

Machine Learning approach to reconstruct Density matrices from Quantum Marginals

Daniel Uzcategui-Contreras

E-mail: daniel.uzcategui@ua.cl

Departamento de Física, Facultad de Ciencias Físicas y Matemáticas, Universidad de Concepción, Concepción, Chile

Millennium Institute for Research in Optics (MIRO), Chile

Antonio Guerra

Departamento de Física, Facultad de Ciencias Físicas y Matemáticas, Universidad de Concepción, Concepción, Chile

Millennium Institute for Research in Optics (MIRO), Chile

Sebastian Niklitschek

Departamento de Estadística, Facultad de Ciencias Físicas y Matemáticas, Universidad de Concepción, Concepción, Chile

Aldo Delgado

Departamento de Física, Facultad de Ciencias Físicas y Matemáticas, Universidad de Concepción, Concepción, Chile

Millennium Institute for Research in Optics (MIRO), Chile

Abstract. In this work, we propose a machine learning-based approach to address a specific aspect of the Quantum Marginal Problem: reconstructing a global density matrix compatible with a given set of quantum marginals. Our method integrates a quantum marginal imposition technique with convolutional denoising autoencoders. The loss function is carefully designed to enforce essential physical constraints, including Hermiticity, positivity, and normalization. Through extensive numerical simulations, we demonstrate the effectiveness of our approach, achieving high success rates and accuracy. Furthermore, we show that, in many cases, our model offers a faster alternative to state-of-the-art semidefinite programming solvers without compromising solution quality. These results highlight the potential of machine learning techniques for solving complex problems in quantum mechanics.

1. Introduction

Machine Learning (ML) has demonstrated remarkable capabilities across a broad range of scientific and technological fields, from classical tasks like regression and classification to sophisticated applications such as large language models (LLMs). Its success in

discovering complex, nonlinear relationships has accelerated advances in fundamental science [1, 2], mathematics [3], materials science [4], and meteorology [5]. In quantum mechanics, ML techniques have been applied to tasks such as quantum state tomography [6, 7] and quantum process tomography [8], both of which suffer from exponential scaling with system size, rendering them impractical for large systems. Additionally, physics-informed neural networks (PINNs) [9] have shown promise for solving quantum control problems through the resolution of partial differential equations like the Schrödinger equation [10].

In this context, the Quantum Marginal Problem (QMP) arises as a fundamental challenge in quantum theory: Given a set of reduced density matrices describing subsystems of a multipartite system, under what conditions are they consistent? Here, consistent means that there is a global density matrix, describing the entire system, that contains the reduced density matrices. This question naturally leads to several others: Can such a global state be uniquely determined? How is entanglement reflected in the marginals? What is the minimal information required to reconstruct the global state?

The consistency problem of the QMP is a decision problem that was shown to be complete for the Quantum Merlin-Arthur (QMA) complexity class [11, 12], highlighting its computational intractability in the general case. Solving the QMP would have profound implications. For example, it could enable more efficient calculations of the ground states of local Hamiltonians [13]. The QMP also plays a central role in quantum chemistry, where it is referred to as the N -representability problem. Understanding the conditions under which reduced wave functions of electron pairs are compatible with an N -electron system could lead to more efficient calculations of binding energies [14]. However, the N -representability problem is also known to be QMA complete, reflecting its computational difficulty [13].

Previous works have addressed special cases of the QMP. For example, Klyachko provided the necessary and sufficient conditions for compatibility of 1-body marginals with a global pure state in the nonoverlapping case [15], while other studies have explored the uniqueness of pure global states determined by reduced marginals [16, 17, 18]. Additionally, symmetric instances of the QMP have been formulated as semidefinite programming (SDP) problems [19].

Machine learning (ML) approaches have been proposed for problems in quantum many-body physics [20, 21, 22, 23]. Deep learning models have been shown to learn N -representability conditions from one- and two-electron systems and predict properties of larger systems [24, 25, 26]. In this work, we introduce a novel deep learning-based approach to approximate solutions to QMP. Our method combines the Marginal Imposition Operator (MIO) [27], a technique designed to impose arbitrary quantum marginals on a matrix, with a convolutional denoising autoencoder (CDAE) architecture. This combination enables the construction of physically valid quantum states (i.e., positive semidefinite with trace one) that are compatible with a prescribed set of marginals, even in the presence of noise or initial inconsistency.

Our architecture is capable of handling mixed quantum states for systems of 3 to 8

qubits. Additionally, we explore transfer learning between models trained on different numbers of qubits, showing that a model trained on 3-qubit systems can be adapted to systems with up to 8 qubits with limited retraining. This indicates that the network captures structural features of the compatibility conditions that are not specific to a fixed Hilbert space dimension.

The results presented in this article suggest that deep neural networks are powerful tools for approximating feasible solutions to the QMP, and they may offer a scalable alternative to traditional optimization-based methods. Furthermore, by integrating explainability techniques such as knowledge extraction [28, 29, 30, 31, 32], we believe that these models can contribute to a deeper theoretical understanding of the quantum marginal landscape.

2. The problem of compatibility

A quantum marginal $\sigma_{\mathcal{J}}$ is the reduced quantum state of a subsystem labeled by $\mathcal{J}$, derived from a larger system with global state $\rho_{\mathcal{I}}$. Let $\mathcal{I}$ denote a set of labels (e.g., $\mathcal{I} = 1, \dots, N$) representing the components of an N -body physical system. The subset $\mathcal{J} \subset \mathcal{I}$ corresponds to the components of a reduced subsystem. When the global state $\rho_{\mathcal{I}}$ is known, the marginal state $\sigma_{\mathcal{J}}$ can be obtained via the partial trace operation as:

$$\sigma_{\mathcal{J}} = \text{tr}_{\mathcal{J}^c} [\rho_{\mathcal{I}}] = \sum_j (\langle j^c | \otimes \mathbb{1}_{\mathcal{J}}) \rho_{\mathcal{I}} (|j^c\rangle \otimes \mathbb{1}_{\mathcal{J}}), \quad (1)$$

where $\mathcal{J} \cup \mathcal{J}^c = \mathcal{I}$. In equation (1), the identity operator $\mathbb{1}_{\mathcal{J}}$ acts on the subsystem $\mathcal{J}$, while $\{|j^c\rangle\}$ is an orthonormal basis tracing the complement $\mathcal{J}^c$.

Let us consider now the opposite problem. This is, given a set of M quantum marginals $\{\sigma_{\mathcal{J}_\alpha}\}_{\alpha=1}^M$, we would like to know whether there is a global quantum state $\rho_{\mathcal{I}}$ such that $\sigma_{\mathcal{J}_\alpha} = \text{tr}_{\mathcal{J}_\alpha^c} [\rho_{\mathcal{I}}]$ for all $\alpha = 1, \dots, M$. This problem can be stated as the SDP [33]

$$\begin{aligned} & \text{maximize} && \text{tr}(\rho_{\mathcal{I}}) \\ & \text{subject to} && \text{tr}_{\mathcal{J}_\alpha^c} [\rho_{\mathcal{I}}] = \sigma_{\mathcal{J}_\alpha}, \quad \alpha = 1, \dots, M \\ & && \rho_{\mathcal{I}} \geq 0 \end{aligned} \quad (2)$$

There are algorithms, such as interior point methods, that are highly efficient for solving SDPs. However, the runtime and memory requirements of these methods scale with the size of the matrix and the number of constraints [34, 35], making them impractical for large-scale instances of the problem. An iterative algorithm based on the Marginal MIO, which will be introduced in the next subsection, was proposed in [27]. Nevertheless, determining the computational complexity of this approach remains a challenging task.

2.1. The Marginal Imposition Operator

The problem of determining a global quantum state from a given set of quantum marginals has been previously studied [27], where the main tool introduced is the MIO.

Let $\sigma_{\mathcal{J}}$ be a $|\mathcal{J}|$ -party quantum state, with $\mathcal{J} \subset \mathcal{I}$. Then, the MIO is defined as follows:

$$\mathcal{Q}_{\mathcal{J}}(\tilde{\rho}_{\mathcal{I}}) := \tilde{\rho}_{\mathcal{I}} - \rho_{\mathcal{J}} \otimes \frac{\mathbb{1}_{\mathcal{J}^c}}{|\mathcal{J}^c|} + \sigma_{\mathcal{J}} \otimes \frac{\mathbb{1}_{\mathcal{J}^c}}{|\mathcal{J}^c|}, \quad (3)$$

where $\tilde{\rho}_{\mathcal{I}}$ is an $|\mathcal{I}|$ -party quantum state, which can be chosen at random or in any convenient fashion, and $\rho_{\mathcal{J}} = \text{tr}_{\mathcal{J}^c} [\tilde{\rho}_{\mathcal{I}}]$.

It is easy to see that $\text{tr}_{\mathcal{J}^c} [\mathcal{Q}_{\mathcal{J}}(\tilde{\rho}_{\mathcal{I}})] = \sigma_{\mathcal{J}}$. In simple terms, the MIO enforces the marginal $\sigma_{\mathcal{J}}$ onto $\tilde{\rho}_{\mathcal{I}}$, although $\mathcal{Q}_{\mathcal{J}}(\tilde{\rho}_{\mathcal{I}})$ may not necessarily represent a valid quantum state. Despite this, the MIO exhibits several interesting properties:

- (i) Hermitian: $\mathcal{Q}_{\mathcal{J}}(\tilde{\rho}_{\mathcal{I}}) = \mathcal{Q}_{\mathcal{J}}(\tilde{\rho}_{\mathcal{I}})^{\dagger}$
- (ii) Trace preserving: $\text{tr} [\mathcal{Q}_{\mathcal{J}}(\tilde{\rho}_{\mathcal{I}})] = 1$.

More generally, for a given set $\{\sigma_{\mathcal{J}_{\alpha}}\}_{\alpha=1}^M$, the composition $\mathcal{Q}_{\mathcal{J}_M} \circ \dots \circ \mathcal{Q}_{\mathcal{J}_1}(\tilde{\rho}_{\mathcal{I}})$ contains all the quantum marginals. This is

$$\sigma_{\mathcal{J}_{\alpha}} = \text{tr}_{\mathcal{J}_{\alpha}^c} [\mathcal{Q}_{\mathcal{J}_M} \circ \dots \circ \mathcal{Q}_{\mathcal{J}_1}(\tilde{\rho}_{\mathcal{I}})], \quad (4)$$

for all $\alpha = 1, \dots, M$. An obvious condition for equation (4) to be satisfied is that for every pair $\sigma_{\mathcal{J}_{\beta}}, \sigma_{\mathcal{J}_{\nu}} \in \{\sigma_{\mathcal{J}_{\alpha}}\}_{\alpha=1}^M$, the quantum states $\sigma_{\mathcal{J}_{\beta} \cap \mathcal{J}_{\nu}}$ in the overlapping subsystems must coincide when $\mathcal{J}_{\beta} \cap \mathcal{J}_{\nu} \neq \emptyset$.

However, there are no guarantees that $\mathcal{Q}_{\mathcal{J}_M} \circ \dots \circ \mathcal{Q}_{\mathcal{J}_1}(\tilde{\rho}_{\mathcal{I}}) \geq 0$. Ensuring positivity may depend on the choice of $\tilde{\rho}_{\mathcal{I}}$, as numerically demonstrated in [27], among other factors. Although one could attempt to find the closest quantum state [36], this approach incurs the computational cost of eigendecomposition. In the next section, we introduce CDAE, an ML model that offers a potential solution to this problem.

3. Model description

3.1. Image processing

ML has emerged as a powerful tool for image processing [37], a field encompassing techniques designed to transform an image into another representation based on a specific objective. The resulting output can be either another image or a classification/discrimination result. Examples of such transformations include resolution enhancement, noise reduction [38], image segmentation [39], and other feature extraction methods. A central goal of image processing is to preserve essential information while enhancing the relevant features of the original image.

A color image can be represented as a set of three matrices corresponding to the RGB color channels, each with the same dimensions, containing a fixed number of pixels in both height and width. This establishes a direct and intuitive correspondence between images and matrices.

Convolutional Neural Networks (CNNs) are designed to process multi-array-type data. For example, color images, which are represented as a set of three matrices corresponding to the RGB color channels, each with the same dimensions, containing a fixed number of pixels in both height and width. Many state-of-the-art models are built

upon CNN architectures, including Autoencoders, U-Nets, Variational Autoencoders (VAEs), and SegNet, each tailored for specific tasks. These models are widely used in applications such as denoising, segmentation, and feature extraction. A key property of CNNs is their ability to progressively capture hierarchical features while reducing the dimensionality of input images, enabling efficient and effective processing.

On the other hand, a fundamental concept in quantum mechanics is the density matrix, which encodes the information of a quantum system. These matrices consist of complex-valued elements and must satisfy specific mathematical properties: they must be PSD and have a unit trace. Consequently, they are Hermitian. For a quantum system formed by N qubits, the dimension of the space of density matrices grows exponentially with N .

A density matrix can be decomposed into its real and imaginary components, forming a multiarray: a set of two real-valued matrices. This suggests a natural analogy between density matrices and images, which in turn implies that image processing techniques may be adapted to quantum data. If CNNs are effective at extracting spatial features from images, it is reasonable to explore their application in learning quantum states.

However, this relationship is not straightforward. The convolutional layers of CNNs exhibit local connectivity, where each neuron is linked only to its neighboring units. This contrasts with fully connected layers, in which each neuron in one layer is connected to all units in the next layer. This localized connectivity is particularly beneficial in image processing, as it allows the extraction of relevant features while preserving spatial relationships, without requiring a global representation of the entire image.

In contrast, quantum density matrices do not inherently exhibit this locality-based structure. However, in QMP, we are primarily interested in global quantum states and their marginals, which are obtained through partial trace operations. This approach bears a conceptual similarity to the locality-based feature extraction performed in CNNs: While CNNs capture local structures within an image, partial traces extract relevant information from subsystems of a quantum state while maintaining a connection to the global system. This suggests that a CNN-based approach may be suitable for learning quantum states.

One of the main advantages of Deep Learning models for image processing is their scalability, as they are designed to handle images containing thousands or even millions of pixels. This scalability is particularly relevant in quantum mechanics, where the Hilbert space grows exponentially with the number of qubits. In this work, we present results for systems between 3 and 8 qubits. Training models for larger quantum systems requires significantly higher computational resources, which are currently beyond our reach. Nevertheless, we provide all the necessary tools for the further development of such models, facilitating their application to more complex quantum systems.

3.2. Autoencoders

Autoencoders are neural networks designed to learn efficient representations of data [40]. They consist of two main components: an encoder function, $\mathbf{h} = f(\mathbf{x})$, which maps the input data $\mathbf{x}$ to a lower-dimensional *latent representation* $\mathbf{h}$, and a decoder function, $\mathbf{r} = g(\mathbf{h})$, which reconstructs the original data from $\mathbf{h}$. A common application of autoencoders is image compression, where the encoder reduces an image to a compact latent representation, and the decoder reconstructs the image from this compressed form.

Autoencoders aim to approximately reconstruct the input data, $\mathbf{r} \approx \mathbf{x}$, as achieving exact reconstruction is often impractical. However, the strength of autoencoders lies in the compressed representation they generate, which can be leveraged for tasks such as denoising, anomaly detection, and feature extraction.

In many real-world applications, input data $\mathbf{x}$ may be corrupted by noise or other distortions, resulting in a noisy version $\tilde{\mathbf{x}}$. Despite the presence of noise, $\tilde{\mathbf{x}}$ often retains valuable information about the original data. Denoising Autoencoders [41] are a specialized type of autoencoder designed to learn a robust representation of $\mathbf{x}$ from noisy input data. This is achieved by minimizing a loss function that enforces the reconstructed output to align with the desired properties of $\mathbf{x}$. In quantum mechanics, Denoising Autoencoders have been successfully applied to quantum state tomography [42, 43].

To address the quantum marginal compatibility problem introduced in Section 2, we propose a CDAE architecture. To represent density matrices in a format suitable for the CDAE, we decompose them into two-channel tensors: one channel containing the real part and the other containing the imaginary part. This representation is analogous to how images are represented as multi-channel tensors, enabling the application of computer vision techniques, such as CNNs [44], to address our problem.

For N -qubit systems, each channel corresponds to a $2^N \times 2^N$ matrix. The full architectural specifications of the implemented model are detailed in Appendix A. Our CDAE architecture supports any N -qubit system with $N > 2$, as demonstrated in Appendix B. Figure 1 illustrates the proposed CDAE architecture. The encoder, which is responsible for extracting the latent representation, comprises the layers to the left of the latent representation in the figure. The decoder, which is tasked with reconstructing the original data from the latent code, consists of the layers to the right of the latent representation.

4. Approach

Let $\tilde{\mathbf{x}}$ and $\mathbf{z}$ be the input and output of our CDAE, respectively, such that $\mathbf{z} = g(f(\tilde{\mathbf{x}}))$. The input $\tilde{\mathbf{x}}$ is the two-channel representation of the corrupted density matrix $\tilde{X}$

$$\tilde{X} = \mathcal{Q}_{\mathcal{J}_M} \circ \dots \circ \mathcal{Q}_{\mathcal{J}_1}(\rho), \quad (5)$$

obtained by composing equation (3) for a given set $\{\sigma_{\mathcal{J}_\alpha}\}_{\alpha=1}^M$ of quantum marginals, with ρ serving as an initial seed state. The term 'corrupted' refers to the presence of

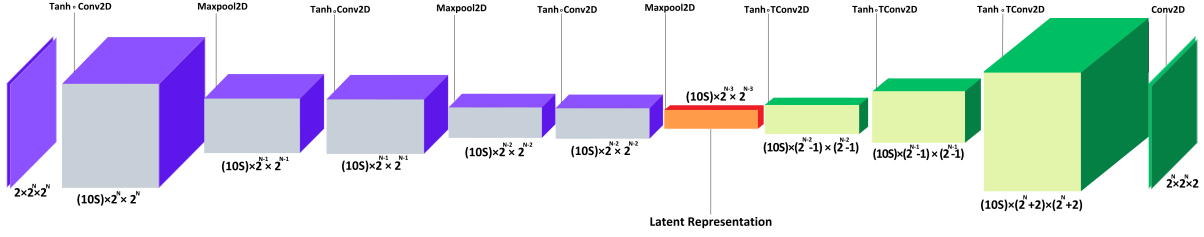


Figure 1. Illustration of the proposed CDAE architecture. The boxes represent tensors with dimensions specified as $(\text{number of channels}) \times \text{height} \times \text{width}$, indicated below each box. The operation applied to the tensor on the left is written above the arrow between two tensors, with the resulting tensor shown on the right. Since the Tanh activation function does not modify tensor dimensions, we simplify the notation by combining convolutional operations (Conv2D and TConv2D) with the Tanh activation function into single operations, denoted as $\text{Tanh} \circ \text{Conv2D}(\dots)$ and $\text{Tanh} \circ \text{TConv2D}(\dots)$. All boxes and operations to the left (right) of the *Latent Representation* correspond to the *encoder* (*decoder*). The leftmost box represents the input $\tilde{\mathbf{x}}$ of the CDAE, while the rightmost box represents its output $\mathbf{z}$.

negative eigenvalues in the matrix $\tilde{X}$, which indicates that it is not a valid quantum state, although it still contains quantum marginals, that is, $\sigma_{\mathcal{J}_\alpha} = \text{tr}_{\mathcal{J}_\alpha^c}[\tilde{X}]$ for all $\alpha = 1, \dots, M$. The output $\mathbf{z}$ is also a two-channel tensor, and its matrix representation Z can be obtained by taking the first channel as the real part and the second channel as the imaginary part.

In addition, we explore an approach that involves using the output of the CDAE as an initial seed for the MIO. Since the MIO is capable of reconstructing global states that match the given marginals with perfect fidelities but often suffers from the issue of producing unphysical, this hybrid strategy could mitigate this issue. By providing a well-structured initial guess from the trained autoencoder, the MIO may produce physically valid density matrices more reliably.

4.1. Data generation

Here, we describe the procedure used to generate the data for the training and testing stages. The set $\{\tilde{\mathbf{x}}_\beta, \mathcal{M}_\beta\}_{\beta=1}^{N_S}$ represents our training set, with N_S denoting the number of samples, and $\mathcal{M}_\beta = \{\sigma_{\mathcal{J}_\alpha}\}_{\alpha=1}^M$. Each set $\mathcal{M}_\beta$ is obtained by calculating all k -body quantum marginals of an N -qubit density matrix $\rho^{(\beta)}$, where $|\mathcal{M}_\beta| = \binom{N}{k}$. We choose $k = N - 1$ to generate the data in all the cases presented in this work. However, the models trained in this way also work for $0 < k < N - 1$, as demonstrated in the results section. We consider r -rank density matrices $\rho^{(\beta)}$, generated according to the following procedure:

- i) Randomly generate an integer r from a uniform distribution in the interval $r_{\min} \leq r \leq 2^N$, where $r_{\min}$ is a pre-defined parameter.
- ii) Generate a probability distribution vector $\mathbf{p}_\beta = (p_1^\beta, \dots, p_r^\beta)^T$.
- iii) Sample r pure states $|\psi_i^\beta\rangle$ from the Haar measure.

iv) Form the density matrix $\rho^{(\beta)} = \sum_{i=1}^r p_i^\beta |\psi_i^\beta\rangle\langle\psi_i^\beta|$.

Thus, we generate $\rho^{(\beta)}$ and then, using the partial trace $\sigma_{\mathcal{J}_\alpha} = \text{tr}_{\mathcal{J}_\alpha^c} [\rho^{(\beta)}]$, we obtain the elements of $\mathcal{M}_\beta$. With $\mathcal{M}_\beta$, we find $\tilde{X}_\beta$ using equation (5) and then its two-channel representation $\tilde{\mathbf{x}}_\beta$. We repeat this procedure N_S times to form the training set.

4.2. The loss function

To train our CDAE, we define a loss function that ensures that the output is a PSD matrix and compatible with the given quantum marginals. Let Z be the matrix representation of the output $\mathbf{z}$ of the CDAE. We can decompose Z into its polar form as $Z = UP$, where U is a unitary matrix and P is a PSD matrix. The loss function is given by:

$$\mathcal{L} = \ell(\text{Re}(U), \mathbb{1}) + \ell(\text{Im}(U), \mathbb{0}) + \sum_i^{|\mathcal{M}_\beta|} \ell(\sigma_{\mathcal{J}_i}, z_{\mathcal{J}_i}), \quad (6)$$

where $\ell(.,.)$ denotes the mean square error, $\mathbb{1}$ is the identity matrix, and $\mathbb{0}$ is the zero matrix. The first two terms in equation (6) ensure that the output matrix Z is PSD, while the third term accounts for the quantum marginals, with $z_{\mathcal{J}_i} = \text{tr}_{\mathcal{J}_i^c} [Z]$ representing the output marginal. Although we could explicitly include a term $\ell(\text{Tr}[Z], 1)$ in equation (6) to guarantee normalization, we find that this constraint is approximately satisfied without the need to explicitly include it. The final output matrix can be renormalized as $Z/\text{Tr}[Z]$ to ensure exact normalization. Additionally, in practice, Z is approximately Hermitian, which can be fixed by setting $Z = (Z + Z^\dagger)/2$, thus avoiding nonreal eigenvalues.

4.3. Training

For training, we used Adam Optimizer with a learning rate of 10^{-4} and a batch size of 100 for all the cases presented. The model was implemented in PyTorch, and all numerical experiments were conducted on a workstation equipped with 125 GB of RAM, an NVIDIA GeForce RTX 4090 GPU and an AMD Ryzen Threadripper 7980X 64-core CPU. The complete code and instructions for reproduction are available in a GitHub repository [45].

The architecture employed in this work demonstrates sufficient generalization capacity, successfully adapting to global quantum states ranging from 3 to 8 qubits without requiring significant structural modifications. As a result, there is no need to design new architectures or increase the model capacity for each case (e.g., by deepening the network or increasing the number of convolutional filters). This flexibility enables the use of transfer learning as a strategy for extending the model to systems with a higher number of qubits.

Transfer learning allows us to initialize training for a model targeting N qubits from a previously trained model with $N - 1$ qubits, rather than training from scratch. Our results indicate that this approach substantially reduces both the training time

and the amount of data required. During the transfer process, certain layers, typically the encoder, are frozen, i.e., their gradients are deactivated, and only selected layers (often in the decoder) are updated. We begin with a base model trained on 3-qubit global states. Once trained to an acceptable level of performance—achieving fidelities of approximately 0.98 for a random case—we use this model as a starting point for training a model for 4-qubit global states. In this case, it was sufficient to retrain only the final decoder layer, yielding fidelities of around 0.97 for a test case. This strategy is repeated to scale up the model. However, in some cases, retraining only the decoder was insufficient. For example, when transitioning from 4 to 5 qubits, retraining the full model was required to achieve average fidelities above 0.95 for randomly selected cases. Interestingly, for larger systems (6, 7, and 8 qubits), retraining only the decoder once again proved sufficient. In these cases, transfer learning from the immediately smaller model led to successful generalization, enabling high-fidelity global state reconstruction based on marginal inputs.

These findings suggest a remarkable property of the network: the encoder appears to capture the relevant marginal information in a way that is transferable across different system sizes. This highlights the model’s ability to internalize the underlying non-linear structure of the QMP and leverage it effectively across varying dimensional scales.

5. Results

In this section, we present the results of our experiments to evaluate the performance of the proposed CDAE in solving the task of finding a quantum state compatible with a given set of quantum marginals. We focus on two key metrics: the success rate and accuracy. The success rate measures the fraction of outputs predicted by the model that are valid PSD matrices.

To assess the accuracy of the predicted quantum marginals, we use the *mean fidelity*, denoted F_{mean} . For each sample β (see 4.1) in the test set, we compute the average fidelity $\bar{F}_\beta$ over all the marginals in that sample as $\bar{F}_\beta = \sum_{i=1}^{|\mathcal{M}_\beta|} F(\sigma_{\mathcal{J}_i}, z_{\mathcal{J}_i}) / |\mathcal{M}_\beta|$, with $F(\sigma_{\mathcal{J}_i}, z_{\mathcal{J}_i}) = \text{tr} [\sqrt{\sqrt{\sigma_{\mathcal{J}_i}} z_{\mathcal{J}_i} \sqrt{\sigma_{\mathcal{J}_i}}}]^2$ the fidelity between the target marginal $\sigma_{\mathcal{J}_i}$ and the predicted marginal $z_{\mathcal{J}_i}$. Finally, the mean fidelity F_{mean} is obtained by averaging $\bar{F}_\beta$ over all samples in the test set; this is $F_{mean} = \sum_{\beta=1}^{N_{test}} \bar{F}_\beta / N_{test}$, where N_{test} is the number of samples. To show variability in the data, we use the standard deviation (SD), calculated as $SD = \sqrt{(N_{test} - 1)^{-1} \sum_{\beta=1}^{N_{test}} (F_{mean} - \bar{F}_\beta)^2}$.

We consider N -qubit global states with k -qubit marginals, denoted as $N\#k\#$. We label the model that uses the full loss function from equation (6) as *model1*, while *model2* uses equation (6) without the marginals term $\sum_i \ell(\sigma_{\mathcal{J}_i}, z_{\mathcal{J}_i})$. We determine the scaling parameter S by experimentation, keeping the minimum value of S that produces satisfactory levels of accuracy and success rate.

To evaluate the performance of *model1* and *model2* in the case $N3k2$, we set the scaling factor $S = 10$ and trained both models for 100 epochs using the same data set (see Section 4.1) of size 10^6 , with weights initialized in the same seed. We then evaluated

the performance of these trained models in both the $N3k1$ and $N3k2$ cases. The values of $F_{mean} \pm SD$, calculated over 10^4 samples per rank, are shown in figure 2a and figure 2b, respectively, for ranks up to 2^N . As expected, predictions from *model1* (purple squares) consistently achieve higher fidelity with target marginals compared to *model2* (green dots) for all ranks, and both models achieve a high success rate. This demonstrates the importance of including the marginals term in the loss function. Moreover, predictions from *model1* outperform *random guessing* (red triangles), which involve generating random N -qubit density matrices, calculating their k -qubit marginals, and computing the fidelity with the target marginals.

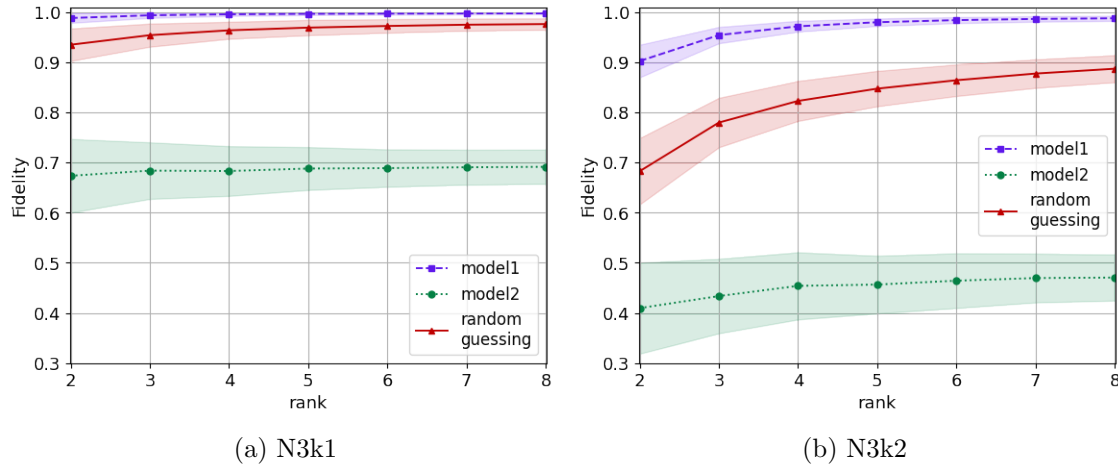


Figure 2. Mean and standard deviation (colored cloud) of the fidelity, obtained over 10^4 samples, versus *rank* for *model1* (purple squares), *model2* (green dots) and *random guessing* (red triangles) for the cases (a) $N3k1$ and (b) $N3k2$.

In figure 3, we compare *model1* to random guessing on systems with $N > 3$, up to $N = 8$ qubits, using the mean fidelity F_{mean} as a performance metric. The comparison is carried out for different values of k . We observe that *model1* consistently outperforms random guessing in all scenarios, with the performance gap increasing as k grows. This suggests that the model retains the information provided by the marginals with high accuracy. Furthermore, *model1* achieved high success rates for all cases shown in figure 3: above 95% for the case $N4k2$, and above 99% for the remaining cases -some of which reached 100%- as detailed in figure 4.

Moreover, we tested a scheme in which the output of *model1* is passed through the MIO (*model1*+MIO). By first applying the MIO and then feeding its output into the CDAE, we obtained a global state that closely reproduces the target marginals with high fidelity. This resulting state can then serve as a new seed for a subsequent pass through the MIO, as given by equation (5), using the given target marginals $\{\sigma_{\mathcal{J}_\alpha}\}_{\alpha=1}^M$. Depending on the number k of qubits in the quantum marginals, some corrupted (i.e. negative) eigenvalues may still persist. However, we observed that, in many cases, the final output contains no corrupted eigenvalues, or at least fewer than those present after

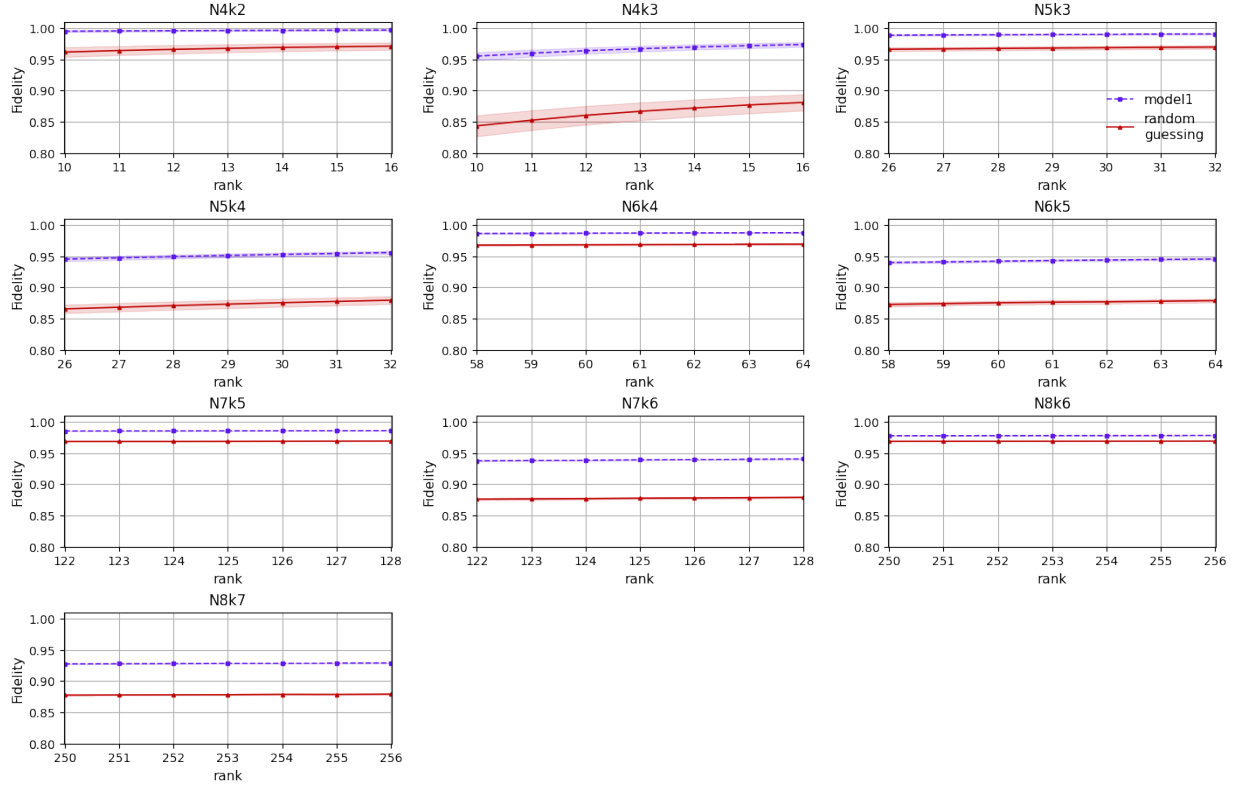


Figure 3. $F_{mean} \pm SD$, over 10^4 samples, as a function of *rank* for *model1* (purple dashed line) and for *random guessing* (solid red line) for the cases $N4k2$, $N4k3$, $N5k3$, $N5k4$, $N6k4$, $N6k5$, $N7k5$, $N7k6$, $N8k6$ and $N8k7$.

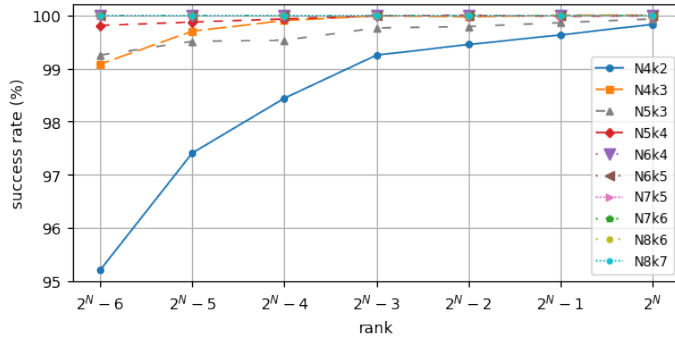


Figure 4. Success rates of *model1* for all the cases presented in Figure 3. *model1* achieved a 100% success rate for all the cases with $N > 5$, with the lines overlapping at the 100% mark.

the initial MIO pass, see Figure 5 for details. Additionally, the magnitude of these corrupted eigenvalues tends to decrease by approximately one order of magnitude.

In figure 6, we show results for cases in which the final output of this scheme (*model1* + MIO) is a valid quantum state whose marginals match the target marginals with perfect fidelity. Success rates of these cases are shown in figure 7. For systems with $N > 5$, all the instances were perfectly reconstructed, achieving a success rate of 100%, whereas for the cases $N4k2$ and $N5k2$ the success rate consistently increases with

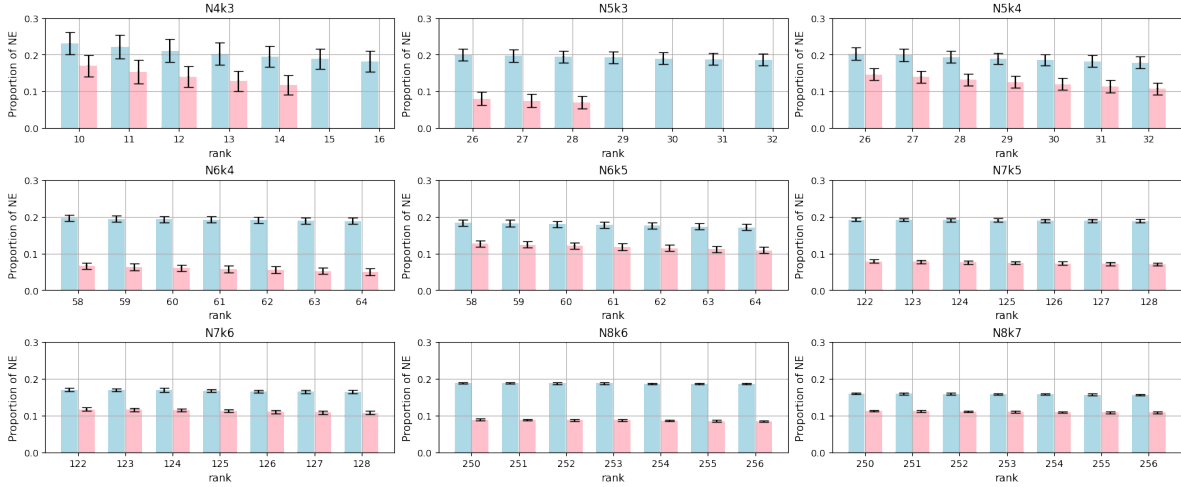


Figure 5. Comparison of the proportion of negative eigenvalues relative to the total number of eigenvalues. Each bar represents the average over a sample of 10^4 tests per rank, with error bars indicating the standard deviation. Light blue bars correspond to the first pass through the MIO, while pink bars correspond to the *model1*+MIO configuration.

the rank. These results suggest that further improvements to the model, such as adding more layers, increasing filter sizes, or training on larger datasets, could potentially lead the model to find exact solutions in all the cases.

We experimented with iterating the CDAE and MIO steps multiple times in cases where exact solutions were not initially found, but observed no significant improvement beyond the *model1*+MIO scheme. This suggests that a single iteration is sufficient to achieve this form of “purification”. We hypothesize that this is because the remaining negative eigenvalues after the second MIO pass are too small to be identified by the CDAE as noise in subsequent iterations.

In addition, we compare the performance of our model against state-of-the-art SDP solvers in terms of computation time. Figure 8 shows the average runtime computed over 10^3 samples for cases with $N < 6$, 100 samples for the case with $N = 6$ and 50 samples for cases with $N > 6$. In all cases, the target marginals were obtained from full-rank density matrices. As shown in figure 8, once trained, both *model1* and *model1*+MIO execute faster than the SDP solver provided by the CVXPY library in Python. While the SDP solver consistently found solutions matching the target marginals with perfect fidelity, it failed to run for the *N8k4* case on our machine. In contrast, our *model1*+MIO scheme successfully found solutions with exact marginals for *N8k4*, with an average runtime of approximately 3.38 seconds.

6. Conclusions and Discussion

We have successfully developed a CDAE and combined it with the MIO [27] to find a global N -qubit density matrix compatible with a given set of quantum marginals. Our

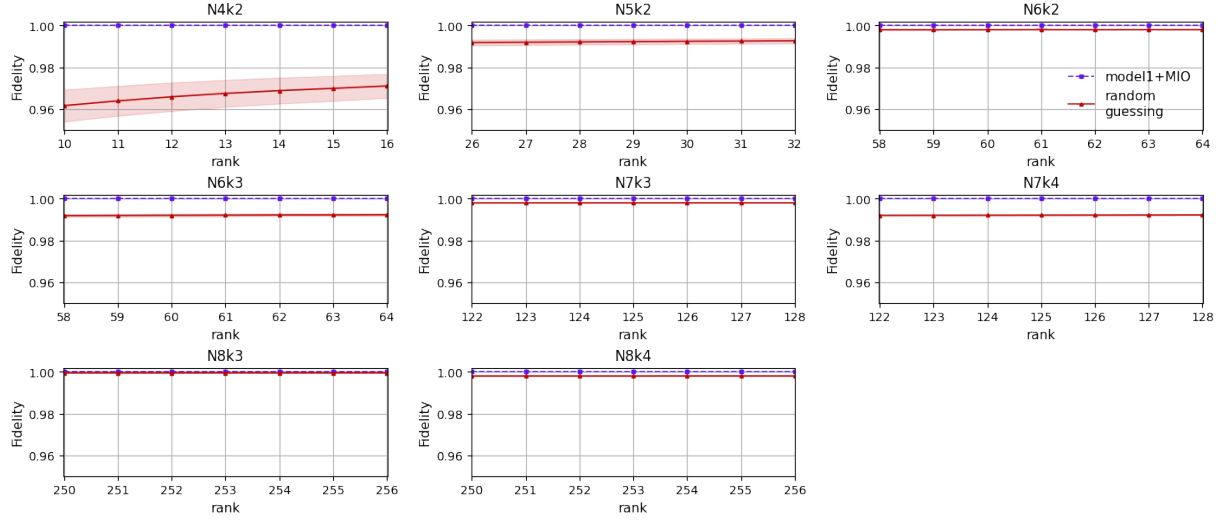


Figure 6. $F_{mean} \pm SD$, over 10^4 samples, as a function of *rank* for *model1* with an additional MIO pass (purple dashed line) and for *random guessing* (solid red line). Results are shown for the cases $N4k2$, $N5k2$, $N6k2$, $N6k3$, $N7k3$, $N7k4$, $N8k3$ and $N8k4$. All the states produced by the *model1*+MIO scheme match the target marginals with perfect fidelity.

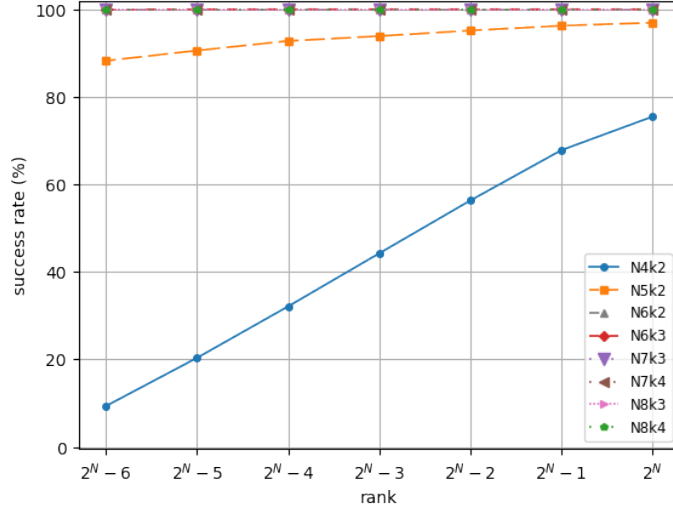


Figure 7. Success rates of the *model1*+MIO scheme for all the cases presented in Figure 6. The scheme achieved a 100% success rate for all the cases with $N > 5$, with the lines overlapping at the 100% mark.

model, which admits any N -qubit global system with more than 2 qubits, transforms, in most cases, the output of the MIO into a PSD matrix, preserving information about the quantum marginals with high accuracy.

Transfer learning proved to be a valuable technique in our approach, allowing us to reduce the size of the training data set and improve computational efficiency. By keeping parts of the CDAE fixed and retraining the rest of the architecture, we were able to leverage the knowledge learned in smaller-scale cases to improve performance on

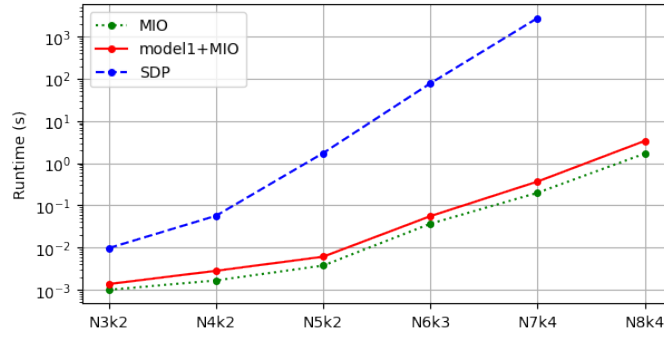


Figure 8. Average computation time of the CVXPY SDP solver (blue dashed line), *model1* (green dotted line), and *model1*+MIO (red solid line) across different system sizes. The vertical axis is in logarithmic scale. The SDP solver failed to run for the 8-qubit case.

larger-scale problems.

Although our model demonstrates promising results, it is important to note that it may struggle with cases where the quantum marginals correspond to pure global states. Previous theoretical studies have shown that the representation of the global state is unique in such cases, making it more challenging for our model to find a specific solution. However, for mixed states, multiple representations exist, increasing the likelihood of finding an acceptable solution.

Our work makes several contributions. First, the two-channel encoding of a density matrix introduced here offers a novel approach for studying quantum states using convolutional neural networks (CNNs). Our numerical results demonstrate that CNNs are capable of capturing fundamental features of density matrices. As noted in [30], the softmax layer in convolutional neural networks and autoencoders, when using tanh or ReLU activation functions, is highly explicable through rule extraction. This suggests that convolutional autoencoders, such as the one used in this work, may be useful for uncovering new insights into challenging problems such as the QMP. Moreover, autoencoders have been successfully applied as anomaly detectors in various domains [46]. Future work could explore the use of convolutional autoencoders to distinguish between compatible quantum marginals and anomalies (incompatible marginals).

We demonstrated that combining the MIO with a CDAE enables the construction of quantum states compatible with a given set of quantum marginals, preserving them with high accuracy. In specific cases, a second application of the MIO further improves the result, yielding quantum states whose marginals match the inputs exactly. Although training the CDAE can be computationally intensive, once trained, the inference process is significantly faster compared to current state-of-the-art SDP solvers, such as those provided by the CVXPY Python library. Therefore, in the long term, our approach presents a promising and more efficient alternative to existing SDP-based methods.

The ability of our model to generate accurate approximations might serve to provide *initial guess* to solve the SDP of equation (2), potentially reducing its computational cost and accelerating its convergence. The technique of using an initial guess close to the

solution is called *warm-start* and is an active subject of research. Although warm-start has found difficulties in being implemented in SDPs, recent work [47] shows its viability.

Acknowledgments

This work was supported by ANID grants No. 2023 – 3230427 and No. 2022 – 21221096, and the ANID Millennium Science Initiative Program ICN17_012. We also thank Proyecto FONDECYT Regular No. 1230586, and Dr. Dardo Goyeneche for his feedback.

Appendix A. Model architecture

In this section, we describe the specifics of the proposed CDAE. The encoder consists of multiple layers, incorporating 2D convolution (*Conv2D*) operations, 2D max pooling (*MaxPool2D*) operations [48], and Hyperbolic Tangent (*Tanh*) activation functions. The decoder includes *Conv2D* operations and *Tanh* activation functions, along with layers utilizing 2D transposed convolutions (*TConv2D*).

The tables A1 and A2 provide detailed specifications of each layer in the encoder and decoder, respectively. All convolutional and transposed convolutional layers use a dilation factor of one. In the decoder, the output padding for transposed convolutions is fixed at zero. The intermediate number of channels in both encoder and decoder is controlled by a scaling factor S , which allows for flexible adjustment of the model’s capacity. Lower values of S reduce the number of trainable parameters, offering a trade-off between model complexity and computational efficiency.

Table A1. Specifications for each encoder layer. Dilation is fixed to one throughout the network.

Layer	Operation	Input channels	Output channels	Kernel size	Stride	Padding
(0)	Conv2D	2	$S \times 10$	3×3	1	1
(1)	Tanh	–	–	–	–	–
(2)	MaxPool2D	–	–	2×2	2	0
(3)	Conv2D	$S \times 10$	$S \times 10$	3×3	1	1
(4)	Tanh	–	–	–	–	–
(5)	MaxPool2D	–	–	2×2	2	0
(6)	Conv2D	$S \times 10$	$S \times 10$	3×3	1	1
(7)	Tanh	–	–	–	–	–
(8)	MaxPool2D	–	–	2×2	2	0

Appendix B. The CDAE compatibility

This appendix provides a detailed analysis of the channel sizes in the CDAE presented in section 4. Each input to the CDAE is a two-tensor channel, where each channel is a $2^N \times 2^N$ matrix. We demonstrate that for $N > 2$, the channel size of the CDAE’s

Table A2. Specifications for each decoder layer. Dilation is set to one, and output padding in transposed convolutions (*TConv2D*) is fixed at zero.

Layer	Operation	Input channels	Output channels	Kernel size	Stride	Padding
(0)	TConv2D	$S \times 10$	$S \times 10$	3×3	2	1
(1)	Tanh	—	—	—	—	—
(2)	TConv2D	$S \times 10$	$S \times 10$	5×5	2	1
(3)	Tanh	—	—	—	—	—
(4)	TConv2D	$S \times 10$	$S \times 10$	6×6	2	0
(5)	Tanh	—	—	—	—	—
(6)	Conv2D	$S \times 10$	2	3×3	1	0

output remains consistent at $2^N \times 2^N$. The size of a channel can change when *Conv2D* operations, *MaxPool2D* operations and *TConv2D* operations are applied to tensors at different layers of the CDAE. Our model considers a setting with square channels, square kernels, the same stride along both axes and the same padding along both axes. Under these conditions, the channel size for *Conv2D* and *MaxPool2D* operations can be calculated using the following formula [48]

$$\mathcal{C}_{out(j)}^{E/D} = \left\lfloor \frac{\mathcal{C}_{in(j)}^{E/D} + 2\mathcal{P} - \mathcal{D}(\mathcal{K} - 1) - 1}{\mathcal{S}} + 1 \right\rfloor, \quad (\text{B.1})$$

with $\lfloor \dots \rfloor$ the floor function. $\mathcal{C}_{in(j)}^{E/D}$ and $\mathcal{C}_{out(j)}^{E/D}$ are the input and output size, respectively, of a channel at layer (j) of the encoder(E)/decoder(D). $\mathcal{P}$ is the padding, $\mathcal{D}$ the dilation, $\mathcal{K}$ the kernel size and $\mathcal{S}$ the stride. The dilation is set to $\mathcal{D} = 1$ across the entire CDAE.

Note that $\mathcal{C}_{in(l)}^{E/D} = \mathcal{C}_{out(l-1)}^{E/D}$ for $l > 0$. This is, the input size of layer (l) is equal to the output size of the previous layer ($l - 1$). The input size of the encoder is $\mathcal{C}_{in(0)}^E = 2^N$. Let us now determine its output size $\mathcal{C}_{out(8)}^E$ using the values for $\mathcal{P}$, $\mathcal{D}$, $\mathcal{K}$ and $\mathcal{S}$ given in Table A1.

$$\begin{aligned} \mathcal{C}_{out(0)}^E &= \left\lfloor \frac{\mathcal{C}_{in(0)}^E + 2 \times 1 - 1 \times (3 - 1) - 1}{1} + 1 \right\rfloor, \\ \mathcal{C}_{out(0)}^E &= \left\lfloor \mathcal{C}_{in(0)}^E \right\rfloor = 2^N. \end{aligned} \quad (\text{B.2})$$

The size of the channels is not affected by *Tanh* operations. Thus, $\mathcal{C}_{out(1)}^E = \mathcal{C}_{out(0)}^E$ and therefore

$$\mathcal{C}_{out(2)}^E = \left\lfloor \frac{\mathcal{C}_{out(1)}^E + 2 \times 0 - 1 \times (2 - 1) - 1}{2} + 1 \right\rfloor, \quad (\text{B.3})$$

$$\mathcal{C}_{out(2)}^E = \left\lfloor 2^{-1} \mathcal{C}_{out(1)}^E - 1 + 1 \right\rfloor = \left\lfloor 2^{-1} \mathcal{C}_{out(1)}^E \right\rfloor, \quad (\text{B.4})$$

$$\mathcal{C}_{out(2)}^E = 2^{N-1}.$$

Then, we calculate $\mathcal{C}_{out(3)}^E$, which is similar to $\mathcal{C}_{out(0)}^E$

$$\mathcal{C}_{out(3)}^E = \left\lfloor \mathcal{C}_{out(2)}^E \right\rfloor = 2^{N-1}. \quad (\text{B.5})$$

Now, using $\mathcal{C}_{out(4)}^E = \mathcal{C}_{out(3)}^E$, we calculate $\mathcal{C}_{out(5)}^E$

$$\mathcal{C}_{out(5)}^E = \left\lfloor 2^{-1} \mathcal{C}_{out(4)}^E \right\rfloor = 2^{N-2}, \quad (\text{B.6})$$

and then $\mathcal{C}_{out(6)}^E$

$$\mathcal{C}_{out(6)}^E = \lfloor \mathcal{C}_{out(5)}^E \rfloor = 2^{N-2}. \quad (\text{B.7})$$

Finally, the output size of the encoder $\mathcal{C}_{out(8)}^E$ is

$$\mathcal{C}_{out(8)}^E = \lfloor 2^{-1} \mathcal{C}_{out(7)}^E \rfloor = 2^{N-3}. \quad (\text{B.8})$$

with $\mathcal{C}_{out(7)}^E = \mathcal{C}_{out(6)}^E$. Note that for $N < 3$ the encoder would render an invalid size. Now, let us calculate the channel's output size of the decoder using the values given in Table A2. This is, the output size at the layer (6) of the decoder. For *TConv2D*, the channel's size $\mathcal{D}_{out(j)}$ changes according to [48]

$$\mathcal{D}_{out(j)} = (\mathcal{D}_{in(j)} - 1) \mathcal{S} - 2\mathcal{P} + \mathcal{D}(\mathcal{K} - 1) + \mathcal{P}_{out} + 1, \quad (\text{B.9})$$

where $\mathcal{P}_{out}$ is the output padding, which is set to zero across the entire CDAE. As for the encoder, $\mathcal{D}_{in(l)} = \mathcal{D}_{out(l-1)}$ for $l > 0$. The input size for the first layer of the decoder is $\mathcal{C}_{out(8)}^E$. Thus,

$$\mathcal{D}_{out(0)} = (\mathcal{C}_{out(8)}^E - 1) 2 - 2 + 1(3 - 1) + 1, \quad (\text{B.10})$$

$$\mathcal{D}_{out(0)} = (2^{N-3} - 1) 2 + 1, \quad (\text{B.11})$$

$$\mathcal{D}_{out(0)} = 2^{N-2} - 1.$$

Now, knowing that $\mathcal{D}_{out(1)} = \mathcal{D}_{out(0)}$, let us determine $\mathcal{D}_{out(2)}$

$$\mathcal{D}_{out(2)} = (\mathcal{D}_{out(1)} - 1) 2 - 2 + 1(5 - 1) + 1, \quad (\text{B.12})$$

$$\mathcal{D}_{out(2)} = (2^{N-2} - 1 - 1) 2 - 2 + 4 + 1, \quad (\text{B.13})$$

$$\mathcal{D}_{out(2)} = 2^{N-1} - 1,$$

and then,

$$\mathcal{D}_{out(4)} = (\mathcal{D}_{out(3)} - 1) 2 - 2 \times 0 + 1(6 - 1) + 1, \quad (\text{B.14})$$

$$\mathcal{D}_{out(4)} = (2^{N-1} - 1 - 1) 2 + 5 + 1, \quad (\text{B.15})$$

$$\mathcal{D}_{out(4)} = 2^N + 2,$$

where $\mathcal{D}_{out(3)} = \mathcal{D}_{out(2)}$. Finally, we calculate the output size at layer (6) using (B.1)

$$\mathcal{C}_{out(6)}^D = \left\lfloor \frac{\mathcal{D}_{out(5)} + 2 \times 0 - 1(3 - 1) - 1}{1} + 1 \right\rfloor, \quad (\text{B.16})$$

$$\mathcal{C}_{out(6)}^D = \lfloor \mathcal{D}_{out(5)} - 2 \rfloor, \quad (\text{B.17})$$

$$\mathcal{C}_{out(6)}^D = \lfloor 2^N + 2 - 2 \rfloor = 2^N,$$

where $\mathcal{D}_{out(5)} = \mathcal{D}_{out(4)}$. Thus, we have shown that the CDAE input and output sizes are equal.

References

- [1] Georgia Karagiorgi, Gregor Kasieczka, Scott Kravitz, Benjamin Nachman, and David Shih. Machine learning in the search for new fundamental physics. *Nature Reviews Physics*, 4(6):399–412, 2022.

- [2] David Pfau, James S. Spencer, Alexander G. D. G. Matthews, and W. M. C. Foulkes. Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Phys. Rev. Res.*, 2:033429, Sep 2020.
- [3] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.
- [4] Amil Merchant, Simon Batzner, Samuel S. Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023.
- [5] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.
- [6] Tobias Schmale, Moritz Reh, and Martin Gärttner. Efficient quantum state tomography with convolutional neural networks. *npj Quantum Information*, 8(1):115, 2022.
- [7] Juan Carrasquilla, Giacomo Torlai, Roger G. Melko, and Leandro Aolita. Reconstructing quantum states with generative models. *Nature Machine Intelligence*, 1(3):155–161, 2019.
- [8] Giacomo Torlai, Christopher J. Wood, Atithi Acharya, Giuseppe Carleo, Juan Carrasquilla, and Leandro Aolita. Quantum process tomography with unsupervised learning and tensor networks. *Nature Communications*, 14(1):2858, 2023.
- [9] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [10] Ariel Norambuena, Marios Mattheakis, Francisco J. González, and Raúl Coto. Physics-informed neural networks for quantum control. *Phys. Rev. Lett.*, 132:010801, Jan 2024.
- [11] Yi-Kai Liu. Consistency of local density matrices is qma-complete. In Josep Díaz, Klaus Jansen, José D. P. Rolim, and Uri Zwick, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 438–449, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [12] Anne Broadbent and Alex Bredariol Grilo. Qma-hardness of consistency of local density matrices with applications to quantum zero-knowledge. *SIAM Journal on Computing*, 51(4):1400–1450, 2022.
- [13] Yi-Kai Liu, Matthias Christandl, and F. Verstraete. Quantum computational complexity of the n -representability problem: Qma complete. *Phys. Rev. Lett.*, 98:110503, Mar 2007.
- [14] Brian R. Judd. Reduced Density Matrices: Coulson’s Challenge. *Physics Today*, 54(2):58–59, 02 2001.
- [15] Alexander Klyachko. Quantum marginal problem and representations of the symmetric group, 2004.
- [16] N. Linden, S. Popescu, and W. K. Wootters. Almost every pure state of three qubits is completely determined by its two-particle reduced density matrices. *Phys. Rev. Lett.*, 89:207901, Oct 2002.
- [17] Lajos Diósi. Three-party pure quantum states are determined by two two-party reduced states. *Phys. Rev. A*, 70:010302, Jul 2004.
- [18] Nikolai Wyderka, Felix Huber, and Otfried Gühne. Almost all four-particle pure states are determined by their two-body marginals. *Phys. Rev. A*, 96:010102, Jul 2017.
- [19] Albert Aloy, Matteo Fadel, and Jordi Tura. The quantum marginal problem for symmetric states: applications to variational optimization, nonlocality and self-testing. *arXiv: Quantum Physics*, 2020.
- [20] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

- [21] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Rev. Mod. Phys.*, 91:045002, Dec 2019.
- [22] Hsin-Yuan Huang, Richard Kueng, Giacomo Torlai, Victor V. Albert, and John Preskill. Provably efficient machine learning for quantum many-body problems. *Science*, 377(6613):eabk3333, 2022.
- [23] Borja Requena, Gorka Muñoz Gil, Maciej Lewenstein, Vedran Dunjko, and Jordi Tura. Certificates of quantum many-body properties assisted by machine learning. *Phys. Rev. Res.*, 5:013097, Feb 2023.
- [24] LeeAnn M. Sager-Smith and David A. Mazziotti. Reducing the quantum many-electron problem to two electrons with machine learning. *Journal of the American Chemical Society*, 144(41):18959–18966, 2022.
- [25] Luis H. Delgado-Granados, LeeAnn M. Sager-Smith, Kristina Trifonova, and David A. Mazziotti. Machine learning of two-electron reduced density matrices for many-body problems. *The Journal of Physical Chemistry Letters*, 16(9):2231–2237, 2025.
- [26] Xuecheng Shao, Lukas Paetow, Mark E. Tuckerman, and Michele Pavanello. Machine learning electronic structure methods based on the one-electron reduced density matrix. *Nature Communications*, 14(1):6281, 2023.
- [27] Daniel Uzcátegui-Contreras and Dardo Goyeneche. Reconstructing the whole from its parts, 2022.
- [28] Z. Boger and H. Guterman. Knowledge extraction from artificial neural network models. In *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, volume 4, pages 3030–3035 vol.4, 1997.
- [29] Vitor A.C. Horta, Ilaria Tiddi, Suzanne Little, and Alessandra Mileo. Extracting knowledge from deep neural networks through graph analysis. *Future Generation Computer Systems*, 120:109–118, 2021.
- [30] Simon Odense and Artur d’Avila Garcez. Layerwise knowledge extraction from deep convolutional networks, 2020.
- [31] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020.
- [32] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017.
- [33] Paul Skrzypczyk and Daniel Cavalcanti. *Semidefinite Programming in Quantum Information Science*. 2053-2563. IOP Publishing, 2023.
- [34] Haotian Jiang, Tarun Kathuria, Yin Tat Lee, Swati Padmanabhan, and Zhao Song. A faster interior point method for semidefinite programming. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 910–918, 2020.
- [35] Richard Y. Zhang and Javad Lavaei. Sparse semidefinite programs with near-linear time complexity. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 1624–1631, 2018.
- [36] M Guță, J Kahn, R Kueng, and J A Tropp. Fast state tomography with optimal error bounds. *Journal of Physics A: Mathematical and Theoretical*, 53(20):204001, apr 2020.
- [37] Ge Wang, Yi Zhang, Xiaojing Ye, and Xuanqin Mou. *Machine Learning for Tomographic Imaging*. 2053-2563. IOP Publishing, 2019.
- [38] F. Scattarella, D. Diacono, and *et. al.* Deep learning approach for denoising low-snr correlation plenoptic images. *Sci Rep*, 13(19645), 2023.
- [39] S. Suganyadevi, V. Seethalakshmi, and K Balasamy. A review on deep learning in medical image analysis. *Int J Multimed Info Retr*, 11:19–38, 2022.
- [40] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [41] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning, ICML ’08*, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery.
- [42] Lotfallahzadeh Barzili. *Quantum State Estimation and Tracking for Superconducting Processors*

- Using Machine Learning*. Ph.d. thesis, Chapman University, 2021.
- [43] Onur Danaci, Sanjaya Lohani, Brian T Kirby, and Ryan T Glasser. Machine learning pipeline for quantum state estimation with incomplete measurements. *Machine Learning: Science and Technology*, 2(3):035014, may 2021.
 - [44] Keiron O’Shea and Ryan Nash. An introduction to convolutional neural networks, 2015.
 - [45] <https://github.com/guerra-antonio/Learning-QM>.
 - [46] Mohammad Saiful Islam and Andriy Miranskyy. Anomaly detection in cloud components. In *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*, pages 1–3, 2020.
 - [47] Rico Angell and Andrew McCallum. Fast, scalable, warm-start semidefinite programming with spectral bundling and sketching, 2024.
 - [48] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning, 2018.