Patched Image Encoding for Quantum Machine Learning

Peiyong Wang* and Udaya Parampalli
UniMelb CIS

Lloyd C. L. Hollenberg
UniMelb Physics

Casey R. Myers
UniMelb CIS and
UNSW SQC

(Dated: September 13, 2023)

Data encoding is one of the central problems of quantum machine learning. Amplitude encoding, although the most qubit-efficient, faces many problems in the NISQ era. It is also "unphyiscal" to use amplitude encoding when applying quantum machine learning algorithms to image data. In this paper, we propose a novel image encoding method for quantum machine learning algorithms, especially variational quantum algorithms, inspired by the way images are divided into smaller patches in classical deep learning models like the vision transformer[1] and its variants, such as MetaFormer [2], as well as methods like the MLP-Mixer [3]. We also point out the "unphysical" aspects of the data encoding method used in quantum convolutional neural networks. We apply our data encoding method to different quantum machine learning tasks, including semi-supervised contrastive learning for image classification and quantum self-supervised learning. We show that concepts and methods in classical deep learning can be used to inspire new research in quantum machine learning.

I. INTRODUCTION

Something in general on the advancement of quantum machine learning and deep learning, not too long

A. Quantum Machine Learning for Image Processing

In this subsection, review quantum machine learning papers involving image data

- Quantum convolutional neural network [4], and tutorial on QCNN [5]
- Quantum self supervised learning paper [6], in which the dimension reduction method is a classical ResNet.
- Quanvolutional neural network [7].
- The paper used matrix product states to compress images for the classification task; see [8]. In this paper, the authors also divided the images into patches. Their method is based on the flexible representation of quantum images (FRQI), which requires $(\lceil \log_2 N/N_p \rceil + 1)\, N_p$, where N is the number of pixels in an image, and N_p is the number of patches. In the paper they used 1 patch for a 32 by 32 image and 11 qubits, and 64 qubits for 2 by 4 patch case. However, with the decreasing number of patches, they showed it would be harder to reconstruct the image. The goal of patch encoding in their paper is to control the number of qubits in the circuit, not based on "physical" intuitions like us. After the encoding, they still just appended a layer of two-qubit gates for classification. We could argue that our research is based on intuition regarding the nature (and symmetry) of image data.
- The papers regarding FRQI, see [9] and [10]. These two can be put in the data encoding section?

B. Deep Learning for Vision Tasks

In this subsection, review the models and methods in deep learning for vision tasks. First briefly go through the development of convolutional neural networks, just major backbones, not task-specific applications; a lot of references will be needed, ordered by time, including LeNet, AlexNet, VGG, Inception (both VGG and Inception are for very deep convolutional nets), ResNet.

Then we come to the vision transfomer [1], MLP-Mixer [3], pointing out that although transformers process the image as a sequence of patches with self-attention, it firstly uses a linear embedding layer to convert the image patches as tokens. This linear embedding layer is just a multilayer perceptron. In MLP-Mixer, the self-attention layer, which is generally thought to be essential for transformers, could be replaced with an MLP that operates on the same channel of different patches of the image,

[*] peiyongw@student.unimelb.edu.au

acting as a mixer of information. Later, MetaFormer united ViT and MLP-like models in the same framework, where the self-attention mechanism in ViT-like models and the mixer in MLP-Mixer-based models are just different choices of mixing the information carried by different tokens.

We can see a common denominator of all these models, which is the embedding layer that transforms the image patches into tokens. Such an embedding layer is often some linear layer, or a simple MLP. Although MLP performs not as well as CNNs on image data, since it is hard for MLP to extract the spatial information from a flattened image (one could argue that in the vast space of parameters for an MLP, there bound to exist some sets of parameters that enable MLP to function like a CNN, but it could be hard for a generic optimiser to find those parameters without some strong prior knowledge on the regularisation of the parameters, like sparsity). It is also found that MLPs are not translation/scaling invariant, making it difficult to learn even a slightly shifted version of the image.

C. Quantum Data Encoding

In this subsection, briefly introduce the approaches to encode classical data into a quantum state in the qubit/circuit model, including

- Encoding (binary) data as basis states;
- Encoding data as rotation angles of parameterized gates;
- Amplitude encoding, or encoding data as quantum states.

However I don't think there will be much reference for these methods. We may need to find some textbook on quantum machine learning, like [11].

II. METHODS

Star with discussing the "unphysical" aspects of these data encoding methods introduced in the previous section when applied to image data, especially for amplitude encoding, since after encoding an (flattened) image into a statevector, the whole quantum neural network after the data encoding layer is essentially a regularised linear layer, just like those linear layers in multilayer perceptrons. We could also point out that in the QCNN model (if you applied it to image processing with amplitude encoding), the two-qubit "convolution" unitary only acts locally in the qubit sense, not in the statevector sense, which means that when the input image is encoded as the initial statevector of the system, the two-qubit convolution unitary acts in a more "global" manner on the input data, making it hard to capture the symmetry of the image. To make the local quantum operations in quantum neural networks like QCNN truly local, we need to either change the data encoding scheme or carefully design a quantum operator that corresponds to the matrix version of the convolution operator in classical CNNs. Since we are stuck with NISQ devices, it would not be feasible to design a complicated, non-local quantum operator that functions like the classical convolution operator when applied to a quantum statevector (need to show the matrix representation of the convolution operator). This means that we will have to "localise" pixel data in a local region of an image onto local cluster of qubits, or a single qubit. This will enable us to use local unitaries to mimic the function of the classical convolutional operators.

In this section, introduce the methods of the encoding method.

A. Data Re-Uploading

This subsection will give a more detailed review on the data reuploading method[12]. It would be like those in [13, 14]. The emphasis should be on the similarity between the data re-uploading classifier and the multilayer perceptron, since our encoding method aims to process image patches with quantum circuit mimicking the behaviour of an MLP.

B. Patch-wise Encoding

In this subsection, we will introduce how we segment images into local patches and sub-local patches. We'll use Fig. 1 to illustrate how we divide images.

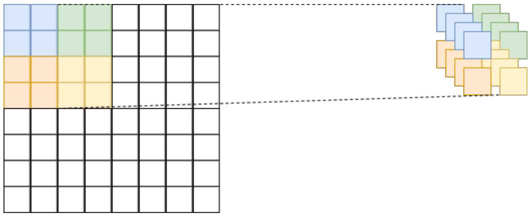

FIG. 1. The figure illustrating how we divide images into patches and how each patch is divided further to incorporate data re-uploading circuits with smaller number of qubits.

We also need to put the data re-uploading circuit here to illustrate the circuit used to encode a 2 by 2 patch and a 4 by 4 patch which is made up with four adjacent 2 by 2 patches.

Even for an 8 by 8 small image, dividing it into 2 by 2 patches still results in 16 patches, which requires 16 qubits if we opt for encoding four pixels into a qubit. We further add another hierarchy of data re-uploading, enabling us to encode four 2-by-2 patches into the state of 4 qubits, see Fig. 3.

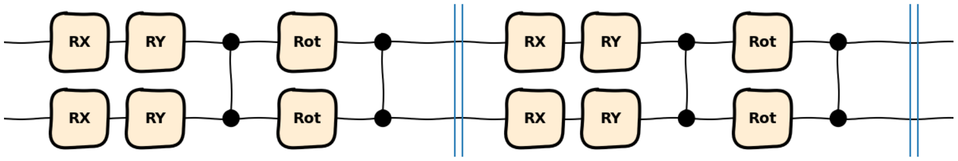

FIG. 2. The two-qubit circuit which can be used to re-upload a two-pixel-by-two-pixel image patch (encoded as rotation parameters of the RX and RY gates). The four Rot gates have different trainable parameters, but the RX and RY gates in the two layers share the same pixel values, hence the re-uploading.

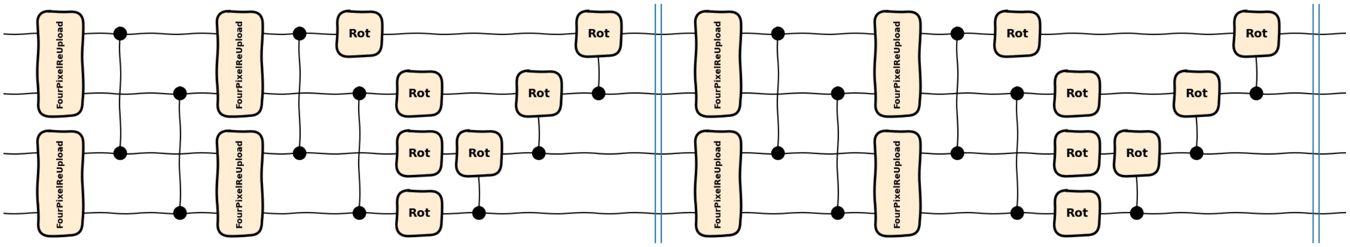

FIG. 3. The four-qubit circuit that encodes 16 pixels (four 2-by-2 patches). In each layer (divided by the doube vertical lines), the four FourPixelReUpload blocks are four quantum circuits as shown in Fig 2, with different pixel data in each block in the same layer, but with the same trainable parameters. The sharing of parameters is inspired by the classical covolutional neural network, in which different regions of the image share the same convolutional kernel. The rotation parameters in the Rot and CRot gates are trainable and different in different layers. We can see that the CRot gates start from the bottom qubit to the top one, meaning that the information on the bottom qubits can be propagated to the top qubits.

C. Image as a Sequence of Patches

After we encode the 16-pixel patch into the state of four qubits, inspired by ViT[1], we will treat the (four) patches in an (8-by-8) image as a sequence, from left to right, top to bottom. At the end of the circuit, the first four qubits will have the "encoded" image information, ready for downstream tasks.

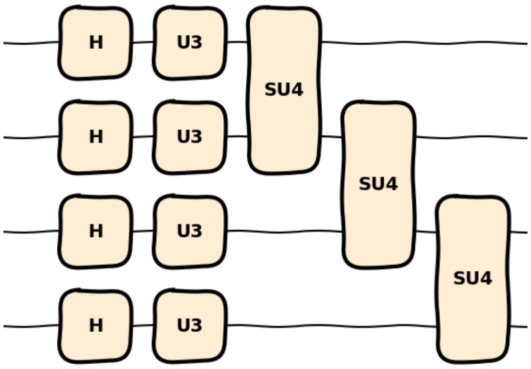

FIG. 5. The memory initialisation circuit InitialiseMemState. This circuit is used to initialise the memory qubits (first four qubits in Fig. 4) to a proper state. All the parameters in the unitaries shown in the figure are trainable.

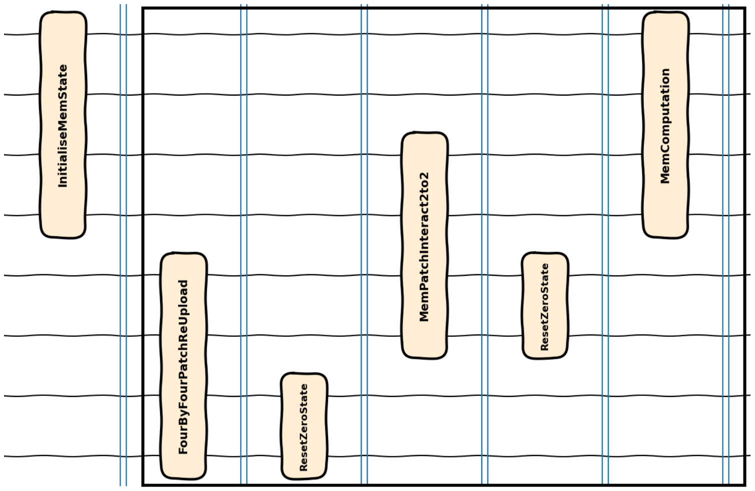

FIG. 4. The full circuit to encode the entire image as a (four-qubit) quantum state. It starts with the $|0\rangle$ state on all eight qubits, then follows by the memory initialisation circuit (see Fig. 5) on the top four qubits, which will be called memory qubits, preparing a proper state for the memory. Then, on the bottom four qubits, we have the `FourByFourPatchReUpload` circuit on the bottom four qubits (patch-encoding qubits), which is the one shown in Fig. 3. Then the bottom two qubits are reset back to $|0\rangle$ state, ready for the next iteration. Then we use the `MemPatchInteract2to2` circuit (shown in Fig. 6) to provide interactions between some of the memory qubits and some of the patch-encoding qubits, allowing information to propagate from the patch-encoding qubits to the memory qubits. Then we will also reset the rest of two qubits in the patch encoding the qubits to the $|0\rangle$ state, ready to accept the encoding circuit for the next patch. Last, there will be some ”in-memory computation” enabled by the `MemComputation` circuit (shown in Fig. ??) on the memory qubits. The circuit in the black box will be repeated N_p times for N_p image patches. The trainable parameters for different image patches are the same.

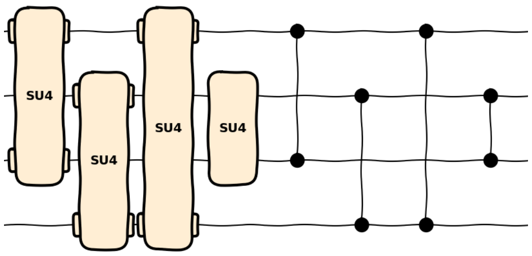

FIG. 6. The quantum circuit `MemPatchInteract2to2` that provides interaction between part of the memory qubits (first two qubits) and part of the patch encoding qubits (bottom two). We can see that there is no interaction within the first two qubits or the bottom two qubits. All the parameters in the unitaries shown in the circuit diagram are trainable.

III. EXPERIMENTS AND RESULTS

In this section, we will talk about the experiments we performed on the data set of tiny handwritten digits. For now, we have done two different kinds of experiments. The first one is the image classification task. The second is the semi-supervised contrastive image classification task (not entirely sure about the name of the task, since the definition of semi-supervised learning is learning with a small amout of labelled data and a large amout of unlabelled data, while we used all the labels, maybe just call it siamese network), where we use the measurement probabilities of the qubits that have encoded image information as input of some distance metric, to minimise the distance of images that share the same label and maximise the distance of images that have different labels. The third one is that we are currently doing, which is a purely self-supervised learning task. However, the SSL framework of the third experiment could subject to change (from BYOL to VICReg).

For the first two experiments, since it was run long ago, with different circuit architectures (but the encoding principle is the same), we may need to re-run them with `Pennylane` and the same circuit architecture mentioned in the previous section.

A. Image Classification

B. Siamese Network

Need to introduce the concept of Siamese network a little bit.

C. Self-Supervised Learning with Quantum Backbone

Need to introduce the SSL framework we are going to use.

IV. DISCUSSION

We first summarise what we have done in this investigation. Then we point out that our method of dealing with image-type data is more intuitive and less relying on classical neural networks, such as the ResNet used in [6].

Since we included self-supervised learning in the experiments, we could also point out, that since only the QNN can actually "see" and interact with the encoded data, there is no need to first encode the classical data completely into a quantum state without any loss. As long as we could get a decent performance for the downstream tasks, there is no need to really care about how the data is encoded. Hence, self-supervised learning can be used as a powerful tool to obtain a backbone quantum neural network which encode the classical data into a quantum state in a lossy way. We further point out that instead of computational complexity, we should seek quantum advantage or quantum usefulness from the quantum representation of classical data, because in the history of deep learning, also it is more time-consuming to use a deep neural network, people still prefer deep models rather than hand-crafted features with faster linear models like the SVM (when the hardware caught up) because deep nets could provide more rich and complex feature/representations of the input data. This shows us that to find some quantum usefulness in deep learning and AI, we should start to seek methods that could provide us with quantum features/representations of the classical data that have properties hard to get from classical features.

ACKNOWLEDGMENTS

The authors acknowledge the support of Pawsey and NCI.

Appendix A: Appendixes

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, in *International Conference on Learning Representations* (2021).

[2] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan, Metaformer is actually what you need for vision, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022) pp. 10809–10819.

[3] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy, MLP-Mixer: An all-MLP architecture for vision, in *Advances in Neural Information Processing Systems*, Vol. 34, edited by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (Curran Associates, Inc., 2021) pp. 24261–24272.

[4] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, Nature Physics **15**, 1273 (2019).

[5] S. Oh, J. Choi, and J. Kim, A tutorial on quantum convolutional neural networks (qcnn) (2020), arXiv:2009.09423 [quant-ph].

[6] B. Jaderberg, L. W. Anderson, W. Xie, S. Albanie, M. Kiffner, and D. Jaksch, Quantum self-supervised learning, Quantum Sci. Technol. **7**, 035005 (2022).

[7] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, Quanvolutional neural networks: powering image recognition with quantum circuits, Quantum Machine Intelligence **2**, 2 (2020).

[8] R. Dilip, Y.-J. Liu, A. Smith, and F. Pollmann, Data compression for quantum machine learning, Phys. Rev. Res. **4**, 043007 (2022).

[9] P. Q. Le, F. Dong, and K. Hirota, A flexible representation of quantum images for polynomial preparation, image compression, and processing operations, Quantum Inf. Process. **10**, 63 (2011).

[10] F. Yan, A. M. Iliyasu, and S. E. Venegas-Andraca, A survey of quantum image representations, Quantum Inf. Process. **15**, 1 (2016).

[11] M. Schuld and F. Petruccione, Representing data on a quantum computer, in *Machine Learning with Quantum Computers*, edited by M. Schuld and F. Petruccione (Springer International Publishing, Cham, 2021) pp. 147–176.

[12] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, enData re-uploading for a universal quantum classifier, Quantum **4**, 226 (2020).

[13] P. Easom-Mccaldin, A. Bouridane, A. Belatreche, and R. Jiang, On depth, robustness and performance using the data Re-Uploading Single-Qubit classifier, IEEE Access **9**, 65127 (2021).

[14] L. Fan and H. Situ, Compact data encoding for data re-uploading quantum classifier, Quantum Inf. Process. **21**, 87 (2022).