Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Raman Spectroscopy Pre-Trained Encoder: A Self-Supervised Learning Approach For Data-Efficient Domain-Independent Spectroscopy Analysis

ABHIRAAM ERANTI[1 1],YOGESH TEWARI [2](Senior Member, IEEE),RAFAEL PALACIOS[3,4],and AMAR GUPTA[5,6](Life Fellow, IEEE)

[1]University of California, Berkeley, 110 Sproul Hall, Berkeley, CA 94720 USA (e-mail: abhiraam_eranti@berkeley.edu)
[2]Cloud Data Engineering, Google, 1600 Amphitheatre Parkway, Mountain View, CA 94043 USA (e-mail: tewariy@google.com)
[3]Cybersecurity at MIT Sloan (CAMS), Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139 USA (e-mail: palacios@mit.edu)
[4]Institute for Research in Technology, Universidad Pontificia Comillas, Alberto Aguilera 23, 28015 Madrid, Spain (e-mail: palacios@mit.edu)
[5]Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139 USA (e-mail: agupta@mit.edu)
[6]Distinguished Professor Institute for Applied AI Innovation University of Texas El Paso, TX, USA

Corresponding author: Abhiraam Eranti (e-mail: abhiraam_eranti@berkeley.edu).

ABSTRACT Deep-learning methods have boosted the analytical power of Raman spectroscopy, yet they still require large, task-specific, labeled datasets and often fail to transfer across application domains. The study explores pre-trained encoders as a solution. Pre-trained encoders have significantly impacted Natural Language Processing and Computer Vision with their ability to learn transferable representations that can be applied to a variety of datasets, significantly reducing the amount of time and data required to create capable models. The following work puts forward a new approach that applies these benefits to Raman Spectroscopy. The proposed approach, RSPTE (Raman Spectroscopy Pre-Trained Encoder), is designed to learn generalizable spectral representations without labels. RSPTE employs a novel domain adaptation strategy using unsupervised Barlow Twins decorrelation objectives to learn fundamental spectral patterns from multi-domain Raman Spectroscopy datasets containing samples from medicine, biology, and mineralogy. Transferability is demonstrated through evaluation on several models created by fine-tuning RSPTE for different application domains: Medicine (detection of Melanoma and COVID), Biology (Pathogen Identification), and Agriculture. As an example, using only 20% of the dataset, models trained with RSPTE achieve accuracies ranging 50%–86% (depending on the dataset used) while without RSPTE the range is 9%–57%. Using the full dataset, accuracies with RSPTE range 81%–97%, and without pre-training 51%–97%. Current methods and state-of-the-art models in Raman Spectroscopy are compared to RSPTE for context, and RSPTE exhibits competitive results, especially with less data as well. These results provide evidence that the proposed RSPTE model can effectively learn and transfer generalizable spectral features across different domains, achieving accurate results with less data in less time (both data collection time and training time).

INDEX TERMS Raman Spectroscopy, self-supervised learning, pre-trained encoder, multi-domain data, clinical diagnostics

I. INTRODUCTION

Raman spectroscopy has emerged as a promising technique for analysis and identification across diverse fields, including agriculture [1], mineralogy [2], chemistry [3], biology [4, 5, 6], and medicine [7, 8, 9]. Its nondestructive, minimally invasive, and highly specific molecular insights enable detailed analysis of chemical composition and structure across various samples. In biological systems, it probes the biomolecular content of cells [7], tissues [10], and biomolecules, facilitating the identification of cellular characteristics, disease markers [11], species [5, 12], and treatment responses [13].

Deep learning employed in Raman spectroscopy have enhanced analytical performance through automated feature extraction and classification. [14, 15, 16]. Conventional deep-learning approaches utilize techniques like Convolutional Neural Networks and ResNets, which can extract features from Raman spectra for efficient classification [2, 5]. Recent approaches leverage techniques like Transformers and Multi-head Attention mechanisms [17, 18] to capture long-range dependencies within spectral data, enabling superior performance over conventional methods in certain benchmarks [19]. In addition, feature engineering and spectral transformation techniques like Wavelet Transforms [20] and Gramian Angular Fields [21] have helped simplify data complexity and enhance subtle spectral features, which have also improved performance [2, 5].

However, conventional approaches typically require large, labeled datasets for training, which can be expensive and time-consuming to acquire (especially for complex biological samples). For example, Ho et al. [4] manually collects over 60,000 samples for training and 3000 samples for fine-tuning; Berlanga et al. aggregates over 80,000 spectra for mineral classification [2]; and Qiu et al. utilizes 20,000 spectra for training and analysis [7]. Reaching the scale of these datasets requires significant time and effort; such label volumes are unattainable for many laboratories, especially when samples are rare, costly, or patient-derived. Moreover, these resultant models are often domain-specific, limiting their adaptability to new analytical tasks. Several methods that improve results on Raman spectroscopy benchmarks do not specifically address the large data issue and focus mainly on improving model or preprocessing steps [4, 5, 19] or creating custom-built training datasets to solve in-domain problems [2, 4, 7].

A promising solution is the recent development of pre-trained encoders in natural language processing and computer vision, which leverage transfer learning by training on large volumes of unlabeled, out-of-domain data. These models, such as BERT in NLP (Natural-Language processing) [22] and CLIP in computer vision [23], learn rich, generalizable, and information-dense representations that can be fine-tuned for a wide variety of downstream tasks [24]. Pre-trained encoders are often trained using self-supervised learning algorithms, which use the structure of unlabeled data itself to learn valuable insights. For example, models may be trained to predict masked words in a sentence [22] or similarities within an image [25].

Models pre-trained on large, unlabeled data learn general patterns and, when fine-tuned on smaller in-domain data, achieve accuracy comparable to training from scratch on large datasets, but with significantly less computing power [24]. It is important to note that while training the pre-trained encoder is time-consuming and computationally expensive, particularly for complex models that rely on several broad, domain-relevant dataset, the process is a worthwhile investment. Pre-training reduces the time and computational cost of developing new models by enabling efficient fine-tuning for specific applications.

This work leverages the benefits of self-supervised out-of-domain pre-training by introducing Raman Spectroscopy Pre-Trained Encoder (RSPTE), a proposed self-supervised pre-trained encoder designed for data-efficient Raman Spectroscopy analysis. It is hypothesized that by pre-training on a large corpus of unlabeled, out-of-domain data, RSPTE can learn key spectral information and transfer well into downstream tasks.

A common occurrence for models trained on limited data is the tendency for the model to overfit on the sampled distribution without considering the distribution of the data as a whole. With such a small sample, even if the model performs well on the sample itself, it may fail to extrapolate its insights on the broader dataset (as shown in Fig. 1). Pre-training improves performance by leveraging insights from related datasets that may belong to different domains but share useful similarities.

By creating a generalizable pre-trained encoder that can learn key spectral representations which capture fundamental information about spectroscopy, RSPTE can be used as a pre-trained backbone, leveraging transferable knowledge to accelerate domain adaptation–potentially allowing for the creation of effective Raman Spectroscopy detectors with reduced data, time, and effort.

The main contributions of this work are as follows:

1) Adaptation of the Barlow Twins framework to 1D Raman spectral data and introduction of a pipeline of five physically motivated spectral transforms designed to simulate raman-specific instrument variability while preserving chemical identity.
2) Axis-robust cross-domain pre-training strategy to learn transferable spectral primitives (peak shape, width, intensity ratios) rather than dataset-specific embeddings.
3) A systematic data-efficiency evaluation across four downstream domains across pathogen identification, medical diagnostics, cancer subtyping, and agricultural classification.
4) Comparative analysis against classical and state-of-the-art approaches (PCA+KNN, SVM, CNN, and RamanNet).

II. MATERIALS AND METHODS

A. DATASETS

RSPTE was pre-trained on a mixed multi-domain Raman spectroscopy dataset compiled from several established sources, including spectra from:

- MDA-MB-231 breast cancer cell data [26], containing approximately 160k spectra for training
- RRuFF [27], containing approximately 8k Raman spectra of various minerals.

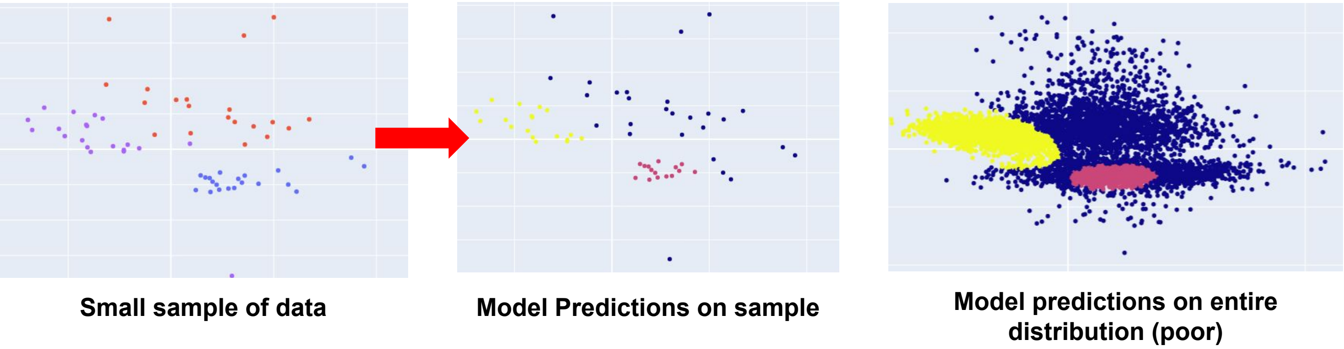

FIGURE 1. Justification for the recommendation of pre-training in data-scarce situations. The model's predictions after from-scratch training on limited data renders suboptimal generalization to the data distribution as a whole. Pre-training could potentially help by supplying transferrable information from other domains about spectral features and characteristics.

- Adenine data [3], containing approximately 4k Raman spectra of different concentrations of the adenine nucleotide.
- MLROD [2], containing approximately 78k Raman spectra of various rocks and minerals.

Together, these datasets provided a sufficiently balanced distribution of Raman spectra originating from a variety of sources, potentially allowing the proposed RSPTE approach to learn fundamental spectral patterns. Once the RSPTE model was pre-trained, it was then used for fine-tuning on different fields in which Raman spectroscopy is used. Several datasets were evaluated to understand the performance of fine-tuning after RSPTE:

- **Bacteria-ID** data [4], which was used as a reference for benchmarks for several spectroscopy papers [4, 5, 19]. The dataset contained Raman spectra from 30 common microbial species. The fine-tune and test splits are relatively small, making this dataset suitable for data-scarce evaluation.
- **Covid-19** data [28], including Raman spectra of individuals diagnosed with Covid-19, individuals suspected of having Covid-19, and healthy individuals as controls. Suspected individuals were not included (thus making the analysis a two-class problem) due to unreliable labels, potentially causing issues with downstream fine-tuning.
- **Melanoma** data [29], consisting of Raman spectra of both healthy (control) and 12 variations of melanoma cells. While the authors collected three spectra per sample using AuMs functionalized with ADT-NH$_2$, ADT-COOH, and ADT-(COOH)$_2$, only spectra collected with ADT-COOH functionalisation were used, as this subset exhibits the highest inter-class similarity and therefore poses a more challenging discrimination task [30]. Evaluating RSPTE's performance under these conditions could potentially provide insight into its ability to distinguish subtle spectral variations.
- **Wheat Lines** data [1], including Raman spectra from a commercial wheat cultivar and a mutant strain, both in raw form and treated with 125mM NaCl. After analyzing the data embeddings, raw and salt-treated spectra for each strain were pooled, yielding a two-class problem (commercial cultivar vs mutant). Raw and salt-treated spectra for each class were pooled for theoretical and experimental reasons. Salt-treating does not significantly change sample properties and is done to enhance analysis - thus, a model trying to identify raw vs salt-treated data would identify treatment-method rather than spectra type, which was the desired task of the dataset. Additionally, experimental observation of the data revealed two large clusters (mutant vs original); raw and salt-treated data were simply slightly shifted from each other yet didn't significantly change spectral properties to warrant treatment as a different class entirely.

As shown in Fig. 2, the spectra used in the dataset exhibited differences in signal-to-noise (SNR), artifacts, peaks, and baseline shifts. Some notable examples include:

- Varying SNR ratios: For example, the spectra from MLROD dataset had a very high SNR, often represented as smooth curves with well-defined peaks; on the other hand, the data from the MDA-MB-231 dataset had a low SNR, with a very noisy signal.
- Artifacts: The data from MLROD sometimes contained cosmic rays, depicted as sharp spikes. These could affect downstream preprocessing tasks as described in Section II-C, and had to be removed.
- Some datasets contained baseline shifts, often from external factors like fluorescence distortion. These shifts were not expressly removed as they did not necessarily impede the preprocessing steps.
- Peaks: Some datasets, like Adenine and RRuFF, exhibited several peaks/spectra, but some only had one principal peak, like the MDA-MB-231 dataset [26]

The choice of the datasets for pre-training and fine-tuning was intentional because of the different domains from which the individual datasets come from. The main objective was to evaluate whether RSPTE could learn generic spectral

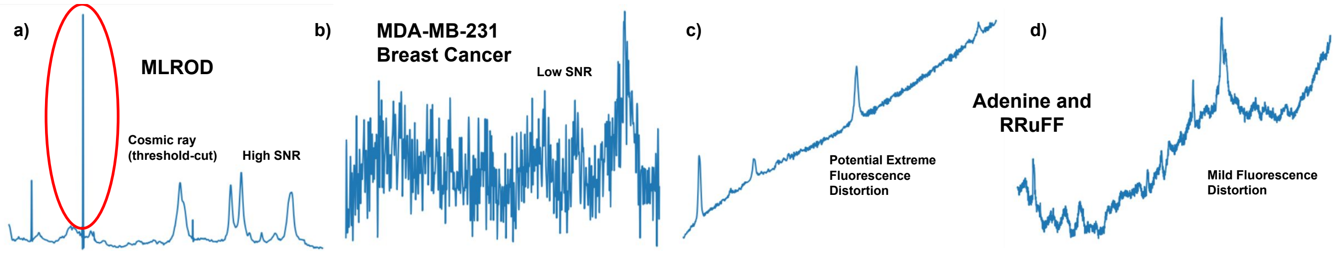

FIGURE 2. Representative spectra used to pre-train RSPTE (intensity vs. Raman shift); left to right: (a) Cosmic-ray artifact from MLROD (characteristic narrow, high-amplitude spike), (b) Breast-cancer spectrum from MDA-MB-231 (low-SNR signal with weak but present bands), (c) Extreme fluorescence distortion (steep, baseline-dominated profile), and (d) Mild fluorescence distortion from Adenine/RRuFF (moderate baseline with clear adenine features).

features from a diverse dataset during pre-training, and successfully transfer such knowledge to several downstream models during fine-tuning. This approach aimed to demonstrate cross-domain transferability without relying on a large quantity of labeled, in-domain data for full training.

B. METHODS

The Barlow Twins approach of self-supervised pre-training from Zbontar et al. [25] was chosen for its independence from positive-negative pairs and explicit focus on redundancy reduction.

1) Overview of Barlow Twins

Many conventional metric-learning algorithms require the use of positive-negative pairs and learn embeddings by training the models to create similar embeddings for positive pairs of images and different embeddings for different ones [31], measuring “similarity” via metrics like negative cosine similarity. However, these approaches often require labels to create the pairings or make assumptions about the similarity of data within batches which are potentially incorrect [31].

Barlow Twins does not require the use of any labels or positive-negative comparisons during the training process. Instead, two augmented versions of the same data sample are created, which are both passed into the model to create a pair of embeddings. Because the embeddings come from the same sample, and are thus from the same class, the two embeddings are expected to be similar to each other. If the embeddings are dissimilar, it means that the model has not yet effectively learned to generate accurate representations.

Barlow Twins uses a cross-correlation matrix to measure both the similarity between two embeddings and their redundancy. As illustrated in Fig. 3, the ideal cross-correlation matrix would be in the form of an identity matrix, where strong correlations exist only along corresponding parts of embeddings and minimal correlation between non-corresponding pairs. To achieve this, the Barlow loss function penalizes values less than one for correlations among corresponding parts and nonzero values for correlations outside the diagonal of the matrix (decorrelating features within the embedding). Detailed pseudocode of the loss function is shown in Algorithm 1.

2) Raman Spectroscopy-specific changes

Several modifications were applied in order to adapt the Barlow Twins architecture to Raman Spectroscopy. Instead of using the ResNet-50 architecture described in the reference paper [25], a base ResNet-34 adapted from [32] was utilized instead. While ResNet is a common architecture that has demonstrated successful results in Raman spectroscopy classification [4, 14], the size of the ResNets used in these established papers are significantly smaller than ResNet-50 (used in Zbontar et al. on ImageNet [25], several orders of magnitude larger and more complex than this study's pre-training data). Given the size and shape of the RSPTE pre-training dataset, a ResNet-34, better adapted for medium-sized datasets, was used instead.

In addition, because pre-training was conducted on 1D Raman Spectra instead of 2D images, the base ResNet-34 architecture was adapted to work on 1D data by substituting 2D layers with corresponding 1D versions, using Conv1D instead of Conv2D; ZeroPadding1D instead of ZeroPadding2D; MaxPool1D instead of MaxPool2D; and AveragePooling1D instead of AveragePooling2D.

Furthermore, conventional image augmentations such as rotation, flipping, and color shifting have no meaningful analogue in 1D spectral data, necessitating domain-specific transforms. Augmentations were chosen to introduce realistic variability while preserving the fundamental spectral identity of each sample. Each spectrum was stochastically augmented using a pipeline of five Raman-specific transforms applied with independent probabilities:

1) Gaussian noise ($\sigma = 0.01$) was added with approximately 50% probability to simulate detector noise.
2) Average blurring with a random kernel size between 2 and 5 was applied with approximately 25% probability to mimic resolution variation across instruments.
3) A wavenumber-axis shift of ±5 points was applied with approximately 25% probability to simulate small calibration offsets. The shift mirrors calibration differences across Raman spectroscopy equipment rather than a large signal translation.
4) Random zeroing, in which 2% of spectral points were set to zero, was applied with approximately 50% probability to encourage robustness to missing data.
5) Random intensity scaling by a factor drawn uniformly

from $[0.9, 1.1]$ was applied with approximately 25% probability to simulate intensity variation.

On top of the ResNet-34 Encoder, a projector was applied that took the output and transformed it into a final embedding of 2048 neurons. The projector is a key part of the Barlow Twins algorithm [25] and consists of 2 Dense-BatchNormalization-Relu blocks, followed by a final Dense layer to generate the output embeddings.

TABLE 1. Raman spectroscopy datasets and spectral windows used in this work. Evaluation-only datasets were not utilized in pre-training or in the UMAP analysis to select the number of spectral features. Heterogeneity was handled through resampling, normalization, and light removal of extreme cosmic rays. Samples within the RRuff, Adenine, and MLROD datasets had different input shapes (amount of spectral features per sample) and spectral windows (starting spectrum and ending spectrum values for the sample), so the range ($\mu \pm \sigma$) of spectral count and spectral window is provided.

Set	Dataset	No. spectra	Spectral window (cm^{-1})
Train	MDA-MB-231	500	100–1800
Train	RRuff	1186 ± 282	108 ± 45–1520 ± 148
Train	Adenine	1756 ± 2525	325 ± 174–2386 ± 748
Train	MLROD	1766 ± 465	102 ± 18–1253 ± 256
Eval	Covid	900	400–2112
Eval	Wheat Lines	1748	296–2043
Eval	Melanoma	2089	605–1722
Eval	Bacteria	1000	382–1792

Algorithm 1 Barlow Twins Loss Function

```
Require: Embedding batches $Z_A, Z_B \in \mathbb{R}^{B \times D}$, redundancy weight $\lambda$
Ensure: Scalar loss $\mathcal{L}$
$\bar{Z}_A \leftarrow \frac{Z_A - \mu(Z_A)}{\sigma(Z_A)}$ {Normalize along batch dimension}
$\bar{Z}_B \leftarrow \frac{Z_B - \mu(Z_B)}{\sigma(Z_B)}$
$\mathcal{C} \leftarrow \frac{\bar{Z}_A^\top \bar{Z}_B}{B}$ {Cross-correlation matrix $\mathcal{C} \in \mathbb{R}^{D \times D}$}
$\mathcal{L}_{\text{inv}} \leftarrow \sum_i (\mathcal{C}_{ii} - 1)^2$ {Invariance: diagonal $\to 1$}
$\mathcal{L}_{\text{red}} \leftarrow \lambda \sum_{i \neq j} \mathcal{C}_{ij}^2$ {Redundancy: off-diagonal $\to 0$}
$\mathcal{L} \leftarrow \mathcal{L}_{\text{inv}} + \mathcal{L}_{\text{red}}$
return $\mathcal{L}$
```

C. TRAINING

In order to use the multi-domain dataset, the samples had to be preprocessed in a way that could be accepted by RSPTE.

The first issue stemmed from the fact that the number of spectral features varied for each dataset. For example, the MDA-MB-231 breast cancer dataset had 500 spectral features, but the **Bacteria-ID** dataset had 1000 spectral features. This was a problem because the RSPTE architecture, which uses a ResNet projection head, could only accept a fixed input-shape, meaning the dataset spectra had to be resampled such that they all had the same number of spectral features. A matter which had to be determined was the final number of spectral features to resample to; if the number chosen was too small, RSPTE could risk losing potentially valuable spectral information; however, if the number was too large, the model could fail to generalize due to the excess input shape of data, potentially "overwhelming" the Resnet-34 with input features. Additionally, excessively large spectral features could cost valuable compute time and memory.

To determine the ideal number of spectral features of pretraining, input shape analysis (Table 1) and multiple test runs of 500, 1000, and 2000 spectral features (representative feature sizes within our data) were conducted. Pretrained encoder performance was evaluated by visualizing the model's embeddings using the UMAP dimensionality-reduction algorithm as an indicator of the model's ability to transfer information. The Bacteria-ID's reference dataset was used for the visualization evaluation because it was not utilized in pre-training or fine-tuning benchmarks. After qualitatively analyzing the UMAP-reduced embeddings, resampling the data to 1000 spectral features was determined to provide an optimal balance between complexity and spectral preservation (Fig. 4).

In order to resample the data to the desired number of spectral features, cubic-spline interpolation was applied. It is important to note that cubic-spline interpolation did not reduce the spectral noise nor aided in spectral smoothing, but only served to standardize the input shapes for the RSPTE model.

Additionally, the range of spectral intensities also varied between datasets. For example, the Adenine dataset had spectral intensities ranging from 0 to about 60000; but the Breast Cancer dataset had spectral intensities from 0 to 0.175. These ranges were not only significantly different for each dataset, but were also not the ideal range of values for pretraining. Deep learning models often train best when the input data is scaled from 0 to 1 because it scales the gradient updates, dependent on the input values, to a manageable and stable form that improves training quality [33].

In order to standardize the intensities, min-max normalization was employed to standardize the dataset intensity range from 0 to 1. See Fig. 5 for a visual representation.

This study intentionally did not align wavelengths across datasets or include wavelength as an explicit feature. Our pretraining objective was to learn axis-robust local spectral primitives (e.g., peak shape, width, contrast, and co-occurrence) rather than dataset-specific absolute anchors, because instruments often differ in coverage, sampling density, and small calibration offsets. Architecturally, the 1D CNN backbone with pooling (utilized in both the current ResNet architecture and many other use-cases) is locally shift-equivariant, which biases it to emphasize such local patterns even under small global shifts of the wavenumber axis. Biochemically, Raman identities are encoded not only in absolute positions but also in relative spacings, intensity ratios, and bandshape, which are quantities that remain stable under approximately uniform axis shifts arising from instrument/environmental factors. In contrast, enforcing a single global range across heterogeneous sources would require trimming/extrapolation and substantial padding, creating easy-to-learn artifacts that could reduce the quality of embeddings, dominate self-supervised objectives, and overfit

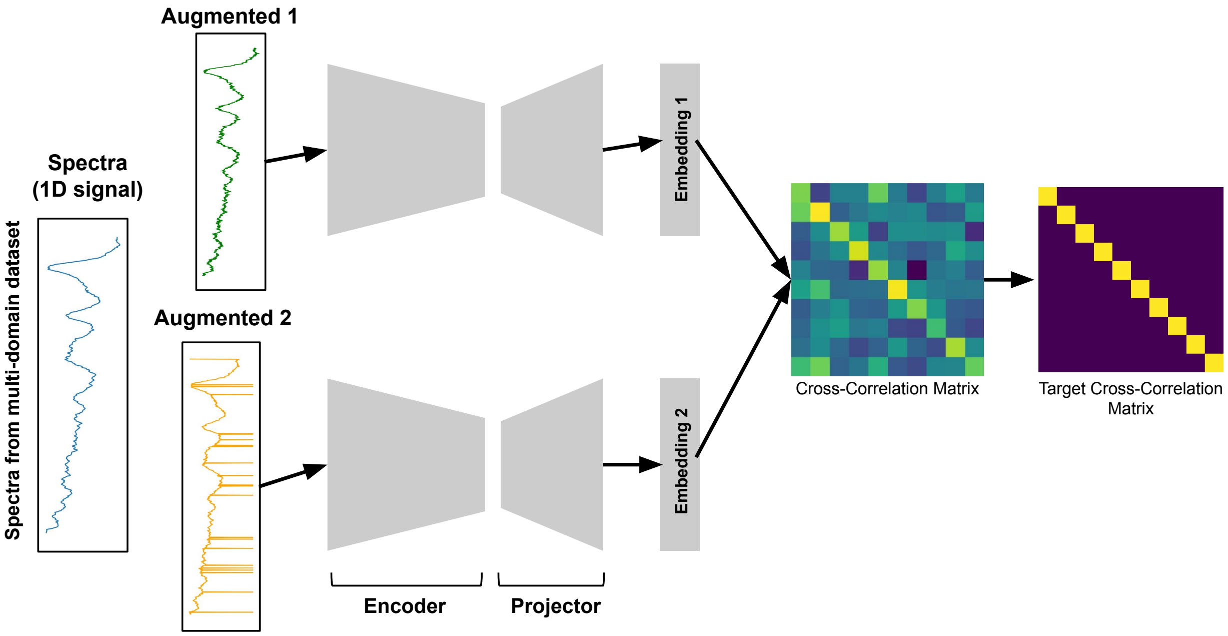

FIGURE 3. Visual demonstration of the Barlow Twins pipeline. The input spectra, represented as a 1D signal, is augmented into two distinct versions (two augmented samples for each real sample), which are both passed through the model containing an encoder and a projector. Two embeddings result from this process. A cross-correlation matrix using the two embeddings is derived and optimized to look like the target through the Barlow Twins loss function.

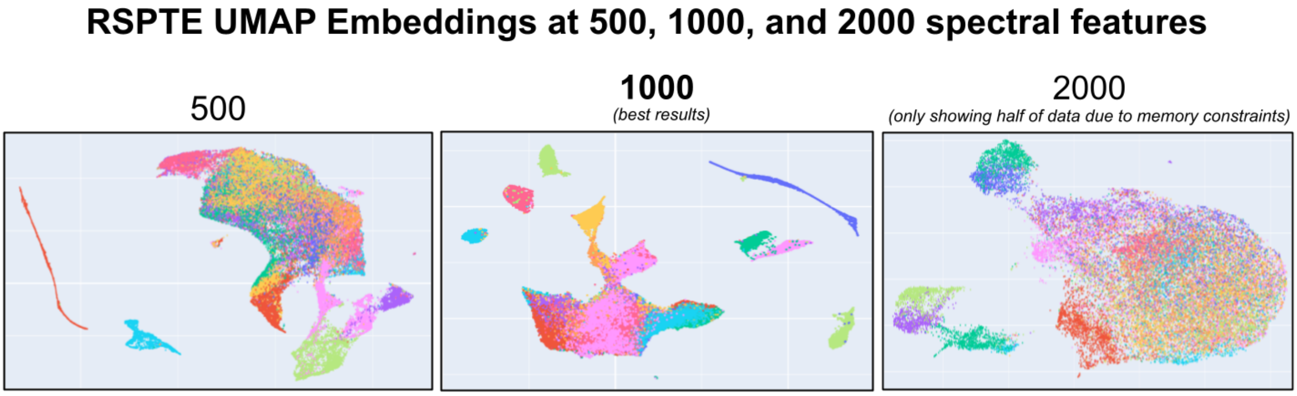

FIGURE 4. UMAP visualization of RSPTE embeddings when pre-training with data interpolated to 5000, 1000, and 2000 spectral features. The 1000 spectral feature model has the most separable clusters. 500 samples returns similar (albeit slightly inferior) results. 2000 samples demonstrates a clear degradation in embedding quality as clusters start to mix

the model to not focus on edges. We therefore restricted preprocessing to resampling for the purpose of length standardization (without smoothing), letting the encoder learn chemistry-relevant local structure that transfers across instruments. Crucially, absolute wavenumber information is reintroduced during fine-tuning on each dataset's native axis (i.e all fine-tuning samples are on the same axis), which "re-anchors" the learned primitives to dataset-specific peak positions when making downstream decisions. Experimentally, the study's choice for handling heterogeneity is supported by the prior input-shape analysis demonstrating a justification for 1000 spectral features, and a cluster-analysis of the UMAP embeddings demonstrating improved separation on fine-tuned data in a different spectral range.

The RSPTE model was created with these specifications:

- The encoder Resnet-34 model had an input shape of size (1000,), corresponding to the decided number of spectral features per spectra.
- The projector network projected intermediate model representations into embeddings of size (2048,). The value 2048 was chosen after reviewing Zbontar et al.'s ablation studies and training setup, in which an embed-

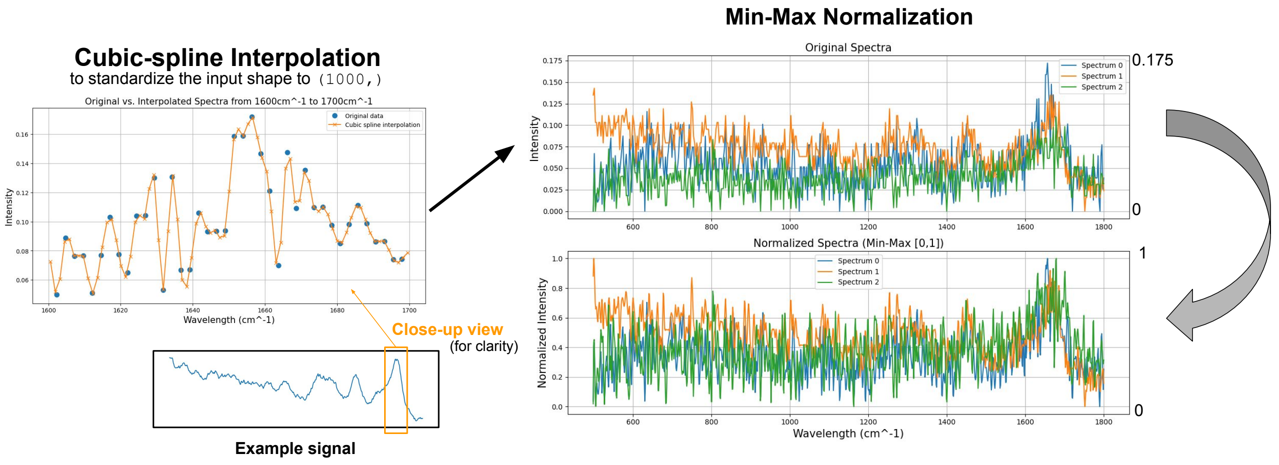

FIGURE 5. Basic preprocessing and normalization steps used to pre-process spectra for pre-training. For each spectrum, a cubic-spline resampling to 1000 spectral features, followed by min-max normalization from 0-1 is applied. .

ding size of 2048 was decided.

- The batch size for training was 1024, after reviewing the Zbontar et al.'s ablation study on batch size, where optimal performance was discovered when using a batch size of 1024 samples.
- The AdamW optimizer was used for gradient descent due to demonstrated improvements over the conventional Adam optimizer in terms of proper generalization and minimizing overfitting [34], and its default parameters for training were utilized.
- RSPTE was pre-trained for 200 epochs. Although the initial plan was to train for 300 epochs following the setup in Zbontar et al., signs of overfitting and potential representation collapse (as shown later in Fig. 6, Results section) were observed. Significantly better results were achieved by reducing the epoch count to 200.
- Basic evaluation of the training results was conducted by analyzing UMAP embeddings [35] of the trained representations. The **Bacteria-ID** reference dataset was selected for evaluation due to its large data count, which facilitated visual identification of clustering. The UMAP algorithm reduced the 2048-feature representations to 2 features while largely preserving the data distribution and enabling visual analysis. (see Fig. 6 in Results section).

Upon completion, the backbone of the pre-trained encoder was saved for subsequent fine-tuning.

D. FINE-TUNING

For fine-tuning, the study utilized the datasets described in Section II-A: **Bacteria-ID**, **Covid-19**, **Melanoma**, and **Wheat Lines**.

In order to use the RSPTE backbone to fine-tune for the datasets, the following setup was required:

1) Load the pre-trained encoder
2) Attach a Dense layer at the end of the pre-trained encoder with N neurons, N being the number of classes the dataset has.

For example, with the **Bacteria-ID** dataset, the pre-trained backbone was loaded, and a Dense layer with 30 neurons was attached to represent the 30 bacterial classes for classification in the downstream task.

To assess RSPTE's effectiveness, fine-tuning was performed using both the RSPTE backbone and a randomly initialized backbone as a control, with both settings trained under identical conditions.

Training was conducted for 400 epochs, determined after analyzing the behavior of models trained using the randomly-initialized backbone, demonstrated below in Fig. 11 at Section III

The batch size of 256 was determined after considering the theoretical effects of batch size [36] on performance for both pre-trained and randomly-initialized models. If the batch size is too small, pre-trained models could have difficulty with generalization due to excessive noise in gradient updates triggered by random variations within the dataset, which may diminish the knowledge obtained during the pre-training step. However, an excessive batch size would make convergence for the randomly-initialized models slower and slightly more prone to over-fitting. While a batch size of 256 seemed like a reasonable compromise and demonstrated accurate results for both pre-trained and randomly-initialized models in certain scenarios, there were several areas where the batch size 256 was greater than the size of the dataset itself, which could make the model at risk of overfitting. To maintain a fair, identical fine-tuning protocol across datasets and splits, we kept the nominal batch size fixed at 256; when the training partition contained fewer than 256 examples, the dataloader formed a single batch of size N (i.e., an effective batch size $\min(256, N)$) per epoch. We did not duplicate samples to fill batches and did not use gradient accumulation in these cases, so the optimizer performed one full-batch update per epoch on the available training examples.

Moderate L1 and L2 regularization were applied in order to help mitigate these effects. Both the L1 and L2 regularization parameters λ_1 and λ_2 were set to 0.05, which helped to both reduce overfitting and improve exploration by causing random perturbations in the model that help it overcome local minima. This regularization is particularly important in the low-N regime induced by $\min(256, N)$ effective batches, where full-batch updates can otherwise overfit. Keeping the batch policy fixed while relying on regularization preserves comparability between the pre-trained and randomly initialized backbones without per-dataset hyperparameter tuning.

To promote maximum generalization, `ReduceLROn-Plateau` was used to decrease the learning rate by a factor of 0.1 if validation loss did not significantly improve for 30 consecutive epochs. The learning rate was reduced until reaching a minimum of 5×10^{-7}, at which point further gradient updates had minimal impact on the loss. This helped prevent the common issue of the loss "bouncing" between two suboptimal extreme conditions instead of reaching the minimum between them–a problem that occurs when the learning rate is too high for the optimizer to have the granularity to reach the minimum.

After fine-tuning, the model was evaluated on the test dataset. For the evaluation, the model was tested on either an independent set provided by the creators of the evaluation benchmark or a partition of the original dataset. The test dataset accuracy was utilized as a metric for performance.

E. ABLATION STUDY

This study evaluated the performance of both the control and pre-trained models fine-tuned with varying amounts of data. Stratified sampling was used to select a percentage of the fine-tuning dataset, allowing performance comparisons across different data proportions. Evaluations were conducted on 10%, 20%, 50%, and 100% of the fine-tuning datasets, except for the **Wheat Lines** dataset. Given its approximately 50,000 samples, smaller percentages (0.1%, 0.2%, 0.5%, 1%, and 10%) were tested to better simulate data-scarce scenarios, where RSPTE was expected to demonstrate its benefits.

To account for variations introduced by data sampling, Stratified K-Fold Cross-Validation with 5 folds was employed, a commonly used approach in prior studies [4, 19]. Four folds were used for training and one for validation, ensuring balanced representation of classes across folds. The fine-tuned model was then evaluated on the independent test set to measure performance.

III. RESULTS

The loss decreased over time during the pre-training process despite minor jumps in the beginning. UMAP projections of the **Bacteria-ID** reference dataset embeddings, as shown in Fig. 6, were created to evaluate the performance of the pre-trained encoder and show signs of unsupervised separation of different spectral types.

In the ablation study, significant improvements in test accuracy (Table 2), precision (Table 3), recall (Table 4), and f1 (Table 5) were observed after fine-tuning with the RSPTE backbone compared to the randomly-initialized control (which mostly failed to generalize under low percentages of data). Additionally, a review of common and SOTA models in current literature was conducted to provide context for the results and what a theoretical maximum could look like. Generally, RSPTE demonstrated comparable results across benchmarks, achieving similar accuracies to established approaches, even with limited context or data. Particularly interesting is RSPTE's performance on the Bacteria-ID dataset, where RSPTE achieved results similar to CNN and SOTA approaches like RamanNet without pretraining on the 60,000 sample reference dataset.

TABLE 2. Accuracy (%) with and without RSPTE. Mean ± std across folds.

Category	Concentration	RSPTE	Random
Bacteria-ID	10%	61.2 ± 1.4	3.3 ± 0.3
	20%	73.6 ± 0.3	36.6 ± 2.8
	50%	78.9 ± 1.0	52.4 ± 1.6
	75%	80.7 ± 0.5	74.0 ± 1.2
	90%	80.2 ± 0.6	63.9 ± 3.1
	100%	81.0 ± 0.6	68.1 ± 5.2
Covid-19	10%	61.3 ± 9.6	49.7 ± 1.6
	20%	75.5 ± 2.6	51.6 ± 0.0
	50%	85.2 ± 4.8	50.3 ± 1.6
	75%	83.9 ± 2.9	51.0 ± 1.3
	90%	85.2 ± 6.0	51.0 ± 1.3
	100%	85.8 ± 1.6	51.0 ± 1.3
Melanoma	10%	40.0 ± 3.5	11.6 ± 3.5
	20%	50.9 ± 4.5	8.8 ± 1.3
	50%	72.8 ± 0.8	13.4 ± 2.9
	75%	87.8 ± 2.9	85.9 ± 2.2
	90%	88.8 ± 1.5	91.3 ± 2.3
	100%	87.5 ± 2.4	94.7 ± 1.3
Wheat Lines	0.10%	82.1 ± 1.1	52.3 ± 0.0
	0.20%	85.5 ± 1.4	51.4 ± 1.9
	0.50%	89.8 ± 0.8	54.8 ± 9.7
	1%	93.2 ± 0.3	92.0 ± 1.7
	5%	96.7 ± 0.1	96.6 ± 0.3
	7%	97.0 ± 0.2	97.1 ± 0.3
	10%	97.3 ± 0.1	97.2 ± 0.2

Fig. 7-Fig 10 present confusion matrices that visually summarize the classification performance across the evaluation datasets. These figures highlight various scenarios encountered during the evaluation process.

An important benchmark to note in which suboptimal results were found for both RSPTE and the randomly-initialized backbone is the **Covid-19** dataset. At 10%, both achieve poor results, with 61% and 50%, respectively. Accuracy of 50% is considered very low, because in a 2-

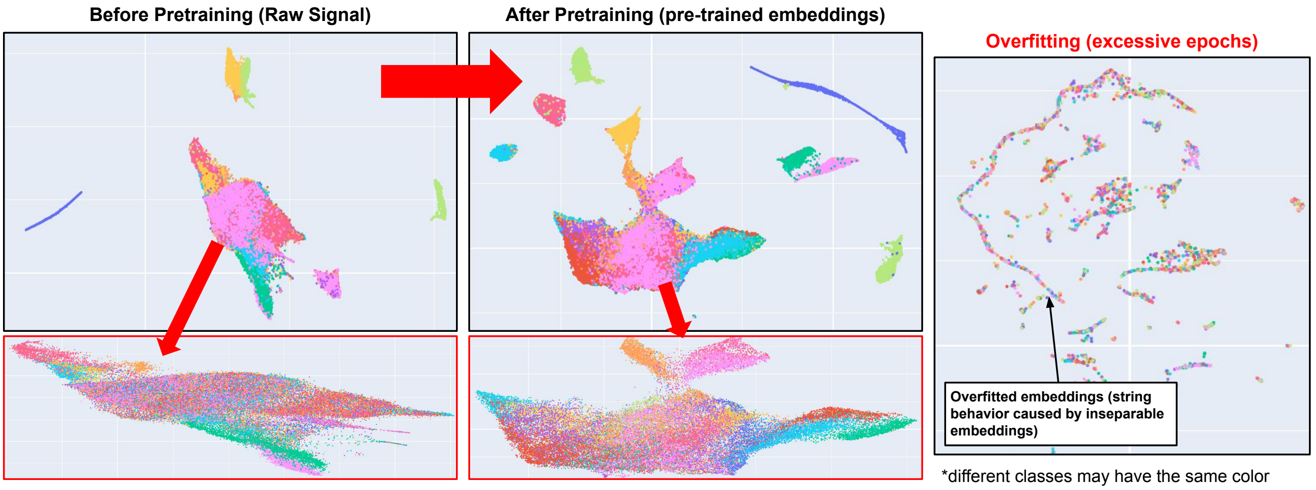

FIGURE 6. UMAP projections of unsupervised **Bacteria-ID** reference dataset embeddings from the pre-trained backbone compared to projections of the raw spectra. Additionally, the effects of overfitting on embedding quality is considered as well. The overfitted embeddings form 'strings'. This behavior often occurs during representation collapse, which occurs when embeddings hold little-to-no useful information.

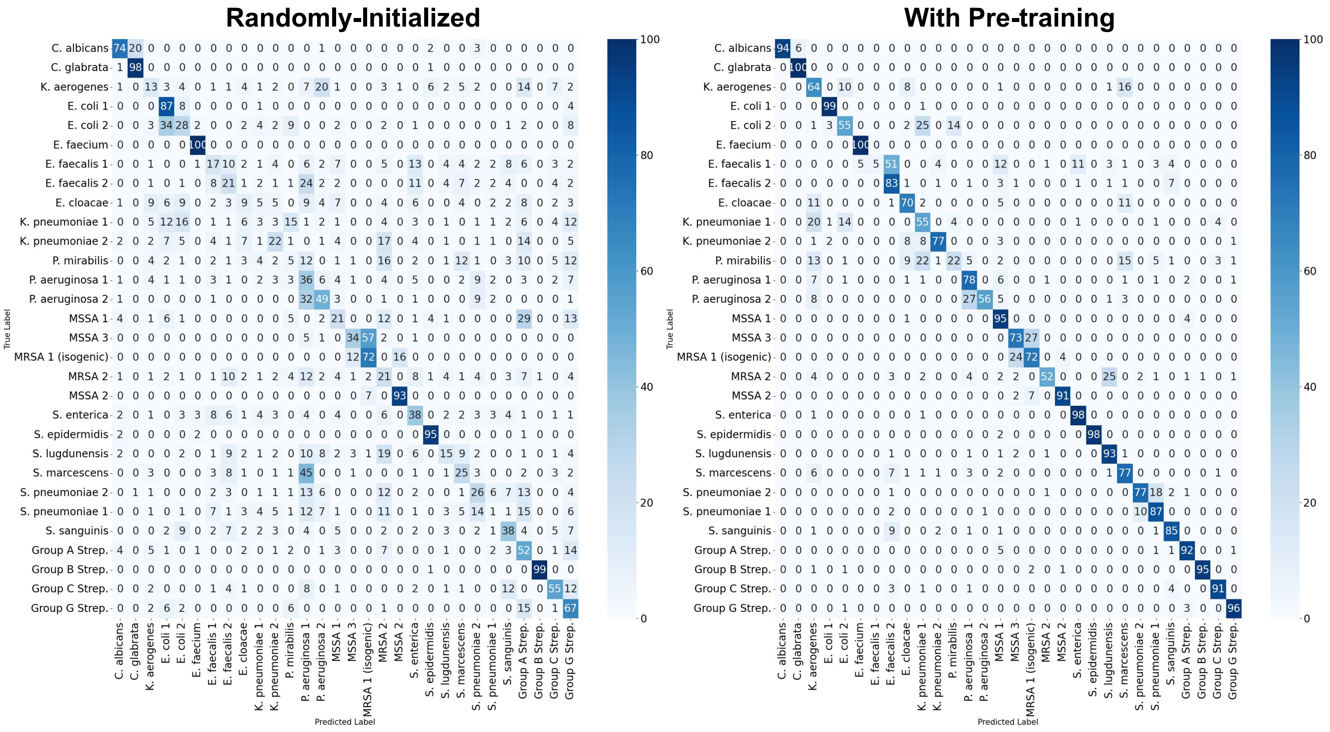

FIGURE 7. Confusion matrix of the **Bacteria-ID** dataset results with 50% of the data

class problem a model that guesses randomly would get an accuracy of 50%. Also suboptimal were the results at 100% of data, in which the randomly-initialized backbone failed to generalize with 51%. This suggests that the setup was suboptimal for both approaches.

A particularly compelling reason may be the over-complexity of the model on a 2-class problem. Given the dataset size and only 2 classes, utilizing a foundation model of this size may not be beneficial as it may not generalize well due to the number of weights being too much for a simple task. Additionally, the high regularization, meant to curb overfitting on multi-class complex datasets, could have prevented the model from reaching a global minimum. For smaller-class problems, utilizing a simpler approach,

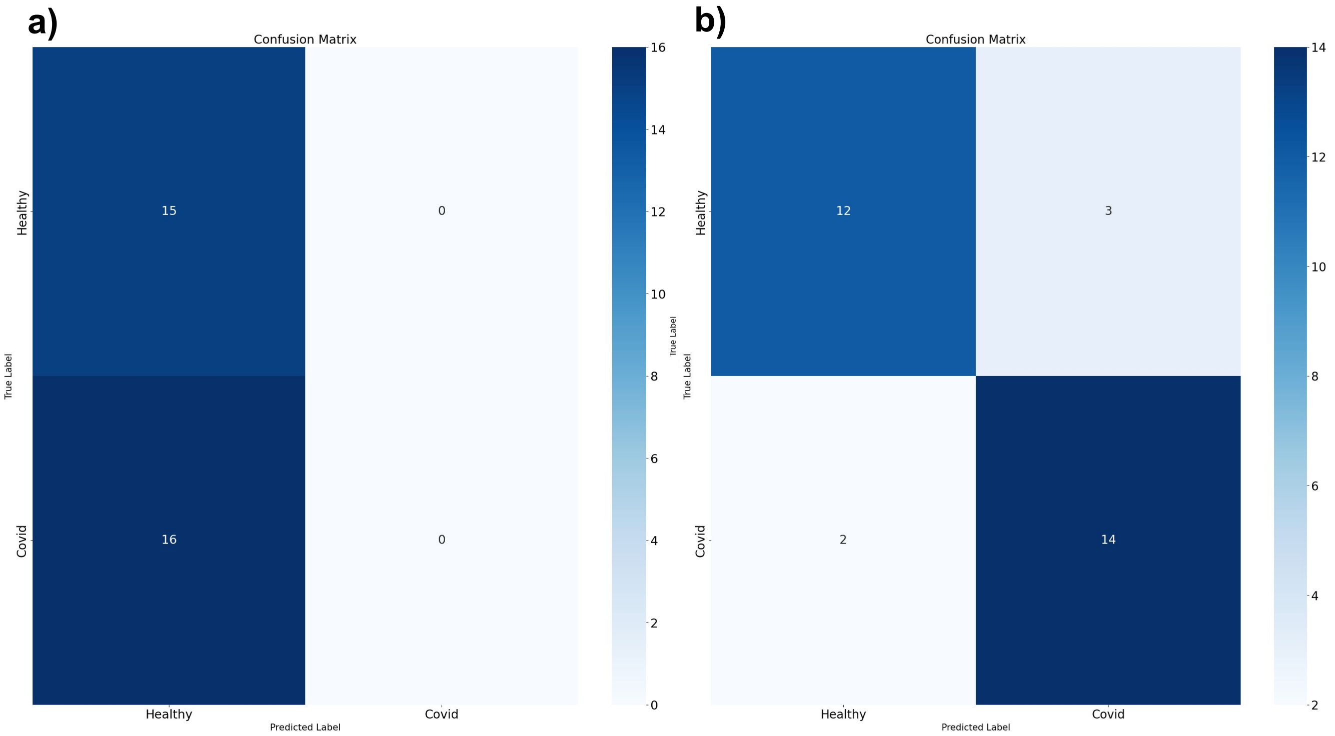

FIGURE 8. Confusion matrix of the **Covid-19** dataset results with 100% of the data; (a) Randomly-Initialized and (b) With Pre-training

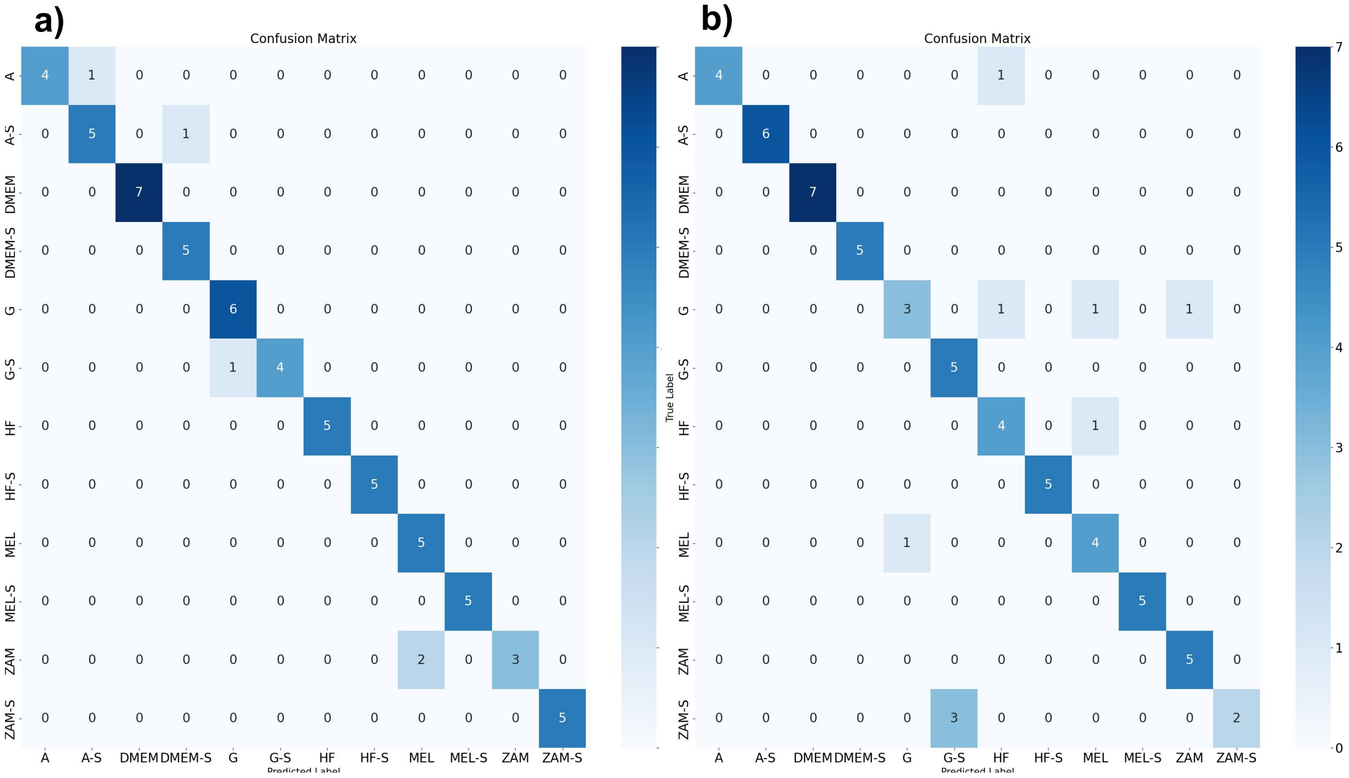

FIGURE 9. Confusion matrix of the **Melanoma** dataset results with 100% of the data; (a) Randomly-Initialized and (b) With Pre-training

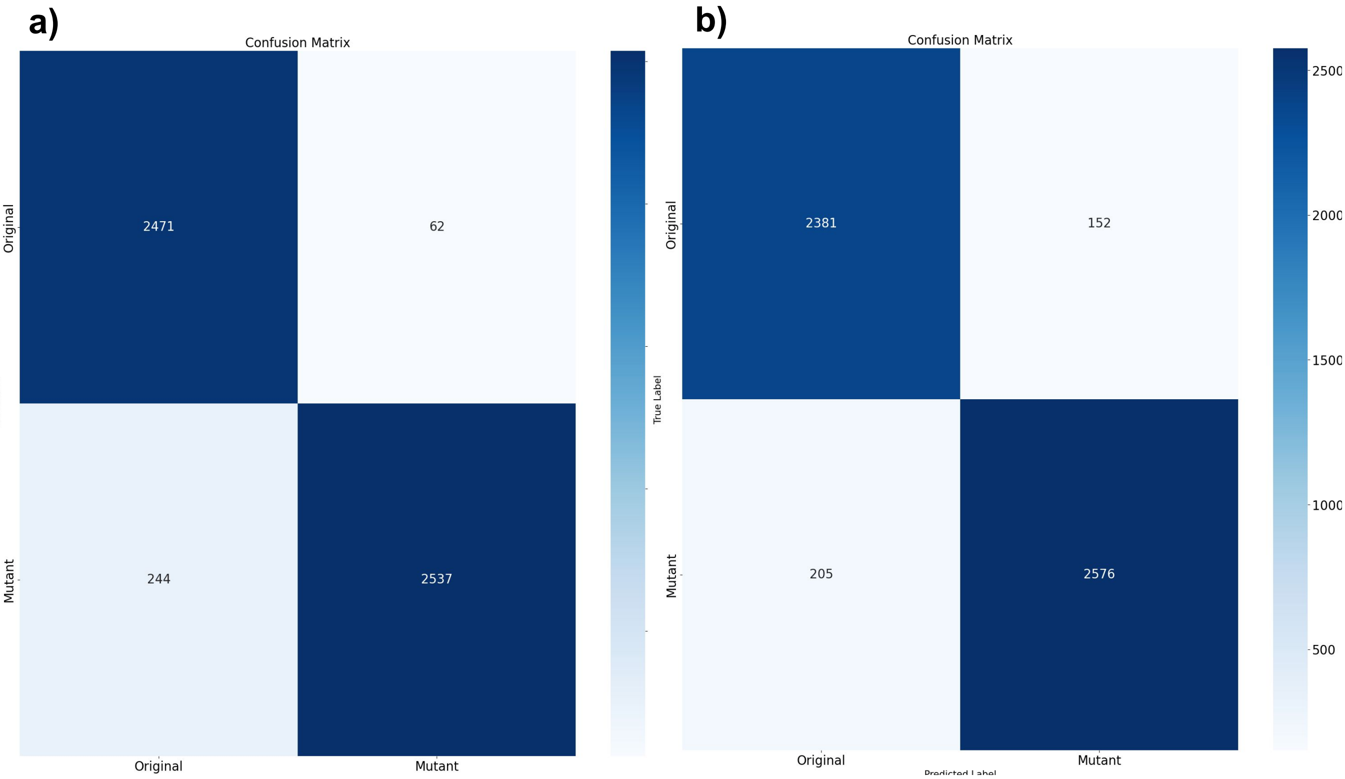

FIGURE 10. Confusion matrix of the **Wheat Lines** dataset results with 1% of the data; (a) Randomly-Initialized and (b) With Pre-training

like PCA for dimensionality reduction, may provide better results.

Furthermore, the accuracy gain when using the RSPTE pre-trained backbone generally decreased as the amount of fine-tuning data increased. In both the **Melanoma** and **Wheat Lines** benchmarks, performance was comparable to the control when trained on 100% of the data. However, in the **Bacteria-ID** benchmark, RSPTE still provided a notable improvement of nearly 10 percentage points, even with the full dataset.

Additionally, even for the cases when RSPTE and the Randomly-initialized backbone exhibited similar results in accuracy, RSPTE often generalized significantly faster than the randomly-initialized counterpart. As it shown in Fig. 11, corresponding to the training process on the **Melanoma** dataset, the randomly-initialized model required close to 250 epochs to finally start to generalize, as opposed to an almost instant convergence with RSPTE which only took approximately 10-30 epochs to reach comparable accuracy levels. Converging in fewer epochs is crucial because it reduces the computational time and resources needed to train a model efficiently.

An interesting point to make is that for both the pre-trained and randomly-initialized curves, the validation accuracy converged below the train accuracy, which was near 100%. It is likely that because we only included spectra collected with ADT-COOH functionalization because that data was significantly less separable than when using all functionalizations together. However it's probable that doing this meant that there simply wasn't enough information for the model to achieve an accuracy comparable to the training data, which it potentially started to memorize in later epochs.

Finally, interest should be taken to the results of the Bacteria-ID dataset, Beyond 75% of data, the randomly-initialized backbone exhibits a bimodal performance distribution. At both 90% and 100%, 4 out of 5 folds converged to suboptimal solutions (60-68% accuracy), while 1 fold achieved 79% accuracy comparable to RSPTE. This 80% failure rate suggests a fundamental optimization challenge and suggests the loss landscape for randomly-initialized backbones at higher data regimes may contain dominant attractors (bad local minima) that almost-always result in suboptimal results. The high variance (90%: ±3.1%, 100%: ±5.2%) reflects this bimodal distribution. In stark contrast, RSPTE achieves 80-81% accuracy consistently across all folds (±0.6%), demonstrating that pre-trained initialization provides reliable and reproducible access to high-quality solutions regardless of random seed or fold selection.

IV. DISCUSSION

The results demonstrated the effective extraction of useful and transferable features from unlabeled out-of-domain data and their successful application to several different Raman spectroscopy domains. This study implemented self-

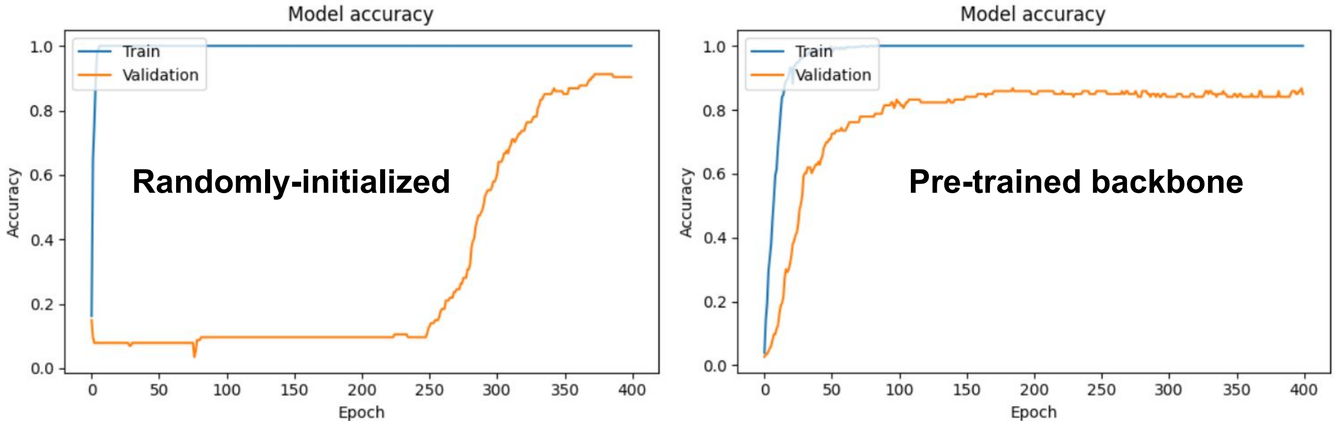

FIGURE 11. Example of accuracy/epoch curves for randomly-initialized and pre-trained backbone on 100% of the **Melanoma** dataset. While both exhibit similar evaluation results, the randomly-initialized backbone took longer to generalize.

TABLE 3. Macro-Precision (%) with and without RSPTE. Mean ± std across folds.

Category	Concentration	RSPTE	Random
Bacteria-ID	10%	64.8 ± 1.4	0.5 ± 0.4
	20%	76.3 ± 0.5	34.9 ± 3.0
	50%	80.7 ± 1.5	51.5 ± 1.7
	75%	82.0 ± 0.6	73.5 ± 1.2
	90%	81.8 ± 0.6	63.6 ± 3.4
	100%	82.1 ± 1.6	68.1 ± 5.6
Covid-19	10%	62.4 ± 10.8	29.9 ± 10.7
	20%	76.3 ± 2.8	25.8 ± 0.0
	50%	85.9 ± 4.4	25.2 ± 0.8
	75%	84.1 ± 2.9	25.7 ± 0.3
	90%	85.4 ± 6.1	25.5 ± 0.6
	100%	86.1 ± 1.8	25.7 ± 0.3
Melanoma	10%	37.6 ± 5.0	1.8 ± 1.3
	20%	53.8 ± 5.4	0.8 ± 0.2
	50%	75.0 ± 3.6	3.2 ± 1.8
	75%	88.8 ± 2.8	89.1 ± 0.8
	90%	89.4 ± 1.7	93.0 ± 2.1
	100%	89.1 ± 2.6	95.3 ± 1.3
Wheat Lines	0.10%	82.7 ± 0.8	26.2 ± 0.0
	0.20%	85.9 ± 1.1	25.7 ± 0.9
	0.50%	89.9 ± 0.8	51.9 ± 23.3
	1%	93.2 ± 0.3	92.5 ± 1.4
	5%	96.7 ± 0.1	96.6 ± 0.3
	7%	97.0 ± 0.2	97.0 ± 0.3
	10%	97.3 ± 0.1	97.2 ± 0.2

TABLE 4. Macro-Recall (%) with and without RSPTE. Mean ± std across folds.

Category	Concentration	RSPTE	Random
Bacteria-ID	10%	61.2 ± 1.4	3.3 ± 0.3
	20%	73.6 ± 0.3	36.6 ± 2.8
	50%	78.9 ± 1.0	52.4 ± 1.6
	75%	80.7 ± 0.5	74.0 ± 1.2
	90%	80.2 ± 0.6	63.9 ± 3.1
	100%	81.0 ± 0.6	68.1 ± 5.2
Covid-19	10%	61.5 ± 9.8	50.3 ± 0.5
	20%	75.4 ± 2.5	50.0 ± 0.0
	50%	85.2 ± 4.8	50.0 ± 0.0
	75%	83.9 ± 2.9	49.4 ± 1.3
	90%	85.1 ± 6.0	50.0 ± 0.0
	100%	85.7 ± 1.5	49.4 ± 1.3
Melanoma	10%	38.8 ± 3.3	11.3 ± 3.7
	20%	49.6 ± 5.1	8.3 ± 0.0
	50%	71.2 ± 2.0	11.1 ± 2.4
	75%	87.7 ± 2.9	85.7 ± 2.6
	90%	88.6 ± 1.5	91.0 ± 2.3
	100%	87.7 ± 2.5	94.7 ± 1.2
Wheat Lines	0.10%	82.4 ± 1.0	50.0 ± 0.0
	0.20%	85.7 ± 1.2	50.0 ± 0.0
	0.50%	89.7 ± 0.8	55.8 ± 9.6
	1%	93.2 ± 0.3	92.3 ± 1.6
	5%	96.7 ± 0.1	96.7 ± 0.3
	7%	97.0 ± 0.2	97.1 ± 0.3
	10%	97.3 ± 0.1	97.2 ± 0.2

supervised pre-training using Barlow Twins on a multi-domain pre-training dataset and fine-tuned a ResNet-34+projector model across multiple evaluation sets. Performance was compared between a self-supervised pre-trained backbone and a randomly initialized one, demonstrating that the RSPTE backbone consistently and significantly outperformed the randomly initialized model, even with limited data.

A. LIMITATIONS

Despite the promising results, it is important to acknowledge some limitations of the current study.

1) Hyperparameter Exploration: RSPTE was pre-trained with a fixed batch size of 256. When fine-tuning on small datasets where the training partition contains fewer than 256 samples, this effectively becomes full-batch gradient descent, which may limit the regularization benefits typically provided by stochastic mini-

TABLE 5. Macro-F1 (%) with and without RSPTE. Mean ± std across folds.

Category	Concentration	RSPTE	Random
Bacteria-ID	10%	61.2 ± 1.4	0.7 ± 0.4
	20%	72.7 ± 0.4	32.6 ± 2.8
	50%	77.7 ± 0.9	50.7 ± 1.6
	75%	79.4 ± 0.5	72.5 ± 1.1
	90%	78.8 ± 0.5	62.7 ± 3.1
	100%	79.7 ± 0.7	66.7 ± 4.9
Covid-19	10%	60.9 ± 9.5	36.5 ± 7.2
	20%	75.2 ± 2.5	34.0 ± 0.0
	50%	85.1 ± 4.9	33.5 ± 0.7
	75%	83.9 ± 2.9	33.8 ± 0.6
	90%	85.1 ± 6.0	33.8 ± 0.6
	100%	85.7 ± 1.6	33.8 ± 0.6
Melanoma	10%	35.6 ± 3.7	2.9 ± 2.0
	20%	49.0 ± 4.6	1.5 ± 0.3
	50%	71.3 ± 1.7	4.2 ± 1.7
	75%	87.2 ± 3.2	85.5 ± 2.8
	90%	88.3 ± 1.6	91.1 ± 2.3
	100%	86.8 ± 2.8	94.7 ± 1.2
Wheat Lines	0.10%	82.1 ± 1.1	34.4 ± 0.0
	0.20%	85.5 ± 1.4	33.9 ± 0.8
	0.50%	89.8 ± 0.8	43.9 ± 15.3
	1%	93.2 ± 0.3	92.0 ± 1.8
	5%	96.7 ± 0.1	96.6 ± 0.3
	7%	97.0 ± 0.2	97.0 ± 0.3
	10%	97.3 ± 0.1	97.2 ± 0.2

batching. Although L1 and L2 regularization were applied during fine-tuning, per-dataset hyperparameter tuning of batch size and regularization strength could yield improved performance on smaller datasets.

2) Over-parameterization: The ResNet-34 backbone, while effective for complex multi-class problems such as Bacteria-ID (30 classes) and Melanoma (12 classes), may be over-parameterized for simpler binary classification tasks. This is evident in the Covid-19 results, where RSPTE achieved 85.8% accuracy compared to 51.0% for the randomly initialized baseline. This is a substantial improvement, but simpler methods like PCA or SVM provide equivalent or better performance. The confusion matrices for Covid-19 reveal persistent false positives and false negatives, suggesting that the model's capacity exceeds what is necessary for a binary task with only 278 samples. For such low-complexity tasks, smaller backbones, like ResNet-18, or simply dimensionality reduction followed by classical classifiers, may be more appropriate.
3) Generalization to diverse acquisition conditions: RSPTE was pre-trained and evaluated exclusively on publicly available datasets. Performance on data acquired under substantially different conditions, such as non-standard wavelength ranges or specialized instrumentation may require additional validation.

Some specific adjustments to the current approach that may be valuable for further study include:

- Exploring new augmentations, like applying baseline shifts or mixing several different spectra together.
- Evaluating further preprocessing steps like signal cleaning; stricter cosmic-ray [2] thresholding; polynomial baseline removal, for removing fluorescence shifts; or low-pass filtering, to reduce noise. However, the impact of these artifacts remains uncertain, as they may either improve model robustness or contribute to noise and overfitting, warranting further study.
- Combining UMAP with other interpretable ML techniques to further understand the effect of pre-trained encoders [37, 38].

TABLE 6. Reference accuracies reported in prior work (typically full-data or specified feature subsets), shown for context alongside our RSPTE (100%) results where available. Protocols differ across studies.

Task	Method	Accuracy (%)	Setup note
Bacteria-ID	RSPTE (ours, 100%)	81	no pretraining conducted
	KNN	40	
	CNN	82.2	Pretrained with 60,000 spectra
	Logistic Regression	75.8	
	SVM	74.9	
	PCA + KNN	82	
	RamanNet	85.5	Pretrained with 60,000 spectra
Covid-19	RSPTE (ours, 100%)	87	
	KNN	79	
	PCA + KNN	85	
	RamanNet	91	
	SVM	90	
Melanoma	RSPTE (ours, 100%)	87	−COOH only
	KNN	77	all features
	PCA + KNN	76	all features
	CNN	93.3	−COOH only
	RamanNet	98.54	−COOH only
	RamanNet	99	all features
Wheat Lines	RSPTE (ours, 10%)	97	100% not available
	Tree Classifier	94.62	

Finally, future research could explore combining the RSPTE pre-training approach with innovative model architectures and novel techniques. Since the Barlow Twins architecture allows for a swappable projector network, integrating novel projectors like Transformers [19], Action Networks [39], or Quanvolutional Neural Networks [40], which have either shown demonstrated improvements on benchmarks or promising results in the field, would be an interesting direction. Additionally, synthetic data generation to further augment the pre-training data, as has been explored through approaches like Generative Adversarial Networks [41], could further improve the resiliency of the pre-trained encoder.

V. CONCLUSION

The results obtained with RSPTE and presented in this paper are promising. The proposed approach demonstrated the ability to achieve heuristic results using only a fraction of in-domain data. In a research setting, this approach could enable faster, earlier, and more cost-effective results by reducing the need for large quantities of manually collected and labeled in-domain data, potentially streamlining developments in the Raman Spectroscopy field and making analysis more accessible to resource-limited settings.

STATEMENTS AND DECLARATIONS

CONFLICT OF INTEREST/COMPETING INTERESTS

The authors declare no conflict of interest with respect to the contents of this paper.

DATA AVAILABILITY

The MDA-MB-231 Breast Cancer dataset used for self-supervised pre-training can be found at DeepeR. The **Bacteria-ID** dataset used for fine-tuning can be found at: Bacteria-ID. Links to other datasets utilized for pre-training can be found at Datasets.

CODE AVAILABILITY

Processed data and code to reproduce the results are available at https://github.com/dewball345/RamanFoundation.

FUNDING

The authors received no funding for the work described in this paper.

REFERENCES

[1] A. Sen, I. Kecoglu, M. Ahmed, U. Parlatan, and M. B. Unlu, “Differentiation of advanced generation mutant wheat lines: Conventional techniques versus raman spectroscopy,” Frontiers in Plant Science, vol. 14, p. 1116876, 2023.

[2] G. Berlanga, Q. Williams, and N. Temiquel, “Convolutional neural networks as a tool for raman spectral mineral classification under low signal, dusty mars conditions,” Earth and Space Science, vol. 9, no. 10, p. e2021EA002125, 2022.

[3] S. Fornasaro, F. Alsamad, M. Baia, L. A. Batista de Carvalho, C. Beleites, H. J. Byrne, A. Chiadò, M. Chis, M. Chisanga, A. Daniel et al., “Surface enhanced raman spectroscopy for quantitative analysis: results of a large-scale european multi-instrument interlaboratory study,” Analytical chemistry, vol. 92, no. 5, pp. 4053–4064, 2020.

[4] C.-S. Ho, N. Jean, C. A. Hogan, L. Blackmon, S. S. Jeffrey, M. Holodniy, N. Banaei, A. A. Saleh, S. Ermon, and J. Dionne, “Rapid identification of pathogenic bacteria using raman spectroscopy and deep learning,” Nature communications, vol. 10, no. 1, pp. 1–8, 2019.

[5] Y. Liu, Y. Gao, R. Niu, Z. Zhang, G.-W. Lu, H. Hu, T. Liu, and Z. Cheng, “Rapid and accurate bacteria identification through deep-learning-based two-dimensional raman spectroscopy,” Analytica Chimica Acta, vol. 1332, p. 343376, 2024.

[6] R. Li, D. Dhankhar, J. Chen, A. Krishnamoorthi, T. C. Cesario, and P. M. Rentzepis, “Identification of live and dead bacteria: A raman spectroscopic study,” IEEE Access, vol. 7, pp. 23 549–23 559, 2019.

[7] X. Qiu, X. Wu, X. Fang, Q. Fu, P. Wang, X. Wang, S. Li, and Y. Li, “Raman spectroscopy combined with deep learning for rapid detection of melanoma at the single cell level,” Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy, vol. 286, p. 122029, 2023.

[8] K. E. Tomnikova and I. A. Matveeva, “Machine learning methods for classifying raman scattering spectra of the skin,” in 2024 X International Conference on Information Technology and Nanotechnology (ITNT). IEEE, 2024, pp. 1–4.

[9] P. M. Conforti, G. Lazzini, P. Russo, and M. D’Acunto, “Raman spectroscopy and ai applications in cancer grading. an overview,” IEEE Access, 2024.

[10] L. Huang, H. Sun, L. Sun, K. Shi, Y. Chen, X. Ren, Y. Ge, D. Jiang, X. Liu, W. Knoll et al., “Rapid, label-free histopathological diagnosis of liver cancer based on raman spectroscopy and deep learning,” Nature Communications, vol. 14, no. 1, p. 48, 2023.

[11] Y. Zhang, Z. Li, Z. Li, H. Wang, D. Regmi, J. Zhang, J. Feng, S. Yao, and J. Xu, “Employing raman spectroscopy and machine learning for the identification of breast cancer,” Biological Procedures Online, vol. 26, no. 1, p. 28, 2024.

[12] A. Nakar, A. Pistiki, O. Ryabchykov, T. Bocklitz, P. Rösch, and J. Popp, “Detection of multi-resistant clinical strains of e. coli with raman spectroscopy,” Analytical and Bioanalytical Chemistry, vol. 414, no. 4, pp. 1481–1492, 2022.

[13] A. Orlando, F. Franceschini, C. Muscas, S. Pidkova, M. Bartoli, M. Rovere, and A. Tagliaferro, “A comprehensive review on raman spectroscopy applications,” Chemosensors, vol. 9, no. 9, p. 262, 2021.

[14] R. Luo, J. Popp, and T. Bocklitz, “Deep learning for raman spectroscopy: a review,” Analytica, vol. 3, no. 3, pp. 287–301, 2022.

[15] S. Weng, H. Yuan, X. Zhang, P. Li, L. Zheng, J. Zhao, and L. Huang, “Deep learning networks for the recognition and quantitation of surface-enhanced raman spectroscopy,” Analyst, vol. 145, no. 14, pp. 4827–4835, 2020.

[16] A. Anjikar, N. P. Rao, R. Paneerselvam, K. Jayaramulu, C. Narayana, T. Yamamoto, and H. Noothalapati, “Deep learning in biomedical applications of raman spectroscopy,” in Biomedical Imaging: Advances in Artificial Intelligence and Machine Learning. Singapore:

Springer, 2024, pp. 209–247.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, and I. Polosukhin, "Attention is all you need," in 31st Conference on Neural Information Processing Systems NIPS 2017, 2017.

[18] J.-B. Cordonnier, A. Loukas, and M. Jaggi, "Multi-head attention: Collaborate instead of concatenate," 2021. [Online]. Available: https://arxiv.org/abs/2006.16362

[19] O. C. Koyun, R. K. Keser, S. O. Sahin, D. Bulut, M. Yorulmaz, V. Yucesoy, and B. U. Toreyin, "Ramanformer: A transformer-based quantification approach for raman mixture components," ACS omega, 2024.

[20] A. Tangborn, "Wavelet transforms in time series analysis," Global Modeling and Assimilation Office, Goddard Space Flight Center: Washington, WA, USA, pp. 1–31, 2010.

[21] Z. Wang and T. Oates, "Imaging time-series to improve classification and imputation," 2015. [Online]. Available: https://arxiv.org/abs/1506.00327

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[23] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., "Learning transferable visual models from natural language supervision," in International conference on machine learning. PMLR, 2021, pp. 8748–8763.

[24] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He et al., "A comprehensive survey on pretrained foundation models: A history from bert to chatgpt," International Journal of Machine Learning and Cybernetics, pp. 1–65, 2024.

[25] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in International conference on machine learning. PMLR, 2021, pp. 12 310–12 320.

[26] C. C. Horgan, M. Jensen, A. Nagelkerke, J.-P. St-Pierre, T. Vercauteren, M. M. Stevens, and M. S. Bergholt, "High-throughput molecular imaging via deep-learning-enabled raman spectroscopy," Analytical chemistry, vol. 93, no. 48, pp. 15 850–15 860, 2021.

[27] B. Lafuente, R. Downs, H. Yang, and N. Stone, "The power of databases: The rruff project," Highlights in Mineralogical Crystallography, pp. 1–30, 11 2015.

[28] G. Yin, L. Li, S. Lu, Y. Yin, Y. Su, Y. Zeng, M. Luo, M. Ma, H. Zhou, L. Orlandini et al., "An efficient primary screening of covid-19 by serum raman spectroscopy," Journal of Raman Spectroscopy, vol. 52, no. 5, pp. 949–958, 2021.

[29] M. Erzina, A. Trelin, O. Guselnikova, B. Dvorankova, K. Strnadova, A. Perminova, P. Ulbrich, D. Mares, V. Jerabek, R. Elashnikov et al., "Precise cancer detection via the combination of functionalized sers surfaces and convolutional neural network with independent inputs," Sensors and Actuators B: Chemical, vol. 308, p. 127660, 2020.

[30] N. Ibtehaz, M. E. Chowdhury, A. Khandakar, S. Kiranyaz, M. S. Rahman, and S. M. Zughaier, "Ramannet: a generalized neural network architecture for raman spectrum analysis," Neural Computing and Applications, vol. 35, no. 25, pp. 18 719–18 735, 2023.

[31] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in International conference on machine learning. PMLR, 2020, pp. 1597–1607.

[32] Yashowardhan, "How to code your resnet from scratch in tensorflow?" https://www.analyticsvidhya.com/blog/2021/08/how-to-code-your-resnet-from-scratch-in-tensorflow/, Oct 2024.

[33] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "Normalization techniques in training dnns: Methodology, analysis and application," IEEE transactions on pattern analysis and machine intelligence, vol. 45, no. 8, pp. 10 173–10 196, 2023.

[34] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," CoRR, vol. abs/1711.05101, 2017. [Online]. Available: http://arxiv.org/abs/1711.05101

[35] L. McInnes, J. Healy, N. Saul, and L. Großberger, "Umap: Uniform manifold approximation and projection," Journal of Open Source Software, vol. 3, no. 29, p. 861, 2018. [Online]. Available: https://doi.org/10.21105/joss.00861

[36] A. Pareja, N. S. Nayak, H. Wang, K. Killamsetty, S. Sudalairaj, W. Zhao, S. Han, A. Bhandwaldar, G. Xu, K. Xu, L. Han, L. Inglis, and A. Srivastava, "Unveiling the secret recipe: A guide for supervised fine-tuning small llms," 2024. [Online]. Available: https://arxiv.org/abs/2412.13337

[37] P. Jin, Y.-T. Yeh, J. Ye, Z. Wang, Y. Xue, N. Zhang, S. Huang, E. Ghedin, H. Lu, A. Schmitt et al., "Strain-level identification and analysis of avian coronavirus using raman spectroscopy and interpretable machine learning," in 2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI). IEEE, 2023, pp. 1–5.

[38] I. A. Matveeva, "Research of the feature informativeness in the classification of raman spectra for the identification of malignant melanoma of the skin," in 2024 8th International Conference on Information, Control, and Communication Technologies (ICCT). IEEE, 2024, pp. 1–4.

[39] J. I. Da Silva Filho, C. Vander Nunes, D. V. Garcia, M. C. Mario, F. Giordano, J. M. Abe, M. T. T. Pacheco, and L. Silveira Jr, "Paraconsistent analysis network applied in the treatment of raman spectroscopy data to support medical diagnosis of skin cancer," Medical & biological engineering & computing, vol. 54, no. 10, pp. 1453–1467, 2016.

[40] S. S. Dutta, S. Sandeep, S. Sridevi, R. Mistry, J. Or-

tega, and G. Kar, "Novel quanvolutional neural network based covid-19 diagnosis from raman spectroscopic signals of human serum," IEEE Access, 2025.

[41] C. McDermott, S. Lovett, and C. Rossa, "Improved bioimpedance spectroscopy tissue classification through data augmentation from generative adversarial networks," Medical & Biological Engineering & Computing, vol. 62, no. 4, pp. 1177–1189, 2024.

ABHIRAAM ERANTI is a freshman at UC Berkeley studying data science, economics, and biomedical engineering. He worked at Chronus Health for 1.5 years creating ML diagnostic software and microfluidic monitoring algorithms for CBC 3-part differential and CMP testing and is a co-inventor on the company's provisional patent application for his contributions. He also has over 4 years of experience as a subcontractor for the US Bureau of Engraving and Printing. He developed EyeNote v4 (eyenote.gov), a mobile app that helps visually-impaired individuals read currency denominations via low-latency deep-learning architectures. He is currently based in the Bay Area establishing accessible support channels for EyeNote and is exploring research and applied opportunities in data science and lab-on-a-chip diagnostics.

YOGESH TEWARI was born in India. He received a B.Tech. degree in Electronics and Communication Engineering from AKGEC, India, in 2008, and an M.S. degree in Artificial Intelligence from Johns Hopkins University in 2022. In his early career, he worked as a Software Engineer in Sydney, Australia, where he used graph databases to help the largest credit reporting agency in Australia and New Zealand develop forensic-grade credit analysis systems. After joining Google in 2018, as a key advisor to The Broad Institute of MIT and Harvard, he helped them design and build one of the largest genomics data warehouse solutions to perform joint calling on millions of human genomes for the All of Us Research Program, optimizing query costs by 40x. As a Cloud Data Engineer at Google, he has led massive-scale migrations for large enterprises, moving petabytes of data to Google Cloud. He has also directly contributed to Google's product development by implementing core SQL features for BigQuery Graph. His current focus is on Graph Neural Networks (GNNs) and their application in achieving Level 5 autonomy for large Telecom service providers, known as Autonomous Network Operations (ANO).

RAFAEL PALACIOS was born in Madrid, Spain, in 1966. He received the B.S. and M.S. degrees in mechanical engineering from the ICAI School of Engineering, Comillas Pontifical University, Madrid, in 1990, and the Ph.D. degree from Comillas Pontifical University, in 1998. He joined the Department of Electronics, ICAI School of Engineering, as an Assistant Professor, and the Institute for Research in Technology, as a Researcher, in 1998. He obtained Tenure, in 2004, and became a Full Professor, in 2020. He has been the Head of the Programs in telecommunications engineering and computer science since 2012. He also helped to create the master's program in cybersecurity and was the coordinator from 2019 to 2021. He is a frequent Visiting Professor with the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, where he carried out research with Sloan School of Management (2001-02 and 2024-25), the Department of Aeronautics and Astronautics (2009-10), and the MIT Energy Initiative (2017-18).

DR. AMAR GUPTA rejoined MIT in 2015 and leads and conducts research activities at MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL). Concurrently, he also serves as Distinguished Professor, University of Texas at El Paso. Earlier, he worked at the MIT Sloan School and then as Thomas R. Brown Endowed Professor of Management and Technology, and tenured Professor of Entrepreneurship at University of Arizona. For parts of this period, he also served as Senior Director for Research and Business Development, Professor of Computer Science, Professor of Latin American Studies, Professor of Law, and Professor of Pharmacy. He was selected as The Best Teacher in 2011 by an unprecedented margin and subsequently by students who took his new course, "Telemedicine and Telehealth for Global Health" at MIT. Between 2012 and 2015, he served as Dean of Seidenberg School of Computer Science and Information Science. He has served as an advisor to multiple agencies of the United Nations and has published 12 books. He has played a pivotal role in nucleating several technologies that are in broad use today, including AI-based Automated Reading and Processing of Bank Cheques (which has led to global savings of over one trillion dollars), clipart, PC-based Presentation Graphics and the first microcomputer-based image database system. His current focus is on: Artificial Intelligence; 24-Hour Knowledge Factory; Telemedicine; and Global Health. Dr. Gupta holds an undergraduate degree in Electrical Engineering, a graduate degree in Management (from MIT) and a doctorate in Computer Science. He is an elected Life Fellow of IEEE.

.

APPENDIX A SOFTWARE

Training was completed on an NVIDIA A100 GPU using Google Colab Pro. The model was trained using Keras. Scikit-learn was used for the fine-tuning data preparation and K-fold cross-validation.

APPENDIX B SAMPLES IN FINE-TUNING DATA

This appendix details the number of samples employed in the training, validation, and test sets for all fine-tuning datasets, including the per–sampling-level breakdown used in our experiments. Unless otherwise noted, validation is a 20% split of the sampled training data (i.e., 80% training / 20% validation). For sampling levels that yield non-integer counts, we round to the nearest integer and allocate validation as $N_{\text{val}} = N_{\text{sample}} - \lfloor 0.8 \cdot N_{\text{sample}} \rfloor$ (so that train + val equals the sampled total).

BACTERIA

Total (train+val pool): 3000 **Test:** 3000
Sampling levels (10, 20, 50, 100% of 3000).

Sampling	Sampled Total	Train (80%)	Val (20%)	Test
10%	300	240	60	3000
20%	600	480	120	3000
50%	1500	1200	300	3000
100%	3000	2400	600	3000

MELANOMA

Total (train+val pool): 569 **Test:** 64
Sampling levels (10, 20, 50, 100% of 569).

Sampling	Total	Train (80%)	Val (20%)	Test
10%	57	45	12	64
20%	114	91	23	64
50%	284	227	57	64
100%	569	455	114	64

WHEAT LINES

Total (train+val pool): 47,820 **Test:** 5,314
Sampling levels (0.1, 0.2, 0.5, 1, 10% of 47,820).

Sampling	Sampled Total	Train (80%)	Val (20%)	Test
0.1%	48	38	10	5314
0.2%	96	76	20	5314
0.5%	239	191	48	5314
1%	478	382	96	5314
10%	4782	3825	957	5314

COVID-19

Total (train+val pool): 278 **Test:** 31
Sampling levels (10, 20, 50, 100% of 278).

Sampling	Sampled Total	Train (80%)	Val (20%)	Test
10%	28	22	6	31
20%	56	44	12	31
50%	139	111	28	31
100%	278	222	56	31

APPENDIX C PER-CLASS RESULTS

This appendix reports per-class accuracy (i.e., class-wise recall, %) as mean ± std across folds. Multi-class datasets (Bacteria-ID, Melanoma) include all evaluated concentrations; binary datasets (COVID-19, Wheat Lines) show class 0/1 recalls per concentration for both RSPTE and Random initializations. R stands for randomly initialized, P stands for pre-trained

TABLE 7. Bacteria-ID (30 classes): Per-class accuracy (recall, %) across concentrations. Mean ± std.

Class	10% R	10% P	20% R	20% P	50% R	50% P	75% R	75% P	90% R	90% P	100% R	100% P
0	6.0±9.1	58.8±3.3	57.2±13.7	81.4±5.1	80.6±3.1	91.4±2.5	91.6±2.9	94.8±1.5	89.4±5.3	92.4±4.9	90.4±5.3	93.8±1.2
1	72.6±32.5	71.8±14.7	95.2±2.8	96.8±2.4	98.8±0.8	100.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0	99.6±0.8	100.0±0.0
2	0.0±0.0	51.6±12.1	3.8±2.3	62.2±11.5	16.0±7.6	52.8±14.0	52.8±5.3	68.6±4.0	33.6±10.0	67.8±4.7	44.4±5.3	67.0±5.6
3	1.4±2.8	57.0±9.6	74.2±8.5	87.4±4.5	92.8±1.7	99.4±0.8	99.0±0.9	100.0±0.0	96.2±1.2	99.4±0.5	98.8±0.8	99.8±0.4
4	0.0±0.0	38.0±5.6	24.4±8.8	34.2±8.3	32.4±5.8	38.8±5.3	36.0±5.8	43.2±4.2	43.2±4.3	39.0±8.1	36.0±7.6	51.2±9.2
5	0.8±1.6	84.8±8.2	88.6±7.0	100.0±0.0	99.8±0.4	100.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0
6	0.0±0.0	27.6±11.7	6.2±4.0	7.2±3.7	20.8±10.3	4.2±1.2	5.8±2.8	4.2±1.7	19.0±7.9	4.2±2.6	10.6±3.6	2.4±2.1
7	0.0±0.0	44.6±6.7	21.0±12.5	74.4±2.7	36.8±7.3	85.4±2.7	64.0±8.1	81.8±3.5	51.4±7.0	81.8±4.7	61.6±10.3	86.8±2.1
8	0.0±0.0	44.2±9.1	16.4±10.1	58.0±7.6	19.4±6.5	72.2±2.6	53.4±5.1	72.6±4.2	33.8±7.5	73.2±6.2	42.4±14.8	75.2±3.3
9	0.0±0.0	34.4±2.4	1.4±1.0	45.6±6.2	10.8±3.5	65.8±6.6	34.2±3.5	65.4±5.5	26.8±7.0	73.0±8.4	35.0±19.8	73.4±6.1
10	7.6±15.2	70.4±2.6	17.4±8.1	80.4±6.5	40.8±5.0	85.4±0.8	78.6±5.6	88.2±4.3	63.6±10.0	83.0±2.8	68.4±9.9	84.2±2.8
11	0.0±0.0	50.4±5.5	9.8±4.7	30.6±3.6	8.2±4.3	29.4±4.0	22.0±6.5	27.8±4.8	17.4±7.0	31.2±4.5	17.2±2.6	34.6±5.6
12	0.0±0.0	61.2±6.2	11.6±9.2	62.4±4.8	38.6±4.0	76.2±2.2	80.2±3.0	81.0±2.2	50.6±6.8	87.8±2.7	62.2±15.4	85.0±4.5
13	0.6±1.2	64.6±5.5	39.0±9.2	68.8±4.9	59.4±10.9	75.6±6.0	69.4±4.8	73.6±1.9	64.2±4.6	71.0±6.5	62.8±6.4	74.2±2.7
14	0.0±0.0	60.8±12.9	10.8±7.7	90.4±3.8	31.2±7.6	97.2±1.0	93.2±3.3	96.8±1.7	56.4±6.0	97.8±1.0	78.4±11.8	96.6±2.5
15	0.0±0.0	51.8±8.0	25.4±12.4	60.4±7.1	59.4±3.0	66.2±3.9	59.6±6.8	67.6±2.7	64.0±6.6	64.8±10.4	62.8±5.6	63.2±2.3
16	0.6±1.2	68.2±4.7	81.8±4.5	72.2±1.7	78.4±4.8	80.4±2.4	83.0±6.2	83.6±3.8	83.4±4.5	83.0±5.6	79.6±4.8	83.2±3.4
17	3.2±6.4	34.2±6.2	8.6±2.6	52.6±1.9	28.8±5.9	57.0±4.3	48.6±2.6	53.0±2.3	37.4±2.6	45.4±6.3	40.2±6.4	44.0±6.9
18	0.4±0.8	80.0±4.3	91.2±1.5	92.4±1.4	90.6±1.0	92.8±2.0	94.4±1.9	93.8±0.4	91.4±1.5	93.4±2.2	93.8±1.2	94.2±1.2
19	0.0±0.0	82.6±6.7	13.8±8.4	97.6±0.5	57.2±14.3	97.6±0.5	93.4±2.0	97.4±0.8	79.8±7.4	97.0±1.6	79.2±10.8	97.2±0.8
20	4.0±8.0	79.2±10.2	92.2±3.3	98.8±1.0	96.8±1.6	98.8±0.8	99.2±0.8	99.6±0.5	99.0±0.6	99.6±0.5	99.2±0.8	99.6±0.8
21	0.0±0.0	56.6±2.9	15.4±8.9	79.8±9.7	39.2±6.4	88.4±5.6	88.2±3.4	92.6±1.9	61.2±5.3	93.8±1.6	75.2±11.2	94.4±2.7
22	0.0±0.0	64.0±10.2	26.8±16.0	73.6±2.8	42.4±6.5	78.8±1.5	70.4±4.2	85.0±4.2	61.4±6.1	84.8±3.4	65.6±11.7	82.2±2.2
23	0.0±0.0	54.6±10.3	13.2±4.2	68.8±3.1	38.8±3.4	72.6±4.6	70.2±6.8	83.6±5.1	54.2±6.2	79.4±6.2	61.2±11.8	84.4±3.8
24	0.0±0.0	37.2±13.6	1.8±1.6	86.0±0.6	13.4±3.3	90.8±2.1	70.8±10.2	93.6±0.8	34.8±5.9	93.0±2.0	51.2±23.9	92.4±1.6
25	0.0±0.0	84.4±3.4	20.0±9.0	86.2±1.6	42.4±4.1	82.2±3.1	84.6±2.7	86.6±2.9	66.2±7.4	82.4±4.4	77.0±5.3	86.8±4.1
26	1.4±2.8	78.6±7.7	33.6±11.7	81.2±5.0	57.6±4.2	93.6±2.4	87.8±3.7	93.2±1.2	75.8±8.7	90.4±5.0	79.8±8.9	92.8±2.1
27	0.0±0.0	79.2±10.9	94.8±3.9	93.4±3.7	99.0±0.6	97.4±1.4	99.8±0.4	96.6±2.1	99.2±0.4	98.0±1.1	99.8±0.4	96.6±1.2
28	0.8±1.6	75.8±3.0	58.0±12.9	90.0±3.6	73.8±7.2	96.2±3.1	91.8±1.7	98.0±0.9	80.8±4.5	98.2±1.3	85.0±5.8	97.0±2.5
29	0.4±0.8	88.2±6.7	43.4±17.4	95.2±1.6	67.6±2.2	98.8±0.4	98.0±1.1	99.6±0.5	82.2±6.4	99.6±0.5	86.6±7.0	98.6±0.5

TABLE 8. Melanoma (12 classes): Per-class accuracy (recall, %) across concentrations. Mean ± std.

Class	10% R	10% P	20% R	20% P	50% R	50% P	75% R	75% P	90% R	90% P	100% R	100% P
0	0.0±0.0	64.0±15.0	0.0±0.0	80.0±17.9	0.0±0.0	96.0±8.0	64.0±15.0	96.0±8.0	84.0±8.0	92.0±9.8	96.0±8.0	92.0±9.8
1	20.0±40.0	13.3±6.7	0.0±0.0	23.3±17.0	0.0±0.0	60.0±8.2	90.0±8.2	96.7±6.7	100.0±0.0	93.3±8.2	90.0±8.2	96.7±6.7
2	0.0±0.0	68.6±14.0	20.0±40.0	65.7±14.6	80.0±33.3	100.0±0.0	97.1±5.7	100.0±0.0	97.1±5.7	100.0±0.0	100.0±0.0	100.0±0.0
3	0.0±0.0	92.0±9.8	20.0±40.0	72.0±16.0	8.0±16.0	84.0±8.0	100.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0
4	40.0±48.9	46.7±6.7	20.0±40.0	40.0±13.3	36.7±45.2	63.3±16.3	76.7±17.0	63.3±19.4	86.7±12.5	70.0±12.5	86.7±6.7	40.0±8.2
5	0.0±0.0	20.0±17.9	0.0±0.0	52.0±16.0	0.0±0.0	76.0±8.0	76.0±8.0	88.0±9.8	80.0±0.0	92.0±16.0	84.0±8.0	96.0±8.0
6	0.0±0.0	24.0±19.6	0.0±0.0	80.0±12.6	0.0±0.0	80.0±0.0	92.0±9.8	84.0±8.0	96.0±8.0	80.0±0.0	96.0±8.0	84.0±8.0
7	0.0±0.0	36.0±8.0	0.0±0.0	48.0±20.4	0.0±0.0	68.0±9.8	100.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0	96.0±8.0
8	0.0±0.0	28.0±20.4	0.0±0.0	28.0±9.8	8.0±16.0	54.0±8.0	76.0±8.0	76.0±8.0	76.0±19.6	68.0±9.8	84.0±8.0	76.0±8.0
9	40.0±48.9	48.0±16.0	20.0±40.0	53.0±18.9	0.0±0.0	76.0±14.9	92.0±9.8	96.0±8.0	100.0±0.0	96.0±8.0	100.0±0.0	100.0±0.0
10	36.0±44.5	32.0±20.4	20.0±40.0	32.0±9.8	0.0±0.0	64.0±19.6	80.0±30.9	96.0±8.0	88.0±16.0	100.0±0.0	100.0±0.0	100.0±0.0
11	0.0±0.0	0.0±0.0	0.0±0.0	44.0±14.9	0.0±0.0	48.0±16.0	84.0±14.9	56.0±14.9	84.0±8.0	68.0±9.8	100.0±0.0	72.0±20.4

TABLE 9. COVID-19 (2 classes): Per-class accuracy (recall, %) by concentration. Mean ± std.

Conc.	RSPTE C0	RSPTE C1	Random C0	Random C1
10%	68.0 ± 18.6	55.0 ± 4.7	68.0 ± 41.2	32.5 ± 41.5
20%	72.0 ± 7.8	78.8 ± 9.4	0.0 ± 0.0	100.0 ± 0.0
50%	85.3 ± 7.8	85.0 ± 9.4	40.0 ± 49.0	60.0 ± 49.0
75%	86.7 ± 4.2	81.3 ± 4.0	0.0 ± 0.0	98.8 ± 2.5
90%	82.7 ± 6.8	87.5 ± 6.9	20.0 ± 40.0	80.0 ± 40.0
100%	82.7 ± 3.3	88.8 ± 4.7	0.0 ± 0.0	98.8 ± 2.5

TABLE 10. Wheat Lines (2 classes): Per-class accuracy (recall, %) by concentration. Mean ± std.

Conc.	RSPTE C0	RSPTE C1	Random C0	Random C1
0.10%	88.6 ± 1.9	76.2 ± 3.3	0.0 ± 0.0	100.0 ± 0.0
0.20%	89.8 ± 2.8	81.7 ± 5.0	20.0 ± 40.0	80.0 ± 40.0
0.50%	88.4 ± 2.1	91.1 ± 1.5	76.9 ± 38.6	34.6 ± 38.0
1%	93.4 ± 0.7	93.0 ± 0.5	98.2 ± 0.6	86.3 ± 3.8
5%	96.8 ± 0.2	96.7 ± 0.2	97.5 ± 0.5	95.8 ± 0.8
7%	96.8 ± 0.1	97.1 ± 0.3	97.8 ± 0.3	96.4 ± 0.7
10%	97.0 ± 0.2	97.5 ± 0.2	97.4 ± 0.3	97.1 ± 0.6