scientific reports

OPEN

Neural networks can learn to utilize correlated auxiliary noise

Aida Ahmadzadegan[1,2,3]✉, Petar Simidzija[4], Ming Li[5] & Achim Kempf[1,3,6]

We demonstrate that neural networks that process noisy data can learn to exploit, when available, access to auxiliary noise that is correlated with the noise on the data. In effect, the network learns to use the correlated auxiliary noise as an approximate key to decipher its noisy input data. An example of naturally occurring correlated auxiliary noise is the noise due to decoherence. Our results could, therefore, also be of interest, for example, for machine-learned quantum error correction.

Our first aim is to show that neural networks that learn a task on noisy data, such as, e.g., image classification, can simultaneously also learn to improve their performance by exploiting access to separate noise that is correlated with the noise in the data, when such auxiliary correlated noise is available. In effect, the network learns to improve its performance by implicitly using the auxiliary correlated noise to subtract some of the noise from the data.

This new approach of 'Utilizing Correlated Auxiliary Noise' (UCAN), has potential applications, for example, whenever noise arising in a measurement is correlated with noise that can be picked up in a vicinity of the measurement. The UCAN approach can also be applied in scenarios where the noise is added intentionally, for example, for cryptographic purposes. In the cryptographic case, the UCAN setup is essentially a generalized one-time-pad protocol[1–3] in which the auxiliary noise plays the role of an approximate key that is correlated with the exact key that is represented by the noise on the data. In effect, the network uses the approximate key represented by the auxiliary noise to decipher the noisy data.

The novel UCAN approach is, therefore, not primarily concerned with traditional denoising, see, e.g.,[4–11], but is instead concerned with new opportunities for neural networks that arise in the event of the availability of correlated auxiliary noise. However, we will here not dwell on the range of possible conventional applications from scientific data taking to signal processing and cryptography.

Instead, our aim here is to provide a proof of principle of the UCAN approach on classical neural networks. In the longer term, our motivation is to explore applying the UCAN approach to quantum and quantum-classical hybrid neural networks. There, one application could be to the main bottleneck for quantum computing technology, the process of decoherence. This is because decoherence consists of the generating of correlated auxiliary noise in degrees of freedom in the immediate environment of the physical qubits. The challenge would be to try to access some of those quantum degrees of freedom and to machine-learn, in a UCAN manner, to re-integrate part of the leaked quantum information into the quantum circuit. This could yield a novel form of machine-learned quantum error correction that is not based on traditional quantum error-correction principles such as utilizing redundant coding or topologiocal stability but that instead tries to access environmental degrees of freedom to re-integrate previously leaked quantum information into the circuit.

In the present work, we aim to lay the ground work by demonstrating a proof of principle on classical machines. To this end, we here demonstrate the feasibility of the utilization of correlated auxiliary noise, i.e., of UCAN, through the intuitive example of convolutional neural networks that classify images. In particular, with a view to prospective applications to quantum noise, we here determine the scaling of the efficiency of the UCAN method as either the level of the noise, the dimensionality of the noise or the complexity of the noise are increased. We find that as the magnitude of the noise is increased, the efficiency of the UCAN approach increases. The efficiency becomes optimal in the regime where the magnitude of the noise is close to the threshold where the noise starts to overwhelm the network, i.e., where the performance of the network without UCAN would drop steeply. Further, we find that also as the dimensionality of the function space from which the noise is drawn is increased, the efficiency of the UCAN approach generally increases. Crucially, we also find that as

[1]Perimeter Institute for Theoretical Physics, Waterloo, ON N2L 2Y5, Canada. [2]ForeQast Technologies Limited, Waterloo, ON N2L 5M1, Canada. [3]Department of Applied Mathematics, University of Waterloo, Waterloo, ON N2L 3G1, Canada. [4]Department of Physics and Astronomy, University of British Columbia, Vancouver, BC V6T 1Z4, Canada. [5]Cheriton School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada. [6]Institute for Quantum Computing, University of Waterloo, Waterloo, ON N2L 3G1, Canada. ✉email: ahmadzadegan.aida@gmail.com

the complexity of the noise is increased, the capacity of a neural network to use UCAN can easily be exhausted on classical computers.

As we will discuss in the Outlook, on theoretical grounds this could offer a potential advantage for quantum over classical computers in UCAN-type applications. The advantage could arise from the ability of quantum computers to store, and quickly draw from, extraordinarily complex probability distributions, even when operating only on a relatively small number of qubits. Further, there may be circumstances where a network performing a UCAN-type task needs to possess quantum components because it needs to operate on quantum information. For example, if a network is to be used for the UCAN-type task of machine-learned quantum error correction, i.e., of trying to re-integrate leaked quantum information from the environment into a quantum circuit, the network would need to possess quantum components. For references on quantum computing, communication, cryptography and error correction, see e.g.,[2,3,12,13].

Application of the UCAN approach to CNNs

We begin with a concrete demonstration of UCAN on classical computers. While UCAN should be applicable to most neural network architectures, we here demonstrate the UCAN approach by applying it to image classification by convolutional neural networks (CNNs).

To this end, we choose the standard Fashion-MNIST 28 × 28 pixel grey level image data set and we add around the image, by zero-padding, a rectangular rim of black pixels which we refer to as a 'bezel'. We choose the bezel to be 6 pixels wide so that the number of pixels in the bezel around the image roughly matches the number of pixels in the image itself. We will refer to an image together with its bezel as a 'panel', which has 40 × 40 pixels. First, we add noise only to the image part of the panels. The image classification performance of a CNN trained on these noisy panels correspondingly diminishes. We then examine to what extent the CNN can recover part of the noise-induced drop of its image classification performance when trained and tested with panels that possess noise on the image as well as noise on the bezel that is correlated with the noise on the image.

Concretely, we generate three sets of labeled data. One set, A, consists of the original set of labeled MNIST images, with the black bezel added. The second set of labeled data, B, consists of the same set of labeled images with noise added only to the images. The third set of labeled data, C, consists of the same set of labeled images but with noise added to both the images and their bezels, with the image noise and the bezel noise generated so as to be correlated. For sets A and B, the results are essentially the same if one removes the black bezel as it carries no information. We added this bezel to make the size of the images included in each set A, B, and C, the same and therefore to enable a fair comparison.

We then train CNNs of identical architecture with the three sets of data and compare their image classification performance on the noisy images. We find that after the image classification performance drops from A to B, as expected, it increases again with C. This means that a CNN trained with noisy images with noisy bezel can outperform a CNN with the same architecture but trained on the noisy images with a noiseless bezel. This demonstrates that CNNs can be trained to use access to correlated noise on the bezel to improve their image classification performance by implicitly subtracting some of the noise from the image.

The amount of performance recovery from B to C, as a fraction of the initial performance drop from A to B, may be called the efficiency of the UCAN method in the case at hand. In our experiments we explored how this efficiency depends on the level of the noise as well as on the dimensionality and the complexity of the noise. We will now discuss how we generate these varying types of noise.

Method to generate correlated noise of varying level, dimensionality and complexity.

- **Noise-to-Signal ratio.** We increase the noise level, i.e., the noise-to-signal ratio, by increasing the noise amplitude range relative to the amplitude range of the pixels of the clear image. The brightness values of the clear image are ranging in the interval from zero (black) to one (white). We therefore lift and compress the brightness values of the clear image (and bezel) pixels to a suitable smaller range so that after the noise (whose amplitudes are allowed to take positive and negative values) is added, the brightness values of the noisy image and bezel is ranging again between zero and one.
- **Dimension of the noise space.** In addition to varying the noise-to-signal ratio, we are also varying the dimension of the space from which the noise is drawn. The dimension of the space of panels of size 40 × 40 is 1600. We choose a set of $N < 1600$ basis vectors in that space and we then generate the noise as a linear combination of these noise basis vectors with coefficients drawn from a Gaussian probability distribution. In order to explore the scaling of the efficiency of the UCAN approach when increasing the dimensionality of the vector space from which the noise is drawn, we find that choosing the number, N, of noise basis functions to be either $5^2 = 25$ or $15^2 = 225$ or $22^2 = 484$ suffices to show the trend. (As shown in the Section entitled 'Generating low and high complexity noise', the squares arise when constructing the basis functions as the product of an equal number of Fourier modes in the x and y directions.)
- **Noise complexity.** In order to vary the complexity of the noise, we choose the noise basis vectors such that the pixel pattern that they represent is either of low or high algorithmic complexity, i.e., such that it is either relatively easy or relatively hard to learn for a machine such as a neural network. On the notion of algorithmic complexity, see, e.g.,[14]. In order to generate relatively low complexity noise, we choose as the basis vectors those pixel patterns that correspond to the first $5^2, 15^2$ or 22^2 sine functions of the discrete Fourier sine transform of the full image with bezel. Recall that sine functions are of low algorithmic complexity as they can be generated by a short program. In order to generate relatively high complexity noise, we span the noise vector space using $5^2, 15^2$ or 22^2 basis vectors that correspond to pixel patterns that approximate white noise. Recall that white noise is algorithmically complex. Correspondingly, it should become harder for a CNN to learn

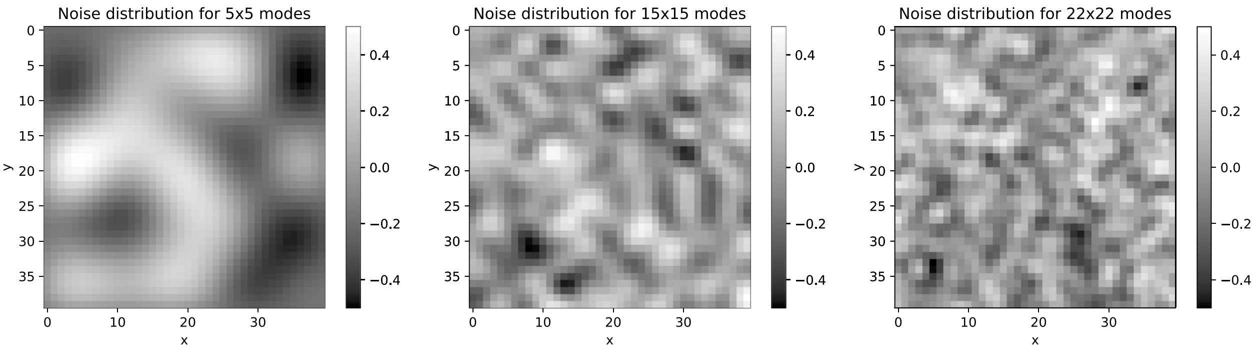

Figure 1. Examples of low complexity noise panels drawn from noise spaces of dimensions (5 × 5), (15 × 15), and (22 × 22) respectively.

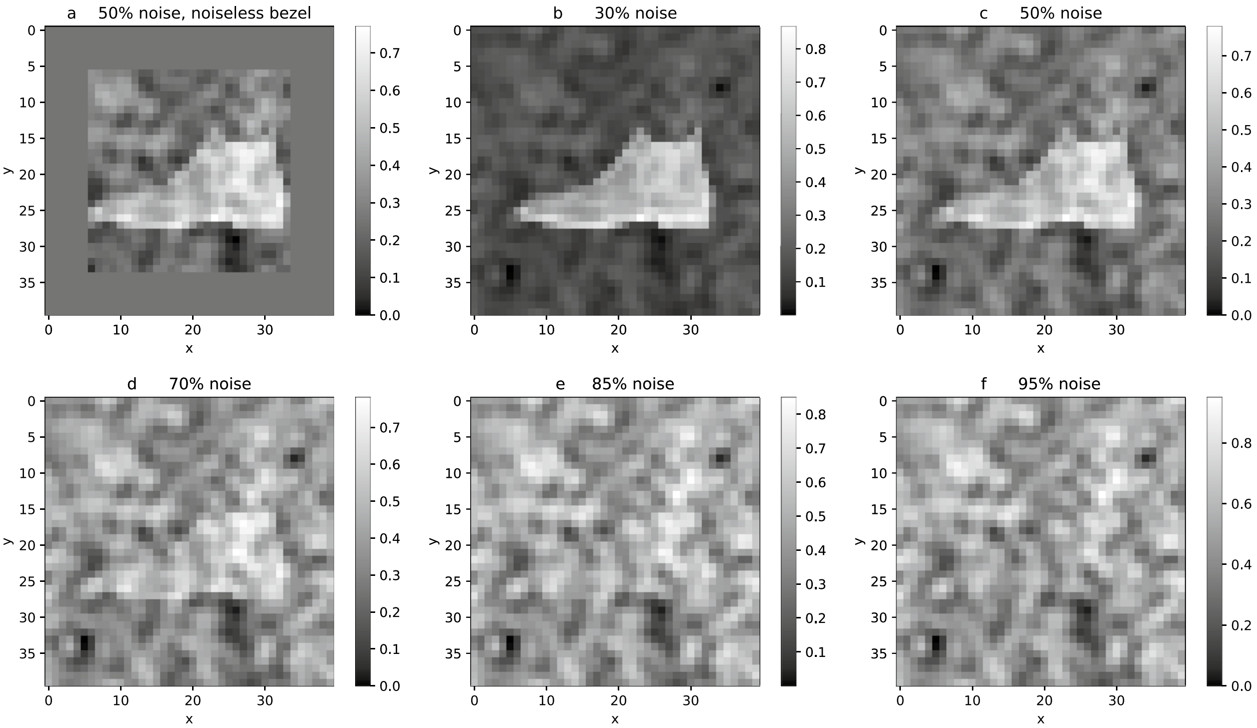

Figure 2. (**a**) Processed image from Fashion-MNIST dataset with added 6 pixel-width initially black bezel. All pixels are rescaled to an interval of [0.25, 0.75] (therefore the grey bezel has pixel values of 0.25). Finally, noise of amplitude interval [−0.5, 0.5] is added to the 28 × 28 image. Panels (**b**)–(**f**) show examples where 30%, 50%, 70%, 85%, and 95% of the panel is noise, respectively.

and utilize the more complex noise. Indeed, as the experimental results discussed in the following section show, the level of noise complexity that we can achieve by the above noise generating method is sufficient to reach the limit of noise complexity that the network architecture which we use in our experiments can accommodate for the purpose of UCAN.

Figure 1 shows examples of panels of relatively low complexity noise drawn from noise spaces of increasing dimension. Figure 2 shows panels of a noisy image and bezel with increasing noise-to-signal ratios. The noise-to-signal ratio is increased until the image is no longer classifiable by human perception. In the experiments, we increase the noise-to-signal ratio until the networks classify no better than chance.

Experiments

Experimental setup. In this section, we detail our implementation of the new UCAN scheme in convolutional neural networks. We use the slightly modified version of CNN architecture given in[15] which contains three convolutional layers and two fully connected layers. Full details regarding our network architecture, training, and evaluation are provided in the Supplementary Information.

In brief, our training data sets are generated from the set of labeled 28 × 28 pixel Fashion-MNIST images[16]. The data set contains 10 different types of fashion items, i.e., if a CNN performs at the level of 10% accuracy then it classifies no better than chance. The data set consists of 10k test images and 60k training images which we divided into a 50k training and a 10k validation set. The Fashion-MNIST images are in grey scale with the pixel values originally ranging between 0 and 255, including both bounding values. We re-scale these values to the interval [0, 1].

To obtain our data sets of type A, we add to the images a black bezel of 6 pixel width by zero-padding. We obtain data sets of type B by adding noise only to the image, as, e.g., in Fig. 2a, and data sets of type C by adding noise to both the image and the bezel, as, e.g., in Fig. 2b–f.

Generating low and high complexity noise. In order to generate noise with relatively low algorithmic complexity, we construct a basis of the noise space by using the orthogonal sine functions of the Fourier sine transform of functions defined on the square $[0, L] \times [0, L]$:

$$b_n(\boldsymbol{x}) = \frac{2}{L} \sin\left(\frac{n_x \pi x}{L}\right) \sin\left(\frac{n_y \pi y}{L}\right) \tag{1}$$

Here, $n = (n_x, n_y)$ is a pair of positive integers that label the choice of basis function. Each basis function $b_n(x)$ yields a 40 × 40 panel, P_n, by evaluating the basis function on the grid of integers: $P_n := [b_n(m_1, m_2)]_{m_1,m_2=1}^{40}$. Each such panel serves as a basis vector in the space of panels from which we draw the noise. In order to avoid needlessly small amplitudes near the boundary (due to the vanishing of all sines there), we choose L slightly larger than 40, at 44. We limit the bandwidth of the noise from above by generating the noise using the first M sine functions in each of the x and y directions. We then generate each noise panel, which we may call r, as a random linear combination of the $N = M^2$ basis panels that are obtained in this way. The pixel values of r are

$$r_{(m_1,m_2)} := \sum_{n_1,n_2=1}^{M} g_{(n_1,n_2)} P_{(n_1,n_2)}(m_1, m_2) \quad \text{where} \quad m_1, m_2 = 1, ..., 40, \tag{2}$$

where we choose the coefficients $g_{(n_1,n_2)}$ from Gaussian probability distributions. We choose the width of the Gaussians to be $\omega(n_1, n_2) := 1/\sqrt{(n_1^2 + n_2^2)}$. This choice lessens the probability of large amplitudes of the coefficients of sines of short wavelength, leading to a pink noise spectrum. We choose these Gaussian distributions since, as discussed in detail in the Supplementary Information, this choice also happens to exactly match the statistics of the quantum vacuum fluctuations of a neutral scalar Klein-Gordon quantum field. Examples of noise panels drawn from noise spaces of different dimensions, $N = M^2$, are shown in Fig. 1. Since we have 60k training and 10k test images, we need 70k such noise panels to add to our total 70k Fashion-MNIST dataset. In order to study the effect of increasing the dimension of the noise space on the performance of the network, we create three such data sets of 70k panels each, with the noise space of dimensions $N = 5^2, 15^2$, and 22^2, respectively.

In order to generate noise with high algorithmic complexity, we proceed exactly as above, except that we use as the basis of the space of noise panels not sine functions but instead panels of fixed approximate white noise. Each of the basis noise panels is generated by drawing for each pixel its grey level from a normal distribution. For later reference, let us note here that the so-obtained basis noise panels are generally not orthogonal, unlike the sine based base noise panels. The 70k noise panels are then generated each as a linear combination of these basis noise panels, with coefficients drawn from a Gaussian probability distribution and truncated so that the grey levels of the noise panel is in the range of $[-0.5, 0.5]$. Analogous to the case of relatively low complexity noise, we generate also the sets of relatively high complexity noise panels by linearly combining, with Gaussian-distributed random coefficients, either $N = 5^2, 15^2$, or 22^2 basis noise panels.

Experimental results. The experimental results, i.e., the performances of our convolutional neural networks as a function of the level, dimensionality and complexity of the noise are shown in Fig. 3. The y-axis indicates the performance of the CNN and the x-axis denotes increasing levels of noise. The left panel, Fig. 3a, shows the performance for noise of relatively low computational complexity, i.e., noise arising as linear combinations of basis noise panels that represent sine functions. The right panel, Fig. 3b, shows the performance for noise of high computational complexity, i.e., for the noise that arises as linear combinations of basis noise panels that each represent approximate white noise. The blue, green, and red curves in Fig. 3a,b represent the choice of $N = 5^2, 15^2$ or 22^2 dimensions for the space of noise panels.

The dashed lines represent the performance of the CNN on the data sets with the noise only on the image while the solid lines represent the performance of the CNN with the noise both on the image and on the bezel. Each data point has been calculated 100 times and the mean value together with its standard deviation in the form of error bars is plotted. The error bars on Fig. 3b are there but they are small, as we will discuss below.

We begin our analysis of the experimental data with the observation that the curves show that, as the level of noise increases, the performance generally drops. In addition, we notice that on the noise-to-signal ratio axis, there are well-defined 'cliffs' where the performance sharply drops to the level of 10% and the network is no longer able to learn to classify better than chance. We also see that the performance drops as the dimensionality of the noise space is increased, i.e., from blue to green to red. As the complexity of the noise is increased, namely from Fig. 3a,b, the performance also drops - except for the red curves, i.e., except if the dimension of the noise space is highest. We will discuss this exception further below.

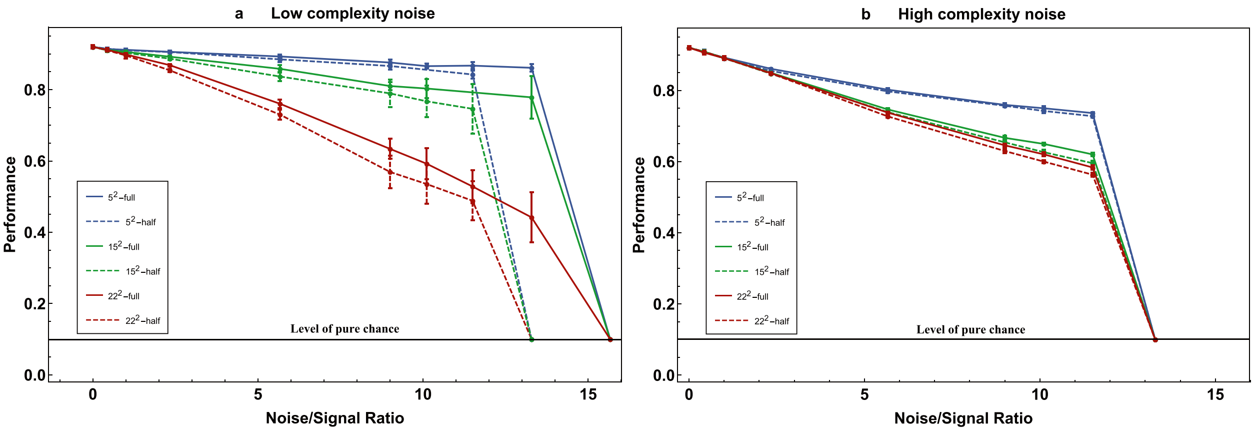

Figure 3. The test accuracy, i.e., the performance of the network on the test dataset, as a function of the noise-to-signal ratio. Notice that, since there are 10 different fashion items, a success rate of 10% indicates that the network classifies at a rate that is equal to pure chance.

The most crucial observation, however, is that all the solid lines are above the dashed lines. This means that the CNNs were able to improve their performance due to UCAN, i.e., when they are given access to correlated noise on the bezel. In particular, we see in Fig. 3a that the efficiency of UCAN, i.e., the gap between the dashed and solid lines of equal color, increases with increasing noise level. Most importantly, we observe that the cliff at which the performance of the network drops sharply is at a higher noise level for the solid lines, with UCAN, than it is for the dashed lines, without UCAN, i.e., without noise on the bezel. Concretely, we observe that there exists a special regime of noise-to-signal levels, here in Fig. 3a around 14. At that level of noise, a CNN without UCAN (dashed lines) cannot learn at all, i.e., its performance drops to 10%, which is the performance level of pure chance. At the same level of noise, however, a CNN of the same architecture but with access to correlated auxiliary noise on the bezel (solid lines) learns to perform considerably well, here with performance levels from about 40% to about 90%, depending on the dimension of the noise space. The upshot is that UCAN possesses its highest efficiency in the regime of such high noise-to-signal ratio, where the network without UCAN starts to fail to learn at all.

It is intuitive that the efficiency of UCAN is best in regimes of high levels of noise. This is because UCAN in effect reduces the network's rate of those misclassifications that are due to noise while, at low noise levels, most misclassifications of a CNN are not primarily due to noise. However, we can only expect the efficiency of UCAN to increase with increasing noise as long as the capacity of the network suffices to learn to utilize the correlations in the noise. Indeed, our experiments showed that the networks struggled to achieve UCAN efficiency in the regime of high noise complexity: in Fig. 3b, the solid lines are barely above the dashed lines. This demonstrates that the UCAN approach can quickly exhaust a classical network's capacity. In the outlook, we will come back to this point in our discussion of the prospect of UCAN on quantum machines, which should possess a much higher capacity to represent complex correlations.

Let us now also discuss why the error bars on Fig. 3b are smaller than those on Fig. 3a. Superficially, the reason is that the performance of the CNNs was more uniform in the case of the high complexity noise on the right. We conjecture the reason to be that the network, when trained on the low complexity noise data, succeeded to learn, to a varying extent, the algorithmically relatively simple long distance correlations between bezel and image noise that are due to the algorithmically relatively simple nature of the sine functions. In contrast, the CNNs appear to have consistently struggled to learn any correlations between the bezel and image noise in the case of relatively high noise complexity.

Finally, let us discuss why the red curves are higher in Fig. 3b than in Fig. 3a. We expect the reason to be that the 22^2-dimensional noise space on the left is spanned by sine functions that are orthogonal while 22^2-dimensional noise space on the right is spanned by 22^2 random white noise panels that are at random angles to another. This means that the noise space is more uniformly sampled for the red curves on the left than for those on the right, which leads to more predictability of the noise on the right and therefore to an advantage for the CNNs on the right. This phenomenon arises only for high-dimensional noise spaces where the directions of random basis vectors start crowding together.

Correlation analysis.

So far, we discussed the efficiency of the UCAN method as a function of the the noise-to-signal-ratio, the noise dimensionality and the algorithmic complexity of the noise. We are now ready to discuss the performance of the UCAN method in terms of the correlations between the noise on the bezel and the noise on the image.

We begin by noting that, since uncorrelated noise is of no use for UCAN, we chose all of the noise in our experiments to be perfectly correlated between the bezel and the image. If the noise on the bezel was known then, in principle, the noise on the image could be perfectly inferred. To see this, let us consider the simple case where the noise space is one dimensional, i.e., where all noise panels are a multiple of just one basis noise panel. In this case, knowing one pixel value anywhere, for example on the bezel, would imply knowing the noise everywhere. More generally, if the noise space is chosen to be N-dimensional, then knowledge of the grey level values of any

N pixels, e.g., N bezel pixels, if the bezel has enough pixels, allows one to infer the noise everywhere, namely by solving a linear system of equations. Since the largest dimension of the noise space that we considered is $N = 22^2 = 484$, while the bezel possesses a larger number of pixels, namely $B = 40^2 - 28^2 = 816$, it is always possible to determine the noise on the image from the noise on the bezel, in principle. However, for a network to infer the image noise from the bezel noise, it would first need to determine the exact noise space. One challenge for the network is that while it is trained with a clear view of the noise on the bezel, its view of the noise on the images is obscured by the presence of the images.

More importantly, some noise spaces are easier for a network to learn than others. For example, if the noise space is one dimensional, then the network needs to learn only one noise basis panel. If this panel is simple, e.g., if the grey level values follow a sine wave, then the panel is easier to learn than when the noise basis panel is of high algorithmic complexity, such as a panel of white noise. The challenge to the network increases as the dimensionality of the noise space is increased. Experimentally, as is clear when comparing Fig. 3a,b, it is indeed easy to overwhelm the network's limited capacity to benefit from UCAN by using basis noise panels of high algorithmic complexity.

Our experiments have been limited, so far, to UCAN applied to CNNs for image classification. It should be very interesting to apply UCAN to other network architectures whenever auxiliary correlated noise is naturally available or can usefully be added, e.g., as the case may be, with RNNs for signal processing, or autoencoders for denoising.

Independently of which suitable neural network architecture the UCAN method is applied to, we are led to conjecture that the efficiency of the UCAN method tends to increase as the amount of noise increases, and that the efficiency of UCAN is highest when the noise reaches the level at which the network without UCAN would start to fail to learn. We are also led to conjecture that when UCAN is applied to any neural network architecture, then even perfect correlations between the noise on the input signal and the auxiliary noise can easily be made sufficiently complex to exhaust the capacity of the network to learn to utilize these correlations.

To support this conjecture, let us discuss to what extent the complexity of the noise could be increased. For example, in the case of the CNNs that we studied here, the noise panels do not need to be generated in the way we did, i.e., by linearly combining noise basis panels with independently distributed coefficients. Instead, in principle, the noise panels could be drawn from any probability distribution over the manifold $[0,1]^{40\times 40}$, i.e., over the 1600-dimensional unit cube. Even if the pixel values are restricted to be 0 or 1, a generic and therefore highly complex probability distribution would require the specification of $2^{1600} \approx 10^{481}$ coefficients. This confirms, in this example, that even if the noise that occurs in practical applications of UCAN is manageable for a suitable network, the complexity of the noise could easily be increased to exceed any network's ability to learn or store or draw from its probability distribution, at least if running on a classical machine.

Outlook

Whenever correlated auxiliary noise is available along with noisy data, or whenever correlated auxiliary noise can be usefully added, the UCAN approach for classical neural networks should be relatively straightforward to implement along the lines presented above.

As we already briefly mentioned, and as we elaborate on in the Supplementary Information, emerging quantum technologies may offer opportunities for further developing the UCAN method. Let us now comment, therefore, on the question of the potential availability of correlated auxiliary quantum noise for applications of UCAN.

In the literature, there are indeed a few examples of uses of auxiliary quantum noise, although so far we know of none that is bringing the power of machine learning to bear, as is our proposal here.

For example, in the quantum energy teleportation (QET) protocol[17,18], an agent invests energy into a local measurement of quantum noise and communicates the outcome to a distant agent who, on the basis of entanglement in the underlying medium or vacuum, uses this information to correspondingly interact with the agent's local quantum noise, enabling that agent to locally extract energy. Quantum energy teleportation has been generalized to aid in algorithmic cooling in quantum processors[19–21]. Also, for example, in quantum optics, see, e.g.,[22,23], the technique of ghost imaging is based on utilizing what is effectively correlated auxiliary classical or quantum noise, see, e.g.,[24,25].

Further, it was shown in[18] that in communication through a quantum field, access to correlated auxiliary quantum noise is always available to the receiver, due to the ubiquitous entanglement in quantum fields and that, in principle, this auxiliary noise is usable to increase the channel capacity. This suggests that classical or quantum implementations of UCAN on quantum machines could be useful, for example, to improve the classical or quantum channel capacity within or between quantum processors or quantum memory. In this case, the quantum noise on the data and the correlated auxiliary quantum noise would arise from the quantum fluctuations of the quantum field that is used for the communication, such as the electromagnetic field, or a quantum field of collective excitations such as the effective phononic field of ion traps[26]. In the context of superconducting qubits, see in particular also, e.g.,[27].

Finally, there is the possibility that UCAN could be used as a method of machine-learned quantum error correction, as we mentioned in the introduction. Indeed, the decoherence of a quantum processor through interaction with its environment consists of creating correlated auxiliary quantum noise in the environment. The experimental challenge then, is to give a quantum neural network, or a quantum-classical hybrid neural network, access to some of the relevant 'environmental' quantum degrees of freedom (which can also be located in the processor itself), as well as access to the quantum processor's noisy quantum output. The computational challenge would be to train the quantum network to undo some of the deleterious effects of the decoherence.

While in our study of UCAN on classical machines above, the decoherence-induced quantum noise in the environment corresponds, of course, to the bezel noise while the noisy output of the quantum processor corresponds to the noisy image, the quantum network's architecture can be very different from that of a CNN. Nevertheless, if we can use our results for classical CNNs of above as guidance, we may speculate that a UCAN approach to machine-learned quantum error correction (as compared to the traditional, scripted approach to quantum error correction that works well at low noise levels) may also work well in the regime of relatively strong noise or strong decoherence. Nevertheless, the quantum neural network will of course itself have to possess suitably low noise levels and it should be very interesting to determine corresponding threshold theorems, see, e.g.,[28].

For recent work on quantum machine learning and quantum neural network architectures, see, e.g.,[29–38]. It should be very interesting to determine which quantum neural network architectures are best suited for quantum UCAN applications, such as quantum machine-learned error correction.

In the near term, of particular interest are potential applications on noisy intermediate-scale quantum (NISQ)[39] devices. These are devices whose number of physical qubits ranges from about 50, which is roughly the number of qubits that classical computers are able to simulate, into the hundreds, which is the number of qubits that is expected to be technologically feasible in the near to medium term. In principle, the application of quantum UCAN on NISQ devices, using gate-model neural networks, will require classical or, preferably, quantum access of the network to the auxiliary noise in the environment of the NISQ device that arises from the partial decoherence of its qubits. At present, experimental setups do not normally provide such access. In the meantime, a proof of principle of quantum UCAN can be pursued by logically tri-partitioning the set of physical qubits in an available NISQ device into one subset that represents the principal quantum processor, a second subset for the gate-model neural network and a third subset that models the environment. Work in this direction is in progress. Of particular relevance in this context are[12,40–52].

Received: 17 September 2020; Accepted: 7 October 2021
Published online: 03 November 2021

References

1. Bellovin, S. M. Frank Miller: Inventor of the one-time pad. *Cryptologia* **35**, 203–222. https://doi.org/10.1080/01611194.2011.583711 (2011).
2. Pirandola, S., Andersen, U. L., Banchi, L., Berta, M. *et al.* Advances in quantum cryptography. arXiv:1906.01645 (2019).
3. Sergienko, A. V. *Quantum Communications and Cryptography* (CRC Press, 2018).
4. Jain, V. & Seung, S. Natural image denoising with convolutional networks. In Koller, D., Schuurmans, D., Bengio, Y. & Bottou, L. (eds.) *Advances in Neural Information Processing Systems 21*, 769–776 (Curran Associates, Inc., 2009).
5. Perez-Cisneros, M., Cocianu, C. & Stan, A. Neural architectures for correlated noise removal in image processing. *Math. Probl. Eng.* **2016**, 6153749 (2016).
6. Krizhevsky, A., Sutskever, I. & Hinton, G. Imagenet classification with deep convolutional neural networks. *Neural Inf. Process. Syst.*, https://doi.org/10.1145/3065386 (2012).
7. Bishop, C. M. *Neural Networks for Pattern Recognition* (Oxford University Press Inc, 1995).
8. Roy, S. S., Ahmed, M. & Akhand, M. A. H. Classification of massive noisy image using auto-encoders and convolutional neural network. In *2017 8th International Conference on Information Technology (ICIT)*, 971–979 (2017).
9. Nazaré, T., De Barros Paranhos da Costa, G., Contato, W. & Ponti, M. *Deep Convolutional Neural Networks and Noisy Images*, 416–424 (2018).
10. Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. **1096–1103**, https://doi.org/10.1145/1390156.1390294 (2008).
11. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. & Manzagol, P. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **11**, 3371–3408 (2010).
12. Gyongyosi, L. & Imre, S. A survey on quantum computing technology. *Comput. Sci. Rev.* **31**, 51–71. https://doi.org/10.1016/j.cosrev.2018.11.002 (2019).
13. Devitt, S. J., Munro, W. J. & Nemoto, K. Quantum error correction for beginners. *Rep. Prog. Phys.* **76**, 076001. https://doi.org/10.1088/0034-4885/76/7/076001 (2013).
14. Li, M. & Vitányi, P. *An introduction to Kolmogorov complexity and its applications*. Texts in Computer Science (Springer International Publishing, 2019), 4th edn.
15. Le, J. Fashion-MNIST. https://github.com/khanhnamle1994/fashion-mnist/commits?author=khanhnamle1994 (2018).
16. Xiao, H., Rasul, K. & Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747 (2017). https://arxiv.org/abs/1708.07747v2.
17. Hotta, M. A protocol for quantum energy distribution. *Phys. Lett. A* **372**, 5671–5676. https://doi.org/10.1016/j.physleta.2008.07.007 (2008).
18. Yamaguchi, K., Ahmadzadegan, A., Simidzija, P., Kempf, A. & Martín-Martínez, E. Super additivity of channel capacity through quantum fields. *Phys. Rev. D* **101**, 105009. https://doi.org/10.1103/PhysRevD.101.105009 (2020).
19. Boykin, P. O., Mor, T., Roychowdhury, V., Vatan, F. & Vrijen, R. Algorithmic cooling and scalable NMR quantum computers. *Proc. Natl. Acad. Sci.* **99**, 3388–3393. https://doi.org/10.1073/pnas.241641898 (2002).
20. Rodríguez-Briones, N. A. *et al.* Heat-bath algorithmic cooling with correlated qubit-environment interactions. *New J. Phys.* **19**, 113047. https://doi.org/10.1088/1367-2630/aa8fe0 (2017).
21. Rodríguez-Briones, N. A., Martín-Martínez, E., Kempf, A. & Laflamme, R. Correlation-enhanced algorithmic cooling. *Phys. Rev. Lett.* **119**, 050502. https://doi.org/10.1103/PhysRevLett.119.050502 (2017).
22. Walls, D. F. & Milburn, G. J. *Quantum Optics* (Springer, 2007).
23. Bachor, H.-A. & Ralph, T. C. *A Guide to Experiments in Quantum Optics* Vol. 1 (Wiley Online Library, 2004).
24. Pittman, T. B., Shih, Y. H., Strekalov, D. V. & Sergienko, A. V. Optical imaging by means of two-photon quantum entanglement. *Phys. Rev. A* **52**, R3429–R3432. https://doi.org/10.1103/PhysRevA.52.R3429 (1995).
25. Bornman, N. *et al.* Ghost imaging using entanglement-swapped photons. *npj Quantum Inf.***5**, 63 (2019).
26. Bruzewicz, C. D., Chiaverini, J., McConnell, R. & Sage, J. M. Trapped-ion quantum computing: Progress and challenges. *Appl. Phys. Rev.* **6**, 021314. https://doi.org/10.1063/1.5088164 (2019).
27. Niu, M. Y. *et al.* Learning non-Markovian quantum noise from moiré-enhanced swap spectroscopy with deep evolutionary algorithm. arXiv:1912.04368 (2019).
28. Knill, E., Laflamme, R. & Zurek, W. H. Resilient quantum computation. *Science* **279**, 342–345 (1998).
29. Benedetti, M., Lloyd, E., Sack, S. & Fiorentini, M. Parameterized quantum circuits as machine learning models. *Quantum Sci. Technol.* **4**, 043001. https://doi.org/10.1088/2058-9565/ab4eb5 (2019).

30. Dunjko, V. & Briegel, H. J. Machine learning & artificial intelligence in the quantum domain: A review of recent progress. *Rep. Prog. Phys.* **81**, 074001. https://doi.org/10.1088/1361-6633/aab406 (2018).
31. Biamonte, J. *et al.* Quantum machine learning. *Nature* **549**, 195–202 (2017).
32. Ciliberto, C. *et al.* Quantum machine learning: A classical perspective. *Proc. Math. Phys. Eng. Sci.***474** (2018).
33. Schuld, M., Sinayskiy, I. & Petruccione, F. The quest for a quantum neural network. *Quantum Inf. Process.* **13**, 2567–2586 (2014).
34. Yang, Z. & Zhang, X. Entanglement-based quantum deep learning. *New J. Phys.* **22**, 033041 (2020).
35. Broughton, M. *et al.* Tensorflow quantum: A software framework for quantum machine learning. arXiv:2003.02989 (2020).
36. Verdon, G., Pye, J. & Broughton, M. A universal training algorithm for quantum deep learning. arXiv:1806.09729 (2018).
37. Verdon, G. *et al.* Quantum graph neural networks. arXiv:1909.12264 (2019).
38. Verdon, G. *et al.* Learning to learn with quantum neural networks via classical neural networks. arXiv:1907.05415 (2019).
39. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum***2**, 79, https://doi.org/10.22331/q-2018-08-06-79 (2018).
40. Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. *arXiv preprint*arXiv:1411.4028 *(2014).*
41. Farhi, E., Goldstone, J., Gutmann, S. & Neven, H. Quantum algorithms for fixed qubit architectures. *arXiv preprint*arXiv:1703.06199 *(2017).*
42. Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors. *arXiv preprint*arXiv:1802.06002 *(2018).*
43. Lloyd, S. Quantum approximate optimization is computationally universal. *arXiv preprint*arXiv:1812.11075 *(2018).*
44. Gyongyosi, L. & Imre, S. Training optimization for gate-model quantum neural networks. *Sci. Rep.* **9**, 1–19 (2019).
45. Farhi, E., Goldstone, J., Gutmann, S. & Zhou, L. The quantum approximate optimization algorithm and the sherrington-kirkpatrick model at infinite size. *arXiv* preprint arXiv:1910.08187 (2019).
46. Gyongyosi, L. & Imre, S. Dense quantum measurement theory. *Sci. Rep.* **9**, 1–18 (2019).
47. Gyongyosi, L. & Imre, S. Quantum circuit design for objective function maximization in gate-model quantum computers. *Quantum Inf. Process.* **18**, 1–33 (2019).
48. Gyongyosi, L. Unsupervised quantum gate control for gate-model quantum computers. *Sci. Rep.* **10**, 1–16 (2020).
49. Gyongyosi, L. & Imre, S. Optimizing high-efficiency quantum memory with quantum machine learning for near-term quantum devices. *Sci. Rep.* **10**, 1–24 (2020).
50. Gyongyosi, L. & Imre, S. Circuit depth reduction for gate-model quantum computers. *Sci. Rep.* **10**, 1–17 (2020).
51. Gyongyosi, L. Quantum state optimization and computational pathway evaluation for gate-model quantum computers. *Sci. Rep.* **10**, 1–12 (2020).
52. Harrigan, M. P. *et al.* Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nat. Phys.* **1–5**, (2021).

Acknowledgements

This research was enabled in part by support provided by Compute Canada (www.computecanada.ca). AK is acknowledging support through the Discovery Program of the National Science and Engineering Research Council of Canada (NSERC), support through the Discovery Project Program of the Australian Research Council (ARC) and through two Google Faculty Research Awards. PS acknowledges support through an NSERC Canada Graduate Scholarship. ML acknowledges support through the Canada Research Chair and Discovery programs of NSERC. AA acknowledges support through an NSERC postdoctoral fellowship.

Author contributions

A.A. and A.K. defined the scientific objectives and conceived the idea. P.S. created the noise panels and A.A conducted the experiments. A.A, A.K., M.L., and P.S. contributed to the interpretation of the results and reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-021-00502-4.

Correspondence and requests for materials should be addressed to A.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.