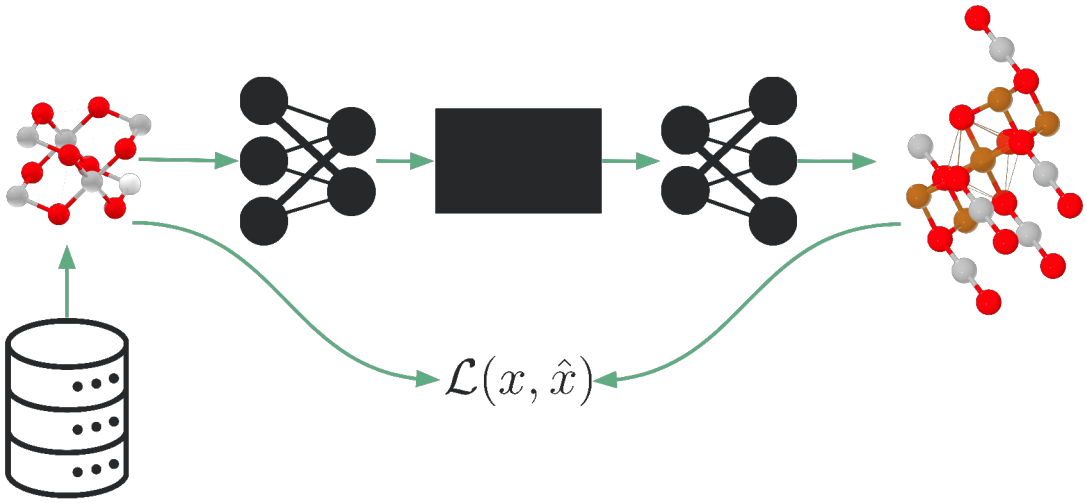
CHALMERS
UNIVERSITY OF TECHNOLOGY



Updating Paulings rules using a machine learning approach

Analysing the latent space of autoencoders to discover new rules for oxide stability

Masters Degree project report in Physics

Pontus Gustafsson

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Degree project report 2026

Updating Paulings rules using a machine learning approach

Analysing the latent space of autoencoders to discover new rules for oxide stability

Pontus Gustafsson

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
Chalmers University of Technology
Gothenburg, Sweden 2026

Updating Paulings rules using a machine learning approach
Analysing the latent space of autoencoders to discover new rules for oxide stability
Pontus Gustafsson

Supervisor: Henrik Klein Moberg, Department of Chemical Physics
Examiner: Anders Hellman, Department of Chemical Physics

Degree project report 2026
Department of Chemical Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: Schematic image of an Autoencoder during training using crystallographic data.

Typeset in LaTeX
Gothenburg, Sweden 2026

Updating Paulings rules using a machine learning approach
Analysing the latent space of autoencoders to discover new rules for oxide stability
Pontus Gustafsson
Department of Physics, Division of Chemical Physics
Chalmers University of Technology

Abstract

Oxides are an important family of materials that have an extremely wide range of applications in for example semiconductors, pigments and catalysis. It is therefore important to have a solid understanding of these ubiquitous materials. In 1929 Linus Pauling proposed five rules for oxide stability that are widely used. These rules are however not good enough to describe oxide stability as only a fraction of stable oxides fulfil them. In this project a machine learning approach was used to attempt to find better rules based on the composition of oxides. This was done by training a set of autoencoders and analysing the latent spaces of these models by sampling new compositions from the models. Three different autoencoders were trained and based on the results, three new rules of thumb are proposed; Oxides containing only reactive non-metals are in general unstable, metals favour stability and heavier cations favour stability.

Keywords: Paulings rules, oxide stability, machine learning, autoencoder, Wasserstein autoencoder, latent space

Acknowledgements

This project has taught me a lot, not just about machine learning, autoencoders and oxides but also about the research process and my own way of working.

Thanks to Anu Peter for our many conversations and brainstorming sessions. You made my time working on this project a lot more fun.

Thanks to Henrik Klein Moberg for your guidance, for answering all of my (many) questions and for teaching me how to take a step back and focus on the important things rather than getting stuck at details. Your help made the process a lot more enjoyable and I always left our meetings with new insights and renewed energy.

Finally, thanks to Anders Hellman for giving me the change to peer into the world of research, for your patience during my rants, your insights and your mentoring. I couldn't have done it without you.

The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

Pontus Gustafsson, Gothenburg, January 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AE	Autoencoder
DDWAE	Double Decoder Autoencoder
EAH	Energy Above Hull
ELBO	Estimated Lower Bound
FID	Fréchet Inception Distance
GAN	Generative Adversarial Network
IMQ	Inverse Multiquadratics kernel
MAE	Mean Absolute Error
MMD	Maximum Mean Discrepancy
PC	Principal Component
PCA	Principal Component Analysis
t-SNE	t-distributed Stochastic Neighbour Embedding
WAE	Wasserstein Autoencoder

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Sets

$\mathcal{X}$	Set of input data
$\mathcal{Z}$	Set of latent space points

Parameters

$N_{\text{atoms}}^{\text{max}}$	Maximum number of atoms allowed in the unit cell for all materials
N_{Epochs}	Number of training epochs
γ	Optimizer learning rate
d_z	Latent space dimensionality
σ_z	Variance of the assumed latent space prior
λ_{MMD}	Weight coefficient for the MMD objective term
N_{Batch}	Batch size for training and validation data

Variables

ΔE_d	Energy above hull
N_{atoms}	Total number of atoms in the unit cell of a material
N_{oxygen}	Number of oxygen atoms in the unit cell of a material
N_{other}	Number of non oxygen atoms in the unit cell of a material
$\vec{C}$	Composition vector
Z	Element number
$\mathcal{L}_{\text{comp}}$	Composition objective loss

$\mathcal{L}_{\text{EAH}}$	Stability objective loss
$\mathcal{L}_{\text{MMD}}$	MMD objective loss

Contents

List of Figures

List of Tables

1
Introduction

Oxides are a class of materials found everywhere with a wide range of applications. Oxides are used as support structures for nanoparticles in catalytic applications, in microelectronics oxides are used as dielectrics or insulators and oxides are used as a pigments, for example Falu red paint gets its classic red color from Fe_2O_3.
Since oxides are everywhere, a solid understanding of the underlying physics governing this vast class of materials is very important.

1.1 Background

Pauling's rules are five, relatively simple, rules that aim to predict the stability of oxides and have been widely used for a long time. However, the rules have a very limited capacity to actually predict the stability of oxides. As found by George et. al. [4], only 20% in a dataset of 5000 oxides fulfil 4 out of 5 rules, and even fewer, 13%, fulfil all five rules.
Machine learning as a tool in physics has exploded in recent years. In materials science, machine learning has mainly been used to create machine learning potentials that greatly reduce computational costs. These systems are, however, largely black box systems.
Autoencoders are a class of generative machine learning architectures that are trained by compressing and decompressing data into and out of a so called latent space. This allows for analysis of the system by analysing the latent space, and if trained correctly, these systems learn patterns in the data due to the forced compression.
In this project, we attempted to leverage the Wasserstein Autoencoder to extract such patterns for oxide stability with the goal of discovering new rules in a similar fashion to Pauling's rules based on the atomic composition of crystalline oxides.

1.2 Research questions

1. Is the latent space structured such that examining a specific feature like stability is feasible?
2. Can new rules of stability be formulated only by examining the latent space of an autoencoder?

1.3 Scope and limitations

Although both the structure and composition are needed to fully describe materials, to limit the scope of the project, the autoencoders used were only trained using oxide compositions.
This reduced model complexity and resulted in more interpretable results when sampling and examining the latent space.

2

Theory

2.1 Pauling's rules

Proposed by Linus Pauling in 1929, the rules aim to predict the stability of oxides. In brief, the rules are:

2.1.1 The radius ratio rule

In ionic compounds such as oxides, the positively charged cations tend to be smaller than the negatively charged anions. Each cation is surrounded by a number of anions forming a polyhedron depending on the coordination number of the cation. Treating the atoms as rigid spheres, the cation must be touching the anions for the oxide to be stable. From geometry, this constraint means that the cation-anion radius ratio $\frac{r_+}{r_-}$ must be large enough.

2.1.2 The electrostatic valence rule

Each cation in a crystal can be considered electrostatically bonded to the coordinated anions that surround it. Defining the electrostatic bond strength to be $s = \frac{z}{\nu}$, with z the cation charge and ν the cation coordination number, the second rule states that for the local electroneutrality to be preserved, the sum over adjacent cations for an anion $\xi = \sum_i s_i$ should equal the charge of the anion.

2.1.3 Third rule; sharing of corners, edges, and faces

The third rule states that two anion polyhedra sharing edges or faces have a negative effect on the stability of the oxide, whilst the sharing of corners does not decrease the stability in a major way. This is due to the larger repulsion between cations in edge-sharing or face-sharing polyhedra as compared to corner-sharing polyhedra.

2.1.4 Fourth rule; crystals with different cations

If a crystal contains different types of cations, the cations with high valency and small coordination numbers are unlikely to share corners, faces or edges with each other. This means that cations with large charges get spread out, reducing repulsion between them.

2.1.5 Fifth rule; the rule of parsimony

The fifth rule states that the number of different cation polyhedra in a crystal tends to be small. This is due to each atom in the crystal being most stable in a specific local environment.

2.2 Measuring stability in oxides

Inorganic solids are defined by composition, which includes the elements in the solid, as well as its structure. One way to measure stability in oxides is using the so-called Energy Above Hull (EAH) [1], denoted ΔE_d. ΔE_d is calculated in relation to the convex hull of the formation energies ΔE_f when mixing some set of elements.
$\Delta E_d >> 0$ implies that a material is highly unstable, while $\Delta E_d = 0$ implies that the material is perfectly stable.

2.3 Autoencoders

Autoencoders are a class of generative neural networks consisting of two types of neural networks: encoder networks and decoder networks. The aim of the encoder network is to compress high dimensional input data $X \in \mathcal{X}$ into a lower dimensional latent representation $Z \in \mathcal{Z}$ according to an encoded distribution $Q_Z := \mathbb{E}_{P_X}[Q(Z|X)]$. This encoded distribution is realized by a neural network, meaning it is trainable.
These latent representations form a latent space that can then be sampled from a prior distribution P_Z and be decoded, which yields a result according to the posterior distribution $P_G(X|Z)$, which takes the form of a neural network, just as for the encoder.
Autoencoders are trained by encoding data into a latent representation and decoding it. The original data x is then compared to the decoded reconstruction $\hat{x}$ using some cost function $c(x, \hat{x})$. There are multiple popular of autoencoders, for example the variational autoencoder [2] which uses Estimated Lower Bound Objective (ELBO) of the KL-divergence between the prior and posterior distributions.
The type of autoencoders that were used in this project, however, are called Wasserstein Autoencoders.

2.3.1 The Wasserstein Autoencoder

The Wasserstein Autoencoder [12] (WAE) is an autoencoder named after the Wasserstein distance. WAEs attempt to optimize two things simultaneously; the cost function c as well as the discrepancy between the aggregated posterior distribution of encoded points in latent space Q_Z and the latent space prior distribution P_Z, which is chosen to take the form of a multidimensional Gaussian with mean $\mu_z = 0$ and variance σ_z. The objective then looks like

$$D_{\text{WAE}}(P_X, P_G) = \inf_{Q(Z|X)\in\mathcal{Q}} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}\left[c(X, G(Z))\right] + \lambda \cdot \mathcal{D}_Z(Q_Z, P_Z) \tag{2.1}$$

where $\mathcal{D}_Z$ is some divergence between the encoded latent space distribution $Q_Z := \mathbb{E}_{P_X}[Q(Z|X)]$ and the assumed latent space prior distribution P_Z and λ is a hyper-parameter.
The encoder Q and decoder G are realised by deep neural networks parametrized by Φ and θ respectively. The original authors of the Wasserstein Autoencoder suggest two different approaches to the divergence $\mathcal{D}_Z$; a Generative Adversarial Network (GAN) based approach and a Maximum Mean Discrepancy (MMD) approach. This project has opted to use a MMD based $\mathcal{D}_Z$, which is given by

$$\mathrm{MMD}_k(P_Z, Q_Z) = || \int_{\mathcal{Z}} k(z, \cdot) dP_Z(z) - \int_{\mathcal{Z}} k(z, \cdot) dQ_Z(z) ||_{\mathcal{H}_k} \tag{2.2}$$

where $k : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ is a positive-definite reproducing kernel and $\mathcal{H}_k$ is the reproducing kernel Hilbert space of real valued functions mapping $\mathcal{Z}$ to $\mathbb{R}$. The kernel used in this project is the *Inverse Multi-Quadratics kernel*, or the IMQ kernel, which is given by

$$k(x, y) = \frac{C}{C + ||x - y||_2^2} \tag{2.3}$$

the scaling constant is chosen as $C = 2d_z\sigma_z^2$, with d_z the latent dimension and σ_z^2 the variance of the prior distribution P_Z. Since it is unrealistic to calculate the integral in equation (2.2) over the entire latent space $\mathcal{Z}$, instead for each datapoint that is encoded, another point in latent space is sampled from the prior P_Z and the distance between them gets added to the loss. The process is outlined in algorithm 1.

Algorithm 1 WAE training algorithm

Require: $\lambda_{\mathrm{MMD}} > 0$, Initialize parameters (Φ, θ)
while Epoch $e < N_{\mathrm{Epochs}}$ **do**
Sample $\{\vec{x}_1, \ldots, \vec{x}_n\}$ from the training set
Sample $\{\vec{z}_1, \ldots, \vec{z}_n\}$ from the prior P_Z
Sample $\bar{\vec{z}}_i$ from $Q_\Phi(Z|\vec{x}_i)$ for $i = 1, \ldots, n$
Update Q_Φ and G_θ by descending:

$$\frac{1}{n}\sum_{i=1}^{n} c(\vec{x}_i, G_\theta(\bar{\vec{z}}_i)) + \frac{\lambda_{\mathrm{MMD}}}{n(n-1)}\sum_{l\neq j} k(\vec{z}_l, \vec{z}_j) + \frac{\lambda_{\mathrm{MMD}}}{n(n-1)}\sum_{l\neq j} k(\bar{\vec{z}}_l, \bar{\vec{z}}_j) - \frac{2\lambda_{\mathrm{MMD}}}{n^2}\sum_{l,j} k(\vec{z}_l, \bar{\vec{z}}_j)$$

end while

2.4 Analysing the latent space

The latent space $\mathcal{Z}$ holds the compressed representations of the input data. In a well-ordered latent space, clusters of latent representations with some property should form. Since the latent space is high-dimensional, it is not possible to visualize it all at once. There are however, multiple strategies to project high-dimensional points onto a 2D or 3D representation.

2.4.1 Principal component analysis

Principal Component Analysis (PCA) is a technique that linearly transforms data onto a new basis where the basis vectors represent directions of maximum variance. The first principal component captures the largest variance in the data, and each subsequent component captures the largest remaining variance subject to orthogonality constraints. The fact that the transformations are linear also means that points in the transformed space can be inversely transformed back into the original space.
For data with k features and n observations, the principal components of the data can be obtained by computing the eigenvalues and eigenvectors of the covariance matrix of the data and sorting the eigenvalue-eigenvector pairs by decreasing eigenvalue.
For a dataset represented by an $n \times k$ matrix X, first calculate the covariance matrix

$$\overline{X} = hu^T \tag{2.4}$$

$$B = X - \overline{X} \tag{2.5}$$

$$C = \frac{1}{n-1} B^T B \tag{2.6}$$

where h is a $n \times 1$ vector containing all ones and u is a vector containing the mean value for all k features in X.
Then the matrices V and D containing the eigenvectors and eigenvalues of C such that

$$V^{-1} C V = D \tag{2.7}$$

are computed using some eigenvalue decomposition algorithm. The columns of V are then sorted by descending corresponding eigenvalue, and the eigenvectors are used as a basis.
In practice, PCA calculations were performed using the Python package scikit-learn [9].

2.4.2 t-SNE

t-SNE, or t-distributed stochastic neighbour embedding, is an iterative, non-linear dimensionality reduction technique. t-SNE attempts to preserve the neighbourhood of points, meaning that points that are close to each other in high-dimensional space end up close to each other in the low-dimensional projection. This means that t-SNE can be a useful tool to identify clusters in high-dimensional data. Just as for PCA, in practice, the scikit-learn implementation of the algorithm was used in this project [10].

2.5 Fréchet inception distance

To evaluate the reconstruction performance of each architecture, it is not enough to only consider the loss.
The Fréchet inception distance (FID) is often used as a score when assessing generative models, especially for image generation. It is however unclear how to calculate

this score for materials generation, although there are suggestions [8] for some applications.
To keep things simple, the distributions involved were assumed to be multidimensional Gaussian distributions such that the closed form of the FID could be used. The closed form expression is given by

$$d_F(\mathcal{N}(\mu,\Sigma),\hat{\mathcal{N}}(\hat{\mu},\hat{\Sigma})) = ||\mu-\hat{\mu}||_2^2 + \text{trace}(\Sigma+\hat{\Sigma}-2(\Sigma\hat{\Sigma})^{\frac{1}{2}}) \tag{2.8}$$

where $\mathcal{N}(\mu,\Sigma)$ denotes the original data distribution with mean μ and covariance matrix Σ and $\hat{\mathcal{N}}(\hat{mu},\hat{\Sigma})$ denotes the distribution of the generated data in the same way.
A lower FID score implies better generative performance of a model.

3
Method

This project aims to find new rules for oxide stability with the help of machine learning. This involves three main steps;

1. Gather oxide data
2. Train a generative neural network using the data to reproduce the original data
3. Sample the neural network and examine what changes in the materials as the stability changes

3.1 Training data

Training data was gathered from the Materials Project database [5] using the following filters

- Materials that include Oxygen
- Energy above hull $\Delta E_d \in [0, 2.5]$ eV/atom
- Formation energy $E_{form} \leq 0$ eV/atom
- Only experimentally observed materials
- Total number of atoms in the unit cell $N_{\text{atoms}} \leq 100$
- Materials containing elements larger than Pb ($Z = 82$) were excluded.

The gathered database contained around 20000 oxides.

Each entry in the database was preprocessed into a vector which contained the number of oxygen atoms N_{oxygen}, the number of non-oxygen atoms N_{other}, the composition as a vector of fractions for each element with oxygen excluded, $\vec{C}$ and the energy above hull ΔE_d. To improve training stability, the vector was normalized according to

$$\overline{N}_{\text{oxygen}} = \frac{N_{\text{oxygen}}}{N_{\text{atoms}}^{\text{max}}} \tag{3.1}$$

$$\overline{N}_{\text{other}} = \frac{N_{\text{oxygen}}}{N_{\text{atoms}}^{\text{max}}} \tag{3.2}$$

$$\overline{\Delta E}_d = \frac{\Delta E_d - \mu_{\Delta E_d}}{\sigma_{\Delta E_d}} \tag{3.3}$$

The distributions of these quantities are shown in figures 3.1, 3.2, 3.3 and 3.4.
The composition vector was constructed as

$$\vec{C}_i = \frac{N_{i+1}}{N_{\text{atoms}}} \tag{3.4}$$

where the index $i = Z - 1$ such that the first index of $\vec{C}$ corresponded to the fraction of H in the material, the second index to the fraction of He and so on.
The element distribution for the database is shown in figure 3.5.
For all experiments, the database was split into training and validation data by randomly shuffling the data (with a fixed seed for reproducibility) and 85% was chosen as training data and 15% as validation data.

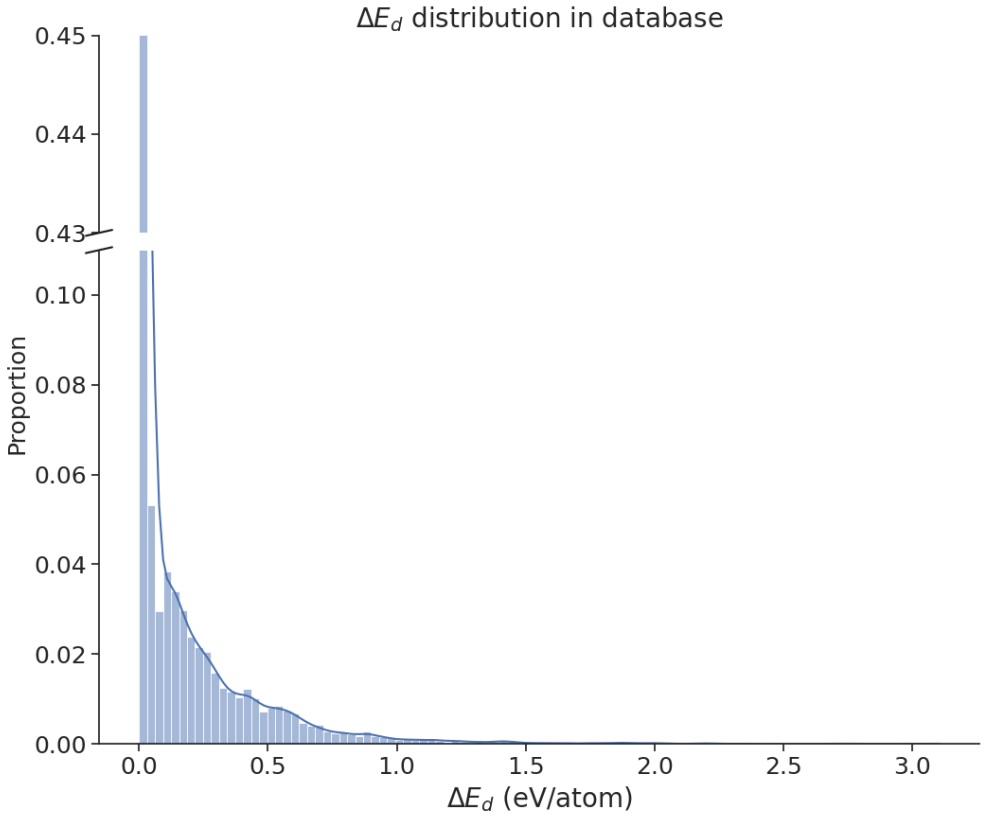


Figure 3.1: Distribution of ΔE_d in the database. Note the broken y-axis.

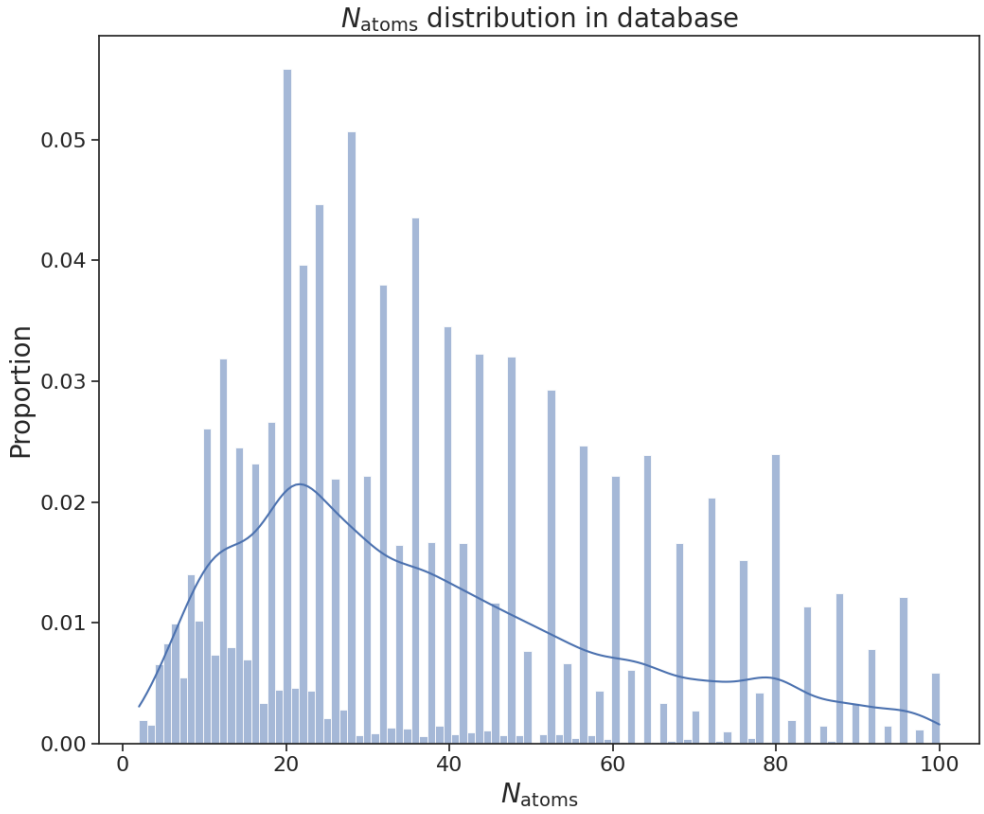


Figure 3.2: N_{atoms} distribution in the database.

3.2 Autoencoder architecture

Several model architectures were tested during the project. Three were selected to gather results, descriptions of these are listed below.

3.2.1 Standard Wasserstein autoencoder

The first architecture used was the standard Wasserstein autoencoder. The encoder and decoder networks used symmetric layer sizes, where the encoder network was mirrored in the decoder network. A schematic image is shown in 3.6. To ensure that the composition vector $\vec{C}$ sums to 1, the softmax function was applied to the raw predicted composition vector $\hat{\vec{C}}$ according to

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_i \exp(x_i)} \tag{3.5}$$

3.2.2 Double decoder Wasserstein autoencoder

Since the objective of the model is to predict stability, it makes sense that ΔE_d should receive extra attention. Therefore a double decoder Wasserstein autoencoder

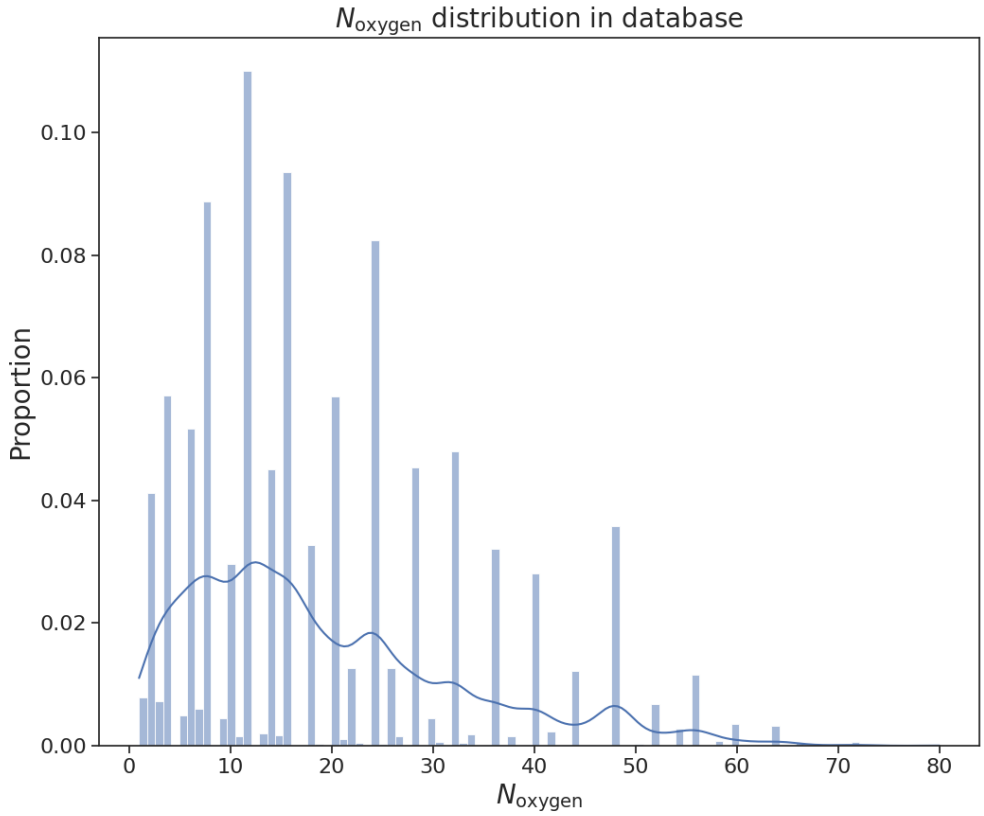


Figure 3.3: N_{oxygen} distribution in the the database.

(DDWAE) was used. DDWAEs tries to minimize two loss functions at once; one for stability and one for composition, two strategies to do this were tested in the project where both of the setups used the same layer sizes. The softmax function was applied to the composition vector according to equation 3.5 in the same way as for the standard WAE. A schematic picture of a DDWAE is shown in Figure 3.7.

3.2.2.1 Combined composition and stability objective

One strategy to train the DDWAE was to simply combine the losses calculated from the two decoders as

$$\mathcal{L}_{\text{objective}} = \mathcal{L}_{\text{composition}} + \lambda_{\text{EAH}} \cdot \mathcal{L}_{\text{EAH}} + \lambda_{\text{MMD}} \cdot \mathcal{L}_{\text{MMD}} \tag{3.6}$$

This approach meant that the model could be trained in the same way as the standard WAE using only one backpropagation per datapoint per epoch when training. The relative importance of the stability objective can also be tuned using the parameter λ_{EAH}. In this project $\lambda_{\text{EAH}} = 1$ was used to get comparable results to the other setups.

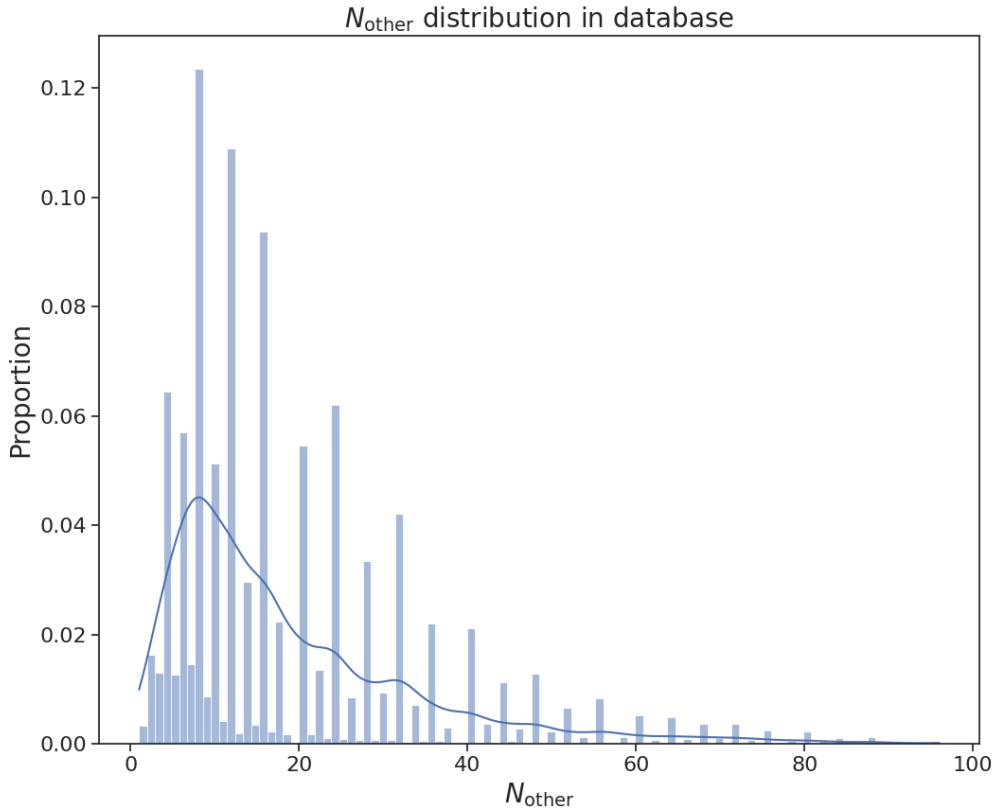


Figure 3.4: N_{other} distribution in the database.

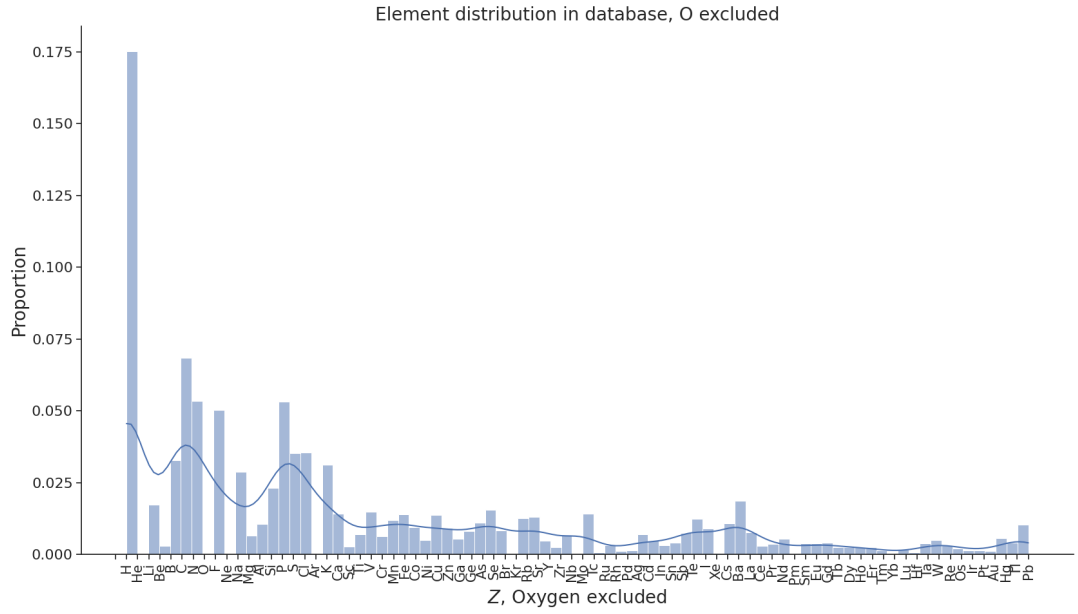


Figure 3.5: Element distribution in the database.

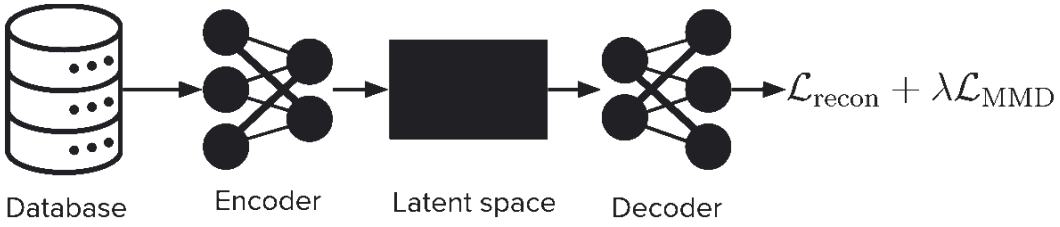


Figure 3.6: Schematic picture of a Wasserstein autoencoder.

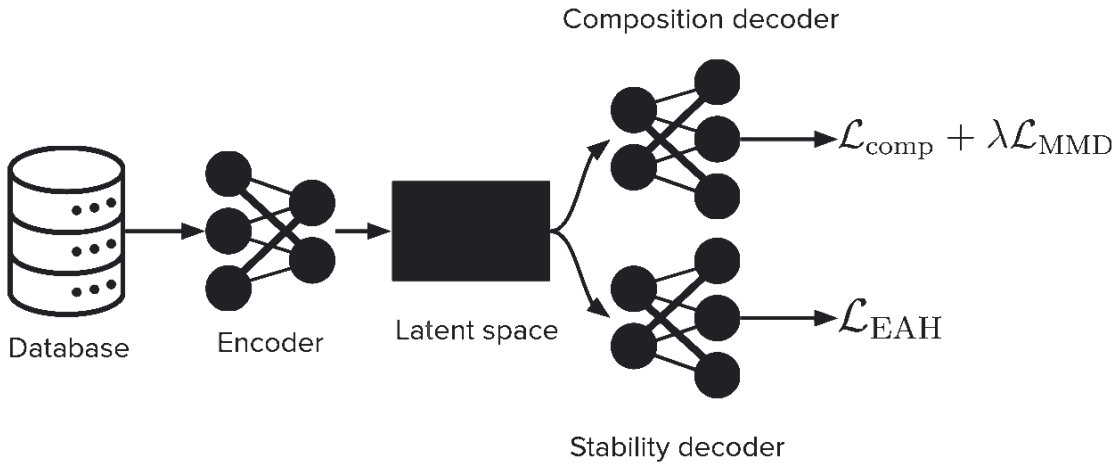


Figure 3.7: Schematic picture of a double decoder Wasserstein autoencoder.

3.2.2.2 Separated composition and stability objectives

Another approach to training the DDWAE is to treat the losses for stability and composition separately and backpropagate twice per datapoint per epoch during training.
Since the encoder network is updated for two different objectives the same training data may get encoded to two different points in latent space within the same epoch, which might lead to issues with training stability and an unorganized latent space. To attempt to mitigate this, some layers in the encoder were frozen during back-propagation. This means that one or more layers in the encoder are updated only for a single objective. Another issue that arises is that applying the Wasserstein regularization (the MMD term) twice for different objectives may also lead to an unorganized latent space.
In this project, Wasserstein regularization was only applied to the composition part of the model.

3.3 Training the models

All of the models were trained using the ADAM optimizer [6] with $\beta_1 = 0.9, \beta_2 = 0.999$. The starting learning rate was set to $\gamma = 10^{-4}$, which was adaptively decreased by a factor of 5 when the validation loss plateaued.
All models were first trained on a smaller dataset with $d_z = 8$ and a small number of layers. The number of layers as well as the layer sizes were then increased until the training losses did not improve significantly. The Wasserstein regularization term was then added and the model was trained on the full dataset. A parameter sweep for the latent dimension was performed for all models. Each model was trained using $d_z = 4, 6, 8, 12, 16, 20, 24, 32, 64, 128$ for 5000 epochs. The average validation loss for the last 100 epochs, as well as the FID score, were used to select the optimal latent dimension d_z. Several other parameters were empirically tested throughout the project, the values selected for these parameters are shown in table 3.1.

Parameter	Value
Activation function	ReLU
σ_z	2.0
λ_{MMD}	10
N_{Batch}	2000
N_{Epochs}	10^4

Table 3.1: Hyperparameters used in all final models.

3.4 Sampling the models and analysing the latent space

In order to capture as much of the variations inside the latent space, PCA transforms were applied to the latent spaces. To identify which principal components (PCs) that contained the most information about stability, the models were sampled for stability along each of the principal components from $-2\sigma_z$ to $2\sigma_z$, as well as uniformly sampled in the planes spanned by each combination of PCs from $-2\sigma_z$ to $32sigma_z$ in each direction. This resulted in a corner plot in which the planes containing the most information about stability could be selected for further sampling.
The selected planes were then sampled for composition along lines from a highly unstable point to a stable point. It was important that these lines were linear combinations of the two PCs spanning the plane (i.e. diagonal lines) such that the information contained in both the PCs was included in the sampling.
The compositions could then be examined with regard to stability and new rules for stability theorized.

4
Results

4.1 Parameter sweep for latent dimension

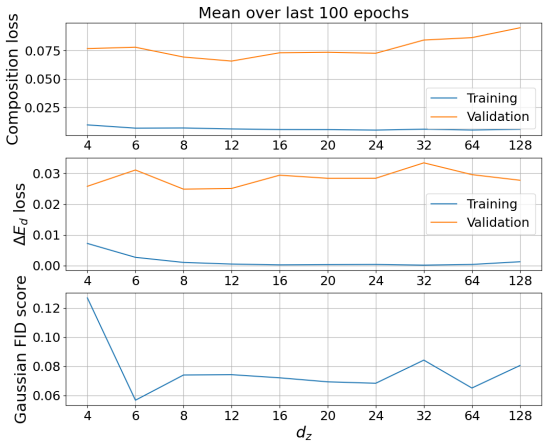
The results of the parameter sweep are shown in Figure 4.1. Three aspects were considered in the choice of d_z for each architecture:

1. The validation losses
2. The FID-score
3. d_z needed to be small enough that analysis of the latent space is feasible

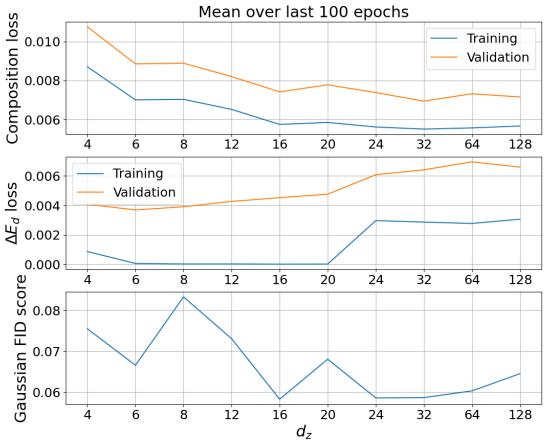
To accommodate the third criterion, a maximum value of $d_z = 12$ was set. Based on these considerations, the values for d_z were chosen as in table 4.1. As is clear for all models, a larger d_z does not yield better performance. In terms of FID score and $\mathcal{L}_{\text{EAH}}$ the DDWAE models perform better than the standard WAE.

Architecture	d_z
Single decoder	6
Mixed loss double decoder	12
Separated loss double decoder	12

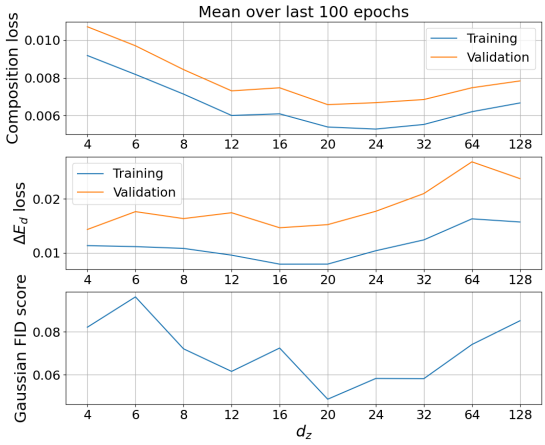
Table 4.1: Choices of latent dimensionality d_z for all models.



(a)



(b)

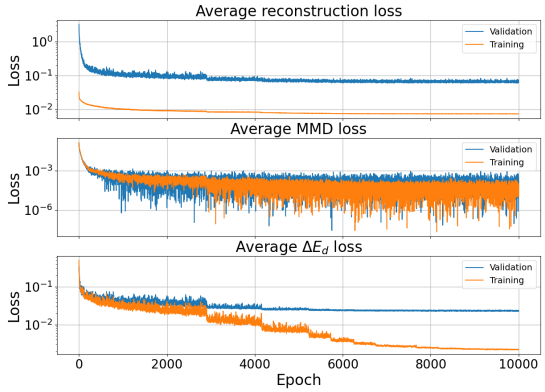


(c)

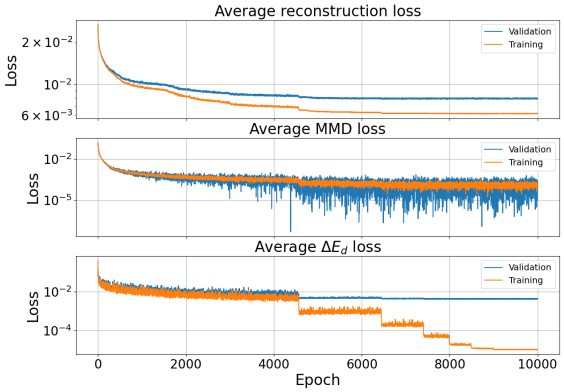
Figure 4.1: Final losses and FID score for parameter sweep over d_z for (a) Single decoder setup (b) Combined loss double decoder setup (c) Separated loss double decoder setup

4.2 Model performances

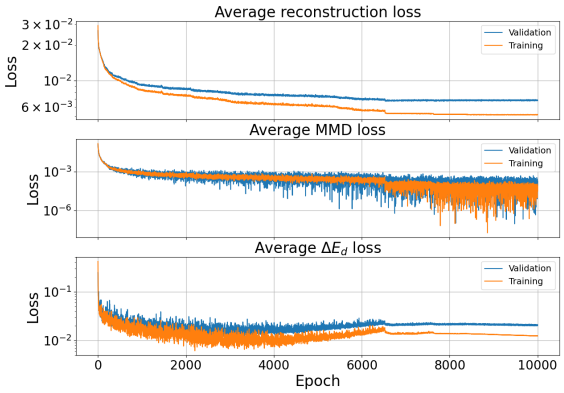
The training and validation losses for each of the architectures are shown in Figure 4.2



(a)



(b)



(c)

Figure 4.2: Training losses for (a) Single decoder setup (b) Combined loss double decoder setup (c) Separated loss double decoder setup

The final losses and FID scores are reported in Table 4.2.

Model	$\mathcal{L}_{\text{comp}}$	$\mathcal{L}_{\text{EAH}}$	$\mathcal{L}_{\text{MMD}}$	FID score
Single decoder	$6.252 \cdot 10^{-2}$	$\mathbf{1.412 \cdot 10^{-4}}$	$2.342 \cdot 10^{-2}$	0.08161
Mixed loss double decoder	$7.911 \cdot 10^{-3}$	$4.505 \cdot 10^{-3}$	$\mathbf{1.771 \cdot 10^{-4}}$	0.06922
Separated loss double decoder	$\mathbf{6.816 \cdot 10^{-3}}$	$2.093 \cdot 10^{-2}$	$1.955 \cdot 10^{-4}$	**0.05823**

Table 4.2: Final losses and FID score for each of the three models. Lower values indicate better performance and the best values are marked as bold.

The DDWAE models perform worse with regards to the stability objective, but better in terms of all other metrics.

The explanation is that composition and ΔE_d are properties fundamentally at different scales and separating these objectives lead to an overall lower FID score and better performance. The higher stability loss for the separated DDWAE is likely due to the MMD term being applied only to the composition objective.

4.3 Sampling the latent space

The latent spaces of the three models were sampled for composition. Below are the results of this sampling.

4.3.1 Standard Wasserstein setup

By first applying a PCA transform and then uniformly sampling the latent space in the planes spanned by the resulting principal components, as well as along the principal components, resulted in the corner plot shown in 4.3, which shows that stability information is concentrated in the last three principal components.

Since the range of ΔE_d values is largest along the last principal component, the latent space was sampled from unstable points to stable points in the planes spanned by the last principal component and every other principal component, i.e. in the planes depicted in the last row of Figure 4.3.

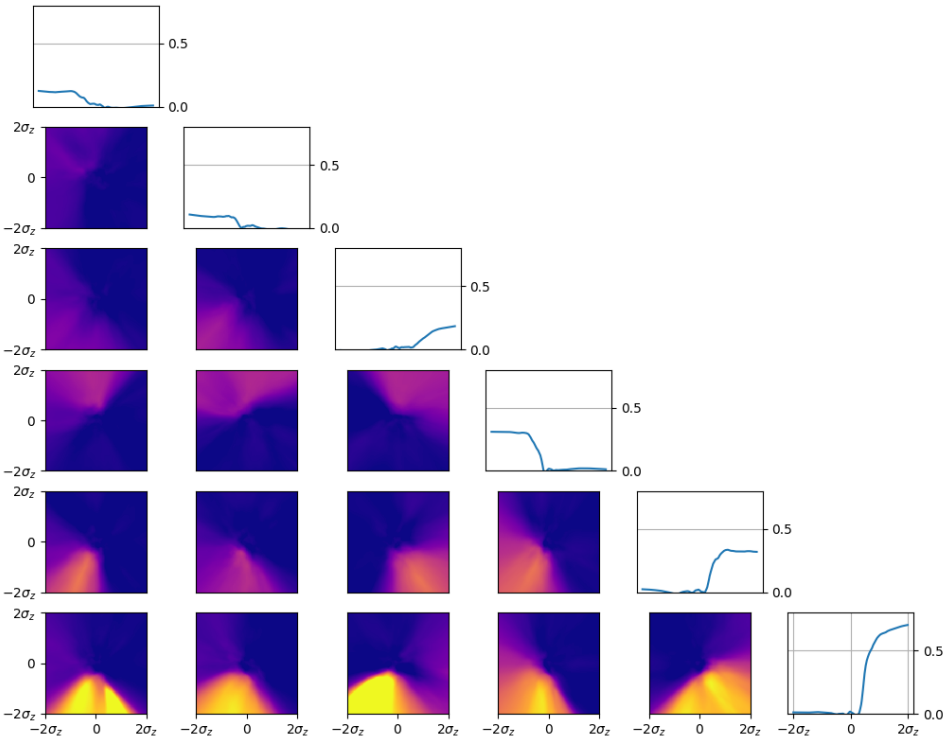
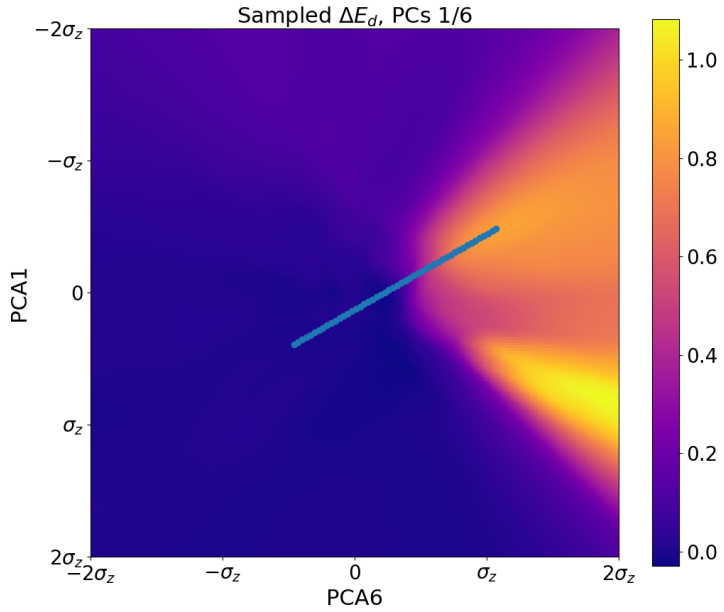
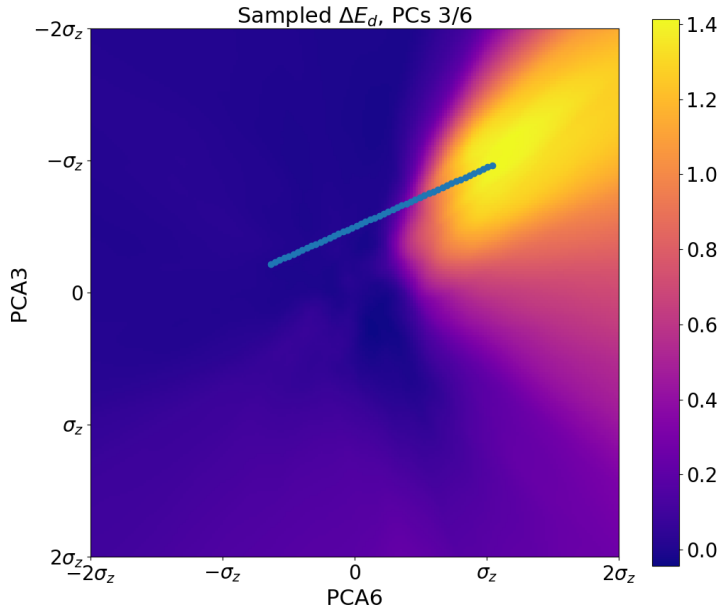


Figure 4.3: Corner plot with ΔE_d sampled uniformly from $-2\sigma_z$ to $2\sigma_z$ along each principal component, as well as in the plane spanned by each combination of principal components.

Two of the sampled paths are shown in figure 4.4 and the resulting compositions and ΔE_d, N_{oxygen}, N_{other} for the unstable point, halfway point (in terms of ΔE_d) and stable point are shown in figure 4.5 and figure 4.6.



(a)



(b)

Figure 4.4: (a) Path sampled in the plane spanned by PC 1/6, (b) Path sampled in the plane spanned by PC 3/6

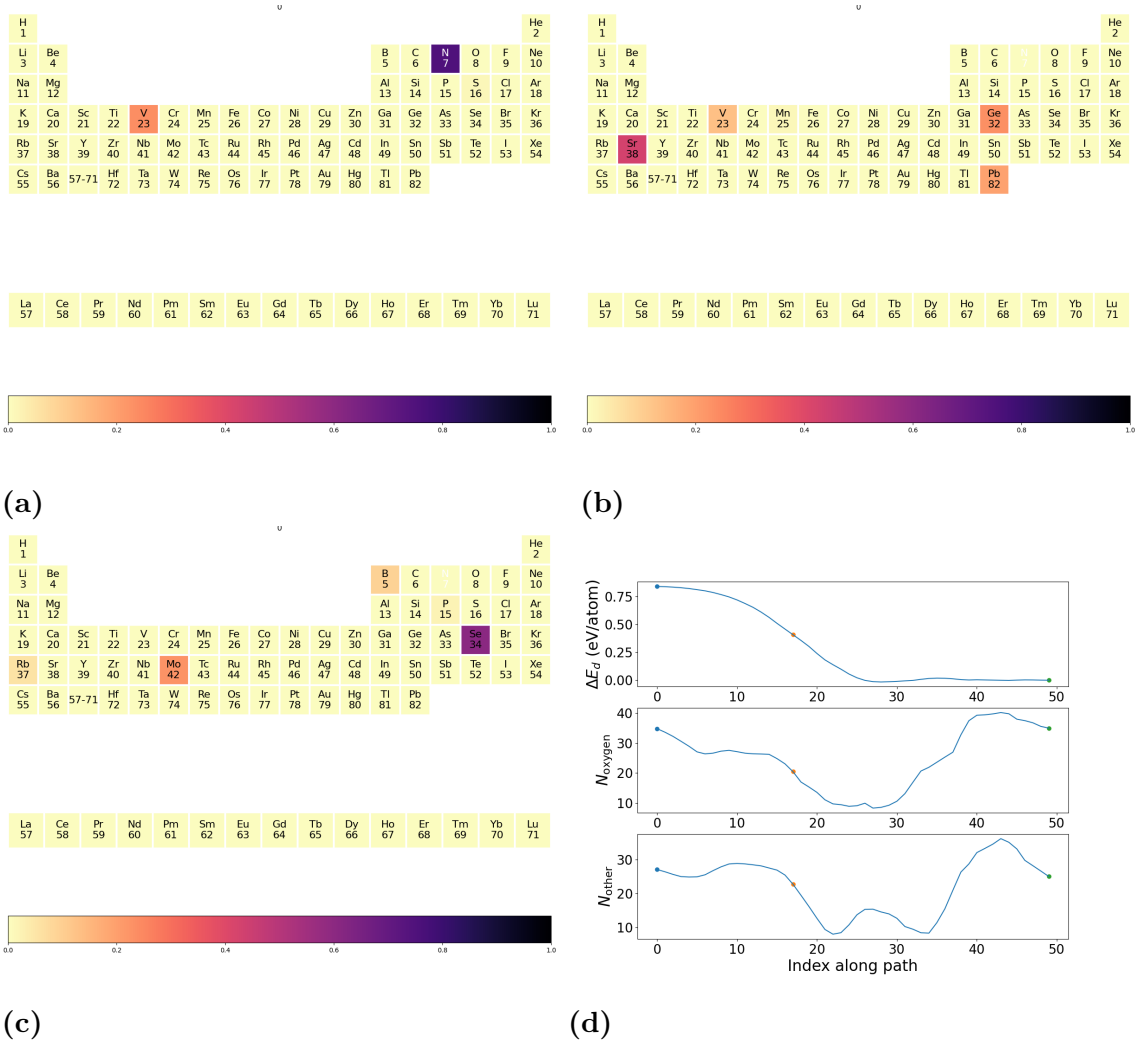


Figure 4.5: Sampled composition for (a) unstable point (b) halfway point (c) stable point for plane spanned by PC1 and PC6.
(d) $\Delta E_d, N_{\text{oxygen}}, N_{\text{other}}$ along the sampled path.

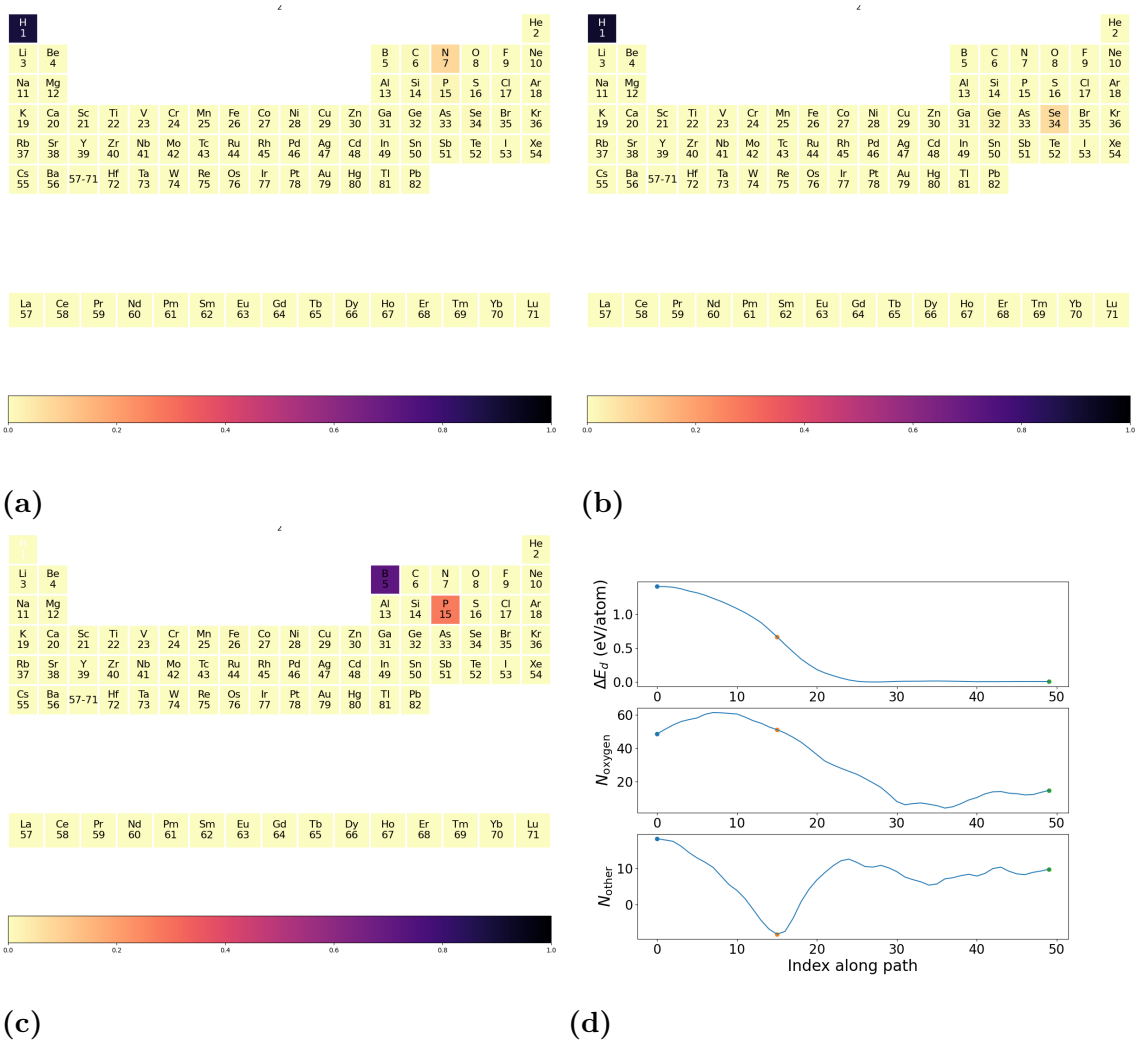


Figure 4.6: Sampled composition for (a) unstable point (b) halfway point (c) stable point for plane spanned by PC3 and PC6.
(d) $\Delta E_d, N_{\text{oxygen}}, N_{\text{other}}$ along the sampled path.

The unstable, halfway and stable compositions for all sampled paths are also shown in a more readable form in 4.3.

Principal components	Unstable composition	Halfway composition	Stable composition
1/6	N_{20} V_6 O_{35}	V_3 Ge_5 Sr_{10} Pb_4 O_{20}	B_2 Se_{15} Rb_1 Mo_6 O_{35}
2/6	O_3	V_2 Mn_1 Ge_7 Sr_5 O_{15}	P_3 V_6 Rb_3 Pb_1 O_{17}
3/6	H_{16} N_1 O_{49}	O_{51}	B_7 P_3 O_{15}
4/6	C_2 N_3 Cl_{11} K_{20} Pb_1	V_2 Mn_1 Ge_7 Sr_6 Pb_2 O_{22}	H_8 Na_{13} K_1 Se_2 O_{27}
5/6	K_{12} O_{55}	Ge_3 O_{33}	Se_1 O_4

Table 4.3: Compositions sampled along all paths for the standard WAE model.

The sampled compositions show a trend where reactive non-metals are substituted for a mix of metals and alkali metals, as well as a trend towards heavier cations.

4.3.2 Combined composition and stability objective

Just as for the standard WAE, the combined DDWAE was first PCA transformed and then uniformly sampled in the planes spanned by the resulting principal components, as well as along the principal components. This resulted in the corner plot shown in Figure 4.7, which shows that stability information is concentrated in the last principal component.

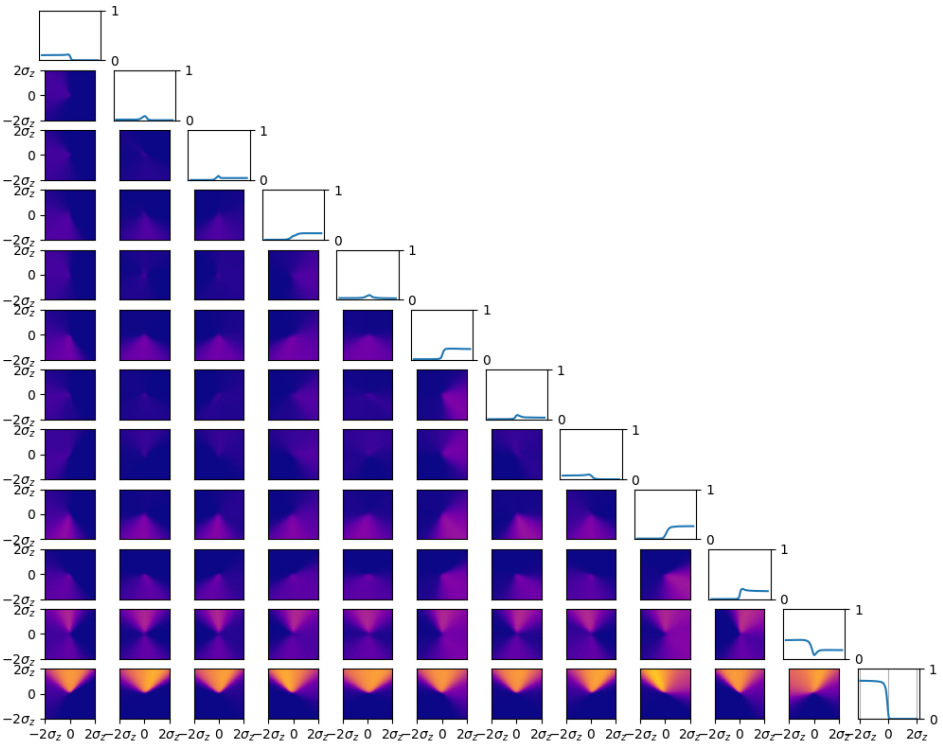
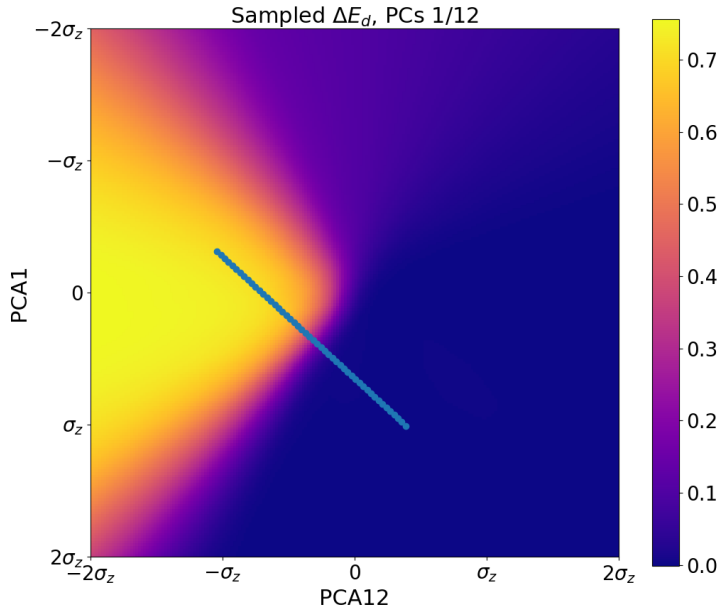
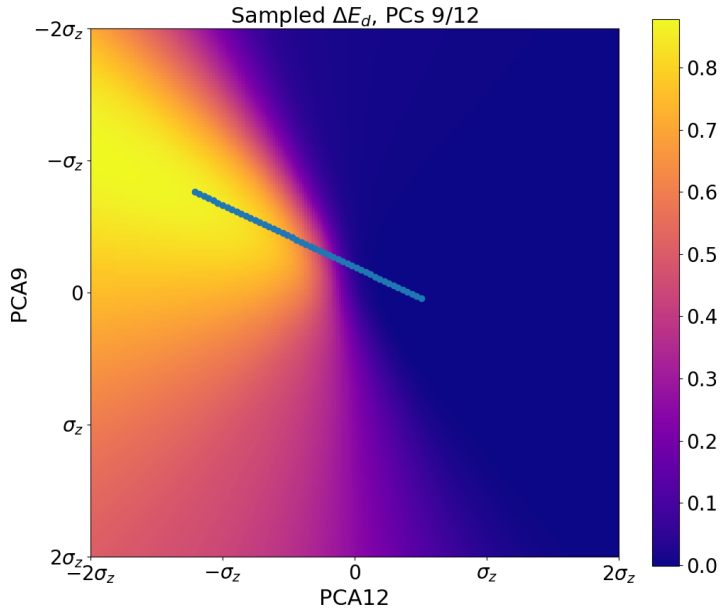


Figure 4.7: ΔE_d sampled along each PC as well as sampled uniformly in the planes spanned by each pair of PCs.

Just as for the standard WAE, the planes spanned by the last principal component and every other principal component were sampled from an unstable point to a stable point along a straight path. Two examples of these paths are shown in Figure 4.8.



(a)



(b)

Figure 4.8: Paths in planes spanned by PCs (a) PC1 and PC12 (b) PC9 and PC12

The sampled compositions as well as $\Delta E_d, N_{\text{oxygen}}, N_{\text{other}}$ for the unstable point, halfway point and stable point along the paths are shown in figures 4.9 and 4.10.

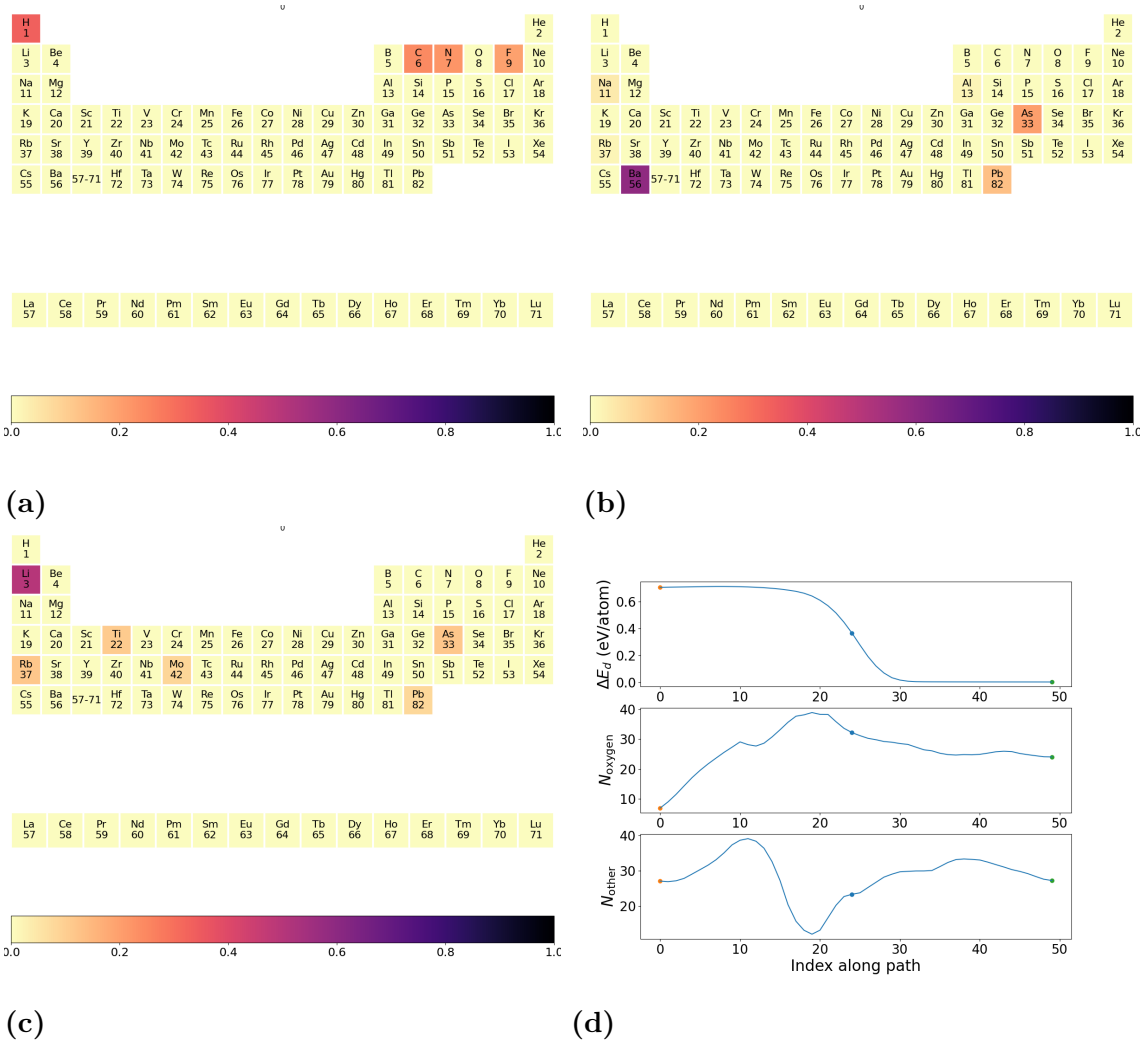


Figure 4.9: Sampled compositions from the combined loss double decoder setup in plane spanned by PC1 and PC12 for (a) the unstable point, (b) halfway point and (c) stable point. (d) $\Delta E_d, N_{\text{oxygen}}, N_{\text{other}}$ along the sampled path.

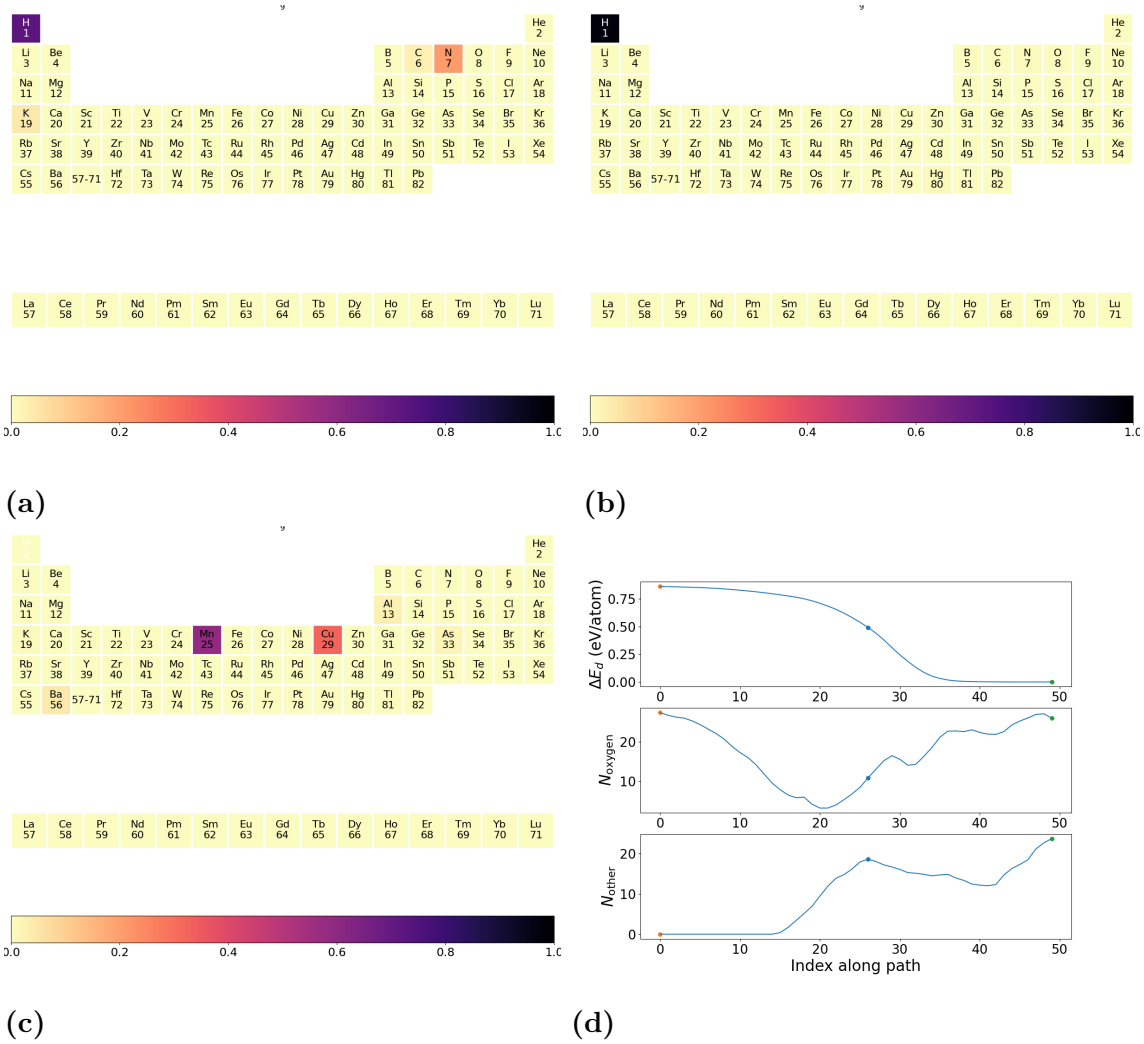


Figure 4.10: Sampled compositions from the combined loss double decoder setup in plane spanned by PC9 and PC12 for (a) the unstable point, (b) halfway point and (c) stable point. (d) $\Delta E_d, N_{\text{oxygen}}, N_{\text{other}}$ along the sampled path.

The sampled compositions for the paths in all of the spanned planes are shown in Table 4.4.

PC combination	Unstable	Halfway	Stable
1/12	$H_9\ C_7\ N_6\ F_5\ O_7$	$Na_1\ As_5\ Ba_{14}\ Pb_3\ O_{32}$	$Li_{14}\ Ti_3\ As_3\ Rb_3\ Mo_2\ Pb_2\ O_{24}$
2/12	$H_3\ N_1\ F_2\ Cl_3\ O_1$	$H_{12}\ P_5\ Fe_1\ As_1\ O_{34}$	$H_2\ P_1\ V_{22}\ Mn_7\ O_{35}$
3/12	$H_4\ C_{33}\ N_{12}\ S_2\ Cl_5\ K_2\ O_{17}$	$P_3\ O_{11}$	$P_6\ O_{20}$
4/12	$H_1\ Si_5\ Ca_3\ O_{14}$	$H_5\ P_1\ Ba_1\ O_{36}$	$K_2\ As_1\ Ag_8\ O_{15}$
5/12	$B_1\ N_{20}\ Na_{11}\ Al_1\ P_{11}\ S_1\ O_{46}$	$H_4\ C_2\ Fe_5\ As_1\ Eu_1\ O_{26}$	$H_3\ Se_8\ O_7$
6/12	$H_6\ C_6\ N_{10}\ O_{34}$	$H_1\ C_1\ P_4\ Cl_1\ Fe_1\ Ba_1\ O_{28}$	$C_2\ Cl_2\ K_1\ O_{20}$
7/12	$H_1\ N_{21}\ O_{21}$	$H_6\ C_1\ Cu_5\ O_{15}$	$Cl_2\ Mn_1\ Cu_5\ O_{15}$
8/12	$F_{31}\ Cl_1\ K_1\ O_{31}$	$H_7\ P_8\ Fe_1\ O_{31}$	$H_5\ K_{16}\ Mn_3\ As_5\ O_{19}$
9/12	O_{27}	$Fe_{14}\ As_3\ O_{11}$	$Mn_{14}\ Cu_6\ As_3\ O_{26}$
10/12	$H_{17}\ C_1\ N_5\ K_1\ O_{12}$	$H_{12}\ O_{39}$	$Al_1\ Mn_{13}\ Cu_7\ Ba_1\ O_{23}$
11/12	$H_{19}\ N_{13}\ O_5$	$H_{16}\ O_{33}$	$F_4\ K_7\ Mn_4\ Cu_1\ O_{12}$

Table 4.4: Sampled compositions for paths in the planes spanned by PC12 and every other PC for the combined loss double decoder setup.

For the combined DDWAE, just as for the standard WAE, the transition away from reactive non-metals is clear. In this case the stable compositions also contain a variety of metals such as Mn, Ag and Cu.

4.3.3 Separated composition and stability objectives

The stability in all the planes spanned by the PCs of the separated DDWAE is shown in figure 4.11.
Just as for combined DDWAE, most of the information concerning stability was contained in the last principal component. The planes spanned by the last principal component and every other principal component were sampled from an unstable point to a stable point along a straight path. Two examples of these paths are shown in Figure 4.12.

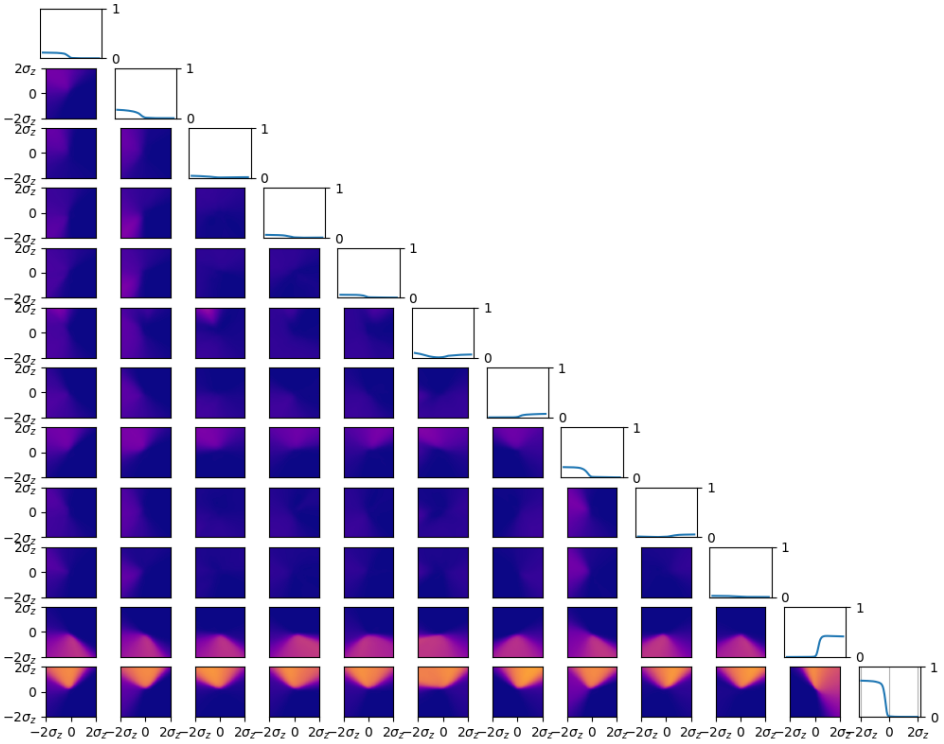
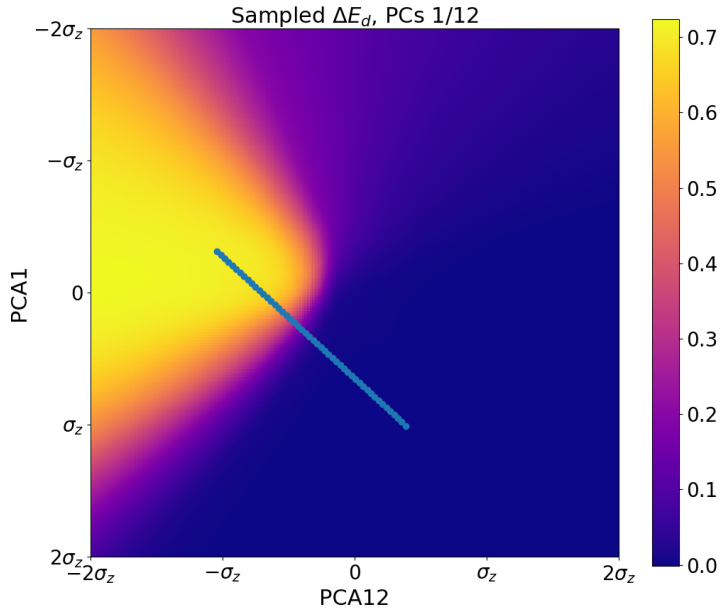
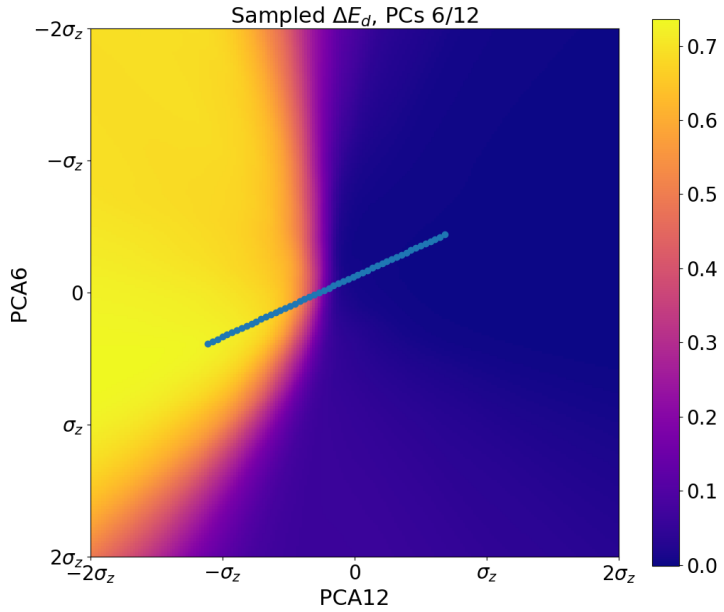


Figure 4.11: ΔE_d sampled along each PC as well as sampled uniformly in the planes spanned by each pair of PCs.

The sampled compositions as well as $\Delta E_d, N_{\text{oxygen}}, N_{\text{other}}$ for the unstable point, halfway point and stable point along the paths shown in Figure 4.12 are shown in Figures 4.13 and 4.14.



(a)



(b)

Figure 4.12: a) Path in PC plane 1/12 b) Path in PC plane 6/12

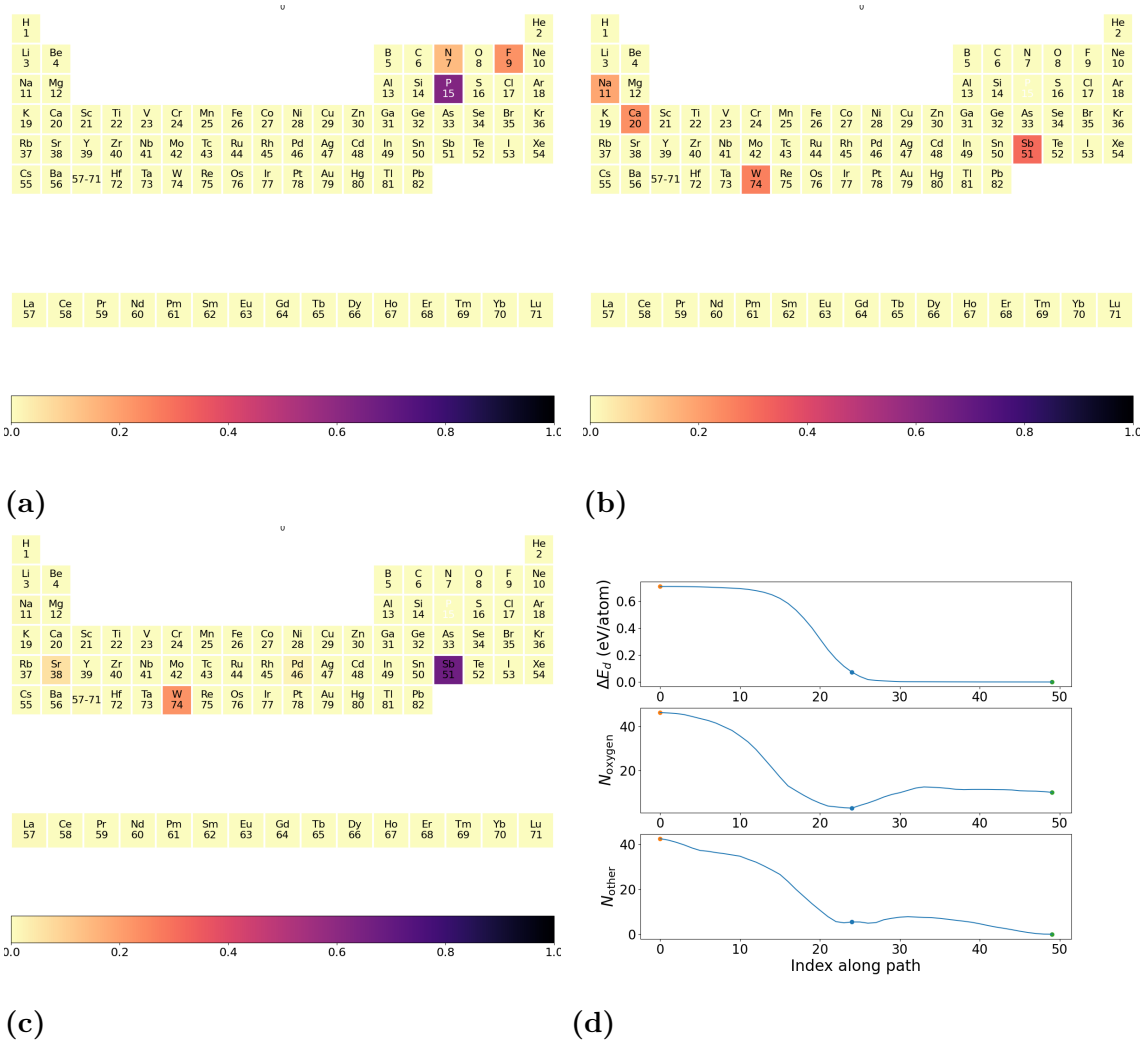


Figure 4.13: a) Unstable composition for path in PC plane 1/12 b) Halfway composition c) Stable composition d) $\Delta E_d, N_{\text{oxygen}}, N_{\text{other}}$ along the sampled path.

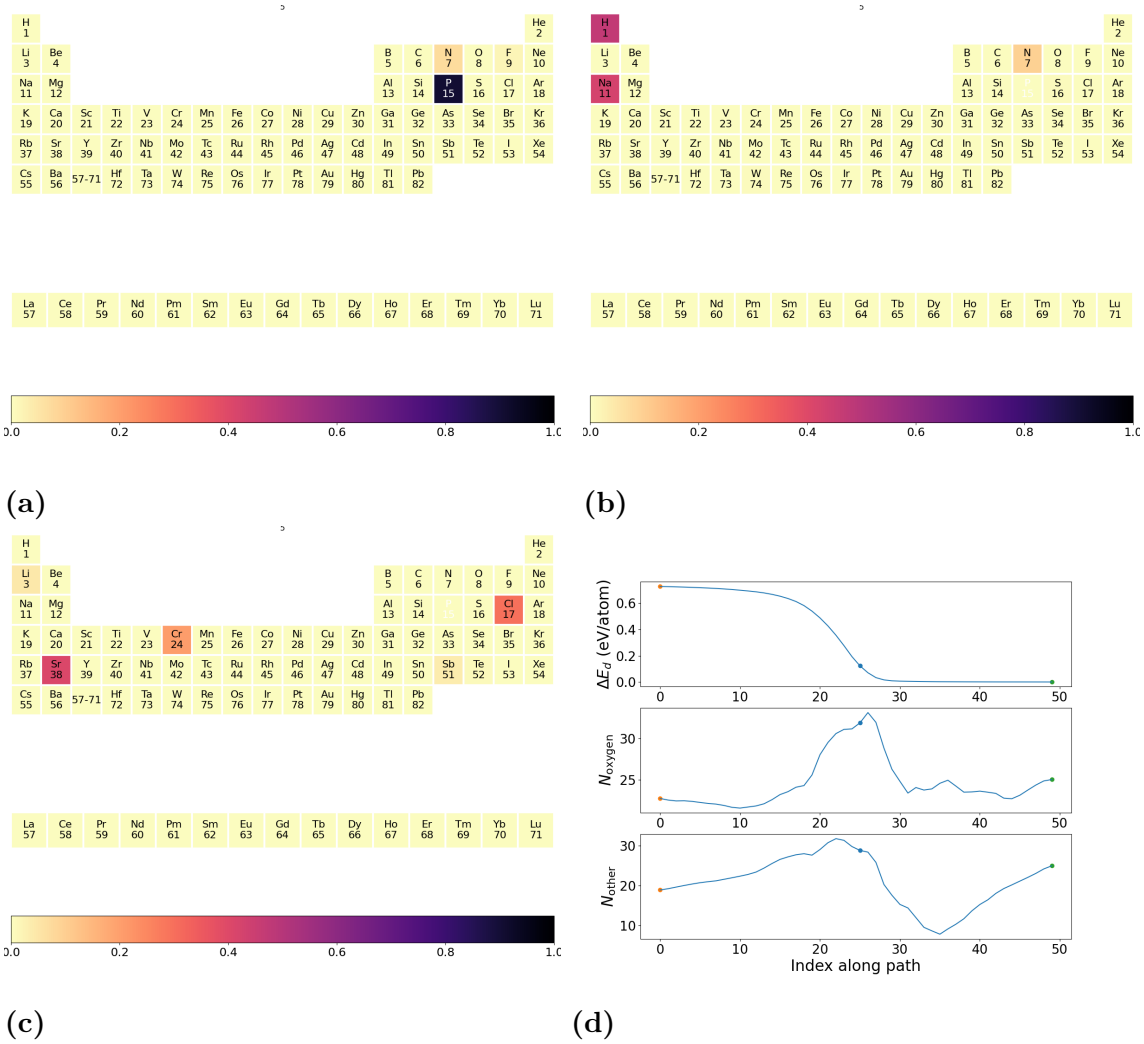


Figure 4.14: a) Unstable composition for path in PC plane 6/12 b) Halfway composition for path in PC plane 6/12 c) Stable composition for path in PC plane 6/12 d) $\Delta E_d, N_{\text{oxygen}}, N_{\text{other}}$ along the sampled path.

The sampled compositions along all paths in all planes are shown in Table 4.5.

PC	Unstable	Halfway	Stable
1/12	$N_5F_8P_{22}O_{40}$	$H_1F_7Na_3Nb_1O_6$	$Sb_3W_1O_{12}$
2/12	$N_3F_{10}P_{18}O_{32}$	$N_4S_{36}K_1Nb_{11}O_{25}$	$S_{14}Nb_4Cs_6Hg_9O_{21}$
3/12	$N_5F_2P_{18}O_{20}$	$Na_{11}S_1W_{11}O_{37}$	$S_1Cr_{12}Te_1Ba_1W_2O_{29}$
4/12	$C_1N_3F_7P_{21}O_{31}$	$N_5Na_5P_{11}W_3O_{31}$	$Fe_{11}Mo_3Te_1O_{30}$
5/12	$N_5F_{18}P_{16}O_{29}$	$H_{11}N_7Na_{12}S_1O_{31}$	$B_{23}O_{31}$
6/12	$N_1P_{17}O_{23}$	$H_{13}N_8Na_{11}S_1O_{31}$	$Li_1Cl_7Cr_5Sr_{10}Sb_1O_{25}$
7/12	$N_{16}P_3Ba_2O_2$	$H_{10}N_{10}Na_{13}S_1O_{35}$	$Na_{19}O_{18}$
8/12	$N_3F_7P_{20}O_{31}$	$Li_9S_2K_1Te_1La_4W_2O_{29}$	Li_7O_9
9/12	$N_3F_{11}Na_8P_7O_{18}$	$H_8N_{15}Na_8O_{30}$	$B_8Cs_1O_{25}$
10/12	$C_2N_1F_{18}P_5O_{15}$	$F_{15}Na_6O_{15}$	$H_1B_{25}O_{44}$
11/12	$N_4F_9P_{18}O_{32}$	$H_2N_3F_1P_{19}O_{35}$	$H_4P_{12}Ca_4O_{32}$

Table 4.5: Sampled starting compositions, midway compositions (the sampled composition with a ΔE_d value closest to half the starting value) and the end composition with $\Delta E_d \approx 0$.

The compositions again transition from reactive non-metals in the unstable compositions to metals and heavier cations such as Sb in the stable compositions.

4.3.3.1 Freezing encoder layers

To mitigate the fact that the encoder might place the same datapoint at different points in latent space when training the decoders separately, freezing some layers in the encoder was tried.
The placement in latent space of the same datapoint after both encoder propagations was tracked over all epochs during training and the L_2 distance between the two points were calculated for each epoch. These distances, when not freezing any layers in the encoder, are shown in Figure 4.15, and the distances when freezing layers are shown in Figure 4.16.

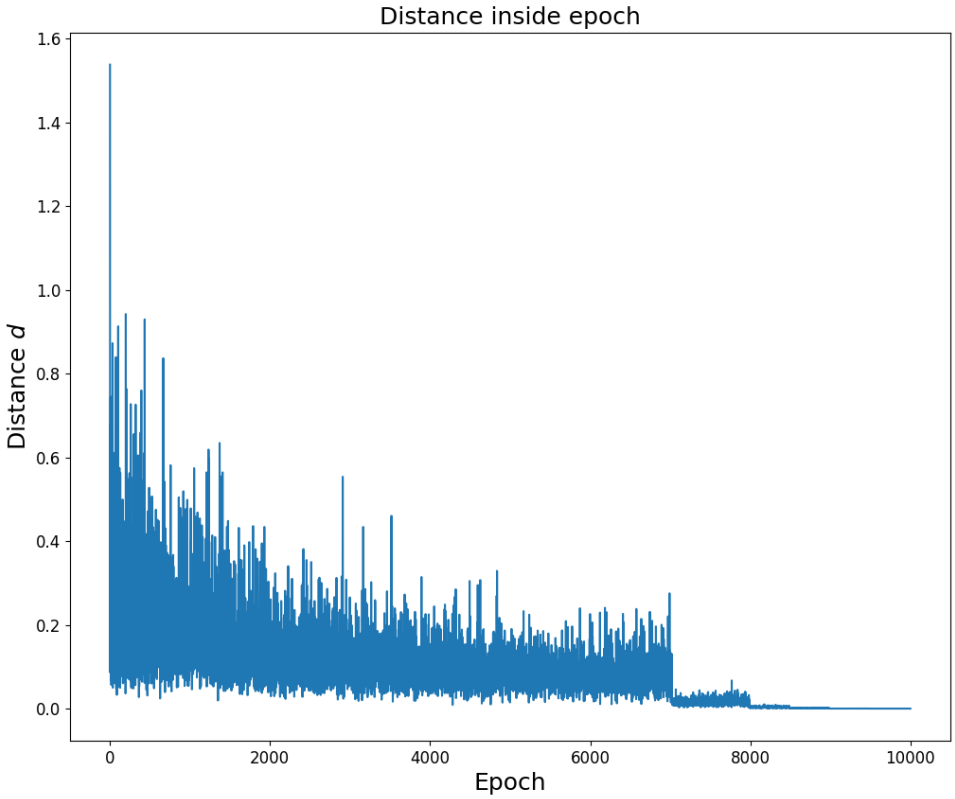


Figure 4.15: Distances within the same epoch for separated loss DDWAE without freezing layers in the encoder

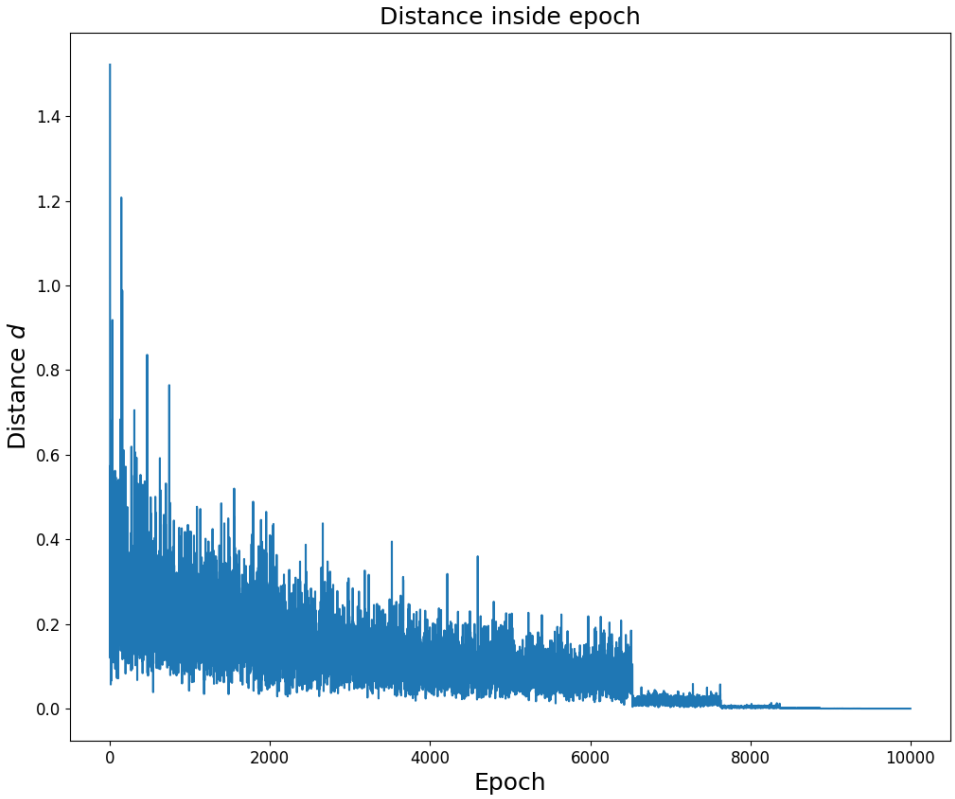


Figure 4.16: Distances within the same epoch for separated loss DDWAE with freezing layers in the encoder

Freezing layers in the encoder had a negligible effect on the convergence of encoder placements in the latent space.

5
Discussion

The sampled compositions produced by the models are not necessarily physically meaningful. This is acceptable in the context of this project, where the objective was to identify qualitative trends between composition and stability. However, generating physically meaningful compositions would be important if the goal would have been to generate new oxides. One approach to do this could be to incorporate constraints into the training objective (e.g. charge neutrality) or by feeding the models more information such as oxidation states and coordination environments, so that the network learns basic chemical rules during generation.

One important assumption in this project is the use of a Gaussian prior for the latent space distribution. Alternative choices, such as a multi-modal Gaussian distribution [7] or an energy based prior [3] could result in a more structured latent space which would be beneficial when sampling compositions in the latent space and could potentially reveal more detailed patterns for stability.

The sampling strategy used in this work is relatively simple, and more sophisticated sampling of the latent space could yield different rules or increase confidence in the observed trends. Because sampling was performed manually, only very small regions of the latent space were explored. By imposing well-defined constraints, the procedure could be automated and much larger regions of latent space could be investigated. One option would be to sample along paths that preserve the total number of atoms in the composition, which would have the added benefit of excluding the number of atoms as a factor impacting stability. Another straightforward alternative would be to sample many points from stable and unstable regions and perform a statistical analysis, although this would partly undermine the motivation for using an autoencoder, since such an analysis could be carried out directly on the original database.

Although some patterns with regard to stability emerge in this project, a complete set of rules requires structural information as well as compositional.
At the beginning of the project, an attempt was made to include both structure and composition, which failed to produce meaningful results, and the decision was therefore made to focus solely on composition. To incorporate structure as well, a graph-based network, especially for the encoder, are likely a good candidate. There are already examples of Variational autoencoders with graph based message passing encoders that succeed in generating novel crystalline materials [13], and a similar approach would be well suited also for the goals of this project.

If the structure is also considered and the models succeed in generating physically valid oxides, the active learning cycle [11] could be used to further improve the model.

6
Conclusion

6.1 Latent space exploration

This project shows that a well ordered latent space of a Wasserstein autoencoder can be used to systematically explore relationships between element composition and thermodynamic stability of oxides and that a double decoder approach can be an effective way to focus on specific properties.

6.2 Suggested new rules of stability based on composition

Using only composition to represent oxides and within the dataset used in this project, three main trends emerged that can be formulated into rules

1. Oxide compositions containing only reactive non-metals tend to be unstable
2. The presence of metals in the composition tends to favour stability
3. Heavier cations appear more often in stable compositions

Bibliography

[1] Christopher J. Bartel. Review of computational approaches to predict the thermodynamic stability of inorganic solids. *Journal of Materials Science*, 57(11):5163–5209, 2022.

[2] P. Kingma Diederik and Welling Max. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, November 2019.

[3] Debottam Dutta, Chaitanya Amballa, Zhongweiyang Xu, Yu-Lin Wei, and Romit Roy Choudhury. Learning energy-based variational latent prior for vaes. *arXiv preprint arXiv:2510.00260*, September 2025.

[4] Janine George, David Waroquiers, Davide Di Stefano, Guido Petretto, Gian-Marco Rignanese, and Geoffroy Hautier. The limited predictive power of the pauling rules. *Angewandte Chemie International Edition*, 59(19):7569–7575, 2020.

[5] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin A. Persson. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1):011002, 07 2013.

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2017.

[7] Nao Nakagawa, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Gromov-wasserstein autoencoders. *arXiv preprint arXiv:2209.07007*, September 2022.

[8] Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet chemnet distance: A metric for generative models for molecules in drug discovery. *Journal of Chemical Information and Modeling*, 58(9):1736–1741, 2018. PMID: 30118593.

[9] scikit-learn developers. Pca — scikit-learn 1.7.2 documentation. `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`, 2024. Accessed: 2026-01-10.

[10] scikit-learn developers. Tsne — scikit-learn 1.7.2 documentation. `https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html`, 2024. Accessed: 2026-01-10.

[11] Yuri Sverchkov and Mark Craven. A review of active learning approaches to experimental design for uncovering biological networks. *PLoS Comput. Biol.*, 13(6):e1005466, June 2017.

[12] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. 2019.

[13] Tian Xie, Xiang Fu, Octavian-Eugen Ganea, Regina Barzilay, and Tommi Jaakkola. Crystal diffusion variational autoencoder for periodic material generation. 2022.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY