Exploring Explicit Coarse-Grained Structure in Artificial Neural Networks

Xi-Ci Yang(杨析辞)[1], Z. Y. Xie(谢志远)[2*], and Xiao-Tao Yang(杨晓涛)[1*]

[1]*College of Power and Energy Engineering, Harbin Engineering University, Harbin 150001, China*
[2]*Department of Physics, Renmin University of China, Beijing 100872, China*

(Received 5 November 2022; accepted manuscript online 30 December 2022)

We propose to employ a hierarchical coarse-grained structure in artificial neural networks explicitly to improve the interpretability without degrading performance. The idea has been applied in two situations. One is a neural network called TaylorNet, which aims to approximate the general mapping from input data to output result in terms of Taylor series directly, without resorting to any magic nonlinear activations. The other is a new setup for data distillation, which can perform multi-level abstraction of the input dataset and generate new data that possesses the relevant features of the original dataset and can be used as references for classification. In both the cases, the coarse-grained structure plays an important role in simplifying the network and improving both the interpretability and efficiency. The validity has been demonstrated on MNIST and CIFAR-10 datasets. Further improvement and some open questions related are also discussed.

DOI: 10.1088/0256-307X/40/2/020501

In the past decade, machine learning has drawn great attention from almost all natural science and engineering communities, such as mathematics,[1–3] physics,[4–10] biology,[11–13] and materials sciences,[14–16] and has been widely used in various aspects of modern society, e.g., automatic driving systems, face recognition, fraud detection, expert recommendation system, speech enhancement, and natural language processing, etc. Especially, the deep learning techniques based on the artificial neural networks[17,18] have become the most popular and dominant machine learning approaches progressively, and their interactions with many-body physics have been intensively explored in recent years. On the one hand, some typical neural networks, such as multilayer perceptron,[18] restricted Boltzmann machine,[19] autoencoder,[20] convolutional neural network,[21] and autoregressive network,[22] have been successfully applied to research of quantum magnetization,[23–26] Fermi–Dirac statistics,[27,28] superconductivity,[29,30] statistical averages,[6] and phase transitions[5,31,32] in physical systems. On the other hand, the ideas and techniques developed in physics are introduced into neural networks to improve the performance as well as interpretability.[33,34] This approach may obtain a deeper insight of the neural networks and is sometimes referred to as the physics-inspired machine learning.[35,36] A successful example is the introduction of tensor-network state into deep learning. It stems from quantum information and develops fast in quantum many-body physics, and recently it has been used to realize the supervised learning,[37–39] generative models,[40–42] and network reconstruction,[43–45] etc. Though there are some limitations and difficulties in the current stage, it can still be expected that the interplay between deep learning and many-body physics will continue to flourish in the next decade.

Among the discussions about machine learning and many-body physics, the connection between deep neural networks and the renormalization group (RG) has been extensively studied in the literature recently.[46–52] This relevance probably stems from the essential similarity between the underlying hierarchical structure of the inference process in supervised learning[18] and the generated coarse-grained structure in the RG flow in physical systems,[53] and can be seen more clearly in the context of tensor renormalization group, where the tensor-network structure and the RG-based techniques are combined together to study the many-body physics.[54–58] In fact, it shows that not only the hierarchical structure but also the backpropagation method employed in the training process of neural networks resemble those of the tensor networks very much,[59] and this actually lays the foundation of the increasing interplays between the two fields.

In this work, we propose to explicitly employ the hierarchical coarse-grained structure, as generated in the RG process in tensor networks similarly, in neural networks, and apply it to image classification and data distillation[60–63] for better interpretability in both physics and mathematics. To be specific, in the classification task, we construct a neural network called TaylorNet, which expresses the mapping from the input data to the output label in terms of the Taylor series approximately without using any nonlinear activation functions. The network is simple and can be expressed as a polynomial manifestly in mathematics. This is very different from the ordinary neural networks whose explicit expressions are difficult to obtain, thus unveiling part of the mysteries of neural networks and providing clear direction for further improvement. In the second part, we design a multi-level distillation process which imitates the coarse-graining (CG) operations in the RG flow and displays the underlying hierarchical structure of the inference process explicitly. It shows that the data distilled from lower-level ab-

*Corresponding authors. Email: qingtaoxie@ruc.edu.cn; yangxiaotao@hrbeu.edu.cn
© 2023 Chinese Physical Society and IOP Publishing Ltd

straction contains much more details than those distilled from higher-level abstraction, and the final data distilled from the highest-level layer can be used as good references for image classification directly. The results obtained in both tasks are rather satisfying, as demonstrated in the MNIST [64] and CIFAR-10 [65] datasets.

In this Letter, first we briefly review the coarse-grained structure generated in the RG process in the context of the matrix product operator. Then, we introduce the TaylorNet and the new setup for data distillation, respectively, and demonstrate their validity in MNIST and CIFAR-10 datasets. Finally, we summarize our work and discuss the possible improvement as well as promising extensions briefly.

Coarse-Grained Structure Generated in the RG Process. The RG is one of the most profound tools conceptually of theoretical physics, [66–70] and its impact spans from high-energy to statistical and condensed matter physics. [66,71–73] Essentially, the RG is a conceptual framework comprising various techniques, such as the original block spin approach, [57,67] functional RG, [73] Monte Carlo RG, [71] density matrix RG, [72] and tensor network RG. [54–56,58] Though these schemes differ substantially in details, they share a same essential feature, namely the RG process aims to identify the relevant degrees of freedom (DOFs), integrate out the irrelevant ones iteratively, and eventually arrive at a low-energy effective theory. The extraction of the relevant information is realized by a set of RG transformations, which map the DOFs in a lower scale to those in its neighboring higher scale. A hierarchical coarse-grained structure is essentially generated during this kind of scale transformation. It can be seen more clearly in the real-space RG schemes, [54,55,57,67,72] as exemplified in the following.

To show the coarse-grained structure mentioned above clearly, let us consider the real-space RG transformations of a matrix product operator [74,75] defined on a one-dimensional lattice, which may represent a many-body Hamiltonian of a quantum system or a transfer matrix of a classical statistical model. In this context, through a series of scale transformations, the RG process aims to find a finite-dimensional representation of the operator, which can preserve the low-energy part of the Hamiltonian or the dominant-eigenvalue part of the transfer matrix approximately. For simplicity, let us assume that the MPO is symmetric, and consider the simple case where a binary mapping is performed in each scale transformation. Then the RG process in such a system with length $L = 8$ under open boundary condition can be illustrated in Fig. 1. At the beginning, the operator is expressed in terms of the variables $\{\sigma^{(0)}\}$ sitting on the blue lines. The 1st scale transformation is composed of four isometries denoted as $U^{(1)}$'s, each of which maps $\{\sigma^{(0)}\}$ sitting on two neighboring blue lines to the variables $\{\sigma^{(1)}\}$ sitting on the corresponding green lines. Similarly, the 2nd scale transformation is composed of two isometries $U^{(2)}$'s, and each $U^{(2)}$ maps $\{\sigma^{(1)}\}$ to variables $\{\sigma^{(2)}\}$ sitting on the red lines. $U^{(3)}$ constitutes the last scale transformation, and maps $\{\sigma^{(2)}\}$ to variables $\{\sigma^{(3)}\}$ sitting on the black lines. Eventually, the operator is represented in terms of $\{\sigma^{(3)}\}$, and this completes the full RG process.

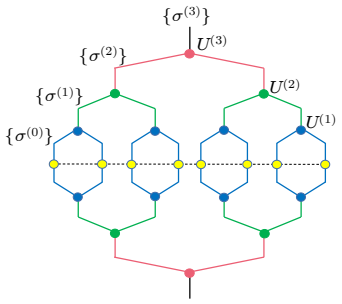

Fig. 1. Hierarchical coarse-grained structure generated in the RG process for a matrix product operator under the open boundary condition. The hollow circles connected by a dashed line and colored by yellow denote the lattice sites where the operator is defined. As described in the main text, the local RG transformations are represented by the rank-3 tensors U's denoted by solid dots, and the DOFs reside on the links between the dots and are denoted as $\{\sigma\}$. For the sake of clarity, the various scales are distinguished by different colors.

In the RG transformations described above, by isometry we mean there are fewer DOFs after the mapping, and the generated variables and DOFs are usually termed as coarse-grained. As clearly sketched in Fig. 1, the whole RG process generates a hierarchical coarse-grained structure with three levels. For a given level, the RG transformations identify the relevant DOFs from lower-level DOFs, and output the identified DOFs to a higher-level transformation for further extraction. This CG operation is an essential gradient of the RG process, and has been heavily employed in the original block spin numerical RG calculations [76–78] and the more recent tensor network RG proposals. [56,79,80]

Without discussing the RG flow in the parameter space and the corresponding fixed-point properties, in this work we just focus on the hierarchical coarse-grained structure described above and emphasize its similarity to the inference process in supervised learning tasks such as image classification. Deep learning falls in the category of representation learning, whose central task is to extract high-level abstract features relevant to the final target from the raw data possessing many irrelevant details and variations, [18] and it solves this problem by constructing higher-level representations out of simpler lower-level representations. More specifically, a representation in a given level is characterized by the output of the previous lower-level neural network layer, and is regarded as the input of a new hidden layer to construct the more abstract higher-level representation. The desired representation is eventually obtained by multistep abstraction, each step of which is realized by a hidden layer and extracts increasingly abstract features from the original input data. This multistep abstraction process is very similar to the RG process discussed before and illustrated in Fig. 1, and also shows an underlying hierarchical coarse-grained structure. This similarity is more evident for the convolutional neural network, where the local structure is emphasized by convolution operations. [21]

In the following, we introduce this hierarchical coarse-grained structure into neural networks manifestly, which makes the conceptual similarity described here more explicit, and the resulting networks are much more easier to understand in both physics and mathematics.

TaylorNet. The quintessential example of a deep learning model is the deep feedforward neural network, and sometimes is referred to as multilayer perceptron model.[18] It represents a mapping $\mathcal{F}$ from the input data to the output result, and generally can be expressed as a composite function of many linear ($\mathcal{L}$) and nonlinear ($\mathcal{N}$) transformations ordered alternately. For example, a feedforward neural network with n layers can be represented as

$$\mathcal{F} = \mathcal{N}_n \mathcal{L}_n \cdots \mathcal{N}_2 \mathcal{L}_2 \mathcal{N}_1 \mathcal{L}_1 \tag{1}$$

where the linear mappings $\mathcal{L}$'s contain many variational parameters that need to be determined, while the nonlinear mappings $\mathcal{N}$'s contain almost no free parameters and are realized by some known operations called activations, such as rectified linear unit, logistic sigmoid, and maxpoolings. The nonlinear mappings $\mathcal{N}$'s are indispensable to approximate a nonlinear $\mathcal{F}$.[18] Apparently, Eq. (1) seems oversimplified, but fortunately, when $\mathcal{F}$ is Borel measurable, the validity is guaranteed by the universal approximation theorem,[81,82] as long as the neural network is sufficiently wide and at least one $\mathcal{N}$ is squashing in some sense. Therefore, Eq. (1) provides a quite general belief to approximate an actual mapping in practical applications of neural networks. However, for a given mapping $\mathcal{F}$, generally there is no clue on either how wide the neural network is or how we can obtain the desired L's. In order to determine the parameters effectively, much effort has been devoted to designing special structures, and this greatly boosted the development of deep neural networks. Successful structures include the convolution operation,[21] shortcut connection in residual network,[83,84] attention structure in the transformer model,[85] etc. Nevertheless, the specific design of structures relies mainly on empirical experience, and there is no theoretical guidance generally, and this is why deep learning is usually regarded as a magic black box.

To reduce the mystery in the structure design, in this work we propose to use another universal expansion, i.e., the multi-variable Taylor formula valid for an arbitrary analytic function f. Expanded at a certain point, the expression can be written as

$$f(X) = f_0 + \sum_{i=1}^{N} a_i^{(1)} x_i + \sum_{i,j=1}^{N} a_{ij}^{(2)} x_i x_j + \sum_{i,j,k=1}^{N} a_{ijk}^{(3)} x_i x_j x_k + \cdots, \tag{2}$$

where X is the input vector with N elements denoted as $\{x_1, x_2, \ldots, x_N\}$, $a^{(n)}$ is the coefficient related to the corresponding n-th order derivative, and f_0 is a collected constant. Equation (2) is also universal, since the nonanalytic functions encountered in our daily life are always expected to have a finite number of singular points and thus can be well approximated by Eq. (2) arguably. Hereafter, just for convenience we simply refer to a and $x_i x_j x_k \ldots$ as the Taylor coefficient and Taylor term, respectively.

The validity of Eq. (2) can be directly verified by an experiment on the classification of MNIST dataset. In the experiment, we regard each image as a vector X, and assume that $f^{(\alpha)}(X^{(i)})$ can be expanded as Eq. (2), where $f^{(\alpha)}(X^{(i)})$ denotes the probability of the i-th image belongs to the α-th category. The whole neural network has only a linear layer that holds the coefficients, and the output can be expressed as

$$f^{(\alpha)}(X^{(i)}) = \sum_{j=1}^{p} W_{\alpha,j} \tilde{X}_j^{(i)}. \tag{3}$$

In the above equation, $\tilde{X}^{(i)}$ is a vector containing p individual Taylor terms corresponding to $X^{(i)}$, which are retained in Eq. (2), and W is a $10 \times p$ weight matrix with element $W_{\alpha,j}$ being the Taylor coefficient corresponding to $\tilde{X}_j^{(i)}$. To make the calculation feasible, we resize the original 28×28 images into 7×7 through the well-established bilinear interpolation technique,[86] perform expansion up to the fourth order, and collect all possible terms in $\tilde{X}$.

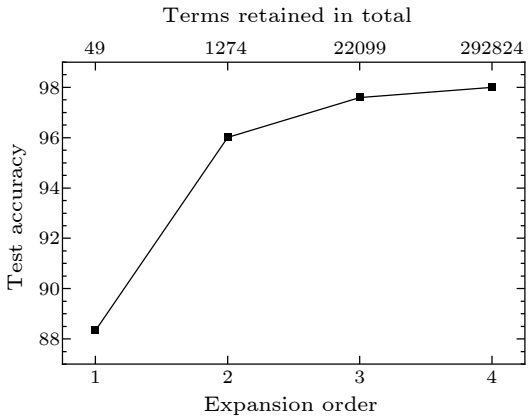

Fig. 2. Test accuracy of the experiment on direct Taylor expansion, as expressed in Eq. (3). The bottom abscissas represent the expansion order kept eventually, and the top abscissas represent the corresponding numbers of Taylor terms. The data is obtained on the MNIST dataset with resized 7×7 images.

The result is shown in Fig. 2. It is clear that the test accuracy can be systematically improved as the expansion order n is increased. When $n = 4$, the number of total terms retained is about 293000, and the obtained accuracy is about 98%. As a comparison, on the same 7×7 MNIST dataset, a residual network with 1.3×10^6 parameters can achieve an accuracy of about 99%. The performance can be further improved by some detailed analysis. Especially, it shows that, though there are about totally 293000 terms retained in the expansion, the contribution of a great number of terms is very small. For example, the distribution of weight corresponding to the quadratic terms is displayed in Fig. 3. For a given term $x_i x_j$, Fig. 3(a) shows that the dominant weight always consumes the least portion no matter how far x_i and x_j are separated, and Fig. 3(b) shows that all the weights have a preferred distribution as a function of the distance between x_i and x_j. This reminds us that there is much redundancy in W, and

thus the number of parameters, N_0, can be greatly reduced by discarding the small weights. In fact, experiments show that the accuracy remains unchanged when N_0 is reduced by half, and drops only by 0.83% when N_0 is reduced to 30%. Even if N_0 is reduced to 10%, we can still obtain an accuracy of about 85%. This actually reflects the spirit of Taylor expansion, since it means that the accuracy can be systematically improved by adding more subtle terms.

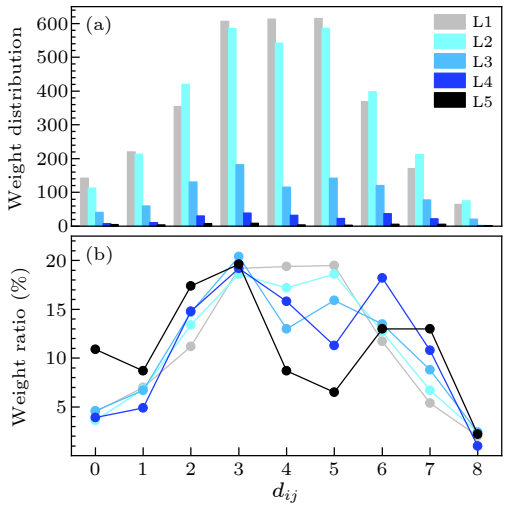

Fig. 3. Distribution of the obtained weight corresponding to the quadratic terms $x_i x_j$, as a function of d_{ij}, the spatial distance between the two pixels x_i and x_j. Weights are ordered by value in ascending order, and equally divided into five groups that are referred to as level-1 (L1) to level-5 (L5), respectively. (a) Weight distribution at each distance d_{ij}. (b) Weight ratio distribution for each level as a function of d_{ij}. The data is obtained on the MNIST dataset with resized 7×7 images. The details of the distribution can be found in Fig. S4 in the Supplemental Material.

To extend Eq. (2) to large scale computer tasks, and to make the above procedure more practical and efficient, we propose the TaylorNet, which realizes Eq. (2) in a multistep manner by utilizing the hierarchical coarse-grained structure described above. Based on the assumption that the dominant parts in the Taylor series mainly correspond to the product of x's in local clusters, as has been partially evidenced in Fig. 3, we introduce a series of intermediate variables x_α with the new index α indicating the hierarchical levels. The variables at a higher level are to be expressed in terms of Taylor series with respect to the variables at the neighboring lower level, and constitute the Taylor expansions of the variables at the neighboring higher level. To be specific, if expanded to the second order, a local cluster indexed as α with m, n-th level variables denoted as $\{x_1^{(n)}, x_2^{(n)}, \ldots, x_m^{(n)}\}$ is mapped to a variable $x_\alpha^{(n+1)}$ at the $(n+1)$-th level, i.e.,

$$x_\alpha^{(n+1)} = c^{(n+1,0)} + \sum_{i=1}^{m} c_i^{(n+1,1)} x_i^{(n)} + \cdots \sum_{i,j=1}^{m} c_{ij}^{(n+1,2)} x_i^{(n)} x_j^{(n)} \tag{4}$$

where $c^{(n,\sigma)}$ denotes the coefficients introduced in the σ-th order terms in the expansion of variables at the n-th level, and $x_i^{(0)}$ is defined as the original input data x_i. Hereafter, Eq. (4) is referred to as a CG operation, and it is illustrated in Fig. 4, in which the usual convolution operation is also illustrated for comparison. Suppose that we are considering the simplest case, i.e., the sizes of the local cluster is 2×2, and there is no overlap between the clusters. In the language of neural networks, this means that the size of both the kernel and the stride is 2×2. In Figs. 4(a) and 4(b), the variables at two neighboring levels are denoted as dots and squares, respectively, and the variables associated with a single local mapping are indicated by the same color. In a convolution operation, a square is a linear combination of four dots, which corresponds to the linear terms in Eq. (4). While in the CG operation, a square is a nonlinear combination of the same four dots, which corresponds to Eq. (4) exactly. To indicate the nonlinear feature, an oval plate is added to distinguish from the convolution, as shown in Fig. 4(b).

The local introduction of the nonlinear mapping has a great advantage over the initial proposals of Taylor expansion, as expressed in Eqs. (2) and (3). On the one hand, the locality puts a strong constraint on the distance among the variables showing up in the Taylor terms retained in the expansion. This greatly reduces the number of parameters that need to be determined, and also removes some unnecessary redundancy, since the contribution from the terms involving variables faraway separated is expected to be small, as partially evidenced in Fig. 3(b). On the other hand, the higher-order terms, as well as the terms involving variables belonging to different local clusters, can emerge naturally in the next several CG operations, which can be seen from Fig. 4(c) explicitly. For example, more complex terms like x_i^3, x_i^4, $x_1 x_3$, $x_1 x_2 x_3$, $x_1 x_2 x_3 x_4$ show up in the expression of z, though y_1 and y_2 are only expanded to the second order locally. The full expression of z can be found in the Supplemental Material. Thus by introducing the nonlinear terms locally, we can generate long-range and very complicated terms in the actual expansions of the variables at the highest level, and there is no need to invoke any magic activations at all.

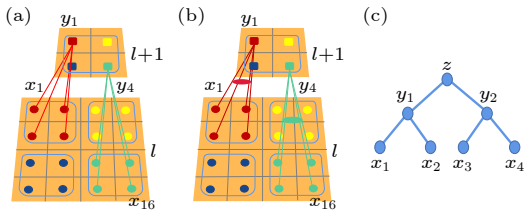

Fig. 4. Illustration of the CG operation in TaylorNet. (a) Convolution operation. (b) CG operation. Clearly, the product of x_1 and x_{16} will emerge in the next CG after the next CG operation. (c) Two successive CG operations on four variables divided into two local clusters. Very complicated terms of x emerge in the expression of z, as shown in Fig. S1 in the Supplemental Material.

In our experiment on MNIST, we use a TaylorNet with four CG layers, each of which maps a 2×2 cluster to a single variable according to second-order Taylor expansion, and then use a linear layer that maps the resulting 2×2 variables to a vector with 10 elements representing the

probabilities. The detailed TaylorNet structure is shown in Fig. 5. On the original 28×28 MNIST dataset, the obtained accuracy is about 99.2%, which is quite satisfactory. On the resized 7×7 MNIST dataset, we can obtain an accuracy of about 97.9% with about 248000 parameters in total, which is much less than the parameters in both the original Taylor expansions (1.46×10^6) and the residual network (1.3×10^6) described above. As to the 32×32 CIFAR-10 dataset, we can obtain an accuracy of about 71.7% with only 1.2×10^6 parameters, and this is also more efficient than the recent MLP-Mixer proposal [87] without pre-training process, which combines the information from the inter-cluster and intra-cluster variables in a similar way. The detailed TaylorNet structure for CIFAR-10 dataset can be found in Fig. S3 in the Supplemental Material. Furthermore, it shows that if we replace the CG operations with the convolution operations in the whole neural network, the accuracy will drop by about 7.3% and 30% immediately as expected, on MNIST and CIFAR-10 datasets, respectively. This clearly demonstrates the power of CG operations in the representations of nonlinear mappings.

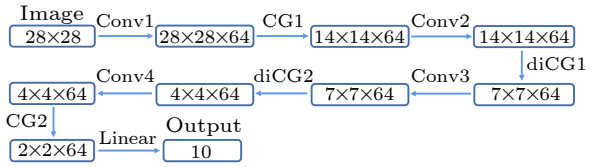

Fig. 5. Sketch of the TaylorNet used in the classification task on MNIST dataset. Hereafter, the numbers in the box denote the representation form of the data, e.g., 28×28×64 denotes 64 feature maps with size 28×28, and the operations sit on the arrows correspond to different neural network layers, e.g., Conv(l_1,l_2,c,s_1,s_2,p) means convolutional layer with kernel size $l_1 \times l_2$, channels c, stride size $s_1 \times s_2$, and padding number p (default 0), similar for CG operation and dilated CG operation as discussed in the main text and Fig. 6. Here, all the four convolutional layers have structure Conv(3,3,64,1,1,1), both CG layers have structure CG(2,2,64,2,2), diCG1 and diCG2 have structures CG(2,2,64,1,1,7) and CG(2,2,64,1,1,3), respectively. The action of a multi-channel convolution operation is illustrated in Fig. S2 in the Supplemental Material, and more details can also be found in Ref. [18].

Similar to the convolution operation, the above CG operation can be performed in slightly different manners. Firstly, the size of the local clusters can be different, and there can be overlaps between different clusters. Moreover, the translation and/or scale invariance of the CG kernels can be employed, that is, the Taylor coefficients in Eq. (4) for CG operations performed at different clusters and/or in different scales can be assumed identical. Secondly, the expansion order in Eq. (4) can be larger than 2, and this depends on the strength of the nonlinearity in the expansion of the actual $\mathcal{F}$.

It is worth mentioning that, in Fig. 5, we have employed two simple ways to further enlarge the effective receptive fields [88] of the CG operations, without changing the size of the local clusters manifestly. One is the introduction of the dilated CG operation, as shown in Fig. 6. In the dilated CG, the variables need to be coarse grained scatter separately in different clusters instead of aggregating locally, which is very similar to the structure of the dilated convolution. [89] The other is performing convolution before the CG operation. This is easy to understand, since the convolution turns each dot in Fig. 4(b) as the linear combination of several dots nearby before the CG operation, and thus each square is effectively expressed in terms of more dots actually. These two techniques may be advantageous in some situations where the inter-cluster product is more important in Eq. (4), and are also considered in Ref. [87] similarly.

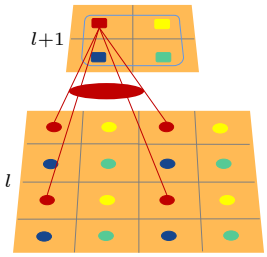

Fig. 6. Sketch of a dilated CG operation, used in Fig. 5. The structure shown in this figure is denoted as CG(2,2,n,1,1,2), which means that the kernel size is 2×2, the number of channels is n, the stride size is 1×1, and the dilation is 2 in both directions.

Data Distillation. The concept of knowledge distillation was originally proposed by Hinton *et al.*, [60] and it aims to train a simpler neural network called student, which is expected to have the same performance approximately to a complex model referred to as teacher. Later, data distillation is proposed to train a smaller dataset from a larger dataset, and expect that the obtained distilled dataset can be used to efficiently train a neural network that has a similar performance to that of a neural network trained from the original larger dataset. [61–63] Though the process of distillation is somewhat complicated, the idea is very simple and reasonable, namely the neural-network-based deep learning is believed to be able to extract some essential features from the original dataset.

In order to make the above idea clearer and to display the inference or abstraction process more explicitly, we propose a new setup of data distillation. Utilizing the hierarchical coarse-grained structure, the new proposal aims to extract the essential features through a multistep process, in which the abstraction is performed progressively from lower levels to higher levels. This is actually the essential spirit of deep learning, [18] as discussed above. It shows that the distilled dataset can be indeed used as references to perform classification task directly.

For concreteness, in the following we describe the distillation strategy applied to the MNIST dataset. The original dataset is composed of ten classes, each of which contains 6000 images, and is denoted as $D^{(0)}(10, 6000)$ hereafter. Firstly we divide each class into 600 groups equally, and select one group from each class to constitute a subset which contains 10 images for each class. Thus totally we obtain 600 subsets, and for simplicity, the i-th subset is denoted as $D_i^{(0)}(10, 10)$ and has 100 images in total. Then perform the usual distillation process on each subset $D_i^{(0)}$

by neural networks, as will be described later, and distill 10 images corresponding to the 10 classes out of each subset. This completes the first level distillation procedure, from which 600 images for each class are distilled, and we denote the distilled dataset as $D^{(1)}(10,600)$ as a whole. Similarly, we further divide $D^{(1)}(10,600)$ into 100 subsets $D_i^{(1)}(10,6)$ on each of which the distillation process is performed and 10 distilled images are obtained, and then we obtain the distilled dataset $D^{(2)}(10,100)$ which contains 100 images for each class at the second level. Repeat this divide-and-conquer strategy, we can obtain dataset $D^{(3)}(10,20)$, $D^{(4)}(10,4)$, and finally reach the highest distilled dataset $D^{(5)}(10,1)$, which contains only a single distilled image for each class and can be regarded as the typical representatives hosting the essential features of that class. The whole process is illustrated in Fig. 7.

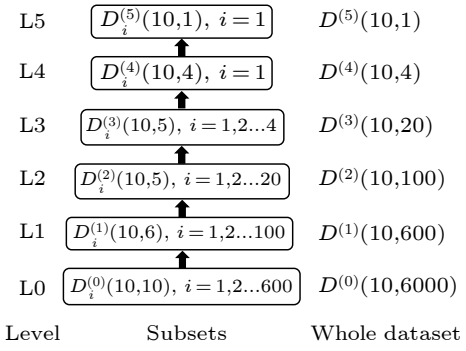

Fig. 7. The data distillation strategy described in the main text, designed for MNIST dataset. The whole distillation process has a five-level structure. The correspondence in the five distillation procedures can be represented as 10-to-1, 6-to-1, 5-to-1, 5-to-1, and 4-to-1 mappings, respectively.

In this work, to perform the distillation procedure on each subset $D_i^{(\alpha)}(10,m)$, we employ the distribution matching method and a similar neural network architecture proposed in Refs. [62,63], as is illustrated in Fig. 8 in detail. The goal of the procedure is to determine n (distilled) images, denoted as Y's, which minimize a lost function defined in the following

$$L=\sum_{\alpha=1}^{n}\left(\lambda d_{\alpha,\alpha}-\sum_{\beta\neq\alpha}^{n} d_{\alpha,\beta}\right),$$
$$d_{\alpha,\beta}\equiv\sum_{i=1}^{m_\beta}\left|f(Y_\alpha)-f(X_{\beta,i})\right|^2, \tag{5}$$

where Y_α denotes the desired image for the α-th class, $X_{\alpha,i}$ denotes the i-th image in the α-th class of the dataset, f denotes the nonlinear mapping represented by the neural network which produces the embedding vector for any given image, as illustrated in Fig. 8. In Eq. (5), n is the number of classes in the dataset, m_α is the number of images belonging to the α-th class, and λ is a hyperparameter to balance the two terms in the bracket, which is set to be 19 in our calculations. In essence, $d_{\alpha,\beta}$ measures the Euclidean distance between the reference Y_α and the images belonging to the β-th class of the original dataset, in the space where the embedding vectors are defined. Therefore, physically the lost function means that each desired reference is required to resemble the images in the same class to the greatest extent, and meanwhile differ from the images in the other classes as much as possible.

The distilled images at different levels are sketched in Fig. 9(a). As expected, it seems that the obtained images obtained from lower-level distillations contain more details and are clearer. When the level of distillation goes up, the details gradually blur and only some indescribable features remain. This tendency is more evident for CIFAR-10 dataset, as is shown in Fig. 9(b). In Figs. S5 and S6 in the Supplemental Material, we also provide direct numerical evidence showing that the similarities shared by the images in the same class become more and more significant when the distillation level increases.

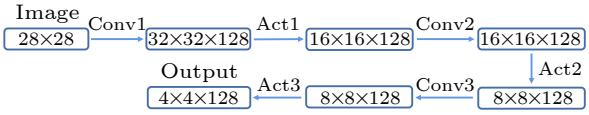

Fig. 8. The neural network structure used in the distillation procedure on each subset of MNIST. Conv1 has structure Conv(3,3,128,1,1,3), both Conv2 and Conv3 have structure Conv(3,3,128,1,1,1). Act1, Act2, and Act3 are three activation layers, each of which is composed of instance normalization, ReLU, and avgpooling. The avgpooling is performed with kernel size 2×2 and stride 2×2. Each image is mapped to an embedded space, and the result is a vector with length 16 and channels 128.

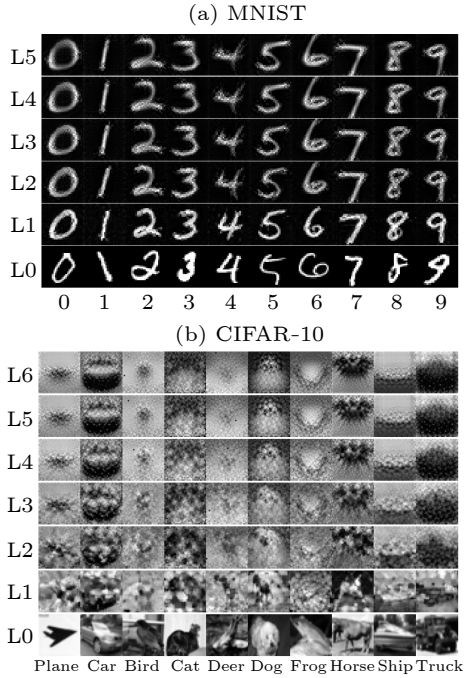

Fig. 9. Distilled figures at each abstraction level. For comparison, one sample of each class in the original dataset is shown in $L0$.

It is reasonable to regard the remaining features in the final distilled images as essential ones which characterize, or even define the dataset in the perfect case. To check this, firstly we train two residual networks, i.e., ResNet18 and ResNet50 on MNIST and CIFAR-10 datasets, respectively, through the usual classification task, and then

use the trained models to collect the output embedding vectors[18] of both the test and distilled images. Without resorting to neural networks further, the final classification is performed by directly comparing the similarity between the embedding vectors of a test image and that of the distilled images, according to the angles in between, and the image is classified into a category whose distilled reference image has the highest similarity to it. It shows that this direct comparison can already give test accuracies of about 98.7% and 86.9% for MNIST and CIFAR-10, respectively. This confirms that the above distillation process can indeed capture some essential features of the original dataset, from lower-level abstraction to higher-level abstraction gradually, and this reflects exactly the spirit of deep learning.

The performance can be further improved in several ways. Firstly, in each distillation procedure, the lost function plays an important role and can be designed more smartly. In this work, the distance between two images is defined as the Euclidean distance in the embedded space, and one can use other measures, such as the Arcface loss,[90] which emphasizes the angular separation and is frequently used in facial-recognition tasks. A preliminary experiment utilizing this lost function can produce an accuracy of about 99% for MNIST, and 89% for CIFAR-10. Secondly, the partition of the dataset, as well as the choice of the hyperparameter λ, may affect the result of the whole distillation. An optimal choice should consider the balance between performance and efficiency in a better way, while in this work, we just adopt the most convenient choice.

In summary, inspired by the similarity between the RG flow in physical systems and the inference process in deep neural networks, we introduce the hierarchical coarse-grained structure into the artificial neural networks manifestly to improve the interpretability without degrading performance. To be specific, in the first part we propose the TaylorNet by introducing the CG operation locally and hierarchically, which extends the linear convolution operation to nonlinear polynomial combinations. It approximates the mapping from the input signal to output result by Taylor expansions effectively, without resorting to the activation functions, and achieves satisfying results in the classification experiments on MNIST and CIFAR-10 datasets. In the distillation task, we propose a setup with a hierarchical coarse-grained structure, and make the inference process from lower levels to higher levels more transparent. It seems that the multistep distillation process is able to capture some essential features of the original dataset, and the distilled images possess less irrelevant details and can be used as reference images in classification tasks. In both the cases, the resulting processes represented by the neural networks are more understandable, and the performance is very acceptable compared to the conventional neural networks.

Besides the specific issues discussed separately, there are some other aspects that can be explored to further improve the performance. For example, we can use more than one TaylorNets to approximate a single mapping, and put the orthogonality constraint on these networks appropriately for higher efficiency. This may be achieved by adding penalties in the lost functions, training in momentum space, or using other orthogonal complete sets like spherical functions. All these topics are interesting and have been discussed in physical systems and they are far beyond the scope of this paper, we would like to leave them as pursuits in the near future.

As to the TaylorNet, it is also worth mentioning that our proposal is very different from the previous work in the literature,[91–97] which have also explored the Taylor series in neural networks. Most of them have specific motivations and work in different frameworks, and there is no explicit hierarchical structure employed there. For example, Chen *et al.*[91] used a single-layer neural network similar to Eq. (3) to approximate the Taylor expansion of a single-variable function. Montavon *et al.*[92] explored the role of Taylor coefficients as derivatives in an ordinary neural network to analyze the importance of a single pixel in the classification task. Tong *et al.*[93] used the Taylor series to approximate the quadratic form of a Hermitian matrix. Rao *et al.*[94] expressed part of the nonlinearity in terms of a direct-product operation and applied it to the study of partial differential equations. Rendle *et al.*[95] and Meng *et al.* [96] proposed similar models to TaylorNet but with quite different emphases and motivations. The closest one to our work is probably Ref. [97], where Novikov *et al.* used the neural network to approximate the Taylor expansions. However, in their work both the Taylor terms and Taylor coefficients were represented as compact matrix product operators approximately. Thus there is no coarse-grained structure emphasized in our work at all.

Finally, though the fixed point is not discussed at all in this work, the introduction of the hierarchical coarse-grained structure does provide the possibility of its existence. It is possible to study the fixed point of the scale transformations introduced in the TaylorNet, as well as the fixed point of the iterative distillation procedure introduced above. Whether the scale invariance can be related to some interesting critical phenomena in this context, as explored in Ref. [46], is an open question deserving attention.

Acknowledgement. We thank Jing Zhang for her contribution in the early stage of this work, and thank Tao Xiang and Ze-Feng Gao for helpful discussions. This work was supported by the National R&D Program of China (Grant Nos. 2017YFA0302900 and 2016YFA0300503), the National Natural Science Foundation of China (Grant Nos. 52176064, 12274458, and 11774420), and the Research Funds of Renmin University of China (Grant No. 20XNLG19).

References

[1] Lei N, Luo Z, Yau S T, and Gu D X 2018 arXiv:1805.10451 [cs.LG]

[2] Lei N, Su K, Yau S T, and Gu D X 2019 *Comput. Aided Geom. Des.* **68** 1

[3] Fawzi A, Balog M, Huang A, Hubert T, Paredes B R, Barekatain M, Novikov A, Ruiz F J R, Schrittwieser J, Swirszcz G, Silver D, Hassabis D, and Kohli P 2022 *Nature* **610** 47

[4] Gao X and Duan L M 2017 *Nat. Commun.* **8** 662

[5] Carrasquilla J and Melko R G 2017 *Nat. Phys.* **13** 431

[6] Wu D, Wang L, and Zhang P 2019 *Phys. Rev. Lett.* **122** 080602

[7] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L, and Zdeborová L 2019 *Rev. Mod. Phys.* **91** 045002

[8] Bedolla E, Padierna L C, and Priego R C 2021 *J. Phys.: Condens. Matter* **33** 053001

[9] Font B, Weymouth G, Nguyen V T, and Tutty O R 2021 *J. Comput. Phys.* **434** 110199

[10] Di Sante D, Medvidovic M, Toschi A, Sangiovanni G, Franchini C, Sengupta A M, and Millis A J 2022 *Phys. Rev. Lett.* **129** 136402

[11] Helmstaedter M, Briggman K L, Turaga S C, Jain V, Seung H S, and DenkHelmstaedter W 2013 *Nature* **500** 168

[12] Webb S 2018 *Nature* **554** 555

[13] Dauparas J, Anishchenko I, Bennett N, Ragotte H B R J, Milles L F, Wicky B I M, Courbet A, deHaas R J, Bethel N, Leung P J Y, Huddy T F, Pellock S, Tischer D, Chan F, Koepnick B, Nguyen H, Kang A, Sankaran B, Bera A K, King N P, and Baker D 2022 *Science* **378** 49

[14] Stanev V, Oses C, Kusne A G, Rodriguez E, Paglione J, Curtarolo S, and Takeuchi I 2018 *npj Comput. Mater.* **4** 29

[15] Wang A Y T, Murdock R J, Kauwe S K, Oliynyk A O, Gurlo A, Brgoch J, Persson K A, and Sparks T D 2020 *Chem. Mater.* **32** 4954

[16] Batra R, Song L, and Ramprasad R 2021 *Nat. Rev. Mater.* **6** 655

[17] LeCun Y, Bengio Y, and Hinton G 2015 *Nature* **521** 436

[18] Goodfellow I, Bengio Y, and Courville A 2016 *Deep Learning* (Cambridge: MIT Press)

[19] Smolensky P 1986 *Information Processing in Dynamical Systems: Foundations of Harmony Theory* (Cambridge: MIT Press)

[20] Bourlard H and Kamp Y 1988 *Biol. Cybern.* **59** 291

[21] Lecun Y, Bottou L, Bengio Y, and Haffner P 1998 *Proc. IEEE* **86** 2278

[22] Bengio Y and Bengio S 2000 *Advances in Neural Information Processing Systems* **12** 400

[23] Carleo G and Troyer M 2017 *Science* **355** 602

[24] Cai Z and Liu J 2018 *Phys. Rev. B* **97** 035116

[25] Liang X, Liu W Y, Lin P Z, Guo G C, Zhang Y S, He L X 2018 *Phys. Rev. B* **98** 104426

[26] Nomura Y and Imada M 2021 *Phys. Rev. X* **11** 031034

[27] Pfau D, Spencer J S, Matthews A G D G, and Foulkes W M C 2020 *Phys. Rev. Res.* **2** 033429

[28] Hermann J, Schatzle Z, and Noe F 2020 *Nat. Chem.* **12** 891

[29] Zeng S M, Zhao Y C, Li G, Wang R R, Wang X M, and Ni J 2019 *npj Comput. Mater.* **5** 84

[30] Konno T, Kurokawa H, Nabeshima F, Sakishita Y, Ogawa R, Hosako I, and Maeda A 2021 *Phys. Rev. B* **103** 014509

[31] van Nieuwenburg E P L, Liu Y H, and Huber S D 2017 *Nat. Phys.* **13** 435

[32] Wetzel S J 2017 *Phys. Rev. E* **96** 022140

[33] Valiant L G 1984 *Commun. ACM* **27** 1134

[34] Bahri Y, Kadmon J, Pennington J, Schoenholz S S, Dickstein J S, and Ganguli S 2020 *Annu. Rev. Condens. Matter Phys.* **11** 501

[35] Hibat-Allah M, Inack E M, Wiersema R, Melko R G, and Carrasquilla J 2021 *Nat. Mach. Intell.* **3** 952

[36] Karniadakis G E, Kevrekidis I G, Lu L, Perdikaris P, Wang S, and Yang L 2021 *Nat. Rev. Phys.* **3** 422

[37] Stoudenmire E and Schwab D J 2016 arXiv:1605.05775 [stat.ML]

[38] Glasser I, Pancotti N, and Cirac J I 2020 *IEEE Access* **8** 68619

[39] Cheng S, Wang L, and Zhang P 2021 *Phys. Rev. B* **103** 125117

[40] Han Z Y, Wang J, Fan H, Wang L, and Zhang P 2018 *Phys. Rev. X* **8** 031012

[41] Cheng S, Wang L, Xiang T, and Zhang P 2019 *Phys. Rev. B* **99** 155131

[42] Vieijra T, Vanderstraeten L, and Verstraete F 2022 arXiv:2202.08177 [quant-ph]

[43] Liu D, Ran S J, Wittek P, Peng C, Garcia R B, Su G, and Lewenstein M 2019 *New J. Phys.* **21** 073059

[44] Gao Z F, Cheng S, He R Q, Xie Z Y, Zhao H H, Lu Z Y, and Xiang T 2020 *Phys. Rev. Res.* **2** 023300

[45] Zunkovic B 2022 *Quantum Mach. Intell.* **4** 21

[46] Saremi S and Sejnowski T J 2013 *Proc. Natl. Acad. Sci. USA* **110** 3071

[47] Beny C 2013 arXiv:1301.3124 [quant-ph]

[48] Lin H W, Tegmark M, and Rolnick D 2017 *J. Stat. Phys.* **168** 1223

[49] Mehta P and Schwab D J 2014 arXiv:1410.3831 [stat.ML] Schwab D J and Mehta P 2016 arXiv:1609.03541 [cond-mat.dis-nn]

[50] Janusz M K and Ringel Z 2018 *Nat. Phys.* **14** 578

[51] Li S H and Wang L 2018 *Phys. Rev. Lett.* **121** 260601

[52] De Koch L M, De Koch R M, and Cheng L 2020 *IEEE Access* **8** 106487

[53] Cardy J 1996 *Scaling and Renormalization in Statistical Physics* (Cambridge: Cambridge University Press)

Kardar M 2007 *Statistical Physics of Fields* (Cambridge: Cambridge University Press)

[54] Levin M and Nave C P 2007 *Phys. Rev. Lett.* **99** 120601

[55] Xie Z Y, Jiang H C, Chen Q N, Weng Z Y, and Xiang T 2009 *Phys. Rev. Lett.* **103** 160601

Zhao H H, Xie Z Y, Chen Q N, Wei Z C, Cai J W, and Xiang T 2010 *Phys. Rev. B* **81** 174411

[56] Xie Z Y, Chen J, Qin M P, Zhu J W, Yang L P, and Xiang T 2012 *Phys. Rev. B* **86** 045139

[57] Efrati E, Wang Z, Kolan A, Kadanoff L P 2014 *Rev. Mod. Phys.* **86** 647

[58] Meurice Y, Sakai R, and Yockey J U 2022 *Rev. Mod. Phys.* **94** 025005

[59] Chen B B, Gao Y, Guo Y B, Liu Y Z, Zhao H H, Liao H J, Wang L, Xiang T, Li W, and Xie Z Y 2020 *Phys. Rev. B* **101** 220409(R)

[60] Hinton G, Vinyals O, and Dean J 2015 arXiv:1503.02531 [stat.ML]

[61] Wang T, Zhu J Y, Torralba A, and Efros A A 2020 arXiv:1811.10959v3 [cs.LG]

[62] Zhao B, Mopuri K R, and Bilen H 2021 arXiv:2006.05929 [cs.CV]

[63] Zhao B and Bilen H 2022 arXiv:2110.04181v3 [cs.LG]

[64] The official website of MNIST is available at http://yann.lecun.com/exdb/mnist

[65] The official website of CIFAR-10 is available at https://www.cs.toronto.edu/ kriz/cifar.html

[66] Gell-Mann M and Low F E 1954 *Phys. Rev.* **95** 1300

[67] Kadanoff L P 1966 *Physics Physique Fizika* **2** 263

[68] Wilson K G 1971 *Phys. Rev. B* **4** 3174

Wilson K G 1971 Phys. Rev. B **4** 3184

[69] Wilson K G 1975 *Rev. Mod. Phys.* **47** 773

[70] Kadanoff L P 1975 *Phys. Rev. Lett.* **34** 1005

[71] Swendsen R H 1979 *Phys. Rev. Lett.* **42** 859

[72] White S R 1992 *Phys. Rev. Lett.* **69** 2863

[73] Wetterich C 1993 *Phys. Lett. B* **301** 90

[74] Verstraete F, García-Ripoll J J, and Cirac J I 2004 *Phys. Rev. Lett.* **93** 207204

[75] Pirvu B, Murg V, Cirac J I, and Verstraete F 2010 *New J. Phys.* **12** 025012

[76] Bray J W and Chui S T 1979 *Phys. Rev. B* **19** 4876

[77] Pan C Y and Chen X Y 1987 *Phys. Rev. B* **36** 8600

[78] Kovarik M D 1990 *Phys. Rev. B* **41** 6889

[79] Yang L P, Liu Y Z, Zou H Y, Xie Z Y, and Meurice Y 2016 *Phys. Rev. E* **93** 012138

[80] Chen B B, Chen L, Chen Z, Li W, and Weichselbaum A 2018 *Phys. Rev. X* **8** 031082

[81] Hornik K, Stinchcombe M, and White H 1989 *Neural Netw.* **2** 359

[82] Cybenko G 1989 *Math. Control Signals Systems* **2** 303

[83] He K M, Zhang X Y, Ren S Q, and Sun J 2016 *Proc. IEEE Conference Computer Vision Pattern Recognition* (Las Vegas, USA, 26 June–1 July 2016) pp 770–778

[84] Huang G, Liu Z, van der Maaten L, and Weinberge K Q 2017 *Proc. IEEE Conference Computer Vision Pattern Recognition* (Hawaii USA, 21–26 July 2017) pp 4700–4708

[85] Vaswani A, Shazeer N, Parmer N, Uszkoreit J, Jones L, Gomez A N, Kaiser L, and Polosukhin I 2017 arXiv:1706.03762 [cs.CL]

[86] Press W H, Teukolsky S A, Vetterling W T, and Flannery B P 2007 *Numerical Recipes: The Art of Scientific Computing* (Cambridge: Cambridge University Press)

[87] Tolstikhin I, Houlsby N, Kolesnikov A, Beyer L, Zhai X, Unterthiner T, Yung J, Steiner A, Keysers D, Uszkoreit J, Lucic M, and Dosovitskiy A 2021 arXiv:2105.01601 [cs.CV]

[88] Luo W, Li Y, Urtasun R, and Zemel R 2016 *Advances in Neural Information Processing Systems* (Barcelona, Spain, 5–8 December 2016)

[89] Wang P, Chen P, Yuan Y, Liu D, Huang Z, Hou X, and Cottrell G 2018 *IEEE Winter Conference on Applications of Computer Vision* (Nevada, USA, 12–15 March 2018) pp 1451–1460

[90] Deng J, Guo J, Xue N, and Zafeiriou S 2019 *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, USA, 16–20 June 2019) pp 4690–4699

[91] Chen X, Ma Q, and Alkharobi T 2009 *IEEE International Conference on Computer Science and Information Technology* (Beijing, China 8–11 August 2009) pp 291–294

[92] Montavon G, Lapuschkin S, Binder A, Samek W, and Muller K R 2017 *Pattern Recognit.* **65** 211

[93] Tong Y, Xiong S, He X, Pan G, and Zhu B 2021 *J. Comput. Phys.* **437** 110325

[94] Rao C, Sun H, and Liu Y 2021 arXiv:2106.04781 [cs.LG]

[95] Rendle S 2010 *IEEE International Conference on Data Mining* (Sydney, Australia, 13–17 December 2010) pp 995–1000

[96] Meng Y M, Zhang J, Zhang P, Gao C, and Ran S J 2021 arXiv:2012.11841v2 [cs.LG]

[97] Novikov A, Trofimov M, and Oseledets I 2017 arXiv:1605.03795v3 [stat.ML]