FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Tatiana Vargicová

Reconstruction of magnetic configurations using machine learning approaches

Department of Condensed Matter Physics

Supervisor of the bachelor thesis: RNDr. Pavel Baláž, Ph. D.

Study programme: Physics
Branch of study: FP

Prague 2022

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In…..... date..... signature

Title: Reconstruction of magnetic configurations using machine learning approaches

Author: Tatiana Vargicová

Department: Department of Condensed Matter Physics

Supervisor of the bachelor thesis: RNDr. Pavel Baláž, Ph. D. Fyzikální ústav AV ČR

Abstract: This work focuses on developing an autoencoder well-suited for reconstruction of magnetic phases with a prospect of future application in phase-recognition task. Specifically, it was investigated how does the autoencoder performance change when Hamiltonian term in added to the loss function, previously computed solely from MSE error. It was found that the effect Hamiltonian inclusion on MSE error is phase specific. Notably for spiral phase, the reconstruction significantly improves. In contrast, for some of the intermediate phases, the reconstruction greatly degrades. This was especially true for the intermediate phase composed of spirals combined with merons. In addition to simple MSE error, it was also investigated whether the reconstruction conserves the energy ascribed to individual spins. It was found that the Hamiltonian term improves the spin-energy conservation for all the magnetic phases.

Keywords: neural networks, autoencoder, Heisenberg model, Hamiltonian

Contents

Introduction

Inspired by human brain morphology and functioning, new computational techniques emerged. When one wishes to stay abstract, one talks about "artificial intelligence" and "machine learning-based" techniques. Otherwise, when one desires to invoke concrete informatics algorithms, the more precise term neural network (NN) is used. NNs represent a powerful tool suitable for variety of tasks, such as data classification, analysis, processing or compressing. In this work, we will use autoencoder NN to compress and reconstruct magnetic configuration.

Specifically, we will investigate, how will the performance of autoencoder change when we supply or model with Hamiltonian term. Since the Hamiltonian measures system's energy, by including it in our model we hope to force the autoencoder better conserve the energy when processing the data. Intuitively, we would expect that this would lead to autoencoder reconstructing the configuration in a more physical way. Nevertheless, as we will learn such interpretation is not straight-forward as different magnetic configurations will be influenced in a dissimilar way with the over effect being very complex.

This work is structured in a following manner: In the section named "The physical background", IT models of magnetic configurations will be explained. We will introduce the Hamiltonian term, describe its components and effect on the configuration. Furthermore, we will describe how do the different magnetic phases look like for the values of external factors we work with. At the end of this section, we will present the work of Kwon et al. 2021 to show our motivation for combining the NN with Hamiltonian term.

In the section called the "IT background", we will explain the basics of neural network functioning, focusing specifically on the type of neural network (the convolutional autoencoder) we will work with.

In the section "Methods" we will describe the features of our dataset, specifying what magnetic phases are present in our dataset and how did we choose to classify them. We will also describe the structure of the two distinct types of autoencoders this work will use.

In the section "Implementation of the Hamiltonian", we will undergo a series of reflexions (including a short experiment) with aim to determine how to best implement the Hamiltonian in the autoencoder. In the end, we will work with a "local Hamiltonian" (newly defined quantity describing the conservation of spin energies) and the results for this model will be presented.

Finally, in the "Discussion" we will discuss these results in context of magnetic phase recognition problematics.

1 The physical background

Artificial intelligence has proven to be of great utility in condensed matter physics. Numerical methods are being used to enhance resolution of grid-system configurations, implement temperature fluctuation, simulate complex many body systems, and examine classical and quantum spin states (as discussed in Kwon et al. 2021a and Iakolev et al. 2018). They are also extremely useful when studying phase transitions, not only for learning order parameters (Wetzel 2017) but also for establishing boundaries between individual phases (Carrasquilla and Melko 2017).

1.1 Studying magnetic systems with machine learning

Specifically, machine learning-based techniques are often being employed to study phase transitions in magnetic compounds. The whole situation is simplified by reducing the dimension of the system to planar case (which however may simulate well thin layers). In 2D, there are two main models used to describe the system: Ising and Heisenberg (Kawamura 2010). In Ising model spins of the atoms are only allowed to acquire binary values of -1 or +1. In Heisenberg model, the spins are thought as three-dimensional vectors.

The organization of the spins within the system respects thermodynamic considerations. One can obtain allowed configuration through Monte Carlo simulations using Metropolis algorithm. As explained in Kapitan et al. 2021, in each step of the Metropolis algorithm, one spin from the configuration is selected. If changing the direction of this spin results in configuration with lower energy, then the direction of the spin is changed. If changing the direction of this spin results in configuration with higher energy, then the direction of the spin is changed with the probability equal to ratio between thermodynamic distributional probability of these two states. (The probability of change is equal to distributional probability of the state with altered spin divided by distributional probability of the state with non-altered spin.) Beside Monte Carlo simulations, there are machine learning-based techniques for obtaining magnetic configuration: For Ising model, such approach was used by Morningstar and Melko 2018. For Heisenberg model, this was done by Kwon et al. 2019a and Kwon et al. 2021.

This work with revolve around Heisenberg model with configurations generated by Monte Carlo simulations. Specifically, we will focus on phase transition in magnetic compounds in vicinity of absolute zero. There one can distinguish three main phases: ferromagnetic phase, spiral phase and skyrmionic phase. The projection of z-spin (out-of-plane) component in these

phases may be seen on Figure 1.1. Figure 1.2 shows the in-plane projection of spins. A thorough analysis of each phase including energetic considerations is provided by Ezawa 2011. In ferromagnetic phase, spins are aligned with the direction of the magnetic field. In spiral phase, spins form sinusoidal waves. In skyrmionic phase, the configuration is composed of skyrmions in hexagonal lattice.

Figure 1.1. Out-of-plane (z) projection of the three main phases as shown on the bwr python colourmap (dark red corresponding to 1, dark blue corresponding to -1). A) ferromagnetic phase B) skyrmionic phase C) spiral phase.

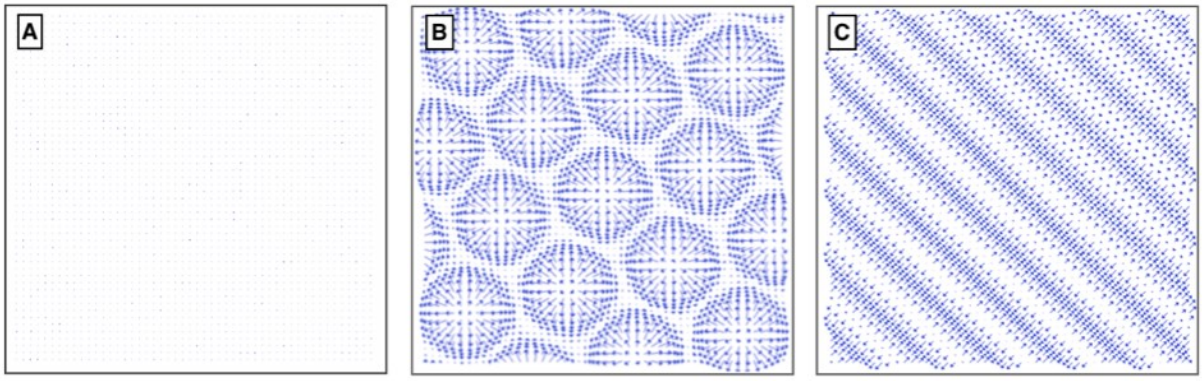

Figure 1.2. In-plane projection of the three main phases shown as a vector plot. A) ferromagnetic phase B) skyrmionic phase C) spiral phase.

1.2 Skyrmions

There is no official definition of skyrmion, but following Evershor-Sitte et al. 2018, we can describe it as "spin structure in which the center of magnetization is in the opposite direction to its boundary and which can be mapped once to the sphere". As a consequence of containing singularity, skyrmions are characterized by topological invariant called topological charge. It is defined via following expression (Mathies 2022):

$$Q = \frac{1}{4\pi} \int S \cdot \left(\frac{\delta S}{\delta x} \times \frac{\delta S}{\delta y} \right) dxdy \tag{1.1}$$

Here S denotes spins of individual atoms and the integration is performed over the region containing the skyrmion. Equation (1.1) is non-well suited for numerical calculation. Typically,

(as was done for instance in Iakovlev et al. 2018 and Heo et al. 2016 one uses form (1.2) as developed by Berg and Lüsher 1981. We will follow this suit too.

$$Q = \frac{1}{4\pi}\sum 2arccos\frac{1+S_i\cdot S_j+S_j\cdot S_k+S_k\cdot S_i}{\sqrt{2(1+S_i\cdot S_j)(1+S_j\cdot S_k)(1+S_k\cdot S_i)}} \quad (1.2)$$

We can also define topological charge density as the quantity calculated by expression (1.1) without performing the integration or by expression (2) without taking the sum. As the Figure 1.3 shows, for skyrmionics phase are plots of topological charge density quite similar to plots of z-magnetisation.

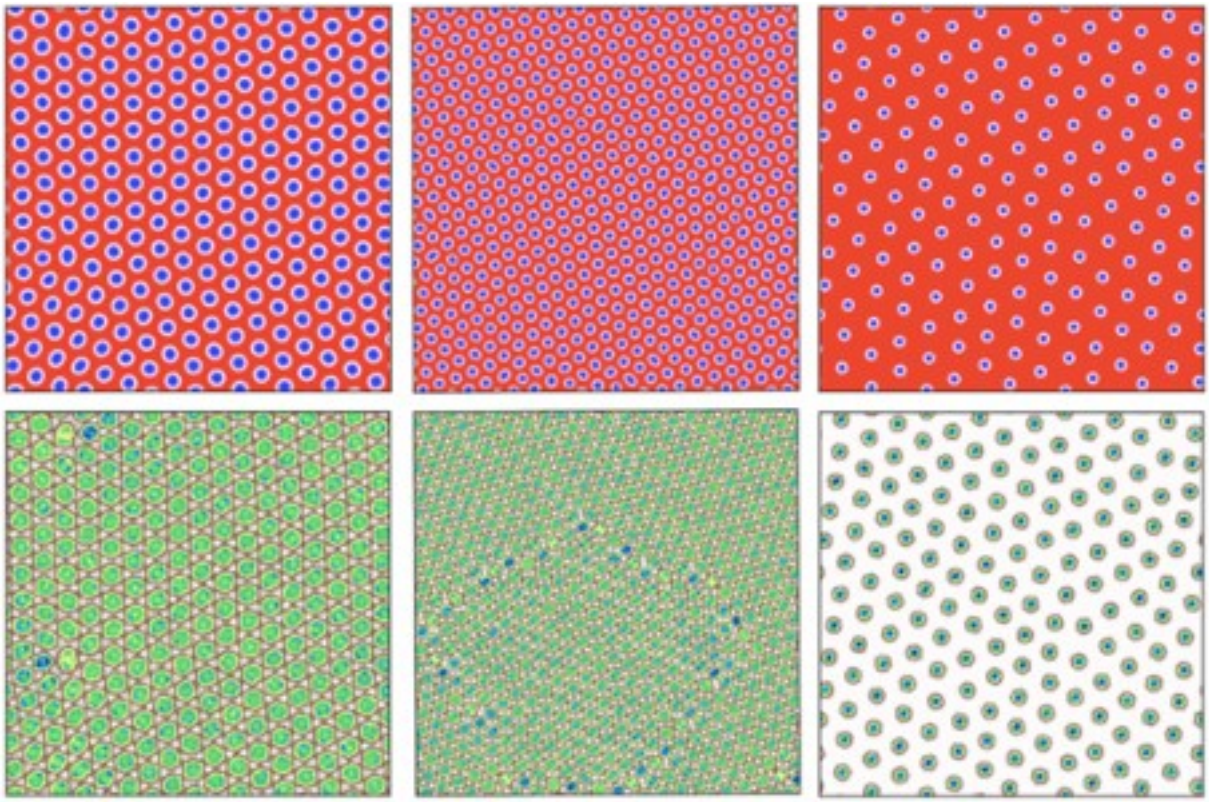

Figure 1.3 Comparison of z-spin projection and topological charge density: The upper row contains plots of z-pin projection; the lower row contains plots of topological charge density.

The exact profile of magnetization inside the skyrmion can take various forms. We differentiate three main types of skyrmions: Bloch skyrmion with Q=-1, Neel skyrmion with Q=-1 and antiskyrmion Q=1 (Kovalev and Sanhoefner 2018). They are depicted on Figure 1.4. These are all plane structures. However, there exist other exotic types of skyrmions, some of them even three dimensional (see Evershor-Sitte et al. 2018). As such, skyrmions are stable configurations whose creation and annihilation can be induced via thermal excitation or favourable tuning of magnetic and electric field. Skyrmions are also capable of later shift – one perceives them as quasi particles performing analogue of Brownian motion. (Mathies 2022). Again, speed and direction of this motion can be regulated through thermal/electric/magnetic gradient or spin

torque (Evershor-Sitte et al. 2018). Since there are ways how to control skyrmions one aspires to use them in racetrack memory - a type of memory where data could be manipulated without having to transfer them from storage from processor (Evershor-Sitte et al. 2018, Cortes-Ortunõ et al. 2017). Other possible applications include: skyrmion based conventional logic devices, skyrmion based transistor, skyrmionic reshuffler for obtaining uncorrelated signals and simulations of biological synapses (Evershor-Sitte et al. 2018).

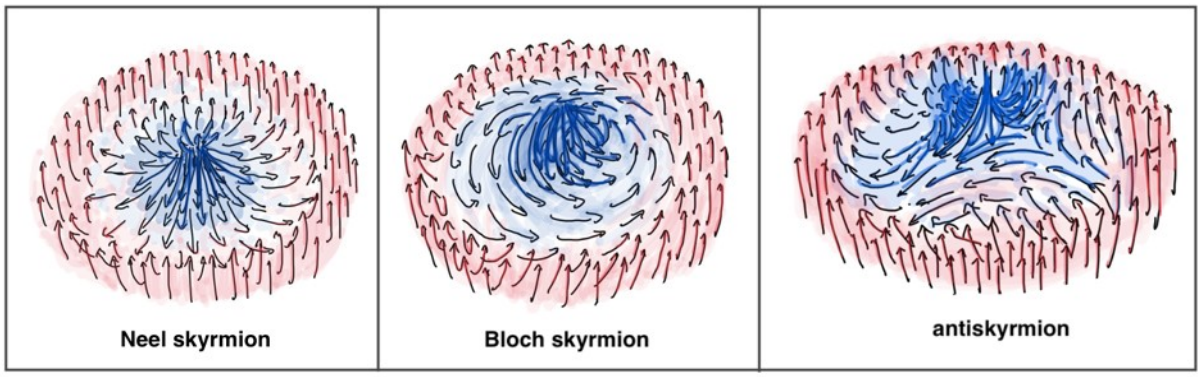

Figure 1.4: Three main basic types of skyrmions – drawings inspired by figures presented in (Evershor-Sitte et al. 2018).

1.3 Hamiltonian of magnetic system

Now that we have introduced notion of skyrmions, we can return to phase transitions. Two-dimensional magnetic compounds are characterized by Hamiltonian with the following form (Kwon et al. 2019, Pepper et al. 2018):

$$H = -J\sum\left(\vec{S}_k \cdot \vec{S}_{k+\vec{u}_x} + \vec{S}_k \cdot \vec{S}_{k+\vec{u}_y}\right) + \sum\left[\left(\overleftrightarrow{D} \cdot \vec{u}_x\right) \cdot \left(\vec{S}_k \times \vec{S}_{k+\vec{u}_x}\right) + \left(\overleftrightarrow{D} \cdot \vec{u}_y\right) \cdot \left(\vec{S}_k \times \vec{S}_{k+\vec{u}_y}\right)\right] - h\sum\left(\vec{S}_k \cdot \vec{u}_h\right) + E_{anis} + E_{dip} + E_{demag} \quad (1.3)$$

(Where J represents exchange interaction constant, $\overleftrightarrow{D}$ represents Dzyaloshinskii-Moriya (DM) tensor and h strength of external magnetic field, with $\vec{e}_h$ being it unitary direction vector. Symbols $\vec{S}_i$ denote spins, $\vec{S}_{i+\vec{e}_x}$ and $\vec{S}_{i+\vec{e}_y}$ are the spins one obtains by transferring along x or y axis by one atom.) First term of (3) represent exchange interaction; the second Dzyalonshinskii-Moriya interaction; the third Zeeman energy; the fourth energy of magnetic anisotropy (preferred direction of magnetic spins characteristic for the material); the fifth energy of dipolar interaction; the sixth energy of demagnetization (changes in magnetic field arising as a consequence of system's own magnetization.) When studying properties of phase transition and individual magnetic phases practically never all contributions of the equation (1.3) are taken into account. Usually, one on first four terms (as was for instance done in Salcedo-Gallo

et al. 2020 or in Kovalev and Sanhoefner 2018) or on the first three terms (as was for instance done in Iakovlev 2018, Kapitan 2021, Albarracín 2022). In this work, we will follow the latter approach, neglecting all but first three terms of the relationship (1.3).

Let us analyse the effect of these three terms closer. It is apparent that exchange interaction would try to align the direction of the neighbouring spin. On the other hand, The Zeeman energy term would attempt to align spins to the direction of the magnetic field which in our case will be that of the z-axis ($\vec{u}_h$=$\vec{u}_z$). The real mystery though, is the Dzyaloshinskii-Moriya term. Since this term contains vector products, one intuitively feels that it would try to compensate the exchange interaction by forcing the neighbouring spins to tilt away from each other. Hence, is not surprising that it is the Dzyaloshinskii-Moriya interaction what facilitates the formations of skyrmions. Nevertheless, how exactly does this happen and what would be the appearance of resulting skyrmions depend on the form of tensor $\overleftrightarrow{D}$ - which is in turn determined by the symmetry of the system. In this work, we will consider the system with geometry alike one shown on Figure 1.5. DM interaction arises from spin-orbital coupling between magnetic and non-magnetic atoms (Güngordü at al. 2016), Kovalev and Sanhoefner 2018). If we denote the distance between first magnetic atom and the nonmagnetic atom $\vec{r}_1$, and the distance between second magnetic atom and the nonmagnetic atom $\vec{r}_2$, with $\vec{u}$ being the distance between the two magnetic atoms, then direction of Dzyaloshinskii-Moriya tensor contracted with $\vec{u}$ will be equal to direction of vector product of $\vec{r}_1$ and $\vec{r}_2$ (Moskvin 2019):

$$\left(\overleftrightarrow{D} \cdot \vec{u}\right) \propto (\vec{r}_1 \times \vec{r}_2) \tag{1.4}$$

Hence in our case, when non-magnetic atoms are positioned as shown on Figure 1.5, the components of DM interaction will be determined by (1.5).

$$\left(\overleftrightarrow{D} \cdot \vec{u}_x\right) = (0,1,0);\ \left(\overleftrightarrow{D} \cdot \vec{u}_y\right) = (-1,0,0) \tag{1.5}$$

With this in mind we can rewrite expression (1.3) as (1.6) – the final form of Hamiltonian we will be working with.

$$H = \sum\left\{-J\left(\vec{S}_{ij} \cdot \vec{S}_{i(j+1)} + \vec{S}_{ij} \cdot \vec{S}_{(i+1)j}\right) + D\left[\left(\vec{S}_{ij} \times \vec{S}_{(i+1)j}\right)_y - \left(\vec{S}_{ij} \times \vec{S}_{i(j+1)}\right)_x\right] - h\left(\vec{S}_{ij}\right)_z\right\} \tag{1.6}$$

The form of $\overleftrightarrow{D}$ we use is known to give rise to Neel skyrmions (Güngordü at al. 2016). How precisely this happens is illustrated on Figure 1.6.

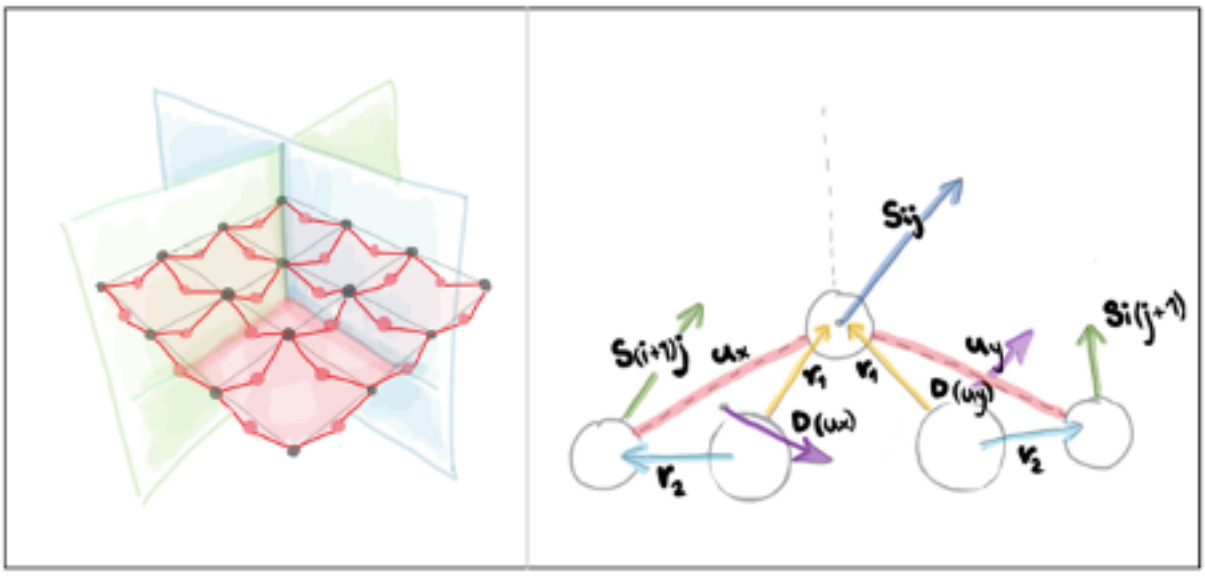

Figure 1.5: On the left, we see positions of non-magnetic atoms which would correspond to symmetry of DM tensor we are working with: The non-magnetic are represented by red spheres; the magnetic atoms are represented by black spheres. The magnetic atoms are positioned directly underneath the line joining the non-magnetic atoms. Hence, this constellation is of C_{2V}.
On the right, we see illustration of the geometrical consideration used to determine the direction of the DM vector. Here, the magnetic atoms are represented by smaller white spheres and non-magnetic atoms are represented by bigger white spheres. We can clearly see that the vector product $(\vec{r}_1 \times \vec{r}_2)$ has the direction (0,1,0) for magnetic atoms lying on the x-axis and the direction (-1,0,0) for magnetic atoms lying on the y-axis.

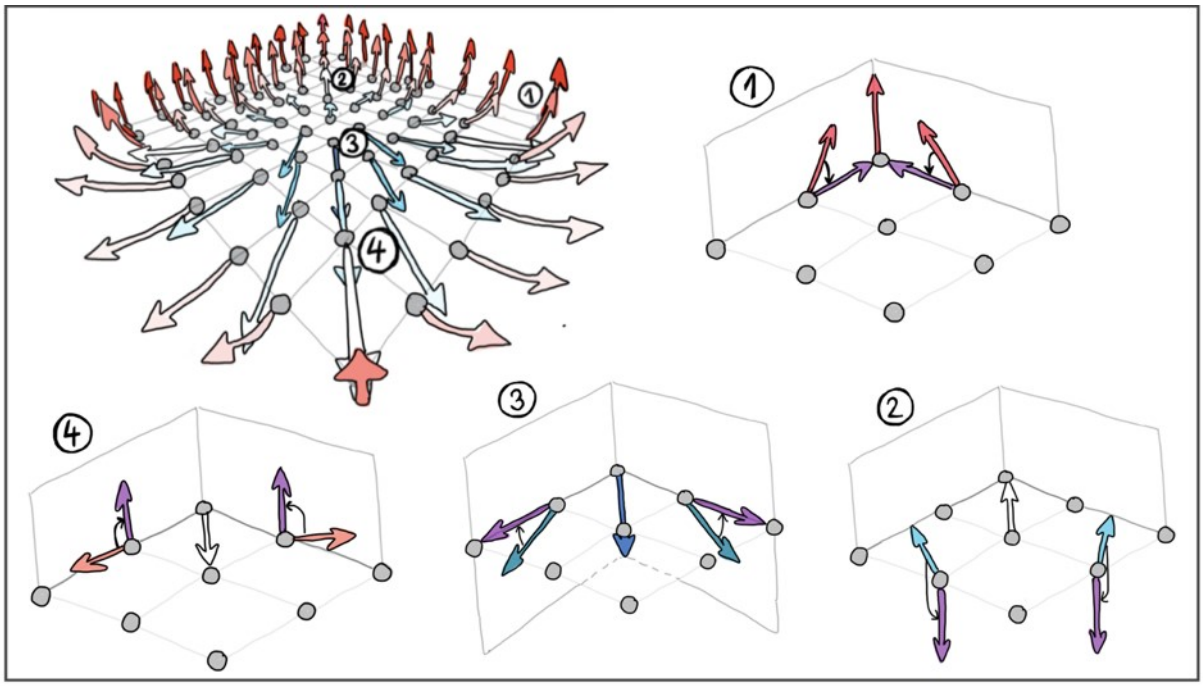

Figure 1.6: Figure showing how DM interaction leads to formation of skyrmions. From the expression (6) we see know, that to minimize the DM term, the vector product of the neighbouring spins must be collinear but of opposite sign to DM vector. This figure shows four cases in which such requirement is easy to satisfy – the DM minimizing vectors are shown in violet. Nevertheless, reaching such optimum would require the direction of neighbouring spins to differ greatly. Such alignment would therefore lead exchange interaction (the first term in expression (6)) achieving energy-unfavourable values. Hence, under the combined effect of these two terms, the neighbouring spins would tilt away from each other in the direction preferred by DM interaction, but this change would be smaller and gradual. The image in the upper left corner of this Figure shows how these small changes can give rise to skyrmion.

1.4 Intermediate phases

In our sample besides the configurations containing pure skyrmions, pure spirals or pure ferromagnetic alignment, there are also intermediate phases. Some configurations contain skyrmions that do not form lattices, but disordered assemblies. They also may be present as solitary objects. Such interphase is termed ‘skyrmionic gas’. It can be seen on Figure 1.7.

Other configurations contain skyrmions which are no longer symmetrical: These appear stretched with one of their axes elongated, so that they resemble stripes rounded ends. If those stripes are long, they can be viewed as finite sized spirals and interpreted as product of transition between skyrmionic and spiral state – they appear on corresponding location of the phase diagram. These ‘stripes with rounded edges’ are called bimerons. On plot of z-spin magnetisation, they appear as one continuous object. Nevertheless, the plot of topological charge density reveals two dense regions at the ends of the bimeron separated by a cleft with no topological charge. These regions of high topological charge density are sometimes perceived as separate objects. One calls them merons. As expected, integration of topological charge density over location containing meron gives the topological number of one half. (The distinction between meron and bimeron can be seen on Figure 1.8)

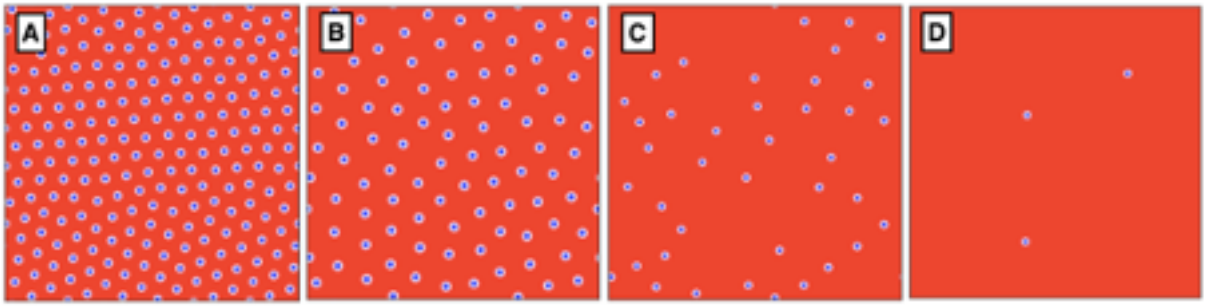

Figure 1.7: Comparison of skyrmionics gas configuration and skyrmion lattice: A) skyrmion lattice B) dense skyrmion gas C) looser form of skyrmion gas with patches corresponding to ferromagnetic state D) very loose form of skyrmion gas with only few skyrmions present.

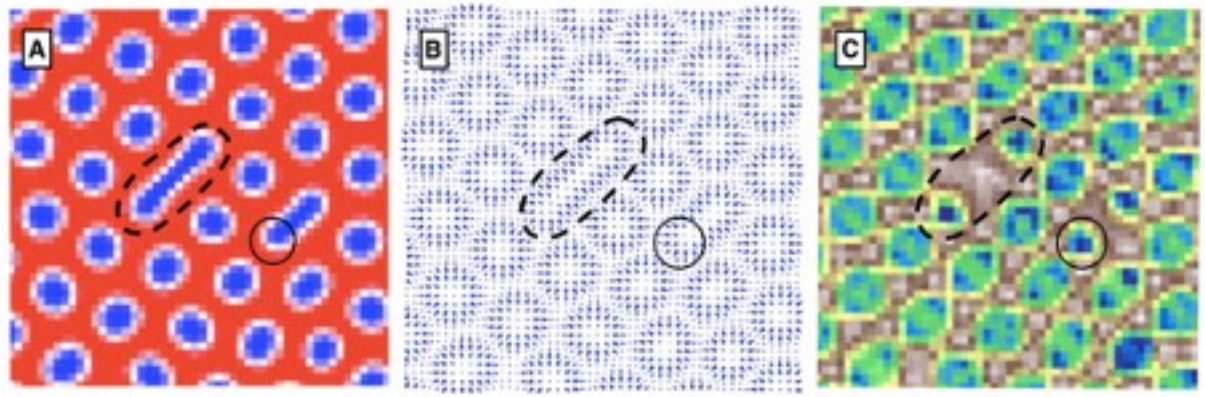

Figure 1.8 Different plots of bimerons. Dotted line shows bimeron, full line shows meron. A) plot of z-spin magnetisation. B) plot of in-plane spin projection C) plot of topological charge density.

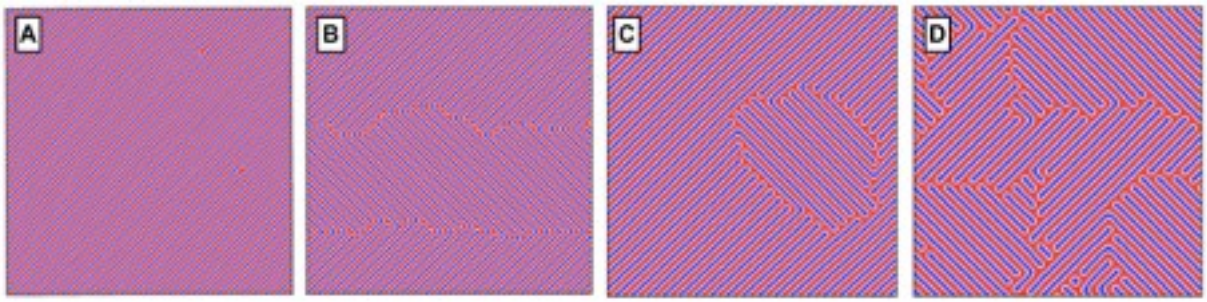

Figure 1.9 Meron-spiral interphases A) spirals with two intercalated merons B) merons forming differently oriented plate C) merons forming a rotated domain D) merons loosely organized.

Anyway, presence of bimerons (merons) in our sample is an unexpected occurrence, since these should not be present when the temperature is equal to absolute zero (Ezawa 2021). But it can be explained by the way our sample was generated: To reach the absolute zero limit, the temperature parameter in the Monte Carlo simulation was gradually decreased to arbitrary low values. But not absolute zero - that is impossible to reach. Low-temperature phases contain bimerons, though they gradually start disappearing through following mechanism described in the following paragraph.

In contrast to ordered phases of magnetic compounds present at vicinity of zero temperature, high temperature phases are characterised by uncorrelated arbitrarily oriented spins. However, as one cools the magnet, symmetries began to appear (Wetzel 2017). Specifically, as shown by Salcedo-Gallo 2020: When transitioning to skyrmionic phase, spin start to align so that individual skyrmions slowly appear, first looking blurred, but gradually smoothing out. In contrast ordering of helical phase is a non-continuous process with non-aligned helical domains appearing and vanishing until the temperature reaches low enough value to enable formation of helical phase with long-ranged order. First, we see assembly of small blurred helical islets (composed of bimerons) rotated with respect to each other. Within these islets direction of individual bimerons changes with further evolution. Finally, for temperatures close to zero, we obtain long-range order of spirals all pointing to the same direction. From this, one can derive a phonological explanation for origin of merons in our sample: Some of the intermediate domains simply remain unaligned as we stopped decreasing the temperature. (Rotated domains in interphase configurations can be seen on Figure 1.9) On the other hand, short bimerons in our sample (present within skyrmion lattices) may have appeared as result of non-perfect ordering of skyrmion lattice (bimerons are known to fill topological defects, see Iakovlev 2018.) Nevertheless, it is difficult to interpret the low-temperature limit of the Monte-Carlo simulations. For the extremely low temperatures situations Monte Carlo simulations are sometimes referred to as unreliable due to being prone to get stuck in local minima (see Kwon

et al. 2019). We will to this energy-stability dilemma, again when it gets more relevant. Let us now explain the main focus of this work.

1.5 Motivation

In this work, we will use neural networks to compress and reconstruct magnetic configuration from our sample. The motivation for what exactly we are going to do is following: Kwon et al. 2021 found out numerical approach tuned to Hamiltonian minimization favours energetically stable states. In their work, they focused on generation of new samples from learned distribution of template data. (They used 'variational autoencoder' - a type of neural network used for generation of new data, as will be soon explained.) Their template data contained topological defects corresponding to non-perfect ordering of skyrmion lattice. Without Hamiltonian, their model would start generating configurations with weird type of defects - such that wouldn't correspond to classical topological defects on account of being created as combination of defected and non-defected configuration. These weird 'intermediate' topological defects are non-physical – they have much higher energy than both non-defected and (classically) defected configurations. By using Hamiltonian, they were able to penalize energetically-disfavoured configuration, obtaining only stable states.

Now let us get back to our reconstructive neural network: Imagine our model is biased towards reconstructing the configuration so that the reconstructed configurations are close the original configuration but possessing non-classical topological defects. That would make the reconstructed data non-physical, which is for many reasons unfortunate. As physicist, we might strive to preserve physicality of data, even though there might be undesired consequences of this effort. Hence in this work, motivated by what Kwon et al. 2021 we will implement Hamiltonian in our neural network and explore whether it improved quality of reconstructed data without causing too many unwanted side effects.

2 IT background

The following is short summary a of all the concepts related to the theory of neural network which the reader should be familiar with to understand our informatic model.

2.1 General concepts

2.1.1 Operation performed by neurons

The most basic type of neural network (referred to as 'fully-connected' or 'dense' neural network – see Chollet 2018, p. 28) can be thought as a series of data transformations represented by layers of neurons. During each transformation *m* inputs x are linearly transformed to *n* values. On each of this *n* values a nonlinear "activation function" f is performed, creating *n* outputs y. We can describe this via equation (Mehta 2019, p.47):

$$y_i = f_i(w_{ij}x_j + b_i) \tag{1.1}$$

The "neuron" is the entity hosting the value $w_{ij}x_j + b_i$. In this sense, the activation function represents eagerness of the neuron to propagate its input further. The coefficients of linear transformation w_{ij} are also called "weights" of the neuron and additive constant b_i "bias" of the neuron (Chollet 2018, p. 151).

Though, theoretically any function can be used activation function, there is a limited collection of the functions that are commonly used as activation functions (Mehta 2019, p. 48). The activation functions are layer specific. There is no fixed protocol on how to choose activation function. Nevertheless, they are extremely useful. Thanks to activation functions f_i, NNs can perform nonlinear operations which greatly widens the range of possible data representation (Chollet 2018, p. 72).

2.1.2 Data processing

To enter the NN, the data must be expressed in form of a tensor with dimensions equal dimensions of the first layer (called the "input" layer) of the NN. Each neuron acts on one data point. After performing the transformation according to (1.7), the data is transferred to the second layer. In this manner the data move through the successive "hidden" layers until reaching the "output" layer.

The character of the output depends on what the NN is constructed to do. In NNs used for the classification (called the classifiers), the ultimate outputs describe the probabilities that the data

belong to specific category. (Individual neurons in the final layer represent the categories; their outputs the corresponding probabilities, Mehta 2019, pp. 29-30) In NNs used for image reconstruction – called the "autoencoders", the final output has the same dimension as the original input and should be as close to it as possible. Related form of NNs - the variational autoencoders - attempts to construct an output belonging to same normal distribution as the input. In this way, we create new data with same properties as the original. (For a beautiful introduction to autoencoders and variational autoencoders, see Chollet 2018, pp. 296-305.)

2.1.3 Loss function

For the network to best achieve its intent, the values w_{ij} and b_i must be tuned to optimal level. For this purpose, the NN undergoes the training. During the training, the NN processes sample data (as part of the "training" dataset) and adjust the w_{ij} and b_i parameters until its output reflects the desired output. The degree of such correspondence is evaluated through the "loss function" (with a low "loss" being an indicator of a good correspondence). Depending on the type of neural network, the loss function can take various forms. For the autoencoders, the most basic type of loss function is a mean squared error (MSE) calculated between input and output. In this case, low loss means the output is reconstructed as a close replica of the input.

2.1.4 Supervised vs unsupervised learning

Depending on the character of the desired output, we differentiate supervised and unsupervised learning (Chollet 2018, p. 94). In unsupervised learning, the training process relies solely on the input data without requiring any additional information. In other words, the character of desired output is fully determined by the input. This is how the training of the autoencoders works – since the input is the same as the output, knowing our input we immediately know what our desired output is. On the other hand, to enable the supervised learning, we must supply the NN with the information about the desired output. This is how the training proceeds for classifiers: Unless provided with the labelled data, the NN wouldn't learn how to classify them.

2.1.5 Gradient descent

The training consists of multiple iterations – epochs. During each epoch, fixed number of data (the "batch", Chollet 2018, p. 96) is fed in the NN. For this batch, the loss function is calculated, and gradient descent method is used to update the values of w_{ij} and b_i. Though the gradient

descent methods for NNs all rely on "backpropagation" technique, there can be small differences in the implementation depending on which "optimizer" is chosen.

As we know, gradient descent is often problematic as it can lead to local, not global minimum. For NNs this threat is to large extent mitigated calculating the gradient for multiple subsets of the data separately, therefore introducing stochasticity. (This is main idea behind working with batches, see Mehta 2019, p. 15). Nevertheless, there are also different ways how to introduce the randomness in the training. For instance, in each iteration we can randomly choose neuron whose outputs would be silenced (called "drop-out", Mehta 2019, p. 52). This would prevent NN from over-relying on constrained number of neurons and give all neurons the opportunity to show their potential for learning. Another way how to achieve this is through "batch-normalization" which controls the outputs f_i, preventing them from reaching unreasonably high values (Mehta 2019, pp. 52-53). Similarly, the magnitude of w_{ij} and b_i can be controlled by adding regularizer term ($\lambda_1\|w\|+\lambda_2\|b\|$) to loss function. (See Chollet at al. 2015, "Layer weight regularizers".)

2.1.6 Validation set and overfitting

In previous we established that the training uses gradient descent method to minimize the loss computed over the training dataset. As one intuitively understands, a prolonged training might lead to w_{ij} and b_i values ending up overfitted, with the NN best prepared for the input encountered during the training. To prevent such scenario, we introduce a "validation" dataset. Throughout the training, the value of loss is calculated not only for the training dataset but also for the validation dataset. In this way, for each epoch we obtain a "training loss" and "validation loss". The former is used for gradient descent; the latter is used to assess prevent overfitting. During the training, the validation loss should not significantly exceed the training loss. In fact, validation loss rising above training loss can be taken as an indication the training is completed. Alternatively, the training can be stopped when the loss seems no longer decreasing. In theory, this should indicate that the performance of the NN would not improve significantly even if the training session was extended. In praxis, this is not guaranteed – sometimes the loss can change abruptly, even after longer periods of remaining unchanged.

2.1.7 Comparing the performance of different NNs

The performance of the NN is assessed by calculating the loss over the validation set, after training is completed. This gives us an estimate for how well NN achieved its intent on an

unfamiliar dataset. (However, in case when the validation loss was used to stop the training, validation set can no longer be perceived as unfamiliar. In such scenarios, one needs to split the dataset to three parts and use only the third part, the so called 'test' dataset for evaluating the performance. (Chollet 2018, pp. 97-98)). Assessing the performance-based validation loss might be biased if the validation set contained only non-challenging (or alternatively too problematic) data. To prevent this from occurring, one may repeat the training so that different validation dataset would be used for each training. Usually, one does this through dividing the dataset to multiple parts (so called "folds") and repeats the training so that each time a different fold will be used as the validation set (with the remainder of the folds merged together and used as the training set). This means that all the data would be at some point included in the dataset, preventing the chance the problematic data will get overseen. However, it wouldn't erase the bias completely since it may still happen that problematic and non-problematic data get coupled together and influence the performance of NN in a nontrivial way. Therefore, one always tries to work with as many folds as the computational cost allows. This form of assessing the performance of NN is called k-fold cross validation; with k standing for the number of folds (Chollet 2018, pp. 87).

2.2 Convolutional neural network

The neural networks are divided to different subclasses according to how the neurons are interconnected. We dense neural network was already describe in the beginning of this section as the simplest type of NN. Now, we will describe the type of NN this work uses. It is called "convolutional neural network" (CNN) work.

2.2.1 Convolution

To illustrate how the CNN works, let us suppose that we have input with represented with three-dimensional tensor of size A_1xA_2xA_3. Let us denote its coordinates *x, y* and *z.* Now suppose we have tensor of size S_1xS_2xA_3, called the "filter" (Mehta 2019, p. 57). Now let us do the following convolutional procedure (the "convolution" as described in Mehta 2019, pp. 56, 57, for illustration see Figure 2.1): From the input tensor, a sub-tensor with size equal to filter size is chosen. Point multiplication followed by contraction is performed, resulting in scalar value. To this scalar, we would add bias *b* and let activation function acting on the result. Next, we would choose another sub-tensor, just adjacent to previous one and perform multiplication, contraction and scalar transformation again (with the same bias *b* as in the previous). We would repeat this

process until the whole input tensor is mapped into matrix of scalars. The size of the output matrix depends on our stride – how far we travelled when moving to next sub-tensor. In the simplest case when we moved just by one point at the coordinate axis, the size of the output would be almost equal to A_1xA_2 – the would be slight size reduction because of effects at boundaries. We would deal with this padding the boundaries with zeros. Now let us suppose that we don't have just one filter, but assembly of N such filters. If we applied them one by one on the input tensor we would obtain an output with size S_1xS_2xN. In the nomenclature of CNNs, the "kernels" are matrices of size S_1xS_2 (and usually S_1=S_2); the assembly of filters is our "convolutional layer" (Chollet 2018, pp. 122-128). The different filters are expected to capture different non-trivial features of the data. Hence, the process of applying filters is termed "feature extraction". In this case described the total number of trainable weights w for such convolutional layer can be calculated as A_3xS_1xS_2xN, total number of trainable biases b would be N, with every filter having its own bias.

2.2.2 Architecture

With the autoencoders, we are interested in creating alternative representations of the original data. The process rescaling the data to our desired representation is called "encoding". After encoding the data are represented in the "latent space". The process of reconstructing the data from latent space back to the original is referred to as "decoding". (Terminology as in Chollet 2018). For encoding of three-dimensional input, we most of the time try to reduce the first two dimensions (A_1 and A_2) while increasing the third (A_3). Of course, very popular usage of autoencoder is image compression which requires the total volume A_1xA_2xA_3 to decreasing to smallest possible values. In the physical problems, we have yet another usage for the autoencoder - we be interested in knowing how would the topological features project on a smaller space. In this sense, the filters can be thought as the operators trying to capture different properties of data, distributed across the space in the non-trivial way.

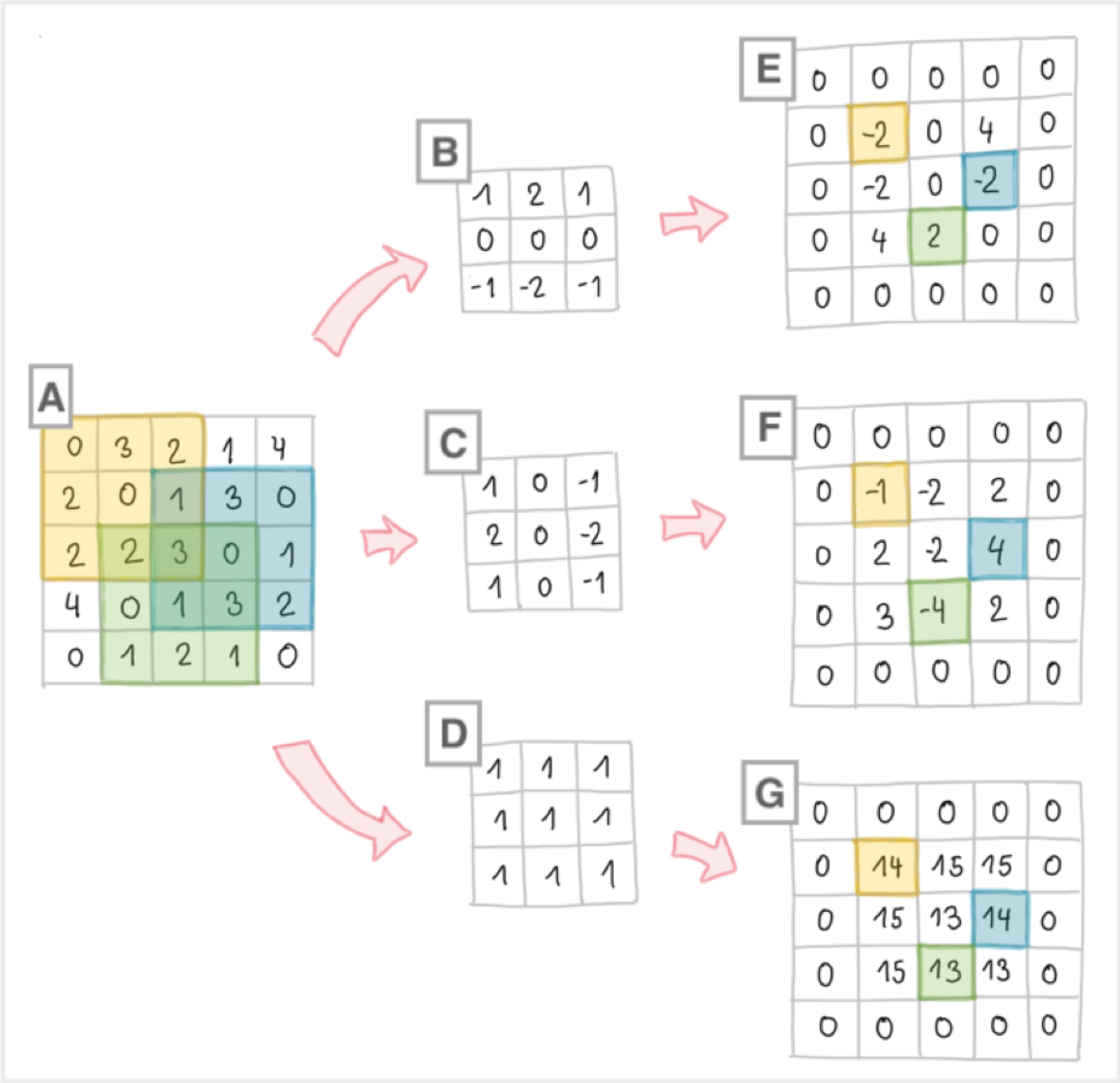

Figure 2.1: Illustration of the convolutional procedure. In this case, the initial tensor (A) is the tensor of size A_1xA_2xA_3=5x5x1. On this tensor we apply the three filters (B), (C), (D) of sizes S_1xS_2xA_3=3x3x1. In each step of the convolutional procedure, we select a subtensor from the matrix A and multiply is with one of the filters, placing the resulting value in the designed position. After performing this process for all available subtensors, we obtain the matrices (E), (F) and (G) which altogether form tensor of size 5x5x3. As we see, the boundaries of the matrices (E), (F) and (G) are padded by zero, because the convolution would otherwise reduce the dimensions.

2.2.3 Handling dimensions

Now let us get back to our problem of dimensionality reduction. We know that to increase the third dimension (A_3) we can make the number of filters larger than the depth of the input: $N>A_3$. (The encoder is usually constructed in such a way that the number of filters is gradually increasing for successive convolutional layers.) To decrease the dimension (A_1 and A_2) we can perform convolutions with filters strolling through the input at higher pace, leaving spaces between sub-tensors. Alternatively, we can use a “pooling layer”. For a pooling layer, we also

specify the kernel size S_1xS_2. In this case, it determines the size of non-overlapping patches to which the input will be divided. In the maxpooling layer, the output represents the maxima of the patches; in the average-pooling layer, the output represents averages over the patches (Chollet 2018, p. 128-129). The most common values of pooling kernel sizes include 2x2 and 3x3, higher sizes are believed to lead to informational loss. (Nagi (2014)) In the decoders, the convolutional and pooling layers are often used in an alternating fashion. (Chollet 2018, p.133) In this way, after each dimensional reduction, the spatial information loss can be compensated be finding new properties characterizing the data: thanks to dimensional reduction we can find the correlation between more and more distant topological features.

Quite obviously, to get from the latent phase representation back to the original dimensions of the data, we should increase the first two dimension A_1, A_2 while decreasing the depth A_3. We can solve the latter via successive application of convolutional layers with gradually decreasing filter numbers. To increase A_1 and A_2, we would use "upsampling" layer which copies each input value into patch of size S_1xS_2. Most of the time S_1xS_2=2x2. In terms of dimension transformation, the decoder can be, and often is, constructed as the mirror image of the encoder.

3 Methods

3.1 Different phases in our dataset

Our dataset contains 600 configurations, each represented by the 40000 distinct spin values carried on 200x200 grid. Out of these 600 configurations, 138 contained skyrmion lattices (for simplicity also called 'skyrmion phase' or just 'skyrmions'), 110 consisted of spirals, 144 showed ferromagnetic alignment, 13 represented skyrmion gas and 205 hosted some form of bimerons.

3.1.1 Subdivision of bimeron-containing interphase

Since bimeron-containing interphase was very non-homogenous, it was decided that it would be split into multiple separate categories. First one was 'skyrmions with few bimerons' where there was only handful bimerons. There was 24 of such configurations. Second was 'skyrmions with some bimerons' where there was enough bimerons to enable their uniform distribution across the configuration, yet there were still large patches containing no skyrmions. There was 43 of such configurations. Third is 'skyrmions with many bimerons'. In contrast to previous category, skyrmion-free patches are small. There was 15 of such configurations. Fourth is 'bimeron with many skyrmions' phase in which there is same number of skyrmions as bimerons. There was 21 of such configurations. Fifth is 'bimerons with few skyrmions', where the configuration is mostly composed of bimerons with few intercalated skyrmions. There was 7 of such configurations. Sixth 'spirals with bimerons' phase where there are bimerons as wells as spirals present. There was 80 of such configurations. Finally, seventh is 'spiral with merons' which consist of spirals with few of them capped by merons. There was 5 of such phases. Figure 3.2 shows all the bimeron-containing interphases that we just defined. Figure 3.1 shows all the phases (intermediate as well as pure) that were defined and identified in our sample.

3.1.2 Alternative subdivision of bimeron-containing interphase

The division of bimeron-containing phase to seven distinct subclasses was performed prior obtaining the results. As we will later see, based on reconstruction error calculated for Hamiltonian-containing autoencoder, the bimeron-containing interphase can be divided to three different subphases. To aid clarity in later parts of this work, these three subphases will be introduced now. First is the 'skyrmion-bimeron' phase. It contains data belonging to 'skyrmions with few bimerons', 'skyrmions with some bimerons' and 'skyrmions with many bimerons'

phases. Hence it hosts all the bimeron-containing configurations in which there are visibly more skyrmions than bimerons. Second is the ‘bimeron-skyrmion’ phase. In contains data belonging to ‘bimeron with many skyrmions’ and ‘bimeron with many skyrmions’. Hence it hosts all the mixtures of skyrmions and bimerons in which the number of skyrmions is less or equal to number of bimerons. Finally, the third ‘bimeron-spiral’ phase contains the ‘spirals with bimerons’ and ‘spiral with merons’ configurations.

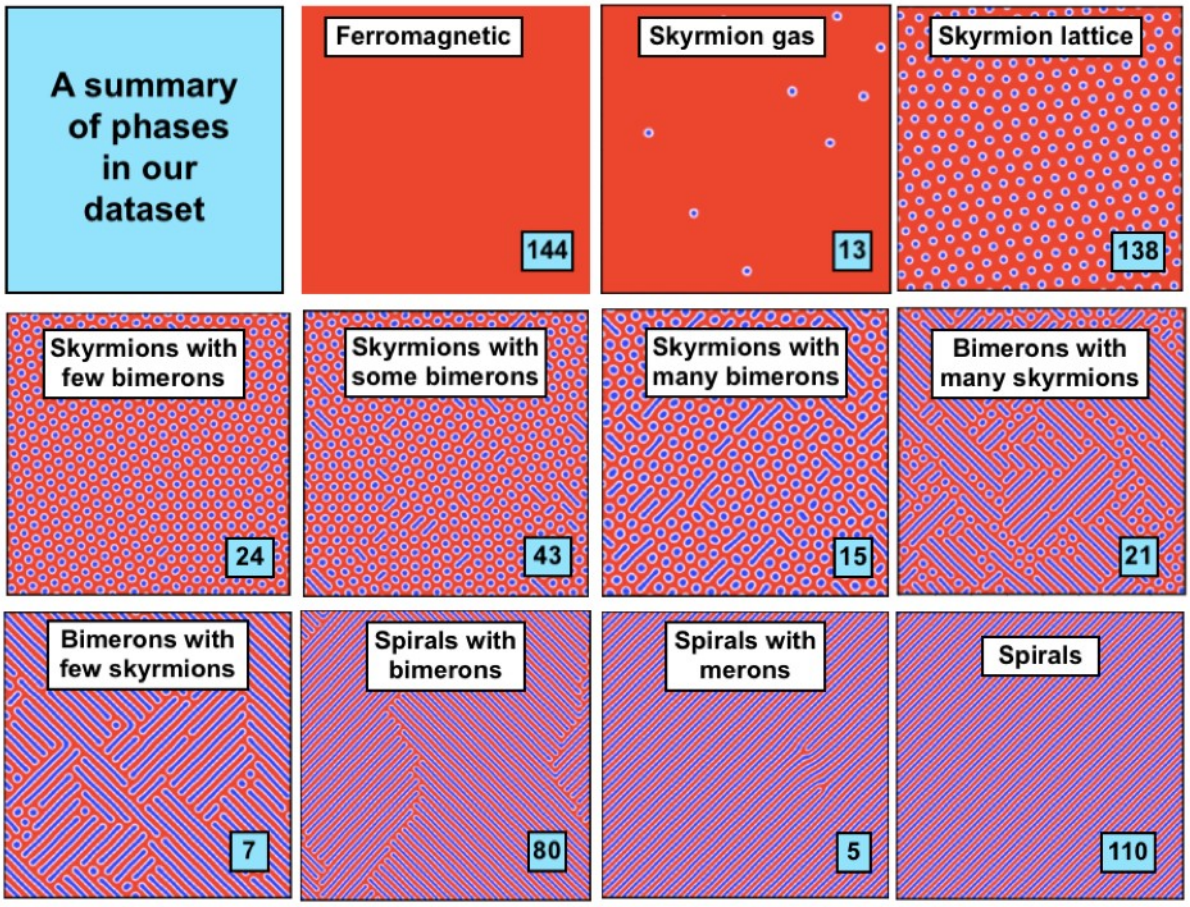

Figure 3.1 A summary of all the phases present in our dataset.

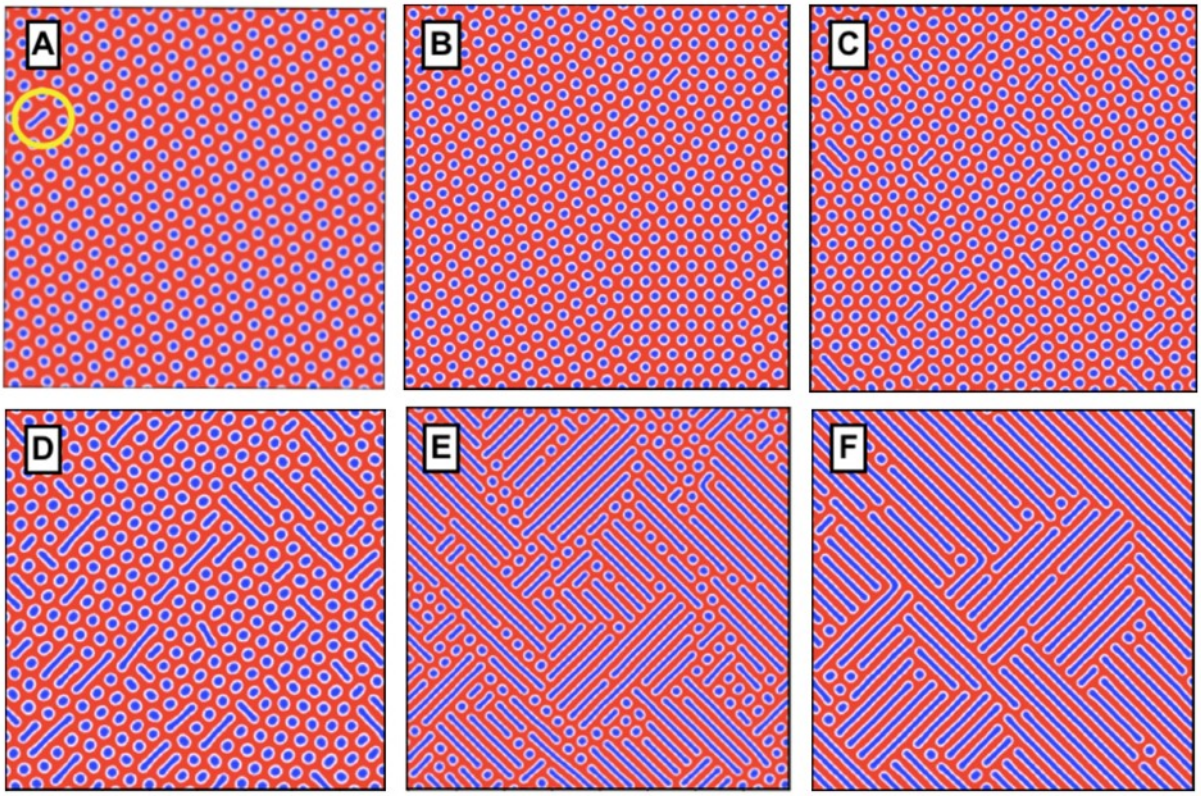

Figure 3.2 A summary of skyrmion-bimeron intermediate phases in our dataset: A), B) skyrmion with few bimerons (a solitary bimeron is shown in a yellow circle). C) skyrmions with few bimerons D) skyrmions with many bimerons E) skyrmion-bimeron F) bimerons with few skyrmions.

3.2 Obtaining the dataset

The assembly of different magnetic configuration was obtained through performing Monte Carlo simulations for differing values of D and h: For different values of D and h, different magnetic phase was thermodynamically favoured. The exact relationship between D, h and identity of magnetic phase, the 'phase diagram', can be seen on Figure 3.3. It is worth to note that the values of D and h were part of dataset – for each configuration there was a supplementary information about the D and h values from which it was obtained. It could not have been done otherwise, since the neural network needed this information to calculate Hamiltonian through equation 1.6. To prevent confusion, it is necessary to stress that the D and h values were not part autoencoder input - they were not part of the data intended for reconstruction. They entered the network through completely different path, as part of the data, the network used calculate loss function.

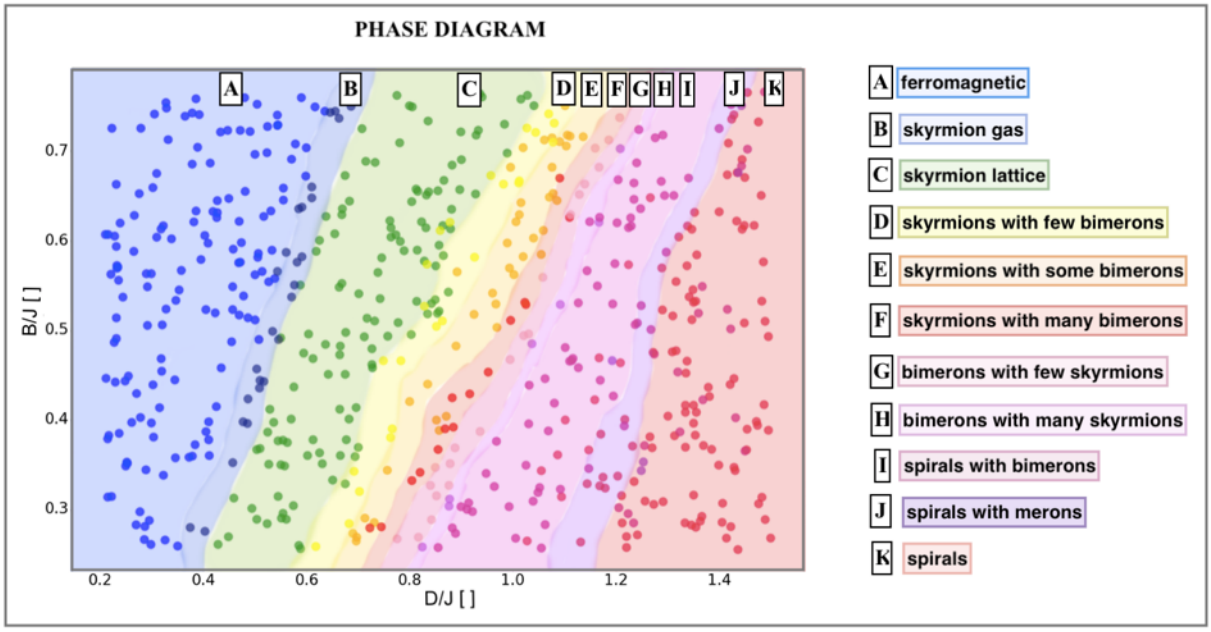

Figure 3.3 The phase diagram

3.3 Skyrmions in our dataset

Let us get back to problematics of different phases: Though in different sense than bimeron-containing interphase, skyrmionic phase was also quite heterogeneous. It contained skyrmions of various sizes and densities. To better address this, the distribution of skyrmionic properties was described and several relevant properties were developed.

In the 138 configurations of skyrmion lattices in our sample, the number of skyrmions within the 200x200 grid varied from 116 – 607 with mean 333. Most numbers fell within the range 200-450 where the distribution can be approximate as uniform. This distribution is so wide because the size of the skyrmions as well as the distance between skyrmions varied. To describe this better, new variable called 'inter-center distance' was defined as the distance between skyrmions d_c measured from the midpoint of one skyrmion to midpoint of adjacent skyrmion. In our dataset the value of inter-center distance ranged from 8 to 22 grid points with most of the values in the interval 10-12 grid points. To derive an explicit formula for inter-center distance, and various correlations were explored. (Corresponding plots showing correlations can be found in). It was found that inter-center distance is best described via equation (3.1).

$$d_c = \frac{total\ number\ of\ grid\ points}{number\ of\ skyrmions} \cdot 0.042 + 6.544 = -\frac{40\,000}{\mathcal{T}} \cdot 0.042 + 6.544 \qquad (3.1)$$

Alternatively, we can measure the distance between edges of skyrmions, the 'inter-edge distance' d_e. In theory d_e should be for same d_c lower for big skyrmions and greater for small

skyrmions. In our sample, d_e is estimated to fall within the 3-12 grid points, with most of the values around 4-5 grid points. Inter-edge distance d_e is well characterized by (3.2) which was obtained in very similar way as (3.1). The total magnetization is the value one obtains by summing all the spins S on the grid. As it would be expected d_e is also correlated with d_c, but this correspondance is weaker than what one obtains with (3.1) or (3.2).

$$d_e = \frac{total\ magnetization}{number\ of\ skyrmions} \cdot 0.040 + 1.924 = -\frac{\sum S}{\mathcal{T}} \cdot 0.042 + 6.544 \tag{3.2}$$

Last property of skyrmions, we can be interested in knowing, is their size. We can distinguish size of whole skyrmion and the size of the inner core. The size of the whole skyrmions ranged from 30 to 70 grid points, with most values estimated to be around 40. Size of the part representing negative values, the inner core, ranged from 21 to 45 grid points.

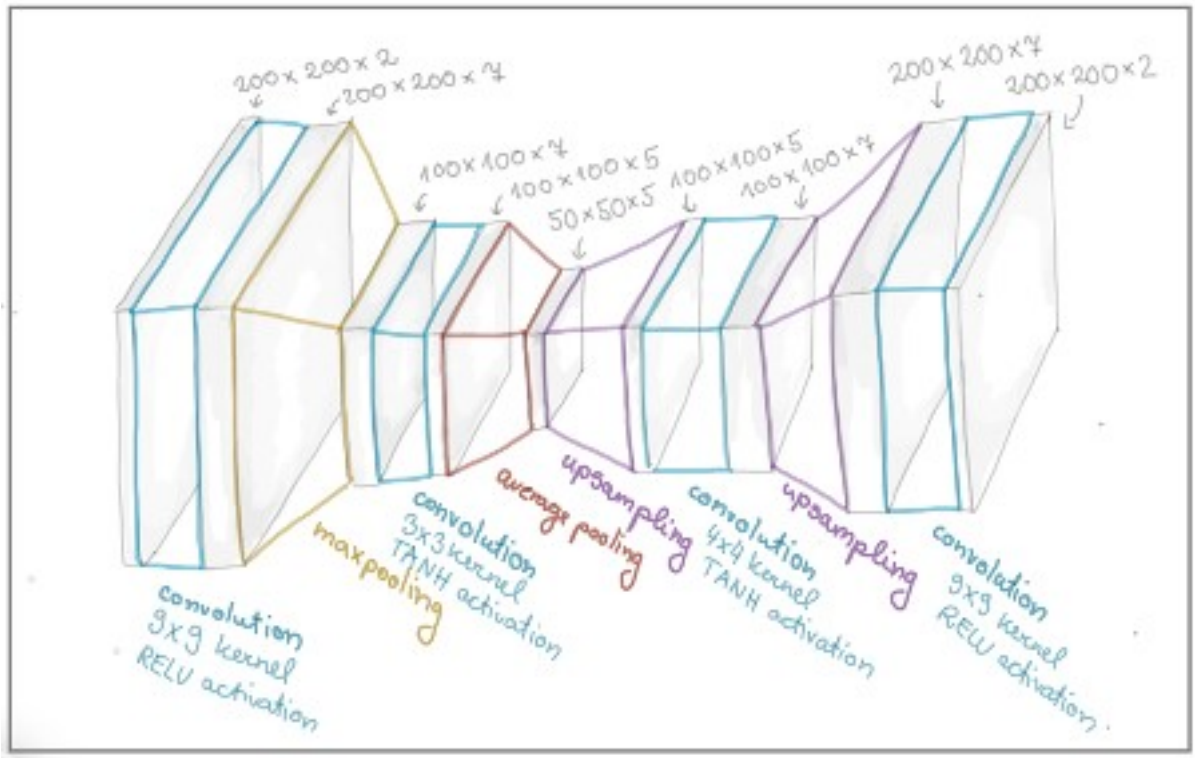

Figure 3.4 Architecture of the four-fold reducing autoencoder.

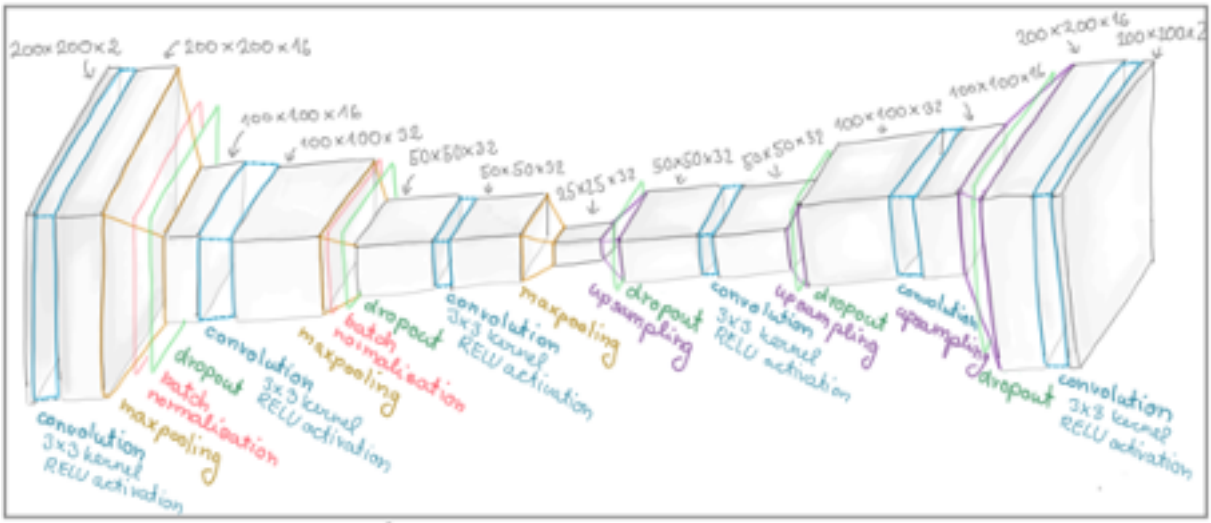

Figure 3.5: Architecture of the eight-fold reducing autoencoder.

3.4 The autoencoders used in this work

Knowing such microscopic properties is important for assessing limitations of dimensional reduction. Suppose we work with autoencoder which decreases the input grid with dimensions 200x200 eight-times, so that in the latent space representation the grid is of size 25x25 and is represented by 625 grid points. As discussed previously, highest number of skyrmions in our dataset is 607, so on the 25x25 grid, each skyrmion will be represented with approximately one grid point. This does not necessarily mean that we would face significant information loss – we can still have the third dimension of the latent space compensating for this reduction. In case of the regular skyrmion lattices, the feature-extraction path can lead to filters capturing lattice symmetry. Nevertheless, some of the bimeron-containing interphases are also very densely packed but possess little symmetry, making the feature-extraction difficult.

From this we see that eight-fold dimensional reduction is a difficult task. It turns that designing four-fold reducing autoencoder that able reconstruct the configurations well (so that the visual appreciation cannot reveal any faults) is an easy task. In contrast, eight-fold reducing autoencoder that able reconstruct the configurations reasonably well is so far unavailable. In this work, we will try implement Hamiltonian term in eight-fold reducing autoencoder with hope to improve its performance.

Preliminary experiments showed that varying the parameters of the network (such adjusting the filter sizes and filter number or changing the activation function) had not as significant effect on the autoencoder performance as did the scaling. The focus of this how the Hamiltonian term influences performance and not in finding best working autoencoder, so choice of specific models was not critical. Hence, we will use two different autoencoders, one eight-fold reducing

(for implementation of Hamiltonian) and one four-fold reducing (for assessing the significance of trends observed), without giving reasons for why their architecture was designed the way it was.

The complete set of parameters for these autoencoders (including kernel sizes, filter sizes and activation functions) is shown on Figure 3.4 and Figure 3.5. The latent space of the four-fold-reducing CNN had dimensions A_1xA_2xA_3=50x50x5, for the eight-fold-reducing it was 25x25x32. Both the autoencoders used ADAM as the optimizer. The batch size was always set to 8 and number of epochs to 200. As for the loss function, the default form is (3.3). (Nevertheless, when trying to assess the effect of Hamiltonian inclusion, the loss function will be modified in the manner that would be specified later.)

$$L = \sum (S_{x\,\mathrm{true}} - S_{x\,\mathrm{pred}})^2 + (S_{y\,\mathrm{true}} - S_{y\,\mathrm{pred}})^2 + (S_{z\,\mathrm{true}} - S_{z\,\mathrm{pred}})^2 = \sum (S_{\mathrm{true}} - S_{\mathrm{pred}})^2 \tag{3.3}$$

In the above expression, L stands for loss and S represent individual spin components. The subscript 'true' means that we refer to the value of the original spin (input), the subscript 'pred' stands for predicted and it means we refer to value of the spin obtained after the reconstruction (output): Our loss is basically the mean square error of the spin matrix. It should that the input of the CNNs is not in form of x, y and z spin components. Instead, we take advantage of the fact the spin as a vector has a constant (unitary) magnitude and represent the spin in spherical coordinates via (normed) azimuthal vector φ and the polar angle θ, reducing the dimension from 3 to 2. Hence when calculating the loss from (3.3), the NN first needs to take perform transformation from spherical to Cartesian coordinates.

Since the direction of the spins corresponds to system magnetisation, squared difference of spins as calculated by the expression (3.3) would in this work be referred to as 'magnetic MSE', 'magnetisation MSE', 'magnetisation error' or simply 'reconstruction error'.

The total number of configurations in our dataset is 600. We split them in a ratio 9:1 to training and validation dataset. When performing cross validation, we have 10 folds, so the sizes of validation and training dataset correspond to this splitting ratio. The size of the input of the CNNs is 540x200x200x2. This means we have training dataset with 540 configurations modelled on a 200x200 grid, so there are 200x200 spin values determined by 2 angle values.

3.5 Comparing four-fold reducing and eight-fold reducing autoencoder

Having discussed the features of our dataset, we can proceed to compare the performance of the two used autoencoders (four-fold reducing vs eight-fold reducing). Knowing that the performance of the four-fold reducing autoencoder is what we strive to achieve, we will later use this comparison to assess whether the inclusion of Hamiltonian changed the performance of eight-fold reducing autoencoder in a significant way.

In previous, we said that instead of comparing the performance of NNs through loss, we will use other metrics. In this case, we will explore the in-plane magnetisation MSE (denoted MSE_{in}) and out-of-plane magnetisation MSE (denoted MSE_{out}) as defined through expressions (3.4) and (3.5). Since our system possess rotational symmetry, MSE of x-spin components and MSE of y-spin components are equivalent and therefore are evaluated together. On the other hand, MSE of z-spin component might behaved differently and therefore is evaluated on its own.

$$MSE_{in} = \sum (S_{x_{\text{true}}} - S_{x_{\text{pred}}})^2 + (S_{y_{\text{true}}} - S_{y_{\text{pred}}})^2 \tag{3.4}$$

$$MSE_{out} = \sum (S_{z_{\text{true}}} - S_{z_{\text{pred}}})^2 \tag{3.5}$$

Figure 3.6 shows the values of mean in-plane MSE for different phases (mean in-plane MSE is obtained by calculating in-plane MSE for each configuration separately and then taking mean over each phase). Figure 3.7 shows this for out-of-plane MSE. As we see for ferromagnetic phase and for skyrmion gas phase, the MSE is very low for both four-fold reducing and eight-fold reducing NN. This is exactly as expected since for these two phases the configuration is (mostly) composed z-axis aligned spins with negligible in-plane projections. (Hence there are almost no opportunities to fail the reconstruction). For the four-fold reducing autoencoder, the in-plane MSEs of all other phases have all very similar values. For the eight-fold reducing autoencoder the mean MSEs are a lot lower for skyrmions and spirals than for the intermediate phases. Nevertheless, if one compares the total MSEs, one will find out that by far the biggest contributor to the total MSE is the spiral-bimeron phase. (Although, arguably if one considered all the skyrmion-bimeron interphases as one, one would obtain similarly high contribution.) The trends in the mean and total out-of-plane MSE (shown on the Figures 3.6 and 3.7) were similar to trends of the in-plane MSE.

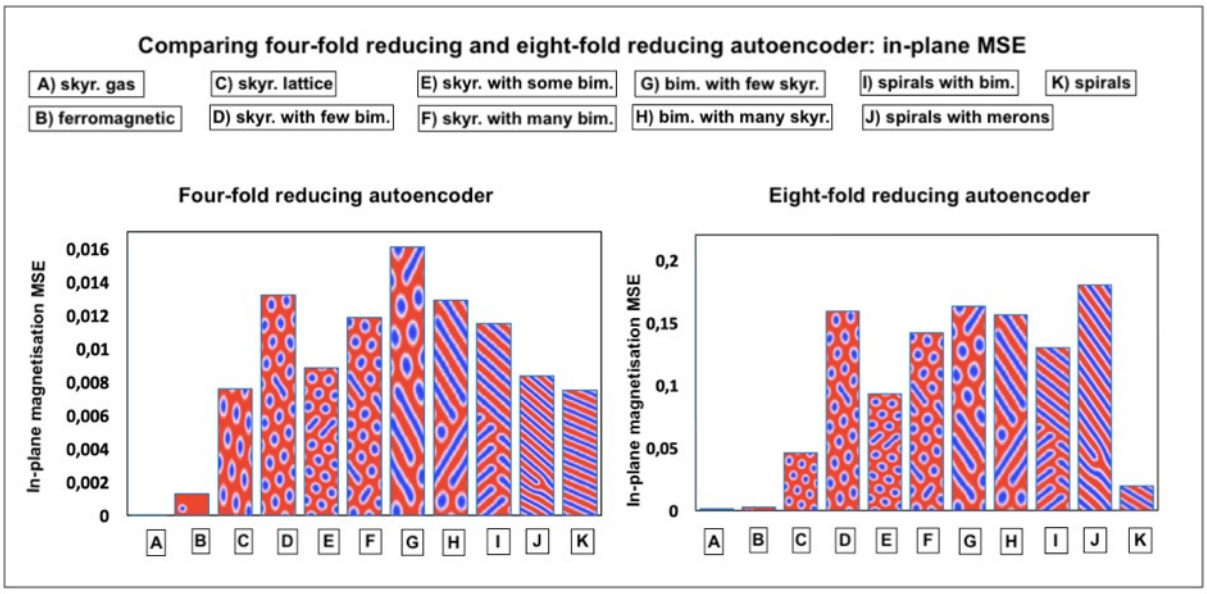

Figure 3.6 Comparing performance of four-fold and eight-fold reducing autoencoder for different phases: In-plane MSE

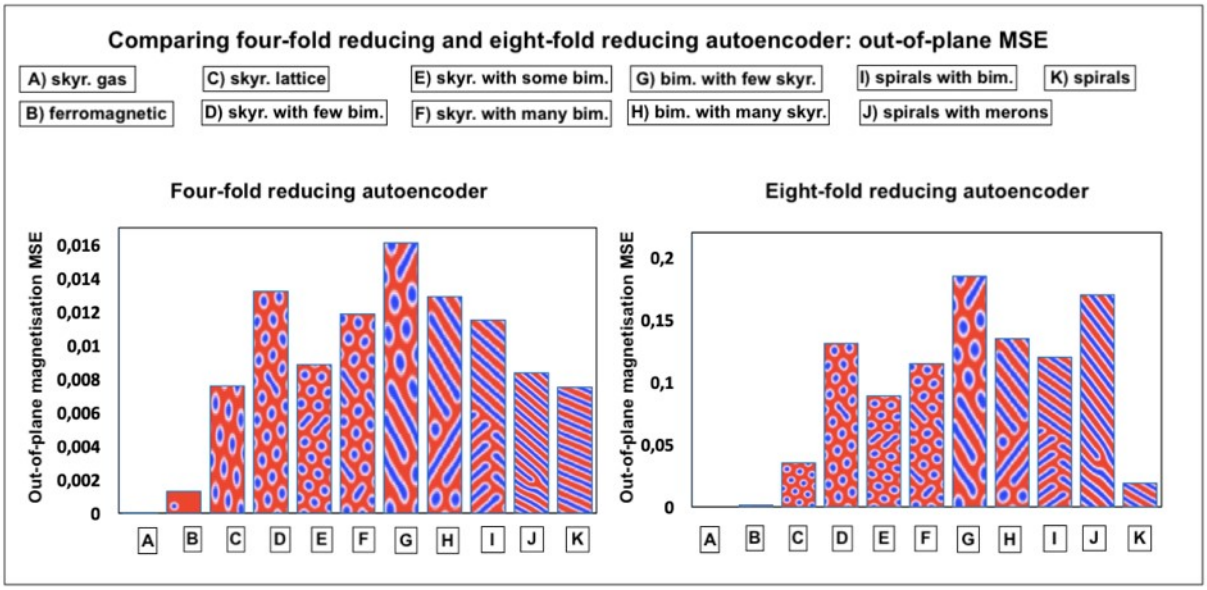

Figure 3.7 Comparing performance of four-fold and eight-fold reducing autoencoder for different phases: Out-of-plane MSE

4 Implementation of Hamiltonian term

4.1 Initial considerations

Before embarking on data collection, it needs to be defined how exactly the Hamiltonian will be implemented in the loss.

4.1.1 Experiment

If we decided to follow the approach of Kwon et al. (2021) we would just supplement the loss function with a term containing Hamiltonian as calculated via expression (1.6). In this way, the energy of the resulting configuration should be minimised. Nevertheless, to better understand how this would influence the training of the CNN, small preliminary experiment was done. Specifically, it was investigated what would happen if the loss function contained only the Hamiltonian term without any contribution of magnetic MSE. This means that the loss had form (4.1), where H was calculated from the expression (1.6).

$$L = H \tag{4.1}$$

It would be impossible train the network in a set-up like that. To overcome this obstacle the network was first trained on the four-fold reducing autoencoder with classical form of loss (expressed by the equation (3.3)). In this way, the network's trainable parameters (the weight w_{ij} and biases b_j values) got adjusted at the optimal level. Then autoencoder with Hamiltonian as loss function was trained, inheriting initial values of w_{ij} and b_j from the trained parameters of the four-fold reducing autoencoder. When the autoencoder trained in this way was applied on our dataset, it reconstructed spirals and ferromagnetic phase in a same way as the four-fold-reducing autoencoder: for the spirals and ferromagnetic phase the performance wasn't influenced by the subsequent training. The story was different for skyrmionic lattices: one epoch of training sufficed in making the autoencoder reconstruct the skyrmions in slightly different way – they got smaller and their shape got regularized (as shown on Figure 4.1, part D) . In the next few epochs, skyrmions gradually disappeared – the autoencoder reconstructed skyrmionic phase as a ferromagnetic phase.

The previous only held for pure skyrmionic lattices. In configuration containing skyrmionic intermixed with bimerons, the skyrmion is got smaller, but never started disappearing, not even after the number of epochs was set to some arbitrary large value like 1000. In contrast, bimerons got progressively smaller and gradually (over the period of around 30 epochs) vanished. This

means that Hamiltonian-trained autoencoder erased the bimerons while leaving the skyrmions intact (as shown on Figure 4.1, part B) Similarly, for spirals containing merons, merons got deleted, but spirals did not. Reading this, one is immediately tempted to try using Hamiltonian-trained autoencoder as a selective bimeron detector with bimeron presence determined through change in total magnetisation (which would decrease after deletion of bimerons). Unfortunately, exploring such idea to sufficient detail is beyond the scope of this work.

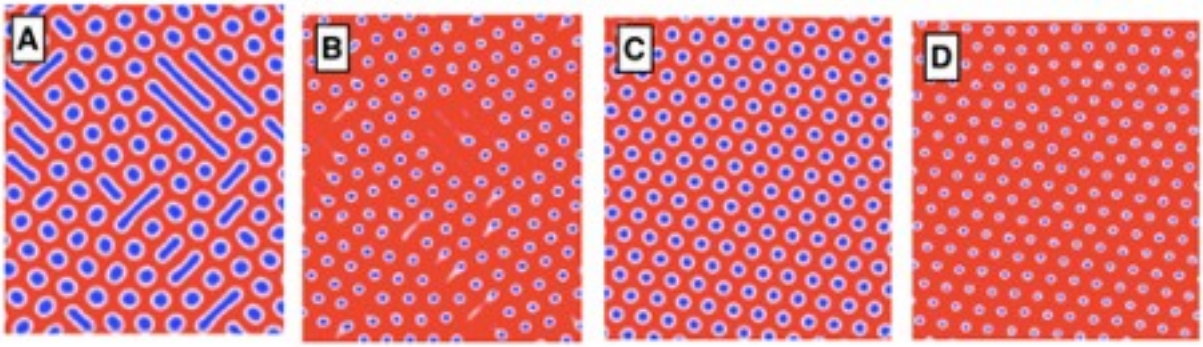

Figure 4.1 Disappearing bimerons. A) original plot of bimeron-containg configuration B) bimeron-containg configuration after reconstruction C) original plot of skyrmion-lattice configuration D) skyrmion-lattice configuration after one epoch.

4.1.2 Explanation

Nevertheless, let us at least try to explain these intriguing results. For this purpose, it was explored which of the phases represents the energetic minimum in the various places on the phase diagram. To be clearer, for each of the eleven observed phases, n configurations were chosen whose D and h values were used to calculate Hamiltonian for all available configurations: In such paradigm Hamiltonians were not calculated for particular configuration and its corresponding D and h values. Instead, D and h values were kept fixed (at nx11 chosen values), while the configuration supplemented to Hamiltonian alternated, until all the 600 available configurations were processed. In this way, we should be able to determine which of the observed phases (as represented by a pool of the configurations) embodies energetic minimum for particular D and h values.

Though n was taken as n=10, out of these 10 samples, one representative was chosen to be shown on Figure 4.2-4.4. As we can see, the results were following: It was found that for D and h values corresponding to ferromagnetic, skyrmion gas and skyrmion lattice phase the energetic minimum is represented ferromagnetic configurations. This means that if all skyrmions in our sample got deleted energy would be released – which is no doubt the reason why Hamiltonian-trained autoencoder performed such deletions. On the other hand, for the skyrmion-bimeron phase (containing mixture of skyrmions and bimerons with prevalence of skyrmions) the

energetic minimum is represented trough skyrmion-lattice phase. Again, this explains why Hamiltonian-trained autoencoder erased bimerons but left skyrmions intact. For remainder of bimeron-containing phases, energetic minimum is represented through mixture of spiral, skyrmion and bimeron-containing phases. There is no straight-forward interpretation of this. Finally, as it would be expected, for pure spirals the energetic minimum is represented through spirals.

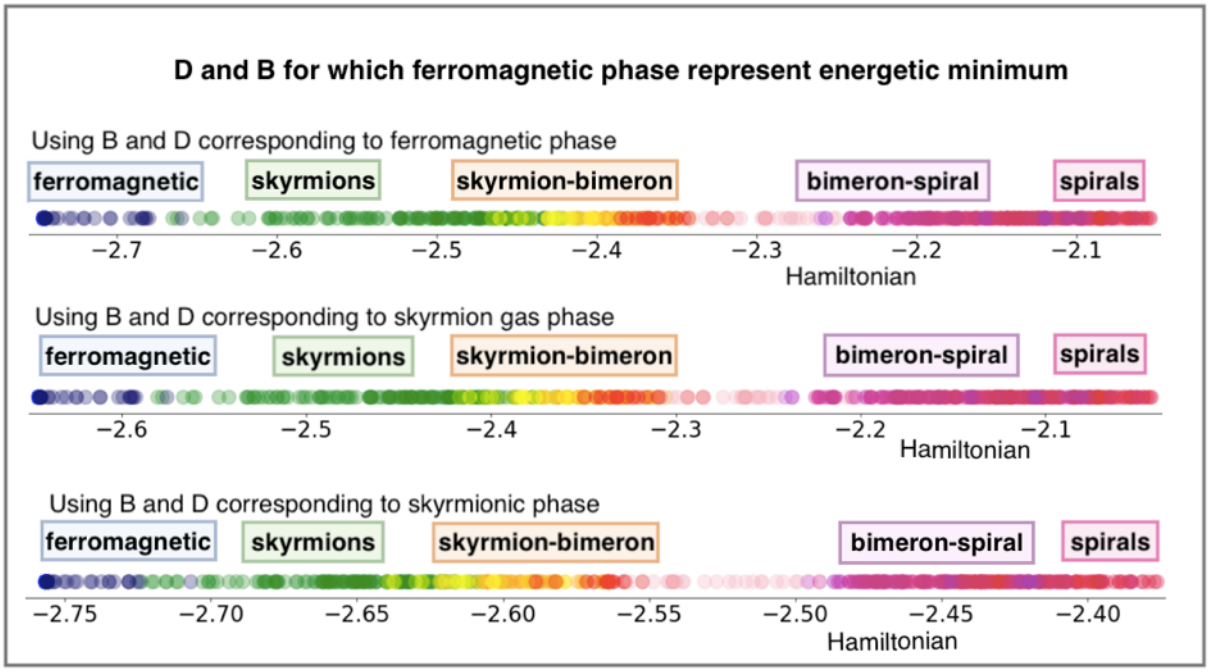

Figure 4.2 Calculating Hamiltonian for D and B of phases for which ferromagnetic phase represents the minimum. For each phase, from the pool of five plots one representative was chosen.

4.1.3 Minimal energy

Contemplating these results, one may get confused as to why Monte-Carlo simulation provided configuration which do not represent energetic minimum for the target value of D and h. Nevertheless, the minimum energy state might not necessarily represent the thermodynamically most favoured state, not even if one approaches absolute zero temperature. This is obvious if one realises that for higher temperatures, states with higher energy are favoured and therefore would be assumed by the system. After subsequent cooling, the system might not have enough energy to undergo phase transition and hence might remain in its current phase, even though when it does not represent energetic minimum. As an illustration of this, let us take an example of skyrmions: Transitioning from skyrmionic to different phase is associated with non-zero energetic barrier. Therefore, skyrmions display hysteretic behaviour (Makino et al. 2017) and may be present even when not representing energetic minimum (Cortes-Ortunõ 2017).

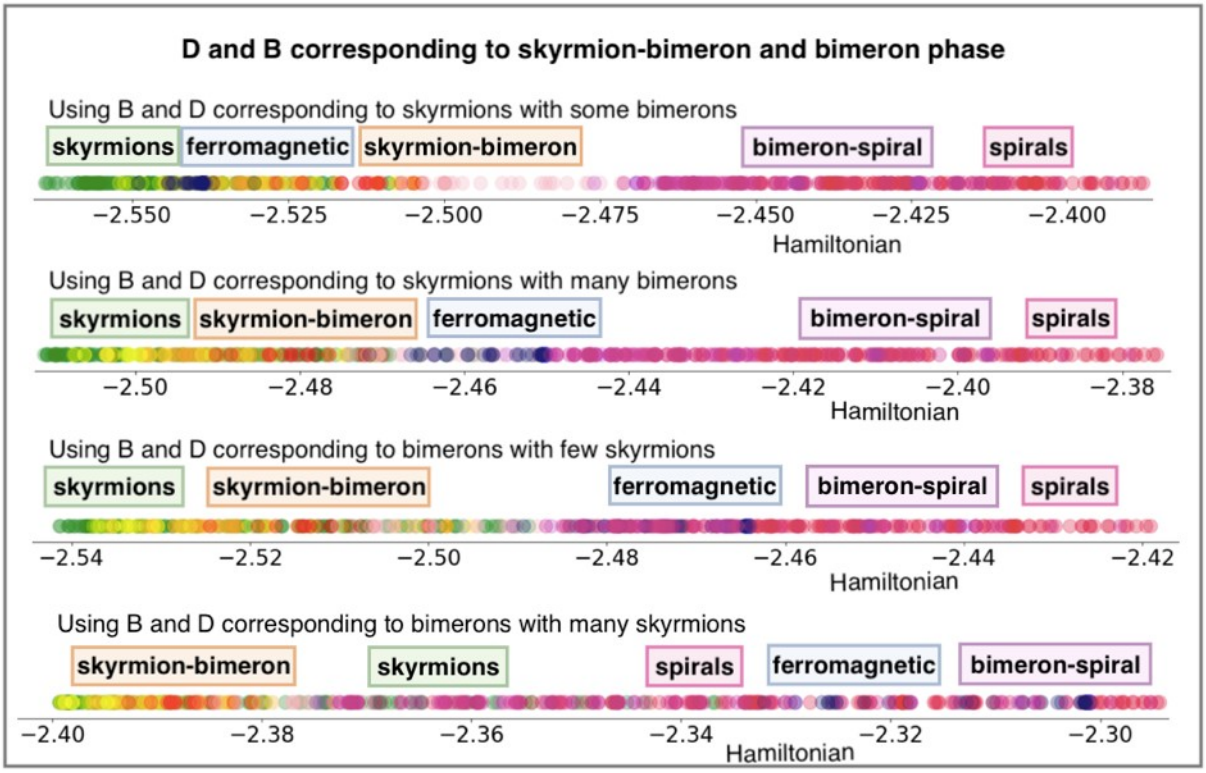

Figure 4.3 Calculating Hamiltonian for D and B of skyrmion-meron interphase. For each phase, from the pool of five plots one representative was chosen.

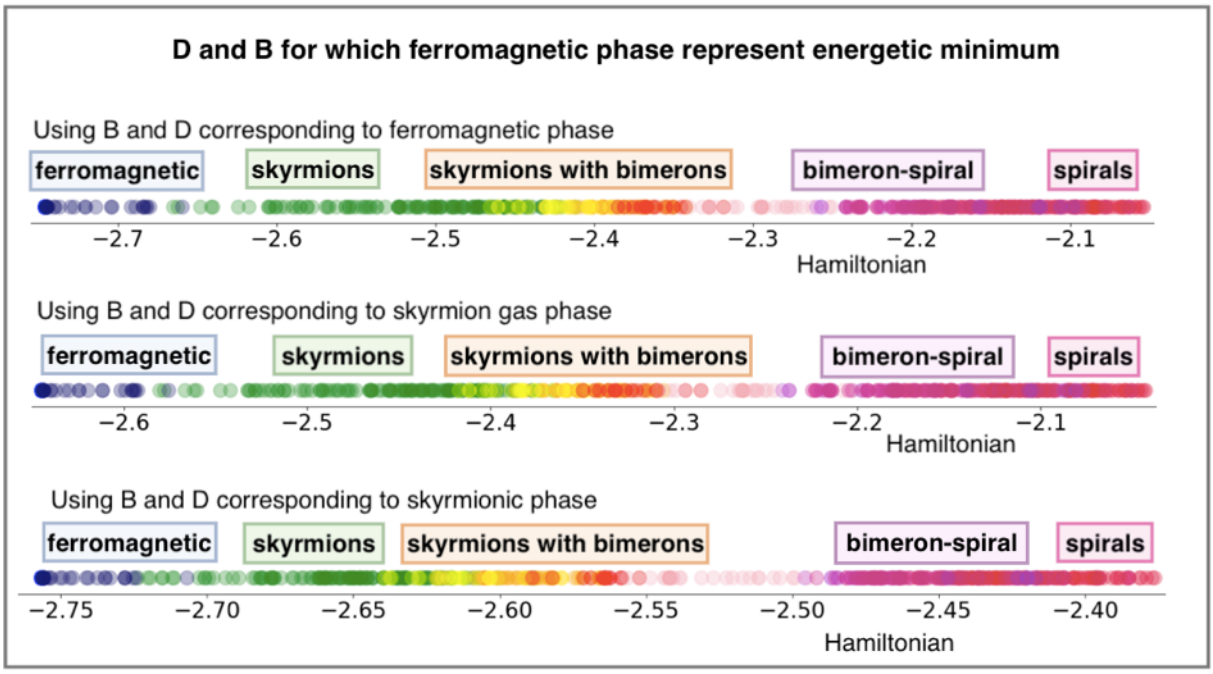

Figure 4.4 Calculating Hamiltonian for D and B of meron-spiral interphases and spirals. For each phase, from the pool of five plots one representative was chosen.

4.1.4 Local Hamiltonian

Returning to our original query, we may argue that basing our loss function on minimization of the Hamiltonian carries certain disadvantages. Specifically, there is a risk that instead of aiding the reconstruction in a productive way, Hamiltonian term might hamper through trying to alter

the configuration towards different phase, as this behaviour was observed when it acted alone. Therefore, this work used a different approach. Namely, instead of attempting to minimize the Hamiltonian on its own, we minimize the difference between energy of original configuration and energy of the reconstructed configuration. Nevertheless, by doing so we are confronted with yet another issue: Arguably, there are many configurations carrying same energy but differing in the topology. Hence the energy difference term would not provide guidance as to how exactly the reconstruction should be altered to produce energetically stable representations of the input. In other words, if faced with an energetically problematic configuration, the Hamiltonian term would alert the network that there is problem but wouldn't offer solution.

To solve the issue of non-locality, new variable termed 'the local Hamiltonian' was defined. It is basically a map of the spin energies, with its MSE representing a squared difference of spin energies prior and after reconstruction. The energy of individual spin (denoted as H_{ij}^{loc}) was calculated through the expression (1.6) without taking the sum, hence obtaining formula (4.2).

$$H_{ij}^{loc} = \left\{-J\left(\vec{S}_{ij} \cdot \vec{S}_{i(j+1)} + \vec{S}_{ij} \cdot \vec{S}_{(i+1)j}\right) + D\left[\left(\vec{S}_{ij} \times \vec{S}_{(i+1)j}\right)_y - \left(\vec{S}_{ij} \times \vec{S}_{i(j+1)}\right)_x\right] - h\left(\vec{S}_{ij}\right)_z\right\} \tag{4.2}$$

From this MSE of local Hamiltonian (denoted MSE_H) can be expressed as (4.3), where the indices 'pred' and 'true' mean we refer to predicted and true value, respectively.

$$MSE_H = \sum (H_{ij,\text{true}}^{loc} - H_{ij,\text{pred}}^{loc})^2 \tag{4.3}$$

Making use of local Hamiltonian, the loss was defined in following way:

$$L = \sum \left[(S_{\text{true}} - S_{\text{pred}})^2 + c * (H_{ij,\text{true}}^{loc} - H_{ij,\text{pred}}^{loc})^2\right] \tag{4.4}$$

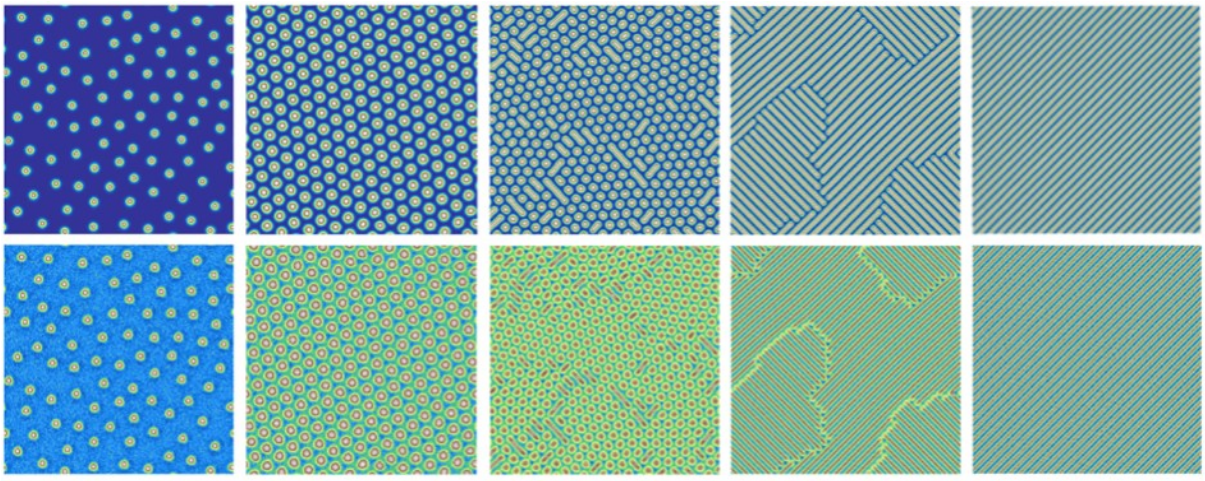

Figure 4.5 Upper row shows plots of z-magnetic component ("terrain" colour-map). Lower row shows plots of local Hamiltonian.

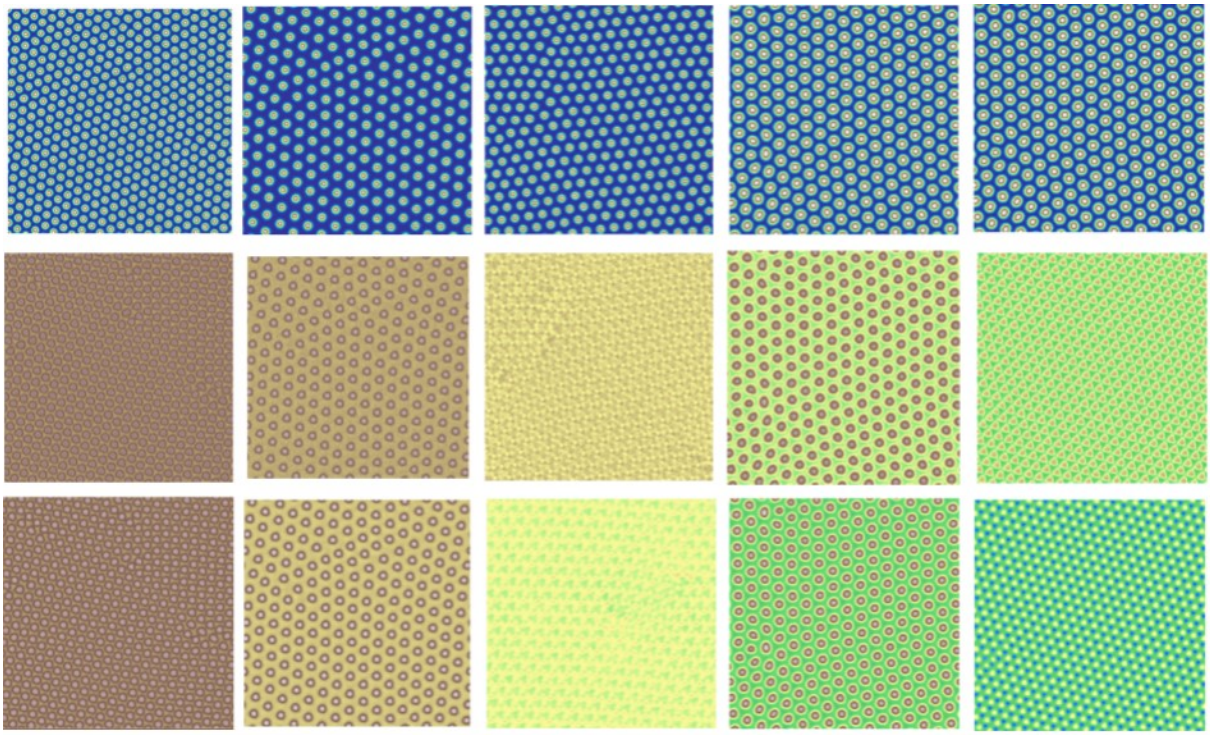

Figure 4.6 First row shows plots of z-magnetic component ("terrain" colour-map). Second row shows plots of local Hamiltonian, third row shows plots of local Hamiltonian averaged over the patch of size 3x3.

Now instead difference between total configuration energy, the difference between energies is calculated for each spin separately. Therefore, if there was a spin (or a group of spins) making troubles the spin MSE would immediately pinpoint the source of the error and provide guidance on how to remove it, so that reconstruction would be altered in a desired way. Of course, this would only work if the plot of the spin energies offered some characteristic pattern, non-trivial on the small scale, ideally manifesting for each phase in a different way. Well, it turns out that plots of local Hamiltonian look very similar to plots of z-component magnetisation. To be more specific, there is almost no different between plots of spirals without defects and homogeneous magnetisation phases. For spiral-meron phases, the interphase between meron caps and spirals get highlighted. This might be exactly what we want, as this is the location where eight-fold-reducing autoencoder often fail to produce good reconstruction. Making this an energetically privileged place (a place where conservation of energy is particularly important) might force the autoencoder to handle it better.

Unfortunately, energy plots of skyrmion phases look a lot less promising. Let us observe Figure 4.5. (Upper row shows plots of z-magnetisation component, lower row shows plots of local Hamiltonian). We can see that for skyrmion gases (and for sparse skyrmion lattices) the space around skyrmions appears grainy, showing that small irregular fluctuation of magnetisation had non- negligible energy contribution – which is arguably not what we want. On the other hand, for dense skyrmion lattices, the individual skyrmions appear less sharp. The lower contrast

means that in terms of energy, dense skyrmions are less important - lesser loss would arise from their deformation. Again, this is not what we would like to observe. And it will only get worse: In previous, the bounds of the colour-map were recalculated separately for each image (so that upper bound corresponded to maximum of the image and lower bound to minimum of the image.) If we set upper bound to maximum of the dataset and lower bound to minimum, we would obtain plots alike second row in Figure 4.6 (where the first row again shows plots of z-magnetisation component.) In this case, the contrast is even lower.

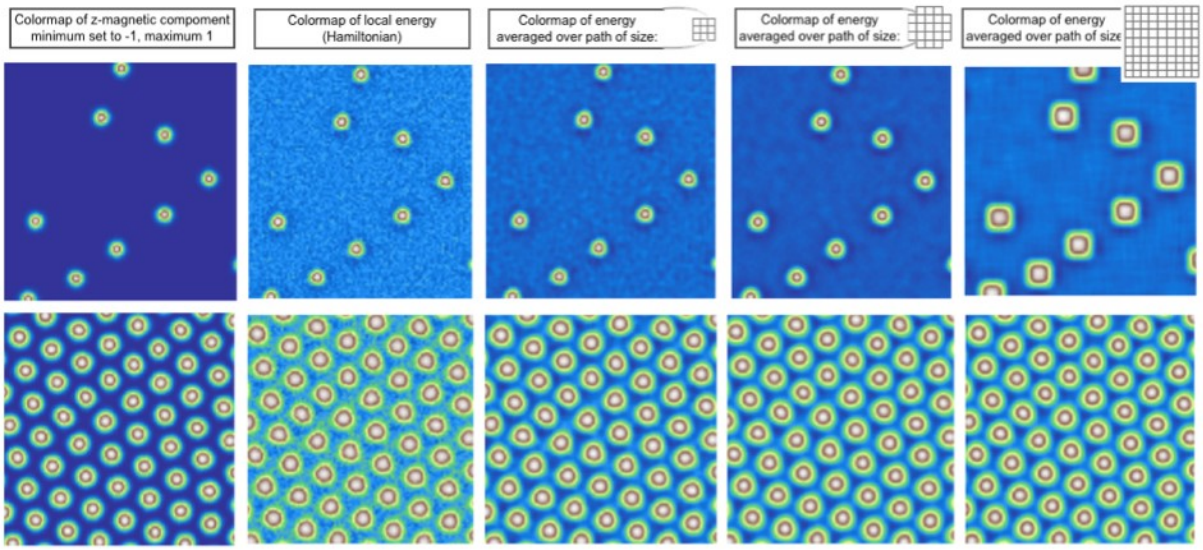

Figure 4.7 First column shows plots of z-magnetic component ("terrain" colour-map). Second columns show plots of local Hamiltonian. The remaining three columns show plots of local Hamiltonian as averaged over different patch sizes.

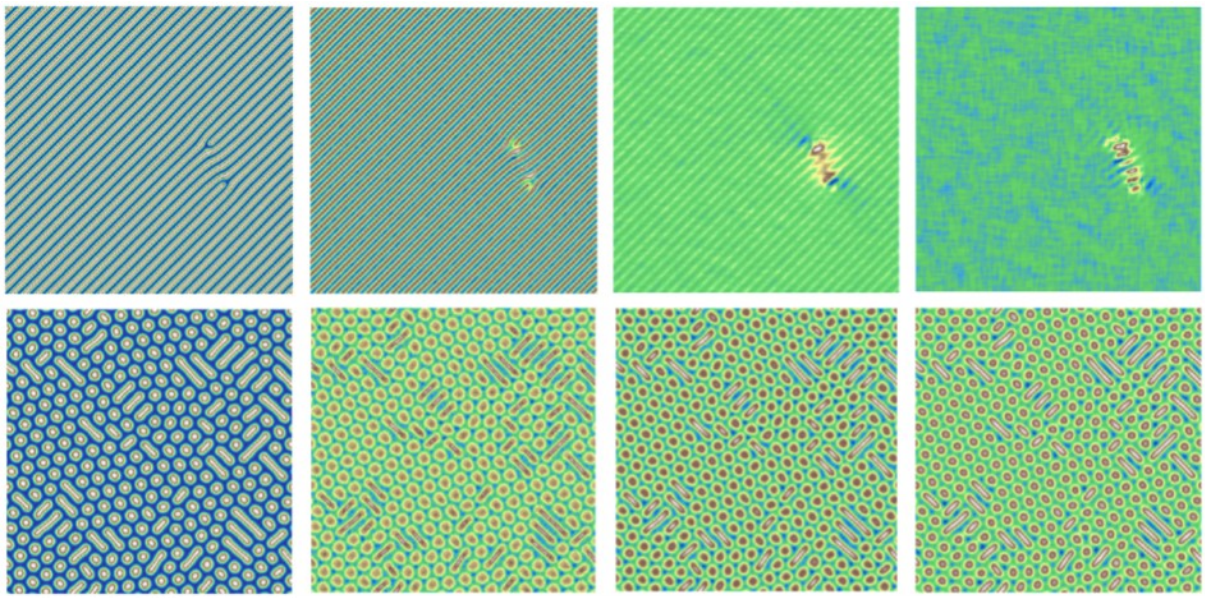

Figure 4.8 First column shows plots of z-magnetic component ("terrain" colour-map). Second columns show plots of local Hamiltonian. The remaining three columns show plots of local Hamiltonian as averaged over different patch sizes.

4.1.5 Averaged Hamiltonian

So, energy plots have two undesirable properties: configuration-dependent low contrast and background noise. Is there a way how to eliminate this? Well, here is the thing: What if we instead of calculating energy for each spin separately, calculated energy for small assembly (like 3x3 matrix) of spins? We would obtain results alike third row in Figure 4.6. We see that for most of the configurations, the contrast is deepened. If we experiment with size of the averaging matrix, we would see that this change is gradual, as Figure 4.7 shows. Unfortunately increasing size of averaging matrix starts blurring helical configuration as Figure 4.8 shows. After all, even Figure 4.7 shows that to large sizes of averaging matrices can produce non-physical results (last column). Moreover, as we can see on Figure 4.6, there are low-contrast configuration for which averaging energy exacerbates the problem, rather than helps it (middle column).

In the end, there are four main reasons why working with local Hamiltonian is preferred to averaged energy: 1) local Hamiltonian has more direct physical interpretation 2) averaging energy blurs topology of spiral phases 3) contrast-sharpening doesn't work on all configurations 4) training CNN with averaged energy as loss function is numerically more demanding (hence the whole process takes significantly longer time even for averaging matrices of smallest sizes).

4.2 Implementing local Hamiltonian

Now, let us contemplate the results. We are going to explore how the performance of CNN changed with increasing the contribution of local Hamiltonian to the total loss. Specifically, the constant c varied in the interval from 0 to 1,4. As the Figures 4.19 and 4.20 show, for the values of c approaching 1,4, the reconstruction error rose drastically. This was the main reason why higher values of c weren't investigated.

4.2.1 Main trends

It was found that with increasing Hamiltonian contribution increased both in-plane magnetisation errors and out-of-plane magnetisation error. This can be seen on Figures 4.19 and 4.20. Of course, this does not tell us whether the CNN failed only at some configuration, or its performance worsened uniformly. If we plot the errors for each phase separately (as it is done in Figures 4.9 – 4.18), we will find out the following: For bimeron-spiral configurations (spirals with bimerons and spirals with merons), MSE of out-of-plane magnetisation rises in the whole interval. For skyrmions, it stays approximately constant for c within the interval 0-0,8, but sharply rises for c>0,8. For skyrmion-bimeron and bimeron-skyrmion configurations,

it rises negligibly for c within the interval 0-0,8, but accelerates for c>0,8. For pure spiral configurations it drops sharply for c within the interval 0-0,3, but decelerates for c>0,3. The trends for MSE of in-plane magnetisation are very similar, except for the fact that it does not seem to rise for spirals with merons configurations.

From this it can be concluded that the optimal value of c should be sought in the interval 0-0,8. In this interval the magnetic MSE not to worsen that drastically for most of the phases. Contemplating the Figures 4.19 and 4.20) again, we see that this also applies to the total errors: Both total MSEs (in-plane as well as out-of-plane) rise most significantly for the c in the interval 0.8-1.4.

For the MSE of local Hamiltonian, the trends were completely different: For increasing value of c, MSE of local Hamiltonian decreased. It is shown on the Figures (42) and (43). This trend was seen for all phases although individual phases differed in terms of how sharply the MSE of Hamiltonian seemed dropping.

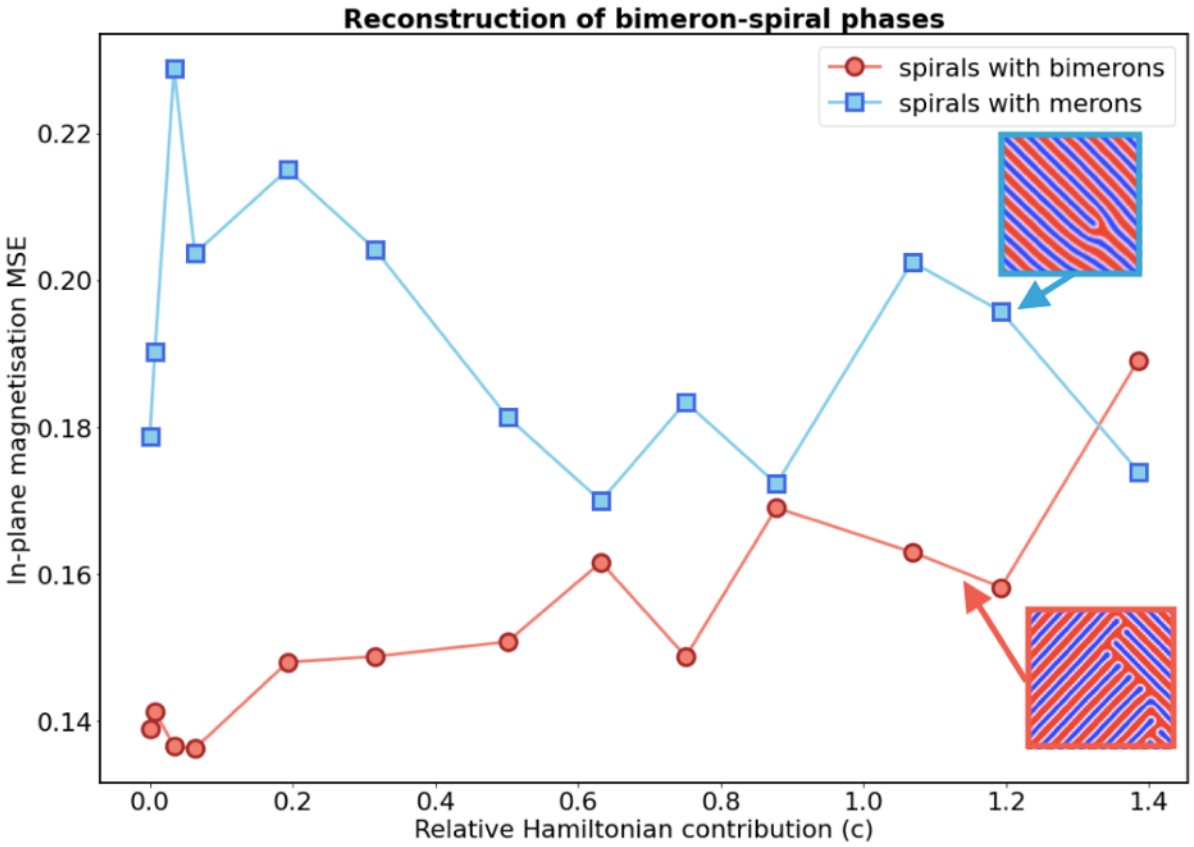

Figure 4.9 In-plane MSE of bimeron-spiral phases.

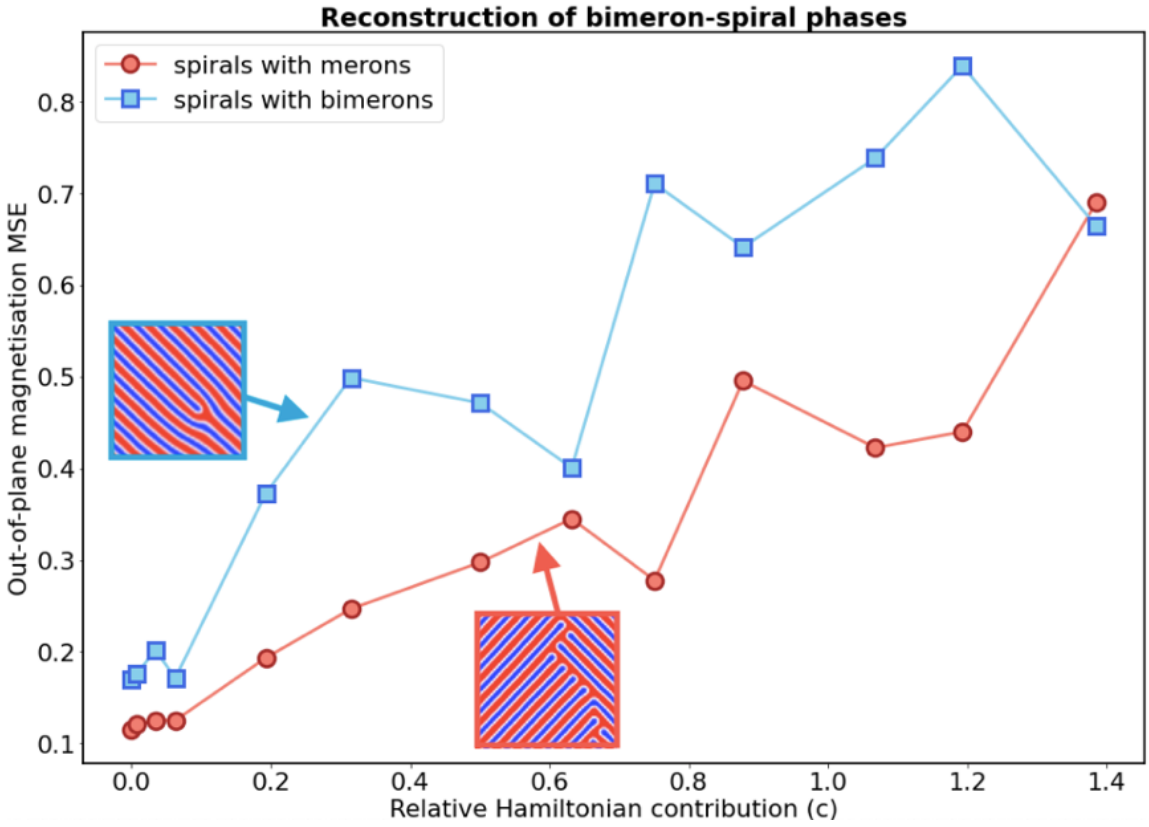

Figure 4.10 Out-of-plane MSE of bimeron-spiral phases.

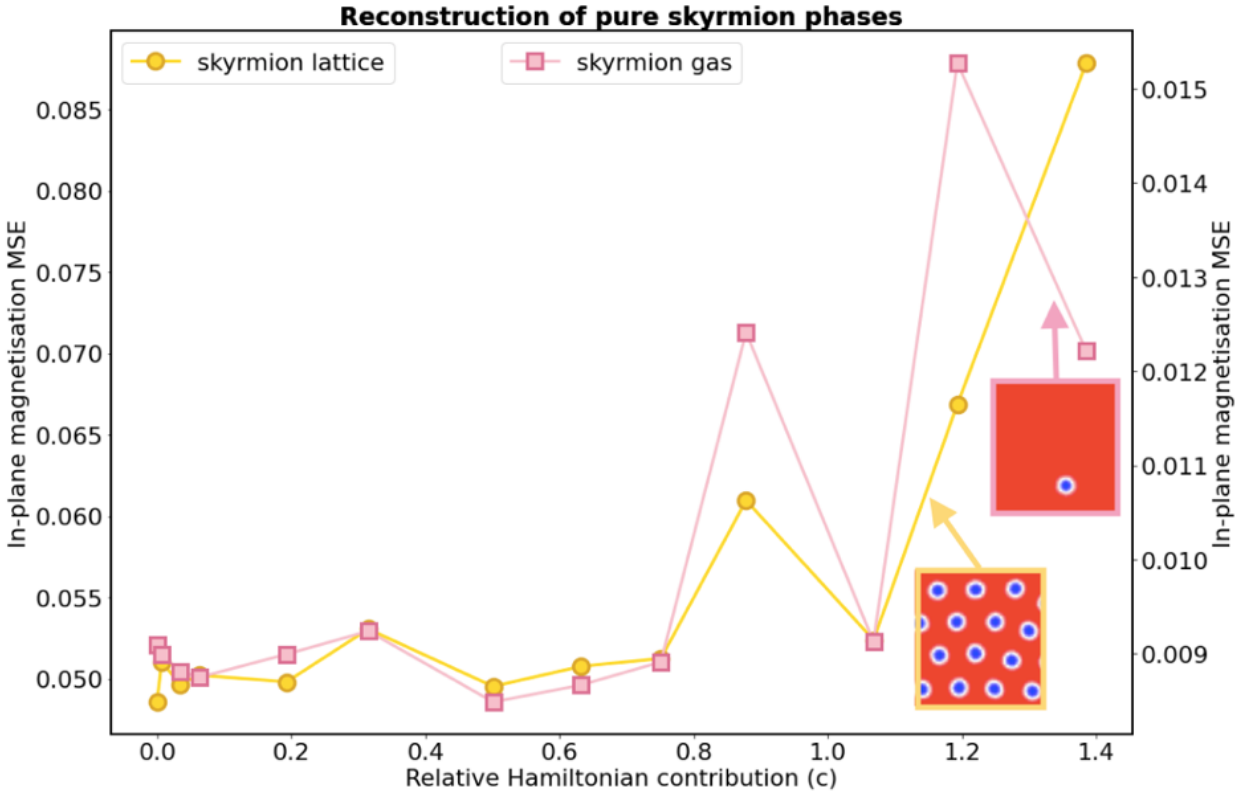

Figure 4.11 In-plane MSE of phases containing only skyrmions.

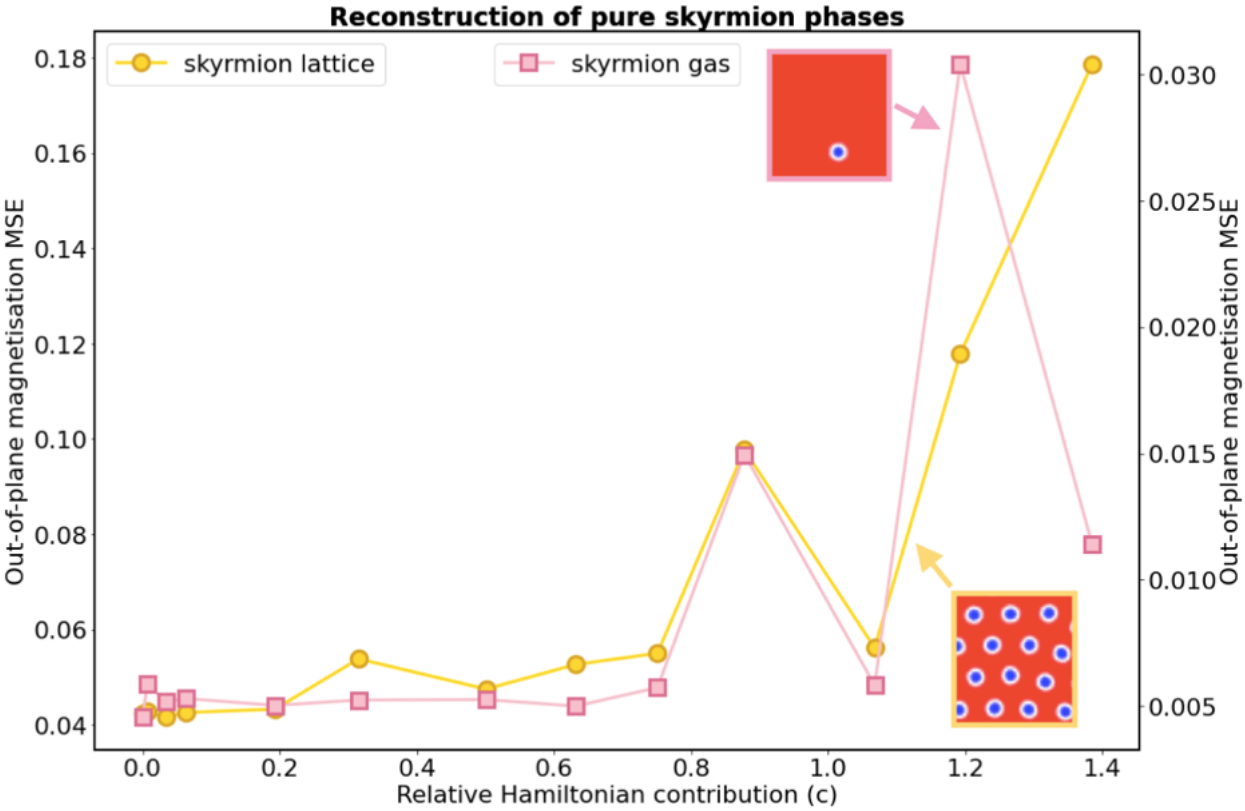

Figure 4.12 Out-of-plane MSE of phases containing only skyrmions.

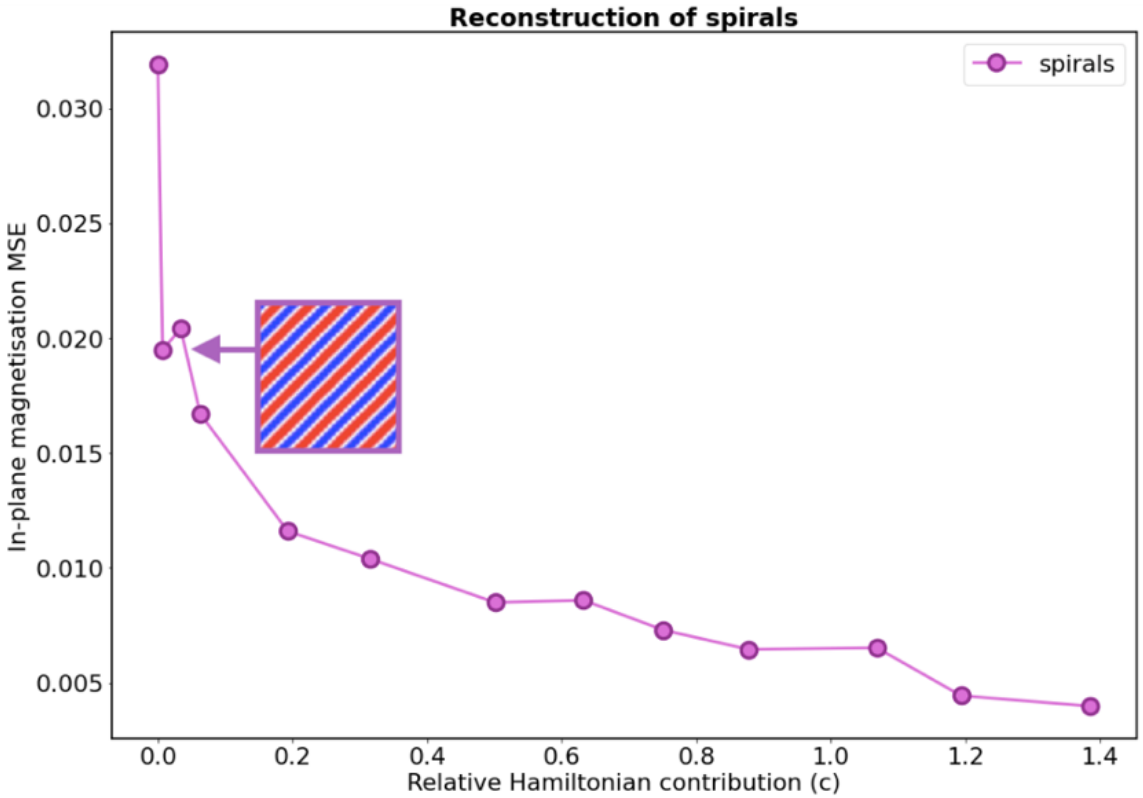

Figure 4.13 In-plane MSE of spirals.

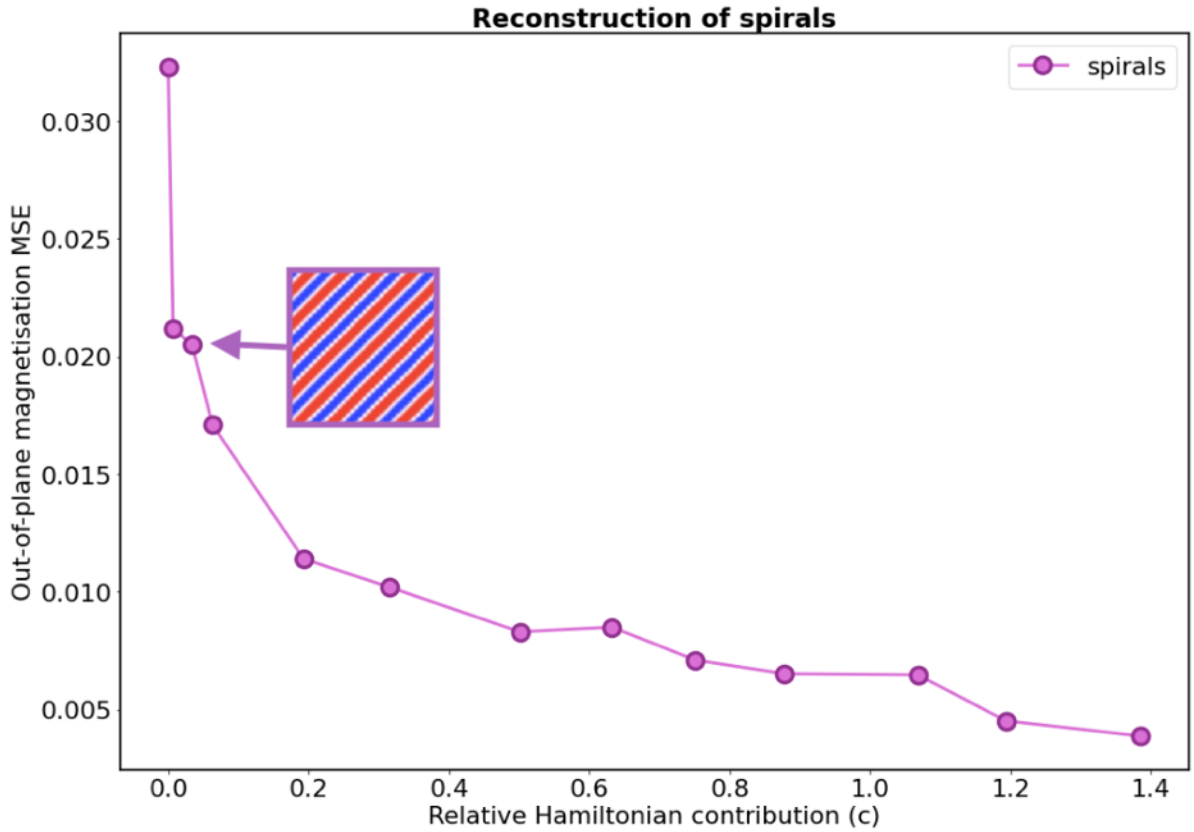

Figure 4.14 Out-of-plane MSE of spirals.

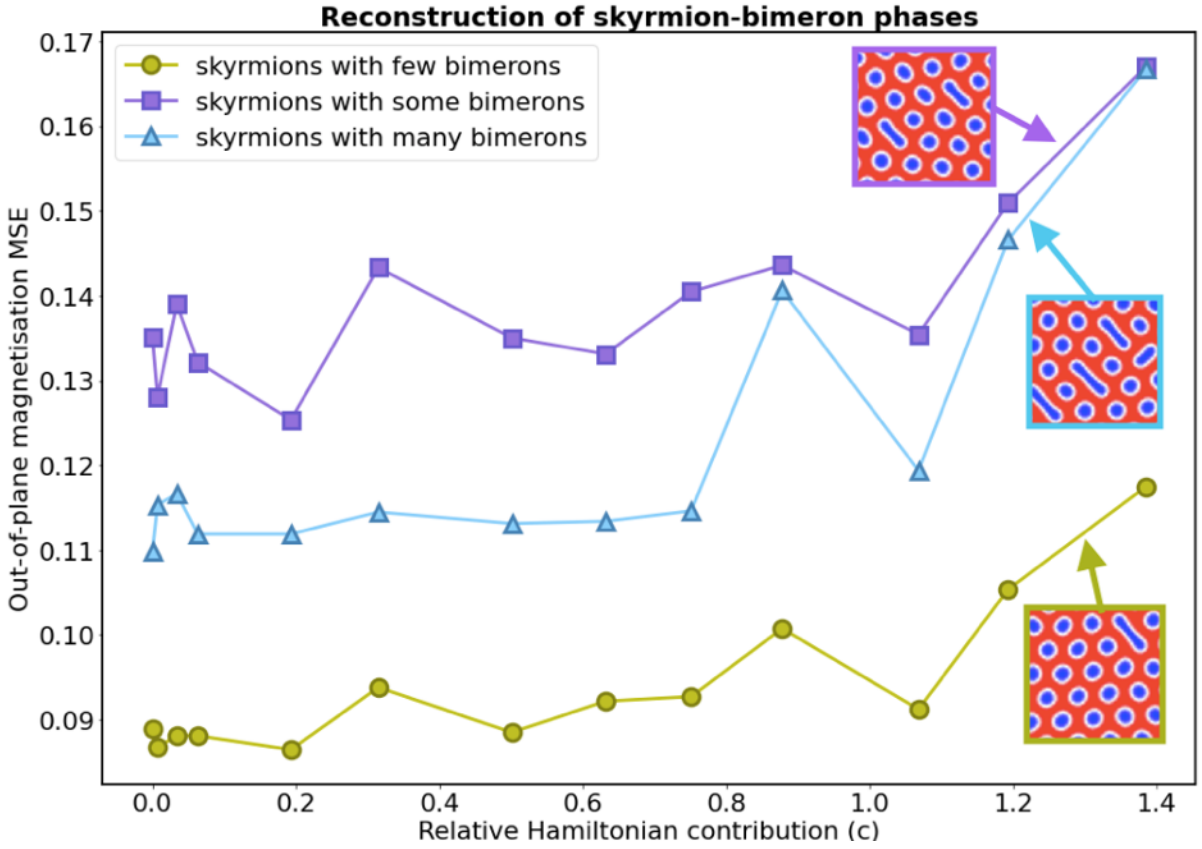

Figure 4.15 In-plane MSE of skyrmion-bimeron phases.

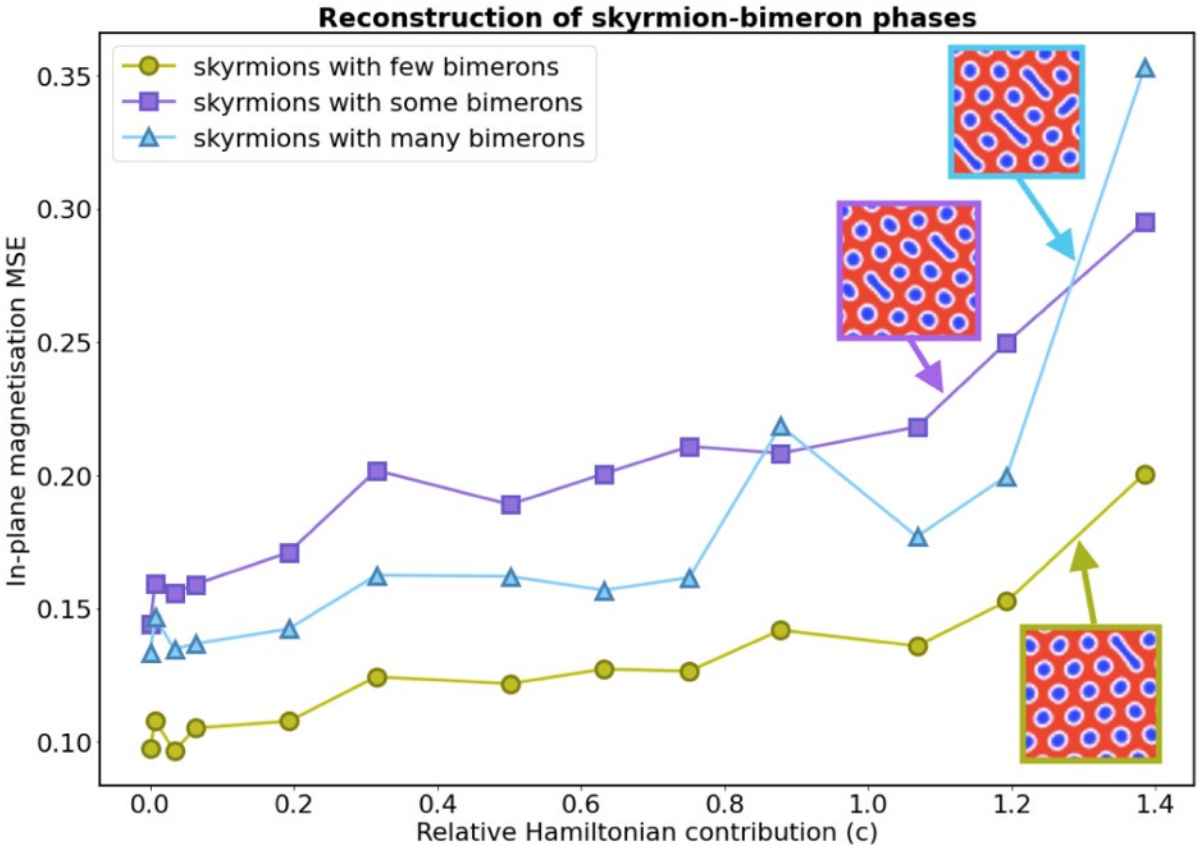

Figure 4.16 Out-of-plane MSE of skyrmion-bimeron phases.

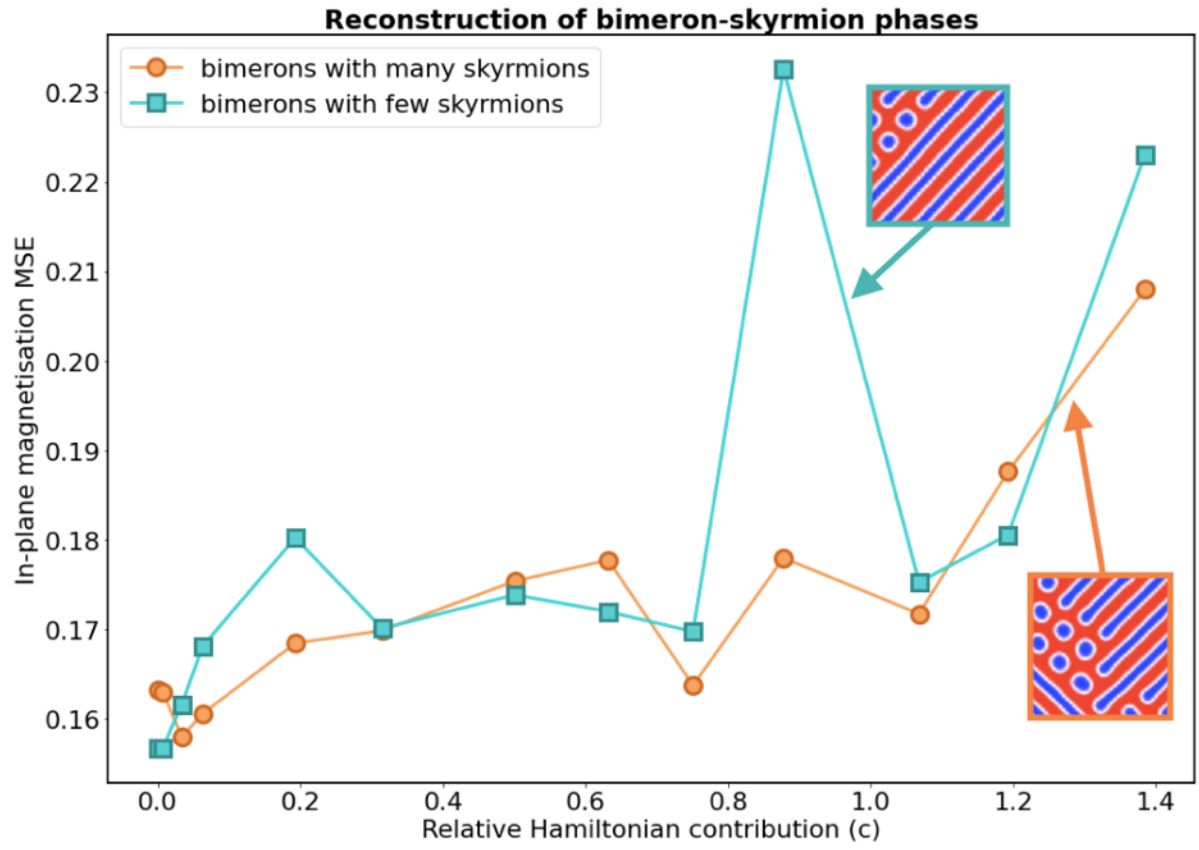

Figure 4.17 In-plane MSE of bimeron-skyrmion phases.

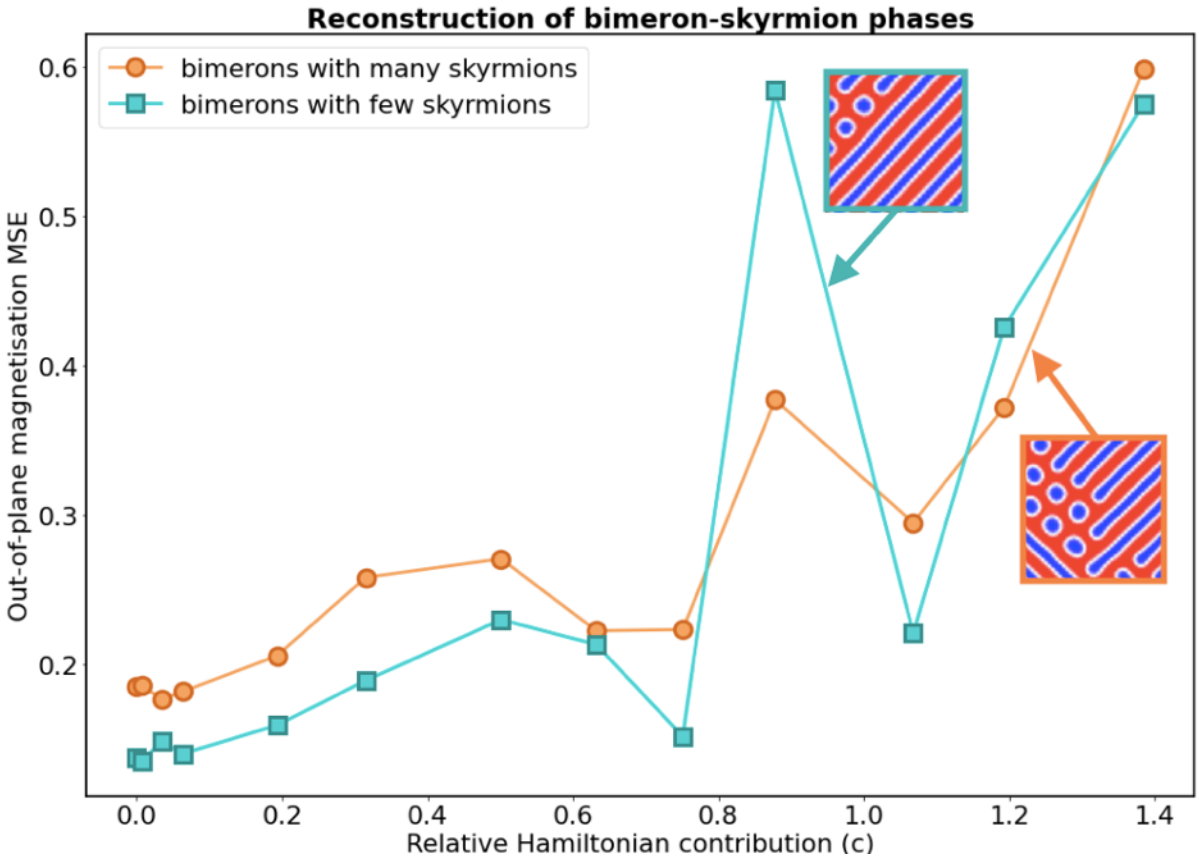

Figure 4.18 Out-of-plane MSE of bimeron-skyrmion phases.

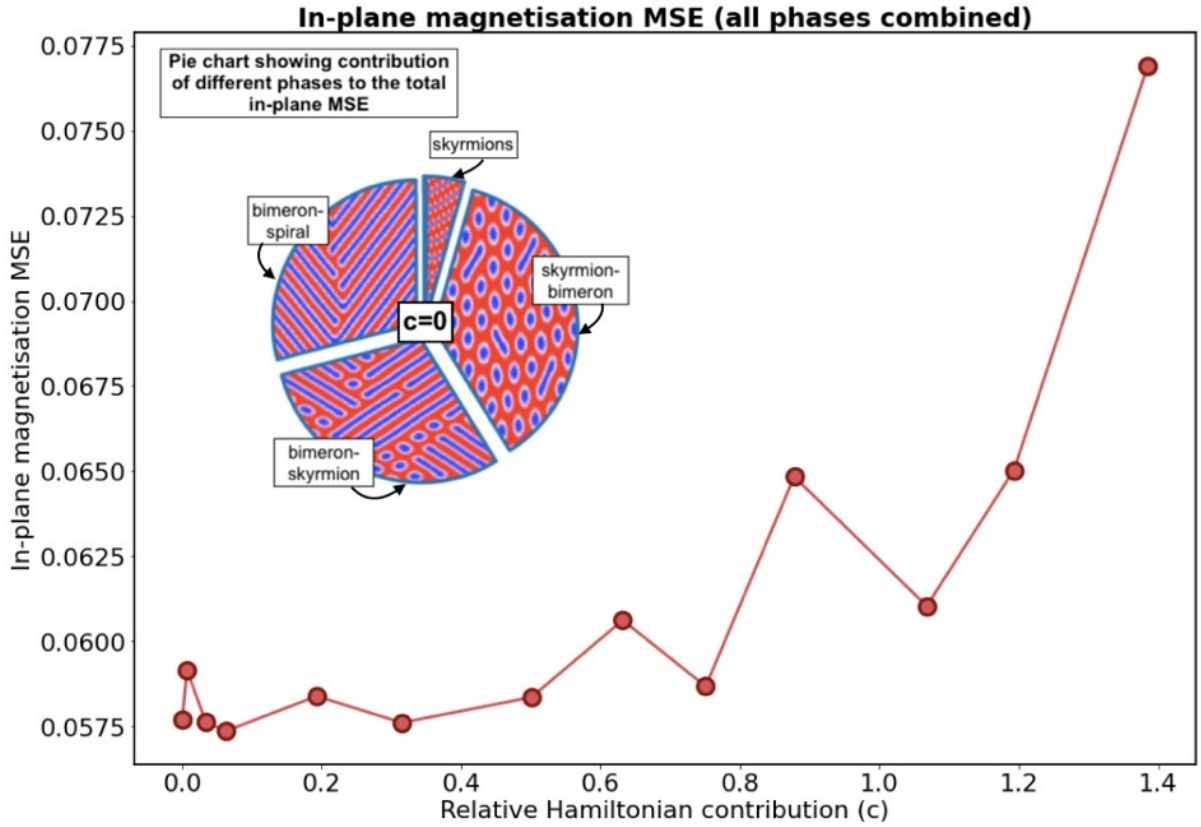

Figure 4.19 In-plane MSE for all phases combined.

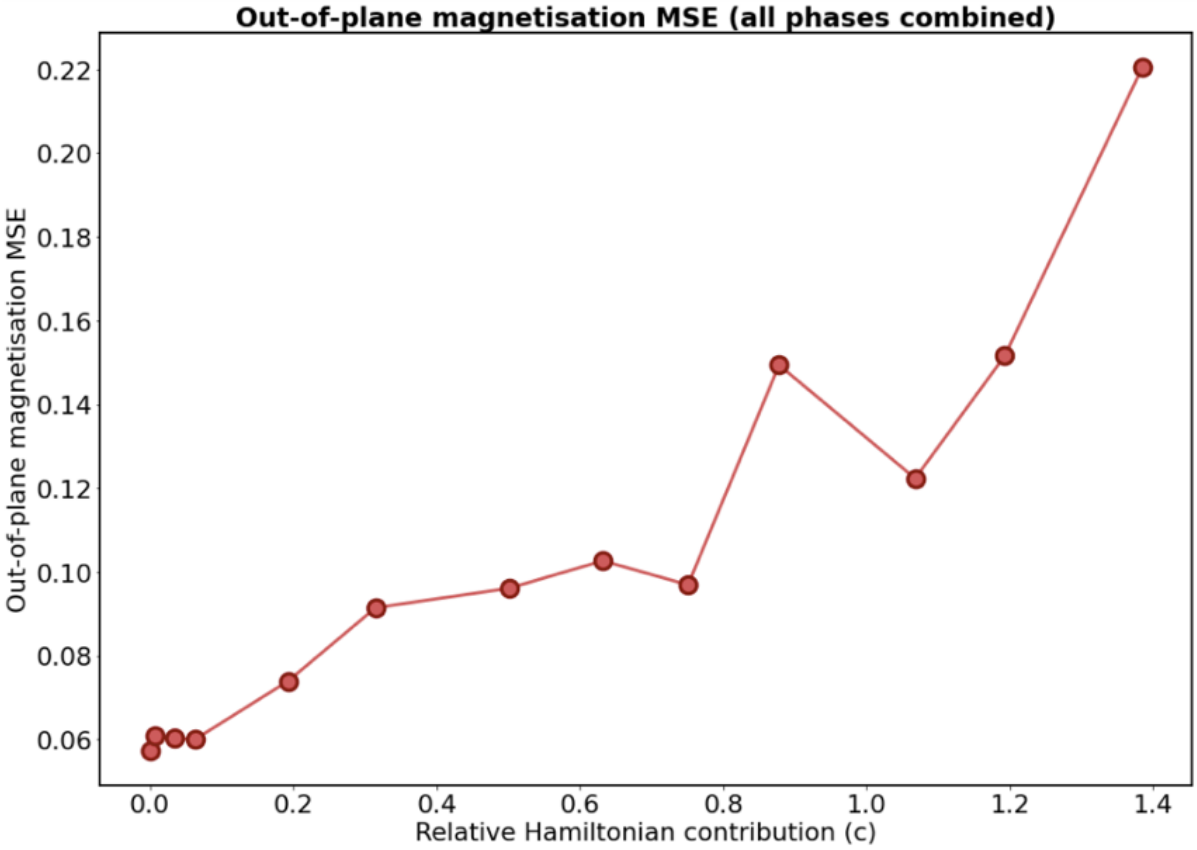

Figure 4.20 Out-of-plane MSE for all phases combined.

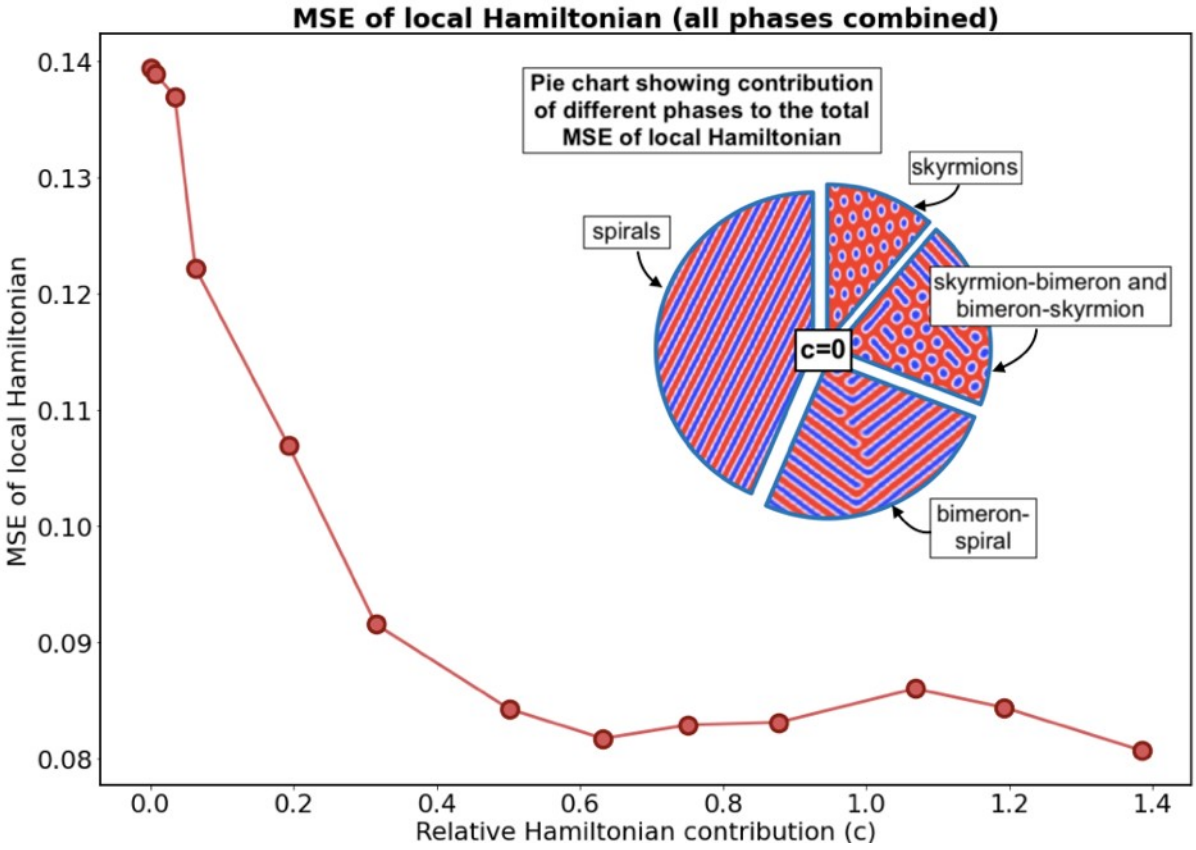

Figure 4.21 MSE of local Hamiltonian for all phases combined.

4.2.2 Contribution of each phase to the total error

For neither in-plane magnetic MSE nor MSE of local Hamiltonian, inclusion of Hamiltonian term didn't change the ratio in which the phases contributed to the total error. Figures 4.19 and 4.21 show the pie charts showing the relative contribution of each phase to the total error. These are plotted for c=0, but for higher values of c, they look very similar. As we can see, the biggest contributor to the total error to the in-plane MSE are the bimeron-containing phases, all main three categories having similar contribution. For the MSE of local Hamiltonian, the biggest contributor were spirals. For the out-of-plane MSE, inclusion of Hamiltonian did change the ratio in which the phases contributed to total error. Specifically, with rising value of c, the contribution of spirals with merons rose until it dominated the total out-of-plane MSE. The gradual change can be seen on the Figure 4.22, which shows the pie charts for differing values of c.

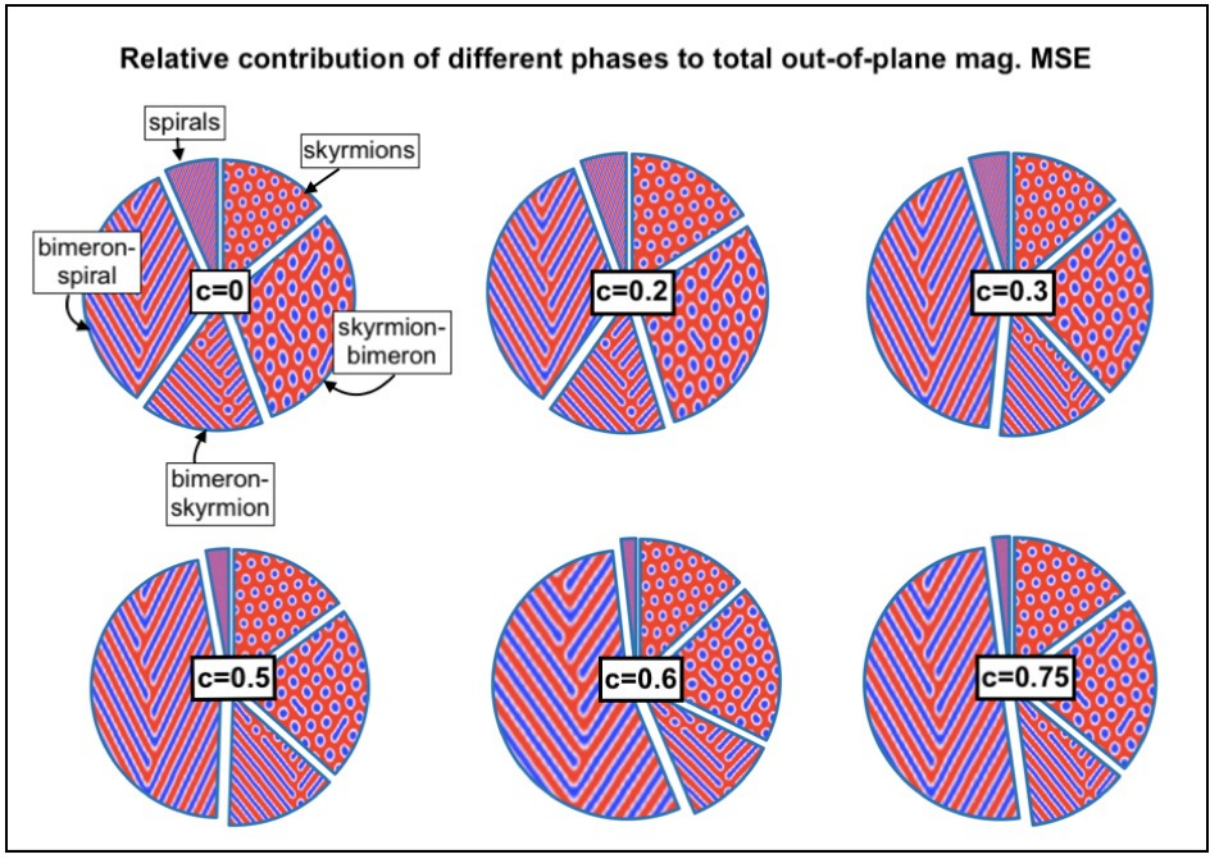

Figure 4.22 Pie charts showing how different phases contribute to total out-of-plane MSE for differing c.

4.2.3 Comparing MSEs of different phases

The Figures A-Z the performance of four-fold reducing autoencoder, eight-fold reducing autoencoder and family of Hamiltonian-containing eight-fold reducing autoencoders with c in

the range 0-0,8 (reflecting the optimum). The bar charts show mean MSEs for different phases. For spirals, both in-plane and out-of-plane magnetic MSEs decrease until reaching values obtained for four-fold reducing autoencoder. For skyrmions, the MSEs increase, but not as significantly as it was observed for the for bimeron-containing interphases: The in-plane magnetic MSEs rise for all interphases, but most significantly for the skyrmion-bimeron phase. On the other hand, the out-of-plane magnetic MSEs rise dramatically for the bimeron-spiral phase, more slowly for bimeron-skyrmion phase and remain unchanged for the skyrmion-bimeron phase.

As for the MSE of local Hamiltonian, the values achieved for higher c are close to what was observed for four-fold reducing autoencoder. Only exceptions of this trend are spiral and spiral-bimeron phases for which MSE of local Hamiltonian remains relatively large.

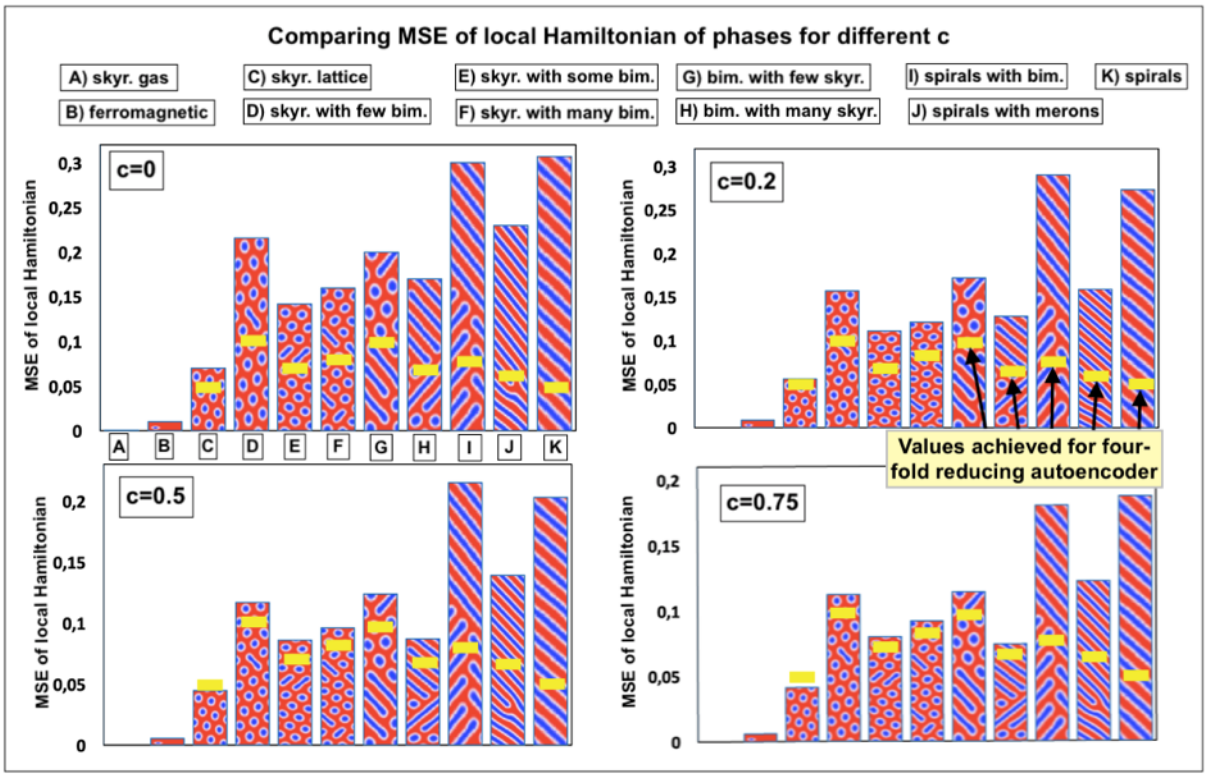

Figure 4.23 Comparing local Hamiltonian MSEs of different phases for differing c.

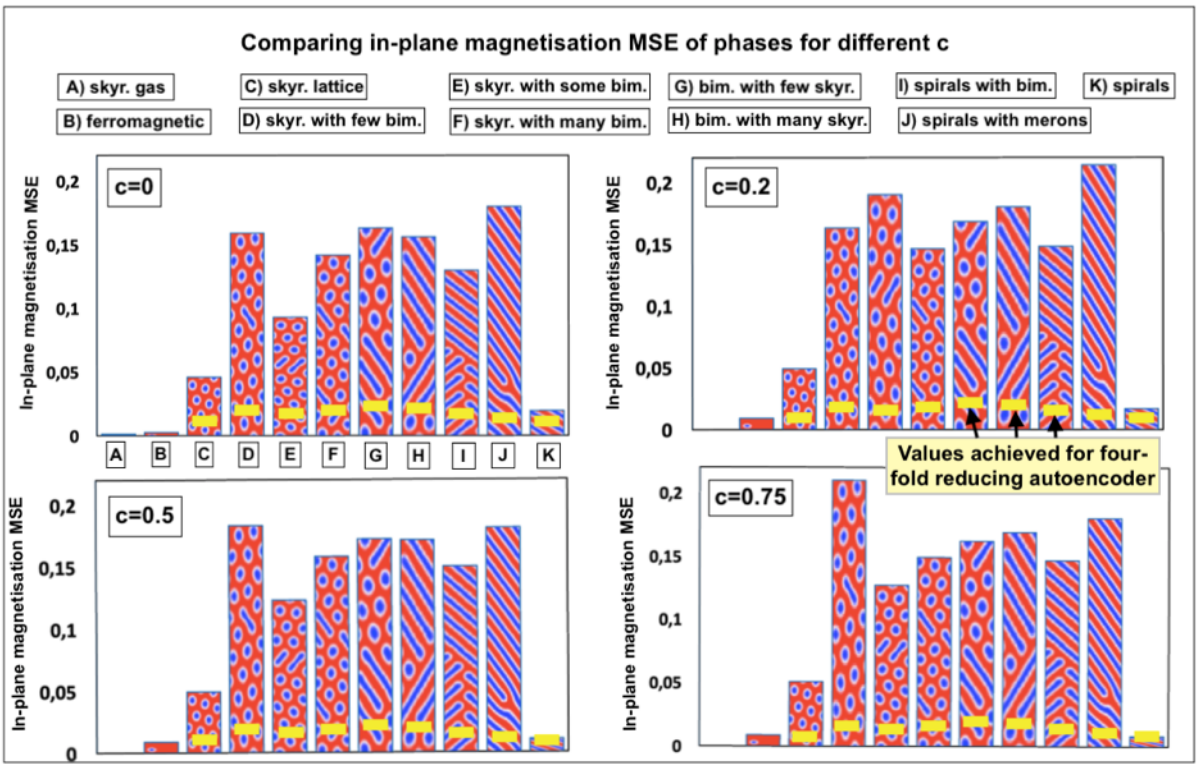

Figure 4.24 Comparing in-plane MSEs of different phases for differing c.

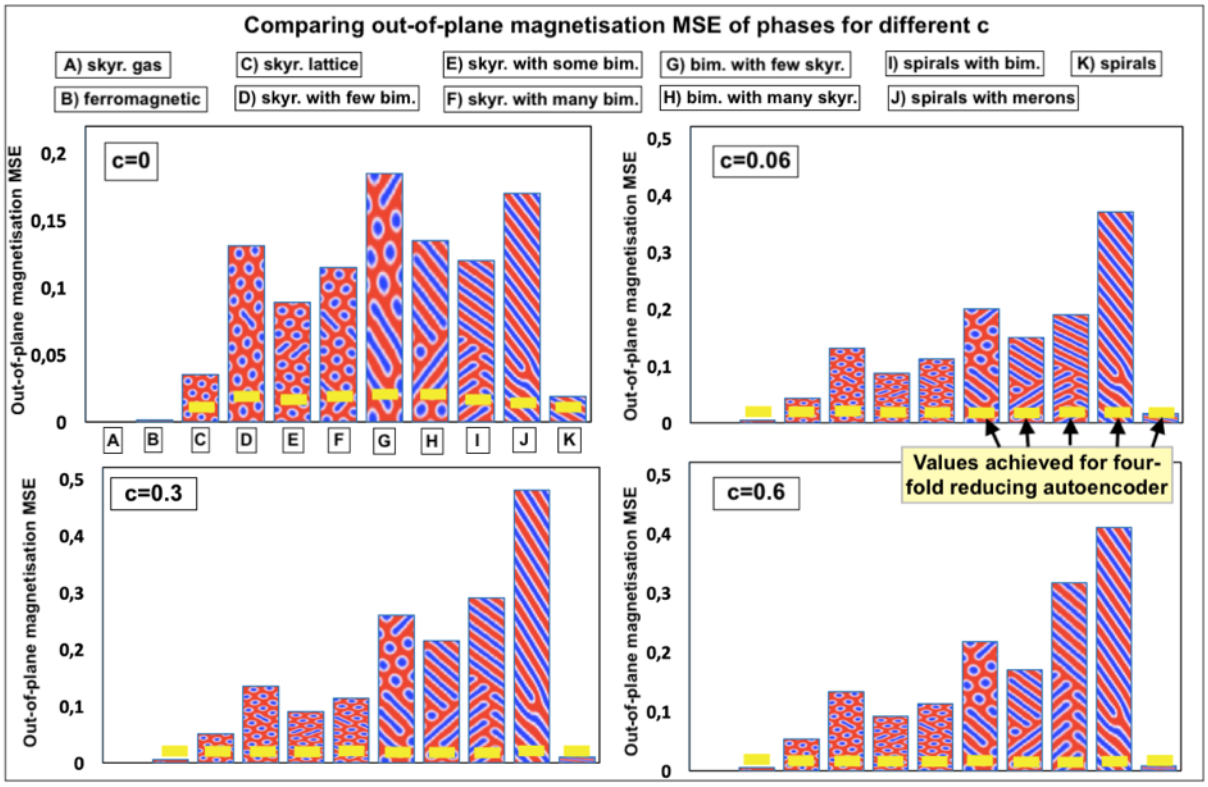

Figure 4.25 Comparing out-of-plane MSEs of different phases for differing c.

4.2.4 Abrupt worsening of reconstruction for c=0.8

In this section we will discuss the problem of abrupt worsening of reconstruction for Hamiltonian contribution constant equal to c=0.8. To do this, let us for a moment concentrate on pure skyrmion lattice configurations. If we plot the value of skyrmion magnetic MSE against skyrmion total magnetisation, we will find out that the value of magnetic MSE is directly proportional to the magnetisation. Which is not surprising: The value of magnetisation tells us how much space is devoted to the skyrmions. In other words: what proportion of the configuration contains nontrivial (non-ferromagnetic) topological elements and hence is expected to carry reconstructive errors.

Very important is following realisation: The configurations with skyrmionic lattices are homogeneous in the sense that within the configuration all small assemblies of skyrmions have similar appearance. (Even though there might be a topological defect breaking the large-scale symmetry, on the small scale all parts of the configuration look very alike.) Therefore, if the CNN was to completely fail the reconstruction, the error would be everywhere very similar. In the extreme case if we imagined the CNN turning skyrmionic lattices to ferromagnetic configuration, there would be a perfect linear dependence of the error on the magnetisation. And such notion is not unrealistic: On the plot of z-magnetisation of the reconstructed configurations it was observed that the most problematic configurations got reconstructed as pure red images with few regular white blobs.

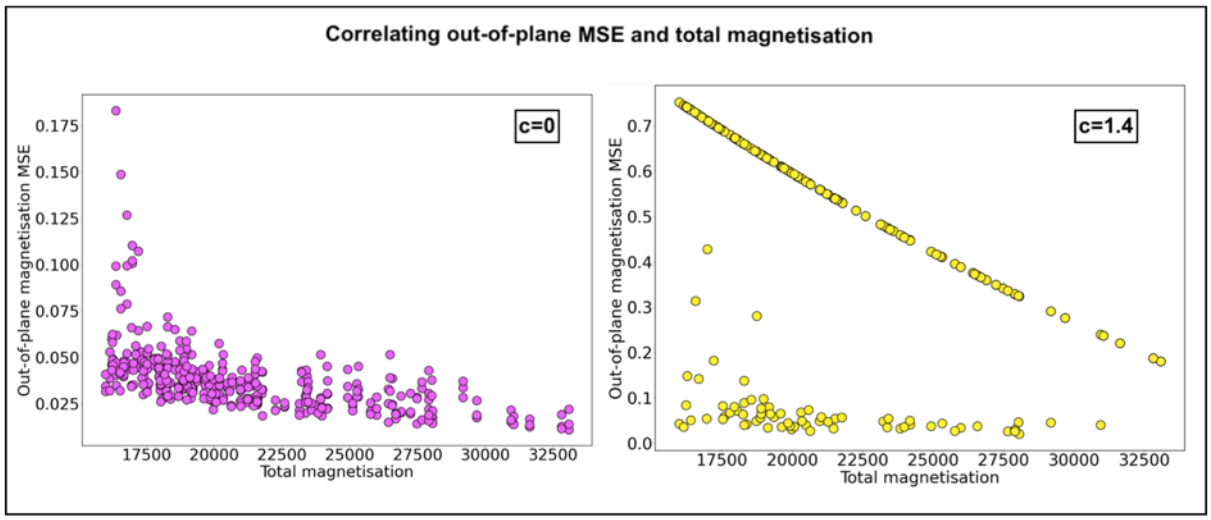

Figure 4.26 Correlating out-of plane MSE with total magnetisation.

More quantitative analysis of the situation might be performed by contemplating the Figure 4.26, which shows that there truly is a linear dependence between magnetisation and magnetisation error with the relationships being almost perfect for truly badly reconstructed

configuration. In fact, the whole plot appears split to parts: the upper part showing the perfect linear dependence (present only for cases when c>0.8) and the lower part showing linear trend with large spread. Here we see that for c>0.8 some of the skyrmionic configuration become outliers as a result of being reconstructed in a non-physical way. Very probably, such outliers occurred in all skyrmion-bimeron interphases, since these interphases also contained skyrmions and their reconstruction also worsened for c>0.8. Presence of outliers was another reason (beside high total MSE) why the values of c greater than 0.8 are disfavoured.

4.2.5 Correlating reconstruction error with skyrmionic properties

In an attempt to explain the large spread of skyrmionic MSE for low values of magnetisation, the dependence of MSE on different skyrmionic properties was explored. These properties were: inter-center distance (as calculated by expression 3.1), inter-edge distance (as calculated by expression 3.2), size (as defined in section “skyrmionic properties”) and inner core size (as defined in section “skyrmionic properties”).

For size and inner core size, no correlation was observed. For the inter-center distance, a linear relationship was found between skyrmion inter-center distance and inverted magnetic MSEs (see Figures 4.27 and 4.28). It held for all values of estimated inter-center distance, but it was particularly strong for the interval 9-11. This is also the sole interval in which one can see a difference in autoencoder performance for different Hamiltonian contribution constant. In particular, with rising c value the linear trend gets shifted downwards (the inverted MSE starts at lower values). This is clearly visible on Figure 4.30 where linear best fit lines are compared for different c. Interestingly, this only holds for z-magnetisation MSE: Though for in-plane magnetisation error, there is also the linear relationships between inverted MSE and inter-center distance, no difference is observed when changing Hamiltonian contribution constant (see Figure 4.29). If instead of magnetic MSE, MSE of Hamiltonian is plotted, the trend-line-shift is in the opposite direction - for higher values of the constant c, trend lines are positioned more towards the top (see Figure 4.31). This is true for whole investigated range, not only for the interval 9-11. Nevertheless, for high values of c situation start to worsen.

The trends observed for inter-edge distance were similar to trends observed inter-center distance.

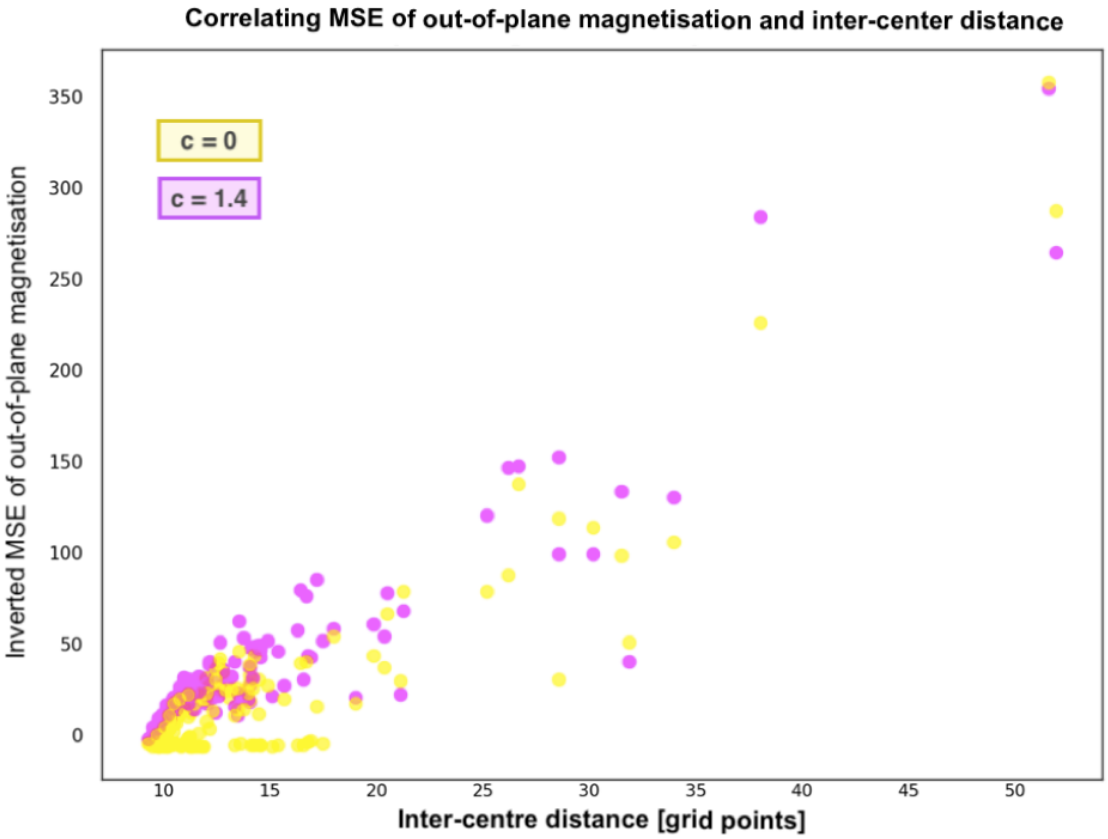

Figure 4.27 Correlating out-of plane MSE with inter-center distance.

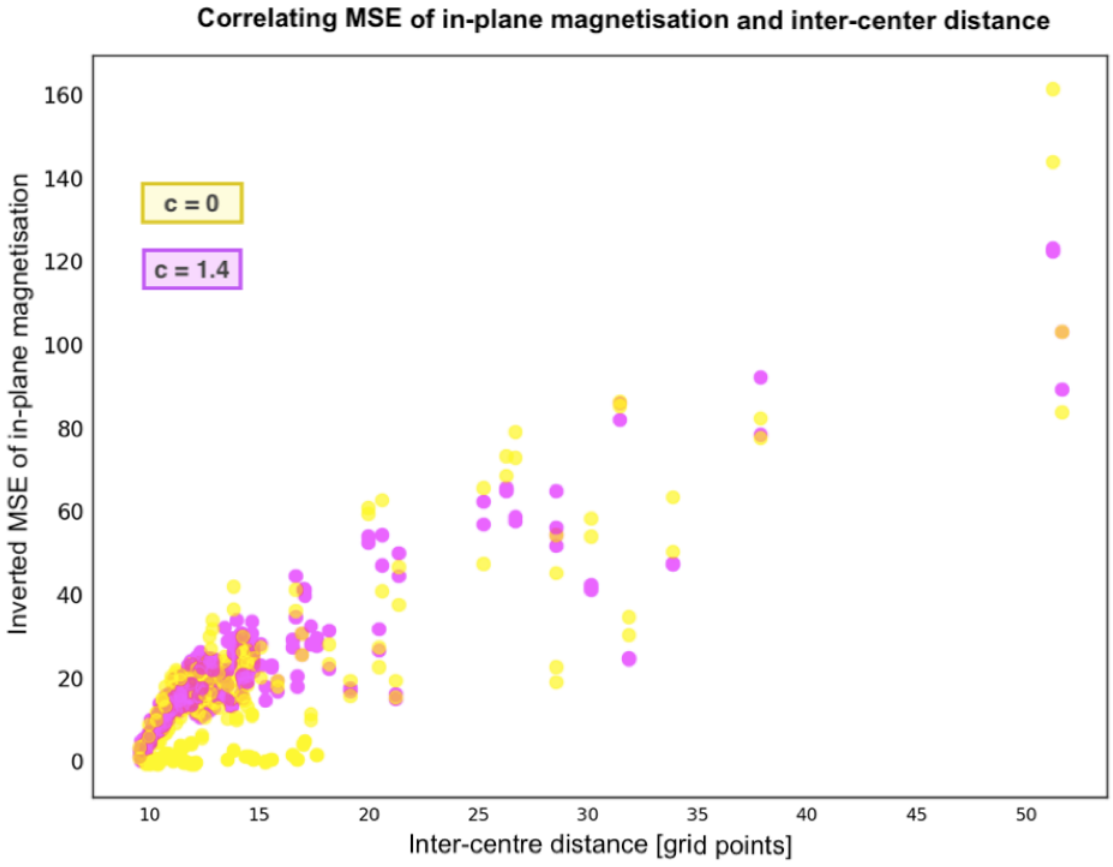

Figure 4.28 Correlating in-plane MSE with inter-center distance.

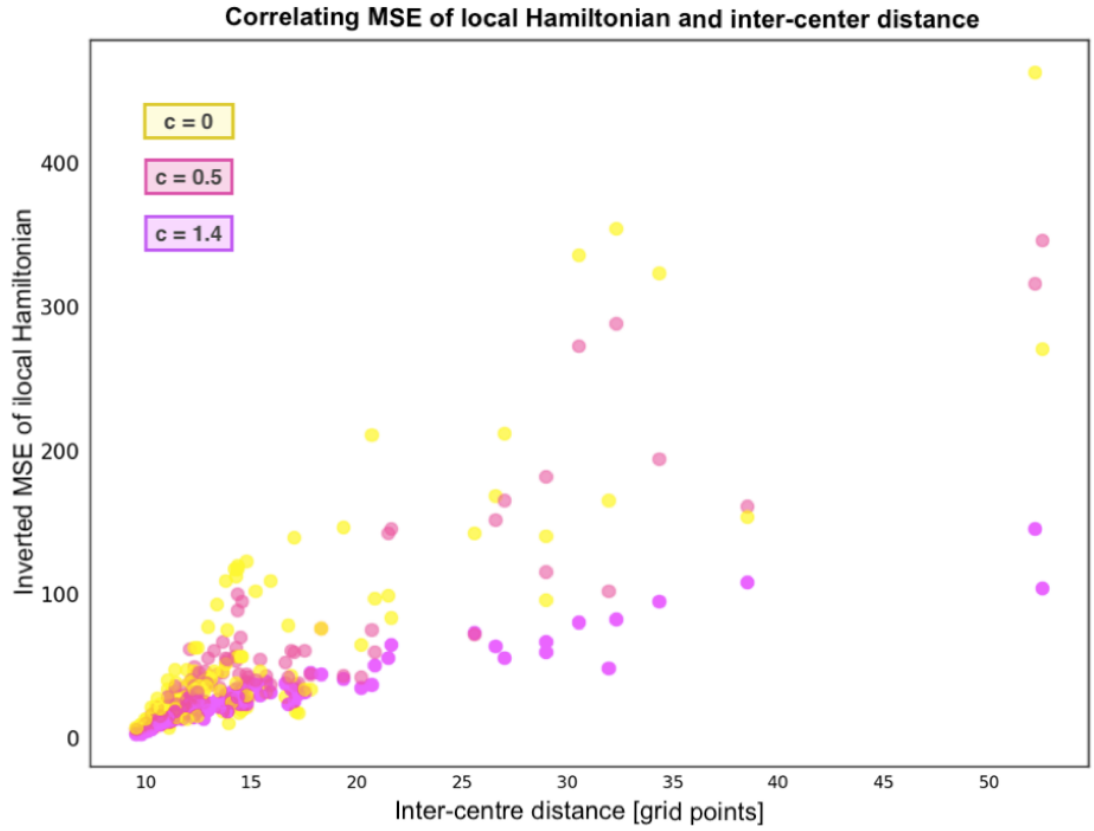

Figure 4.29 Correlating MSE of local Hamiltonian with inter-center distance.

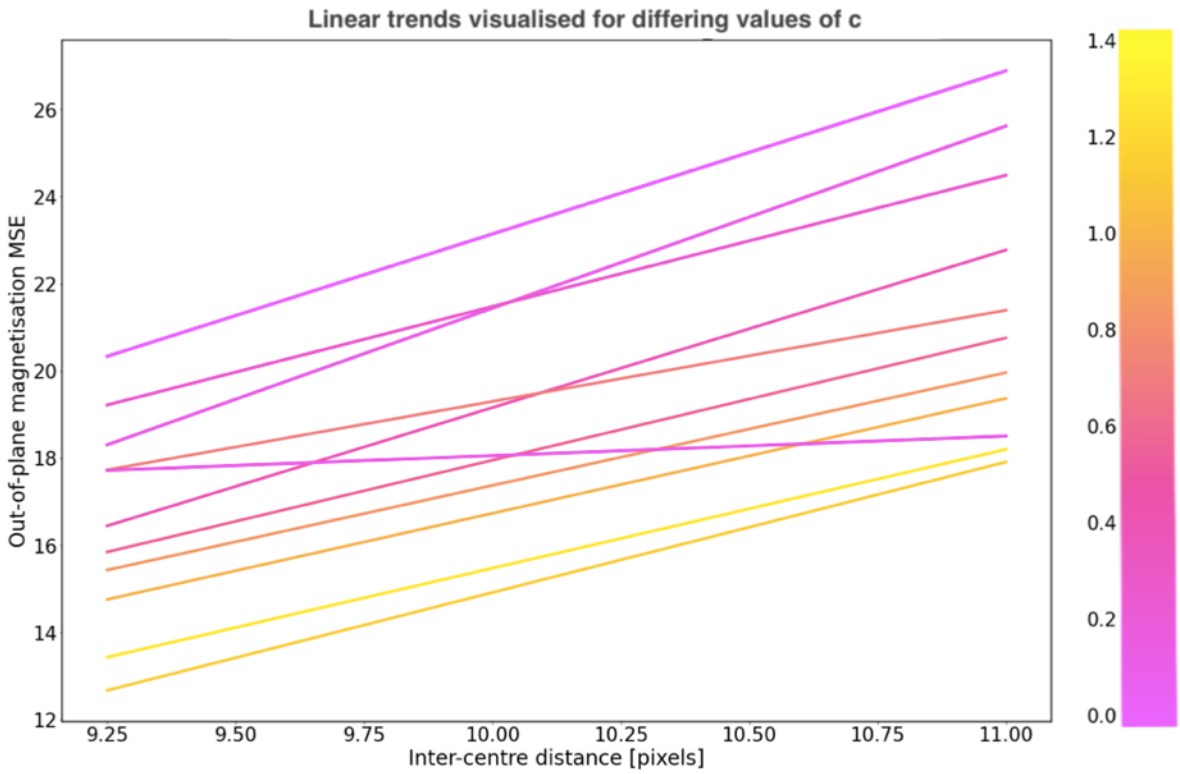

Figure 4.30 Visualising linear trends for out-of-plane MSE

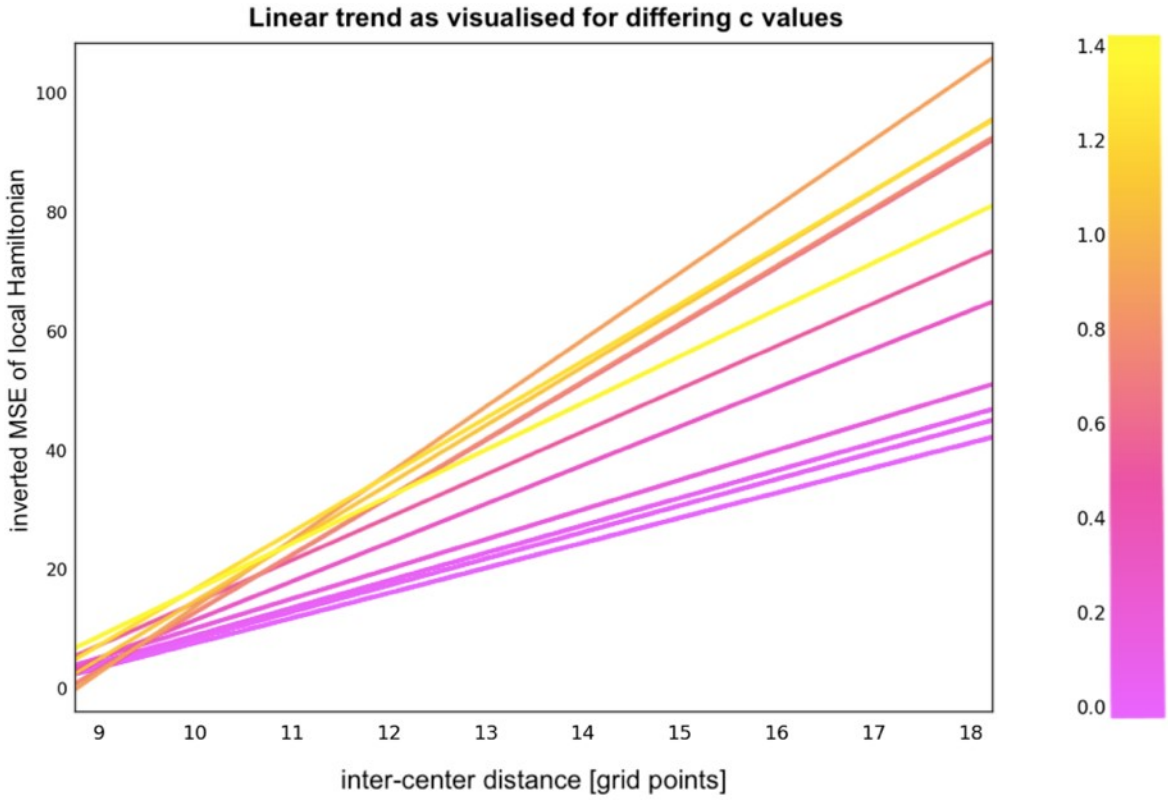

Figure 4.31 Visualising linear trends for MSE of local Hamiltonian

5 Discussion

For the classification of magnetic phase configurations, simple neural networks with only few layers are often employed. This approach allows magnetic phase recognition based on global parameters, such as global magnetisation or topological charge. To aid classification of complex dataset, the data can be transformed to alternative representation. (Zhang at al. 2014) One possibility is to work with latent space of autoencoder neural network which encodes complex data properties obtained through feature-extraction path. In theory, using latent space representation as an input for classifier, should help with discriminating between data differing only in a non-trivial way. Such approach was successfully applied for diagnosis of prostate cancer from MRI images (Abrahan and Nair 2018), diagnosis of osteoporosis disease (Nasser et al. 2017) and speech emotion recognition (Cibau et al. 2013) In this work, we investigated autoencoder reconstructing magnetic configurations, with a prospect of using its latent space representation to facilitate recognition based on complex topological features rather than global parameters.

For the set-up used in this work, one has two main options: Either to work with four-fold reducing autoencoder or with eight-fold reducing. The four-fold reducing offers reliable reconstruction, but less complex representation. Hence, with the motivation explained above, we might be very interested in knowing whether the eight-fold reducing autoencoder could be somehow modified so that its reconstruction improves. Inspired by the work of Kwon 2021b, we explored the possibility of doing this via Hamiltonian term.

The limitations of eight-fold autoencoder were most apparent on the skyrmionic phase, where there was a strong linear relationship between inverted magnetic MSE and inter-center distance for the interval of inter-center distance equal to 9-11 grid points (corresponding to smallest distances in our dataset). This was interpreted as densely packed skyrmions having so rich topology that it cannot be constrained smaller-scale xy-plane. (Since MSE didn't correlate with skyrmionic size, this must be a problem of scaling not resolution). For the Hamiltonian-containing autoencoder linear trend persevered, but trend-lines got shifted downwards. Interestingly, MSE didn't change outside the interval of inter-center distance 9-11, implying that sparse skyrmions were resistant to detrimental effect of Hamiltonian. As for the local Hamiltonian MSE, inclusion of the Hamiltonian had distance-independent positive effect. All this was explained via following:

As we saw on the plots in the beginning of this section, the Hamiltonian metric is more sensitive to small z-axis spin disturbances. (Energy plots were very similar to z-magnetisation plots. In

contrast to z-magnetisation plot, the energy plot appeared grainy - showing that it picked up small changes relative position of neighbouring spins.) Hence even the changes which would not influence magnetisation MSE greatly, can have big effect on local Hamiltonian. Including Hamiltonian in the loss function might therefore prompt the spins to organize more neatly.

Nevertheless, such attention on the detail probably would have come at cost of having less learning capacity to focus on truly problematic configuration. That might be the reason why Hamiltonian-containing autoencoder did not do so badly on the configurations with non-dense skyrmionic lattices - loose lattices never presented a problem for the autoencoder. On the other hand, increasing Hamiltonian contribution resulted in worsened performance on dense skyrmionic lattices which were autoencoder's weak point.

This discussion can also be extended to non-skyrmionic phases: Intermediate phases possess low symmetry, therefore are more topologically rich and harder to reconstruct. Hence as it would be expected, these were the phases for which the detrimental effect of the Hamiltonian was most clear to see. Particularly interesting was how the MSE for spirals lowered while MSE for bimeron-spiral phases increased. Both, bimeron-spiral and spiral phase had for some reason high Hamiltonian MSE. This means that Hamiltonian-containing autoencoder would try harder to reconstruct these two phases well. Nonetheless, this did not work out for bimeron-spiral indicating that they are extremely hard to reconstruct, possibly on account of their high topologically complexity.

Yet, this is a conjecture, similarly as most other explanations presented in this section. In the end, we don't know whether inclusion of Hamiltonian made our data more physical. Despite this, the prospect of using Hamiltonian-containing autoencoder for phase recognition might not be as bad as would the high reconstruction error indicate: For the phase-reconstruction task we don't need a perfect latent space representation, we just need it to be characteristic for each phase. (This was also the reason why we worked with so many phases – we wanted to be able to perform the evaluation for each phase separately.) From the fact that the reconstruction errors differ, we can infer that there are qualitative differences in how the phases are represented. Perhaps it could be even possible to perform the phase-recognition solely on a basis of reconstruction errors without requiring a classifier, as it was done in Betechuaoh et al. 2016.

Conclussion

To reach the goal of modifying the autoencoder with Hamiltonian term, the Hamiltonian was added in form of local Hamiltonian – a quantity representing energies of spins. Specifically, the loss was constructed so that it consisted of simple MSE term and MSE of local Hamiltonian multiplied by constant c, which varied in the range 0-1.4. For the c in the interval 0.8-1.4, the total magnetic MSE rose drastically with this trend applying to most of the phases when analysed separately. For the skyrmionic phase, this interval was characteristic by the presence of badly reconstructed outliers.

For the c in range 0-0.8, trends in magnetic MSE were phase specific. For spirals, MSE decreased until reaching levels expected for reliable reconstruction. For skyrmions, MSE rose insignificantly. For the bimeron-containing phase the errors rose drastically, with in-plane MSE rising most for the skyrmion-bimeron phase and off-plane MSE rising most for the bimeron-spiral phase. The errors of ferromagnetic and skyrmion gas phase remained negligible.

The quantitative differences in MSE profiles were also found for different densities of skyrmion lattices. For higher densities, MSE was high and was indirectly proportional to distance between skyrmions. This was interpreted as dense skyrmionic lattices containing so many nontrivial topological elements that it is impossible to represent them on the constrained latent space without facing information loss. Inclusion of Hamiltonian term had negative effect on off-plane MSE, but not in-plane MSE, and only for high density configurations.

As for the MSE of local Hamiltonian, inclusion of Hamiltonian term had phase-independent positive effect, with the gradient of change steepest for the lower values of c.

It was concluded that if the Hamiltonian-containing autoencoder is to be used in further applications (for instance in phase-recognition task, where the prospects seem hopeful), one should focus on the c in the interval 0-0.8.

Bibliography

Albarracín, F. A. G., Rosales, H. D. (2022). Machine learning techniques to construct detailed phase diagrams for skyrmion systems. Phys. Rev., B, 105, Article 214423.

Bejoy, A. Nair, M. S. (2018).Computer-aided diagnosis of clinically significant prostate cancer from MRI images using sparse autoencoder and random forest classifier. Biocyber. and Biomed. Eng. 38, Article 3.

Berg, B., Lüsher M. (1981). Definition and statistical distrubution of a topological number in lattice O(3) σ-model. Nucl. Phys. B., 190, p. 412-424.

Betechuoh, B. L. Tettey, T.Marwala T. (2006). Autoencoder networks for HIV classification. Curr. Sci., 91, Article 11.

Carrasquilla, J., Melko, R. (2017). Machine learning phases of matter. Nat. Phys., 13, Pages 431-434.

Chollet, F. & others, 2015. Keras.

Chollet, Francois. 2017. Deep Learning with Python. New York, NY: Manning Publications.

Cibau, N. E. Albornooz, E. M., Hugo, L. Speech emotion recognition using a deep autoencoder. Anales de la XV Reunion de Procesamiento de la Informacion y Control. 16. Pages 934-939.

Cortés-Ortuño, D., Wang, W., Beg, M., Pepper, R. A., Bisotti, M., Carey, R., Vousden, M., Kluyver, T., Hovorka O., Fangohr, H. (2017). Thermal stability and topological protection of skyrmions in nanotracks. Sci. Rep., 7, Article 4060.

Everschor-Sitte, K., Masell, J., Reeve R. M. (2018). Perspective: Magnetic skyrmions – Overview of recent progress in an active research field. J. Appl. Phys., 124, Article 240901.

Ezawa, M. (2011). Compact merons and skyrmions in thin chiral magnetic films. Phys. Rev. B., 83, Article 100408.

Güngördü, U., Nepal, R., Tretiakov, O. A., Belashchenko, K., Kovalev, A. A. (2016). Stability of skyrmion lattices and symmetries of quasi-two-dimesional chiral magnets. Phys. Rev., B, 93, Article 064428.

Heo, Ch., Kiselev, N. S., Nandy A. K., Blügel, S., Rasing, T.(2016).Switching of chiral magnetic skyrmions by picosecond magnetic field pulses via transient topological states. Sci. Rep., 6, Article 27146.

Iakolev, I. A., Sotnikov, O. M., Mazurenko, V. V. (2018) .Supervised learning approach for recognizing magnetic skyrmion phases. Phys. Rev.,96, Article 174411.

Iakovlev, I. A., Sotnikov, O. M., Mazurenko V. V. (2018). Bimeron nanoconfined design. Phys. Rev., 97, Article 184415.

Kapitan, V., Vasiliev, E., Perzhu, A., Kapitan, D., Rybin, A., Korol, A., Soldatov, K., Shevchenko, Yu.(2021).Numerical simulation of magnetic skyrmions on flat lattices. AIP Adv. ,11, Article 015041.

Kawamura, H. (2010). Two models of spin glasses – Ising versus Heisenberg. J. Phys. Conf. Ser., 233, Article 012012.

Kovalev, A. A., Sandhoefner, S. (2018). Skyrmions and antiskyrmions in quasi-two dimesional magnets. Front. Phys., 6, Article 98.

Kwon, H. Y., Kim, N. J., Lee, C. K., Won, C. (2019).Searching for magnetic states using an unsupervised machine learning algorithm with the Heisenberg model. Phys. Rev.,96, Article 024423.

Kwon, H. Y., Kim, N. J., Lee, C. K., Yoon, H. G., Choi, J. W., Won, C.(2019).An innovative magnetic state generator using machine learning techniques. Sci. Rep., 9, Arcticle 16709.

Kwon, H. Y., Yoon, H. G., Park, S. M., Lee, D. B., Choi, J. W., Won, Ch.(2021).Magnetic state generation using Hamiltonian guided variational autoencoder with spin structure stabilization. Adv. Sci.,8, Article 2004795.

Makino, K., Reim, J. D., Higashi, D., Okuyama, D., Sato, T. J., Nambu, Y., Gilbert, E. P., Booth, N., Seki, S., Tokura, Y. (2017). Thermal stability and irreversibility of skyrmion-lattice phases in Cu2OSeO3. Phys. Rev., B, 95, Article 134412.

Mathies, T., Schäffer, A. F., Posske, T., Wiesendager, R., Vedmedenko E. Y.(2022). Topological characterization of dynamic chiral magnetic textures using machine laerning. Phys. Rev. Applied, 17, Article 054022.

Mehta P, Wang CH, Day AGR, Richardson C, Bukov M, Fisher CK, Schwab DJ. (2019). A high-bias, low-variance introduction to Machine Learning for physicists. Phys Rep. 2019 May 30;810:1-124

Morningstar, A., Melko, R. G. (2018). Deep learning the Ising model near criticality. J. Mach. Learn. Res., 18, Article 5975.

Moskvin, A. (2019). Dzyaloshinskii-Moriya coupling in 3d insulators. Condens. Matter, 4(4), Article 84.

Pepper, A. R., Beg, M., Cortés-Ortuño, D., Kluyver, T., Bisotti, M., Carey, R., Vousden, M., Albert, M., Wang, W., Hovorka, O., Fangohr, H. (2018). Skyrmion states in thin confined polygonal nanostructures. J. Appl. Phys., 123, Article 093903.

Salcedo-Gallo, J. S., Galindo-González, C. C., Restrepo-Parra, E.(2020).Deep learning approach for image classification of magnetic phases in chiral magnets. J. Magn. Magn. Matter,501, Article 166482.

Wang, D., Wey, S., Yuan, A., Tian, F., Cao, K., Zhao, Q., Zhang, Y., Zhou, Ch., Song, X., Xue, D., Yang, S.(2020).Machine learning magnetic parameters from spin configurations. Adv. Sci.,16, Article 2000566.

Wang, L. (2016). Discovering phase transitions with unsupervised learning. Phys. Rev., B, 94, Article 195105.

Wetzel S. J.(2017).Unsupervised learning of phase transitions: from principal component analysis to variational autoencoders. Phys. Rev.,96, Article 022140.

Y. Nasser, M. E. Hassouni, A. Brahim, H. Toumi, E. Lespessailles and R. Jennane, 2017.Diagnosis of osteoporosis disease from bone X-ray images with stacked sparse autoencoder and SVM classifier, 2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP

Zhang, Y. Jianxin, W. Jianfei, C. (2014) Compact Representation for Image clssification: To Choose or to Compresss. 2014 IEEE Conference on Computer Vision and Pattern Recognition Pages. 907-914.

List of Abreviantions

CNN	Convolutional neural netwok
DM	Dzyaloshinskii-Moriya
MSE	Mean squared error
NN	Neural network