may introduce.

The number of nearest neighbours k = 1 and the Euclidean metric (L2)[12] are chosen for all tests. This choice of parameters was determined after our validation tests, during which we set the k-values to 3, 5, and 7. The predicted class of the query was assumed to be the most frequently represented. However, due to significant variations in the dataset's number of responses per class, responses in the rare classes were easily misclassified into more common courses. Consequently, we opted to assign the predicted class based on the class where the nearest neighbours are located for the test responses.

3.3 Experiments

This subsection will introduce our experimental procedure, including computational load, evaluation and implementation.

3.4 Computational Load

Most of our experiments were conducted on a Tesla V100-SXM2-16GB GPU. This GPU operates at a maximum power usage of 300W and is equipped with 16GB of memory, providing the computational capacity required for handling large-scale data efficiently.

In the pair setting process, we experimented with 10%, 30%, 50%, and the entire dataset to assess the impact of different training data volumes on the fine-tuned model. The pairing process took approximately 1 minute 52 seconds, 20 minutes 23 seconds, 55 minutes 30 seconds, and 3 hours 54 minutes 21 seconds, respectively. For the model fine-tuning stage, we set the training to 5 epochs. The time required for each epoch was approximately 23 minutes, 1 hour 14 minutes, 1 hour 53 minutes, and 3 hours 50 minutes, respectively, corresponding to the different dataset sizes mentioned earlier.

In the embedding stage, we employed a mean pooling strategy that considers the attention mask to enhance the quality of the embeddings produced by the BERT model. This ensures that the averaging process accounts for only the actual tokens, excluding padding tokens. For the training set embeddings, we divided the training set into ten subsets, each containing approximately 200,000 reactions, and performed embeddings for each subgroup separately. The combined embedding process took around 30 minutes. The embedding processes for the validation and test sets took approximately 4 minutes each. The classification step required nearly 2 hours. In the baseline experiments, we set the fingerprint length to 256 to ensure the performance of different embeddings is not due to their size. The time required for embedding and classification was similar to that for the fine-tuned model.

3.4.1 Evaluation. To evaluate the performance of the different models, we employ seven metrics commonly used in classification problems. These metrics include overall classification accuracy and both the micro and macro versions of the F1 score, recall, and precision. The main focus is increasing the F1 score as it provides a balanced view of the algorithm's performance. Given the imbalanced nature of the problem, the recall and precision metrics also offer valuable insights. We also explore the performance per class to identify specific strengths and weaknesses.

3.4.2 Implementation. The models were implemented using Python 3.8.8 and PyTorch 2.3.0. For more information on the packages used and their implementation, please refer to the GitHub repository associated with this project.

4 RESULTS

In this section, we will evaluate the model performance and behaviour.

4.1 Model Performance

Table 5 presents the evaluation metrics for the various models tested on our chemical reaction classification task. The baseline models are included for comparison. Our primary observation is that the CFP consistently outperforms these baselines across most metrics.

The performance of the CFP with varying sizes of the pair-dataset used for fine-tuning (50%, 30%, and 10%) indicates that the model maintains high accuracy and F1 scores even with reduced data. The CFP fine-tuned on just 10% of the pair-dataset achieves a macro F1-Score of 0.8812, demonstrating the effectiveness of our contrastive learning approach in leveraging smaller datasets without significant performance loss. These findings align with those of Wen et al.[28], who used contrastive self-supervised learning to enhance machine learning performance on small chemical reaction datasets. This result is crucial for practical applications where large labelled datasets are unavailable.

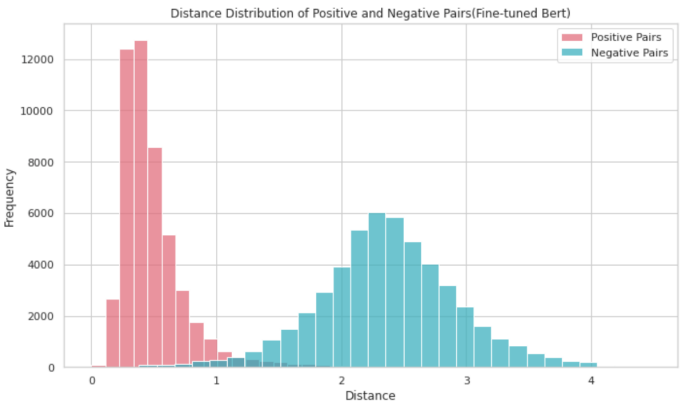

Figure 5: Distance distribution plot (CFP)

For the selection of negative pairs, we compared the results obtained using two different methods with the same 10% of the database. The hard pairing method consistently outperformed the soft pairing method across all evaluation metrics. Therefore, we determined that the hard pairing method is more suitable for identifying dissimilar reactions in this project.

Interestingly, the CFP fine-tuned on 30% of the pair-dataset delivers the highest macro F1-Score of 0.8813. This suggests that providing the model with more pairs might decrease prediction accuracy for classes with fewer reactions. This observation highlights the need for more appropriate methods for setting positive and negative reaction pairs, which will be discussed in detail in the discussion section.

Furthermore, we visualized the distance distributions of positive and negative pairs again, selecting the same 100,000 pairs of each type to observe the changes introduced by the CFP as shown in Figure 5.

Compared to Figure 4, we can observe that the distance between positive and negative reaction pairs has become significantly more distinct. Most positive pairs have distances within 1, while the negative pairs have distances between 1 and 4. This demonstrates that the fine-tuning process has been effective, enabling the CFP to provide superior embedding for the classifier. Consequently, this allows the classifier to perform the classification tasks more accurately.

4.2 Model Behavior Analysis

4.2.1 Superclass Level. We first analyze the model's performance at the superclass level, where superclass labels are obtained by mapping the predicted class IDs to their corresponding superclass categories.

A confusion matrix across the superclasses in the dataset demonstrates that the high classification accuracy achieved using the CFP as an embedding method is consistent across most superclasses, as shown in Figure 6. The accuracy significantly surpasses that obtained using the pre-trained BERT as the embedding model.

In the confusion matrix, the rows represent the actual superclasses, with the numbers in parentheses indicating the correct classification accuracy for each superclass. The columns denote the predicted superclasses. The colour intensity of each cell reflects the frequency of predictions, with darker shades indicating higher frequencies.

The confusion matrix for the CFP embedding method exhibits a pronounced diagonal pattern, where the actual and predicted superclasses align, as indicated by the darkest blue cells. In contrast, while

the confusion matrix for the pre-trained BERT embedding also shows a diagonal pattern, it contains more light blue cells, indicating a higher frequency of misclassifications.

A notable exception for the CFP embedding method is observed for the 'AG6NITRA' superclass. Approximately 15% of reactions in this superclass are misclassified as belonging to the 'AG1ARYL' superclass. This misclassification likely arises because, within our taxonomy, the two classes may collapse into one due to sharing the same 'transform ID' label. Additionally, the 'AG1ARYL' superclass is significantly larger, encompassing approximately 22,500 reactions, compared to the 'AG6NITRA' superclass, which contains about 3,500 reactions. This substantial size discrepancy contributes to the confusion between these two classes, as the more extensive 'AG1ARYL' class potentially overwhelms the smaller 'AG6NITRA' class, leading to their misidentification.

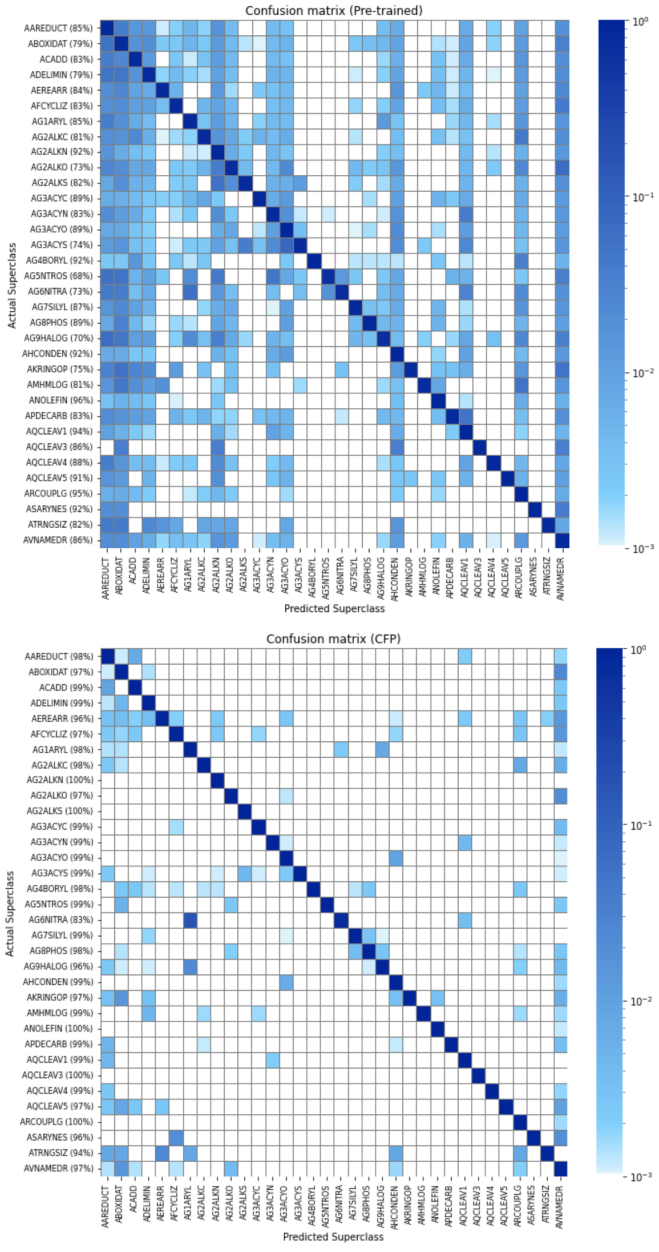

Figure 6: Confusion matrix at superclass level (Pre-trained BERT vs CFP)

4.2.2 Class Level. The analysis in Table 2 highlights the reaction classes with the lowest prediction accuracies when using Pre-trained BERT or CFP as embedding methods. By employing CFP as the embedding method, we observed a notable improvement in the F1-scores of the ten most challenging prediction classes. Except for the 'Swern oxidation of alcohols to ketones' and 'Swern aldehyde synthesis' reaction classes, all other reactions achieved an F1-score exceeding 71%. In contrast, the F1-scores for the ten most poorly performing reactions using Pre-trained BERT embeddings were all below 62%.

Many misclassifications occur between reactions that share similar reactants or transformation steps. For instance, Swern oxidation, Dess-Martin oxidation, and Corey-Suggs reactions describe chemically similar transformations from alcohol to aldehyde or ketone. The primary difference among these reactions lies in the used reagents, which are not in our reaction descriptions and are limited to reactants and products. Consequently, substantial mix-ups between these classes are expected.

Additionally, our CFP model exhibits difficulty accurately distinguishing reversed reactions. This challenge is particularly evident in the misclassification of reduction and oxidation reactions, which are frequently confused with one another. The discussion section will elaborate on potential strategies to enhance the embedding quality for these poorly performing reactions.

4.3 Performance on Public dataset

To demonstrate the broad applicability of our novel embedding method based on reaction embedding space distance generated through contrastive learning, we further fine-tuned the pre-trained BERT model on the USPTO 1k TPL dataset using the same contrastive learning approach. It is important to note that the pre-trained model was initially trained on the Reaxys database, influencing its performance. Our primary goal was to showcase the effectiveness of our distance-based contrastive learning fine-tuning method in improving reaction embeddings for classification tasks, so we didn't pre-train on the MLM task again on the USPTO 1k TPL dataset.

Table 3 presents the classification accuracy of this fine-tuned model on the USPTO 1k TPL dataset, compared to the pre-trained BERT embeddings. By evaluating the 5-nearest neighbour classification benchmark on the TPL dataset, we observed that the fine-tuned model improved classification accuracy using pre-trained BERT embeddings by 15%.

Model	Classifier	Accuracy
Fine-tuned Bert	5-NN	0.799
Pre-trained Bert	5-NN	0.643
RXNFP	5-NN	0.989
DRFP	5-NN	0.919
Traditional FP(AP3 256)	5-NN	0.295

Table 3: Reaction classification accuracy on the USPTO 1k TPL data set

These results align with our expectations. Our reaction descriptions focus solely on reactants and products, excluding solvent and catalyst information. Consequently, the performance of our methods on the USPTO 1k TPL dataset falls short of that achieved by RXNFP and DRFP. However, it still significantly outperforms traditional fingerprints.

5 DISCUSSION

This section will address the limitations identified during our experiments and discuss potential areas for future work to enhance our approach further.

5.1 Limitations and Future Work

Firstly, our method for selecting negative samples may not be optimal. Negative pairs are randomly selected from other superclasses, which might not provide the most challenging comparisons. A recent publication introduced a novel approach for negative sample pairing known as Approximate Nearest Neighbor Contrastive Estimation (ANCE)

Pre-trained model

Class	F1-score	Most frequent incorrectly predicted class
Dess-Martin aldehyde synthesis	0.52	Reduction of aldehydes to alcohols
Swern aldehyde synthesis	0.53	Dess-Martin aldehyde synthesis
Oxidation of dialkyl sulfides to sulfoxides	0.57	Oxidation of dialkyl sulfides to sulfones
Lindgren-Pinnick oxidation of aldehydes to carboxylic acids	0.57	Reduction of aldehydes to alcohols
Corey-Suggs Reagent	0.58	Reduction of aldehydes to alcohols
Oxidation of alkyl aryl sulfides to sulfoxides	0.61	Oxidation of alkyl aryl sulfides to sulfones
Appel halogenation	0.62	Reduction of aldehydes to alcohols
Oxidative fission of alkenes to aldehydes	0.63	Reduction of aldehydes to alcohols
Reduction of aldehydes to alcohols	0.64	Oxidative fission of alkenes to aldehydes

CFP

Class	F1-score	Most frequent incorrectly predicted class
Swern oxidation of alcohols to ketones	0.56	Dess-Martin ketone synthesis
Swern aldehyde synthesis	0.67	Dess-Martin aldehyde synthesis
Corey-Suggs Reagent	0.71	Dess-Martin ketone synthesis
Dess-Martin aldehyde synthesis	0.75	Swern aldehyde synthesis
O-silylation of tertiary alkanols	0.76	O-silylation of secondary alkanols
Condensation of heteroaryl alkyl ketones with hydrazines	0.78	Condensation of phenyl alkyl ketones with hydrazines
Dess-Martin ketone synthesis	0.81	Corey-Suggs Reagent
Oxidation of heteroaryl aryl sulfides to sulfones	0.81	Oxidation of diaryl sulfides to sulfones
Osmylation of alkenes to 1,2-diols	0.82	Permanganate oxidation of alkenes to 1,2-diols
Condensation of heteroaryl aldehydes with hydrazines	0.83	Condensation of aromatic aldehydes with hydrazines

Table 2: Worst-predicted reaction classes

[29]. ANCE improves retrieval performance by dynamically selecting the most challenging negative samples from the entire dataset using Approximate Nearest Neighbor (ANN) techniques, rather than relying on random selection. This method enables the model to distinguish between relevant and irrelevant data better, thereby enhancing retrieval precision.

Secondly, our CFP model generates reaction embeddings without directly performing the classification task. In Schwaller et al.'s RXNFP model [22], a classification layer was added at the end, enabling direct classification. In future experiments, we plan to explore a similar approach to potentially improve our model's performance by incorporating a classification layer directly into the embedding generation process.

Thirdly, there are certain limitations in the choice of hyperparameters for our current experiments. Due to the long training cycles and limited experimental time, we could not thoroughly explore various hyperparameter combinations. Our current settings are based on recommendations from previous studies. Future work will involve more comprehensive experiments to identify the most effective hyperparameters for our model, allowing for a more tailored and potentially more performant configuration.

Additionally, the database used to train the current model does not include reaction conditions, which are crucial for accurately classifying reaction types. Future work must integrate reaction conditions into the dataset to address this limitation.

Another limitation arises from the difficulty in differentiating reactions with minor structural differences far from the reaction centre. For instance, secondary and tertiary alcohols can be confused due to the subtle difference in the number of neighbouring atoms, as shown in Table 2 on the fifth line in CFP worst-predicted reaction classes (e.g., HO-C with two neighbours vs HO-C with three neighbours). Additionally, distinguishing between heteroaryl and aryl substituents presents a challenge as this difference, although chemically significant, is minor and distant from the reaction centre. Aryl is an aromatic ring containing only carbon atoms, whereas heteroaryl includes at least one non-carbon atom (e.g., nitrogen or oxygen). This nuance can be critical for certain reaction classes but may not be a reliable differentiator for others.

The classification task is further complicated by the chemical similarities between different classes, which can introduce noise and make it difficult to define class boundaries clearly. Some features of distant atoms might be crucial for specific reactions (e.g., the distinction between aryl and heteroaryl groups), while in other cases, they may not be as important. This variability underscores the need for a more sophisticated approach to feature selection and classification, which we aim to address in future research.

6 CONCLUSION

In conclusion, this study addresses the research question: To what extent can the CFP improve classification performance on chemistry databases compared to other methods?

Our findings demonstrate that contrastive fine-tuning significantly improves reaction classification accuracy compared to embeddings pre-trained only on the MLM task across multiple datasets. The CFP method improved classification accuracy by 13% on the Reaxys dataset and by 15% on the USPTO 1k TPL dataset compared to pre-trained BERT embeddings. The CFP also outperformed the DRFP and the RXNFP on Reaxys, achieving an accuracy of 96.18%, which surpasses the 95.79% accuracy of the RXNFP model and the 92.69% accuracy of the DRFP model.

However, limitations in our study include the suboptimal selection of negative samples and the absence of a classification layer in the current CFP model. Future work should focus on integrating dynamic negative sample selection techniques such as ANCE and incorporating a classification layer directly into the embedding generation process. Additionally, more comprehensive hyperparameter tuning and including reaction conditions in the training dataset are necessary to enhance the model's performance further.

Our study contributes to chemical reaction classification by introducing a novel, effective, and broadly applicable embedding method. Further research and optimization are required to fully realize the CFP model's potential in various practical applications.

REFERENCES

[1] ChemAxon. 2023. Reaction Fingerprint RF. https://docs.chemaxon.com/display/ltsargon/Reaction+fingerprint+RF.

[2] Lingran Chen. 2003. Reaction classification and knowledge acquisition. *Handbook of Chemoinformatics: From Data to Knowledge in 4 Volumes* (2003), 348–390.

[3] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems* 28 (2015).

[4] Elsevier. 2024. Reaxys. https://www.elsevier.com/products/reaxys. Accessed: 2024-03-03.

[5] Jonathan Goodman. 2009. Computer software review: Reaxys.

[6] Thamme Gowda, Weiqiu You, Constantine Lignos, and Jonathan May. 2021. Macro-average: rare types are important too. *arXiv preprint arXiv:2104.05700* (2021).

[7] Reymond Group. Year. CASP and dataset performance analysis. https://github.com/reymond-group/CASP-and-dataset-performance. Accessed: date-of-access.

[8] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2020. A survey on contrastive self-supervised learning. *Technologies* 9, 1 (2020), 2.

[9] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[10] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems* 33 (2020), 18661–18673.

[11] Hans Kraut, Josef Eiblmaier, Guenter Grethe, Peter Löw, Heinz Matuszczyk, and Heinz Saller. 2013. Algorithm for reaction classification. *Journal of chemical information and modeling* 53, 11 (2013), 2884–2895.

[12] Nathan Krislock and Henry Wolkowicz. 2012. *Euclidean distance matrices and applications*. Springer.

[13] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. 2020. Contrastive representation learning: A framework and review. *Ieee Access* 8 (2020), 193907–193934.

[14] Zhizhe Liu, Zhenfeng Zhu, Shuai Zheng, Yang Liu, Jiayu Zhou, and Yao Zhao. 2022. Margin preserving self-paced contrastive learning towards domain adaptation for medical image segmentation. *IEEE Journal of Biomedical and Health Informatics* 26, 2 (2022), 638–647.

[15] Vipul Mann and Venkat Venkatasubramanian. 2023. AI-driven hypergraph network of organic chemistry: network statistics and applications in reaction classification. *Reaction Chemistry & Engineering* 8, 3 (2023), 619–635.

[16] NextMove Software. 2019. *Pistachio*. http://www.nextmovesoftware.com/pistachio.html

[17] Lucas Pereira and Nuno Nunes. 2017. A comparison of performance metrics for event classification in non-intrusive load monitoring. In *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 159–164.

[18] Daniel Probst and Jean-Louis Reymond. 2018. A probabilistic molecular fingerprint for big data settings. *Journal of cheminformatics* 10 (2018), 1–12.

[19] Daniel Probst, Philippe Schwaller, and Jean-Louis Reymond. 2022. Reaction classification and yield prediction using the differential reaction fingerprint DRFP. *Digital discovery* 1, 2 (2022), 91–97.

[20] David Rogers and Mathew Hahn. 2010. Extended-connectivity fingerprints. *Journal of chemical information and modeling* 50, 5 (2010), 742–754.

[21] Nadine Schneider, Daniel M Lowe, Roger A Sayle, and Gregory A Landrum. 2015. Development of a novel fingerprint for chemical reactions and its application to large-scale reaction classification and similarity. *Journal of chemical information and modeling* 55, 1 (2015), 39–53.

[22] Philippe Schwaller, Daniel Probst, Alain C Vaucher, Vishnu H Nair, David Kreutter, Teodoro Laino, and Jean-Louis Reymond. 2021. Mapping the space of chemical reactions using attention-based neural networks. *Nature machine intelligence* 3, 2 (2021), 144–152.

[23] Philippe Schwaller, Daniel Probst, Alain C Vaucher, Vishnu H Nair, Teodoro Laino, and Jean-Louis Reymond. 2019. Data-driven chemical reaction classification, fingerprinting and clustering using attention-based neural networks. (2019).

[24] Yun-Fei Shi, Zheng-Xin Yang, Sicong Ma, Pei-Lin Kang, Cheng Shang, P Hu, and Zhi-Pan Liu. 2023. Machine learning for chemistry: basics and applications. *Engineering* (2023).

[25] Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2495–2504.

[26] Jennifer N Wei, David Duvenaud, and Alán Aspuru-Guzik. 2016. Neural networks for the prediction of organic chemistry reactions. *ACS central science* 2, 10 (2016), 725–732.

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

[28] Mingjian Wen, Samuel M Blau, Xiaowei Xie, Shyam Dwaraknath, and Kristin A Persson. 2022. Improving machine learning performance on small chemical reaction data with unsupervised contrastive pretraining. *Chemical science* 13, 5 (2022), 1446–1458.

[29] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[30] Rui Zhang, Yangfeng Ji, Yue Zhang, and Rebecca J Passonneau. 2022. Contrastive data and learning for natural language processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorial Abstracts*. 39–47.

Appendix A MODEL PERFORMANCE AT THE SUPERCLASS LEVEL (CFP)

SUPER_TRANSFORM_ID	weighted_precision	weighted_recall	weighted_f1-score	total_test_support	total_train_support	class_count
AAREDUCT	0.980907	0.979424	0.980101	36644	250984	53
ABOXIDAT	0.897971	0.899569	0.898458	24813	175151	105
ACADD	0.971703	0.976918	0.974235	15640	114981	37
ADELMIN	0.981364	0.978690	0.980110	10650	67085	21
AERARR	0.959260	0.960891	0.959308	2486	17393	36
AFCYCLIZ	0.957406	0.945107	0.950955	4026	31313	30
AG17ARYL	0.954370	0.963109	0.958606	7373	55032	20
AG24LKC	0.894514	0.899479	0.895408	3840	26254	80
AG24LKN	0.975265	0.974710	0.974928	35864	270086	16
AG24LKO	0.928641	0.925962	0.926343	7226	54448	39
AG24LKS	0.931497	0.937345	0.934037	1947	15042	21
AG3ACYC	0.933136	0.935629	0.934037	2004	13701	24
AG3ACYN	0.980564	0.980013	0.980223	8956	70353	16
AG3ACYO	0.931288	0.931803	0.930882	13329	101557	75
AG3ACYS	0.865557	0.863265	0.863129	980	7197	24
AG4BORYL	0.944024	0.939925	0.941500	799	5708	10
AG5NTROS	0.927267	0.915119	0.918202	377	3008	13
AG6NITRA	0.878821	0.800687	0.833092	291	2241	9
AG7SILYL	0.899593	0.905109	0.902143	2877	22516	17
AG8PHOS	0.932321	0.925265	0.927990	3024	23495	25
AG9HALOG	0.933599	0.922372	0.927510	3710	26322	28
AHCONDEN	0.936028	0.936213	0.935906	24378	190088	71
AKRINGOP	0.966528	0.951662	0.958742	331	2364	17
AMHMLOG	0.981784	0.983416	0.982556	603	4261	9
ANOLEFIN	0.919591	0.915500	0.915121	7456	52997	10
APDECARB	0.945395	0.939639	0.939997	1607	9347	39
AQCLEAV1	0.988968	0.987943	0.988424	20155	147994	19
AQCLEAV3	0.965685	1.000000	0.982493	28	188	2
AQCLEAV4	0.963177	0.961993	0.962402	3473	25025	6
AQCLEAV5	0.961659	0.920290	0.939516	276	1036	4
ARCOULP	0.993291	0.994258	0.993763	49108	349505	34
ASARYNES	0.966434	0.981818	0.973818	55	399	4
ATRNGSIZ	0.906945	0.888080	0.893356	125	995	14
AVNAMEDR	0.957149	0.957507	0.956888	36359	267140	383

Table 4: Model performance metrics (CFP)

Appendix B MODEL PERFORMANCE AT THE SUPERCLASS LEVEL (PRE-TRAINED BERT)

SUPER_TRANSFORM_ID	weighted_precision	weighted_recall	weighted_f1-score	total_test_support	total_train_support	class_count
AAREDUCT	0.803098	0.809328	0.805795	36644	250984	53
ABOXIDAT	0.677883	0.668238	0.671989	24813	175151	105
ACADD	0.814000	0.804476	0.808679	15640	114981	37
ADELMIN	0.784368	0.776996	0.780841	10650	67085	21
AERARR	0.877156	0.851167	0.862157	2486	17393	36
AFCYCLIZ	0.845355	0.796324	0.818352	4026	31313	30
AG17ARYL	0.829620	0.790798	0.811905	7373	55032	20
AG24LKC	0.802303	0.769010	0.785250	3840	26254	58
AG24LKN	0.869125	0.860660	0.874731	35864	270086	16
AG24LKO	0.698992	0.684750	0.691074	7226	54448	39
AG24LKS	0.784333	0.751296	0.766184	1947	15042	21
AG3ACYC	0.870933	0.874750	0.871344	2004	13701	24
AG3ACYN	0.800558	0.777579	0.788355	8956	70353	16
AG3ACYO	0.798432	0.822117	0.809102	13329	101557	75
AG3ACYS	0.735298	0.681633	0.707236	980	7197	24
AG4BORYL	0.896033	0.901126	0.896521	799	5708	10
AG5NTROS	0.700919	0.636605	0.658001	377	3008	13
AG6NITRA	0.780343	0.697595	0.731568	291	2241	9
AG7SILYL	0.794845	0.782760	0.787155	2877	22516	17
AG8PHOS	0.817907	0.797950	0.806949	3024	23495	25
AG9HALOG	0.747262	0.654987	0.679571	3710	26322	28
AHCONDEN	0.857031	0.864099	0.860171	24378	190088	71
AKRINGOP	0.840561	0.770393	0.799440	331	2364	17
AMHMLOG	0.827958	0.802653	0.812965	603	4261	9
ANOLEFIN	0.886013	0.912402	0.898703	7456	52997	10
APDECARB	0.853496	0.823895	0.836453	1607	9347	39
AQCLEAV1	0.878669	0.915654	0.896446	20155	147994	19
AQCLEAV3	0.896978	0.928571	0.912465	28	188	2
AQCLEAV4	0.874072	0.856032	0.864719	3473	25025	6
AQCLEAV5	0.912033	0.878612	0.893389	276	1036	4
ARCOULP	0.915009	0.925348	0.919963	49108	349505	34
ASARYNES	0.841481	0.872727	0.857100	55	399	4
ATRNGSIZ	0.839444	0.800000	0.813181	125	995	14
AVNAMEDR	0.845969	0.837179	0.840195	36359	267140	383

Table 5: Model performance metrics (Pre-trained BERT)