Eiaki V. Morooka

Quantum Chemistry Preprocessing for Industrial Data: A Case Study with SOAP and MNIST

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor’s thesis

23.5.2025

Abstract

Author:	Eiaki V. Morooka
Title:	Quantum Chemistry Preprocessing for Industrial Data: A Case Study with SOAP and MNIST Point Clouds
Number of Pages:	5 pages + 1 appedix
Date:	23 May 2025
Degree:	Bachelor of Engineering
Degree Programme:	Information Technology
Professional Major:	Game Development
Supervisors:	Miikka Mäki-Uuro, Senior Lecturer

The goal of this study was to explore the application of quantum chemistry-inspired feature extraction techniques to structured point cloud data, using the Smooth Overlap of Atomic Positions (SOAP) descriptor for classification tasks. The work focuses on adapting SOAP, traditionally used for atomic structures, to process 2D grayscale MNIST handwritten digits transformed into 3D point clouds. The aim is to achieve invariant representations under rotation, translation, and mirror transformations while eliminating the need for data augmentation.
Methodologically, the MNIST digits were converted into 3D point clouds, with SOAP applied to generate rotationally and translationally invariant feature vectors. The resulting SOAP power spectra are used as inputs for classification models. Dimensionality reduction was investigated using autoencoders to analyze the trade-off between feature compression and information retention. Additionally, the robustness of the approach was tested by introducing Gaussian noise to evaluate classification performance under data perturbations.
The findings demonstrate that SOAP-based feature extraction effectively captures local structural information, enabling accurate classification without reliance on augmented data. Dimensionality reduction preserves essential features while significantly compressing the data. The method exhibits stability under noise, maintaining performance despite perturbations.
The study highlights the versatility of SOAP for non-chemistry-based point cloud data, offering a robust alternative to traditional feature extraction techniques. The results suggest potential applications in other domains requiring invariant representations of structured data. Further research could explore scalability to larger datasets or different data modalities.

Keywords: SOAP, Auto Encoding, Point Cloud

Contents

1 Introduction

Feature extraction plays a crucial role in machine learning and data processing, shaping how models interpret structured data. Traditional image classification tasks rely on convolutional neural networks (CNNs) [1], often requiring extensive data augmentation to account for transformations such as rotation, translation, and mirror symmetry. However, augmentation is computationally expensive and does not inherently guarantee invariance.

In quantum chemistry and materials science, the Smooth Overlap of Atomic Positions (SOAP) descriptor [2] has been widely used to describe atomic structures in a rotational, translational, and mirror-invariant manner. This study explores the application of SOAP for point cloud-based feature extraction in image analysis, using MNIST handwritten digits as a benchmark dataset [3].

By treating pixel positions in grayscale images as point clouds and applying SOAP descriptors, the aim is to extract robust features that naturally encode spatial relationships while eliminating the need for augmentation. Additionally, how feature compression impacts classification accuracy and assess robustness under noisy conditions was investigated.

The full paper can be found at https://www.preprints.org/manuscript/202502.2316/v1[4], and also in the appendix.

1.1 Project Overview

This project investigates a novel preprocessing technique inspired by quantum chemistry for feature extraction in point cloud-based machine learning. The key steps include:

- Converting MNIST grayscale images into 3D point clouds, where pixels are treated as spatial entities.

- Applying SOAP descriptors to extract rotational, translational, and mirror-invariant features.
- Evaluating classification performance using SOAP power spectra as feature vectors.
- Analyzing the effect of feature compression using autoencoders and PCA.
- Studying robustness by introducing Gaussian noise to point cloud positions and assessing classification stability.

1.2 Background

The SOAP descriptor was originally developed to describe atomic environments by computing high-dimensional feature vectors that remain invariant to spatial transformations. SOAP-based representations have been extensively used in materials science and molecular simulations. The key properties of SOAP vectors include:

- Rotational, translational, and mirror invariance.
- Rich local geometric encoding without explicit data augmentation.
- Compatibility with machine learning models for classification and clustering tasks.

While previous research has focused on molecular applications, this study repurposes SOAP for image processing by treating pixel distributions as point clouds, enabling a new form of feature extraction for classification tasks.

Unlike traditional image descriptors, SOAP-based representations inherently encode local spatial relationships, making them robust to geometric transformations. Given the increasing interest in efficient preprocessing techniques, the potential of SOAP descriptors in reducing computational overhead and improving model generalization is significant.

1.3 Experiments and Results

1.3.1 Experiment 1: SOAP Feature-Based Classification

Objective: To classify MNIST digits using SOAP-based feature extraction.

Method: Each MNIST digit was transformed into a 3D point cloud. SOAP descriptors were computed for each local neighborhood, and classification was performed using a neural network trained on the SOAP power spectra.

Results: The best-performing configuration achieved a validation accuracy of 68.44%. The confusion matrix analysis showed consistent classification across digits, with misclassifications primarily occurring between visually similar digits. These results highlight the effectiveness of SOAP descriptors in capturing local structures within digit images.

1.3.2 Experiment 2: Feature Compression Using Autoencoders

Objective: To reduce SOAP descriptor dimensionality while preserving classification performance.

Method: Principal Component Analysis (PCA) and autoencoders were used to compress the high-dimensional SOAP vectors. The impact on classification accuracy was measured as a function of compression ratio.

Results: Linear compression via PCA retained over 90% accuracy with a 50% reduction in dimensionality, while deep autoencoders achieved similar compression with marginally better retention of information. The trade-off between compression and predictive performance was found to be dependent on the number of retained principal components or latent space size, with diminishing returns at extreme compression levels.

1.3.3 Experiment 3: Robustness to Noise Perturbations

Objective: To assess classification performance under noisy conditions.

Method: Gaussian noise was added to the pixel positions in the 3D point cloud, simulating real-world distortions.

Results: The classification accuracy exhibited gradual degradation as noise intensity increased, demonstrating the robustness of SOAP-based features compared to traditional pixel-based representations. Notably, even under high noise conditions, the SOAP-based classifier maintained significantly higher accuracy than standard pixel-based approaches, reinforcing its resilience to perturbations.

1.4 Conclusion

This study demonstrates the feasibility of using SOAP descriptors for point cloud-based feature extraction in image classification tasks. By leveraging quantum chemistry-inspired techniques, a rotationally and translationally invariant representation that eliminates the need for extensive data augmentation was achieved. Additionally, the experiments show that SOAP descriptors can be effectively compressed while preserving classification accuracy and remain robust against noise perturbations.

The findings suggest that SOAP-based preprocessing could be valuable in industrial applications requiring efficient and invariant feature extraction. Future work could explore its integration with deep learning architectures beyond simple classification models and its application to more complex datasets.

References

1 O’Shea, Keiron & Nash, Ryan. 2015. An Introduction to Convolutional Neural Networks. arXiv: 1511.08458 [cs.NE]. <https://arxiv.org/abs/1511.08458>.

2 Bartók, Albert P.; Kondor, Risi & Csányi, Gábor. 2013. ”On Representing Chemical Environments”. Physical Review B 87.18. <http://dx.doi.org/10.1103/PhysRevB.87.184115>.

3 R, Amarnath & V, Vinay Kumar. 2023. Pruning Distorted Images in MNIST Handwritten Digits. arXiv: 2307.14343 [cs.CV]. <https://arxiv.org/abs/2307.14343>.

4 Morooka, Eiaki V.; Omae, Yuto; Hämäläinen, Mika & Takahashi, Hirotaka. Helmikuu 2025. ”Benchmarking Point Cloud Feature Extraction with Smooth Overlap of Atomic Positions (SOAP): A Pixel-Wise Approach for MNIST Handwritten Data”. Preprints. <https://doi.org/10.20944/preprints202502.2316.v1>.

Appendix: Benchmarking Point Cloud Feature Extraction with Smooth Overlap of Atomic Positions (SOAP): A Pixel-Wise Approach for MNIST Handwritten Data

Benchmarking Point Cloud Feature Extraction with Smooth Overlap of Atomic Positions (SOAP): A Pixel-Wise Approach for MNIST Handwritten Data

Eiaki V. Morooka , Yuto Omae, Mika Hämäläinen and Hirotaka Takahashi

April 24, 2025

Abstract

In this study, we introduce a novel application of the Smooth Overlap of Atomic Positions (SOAP) descriptor for pixel-wise image feature extraction and classification as a benchmark for SOAP point cloud feature extraction, using MNIST handwritten digits as a benchmark. By converting 2D images into 3D point sets, we compute pixel-centered SOAP vectors that are intrinsically invariant to translation, rotation, and mirror symmetry. We demonstrate how the descriptor's hyperparameters—particularly the cutoff radius—significantly influence classification accuracy, and show that the high-dimensional SOAP vectors can be efficiently compressed using PCA or autoencoders with minimal loss in predictive performance. Our experiments also highlight the method's robustness to positional noise, exhibiting graceful degradation even under substantial Gaussian perturbations. Overall, this approach offers an effective and flexible pipeline for extracting rotationally and translationally invariant image features, potentially reducing reliance on extensive data augmentation and providing a robust representation for further machine learning tasks.

1 Introduction

Feature extraction is often used in machine learning and data analysis, shaping the quality and relevance of the input data for a given task. In the field of image processing, training robust models often requires addressing the challenges posed by spatial transformations such as translation, rotation, and mirror symmetry [1]. These transformations can significantly affect pixel intensities and spatial relationships within an image, creating challenges for machine learning models to generalize effectively. To mitigate these issues, data augmentation techniques are commonly employed [2, 3], but they introduce their own limitations:

- **Translation invariance:** Images may undergo shifts in spatial position, causing pixel values to move across the image grid. Training models to handle translation typically involves augmenting the dataset with translated versions of the original images.
- **Rotation invariance:** Images can appear in different orientations. Achieving robustness to rotations requires augmenting the dataset with rotated images, increasing computational cost and memory requirements.
- **Mirror symmetry:** Certain images may appear as mirror reflections. Training models to handle such transformations often involves flipping the images horizontally or vertically, further expanding the dataset.

While these augmentation techniques are effective to some extent, they are computationally expensive and do not inherently guarantee invariance [4]. There is a growing need for feature extraction techniques that are intrinsically invariant to such transformations, reducing the reliance on augmentation and enhancing model efficiency.

In quantum chemistry and materials science, the Smooth Overlap of Atomic Positions (SOAP) descriptor [5, 6, 7] has revolutionized the way local structural environments around atoms are encoded. Originally designed to represent atomic configurations in molecular and crystalline systems, SOAP has found success in a variety of machine learning tasks, including potential energy surface modeling [8], molecular similarity analysis [9], and structure-property predictions [10].

SOAP encodes structural information by representing atomic environments as high-dimensional, rotationally and translationally invariant features derived from smooth atomic density overlaps. These descriptors are computed using expansions in angular basis and radial basis functions, creating a rich representation of the local geometry and chemistry around atoms. Their continuous, differentiable nature makes SOAP particularly attractive for machine learning workflows that require robust and transferable representations.

While SOAP has primarily been applied to atomistic systems, this work presents a novel application of SOAP descriptors to the domain of image analysis. Specifically, we propose using SOAP-inspired spectra for pixel-wise feature extraction, introducing a new methodology for representing local pixel environments in images. Analogous to

atomic neighborhoods, each pixel can be treated as a "local environment" characterized by the intensity values and spatial relationships of its neighboring pixels. By extending the principles of SOAP to these pixel neighborhoods, we derive rotationally, translationally, and mirror invariant descriptors capable of capturing rich, spatially-aware features.

The novelty of this approach lies in its ability to bridge concepts from quantum chemistry with computer vision, creating a new paradigm for pixel-wise feature extraction. Unlike traditional image descriptors that rely on predefined filters or convolutional kernels, the SOAP framework offers a fundamentally different perspective by encoding the spatial "overlap" of pixel distributions. This enables the extraction of high-dimensional features that are both robust to noise and sensitive to local variations, making them ideal for complex tasks such as segmentation, classification, and object recognition.

In this paper, we predict MNIST handwritten data [11], pixel-wise, using SOAP spectra as a feature extraction technique. We observe that the correlation matrix of the SOAP vectors reveals a high degree of correlation among its elements. To address this, we measure the compression efficiency of the SOAP descriptors by comparing three methods: linear autoencoding, principal component analysis (PCA) , and deep autoencoding [12, 13]. Additionally, we analyze the prediction accuracy in relation to the degree of compression. Finally, we evaluate the robustness of the approach by introducing noise into the dataset by perturbing pixel positions with Gaussian random distributions [14] and assess the predictive performance under these conditions.

Using the mathematical rigor and invariance properties of SOAP, this study introduces a novel feature extraction technique that offers a new perspective in image processing. To our knowledge, this is the first application of SOAP-based methodologies in the context of pixel-wise image analysis. This interdisciplinary approach not only enhances the toolbox of image processing techniques but also demonstrates the potential for repurposing advanced descriptors from quantum chemistry for entirely new domains.

2 Related work

Data augmentation has long served as a crucial technique in machine learning for mitigating the challenges of limited data and overfitting. Initially introduced as a statistical method to facilitate maximum likelihood estimation from incomplete data [15, 16], augmentation techniques soon found applications in Bayesian analysis [15] and later evolved to become a staple in modern machine learning workflows. Early approaches in image processing, for instance, focused on perturbing data through affine transformations to simulate different viewpoints and enhance training datasets [17]. These geometric transformations—comprising rotations, translations, and mirror reflections—were adopted to instill invariance in convolutional neural networks (CNNs), despite the increased computational and memory overhead that comes with augmenting the dataset with multiple modified copies of each image.

The evolution of data augmentation techniques saw the integration of more sophisticated methods such as elastic distortions [17], color space adjustments, and noise injection, all aimed at enhancing the diversity of training data. These methods have been instrumental in addressing issues such as class imbalance, where techniques like the Synthetic Minority Over-sampling Technique (SMOTE) generate new synthetic examples by interpolating between minority class samples [18]. Such synthetic oversampling methods have proven particularly effective in domains where data scarcity is pronounced, including medical diagnosis and signal processing [19].

More recent research has turned to generative models, such as Generative Adversarial Networks (GANs) [20], to produce high-fidelity synthetic data. These approaches have not only been applied to image classification tasks but also extended to the augmentation of biological and mechanical signals, thereby enhancing model performance in applications ranging from EEG-based emotion recognition [21] to industrial control systems [22].

Despite the broad success of these data augmentation strategies, a persistent challenge remains: while the augmentation process can enrich the dataset, it does not inherently confer invariance to spatial transformations such as translation, rotation, and mirror symmetry. This limitation often necessitates large-scale data augmentation to achieve robustness, which in turn incurs significant computational costs.

Another influential development in pixel-wise machine learning is the U-Net architecture, which has become a benchmark for image segmentation tasks, particularly in biomedical imaging [23]. U-Net employs an encoder-decoder structure with skip connections that efficiently combine low-level spatial information with high-level semantic features, enabling precise localization and robust segmentation even with limited training data. Its success has spurred numerous variants and inspired a range of applications in pixel-level prediction tasks. However, U-Net and similar architectures typically rely on extensive data augmentation and complex network designs, which can be computationally demanding and may still not guarantee complete invariance to spatial transformations [23].

While traditional data augmentation techniques and architectures such as U-Net enhance model robustness by artificially expanding training datasets and leveraging complex encoder-decoder frameworks, SOAP-inspired descriptors operate on a fundamentally different principle. Rather than relying on extensive augmentation to enforce invariance, SOAP directly encodes spatial relationships through its mathematical formulation. By embedding invariance to translation, rotation, and mirror symmetry at the feature representation level, SOAP circumvents the need for excessive data manipulation and augmentation. This not only streamlines model training but also provides a more structured and theoretically grounded approach to capturing local geometric patterns in image data.

3 Methodology

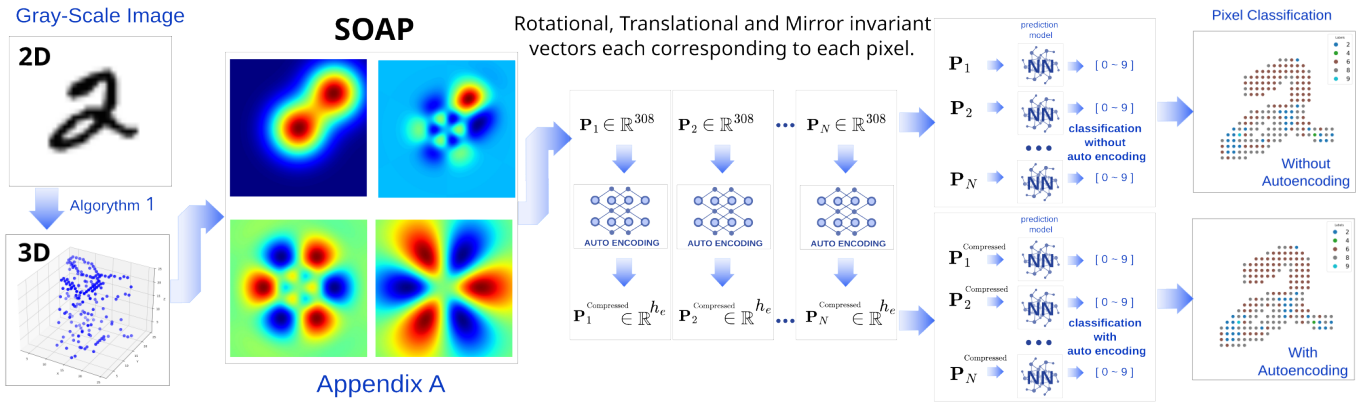


Figure 1: In our study, we take grayscale MNIST handwritten images as input and project them into 3D point clouds. These points are then processed using the SOAP algorithm to generate SOAP spectra—feature vectors that encode local environments while remaining invariant to rotation, translation, and mirror symmetry. Each vector can be labeled and used for classification or regression tasks with models such as feed-forward neural networks. Due to SOAP's inherent symmetry invariance, data augmentation for rotation, translation, and flipping is unnecessary during training. Additionally, since some SOAP components are highly correlated, dimensionality reduction techniques such as autoencoding or PCA can be applied for compression.

Our objective is to extract the local information on a pixel, by getting the SOAP vector (or SOAP spectrum), on an image Figure 2a. In this section, we will go through the mothodology of how SOAP spectra are acquired and how the images are projected from 2D to 3D to make that possible. An overview of our methodology is shown in Figure 1.

3.1 SOAP Formulation

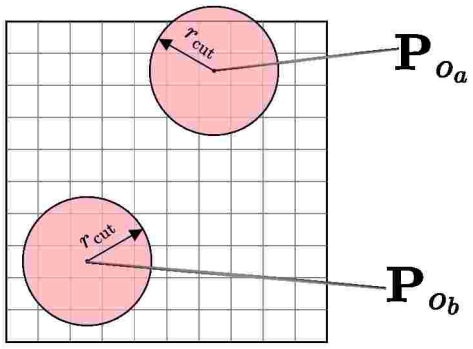
The Smooth Overlap of Atomic Positions (SOAP) descriptor provides a robust framework for encoding local environments, representing them as rotationally, translationally and mirror symmetry invariant features. Originally designed for quantum chemistry applications, the SOAP descriptor was adapted in this study for pixel-wise feature extraction in images. This section outlines the mathematical formulation of SOAP.

3.1.1 Density Function

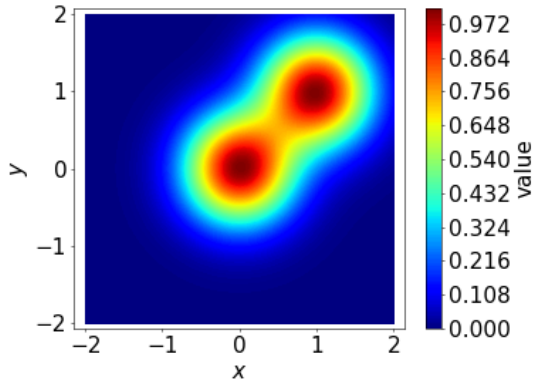
We describe the local environment around a reference point $\mathbf{r}_o$ using a density function ρ_o, where the contributions from surrounding points within a hyperparameter r_{cut}, are smoothly distributed through Gaussian smoothing:

$$\rho_o(x_o, y_o, z_o) = \sum_i \exp\left(-\frac{\|\mathbf{r}_o - \mathbf{R}_i\|^2}{2\sigma_p^2}\right), \tag{1}$$

where o represents the local point, $\mathbf{R}_i = (x_i, y_i, z_i)^\top$ are the positions of neighboring points, σ_p is a hyperparameter that determines the width of the Gaussian smoothing, and $\mathbf{r}_o = (x_o, y_o, z_o)^\top$. An example can be seen in Figure 2b).



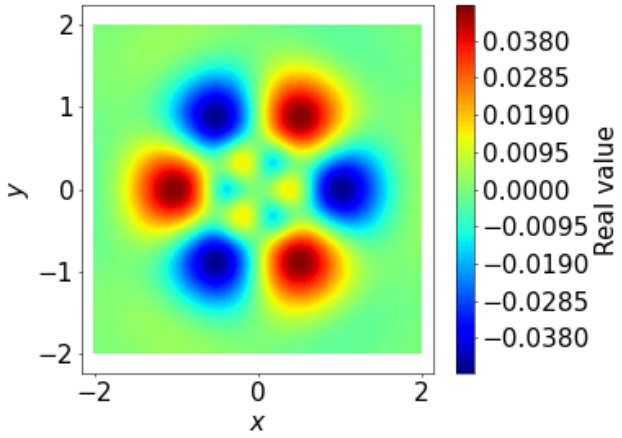
(a) A 10 by 10 image, with two SOAP spectra. The information contained in the SOAP spectra is determined solely by the length of r_{cut}, a hyperparameter; information beyond that is not captured.



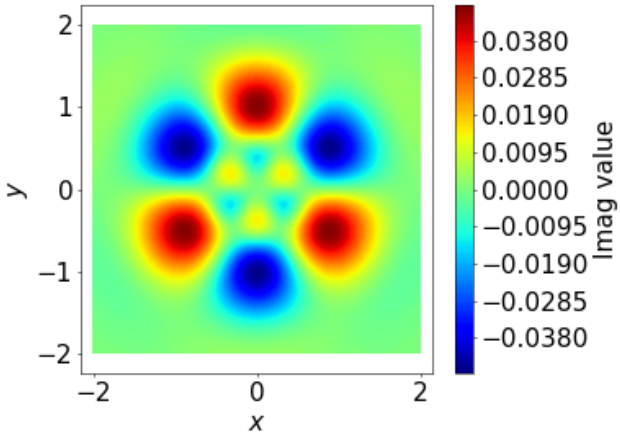
(b) Example of a cross section of a Density Function, ρ_{o}, with 2 points. $P_1 = (0, 0, 1)^\top$ and $P_2 = (1, 1, 1)^\top$ with $\sigma_p = 0.5$.

Figure 2: (a) Illustration of our approach for extracting features from a pixel. $\mathbf{P}_{o_a}$ and $\mathbf{P}_{o_b}$ represent independent SOAP vectors that encode local structural information up to a distance of r_{cut}. (b) Example of a density function with two sample points.

3.1.2 Spatial Basis Function



(a) Real Φ_{053} with a cross section at $z = 1.0$



(b) Imag Φ_{053} with a cross section at $z = 1.0$

Figure 3: Example of a cross section of a Spatial Basis Function, using Spherical Harmonics and GTO radial basis function.

The spatial basis function $\Phi^o_{nlm}(x_o, y_o, z_o)$ is defined as the product of two components: a radial function $g^o_{nl}(r_o)$ and an angular function $Y^o_{lm}(\theta_o, \phi_o)$. These components are combined as follows:

$$\Phi^o_{nlm}(x_o, y_o, z_o) = \Phi^o_{nlm}(r_o, \theta_o, \phi_o) = g^o_{nl}(r_o) Y^o_{lm}(\theta_o, \phi_o), \tag{2}$$

where, $g^o_{nl}(r_o)$ captures the radial variation, while $Y^o_{lm}(\theta_o, \phi_o)$ encodes the angular dependence in spherical coordinates, where $r_o = \sqrt{x_o^2 + y_o^2 + z_o^2}$ is the distance from the reference point, $\theta_o = \arccos(z_o/r_o)$ is the polar angle, and $\phi_o = \arctan 2(y_o, x_o)$ is the azimuthal angle. See Figure 3) as an example.

3.1.3 Radial Basis Functions

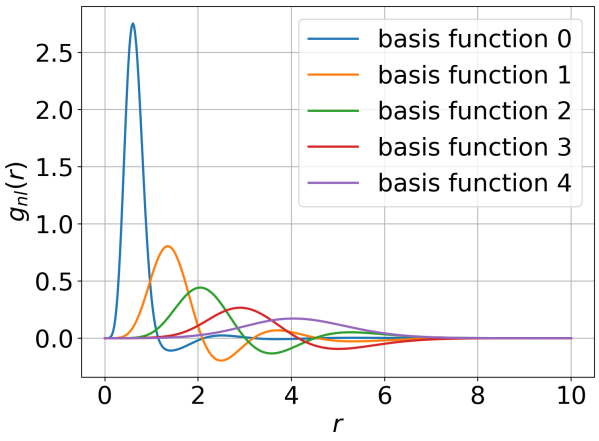
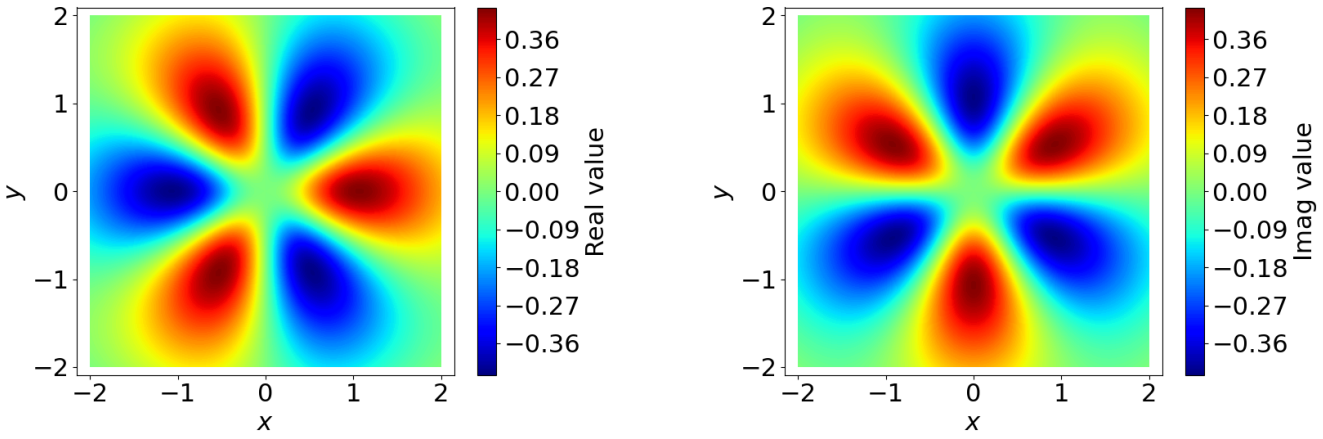


Figure 4: Example of Radial Basis Functions with $r_{\text{cut}} = 10$.

The radial basis functions $g^o_{nl}(r_o)$ capture the radial dependencies of the local environment. These functions may either depend on the angular number l (denoted as $g^o_{nl}(r_o)$) or be independent of l (denoted as $g^o_n(r_o)$). Orthonormality[1] is a key property of these basis functions, ensuring that the expansion coefficients are unique and non-redundant (See Figure 4)) .

3.1.4 Angular Basis Functions



(a) Real Y_{53} with a cross section at $z = 1.0$

(b) Imag Y_{53} with a cross section at $z = 1.0$

Figure 5: Spherical Harmonics as Angular Basis Functions.

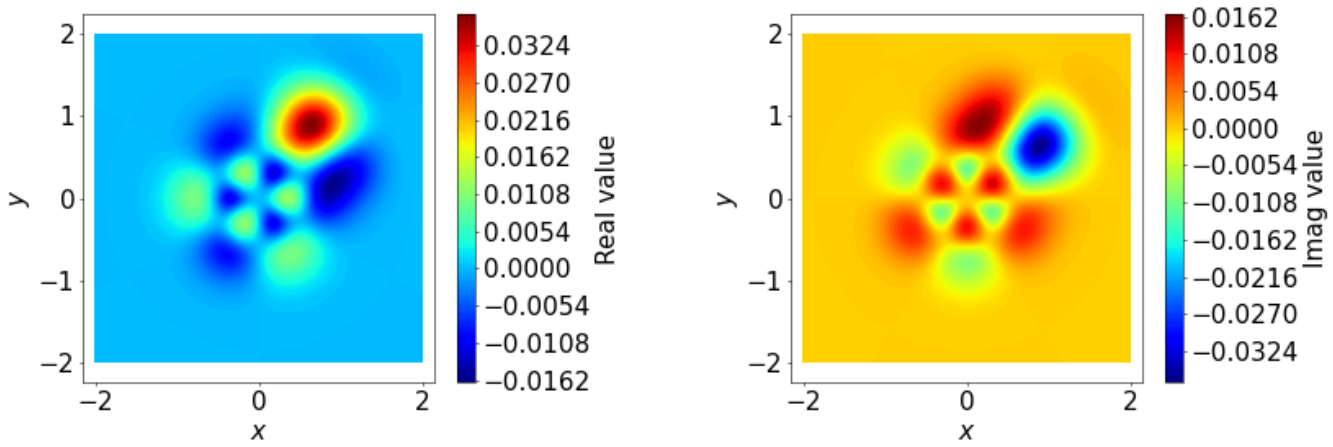
The angular basis functions $Y^o_{lm}(\theta_o, \phi_o)$ encode the angular dependencies of the local environment. These functions are constructed to represent directional information and are parameterized by two indices: l, which controls the level of angular detail, and m, which distinguishes variations within each level. This is analogous to the frequencies of sine and cosine functions around a sphere. In our case, we use spherical harmonics $Y_{lm}(\theta, \phi)$ as the angular basis functions (See Figure 5).

A key property of the angular basis functions is their orthonormality, ensuring that the components of the representation remain independent and non-redundant. Additionally, spherical basis functions depend only on angular coordinates, meaning they are invariant to scaling of the input vector: for any constant a,

$$Y_{lm}(x, y, z) = Y_{lm}(ax, ay, az). \tag{3}$$

[1] A set of functions $\{f_i\}$ is orthonormal if it satisfies $\langle f_i, f_j \rangle = \int_a^b f_i(x) f_j(x)\, dx = 0$ for $i \neq j$ (orthogonality) and $\langle f_i, f_i \rangle = \int_a^b f_i^2(x)\, dx = 1$ (normalization).

3.1.5 SOAP Expansion Coefficients



(a) Real $\rho_o \Phi^o_{053}$ with a cross section at $z = 1.0$

(b) Imag $\rho_o \Phi^o_{053}$ with a cross section at $z = 1.0$

Figure 6: Example of a cross section of an integrand $\rho_o \times \Phi^o_{053}$, using the density function in Figure 2b.

The expansion coefficients c^o_{nlm} are key to representing the local environment in the SOAP formulation. These coefficients quantify the projection of the local environment density function $\rho_o(x_o, y_o, z_o)$ onto the spatial basis functions $\Phi^o_{nlm}(x_o, y_o, z_o)$, which combine radial and angular components. This projection ensures that the complex spatial information encoded in ρ_o is transformed into a compact and expressive feature representation:

$$c^o_{nlm} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \rho_o(x_o, y_o, z_o) \Phi^o_{nlm}(x_o, y_o, z_o)\, dx_o\, dy_o\, dz_o. \tag{4}$$

The orthonormality of the basis functions ensures that these coefficients are unique and non-redundant, making them an efficient and interpretable representation of the local environment. An exmaple of an integrant is shown in Figure 6.

3.1.6 SOAP Power Spectrum

The SOAP power spectrum is a descriptor that is rotationally, translationally, and mirror invariant. It is computed as the inner product of the expansion coefficients over m, capturing the essential characteristics of the local environment. The power spectrum is defined as:

$$\mathbf{P}^{\text{SOAP}}_o = P^o_{nn'l} = \pi \sqrt{\frac{8}{2l+1}} \bar{\mathbf{c}}^{\top}_{nl} \mathbf{c}_{n'l} = \pi \sqrt{\frac{8}{2l+1}} \sum_m \bar{c}^o_{nlm} c^o_{n'lm}, \tag{5}$$

where $\bar{c}^o_{nlm}$ is the complex conjugate of c^o_{nlm}. The inner product over m ensures that the power spectrum encodes information about the radial and angular dependencies while removing orientation-specific details.

The resulting descriptor, $\mathbf{P}^{\text{SOAP}}_o$, is a high-dimensional, invariant feature vector that represents the local environment around the reference point $\mathbf{r}_o$. This invariance is critical for tasks requiring consistent feature extraction across different orientations and positions.

A more detailed examples of the radial basis functions, angular basis functions, and coefficients, including their computation and role in feature construction, are provided in Appendix A.

3.2 Converting Images to 3D Points and Computing SOAP Descriptors

To adapt the SOAP formulation for image analysis, the pixel intensities of 2D images are converted into 3D point representations. These 3D points serve as the input for computing SOAP descriptors. This section explains the methodology for these steps.

3.2.1 Converting Gray-Scale Images to 3D Points

Each image is represented as a collection of 3D points, where the x and y coordinates correspond to the pixel positions in the image, and the z-coordinate is derived from the gray-scale pixel intensity which are the values from maximum 255 divided by 10 to maximum 25.5, independent of Gaussian scaling. Algorithm 1 describes the procedure for generating

the 3D representation, including an optional Gaussian displacement to account for variability or noise in the data. Variable descriptions are show in detail in Table 2 and Table 3.

For a single $M_0 \times M_1$ image, the intensity values of each pixel are scaled and mapped to the z-axis, while the x and y coordinates retain their pixel positions. A Gaussian displacement with standard deviation $\sigma_{\text{disturbance}}$ is applied to the x, y, and z coordinates to introduce variability. This random displacement is particularly useful when studying the divergence of SOAP features under small perturbations. Only non-zero intensity pixels are considered in the transformation, ensuring computational efficiency by excluding irrelevant regions.

The result of this process is a 3D structure $\mathbf{F}_k$, where each point $(x_i, y_i, z_i)^\top$ corresponds to a pixel in the original image. This step bridges the gap between the 2D image space and the 3D local environments required for SOAP descriptor computation (Figure 7).

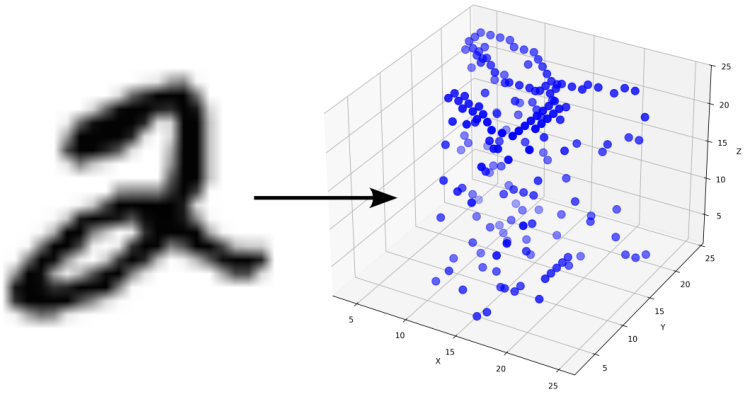


Figure 7: Example of a projection of a 2D image to a 3D Structure.

Algorithm 1 ConvertImagesToXYZ (3D structure)

Require: • $\{I_k\}$: A collection of n images, each of size $M_0 \times M_1$.
• $\sigma_{\text{disturbance}}$: Standard deviation for Gaussian displacement.

Ensure: • Points $\{(x_i, y_i, z_i)^\top\}$ derived from the intensity values of each pixel in each image, with optional random displacement.

function ImageToXYZ($I, \sigma_{\text{disturbance}}$) ▷ Converts a single $M_0 \times M_1$ image I into 3D points.
  Initialize an empty list $\mathcal{P}$ for points.
  for $i \leftarrow 0$ **to** $M_0 - 1$ **do**
    for $j \leftarrow 0$ **to** $M_1 - 1$ **do**
      $z \leftarrow \lfloor I[i,j]/10 \rfloor$ ▷ Squashed intensity from 255 to a scale closer to the image dimension.
      if $z > 0$ **then** ▷ Eliminated pixels that are zero
        $x \leftarrow j + \mathcal{N}(0, \sigma^2_{\text{disturbance}})$
        $y \leftarrow i + \mathcal{N}(0, \sigma^2_{\text{disturbance}})$
        $z \leftarrow z + \mathcal{N}(0, \sigma^2_{\text{disturbance}})$
        Append (x, y, z) to $\mathbf{F}$.
      end if
    end for
  end for
  return F
end function

▷ Main procedure: convert all images $\{I_k\}$ into 3D point sets.
for $k \leftarrow 1$ **to** n **do**
  $\mathbf{F}_k \leftarrow$ ImageToXYZ($I_k, \sigma_{\text{disturbance}}$)
end for

3.2.2 Computing SOAP Descriptors for Image 3D Structures

Once the images are converted into 3D structures, SOAP descriptors are computed for each point in the 3D space. Algorithm 2 provides a detailed procedure for this computation. Each 3D structure $\mathbf{F}_k$ is processed to extract per-point SOAP descriptors, capturing the local spatial arrangement of points.

For each point $\mathbf{r}_o = (x_o, y_o, z_o)^\top$ in $\mathbf{F}_k$, the SOAP formulation outlined earlier is applied. Using the radial basis functions $g^o_{nl}(r_o)$ and angular basis functions $Y^o_{lm}(\theta, \phi)$, the density function is expanded into orthonormal basis functions. The expansion coefficients c^o_{nlm} are then used to compute the power spectrum $\mathbf{P}^{\text{SOAP}}_o$ as described in Equation (6).

The result is a matrix $\mathcal{P}_k$, where each row represents the SOAP descriptor $\mathbf{P}_o$ for a single point in the 3D structure. The descriptors encode local spatial patterns in a manner invariant to rotations, translations, and mirror symmetry. By aggregating the SOAP descriptors across all points, a comprehensive feature set for the image is obtained, enabling pixel-wise classification and other machine learning tasks.

This approach leverages the robustness of SOAP descriptors to provide a high-dimensional, invariant representation of image features, merging methodologies from quantum chemistry and computer vision.

Algorithm 2 Compute SOAP Matrices for a Set of 3D Structures

Require: • A collection of 3D structures $\{\mathbf{F}_k\}_{k=1}^n$, where each structure $\mathbf{F}_k$ is an $N_k \times 3$ matrix containing coordinates in 3D space:

$$\mathbf{F}_k = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_{N_k} & y_{N_k} & z_{N_k} \end{pmatrix}.$$

• Parameters for the SOAP descriptor: $\{\, r_{\text{cut}}, n_{\max}, l_{\max}, \sigma_p \,\}$.

Ensure: • A set of SOAP matrices $\{\mathcal{P}_k\}_{k=1}^n$, where each $\mathcal{P}_k$ is an $N_k \times d$ matrix holding the per-point SOAP vectors $\{\mathbf{P}_o\}$ for $\mathbf{F}_k$. In other words, $\mathcal{P}_k = \begin{pmatrix} \mathbf{P}_{o1} \\ \mathbf{P}_{o2} \\ \vdots \\ \mathbf{p}_{oN_k} \end{pmatrix}$, $\mathbf{P}_o \in \mathbb{R}^d$.

▷ **Algorithm Steps**

Initialize an empty set $\{\mathcal{P}_k\}_{k=1}^n$ store all SOAP matrices.
for $k \leftarrow 1$ **to** n **do**
  Let $\mathbf{F}_k$ be of size $(N_k \times 3)$.
  Define $\mathcal{P}_k$ as an $(N_k \times d)$ matrix.
  for $o \leftarrow 1$ **to** N_k **do** ▷ Compute SOAP vector $\mathbf{P}_o$ for the o-th point of structure $\mathbf{F}_k$.
    $\mathbf{P}_o \leftarrow \text{SOAP}(\mathbf{F}_k;\, r_{\text{cut}},\, n_{\max},\, l_{\max},\, \sigma_p)$
    ▷ Compute SOAP Spectra for each non-zero pixel.
    Insert $\mathbf{P}_o$ (a $1 \times d$ vector) into the o-th row of $\mathcal{P}_k$.
  end for
  Store $\mathcal{P}_k$ in the overall result set $\{\mathcal{P}_k\}_{k=1}^n$.
end for
return $\{\mathcal{P}_k\}_{k=1}^n$ ▷ Each $\mathcal{P}_k$ is a matrix capturing SOAP descriptors of $\mathbf{F}_k$.

4 Experiments and Results

To evaluate the performance of SOAP-based pixel-wise classification, a series of experiments were conducted. This section details the creation of the training datasets and outlines the experimental setups used to validate the methodology.

To systematically assess the effectiveness of SOAP-based pixel-wise classification, we conduct three key experiments. The first experiment focuses on hyperparameter optimization and pixel predictions, where we explore the impact of various SOAP descriptor parameters on model performance to determine an optimal configuration, then predict their classification of each pixel on the validation and test set. The second experiment investigates SOAP vector compression, evaluating different dimensionality reduction techniques—including PCA, linear autoencoding, and deep autoencoding—to quantify the trade-off between compression and classification accuracy. Finally, the third experiment examines robustness to pixel position perturbations, introducing Gaussian noise to pixel coordinates to assess the stability of SOAP-based feature extraction in the presence of spatial distortions.

4.1 Training Data Preparation

The training data for the experiments was derived from the SOAP descriptors computed for the 3D structures obtained from the MNIST dataset of handwritten digits. This dataset consists of 60,000 grayscale images of size 28×28 pixels. Each image was converted into a 3D structure following the methodology described in Section 2.3. 120,000 random SOAP spectra were collected, and split into 0.8:0.2 training and validation sets. The test dataset was collected from the MNIST handwritten dataset of 10,000 gray-scale images, and 10,000 random SOAP spectra were collected as a test set. The processes for generating the datasets in each experiment are detailed in Algorithm 3.

A for the training and validation, collection of SOAP matrices, $\{\mathcal{P}_k\}_{k=0}^{N_k-1}$, was computed using the `Dscribe` Python package [6, 24] and then randomly sampled to extract $T = 12 \times 10^4$ descriptors. These descriptors form the training feature matrix $\mathbf{X} \in \mathbb{R}^{T \times d}$. Each SOAP vector $\mathbf{P}_t$ was assigned a corresponding label ℓ_r, indicating its association with the r-th digit class in the MNIST dataset.

By using the MNIST dataset, this study leverages the well-established benchmark for handwritten digit recognition, enabling a rigorous evaluation of the proposed methodology and facilitating comparisons with other approaches.

To ensure numerical stability and facilitate model convergence, the SOAP descriptors were rescaled using a robust rescaling procedure, `RobustRescalor`, which adjusts the data based on the distribution of feature values. The rescaled descriptors and their corresponding labels constitute the final dataset, $(\mathbf{X}, \mathbf{y})$, with the parameters for robust rescaling for later use $\mathbf{s}_{RR}$ used in subsequent experiments.

The creation of this dataset ensures diversity in the sampled descriptors and maintains a balanced representation across the different input structures, facilitating robust model training and evaluation.

Algorithm 3 Random Extraction of SOAP Spectra with Rescaling

Require: Collection of SOAP vectors from 3D structures, $\{\mathcal{P}_k\}_{k=0}^{N_k-1} = \{\mathcal{P}_0, \ldots, \mathcal{P}_{N_k-1}\}$.
Ensure: Extracted and rescaled SOAP descriptors $\mathbf{X} \in \mathbb{R}^{T \times d}$, and corresponding labels $\mathbf{y} \in \mathbb{R}^T$.

Initialize $\mathbf{X} \leftarrow \mathbf{0}_{T \times d}$
Initialize $\mathbf{y} \leftarrow \mathbf{0}_T$
for $t \leftarrow 1$ **to** T **do**
    Pick a random index $r \in \{0, \ldots, N_k - 1\}$
    Select a random descriptor $\mathbf{P}_t \in \mathbb{R}^d$ from file $\mathcal{P}_r$
    $\mathbf{X}[t, :] \leftarrow \mathbf{P}_t$
    $\mathbf{y}[t] \leftarrow \ell_r$
end for
$(\mathbf{X}, \mathbf{s}_{RR}) \leftarrow \text{RobustRescalor}(\mathbf{X})$
return $\mathbf{X}, \mathbf{y}, \mathbf{s}_{RR}$

4.2 Experiment 1: Hyperparameter Optimization for SOAP and Predictions

4.2.1 Objective

Our objective is to identify the optimal SOAP descriptor parameters (r_{cut}, $n_{\max}$, $l_{\max}$, and σ_p) for pixel-wise digit classification and to evaluate their impact on model performance. This experiment aims to establish the sensitivity of the model to these parameters and determine the configurations that maximize validation accuracy while minimizing redundancy.

4.2.2 Methods

In this experiment, we employed a hyperparameter search using a Monte Carlo sampling strategy [25] over 168 trials. The search explored the following ranges: the neighborhood radius r_{cut} was varied between 2 and 100, the radial basis count $n_{\max}$ between 2 and 11, the angular resolution $l_{\max}$ between 2 and 11, and the Gaussian width σ_p between 1 and 10. The model architecture used was the pixel-wise Classification Model (see Figure 8), trained on 120,000 data points consisting of SOAP descriptors derived from MNIST images. The training protocol included the Adam optimizer [26] with a learning rate of 0.001, a batch size of 128, and 300 training epochs, with the data split into 80% for training and 20% for validation. Validation accuracy served as the primary metric, and the influence of individual hyperparameters was analyzed by correlating them with accuracy trends across the trials.

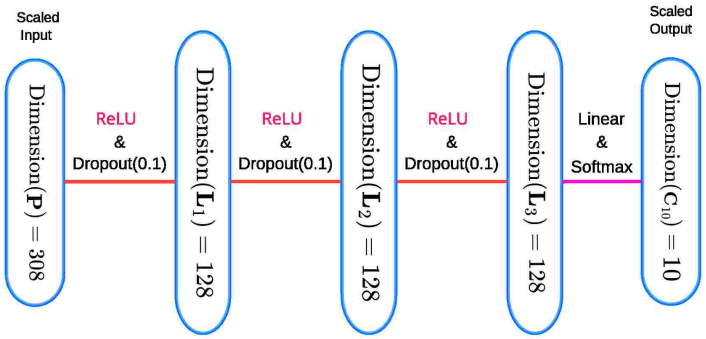


Figure 8: Pixel-wise prediction model used for all Experiments. ReLU activation functions, and Droptout (0.1) was used for the hidden layers.

4.2.3 Results

The best combination of hyperparameters, listed in Table 2, yielded a validation accuracy of 0.6844. Notably, r_{cut} exerted the greatest influence on accuracy, while σ_p performed best between 2 and 5. The parameters n_{max} and l_{max} had less impact, provided they were larger than approximately 6. The size of the pixel-wise SOAP spectra with the optimal parameters was $7(7+1)/2 \times (10+1) = 308$. The results are summarized in Table 1 and the confusion matrix on the test set is shown in Figure 9). Figure 10 presents scatter plots illustrating the relationships between the different hyperparameters (n_{max}, l_{max}, r_{cut}, and σ_p) and their effect on validation accuracy.

Table 1: Test values for the optimal hyper parameters found by Monte Carlo search using validation accuracy as benchmark. The parameters found are $r_{\text{cut}} = 63$, $n_{\text{max}} = 7$, $l_{\text{max}} = 10$, $\sigma_p = 3$. The validation accuracy yielded 0.6844 while test accuracy yielded 0.6863.

Accuracy	Recall	Precision	F1 Score
0.6863	0.6863	0.6821	0.6832

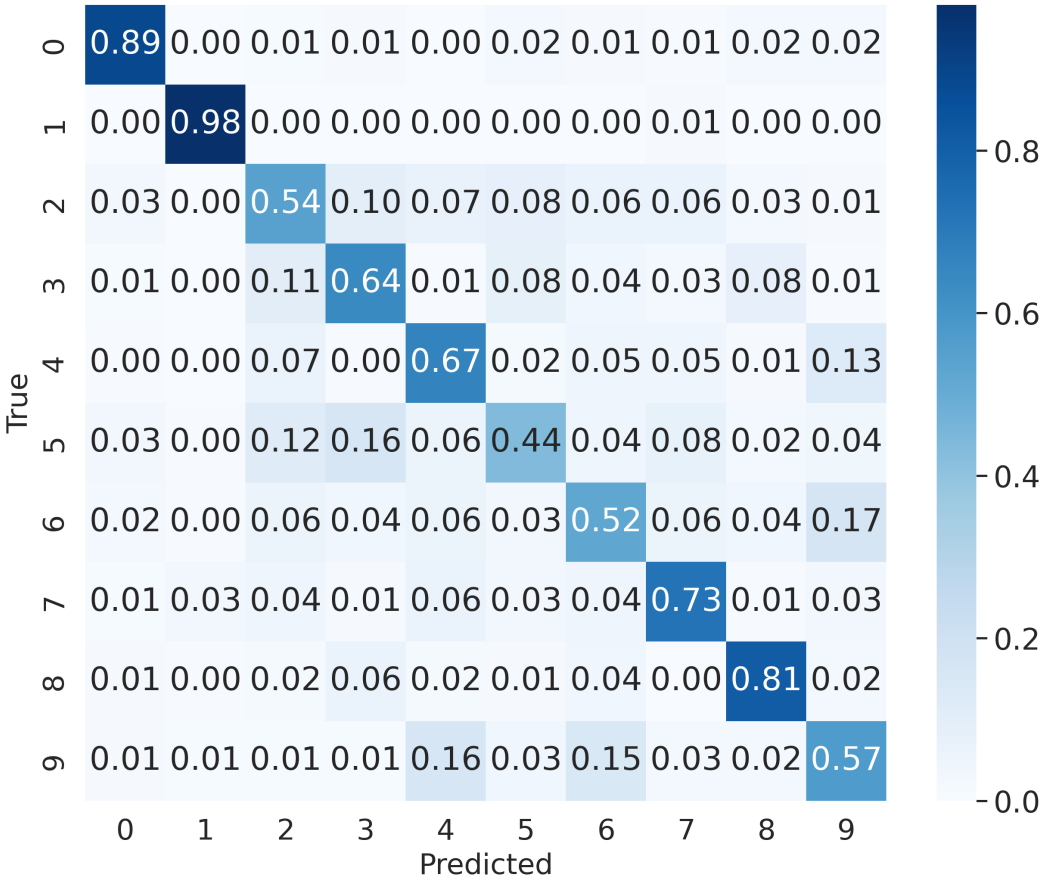
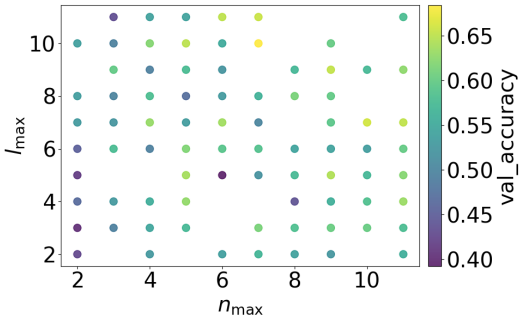
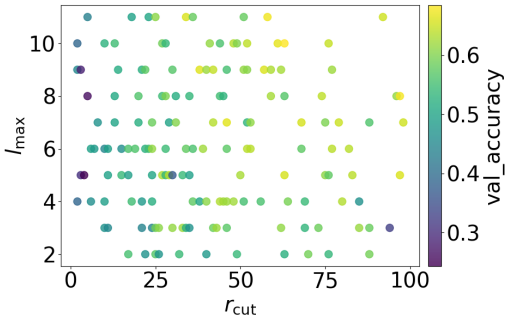


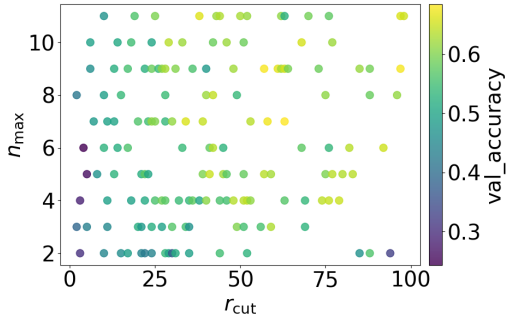
Figure 9: Confusion matrix of 10000 randomly selected test set from the MNIST handwritten data, normalized by each row. Accuracy: 0.6863, Recall: 0.6863, Precision: 0.6821, F1 Score: 0.6832. It can be seen for example, predictiong between 6 and 9 is particularly hard, because SOAP cannot distinguish between symmetries.



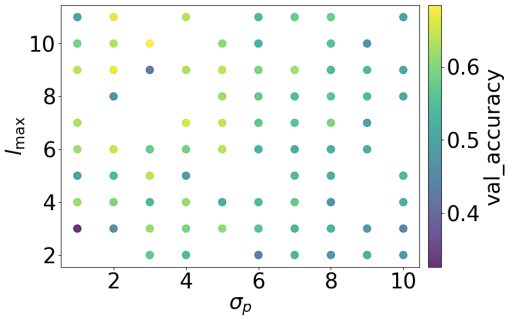
(a) Scatter plot of $n_{\max}$ vs $l_{\max}$ colored by validation accuracy.



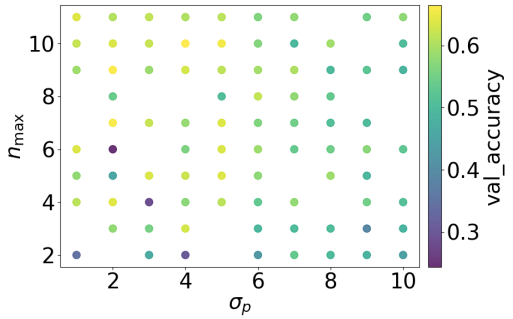
(b) Scatter plot of r_{cut} vs $l_{\max}$ colored by validation accuracy.



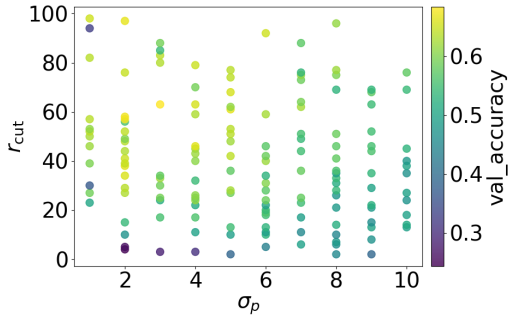
(c) Scatter plot of r_{cut} vs $n_{\max}$ colored by validation accuracy.



(d) Scatter plot of σ_p vs $l_{\max}$ colored by validation accuracy.



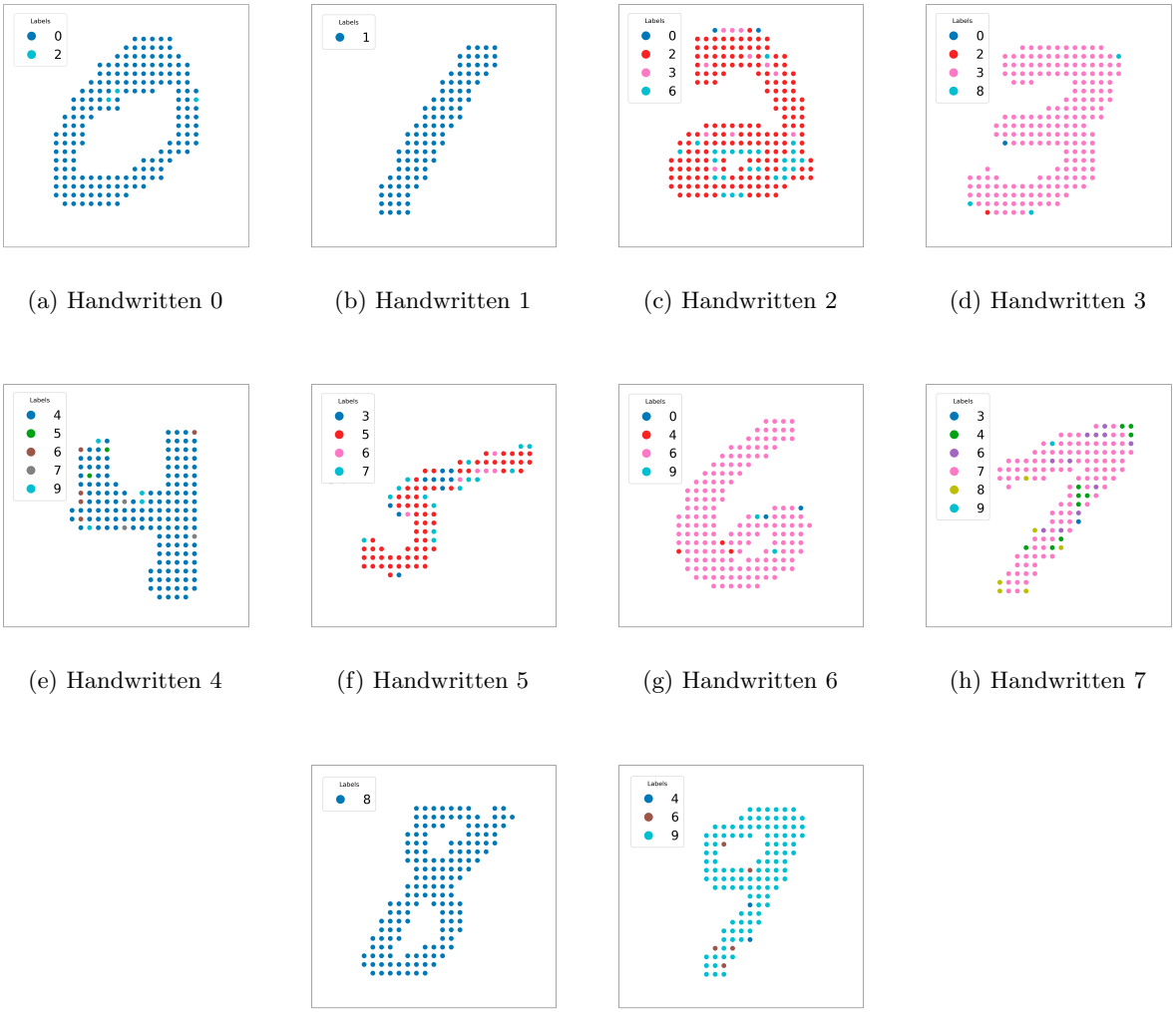
(e) Scatter plot of σ_p vs $n_{\max}$ colored by validation accuracy.



(f) Scatter plot of σ_p vs r_{cut} colored by validation accuracy.

Figure 10: Comparison of various scatter plots showing relationships between different parameters ($n_{\max}$, $l_{\max}$, r_{cut}, and σ_p) and their effect on validation accuracy. Each plot visualizes one pair of parameters, with color indicating the validation accuracy achieved. r_{cut} is the most important parameter, while σ_p tends to do well between 2 and 5.

Figure 11 shows examples of handwritten digits that are relatively easy to classify, indicating near-perfect predictions on clear and unambiguous shapes. These images highlight situations where SOAP-based features can successfully capture local environments without requiring additional data augmentation (e.g., rotation or flipping).



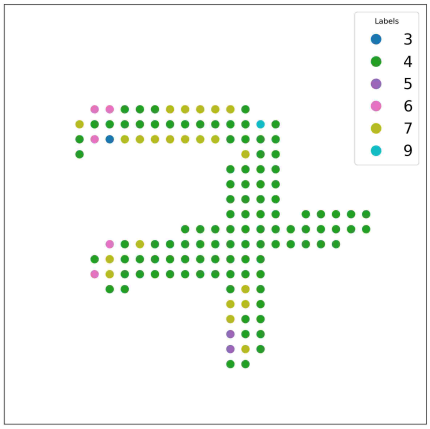
(a) Handwritten 0 (b) Handwritten 1 (c) Handwritten 2 (d) Handwritten 3

(e) Handwritten 4 (f) Handwritten 5 (g) Handwritten 6 (h) Handwritten 7

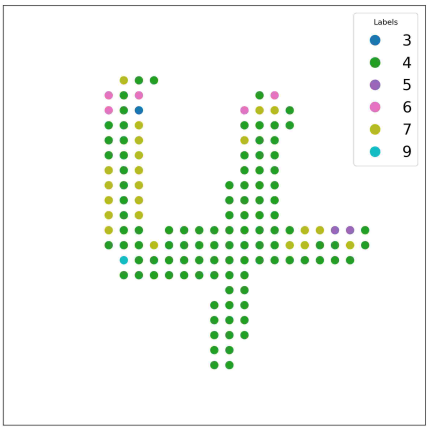
(i) Handwritten 8 (j) Handwritten 9

Figure 11: Predictions on the validation set for easy-to-classify shapes. For clear and unambiguous shapes, the model is very accurate.

Figure 12 demonstrates a challenging case where a handwritten 7 (subfigure a) can be rotated 90 degrees and mirror-flipped (subfigure b), causing the model to misclassify it as a 4. This misclassification arises because SOAP features do not inherently distinguish between these symmetries.



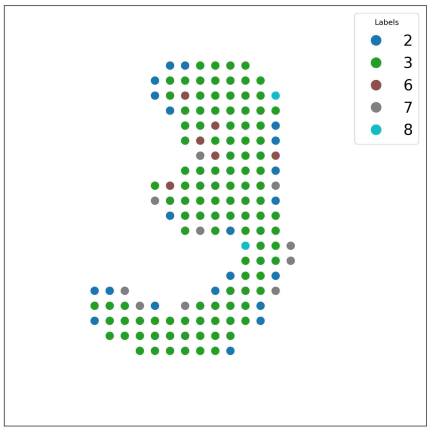
(a) Handwritten 7



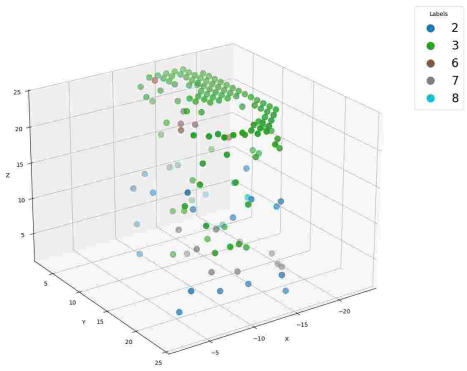
(b) The same image as (a) but rotate clockwise 90 degree and mirror flipped on xaxis.

Figure 12: Because SOAP cannot distinguish between certain symmetries, the model misclassifies the rotated and flipped 7 as a 4.

Figure 13 displays another example where points far from the handwritten shape (digit 3) tend to be predicted less accurately. These edge points do not strongly resemble any digit, indicating that SOAP features, while robust, still depend on local geometry and can produce errors on pixels far from the number's main structure.



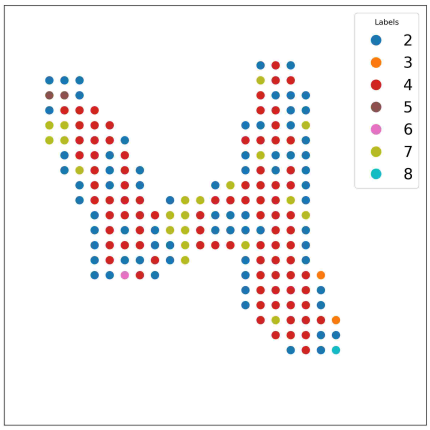
(a) Handwritten 3



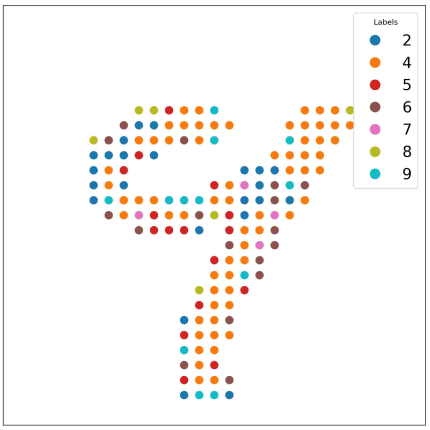
(b) 3D projection of (a)

Figure 13: (a) A simple handwritten 3. (b) A 3D projection of the pixel data, illustrating that points far from the primary shape are predicted less accurately.

Finally, Figure 14 shows ambiguous shapes of handwritten digits (e.g., a 4 and a 9) that can confuse not only the model but also human observers. In such cases, even the most sophisticated feature extraction approaches may fail if the digit is too ambiguous.



(a) Ambiguous Handwritten 4



(b) Ambiguous Handwritten 9

Figure 14: Examples of highly ambiguous handwritten digits (4 and 9). Even human observers may find these shapes confusing.

4.2.4 Discussion

The results underscore the importance of selecting an appropriate r_{cut} and ensuring σ_p lies in the range of 2–5 for improved accuracy. By leveraging SOAP features, our model does not require augmentation for training, such as rotation, translation, or mirror flipping. This is because SOAP naturally encodes local geometric information of each pixel or point in the handwritten digits.

However, the same property that makes SOAP robust against certain transformations also introduces challenges when symmetrical orientations are key to correct identification. For instance, as shown in Figure 12, a handwritten digit 7 rotated 90 degrees and mirror-flipped closely resembles a 4. Humans also tend to misinterpret it in such an orientation [27] , but in deep learning-based models without built-in symmetry handling, such misclassifications can be frequent. Moreover, SOAP struggles with highly ambiguous handwriting (see Figure 14), although this limitation is not unique to SOAP.

In summary, SOAP-based feature extraction presents a strong option for digit classification tasks, particularly for reducing the need for data augmentation. It is especially effective for clear, unambiguous shapes and for learning from relatively limited data. Yet, there is limitations for SOAP (like not being able to distinguish 6s and 9s some times due to rotational invariance), and additional strategies to account for orientation or symmetries may be required to further improve accuracy.

4.3 Experiment 2: SOAP Vector Compression and Impact on Prediction Accuracy

4.3.1 Objective

The high-dimensional nature of SOAP descriptors (308 dimensions in our optimal configuration) introduces computational challenges for downstream machine learning tasks. This experiment evaluates the compressibility of SOAP vectors by comparing three encoding methods—principal component analysis (PCA), linear autoencoding, and deep autoencoding—and quantifies the trade-off between compression ratio and reconstruction accuracy. We further analyze how compression impacts the performance of digit classification.

4.3.2 Methods

For this experiment, we use a subset of 120,000 SOAP descriptors from Experiment 1, which is divided into training (80%) and validation (20%) sets. The compression techniques considered include PCA, which performs linear dimensionality reduction via singular value decomposition; a linear autoencoder, implemented as a single-layer neural network with h_e hidden units and linear activation (see Figure 15); and a deep autoencoder, which employs a non-linear architecture with an encoder defined as $f_e : \mathbb{R}^{308} \to \mathbb{R}^{308h_m} \to \mathbb{R}^{h_e}$ and a decoder defined as $f_d : \mathbb{R}^{h_e} \to \mathbb{R}^{308h_m} \to \mathbb{R}^{308}$, where $h_m \in \{2, 4, 10\}$ controls the hidden layer capacity (see Figure 16). The evaluation metrics include the reconstruction loss, measured as the mean squared error (MSE) [28] between the original and reconstructed SOAP vectors, and the classification accuracy of Model A (from Experiment 1) when using the compressed features. All autoencoders are implemented using the Adam optimizer with a learning rate of 0.0001 and a batch size of 512, and they are trained for 10,000 epochs.

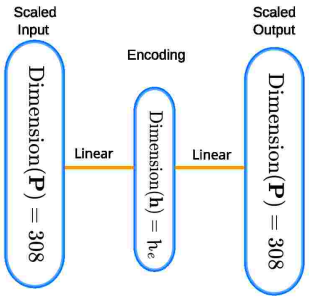


Figure 15: Our Linear Auto Encoder/Decoder Model

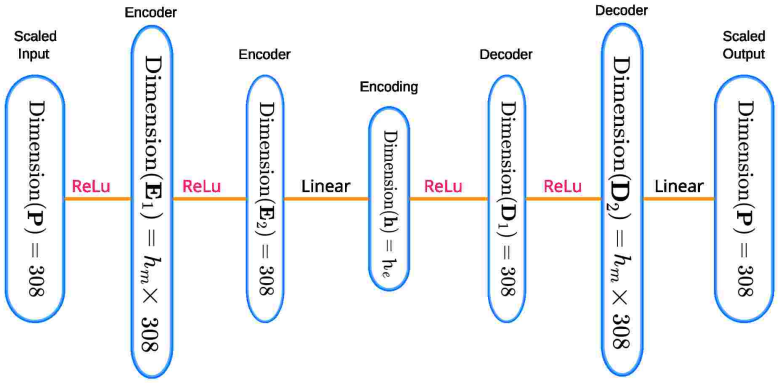


Figure 16: Our Deep Auto Encoder/Decoder Model

4.3.3 Results

Figure 17 reveals strong correlations between SOAP vector components, suggesting significant redundancy and motivating compression to eliminate redundant dimensions without sacrificing predictive power. Figure 18 shows the relationship between the encoding dimension h_e and the reconstruction loss, where both PCA and a linear autoencoder exhibit identical performance for $h_e < 200$, with PCA becoming superior at higher dimensions due to its optimal linear subspace identification, while a deep autoencoder outperforms linear methods for $h_e < 50$ by leveraging non-linear mappings to preserve information. Furthermore, Figure 19 demonstrates the impact of compression on classification accuracy: for high dimensions ($h_e > 50$), all methods achieve more than 95% of the baseline accuracy (308 dimensions), with PCA slightly outperforming autoencoders, whereas under aggressive compression ($h_e < 50$), test accuracy suddenly drops and deep autoencoding outperforms PCA.

4.3.4 Discussion

SOAP vectors exhibit substantial redundancy, enabling compression to approximately 100 dimensions (one-third of the original size) without any loss in accuracy. The key findings include computational efficiency—since principal component analysis (PCA) provides optimal compression for $h_e > 50$, requiring no training and minimal implementation effort—and performance in the high compression regime, where deep autoencoders outperform linear methods for $h_e < 50$, albeit at the cost of increased model complexity. Moreover, for MNIST classification, compressing to $h_e = 100$ results in nearly no loss in prediction accuracy (98% of the baseline). This analysis confirms that SOAP's rotational and translational invariance does not preclude efficient compression, and it indicates that the choice between linear and non-linear compression depends on the target dimensionality and acceptable accuracy trade-offs. Future work could explore hybrid approaches or task-specific compression to further optimize this balance.

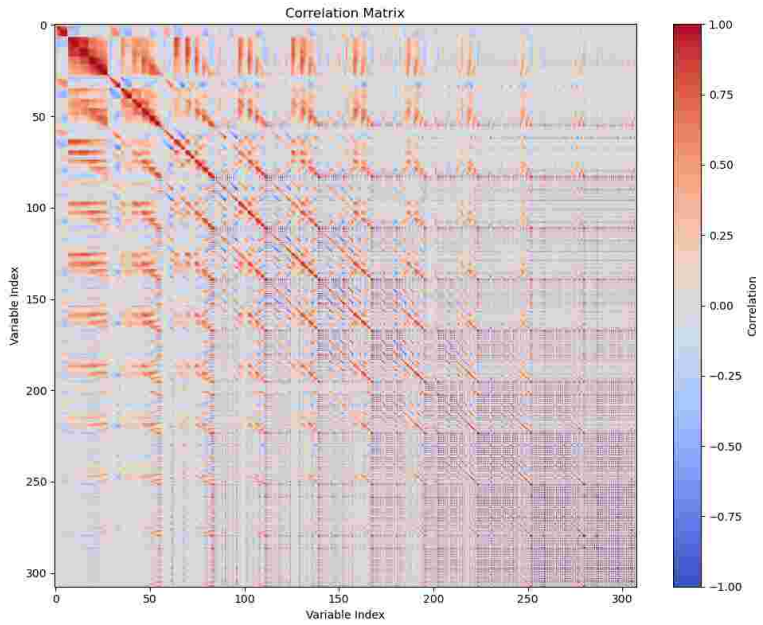
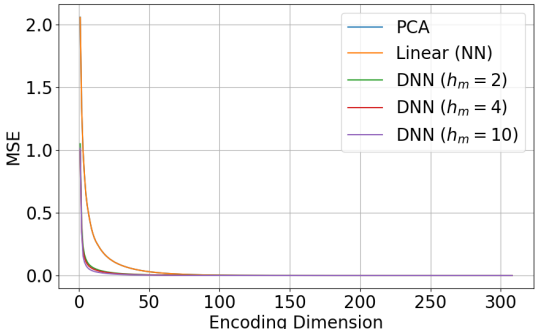
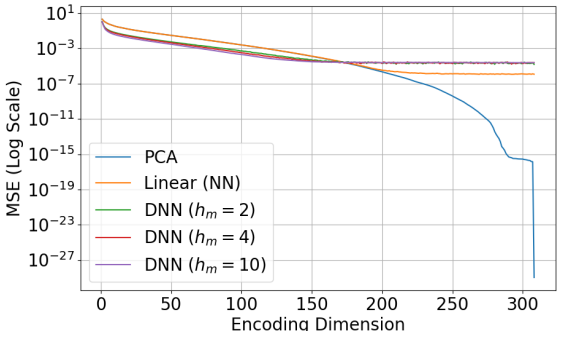


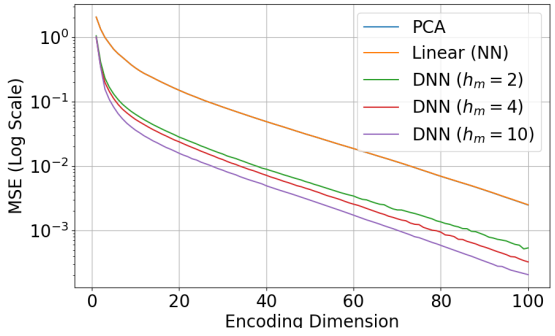
Figure 17: Correlation Matrix of the SOAP vectors for the 120,000 samples. Many of the elements are correlated, which suggests that it is highly compressible.



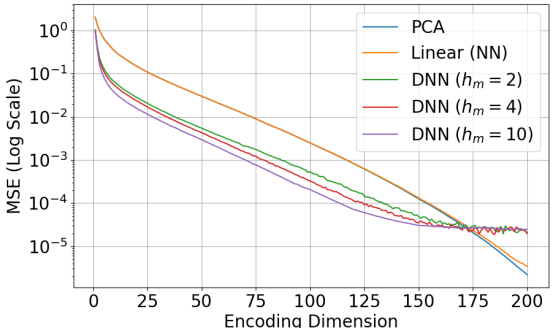
(a) Encoding MSE from 0 to 308, Linear-Scale.



(b) Encoding MSE from 0 to 308, Logarithmic-Scale.

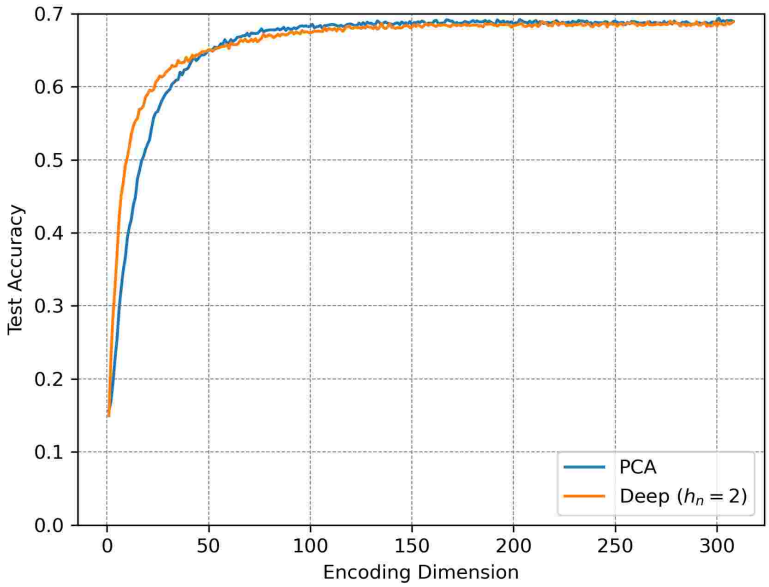


(c) Encoding MSE from 0 to 100, Logarithmic-Scale.

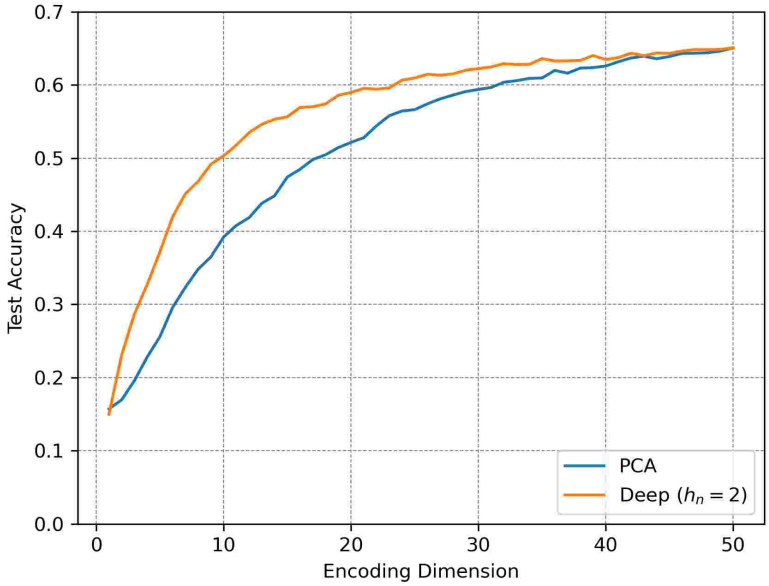


(d) Encoding MSE from 0 to 200, Logarithmic-Scale.

Figure 18: PCA dominates the MSE accuracy from 308 until around 200, then PCA and linear model become identical. Below around 175, Deep Autoencoding becomes more accurate, and there is not much difference between $h_m = 2, 4$ or 10.



(a) Test accuracy for compression dimension between 0 and 308 with Linear model (PCA) and Deep autoencoding ($h_m = 2$).



(b) Test accuracy for compression dimension between 0 and 50 with Linear model (PCA) and Deep autoencoding ($h_m = 2$).

Figure 19: The Linear Model and Deep Model both forform similarily, with PCA giving slightly better accuracy over $h_e = 50$, and Deep Autoencoding giving slightly better accuracy over $h_e = 50$.

4.4 Experiment 3: Robustness to Pixel Position Perturbations

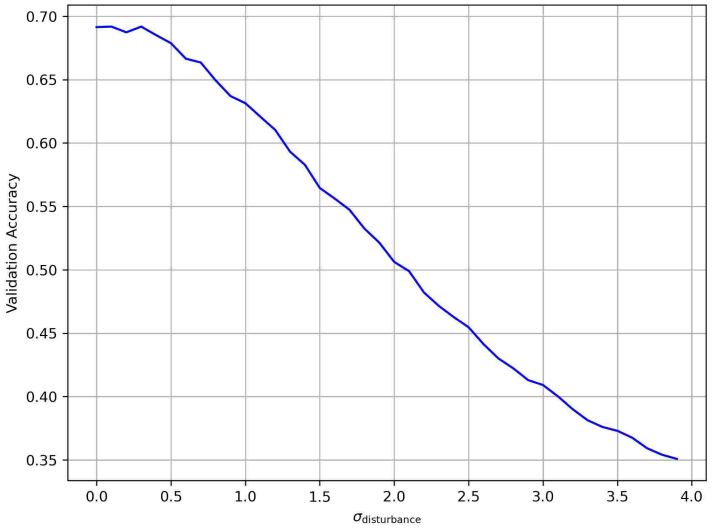


Figure 20: Validation Accuracy with noise.

4.4.1 Objective

To evaluate the robustness of SOAP-based feature extraction against noise, we introduce Gaussian perturbations to pixel positions and measure the impact on validation accuracy. This experiment tests whether the method gracefully degrades with increasing noise, thereby reflecting its stability in real-world scenarios with imperfect data.

4.4.2 Methods

In our approach, noise is injected into each image by perturbing the pixel coordinates (x, y) and the intensity-derived z-values with additive Gaussian noise according to the equation

$$\mathbf{r}_i' = \mathbf{r}_i + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2_{\text{disturbance}}\mathbf{I}),$$

where $\sigma_{\text{disturbance}}$ controls the noise magnitude (tested over a range from 0.1 to 5.0 in 20 logarithmic steps). The dataset consists of 10,000 MNIST test images converted to 3D structures with noise using the same SOAP parameters as in Experiment 1 ($r_{\text{cut}} = 63$, $n_{\text{max}} = 7$, $l_{\text{max}} = 10$, $\sigma_p = 3$), and the model employed is our three-layer prediction model (see Figure 8). The primary metric for evaluation is the validation accuracy as a function of $\sigma_{\text{disturbance}}$.

4.4.3 Results

The results, as shown in Figure 20, indicate that validation accuracy decreases smoothly with increasing $\sigma_{\text{disturbance}}$. At a noise level of $\sigma_{\text{disturbance}} = 1.0$, accuracy remains at 92% of the baseline (i.e., the case when $\sigma_{\text{disturbance}} = 0$), demonstrating robustness to moderate noise; however, performance drops to chance levels (approximately 51%) at $\sigma_{\text{disturbance}} = 3.9$, a point where local pixel neighborhoods are irrecoverably distorted. Additionally, a critical threshold is observed: accuracy declines sharply beyond $\sigma_{\text{disturbance}} = 0.5$.

4.4.4 Discussion

These findings demonstrate that SOAP-based features exhibit gradual performance degradation under controlled noise, confirming their stability for practical applications. The smooth decline in accuracy, rather than a catastrophic failure, validates the method's suitability for scenarios with noisy data and positional uncertainty, and suggests that future work could couple SOAP with denoising techniques to further enhance robustness.

5 Future Work

While this study has demonstrated the potential of SOAP-based descriptors for pixel-wise classification as a benchmark, several extensions and improvements can be explored in future work. One intriguing direction is the adaptation of SOAP for RGB images rather than grayscale. Since SOAP includes species as a hyperparameter, different channels of an RGB image could be encoded using distinct species. For instance, one could draw an analogy by assigning the

red, green, and blue channels to chemical species such as hydrogen (H), helium (He), and lithium (Li), respectively. This approach may introduce a richer feature space by allowing inter-channel interactions to be represented in a way similar to multi-species atomic environments.

Another key limitation of SOAP is its inherent invariance to symmetry transformations, which may discard crucial orientation-dependent information. To address this, a strategy of *forced symmetry breaking* could be employed. One possible method is to introduce auxiliary points near each pixel, such as a structured line below a handwritten digit, to provide directional context. This additional information could help encode spatial orientation, enabling the descriptors to retain some asymmetry where needed.

Beyond pixel-wise classification, future work could explore leveraging SOAP vectors to construct global representations for entire images. For example, one could compute an aggregate representation by averaging SOAP vectors across all pixels in an image, creating a holistic descriptor that remains invariant yet captures key structural patterns. Alternatively, more sophisticated approaches such as graph neural networks could be applied to learn higher-order relationships between SOAP descriptors, potentially enhancing performance in global classification tasks.

Additionally, in this study, we utilized the SOAP power spectrum, which provides a robust yet relatively compact representation of local environments. However, SOAP also offers a more expressive addition known as the bispectrum, which retains higher-order structural correlations and can encode more intricate geometric details. Future work could investigate whether incorporating the SOAP bispectrum leads to improved classification performance, particularly in tasks where capturing finer structural nuances is critical.

Finally, another promising avenue is the direct application of SOAP-based descriptors to point cloud classification tasks. Given that SOAP was originally designed for atomic-scale modeling, its extension to three-dimensional point clouds in computer vision could be a natural progression. This could involve adapting SOAP to tasks such as 3D object recognition, scene reconstruction, or LiDAR data analysis, where local geometric structures play a crucial role in classification.

These directions illustrate the versatility of SOAP-based feature extraction and open up exciting possibilities for extending its applications beyond grayscale image classification to more complex and structured data representations.

6 Conclusions

In this work, we have demonstrated how the Smooth Overlap of Atomic Positions (SOAP), originally developed for atomic-scale modeling in chemistry and materials science, can be adapted to extract pixel-wise descriptors for images. By viewing each pixel as a local "environment" and lifting 2D image data into 3D space, we obtain SOAP vectors that capture rich local structure while maintaining invariance to translation, rotation, and mirror symmetry. One of the primary strengths of this method is that it obviates the need for extensive data augmentation for these transformations, allowing us to train models effectively without having to create or include rotated, translated, or mirrored variants of the input images. However, if mirror-flipping information (or other orientation-dependent cues) is intrinsically relevant to the classification task, SOAP's invariant nature can become a limitation, since it effectively discards such distinguishing orientation-specific features.

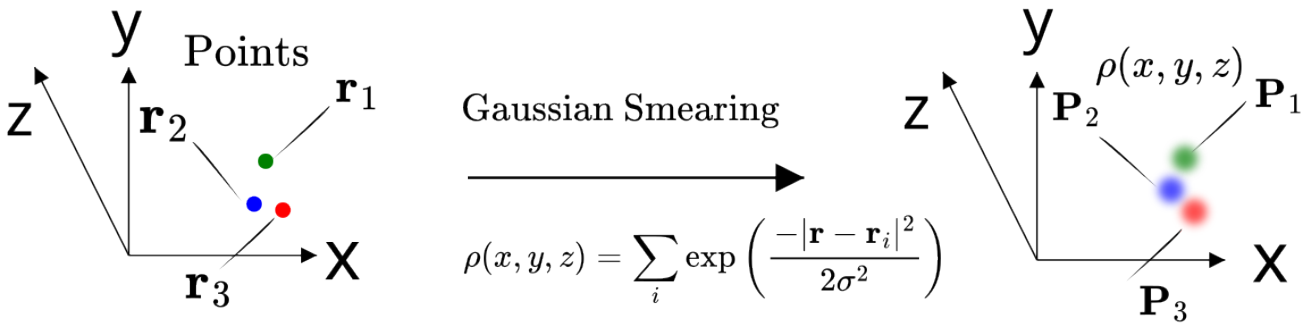
Our experiments on MNIST show that careful tuning of SOAP hyperparameters, especially the cutoff radius, is critical for optimal classification performance. Furthermore, we have illustrated the high compressibility of SOAP features via PCA and autoencoders, reducing dimensionality without significantly degrading predictive accuracy. We also investigated the robustness of SOAP-based descriptors to positional noise. Perturbing the pixel coordinates with Gaussian noise revealed a smooth decline in accuracy, confirming that SOAP gracefully handles moderate spatial uncertainties. This resilience is valuable for real-world datasets where image acquisition or labeling may be imperfect.

A major strength of this approach is its general applicability to any set of data points that can be projected into 3D space. Beyond images, the same pipeline can be readily applied to diverse domains such as 3D object recognition, geospatial data analysis, or even higher-dimensional biomedical images where pixel or voxel intensities can be mapped into spatial coordinates. By combining inherent invariance, robust local feature encoding, and flexible dimensionality reduction, SOAP-based descriptors provide a powerful framework for learning tasks that rely on capturing local patterns in a manner invariant to common image transformations. The results presented here open a promising avenue for future work in computer vision and related fields, where the capacity to incorporate sophisticated descriptors from quantum chemistry can lead to robust, efficient, and interpretable representations.

A

Example of C's

In this appendix, we will derive the close form of SOAP with Spherical Harmonics as spherical basis functions, and Gaussian Orbital Type (GTO) functions as radial basis functions [24] (See Figure 21).



$$\mathbf{P}_1 = \mathrm{SOAP}(\rho(x,y,z), \mathbf{r}_1)$$
$$\mathbf{P}_2 = \mathrm{SOAP}(\rho(x,y,z), \mathbf{r}_2)$$
$$\mathbf{P}_3 = \mathrm{SOAP}(\rho(x,y,z), \mathbf{r}_3)$$

Figure 21: The SOAP algorithm takes a Gaussian-smeared representation of points and computes a rotational, translational, and mirror-symmetric invariant vector $\mathbf{P}_i$ (SOAP power spectrum) at a given reference point, typically located at an existing data point. This SOAP vector encodes the structural environment surrounding the reference point.

Spherical Harnmonics is defined as:

$$Y_l^m(\theta,\phi) = (-1)^m \sqrt{\frac{(2l+1)}{4\pi}\frac{(l-m)!}{(l+m)!}}\, P_l^m(\cos\theta) e^{im\phi}, \tag{6}$$

where $-l \leq m \leq l$ and $P_l^m(x)$ is the associated Legendre polynomials.

GTO basis function is defined as:

$$g_{nl}(r) = \sum_{b=1}^{N_b} \beta_{lbn} r^l e^{-\alpha_{bl} r^2}, \tag{7}$$

where α_{bl}s are hyper parameters that need to be designed, and β_{lbn}s are arthonomalization constants, which cab be obtained as:

$$\beta_{lnn'} = \mathbf{S}_l^{-1/2}, \tag{8}$$

where

$$S_{lnn'} = \int_0^\infty r^{2l} e^{-(\alpha_{ln}+\alpha_{ln'})r^2}\, dr \tag{9}$$

$$= \frac{1}{2}(\alpha_{ln}+\alpha_{ln'})^{-(2l+1)/2}\Gamma\left(\frac{2l+1}{2}\right), \tag{10}$$

is the overlap matrix, where Γ is the gamma function.

By using the density function:

$$\rho(\mathbf{r}) = \sum_{p=1}^{N_p} e^{-\frac{|\mathbf{r}-\mathbf{R}_p|^2}{2\sigma_p^2}}, \tag{11}$$

where $R_p = \sqrt{x_p^2+y_p^2+z_p^2}$, we can get a closed form of coefficents in Eq.(4) by integration:

$$c_{nlm} = \lambda_{lm}(-1)^m \sqrt{2\pi\sigma_p^2}^3 \sum_{b=1}^{N_b} \frac{\beta_{lbn}}{\sqrt{1+2\alpha_{lb}\sigma_p^2}^{2l+3}} \sum_{p=1}^{N_p} e^{\frac{-\alpha_{lb}}{1+2\alpha_{lb}\sigma_p^2}R_p^2} (x_p+iy_p)^m R_p^{l-m} \sum_{k=m}^{l} \xi_{lmk} z_p^{k-m} R_p^{m-k}, \tag{12}$$

where

$$\lambda_{lm} = 2^l \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}}, \tag{13}$$

$$\xi_{lmk} = \frac{\frac{l+k-1}{2}!}{(k-m)!(l-k)!(\frac{l+k-1}{2}-l)!}. \tag{14}$$

$$\tag{15}$$

When $l + k =$ even, $\xi_{lmk} = 0$. α_{bl}'s are a hyper parameter that depend on the r_{cut} and a design choice. How the parameters are chose in the Dscribe [6, 24] package is shown in algorithm 4 (Alternatively, there are other packages that use different basis functions such as QUIP [30, 7]).

Algorithm 4 GetBasisGTO (r_{cut}, $n_{\max}$, $l_{\max}$)

Require:
- $r_{\text{cut}} \in \mathbb{R}^+$: The radial cutoff distance.
- $n_{\max} \in \mathbb{N}$: The number of GTO radial basis functions.
- $l_{\max} \in \mathbb{N}$: The maximum angular momentum quantum number.

Ensure:
- $\{\alpha_{l,i}\}$: A $(l_{\max}+1) \times n_{\max}$ array of radial decay exponents.
- $\{\beta_{l,i,j}\}$: A $(l_{\max}+1) \times n_{\max} \times n_{\max}$ array of Löwdin-orthonormalization factors.

function GetBasisGTO(r_{cut}, $n_{\max}$, $l_{\max}$)
  threshold $\leftarrow 10^{-3}$ ▷ Fixed decay threshold for the Gaussian functions.
  Initialize the array $\{a_i\}_{i=1}^{n_{\max}}$
  for $i \leftarrow 1$ **to** $n_{\max}$ **do**
    $a_i \leftarrow 1 + \dfrac{(i-1)\,(r_{\text{cut}}-1)}{n_{\max}-1}$ ▷ Equally spaced radial points from 1 to r_{cut}.
  end for
  Initialize $\alpha_{l,i} \leftarrow 0$ for $l = 0, \ldots, l_{\max}$ and $i = 1, \ldots, n_{\max}$
  Initialize $\beta_{l,i,j} \leftarrow 0$ for $l = 0, \ldots, l_{\max}$ and $i, j = 1, \ldots, n_{\max}$
  for $l \leftarrow 0$ **to** $l_{\max}$ **do**
    for $i \leftarrow 1$ **to** $n_{\max}$ **do**
      $\alpha_{l,i} \leftarrow -\dfrac{\ln\left(\dfrac{\text{threshold}}{a_i^l}\right)}{a_i^2}$ ▷ Choose $\alpha_{l,i}$ so that

$$a_i^l \exp\left(-\alpha_{l,i}\, a_i^2\right) = \text{threshold}\,.$$

    end for
    Initialize the matrix $M_{i,j}$ for $i, j = 1, \ldots, n_{\max}$
    for $i \leftarrow 1$ **to** $n_{\max}$ **do**
      for $j \leftarrow 1$ **to** $n_{\max}$ **do**
        $M_{i,j} \leftarrow \alpha_{l,i} + \alpha_{l,j}$
      end for
    end for
    Initialize the matrix $S_{i,j}$ for $i, j = 1, \ldots, n_{\max}$
    for $i \leftarrow 1$ **to** $n_{\max}$ **do**
      for $j \leftarrow 1$ **to** $n_{\max}$ **do**

$$S_{i,j} \leftarrow 0.5\,\Gamma\left(l + \frac{3}{2}\right)\left(M_{i,j}\right)^{-\left(l+\frac{3}{2}\right)}.$$

      end for
    end for
    Compute the inverse S^{-1} of S ▷ Use any standard matrix inversion algorithm.
    Compute $\beta^{\text{temp}} \leftarrow \sqrt{S^{-1}}$ ▷ This denotes the matrix square root of S^{-1} (Löwdin orthonormalization).
    if any entry of β^{temp} is complex **then**
      raise an error: "Could not calculate real-valued normalization factors."
    end if
    for $i \leftarrow 1$ **to** $n_{\max}$ **do**
      for $j \leftarrow 1$ **to** $n_{\max}$ **do**
        $\beta_{l,i,j} \leftarrow \beta_{i,j}^{\text{temp}}$
      end for
    end for
  end for
  return $\{\alpha_{l,i}\}$, $\{\beta_{l,i,j}\}$
end function

A

Table of Variables

Table 2: List of Variables for SOAP Descriptor Computation

Variable	Type/Dimension	Description
$\{\mathbf{F}_k\}_{k=1}^n$	Collection of $N_k \times 3$ matrices	3D structure.
$\mathbf{F}_k$	$N_k \times 3$ matrix	The k-th 3D structure containing coordinates (x, y, z) of each 3D-pixel.
$\{ r_{\text{cut}}, n_{\max}, l_{\max}, \sigma_p \}$	Scalars	Parameters defining the SOAP descriptor computation.
$\mathcal{P}_k$	$N_k \times d$ matrix	SOAP descriptors for each 3D-pixel in the k-th structure.
$\mathbf{P}_o$	$1 \times d$ vector	SOAP descriptor for the o-th 3D-pixel in structure $\mathbf{F}_k$.
k	Integer	Index for iterating over each structure ($1 \leq k \leq n$).
o	Integer	Index for iterating over each 3D-pixel within a structure ($1 \leq o \leq N_k$).
$\{\mathcal{P}_k\}_{k=1}^n$	Collection of $N_k \times d$ matrix	Output set of SOAP descriptors for all structures and their 3D structure.

Table 3: Variables for SOAP Extraction with Rescaling

Variable	Type/Dim	Description
$\mathbf{X}$	$T \times d$	Final collection of extracted and rescaled SOAP descriptors, where in our case $T = 1.2 \times 10^5$ for the training data and validation data, and $T = 1 \times 10^4$ for the test data.
$\mathbf{y}$	T	Labels for each row of $\mathbf{X}$.
$\mathbf{P}_t$	$1 \times d$	A single descriptor randomly chosen from $\mathcal{P}_r$.
$\mathbf{s}_{RR}$	p	Robust Rescale Parameters for later use.

References

[1] Quiroga, F.; Ronchetti, F.; Lanzarini, L.; Bariviera, A. F. *Revisiting data augmentation for rotational invariance in convolutional neural networks*. In *Modelling and Simulation in Management Sciences: Proceedings of the International Conference on Modelling and Simulation in Management Sciences (MS-18)*, Springer: 2020; pp. 127–141.

[2] Maharana, K.; Mondal, S.; Nemade, B. *A review: Data pre-processing and data augmentation techniques. Global Transitions Proceedings* **2022**, *3*, 91–99.

[3] Omae, Y.; Saito, Y.; Fukamachi, D.; Nagashima, K.; Okumura, Y.; Toyotani, J. *Impact of chest radiograph image size and augmentation on estimating pulmonary artery wedge pressure by regression convolutional neural network*. In *AIP Conference Proceedings*, AIP Publishing: 2023; *2872*, 1.

[4] Yoo, J.; Kang, S. *Class-adaptive data augmentation for image classification. IEEE Access* **2023**, *11*, 26393–26402.

[5] Bartók, A. P.; Kondor, R.; Csányi, G. *On representing chemical environments. Physical Review B—Condensed Matter and Materials Physics* **2013**, *87*, 184115.

[6] Himanen, L.; Jäger, M. O. J.; Morooka, E. V.; Canova, F. F.; Ranawat, Y. S.; Gao, D. Z.; Rinke, P.; Foster, A. S. *DScribe: Library of descriptors for machine learning in materials science. Computer Physics Communications* **2020**, *247*, 106949.

[7] Caro, M. A. *Optimizing many-body atomic descriptors for enhanced computational performance of machine learning based interatomic potentials. Physical Review B* **2019**, *100*, 024112.

[8] Jäger, M. O. J.; Morooka, E. V.; Federici Canova, F.; Himanen, L.; Foster, A. S. *Machine learning hydrogen adsorption on nanoclusters through structural descriptors. npj Computational Materials* **2018**, *4*, 37.

[9] De, S.; Bartók, A. P.; Csányi, G.; Ceriotti, M. *Comparing molecules and solids across structural and alchemical space. Physical Chemistry Chemical Physics* **2016**, *18*, 13754–13769.

[10] Caruso, C.; Cardellini, A.; Crippa, M.; Rapetti, D.; Pavan, G. M. *TimeSOAP: Tracking high-dimensional fluctuations in complex molecular systems via time variations of SOAP spectra. The Journal of Chemical Physics* **2023**, *158*, 21.

[11] LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. *Gradient-based learning applied to document recognition. Proceedings of the IEEE* **1998**, *86*, 2278–2324.

[12] Gewers, F. L.; Ferreira, G. R.; Arruda, H. F. D.; Silva, F. N.; Comin, C. H.; Amancio, D. R.; Costa, L. F. *Principal component analysis: A natural approach to data exploration. ACM Computing Surveys (CSUR)* **2021**, *54*, 1–34.

[13] Berahmand, K.; Daneshfar, F.; Salehi, E. S.; Li, Y.; Xu, Y. *Autoencoders and their applications in machine learning: A survey. Artificial Intelligence Review* **2024**, *57*, 28.

[14] Malik, J. S.; Hemani, A. *Gaussian random number generation: A survey on hardware architectures. ACM Computing Surveys (CSUR)* **2016**, *49*, 1–37.

[15] Tanner, M. A.; Wong, W. H. *The calculation of posterior distributions by data augmentation. Journal of the American Statistical Association* **1987**, *82*, 528–540.

[16] Wei, L. *Empirical Bayes test of regression coefficient in a multiple linear regression model. Acta Mathematicae Applicatae Sinica* **1990**, *6*, 251–262.

[17] Simard, P. Y.; Steinkraus, D.; Platt, J. C. *Best practices for convolutional neural networks applied to visual document analysis.* In *Proceedings of ICDAR*; Edinburgh: 2003; *3*, 2003.

[18] Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; Kegelmeyer, W. P. *SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research* **2002**, *16*, 321–357.

[19] Elreedy, D.; Atiya, A. F. *A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance. Information Sciences* **2019**, *505*, 32–64.

[20] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. *Generative adversarial nets. Advances in Neural Information Processing Systems* **2014**, *27*.

[21] Bao, G.; Yan, B.; Tong, L.; Shu, J.; Wang, L.; Yang, K.; Zeng, Y. *Data augmentation for EEG-based emotion recognition using generative adversarial networks. Frontiers in Computational Neuroscience* **2021**, *15*, 723843.

[22] Chen, L.; Li, Y.; Deng, X.; Liu, Z.; Lv, M.; Zhang, H. *Dual auto-encoder GAN-based anomaly detection for industrial control system. Applied Sciences* **2022**, *12*, 4986.

[23] Ronneberger, O.; Fischer, P.; Brox, T. *U-Net: Convolutional networks for biomedical image segmentation.* In *Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2015)*, Springer: Munich, Germany, 2015; pp. 234–241.

[24] Laakso, J.; Himanen, L.; Homm, H.; Morooka, E. V.; Jäger, M. O. J.; Todorović, M.; Rinke, P. *Updates to the DScribe library: New descriptors and derivatives. The Journal of Chemical Physics* **2023**, *158*.

[25] Shapiro, A. *Monte Carlo sampling methods. Handbooks in Operations Research and Management Science* **2003**, *10*, 353–425.

[26] Kingma, D. P. *Adam: A method for stochastic optimization. arXiv preprint* **2014**, arXiv:1412.6980.

[27] Kumar, V. *Pruning Distorted Images in MNIST Handwritten Digits. arXiv preprint* **2023**, arXiv:2307.14343.

[28] Hodson, T. O.; Over, T. M.; Foks, S. S. *Mean squared error, deconstructed. Journal of Advances in Modeling Earth Systems* **2021**, *13*, e2021MS002681.

[29] Zenodo Dataset. Available: https:zenodo.orgrecords14916887

[30] Klawohn, S.; Darby, J. P.; Kermode, J. R.; Csányi, G.; Caro, M. A.; Bartók, A. P. *Gaussian approximation potentials: Theory, software implementation and application examples. The Journal of Chemical Physics* **2023**, *159*.