2024 International Conference on Machine Learning and Applications (ICMLA) | 979-8-3503-7488-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICMLA61862.2024.00043

Oracle Embeddings for Chemical Detection

1st Cate Dunham
Data Science
WPI
Worcester MA, USA
cmdunham@wpi.edu

2nd Maria Barger
Data Science
WPI
Worcester MA, USA
mtbarger@wpi.edu

3rd Randy Paffenroth
Mathematical Sciences
Computer Science,
and Data Science
WPI
Worcester MA, USA
rcpaffenroth@wpi.edu

4th Joshua Uzarski
Research Chemist
US Army
DEVCOM Soldier Center
Soldier Protection Division
Natick, MA, US
joshua.r.uzarski.civ@army.mil

5th Chia-Wei Tsai
Interdisciplinary
Physical Scientist
CIV DTRA RD
Fort Belvoir, VA, USA
chiawei.tsai.civ@mail.mil

***Abstract*—The accurate detection of chemical agents promotes many national security and public safety goals, and robust chemical detection methods can prevent disasters and support effective response to incidents. Mass spectrometry is an important tool in detecting and identifying chemical agents. However, there are high costs and logistical challenges associated with acquiring sufficient lab-generated mass spectrometry data for training machine learning algorithms, including skilled personnel, sample preparation and analysis required for data generation. These high costs of mass spectrometry data collection hinder the development of machine learning and deep learning models to detect and identify chemical agents.**

Accordingly, the primary objective of our research is to create a mass spectrometry data generation model whose output (synthetic mass spectrometry data) would enhance the performance of downstream machine learning chemical classification models. Such a synthetic data generation model would reduce the need to generate costly real-world data, and provide additional training data to use in combination with lab-generated mass spectrometry data when training classifiers.

Our approach is a novel combination of autoencoder-based synthetic data generation combined with a fixed, apriori defined hidden layer geometry. In particular, we train pairs of encoders and decoders with an additional loss term that enforces that the hidden layer passed from the encoder to the decoder match the embedding provided by an external deep learning model designed to predict functional properties of chemicals.

We have verified that incorporating our synthetic spectra into a lab-generated dataset enhances the performance of classification algorithms compared to using only the real data. Our synthetic spectra have been successfully matched to lab-generated spectra for their respective chemicals using library matching software, further demonstrating the validity of our work.

***Index Terms*—data generation, chemical identification, mass spectrometry, Chemception**

I. Introduction

The ability to detect chemicals in real time is important in many real-world situations. While many machine learning, and deep learning, methods have been created in attempts to solve the problem of real-time chemical identification, those algorithms often require a large corpus of training data to be effective. However, an in many real-world problems in the physical sciences, such training data can be difficult and costly to obtain. Accordingly, herein we study the use of synthetic data in the training of chemical detection algorithms.

Our research was prompted by the challenging problem of utilizing mass spectrometry data and classification models for the detection of new and potentially hazardous chemicals [10], [12]. Mass spectrometry data is used to determine the structure of a molecule from its ionization fragments stemming from its interaction with ionizing energy. It is provided as a measurement of the mass-to-charge ratio of a molecule, and can be used to identify molecular structures [7]. However, due to the high monetary and temporal costs associated with generating mass spectrometry data, it is impractical to manually generate a dataset large enough to train a classifier and detect those chemicals in the field [2], [3]. As a result, we have turned our attention to synthetic mass spectrometry data.

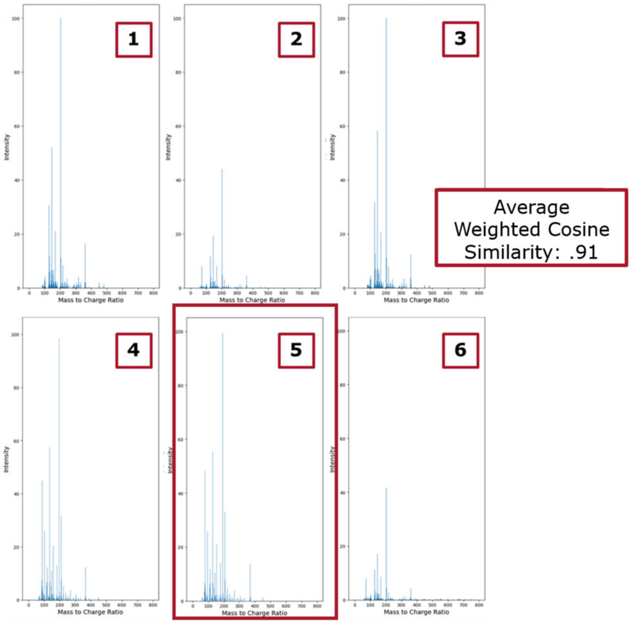

Fig. 1. The outlined spectrum in the figure above represents a synthetic spectrum generated by one of our models. All other spectra in the figure represent lab-generated data for the same chemical. The synthetic spectrum's weighted cosine similarity to the other spectra in the plot is .91 - out of a maximum score of 1 - indicating close resemblance between the spectra.

Studies such as McEachran et al. (2019) [13] and Wei et al. (2019) [25] have proposed various approaches to creating artificial mass spectrometry data. Inspired by these efforts as well as Goh et al.'s Chemception model - detailed in Section III-A - and Moore et al.'s body of work, we aimed to develop a method of generating synthetic mass spectra using chemical representations generated by the Chemception model, which have shown promise in capturing chemical structure information effectively [6], [15]–[18].

To address our objectives we leveraged the strengths of Chemception and explored innovative techniques for synthetic data generation [1], [6], [24]. Our work builds upon the foundations laid by previous researchers in the field and seeks to provide a cost-effective and efficient solution to the identification of chemical analytes. [13], [25].

In particular, we leverage Chemception to provide a *proscribed embedding* $\mathbf{h}$ that supplies our model with information regarding chemical structure and properties, allowing for more accurate data generation than would otherwise be possible for similar models not incorporating Chemception.

1946-0759/24/$31.00 ©2024 IEEE
DOI 10.1109/ICMLA61862.2024.00043

II. Mass Spectrometry and Data Background

A. Mass Spectrometry

Our research considers synthetic data generation specifically for mass spectrometry - an analytical technique widely used to provide insights into a molecule's composition and structure. In mass spectrometry analysis, the molecules within a sample are ionized, causing fragmentation, and the resulting ions are separated based on their mass-to-charge ratios. A data point generated from mass spectrometry is referred to as a "spectrum" and is often represented graphically as a series of peaks, as shown in Figure 2. Herein we will provide details on the physics of a mass spectrometry sensor, and refer the interested reader to [21] for details. However, we merely observe that height and distribution of the peaks give insight into the abundance of the ions in the sample. In particular, the chemical structure shown in Figure 2 has a unique mass spectrum is characteristic of how the various bonds in the chemical structure are broken in the mass spectrometry sensor.

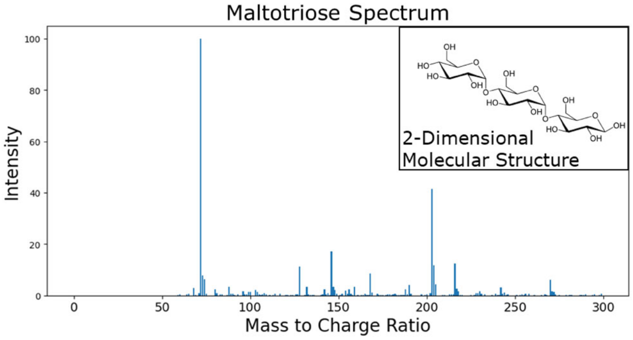

Fig. 2. Example of a mass spectrum with mass-to-charge Ratio along the x-axis and Intensity on the y-axis. The insert in the top right corner is the 2-dimensional molecular structure of the same chemical.

The most intense peak in a spectrum, referred to as the "base peak", represents the most abundant ion fragment detected in the sample. The base peak is assigned a relative intensity of 100%, meaning that intensities for all other peaks in the sample are measured relative to this base peak. Base peak location and the intensity of other peaks relative to the base peak are essential for chemical identification [21].

For our purposes, the mass spectrometry signature is a noisy high-dimensional vector, in our cases residing in $\mathbf{x} \in \mathbb{R}^{797\times 1}$, where 797 was determined by the highest non-zero mass-to-charge ratio observed among the spectra.

B. Data Collection and Preparation

All data used in this research was sourced from the MassBank of North America's (MoNA) database of experimental Gas Chromatography-Mass Spectra [14]. Each molecule in MoNA's database contains mass spectral peaks for the sample, as well as metadata describing the chemical (e.g. molecular weight) and sample preparation (e.g. type of machine used to generate the sample).

We observed discrepancies in chemical naming conventions between samples, likely due to the fact that samples in MoNA's database are collected from many different labs. To ensure consistency and accuracy in our data organization, we opted to organize chemicals based on the unique chemical identifier included in each sample's metadata rather than relying on chemical name.

Many chemicals in the database were represented by only a few samples. 88% of the chemicals in the database had 3 or fewer spectra, while only 2% had more than 5 spectra available. Machine learning prediction models depend on the training data to learn the characteristics of each chemical or class. Insufficient data hinders the model's ability to capture the true variability of each class, impeding its ability learn a meaningful representation of each chemical. Consequently, we chose to include only those chemicals in our dataset that had a minimum number of spectra. To achieve a balance between class representation and dataset size, we included chemicals that had at least 5 spectra. The resulting dataset contained 335 total spectra split between 40 different chemicals as shown in Figure 3.

C. Similarity Metrics

We use weighted cosine similarity in our work to evaluate the quality of synthetic spectra. We calculate weighted cosine similarity as:

$$Sim(X_s, X_t) = \frac{\sum_{k=1} m_k \cdot X_{s,k} \cdot X_{t,k}}{\sqrt{\sum_{k=1} m_k \cdot X_{s,k}^2} \cdot \sqrt{\sum_{k=1} m_k \cdot X_{t,k}^2}} \tag{1}$$

Let X_s denote a synthetic spectrum and X_t represent the true, target spectrum. X_{sk} and X_{tk} correspond to the intensity values for each spectrum at index k. The variables m_k serves as a weight parameter, for which we employed the mass-to-charge ratio (MtCR). Our choice of MtCR as a scaling factor was informed by the fact that values at high MtCR indices are more informative and important for chemical classification than values at low MtCR indices. Using mass-to-charge ratio as a weight parameter in our similarity metric provided insight into the suitability of our synthetic spectra as potential training data points for a classification model. Weighted cosine loss in this paper will refer $1-$ Equation 1.

Weighted cosine similarity, while useful for comparison, does not account for inherent noise in the true data due to varying experimental conditions or differences between samples. To account for inherent noise we applied a ratio of predicted to overall similarity as an additional similarity metric for evaluating synthetic spectra. The predicted to overall similarity ratio compares the difference between target and predicted spectra to the average pairwise difference between all true spectra for that chemical. A ratio close to 1 would indicate that the distance between the predicted and true spectra was the same as the average distance between all true spectra for that chemical.

A widely employed approach to chemical identification using mass spectrometry data involves comparing the query spectrum to reference spectra within a library [25]. The query spectrum is systematically compared to each spectrum in the library using a similarity metric such as weighted cosine similarity, and a prediction is made based on the identity of the spectrum that is most similar to the query [25]. In our research, we applied reference library comparison using the Human Metabolome Database's (HMDB) spectrum match tool, which facilitates comparison between a user-supplied query spectrum and the spectra contained within the HMDB library [9].

III. Methodology

A. Chemception

In this paper we make extensive use of Goh et al.'s Chemception model and Moore et al.'s novel ways of interacting with it [6], [17], [18]. Chemception is a chemistry model trained to predict the toxicity, activity, and solvation properties of its training dataset of molecular structures represented by 2-dimensional images. In a crucial step prior to prediction, Chemception generates 512-dimensional embeddings that

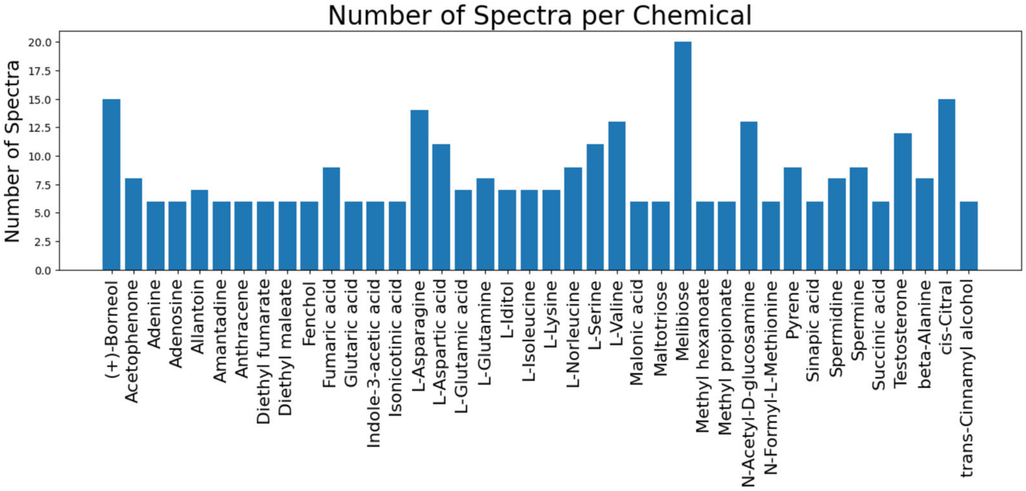

Fig. 3. Mass spectrometry data used in this research were collected from the Mass Bank of North America's database of GC-MS spectra. The dataset contained spectra for 40 chemicals and had a total of 335 spectra.

store the wealth of information the model learned about each data point during training [6]. Moore et al. extended the utility of these embeddings, discovering new methods to extract the stored information and use it for applications beyond the original prediction purpose [17], [18].

As shown in Figure 4, the relative positions of Chemception embeddings provide insights into the similarity or dissimilarity between chemicals. Chemicals with high structural similarity have more similar embeddings than chemicals with low structural similarity.

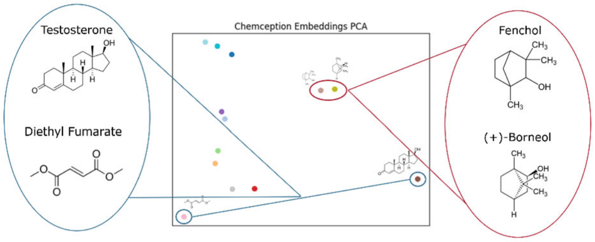

Fig. 4. Chemical embeddings' relative positions reflect similarity: distant for dissimilar chemicals like Testosterone and Diethyl Fumarate, and closer for similar chemicals like Fenchol and (+)-Borneol. These relationships can also be observed by comparing the chemicals' structural diagrams.

In our work, we leverage the 512-dimensional Chemception embeddings as a fixed, universal hidden layer with the aim of facilitating the "translation" of chemical data across formats. We hypothesized that embeddings produced by the Chemception model, which was trained on molecular images that represent chemical structure and composition, would capture a richer representation of chemical compounds compared to embeddings generated by a model trained on our small dataset of a few hundred spectra.

B. Autoencoders

Autoencoders [5] are a key source of inspiration in our work. In particular, in a traditional autoencoder, we have mappings E (often called the encoder) and D (often called the decoder), s.t. given data x, for example a spectra as in Figure 2, we compute

$$\hat{\mathbf{x}} = D(E(\mathbf{x})), \tag{2}$$

the encoder encoder E and decoder D are trained together and share an objective of minimizing $\|\mathbf{x} - D(E(\mathbf{x}))\|$, the difference between the target data and the model's output. Such methods are widely used in many generative data problems. A prime example is the Variational Autoencoder (VAE), which learns a latent space based on the distribution of the data, allowing for generation of synthetic data by sampling from the latent space.

However, in our work we take a somewhat different approach. Namely, in a standard autoencoder the dimension of $\mathbf{h} = E(\mathbf{x})$ has two properties. First, it is low-dimensional with $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{h} \in \mathbb{R}^m$ and $m << n$. Second, the embedding h is learned from the data x.

In our work be break both of these assumptions. First, in our work m and n are much closer in size, with $\mathbf{x} \in \mathbb{R}^{797}$, $\mathbf{h} \in \mathbb{R}^{512}$. In fact, in our methodology, the size of n was determined by the highest non-zero MtCR observed among the spectra, but the size of m is taken to be fixed and proscribed.

Second, and more importantly, for each chemical type, the value of $\mathbf{h}$ is also proscribed; Herein lay the key novelty of our approach. We take $\mathbf{h}$ as given when computing the maps E and D, as opposed to being learned as part of the back-propagation process. For many autoencoder base problems one merely has access to the data $\mathbf{x}$, but in our case we also leverage the Chemception embedding [6] to determine $\mathbf{h}$.

In particular, assuming we have a chemical c, we proceed by determining the 512-dimensional Chemception embedding for c, which we denote as $\mathbf{h}_c \in \mathbb{R}^{512}$. For all spectra pertaining to c, the embedding $\mathbf{h}_c$ remains constant.

C. Loss Function for a Proscribed h

Distinguished from a standard autoencoder with a loss function such as

$$\|\mathbf{x} - D(E(\mathbf{x}))\|,$$

we instead have a two part loss function, namely

$$\|\mathbf{h}_c - E(\mathbf{x})\| + \|\mathbf{x} - D(\hat{\mathbf{h}})\|, \tag{3}$$

where $\hat{\mathbf{h}} = E(\mathbf{x})$ and $\mathbf{h}_c$ is the Chemception embedding for chemical c.

There are several important items to note in (3). First, the loss function is separable in that $\|\mathbf{h}_c - E(\mathbf{x})\|$ can be trained independently of the rest of the loss function, just given the Chemception embedding $\mathbf{h}_c$. While not a focus of this paper, this separability leads to an important idea in that the hidden layer is independent of the particular sensor modality used to create the input signal $\mathbf{x}$. In particular, it is interesting to note that the embedding $\mathbf{h}_c$ was computed without regard for the Mass Spectrometry sensors that we use. It was computed based upon a totally different principle, as we will detail in the next section. However, $\mathbf{h}_c$ is a rich representation of chemical properties, making it quite advantageous for synthetic data generation.

D. Extensions to Other Sensor Modalities

While not a focus of the current text, we observe that the ideas described herein can extended to other additional sensor modalities. Chemception embeddings were computed based upon the 2-dimensional representations of molecular structure mentioned in Figure 2 for the prediction of chemical properties, and without regard to a chemical's representation by a specific sensor. As a result, Chemception embeddings contain details including chemical structure, molecular geometry, similarity or dissimilarity to other chemicals, and toxicity, activity and solvation properties.

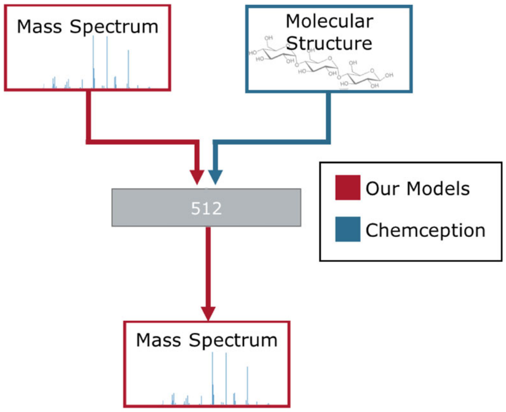

Fig. 5. The modular architecture for translating chemical data between formats using Chemception embeddings as a shared hidden layer. The red and blue branches indicate our implemented models and the Chemception model, respectively.

In our current architecture, as described in Figure 5, each branch encodes to or decodes from Chemception embeddings. We believe Chemception space is rich enough to be used as a latent space for models constructing many different sensor representations. Our modular design allows for the incorporation of a wide range of new data types. Each branch operates independently from the others, enabling the addition of specialized models for translating between any data type and Chemception. In Subsection VI-A we discuss our proposed translation architecture.

IV. Hyper-parameter Tuning

A. Encoder Neural Network Parameters

A chemical's Chemception embedding represents the point in Chemception space where a theoretical noiseless sample for that chemical would encode. Given that our training data correspond to true samples generated in a lab, we expected a certain amount of noise in the spectra. Our encoder's objective was not to map each spectrum directly to its true embedding, but rather to map it *close* to the true embedding, in a position that represents that spectrum's variation from the theoretical noiseless sample.

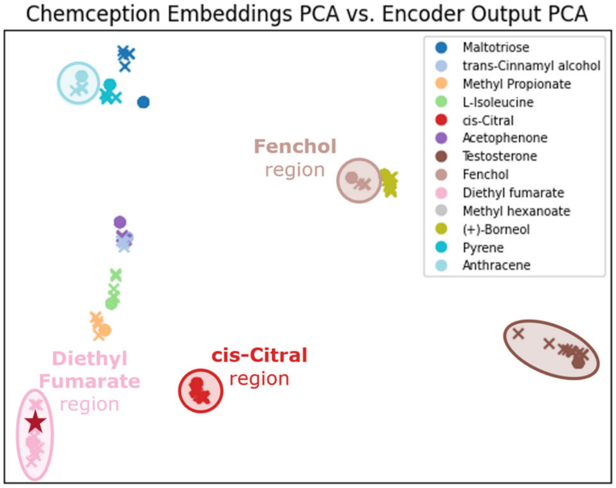

Fig. 6. The encoder accurately mapped spectra, represented by Xs, to their respective Chemception embeddings, represented by *circles* in the chart above. Each chemical's embeddings occupy a different "region" of the Chemception map. The *star* in Diethyl Fumarate's region represents a new embedding generated by the encoder for a noisy input spectrum.

Throughout encoder training we systematically explored various model architecture and hyperparameters to identify an optimal configuration. We chose hyperparameters as displayed in Table I, with special attention given to the number of hidden layers in the model. We found that we were able to increase encoder performance by increasing the number of hidden layers up to seven. Models with more than seven hidden layers began to overfit to the training data.

TABLE I
Encoder Hyperparameters

Size of input layer	797 x 765
Size of output layer	541 x 512
Number of hidden layers	7
Sizes of hidden layers	$765 \times 733, 733 \times 701, 701 \times 669,$ $669 \times 637, 637 \times 605,$ $605 \times 573, 573 \times 541$
Learning rate	1×10^{-5}
Training time	300 epochs
Batch size	32
Activation function	Leaky ReLU
Loss criterion	Mean Square Error

Once the training spectra were encoded, as shown in Figure 6, we could visually observe the "regions" of Chemception space occupied by each chemical's Chemception embedding and encoder-generated embeddings. Initially, we intended to define formal boundaries for each chemical's region and generate synthetic spectra by decoding randomly selected embeddings from within those regions.

If we assume that our training dataset is representative of the broader population, we can infer that embeddings for spectra not seen by our model would fall within their corresponding chemical's region. Furthermore, if we presume that all points within a chemical's region correspond to that chemical, we should be able to decode a randomly selected embedding from within a chemical's region to generate a new, synthetic spectrum for that chemical.

B. Selection of Embeddings for Data Generation

To ensure that our decoder had direct points of comparison for its output, we determined that, rather than selecting embeddings randomly from within chemical regions, the embeddings selected for decoding must correspond in some capacity to the spectra in our training dataset. Decoding the embeddings generated by our encoder based on the training spectra would merely reproduce the training data rather than generating new spectra. Instead, the embeddings needed to represent spectra that were similar, but not identical, to the original, allowing the decoder to generate synthetic data while still being able to accurately calculate loss based on the training spectra.

A straightforward technique for identifying such similar-but-not-identical spectra was to create noisy versions of the training spectra. The noisy spectra shared important features with the original data, while still forcing the decoder to generate new synthetic spectra.

When determining the optimal method for introducing disturbances to our spectra, we were weary of adding noise directly, as the addition of noise would have meant the addition of peaks to the spectra. As mentioned in Subsection II-A, the introduction of additional peaks to a spectrum represents a significant perturbation rather than a minor addition of noise. Instead of *adding* noise, we created noise in the spectra by randomly removing a certain percentage of peaks. Perturbing our spectra by removing peaks allowed the encoder to receive informative signals without introducing additional peaks that could misrepresent the spectra as belonging to different chemicals.

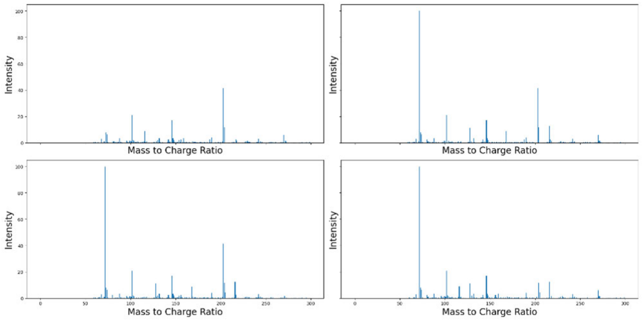

Fig. 7. The examples above are four different noisy spectra all based on the same original spectrum. The original base peak is present in all but one of the noisy spectra.

Figure 7 shows four different noisy versions of the same input spectrum. Each noisy spectrum contains the majority of the peaks from the original spectrum but omits some of the original peaks. Encoding m noisy spectra for each of n original spectra resulted in $n * m$ embeddings, significantly increasing the number of synthetic spectra we could generate.

Our decoder requires one embedding to generate one synthetic spectrum. Had we encoded only the n original spectra in our dataset, we would have produced n embeddings, thereby limiting the number of synthetic spectra to the number of original spectra. To overcome this limitation, we created multiple noisy versions of each training spectrum.

The original decoder architecture, outlined in Table II, appeared to converge and generate synthetic spectra that visually resembled the target lab-generated spectra. When evaluated by Human Metabolome Database's (HMDB) spectrum match tool, however, the synthetic spectra were not correctly identified. While our primary objective was to generate synthetic spectra that would improve the performance of a classifier model - a lower threshold than correct identification by a spectrum match tool - we were also interested in creating realistic spectra. Consequently, we explored alternative approaches in an effort to address this limitation.

Final hyperparameters selected for generalized decoder are displayed in Table II. One particularly impactful hyperparmeter was the percent of peaks removed from the spectra used to generate the decoder's input embeddings. We found that removing 15% of peaks from the spectra used to generate embeddings struck a balance between information retention and simply reproducing the original spectra.

TABLE II
GENERALIZED DECODER HYPERPARAMETERS.

Size of input layer	512 x 541
Size of output layer	765 x 797
Number of hidden layers	7
Sizes of hidden layers	$541 \times 573, 573 \times 605, 605 \times 637,$ $637 \times 669, 669 \times 701,$ $701 \times 733, 733 \times 765$
Learning rate	1×10^{-3}
Training time	500 epochs
Batch size	32
Activation function	Leaky ReLU
Loss criterion	MSE + λ Weighted Cosine Loss
% of peaks removed from embedding generation spectra	15%

C. Per chemical Specialized Decoders

As part of our work we noted that training a generalized decoder to generate spectra for all chemicals in our dataset lead to sub-optimal results. While the spectra created by the generalized decoder visually resembled their lab-generated target spectra, they were not correctly identified by the HMDB's spectrum match tool. As we learned more about the complex and nuanced patterns present within mass spectra for a single chemical, we determined that this level of complexity, scaled across the 40 chemicals in our dataset, may pose a challenge for a single generalized model to accurately learn and represent.

Informed by these experiments, we opted to forgo our generalized decoder for a series of more specialized decoders, each trained to generate spectra for a specific chemical or group of similar chemicals. We posited that these specialized models would capture complexities that the generalized model was not able to.

Although the specialized models converged and generated spectra with high weighted cosine similarities to the original spectra, we observed an unintended behavior when we examined the generated spectra. We noticed that each model's output consisted of a small number of identical or near identical spectra that did not represent the diversity of the training data. Figure 8 shows an example of two identical spectra generated by our model for different input spectra. The decoders had learned to generate a limited set of generic spectra that matched many of the training examples, thereby limiting the loss on the training examples without capturing the underlying patterns in the data.

This behavior, known as memorization, is a frequent issue for models trained on small datasets [19]. A common approach to preventing memorization in low-data scenarios is to allow the model with limited training data to leverage features learned by a base model trained on a similar task from a

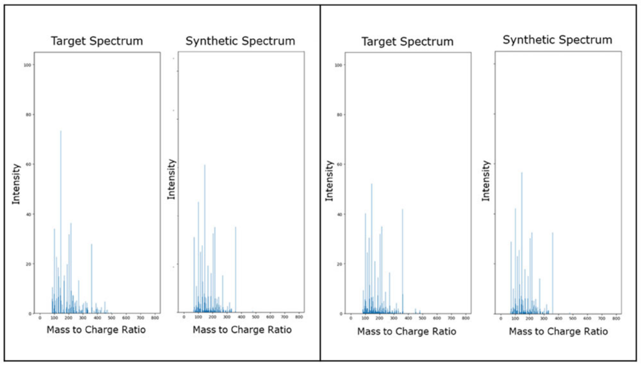

Fig. 8. The figure above shows two synthetic spectra and their respective target spectra. We observe that, while the target spectra are distinct from one another, the two synthetic spectra are identical.

domain where data is more easily obtained [20], [26]. We applied this approach in our work by first training a general model on the entire dataset. We then leveraged the weights from the general model as a base that each of the specialized models could fine-tune to incorporate the complexities of a single chemical's spectra.

The architecture and hyperparameters of the general decoder can be found in Table II. The specialized decoders initialized their weights using the weights learned by the general decoder. Hyperparameters were the same across the general and specialized models, except for those hyperparameters specified in Table III. A notable difference between the general and specialized models is that a subset of the specialized decoders' layers were frozen, preventing the model from updating weights for those layers during training. This strategy was implemented to prevent overfitting and preserve low-level features learned by the base model that were potentially applicable to any of the specialized models [8].

TABLE III
SPECIALIZED DECODER HYPERPARAMETERS THAT DIFFER FROM THE GENERAL DECODER.

Sizes of trainable hidden layers	$512 \times 541, 541 \times 573, 573 \times 605,$ $605 \times 637, 637 \times 669$
Sizes of frozen hidden layers	$669 \times 701, 701 \times 733,$ $733 \times 765, 765 \times 797$
Learning rate	1×10^{-4}
Training time	300 epochs

V. Numerical Results

A. Accuracy of Synthetic Spectra Compared to Target Spectra

Using the training scheme outlined in Section IV, we generated synthetic spectra with high similarity scores that were not overfit to the training data. The outlined spectrum in Figure 1 is an example of the synthetic spectra generated by our specialized decoders from embeddings corresponding to spectra reserved as test data. The quality of our generated spectra was further validated through correct classifications on 88% of the synthetic spectra identified by the HMDB's spectrum match tool.

As a benchmark for evaluating our model's performance, we compared the spectra generated by our model to spectra generated by a Variational Autoencoder trained on the same dataset. All hyperparameters except for loss criterion were consistent between the models. The loss criterion for our encoder was Mean Squared Error (MSE), and our decoder used a combined MSE and weighted cosine loss. The VAE's loss criterion was a combination of reconstruction loss (MSE) and a regularization term expressed as the Kullback-Leibler divergence [23].

TABLE IV
SIMILARITY OF SYNTHETIC SPECTRA TO TARGET SPECTRA ACROSS MODELS.

Synthetic Spectra Generation Model	Average Weighted Cosine Similarity	Average Mean Squared Error	Spectra Above .9 Similarity Ratio
Decoders	.82	5.8	86%
VAE	.43	12.74	11%

Table IV compares the Average Weighted Cosine Similarity, Mean Squared Error and similarity ratio between synthetic spectra and their target spectra for both our models and the Variational Autoencoder. Our models significantly outperformed the VAE in both similarity metrics.

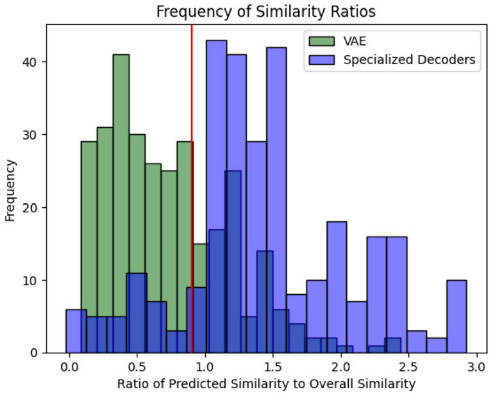

Fig. 9. The figure above compares the similarity between predicted and true spectra to the similarity between all true spectra for the same chemical for spectra generated by decoders and the VAE. Decoder-generated spectra had higher similarity ratios than the VAE spectra, indicating closer resemblance between decoder spectra and target spectra than between VAE spectra and target spectra. A similarity ratio of 1 indicates that the predicted spectrum is as similar to the target spectrum as all of that chemical's spectra are to each other. The frequency of high similarity ratios for decoder-generated spectra is likely due to the fact that GC-MS spectra can vary significantly between samples.

These similarity results are consistent with the hypothesis that Chemception embeddings capture a richer representation of chemical compounds compared to embeddings a model learned directly from the data. Our findings suggest that this richer representation enabled the creation of higher-quality synthetic data than a similar model trained on our limited dataset.

B. Impact of Synthetic Spectra on Classifier Accuracy

In the analysis of our synthetic spectra we addressed two objectives:

1) Determine the impact of our spectra on the performance of a chemical classifier when incorporated into its training data. Evaluating the impact of our spectra allowed us to determine whether we had met our original research goal of creating a mass spectrometry data generation model whose output would enhance the performance of a chemical classifier.
2) Evaluate the impact of our synthetic spectra relative to synthetic spectra generated by another model. Our analysis provided insight into the comparative strength

TABLE V
COMPARISON OF CLASSIFIER ACCURACY DATASETS AUGMENTED BY VAE AND SPECIALIZED DECODER-GENERATED SPECTRA.

Classification Model	Data Generation Model	Random Guess	Real Data Only	Synthetic Data Only	Real and Synthetic Data	Δ Classifier Accuracy
Random Forest	**Our Decoder**	2.5%	72%	57%	79%	7%
	VAE			10%	72%	0%
Neural Network	**Our Decoder**		75%	74%	84%	9%
	VAE			9%	76%	1%

of our model's output against existing data generation models.

To conduct our analysis we compared the performance of Random Forest and Neural Network classifiers on datasets of exclusively lab-generated spectra, exclusively synthetic spectra, and a mix of lab-generated and synthetic spectra. We established four metrics to act as performance baselines. Model performance on each baseline is displayed in Table V and described below:

The first baseline metric was probability of chance identification, or the likelihood of correctly identifying a spectrum by random chance, estimated at $1/40$ given the 40 chemicals in our training dataset. Achieving higher than random accuracy is particularly relevant for the models trained exclusively on synthetic data as it indicates that the synthetic data provide non-trivial information to the classifier. Classifier accuracy exceeded the accuracy that could be achieved by randomly guessing for all models and datasets tested.

Our second baseline was the test accuracy achieved by classifiers trained exclusively on VAE-generated spectra. We compared our second baseline against the test accuracy of classifiers trained exclusively on decoder-generated data. We found that both the Random Forest and Neural Network exhibited higher test accuracy when trained exclusively on decoder spectra compared to their counterparts trained on VAE data, suggesting that decoder-generated spectra may capture spectral features more effectively than VAE-generated spectra. The difference in accuracy was particularly striking for the Neural Network, whose test accuracy was 65% higher when trained on decoder spectra compared to VAE spectra.

The third baseline we employed was the test accuracy of classifiers trained only on lab-generated data. By comparing the test accuracy of classifiers trained on decoder-augmented datasets against this baseline, we were able to address our first analytical objective regarding the impact of our synthetic data on classifier performance. We found that both the Random Forest and the Neural Network were more accurate when their training data was augmented with decoder-generated spectra, establishing that the output of our mass spectrometry data generation model enhances the performance of a chemical classifier.

As our fourth, and arguably most important, metric, we employed the test accuracy achieved by classifiers trained on a VAE-augmented dataset. Comparing classifier performance on a VAE-augmented datasets and decoder-augmented datasets allowed us to evaluate the impact of our decoder spectra against the impact of synthetic spectra generated by another model. The classifiers trained on decoder-augmented datasets exhibited higher test accuracy than the classifiers trained on VAE-augmented datasets. The difference in test accuracy between the two training datasets was 7% for the Random Forest and 8% for the Neural Network. Here we address our second analytical objective to determine that the accuracy impact of augmenting with decoder spectra was greater than the impact of augmenting with spectra generated by another model.

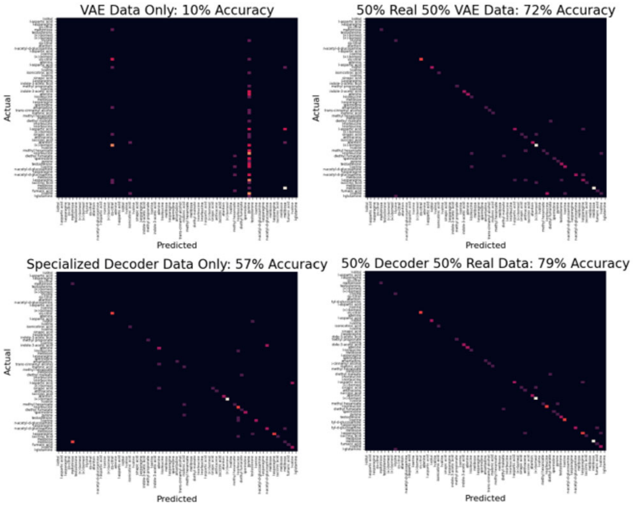

Fig. 10. Confusion matrices for the Random Forest classifier when trained using synthetic data from the VAE vs. synthetic data from specialized decoders. The models trained on specialized decoder-generated spectra demonstrated higher accuracy scores compared to models trained on VAE-generated spectra.

During testing we determined that augmenting with increasing amounts of synthetic data generated increases in classifier accuracy until the number of synthetic spectra in the training dataset equaled the number of lab-generated spectra. After that point we observed no further improvement in accuracy.

Table V shows a comparison of classifier accuracy across each of the 10 models trained. For both the Random Forest and the Neural Network, the highest performing model was the model trained on an augmented dataset of lab-generated and specialized decoder-generated spectra. Figure 10 visually demonstrates the increased accuracy of the Random Forest model when trained on decoder-generated spectra compared to VAE-generated spectra.

It is important to clarify that we do not contend that incorporation of our synthetic spectra leads to greater improvements in classifier accuracy than incorporation of additional lab-generated spectra. Our central claim is that, in scenarios where additional lab-generated data is not available, supplementing with our synthetic spectra can effectively bridge this gap and bolster classifier performance.

VI. CONCLUSION

The outcomes of this research suggest that using a predetermined embedding in an autoencoder-like framework, can lead to effective generative algorithms for real-world problems, such as mass spectrometry spectra generation. In particular, our numerical results lead us to believe that our model is useful in improving real world outcomes for using mass spectrometry data to identify hazardous chemicals using machine learning algorithms. In fact, the flexibility supplied to autoencoders by a predetermined hidden layer provide several benefits, including separability in training and superior generative performance. However, there is a additional important benefit for future work, namely the ability to map multiple data sets to single representation.

A. *Future Work*

Chemical classification models are typically trained on data from a single sensor type (e.g. mass spectrometry, polymer sensor, etc.). As a result, data collected on other types of sensors would not be accessible to the classifier. Our larger goal, beyond data generation, was to design a data generation method that could also be used to translate data between sensor formats, making data from multiple formats accessible to the same classifier. We theorized that taking advantage of the information stored within the Chemception embeddings might assist us in creating such a data generation and format translation method.

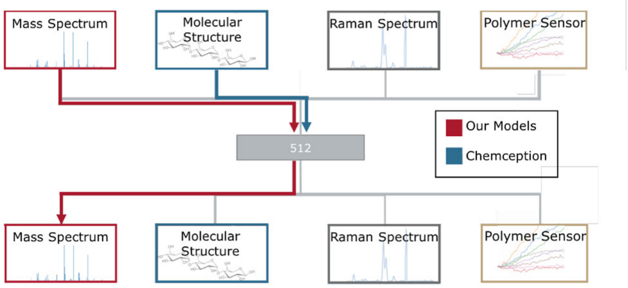

Fig. 11. The above figure shows the proposed modular architecture for translating chemical data between chemical sensor formats using Chemception embeddings as a shared hidden layer. The red, blue and gray branches indicate our implemented models, the Chemception model and proposed future extensions, respectively.

Figure 11 depicts the conceptual design of our translator, where each of the upper branches represents an encoder E mapping data *to* 512 dimensional Chemception embeddings. Each of the bottom branches represents a decoder D mapping *from* a 512 dimensional Chemception embedding to its respective data type. The branch highlighted in blue corresponds to the Chemception model, which maps stick and ball data to the shared hidden layer. The red branches indicate our implemented and validated models, which map mass spectrometry data to and from Chemception embeddings. The remaining gray branches represent proposed future extensions to the architecture.

Further areas for exploration include the classification accuracy impact of varying ratios of real to synthetic data within the classifier's training dataset. In addition, we are interested to examine what effect augmenting our fixed hidden layer with sensor-specific embeddings might have on data generation.

VII. Acknowledgements

This manuscript has been authored under partnership with DEVCOM SC funded by the Defense Threat Reduction Agency (DTRA). The publisher acknowledges that the US Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. Approved for public release. Distribution unlimited.

References

[1] G. Andrew, R. Arora, and J. Bilmes, "Deep canonical correlation analysis," in *International Conference on Machine Learning*, PMLR, 2013.

[2] T. Annesley, "Mass Spectrometry in the Clinical Laboratory: How Have We Done, and Where Do We Need to Be?," *Clinical Chemistry*, vol. 55, no. 6, pp. 1236–1239, 2009. DOI: 10.1373/clinchem.2009.127522.

[3] R. Babbar and B. Schölkopf, "Data Scarcity, Robustness and Extreme Multi-label Classification," *Machine Learning*, vol. 108, pp. 1329–1351, 2019. DOI: 10.1007/s10994-019-05791-5.

[4] Defense Threat Reduction Agency, "Defense Threat Reduction Agency," https://www.dtra.mil/.

[5] B. M. Dillon et al., "Better Latent Spaces for Better Autoencoders," *SciPost Physics*, vol. 11, no. 3, p. 061, 2021.

[6] G. B. Goh, C. Siegel, A. Vishnu, N. O. Hodas, and N. Baker, "Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-Developed QSAR/QSPR Models," *arXiv preprint*, arXiv:1706.06689, 2017.

[7] Garg E, Zubair M. Mass Spectrometer. [Updated 2023 Jan 21]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2024 Jan-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK589702/

[8] K. Goutam, S. Balasubramanian, D. Gera, and R.R. Sarma, (2020). Layerout: Freezing layers in deep neural networks. SN Computer Science, **1**(5), 295.

[9] Human Metabolome Database, "Human Metabolome Database (HMDB) - MS Search," https://hmdb.ca/spectra/c_ms/search, Accessed: 2024-05-20, 2024.

[10] M. Ljoncheva et al., "Machine Learning for Identification of Silylated Derivatives from Mass Spectra," *Journal of Cheminformatics*, vol. 14, no. 1, p. 62, 2022.

[11] M. G. Madden and T. Howley, "A Machine Learning Application for Classification of Chemical Spectra," in *Applications and Innovations in Intelligent Systems XVI. SGAI 2008*.

[12] B. P. Mayer et al., "Toward Machine Learning-Driven Mass Spectrometric Identification of Trichothecenes in the Absence of Standard Reference Materials," *Analytical Chemistry*, vol. 95, no. 35, pp. 13064–13072, 2023.

[13] A. D. McEachran, I. Balabin, T. Cathey, et al., "Linking in silico MS/MS spectra with chemistry data to improve identification of unknowns," *Scientific Data*, vol. 6, p. 141, 2019. DOI: 10.1038/s41597-019-0145-z.

[14] MO NA - Fiehn Lab. Retrieved from https://mona.fiehnlab.ucdavis.edu/downloads
Accessed: 2024-06-13.

[15] A. Moore, R. Paffenroth, K. Ngo, and J. Uzarski, "ACGANs Improve Chemical Sensors for Challenging Distributions," in *IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2022.

[16] A. Moore, R. Paffenroth, K. Ngo, and J. Uzarski, "Cycles Improve Conditional Generators: Synthesis and Augmentation for Data Mining," in *Advanced Data Mining and Applications (ADMA)*, 2022.

[17] A. Moore, R. Paffenroth, K. Ngo, and J. Uzarski, "ChemTime: Semantic Sequences Outperform Multivariate Time Series Classifiers for Chemical Sensing," *arXiv preprint*, arXiv:2312.09871, 2023.

[18] A. M. Moore et al., "ChemVise: Maximizing Out-of-Distribution Chemical Detection with a Novel Application of Transfer Learning," in *2023 International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2023.

[19] Radhakrishnan, A., Yang, K., Belkin, M., and Uhler, C. (2018). Memorization in overparameterized autoencoders. *arXiv preprint arXiv:1810.10333*.

[20] A. S. B. Reddy and D. S. Juliet, "Transfer Learning with ResNet-50 for Malaria Cell-Image Classification," 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019, pp. 0945-0949, doi: 10.1109/ICCSP.2019.8697909.

[21] W. Reusch, "Mass Spectrometry," https://www2.chemistry.msu.edu/faculty/reusch/virttxtjml/spectrpy/massspec/masspec1.htm.

[22] University of Hawaii at Manoa. "Compare, Contrast, Connect: Chemical Structures - Visualizing the Invisible." Exploring Our Fluid Earth. Accessed June 30, 2024. https://manoa.hawaii.edu/exploringourfluidearth/chemical/chemistry-and-seawater/covalent-compounds/compare-contrast-connect-chemical-structures-visualizing-invisible.

[23] Van Erven, T., & Harremoes, P. (2014). Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, 60(7), 3797-3820.

[24] Z. Wan, Y. Zhang, and H. He, "Variational Autoencoder Based Synthetic Data Generation for Imbalanced Learning," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2017.

[25] J. N. Wei, D. Belanger, R. P. Adams, and D. Sculley, "Rapid Prediction of Electron–Ionization Mass Spectrometry Using Neural Networks," *ACS Central Science*, vol. 5, no. 4, pp. 700–708, 2019. DOI: 10.1021/acscentsci.9b00085.

[26] Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. Journal of Big data, 3, 1-40.