**Supplementary**

Table S 1: The definition of the symbols used in the current work.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Symbol** | **Component** | **Definition** | **Symbol** | **Component** | **Definition** |
|  |  | Cycle number in the dataset, |  |  | Human-picked features in cycle *m* |
|  |  | Number of shifted cycles from used as the first cycle of input to the DNNs. |  |  | Charge capacity in cycle *m* |
|  |  | Cycle number in the input dataset to the DNNs, where . |  |  | Discharge capacity in cycle *m* |
|  |  | Total number of cycles used as input to the DNNs, i.e. . |  |  | Average temperature in cycle *m* |
| *t* |  | Time step |  |  | Temperature minimum in cycle *m* |
|  |  | End-of-life |  |  | Temperature maximum in cycle *m* |
|  |  | Residual useful life |  |  | Total charge time in cycle *m* |
|  |  | Discharge power of cycle number *n* |  |  |  |
|  |  |  |  |  |  |
|  |  | Continuous-*t* data in discharge half-cycle *m* |  |  | Data-driven features in discharge half-cycle *m* |
|  |  | Capacity curve in discharge half-cycle *m* |  | / | Data-driven feature for end-of-life/ residual useful life in discharge half-cycle *m* |
|  |  | Voltage curve in discharge half-cycle *m* |  |  | Data-driven feature for total charge time in discharge half-cycle *m* |
|  |  | Current curve in discharge half-cycle *m* |  |  | Data-driven feature for age in discharge half-cycle *m* |
|  |  | Temperature curve in discharge half-cycle *m* |  |  |  |
|  |  |  |  |  |  |
|  |  | Continuous-*t* data in charge half-cycle *m* |  |  | Data-driven features in charge half-cycle *m* |
|  |  | Capacity curve in charge half-cycle *m* |  | / | Data-driven feature for end-of-life/ residual useful life in charge half-cycle *m* |
|  |  | Voltage curve in charge half-cycle *m* |  |  | Data-driven feature for total charge time in charge half-cycle *m* |
|  |  | Current curve in charge half-cycle *m* |  |  | Data-driven feature for age in charge half-cycle *m* |
|  |  | Temperature curve in charge half-cycle *m* |  |  |  |

**The illustration of *n*, *s*, *m*, and *a***

Since we have enabled the flexibility of the input data for the inference, four symbols related to each cycle, i.e. *n*, *s*, *m*, and *a* are defined, and each of which has different definition. Symbol *n* represents cycle number in the dataset, and thus, the range of *n* for a given battery is . Symbols *s*, *m*, and *a* are defined for sampling a segment of cycles from the entire life time of the given battery. Symbol *s* is defined as the number of shifted cycles from cycle number used as the first cycle of input for either inference or training. Symbol *a* is defined as the total number of cycles used as input. Symbol *m* is defined to describe each cycle used as input, and thus, the range of *m* for a given input is .

Fig. S 1 shows two examples. The first case of *s*=0, *a*=5 indicates that the first five cycles of the battery (cycle number: 1…5 in the dataset) are extracted and with the features in the fifth cycle padding to form the input data for the inference of EoL, … etc. by Discharge DNN. The second case of *s*=5, *a*=48 indicates that 48 cycles of the battery from the cycle number 6 (cycle number: 6…53 in the dataset) are extracted and with the features in the 48th cycle padding twice to form the input data for the inference of RUL, *s*… etc. by Full RUL DNN.

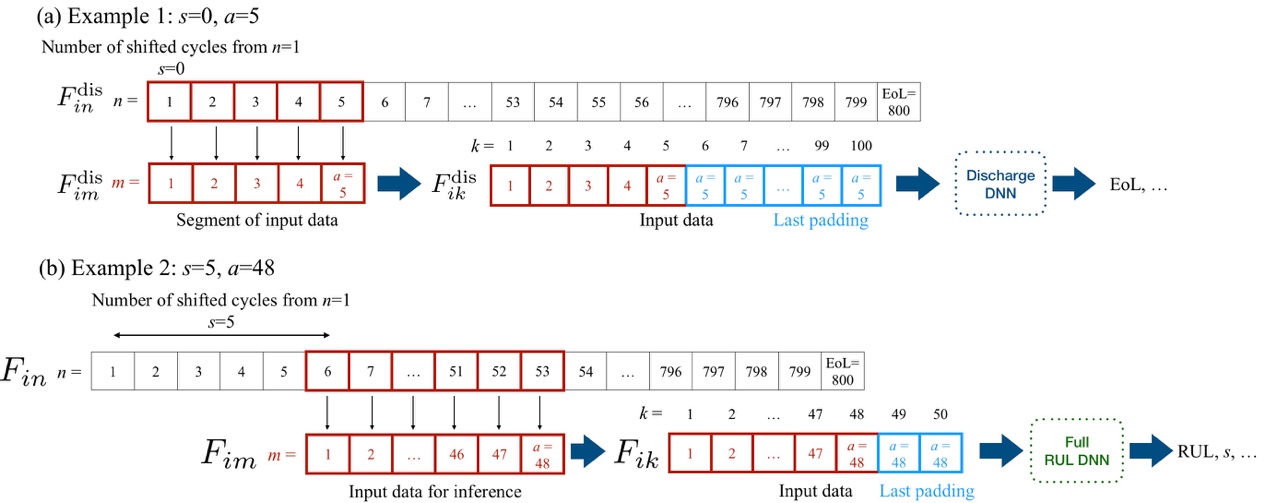
**

Fig. S 1: The illustration of two examples. (a) *s*=0, *a*=5 and (b) *s*=5, *a*=48 for the inference by Discharge DNN and Full RUL DNN, respectively.

**The detail of each NN pipeline**

The workflow of the three DNNs in the current work, as shown in Fig. S 2. Where cubes represent data in tensor form and rounded rectangles represent neural networks. The elements/building blocks/inputs/outputs etc. of each NN pipeline have been illustrated in Fig. S 3 to Fig. S 13, showing all the detail of the current DNNs. Note that, due to the limitation of the article length, and predicted by Predictor 4 for cycle-by-cycle information are not mentioned in manuscript and shown in supplementary only. The codes for prediction and the two experiments in supplementary are released in GitHub: <https://github.com/acctouhou/Prediction_of_battery>

The entire pre-trained models and dataset of this work are available as follow:

<https://drive.google.com/drive/folders/1Aq-wfoQ8ltDqziyHUcka7oUncQ7NgSP8?usp=sharing>

Note that this link is shared by author’s personal Google Drive and only available for the current review process.

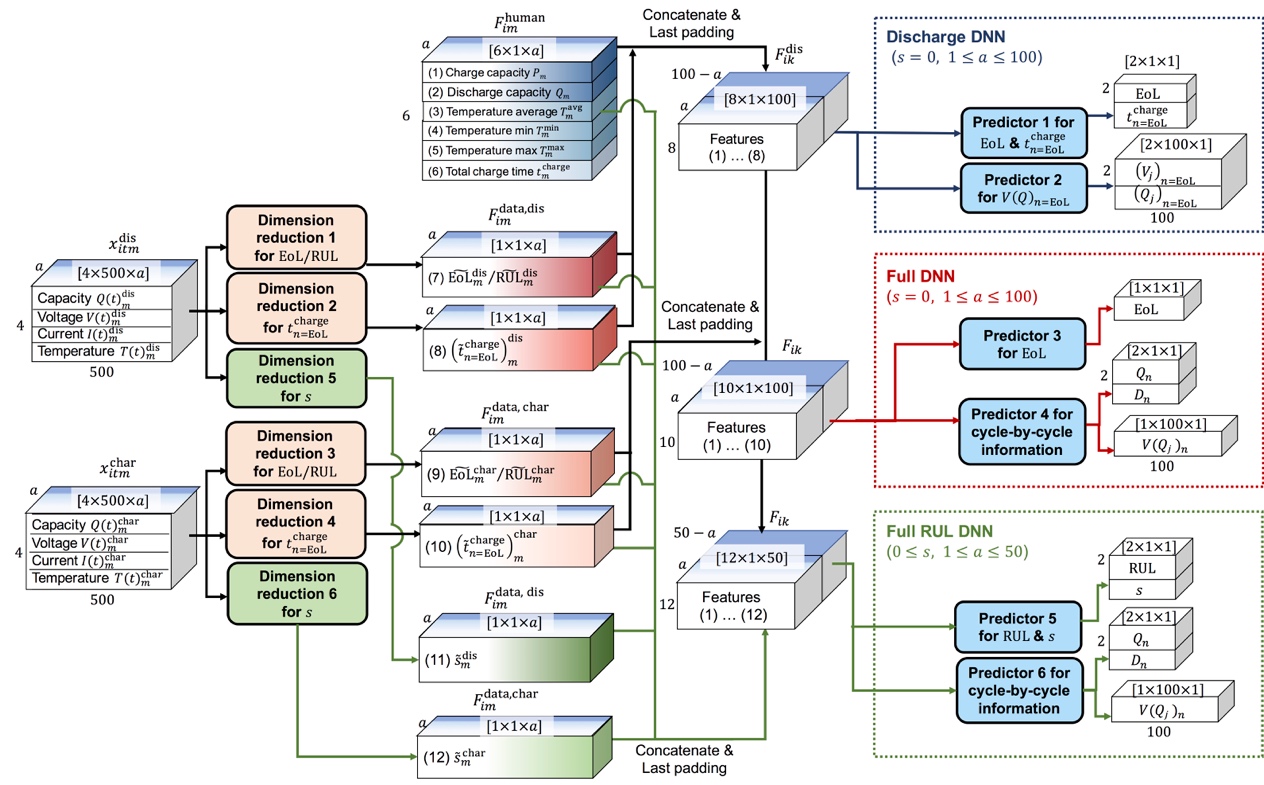


Fig. S 2: The workflow of the three DNNs in the current work. Where cubes represent data in tensor form and rounded rectangles represent neural networks.

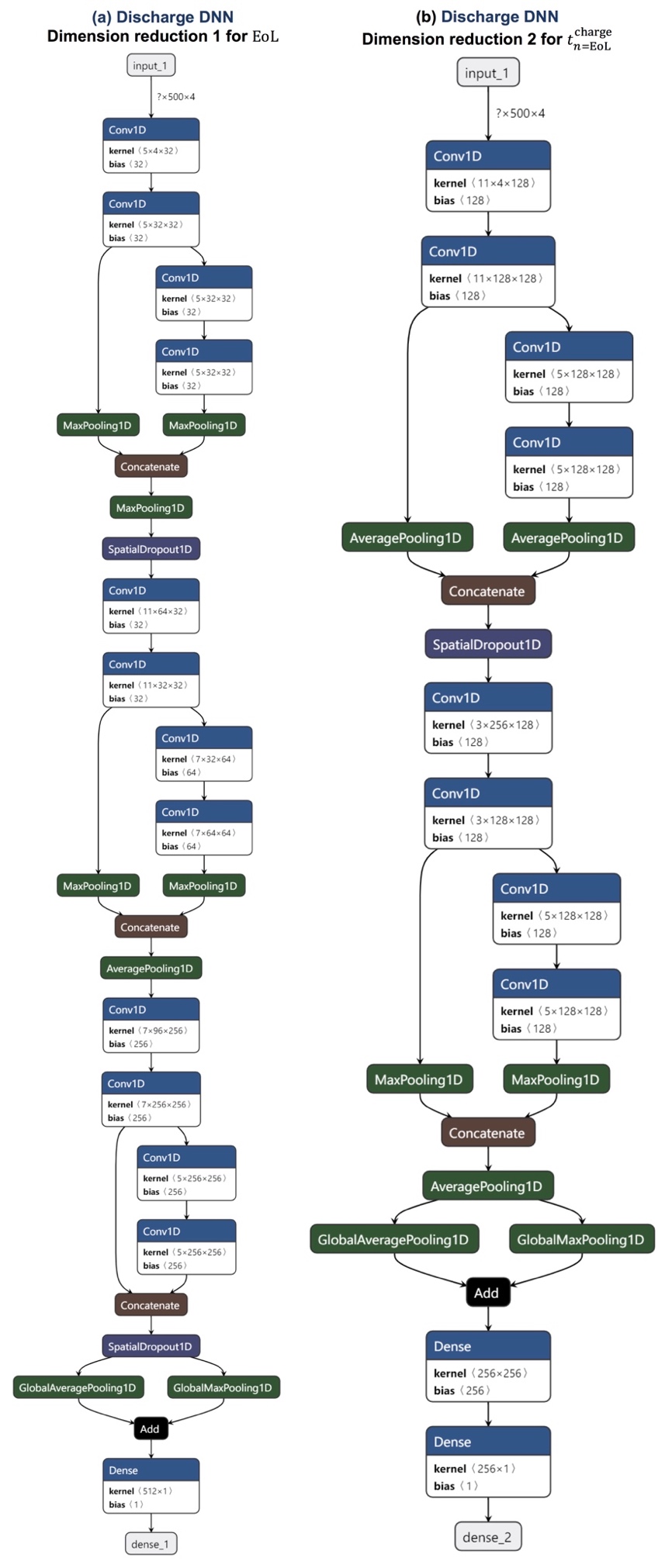


Fig. S 3: The architecture of Dimension reduction (a) 1 and (b) 2 in Discharge DNN.

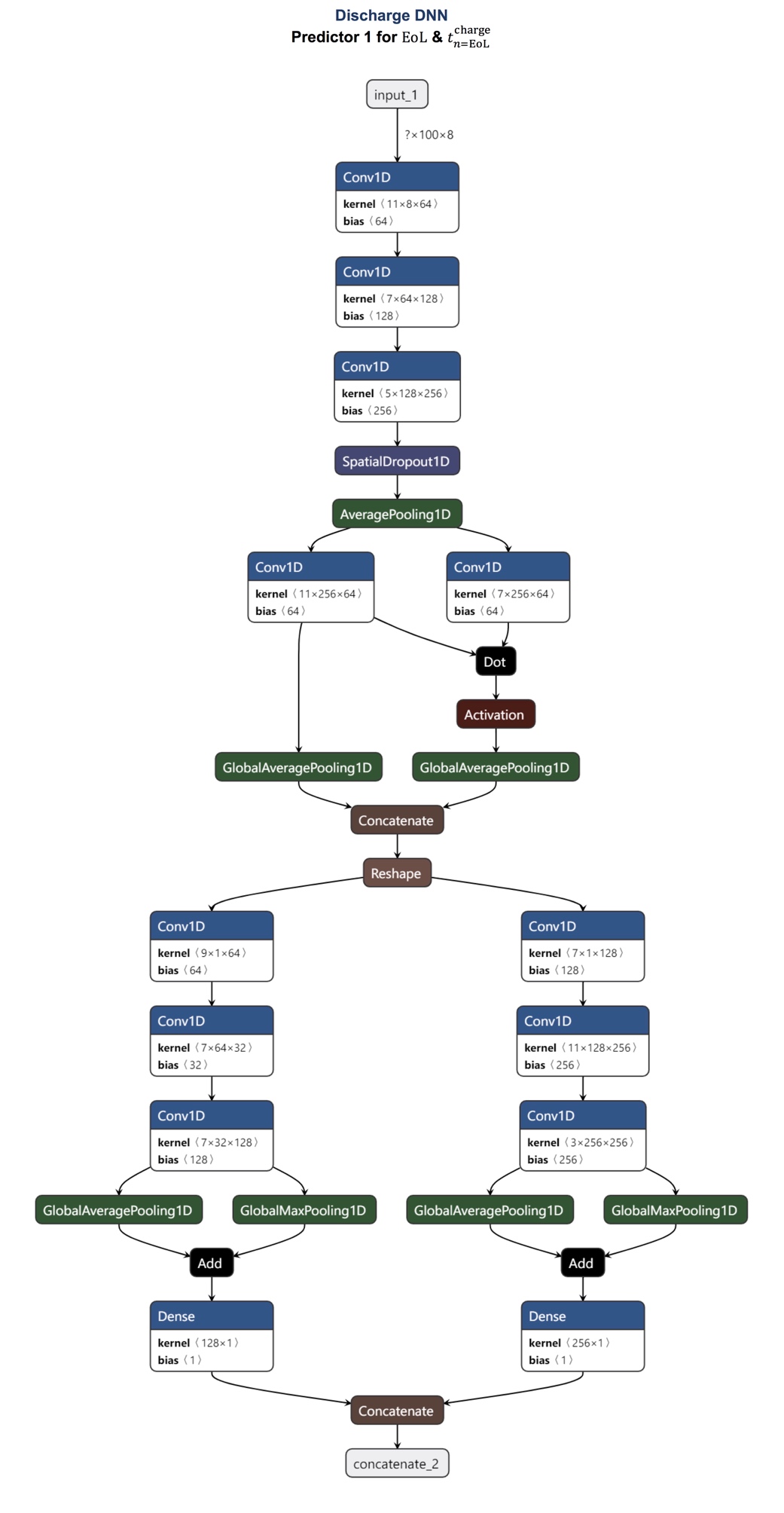


Fig. S 4: The architecture of Predictor 1 in Discharge DNN.

.

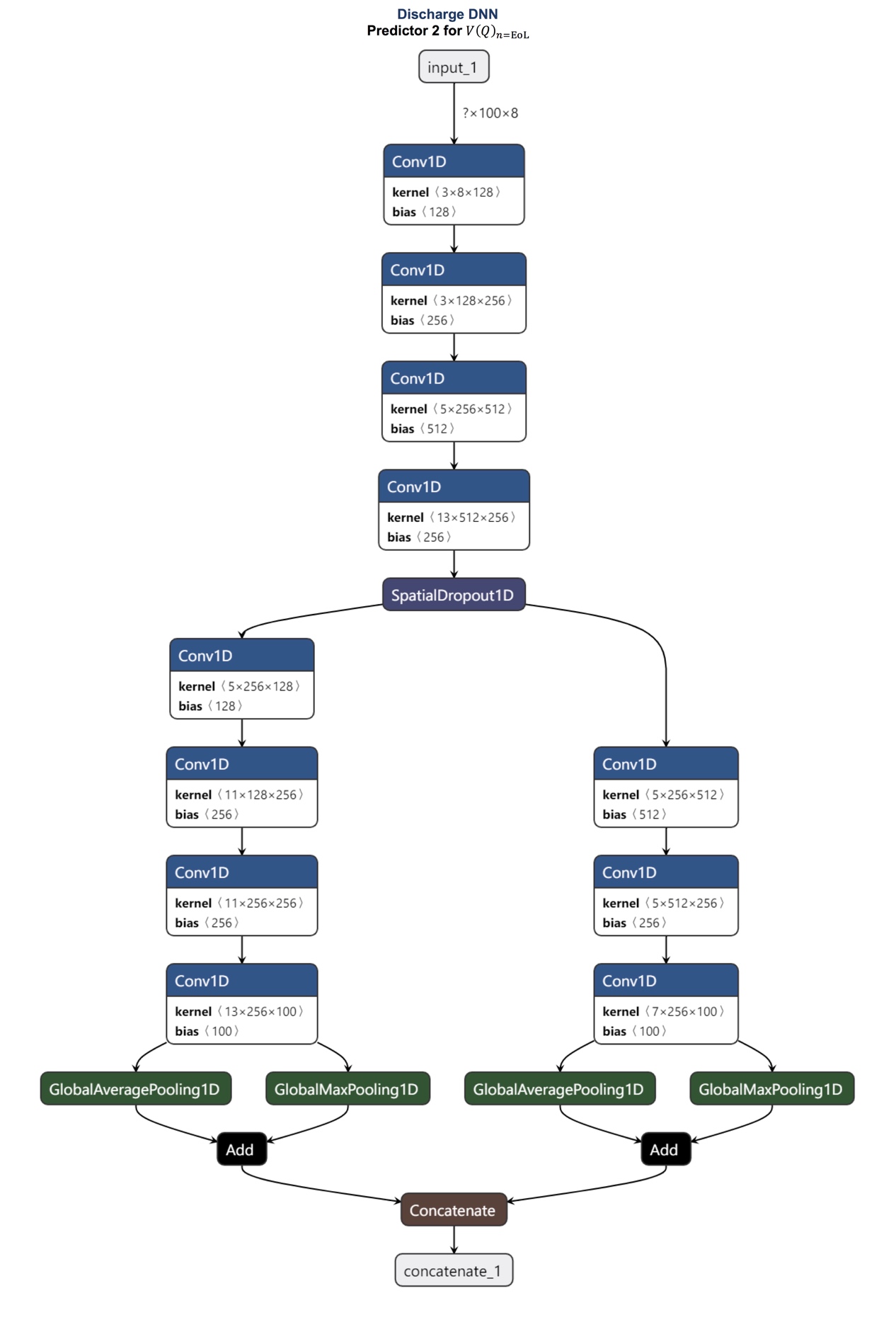


Fig. S 5: The architecture of predictor 2 in Discharge DNN.

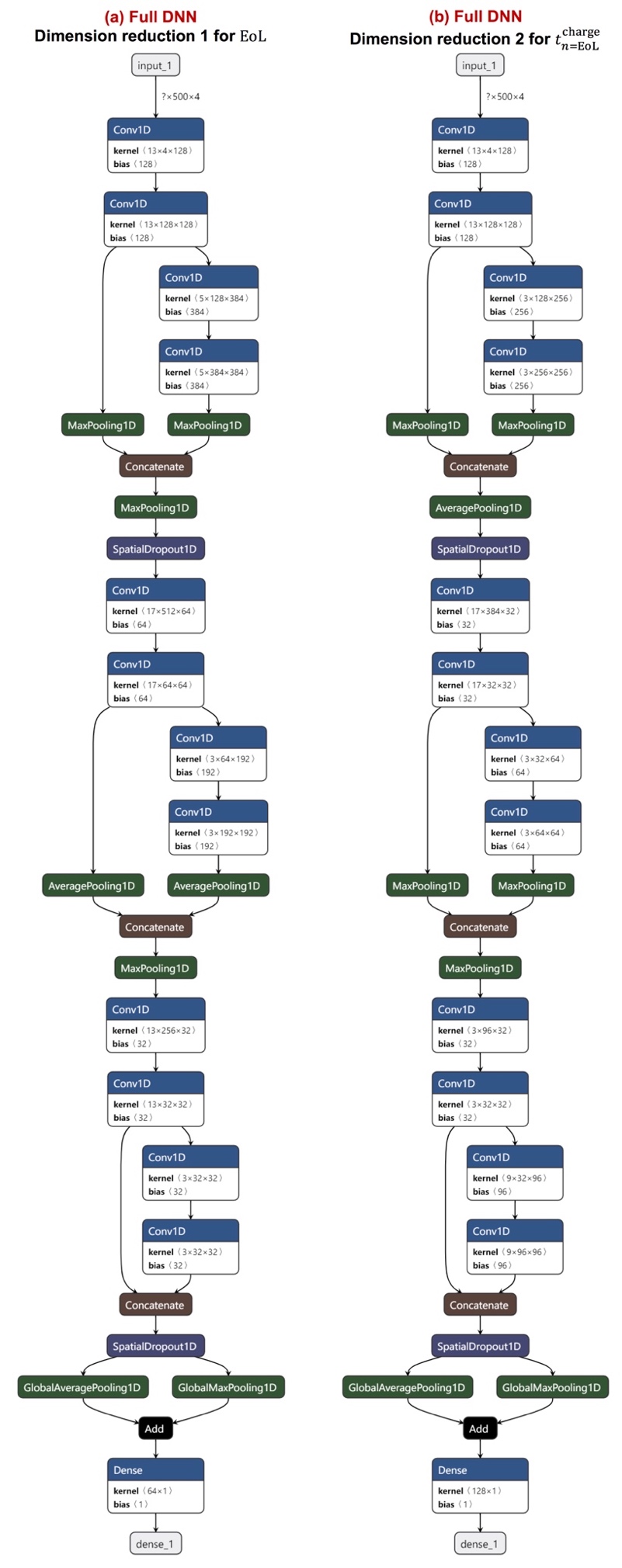


Fig. S 6: The architecture of Dimension reduction (a) 1 and (b) 2 in Full DNN.

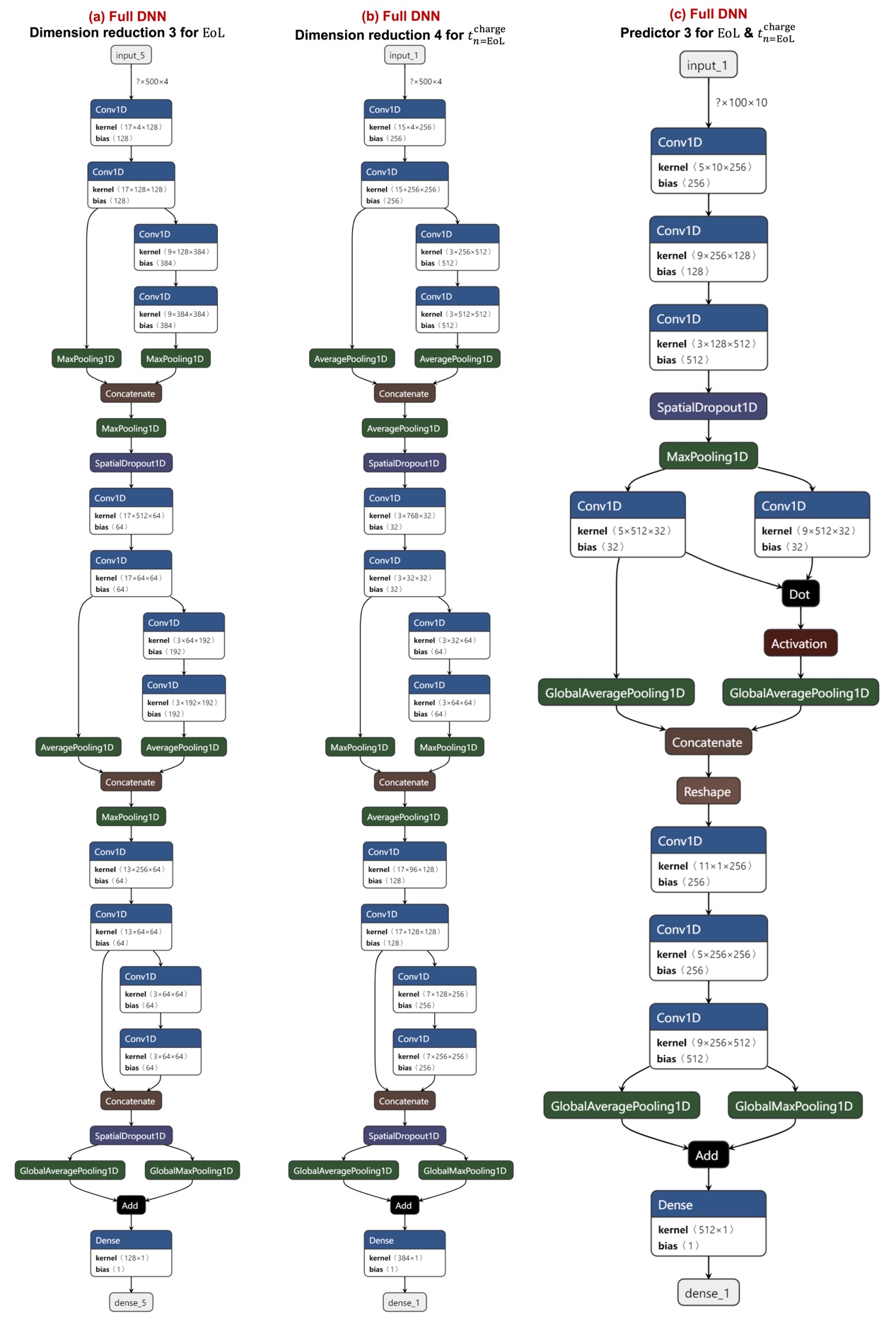


Fig. S 7: The architecture of Dimension reduction (a) 3 and (b) 4 in Full DNN.

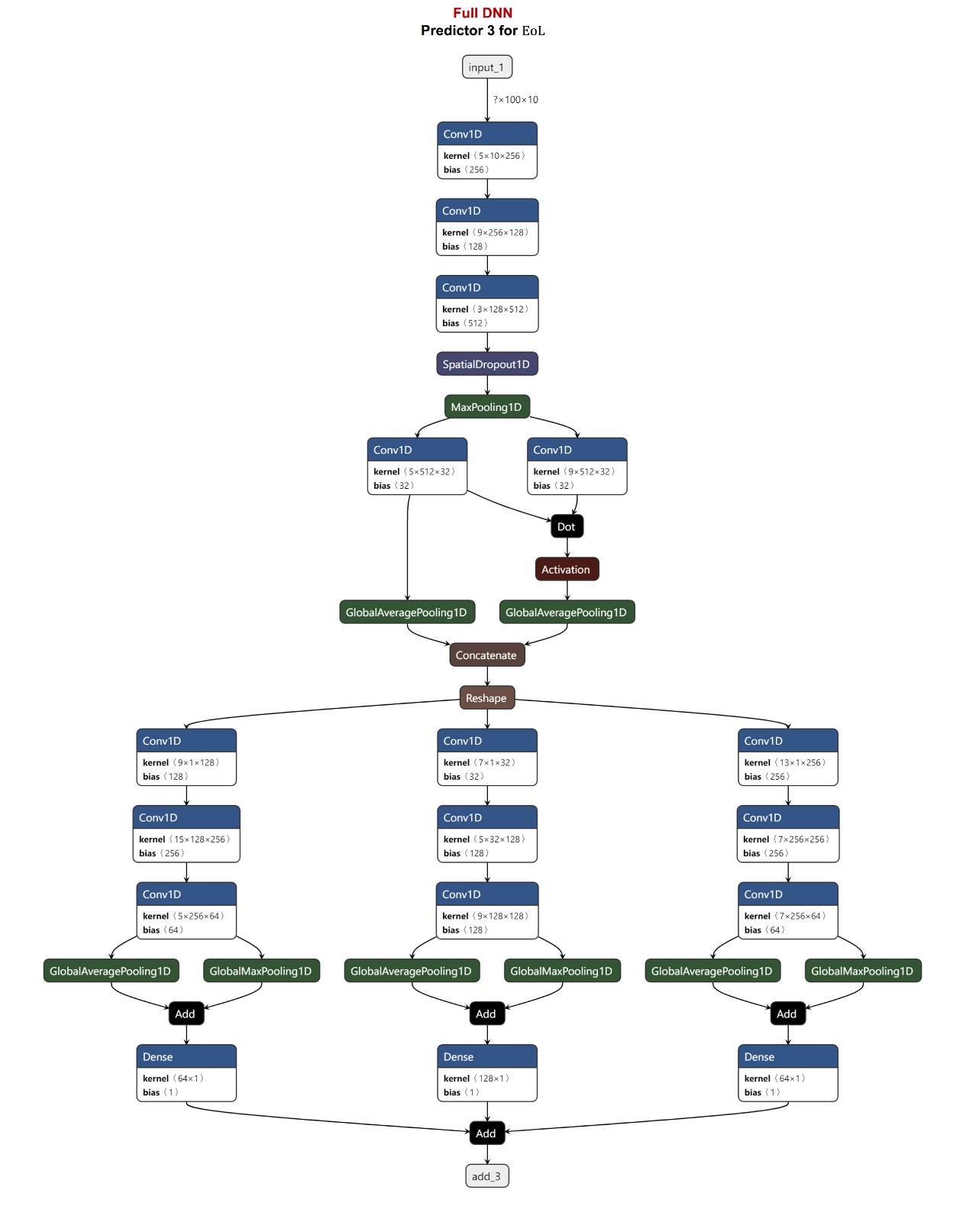


Fig. S 8: The architecture of Predictor 3 in Full DNN.

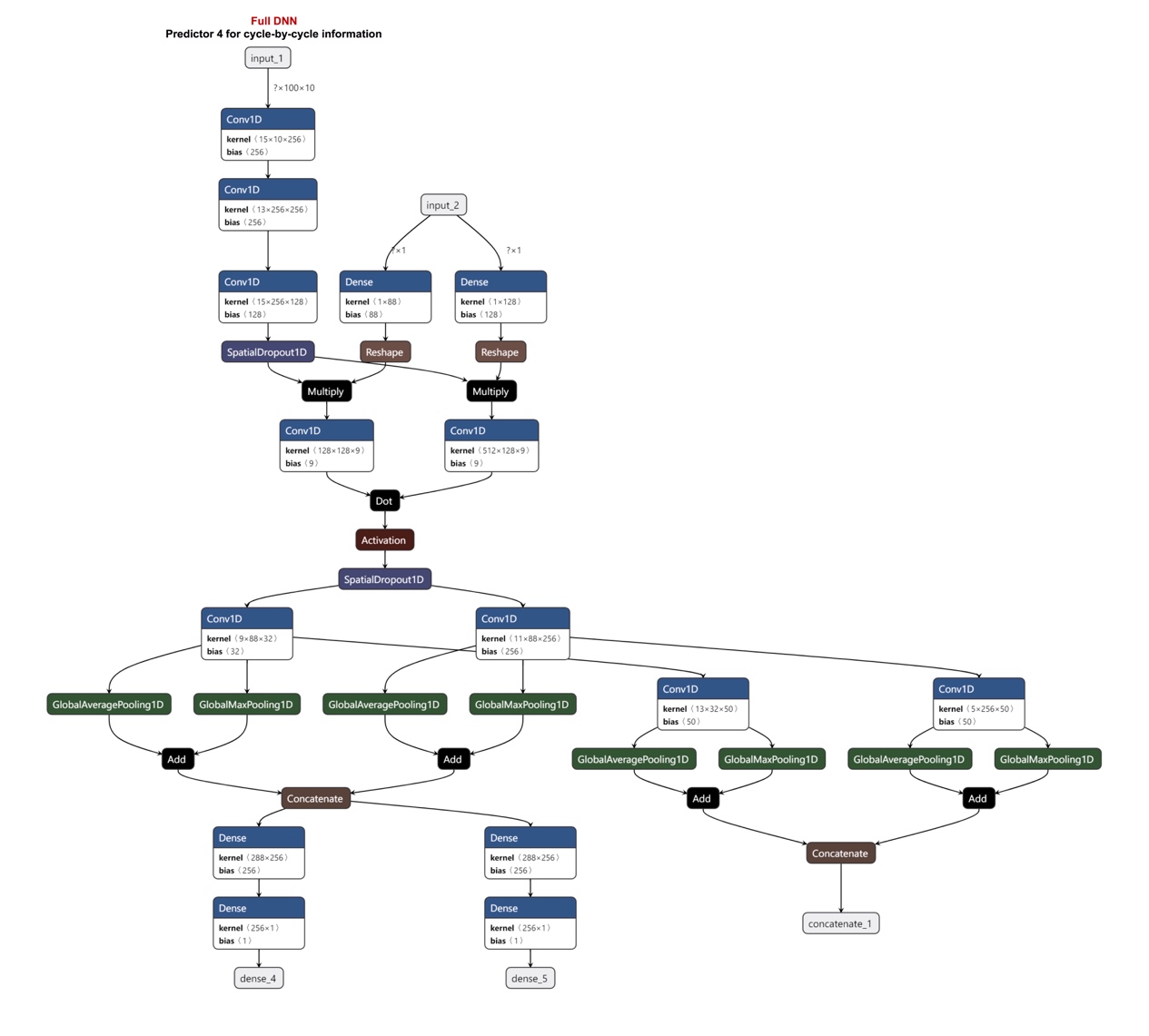


Fig. S 9: The architecture of Predictor 4 in Full DNN.

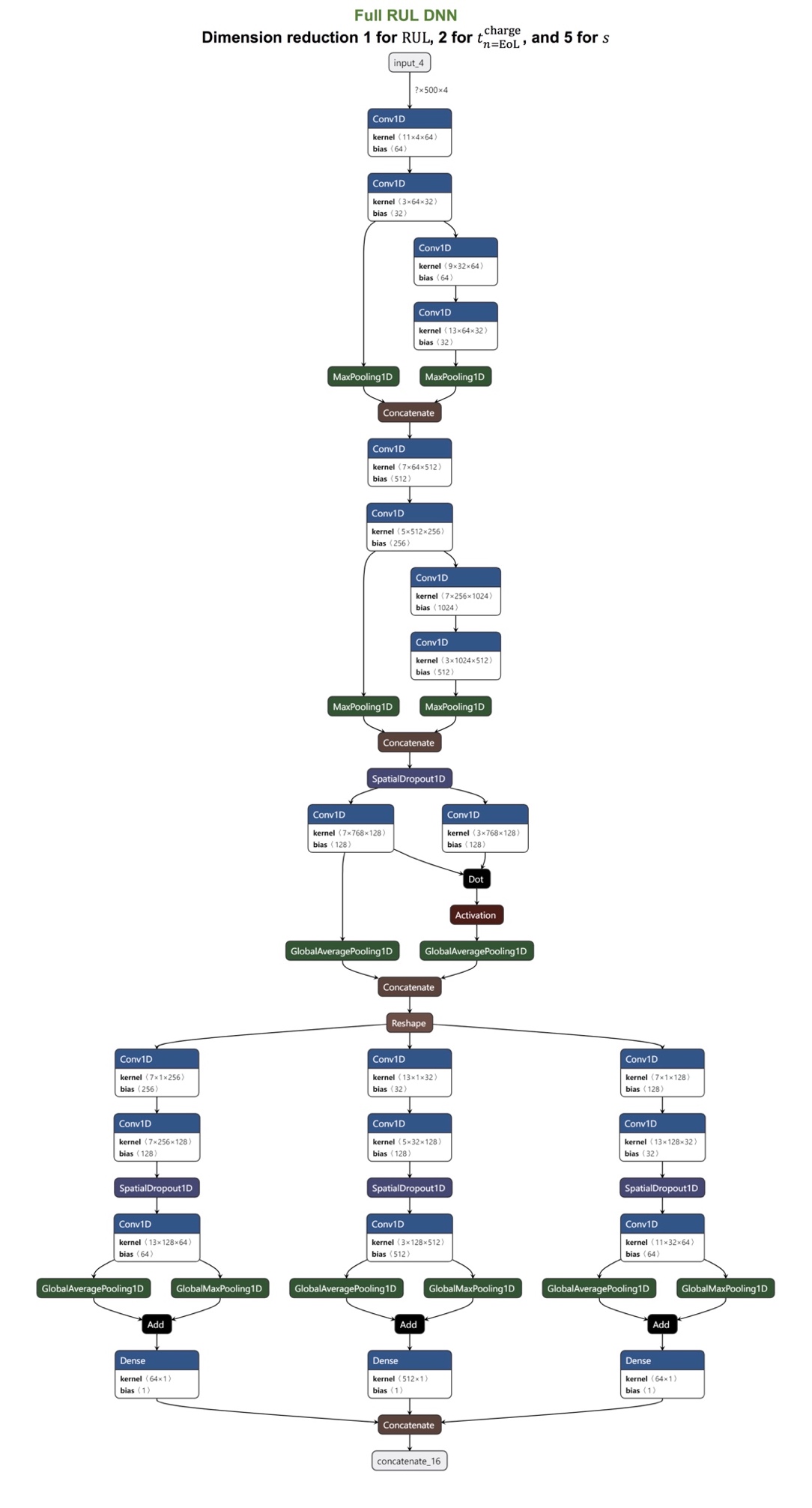


Fig. S 10: The architecture of Dimension reduction 1, 2, and 5 in Full RUL DNN.

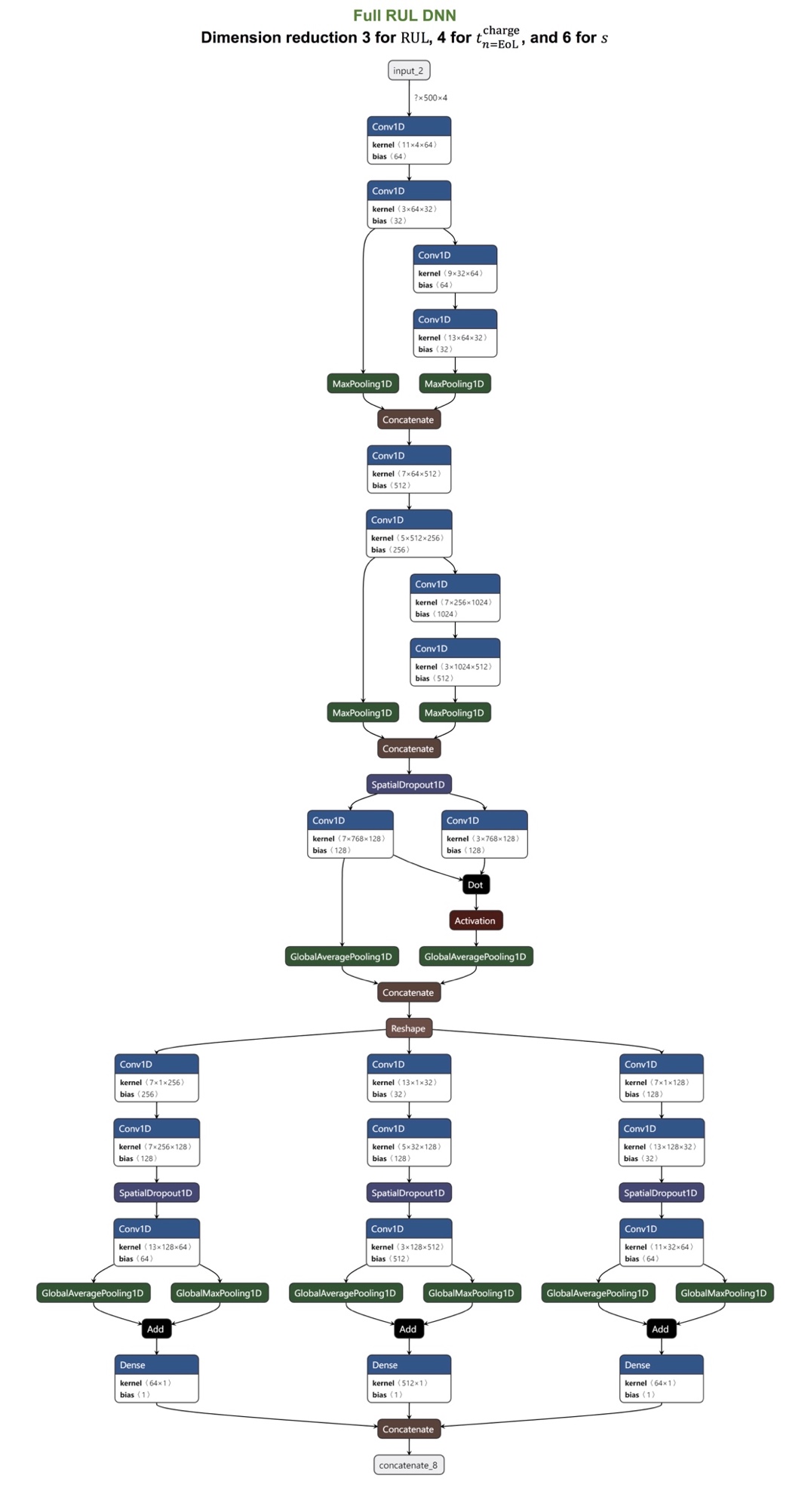


Fig. S 11: The architecture of Dimension reduction 3, 4, and 6 in Full RUL DNN.

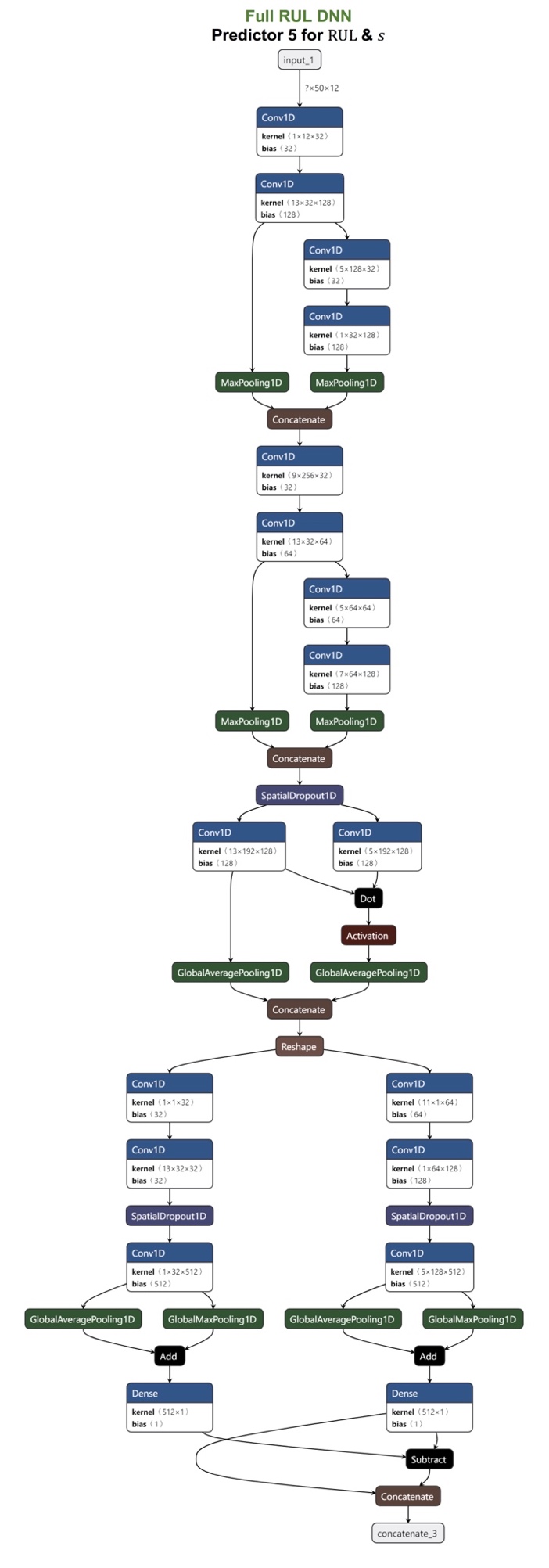


Fig. S 12: The architecture of Predictor 5 in Full RUL DNN.

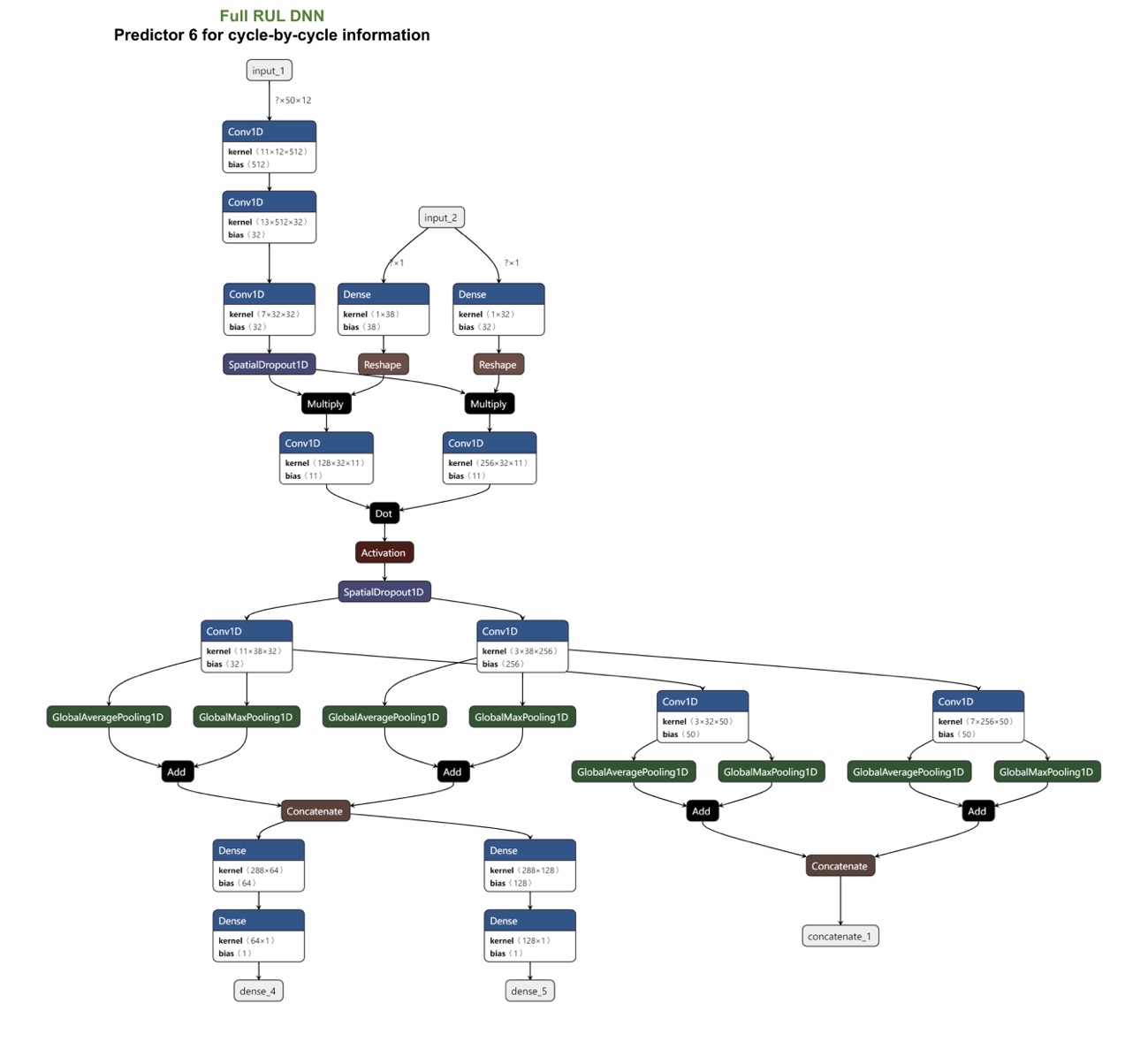


Fig. S 13: The architecture of Predictor 6 in Full RUL DNN.

**The illustration of the testing strategies used in the current DNNs**

As mentioned in the main texts, three different testing strategies were adopted in the DNNs. The illustration of the three selection methods for training and testing datasets was shown in Fig. S 14. Where each grid represents a single cycle, and each column represents the case of a battery. Fig. S 14 (a) shows that testing strategy <1> unselected the entire sequences of randomly chosen batteries; Fig. S 14(b) shows that testing strategy <2> unselected the entire sequences of all the batteries with randomly chosen charging policies. Sampling started from the first cycle () and the length of each sampling may vary from . Fig. S 14 (c) shows that testing strategy <3> was similar to strategy <2>. However, convolutional sampling was adopted here where *s* may vary and , corresponding to Full RUL DNN. Note that the convolution sampling for each selected battery continued until the cycle of was reached, in order to avoid the unstable properties and behavior of the battery near its EoL. These testing strategies were designed to force DNNs to achieve their own objectives.

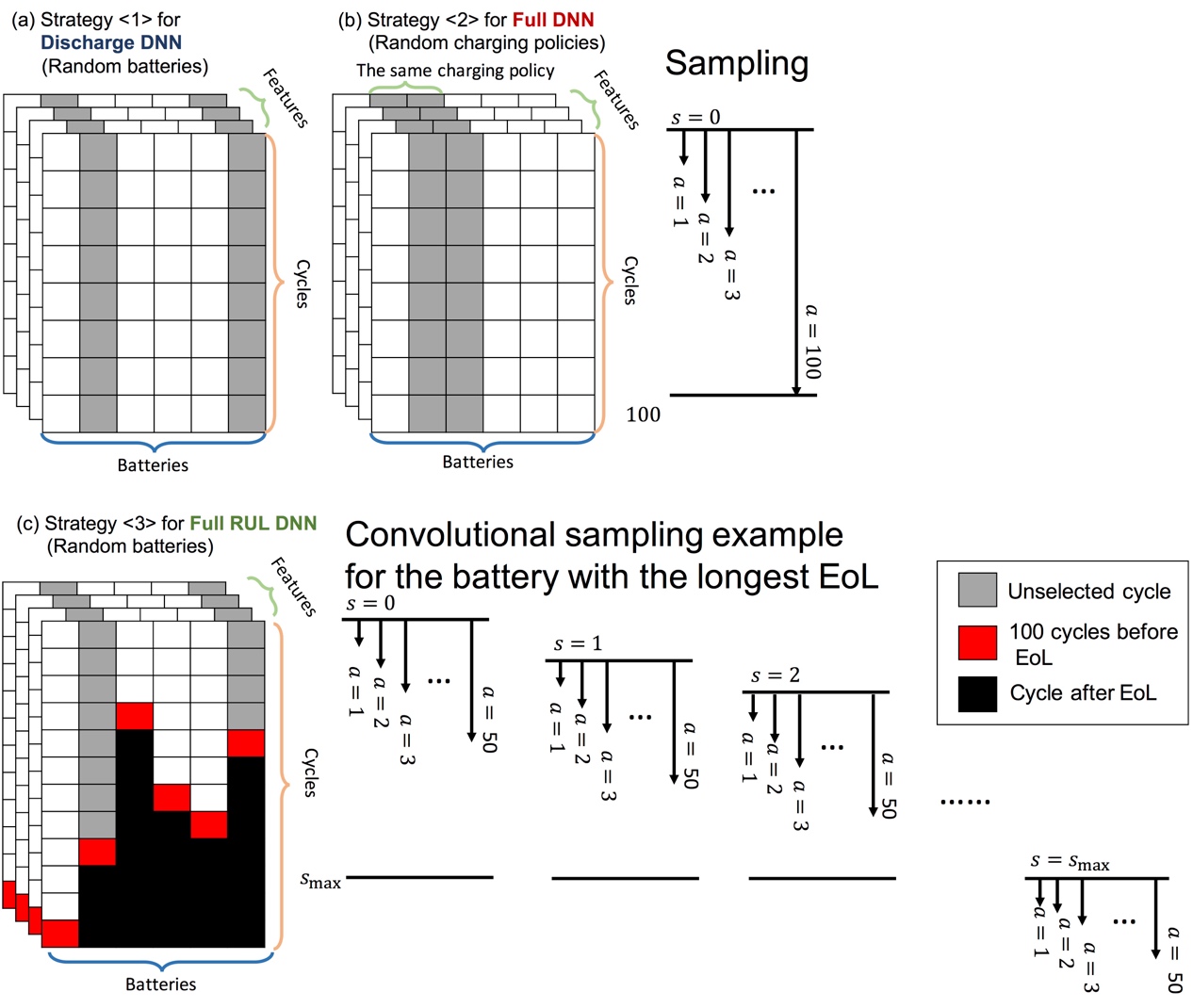


Fig. S 14: Three testing strategies. The grids colored in gray will be used as testing sets; the white grids will be used as training sets; the red grids are the 100 cycles before EoL; the black grids are the cycles after EoL of that battery.

**The ability of extrapolation of Full DNN for unlearned charging policies**

There were 68 different charging policies in the datasets provided by Severson et al. Full DNN adopted special network structure and considered the data of charge process, enabling accurate prediction of EoL even charging policies that have not been learned. This remarkable extrapolation ability for EoL prediction can be illustrated by Fig. S 15. The three axes of the figures (*C*1, *C*2, and *Q*1) represent the first and second applied discharge currents, and the state of charge current (%) where the current switched, respectively (the illustration for these symbols is given in Fig. S 15(c)). This indicates that each point specifies a charging policy. Fig. S 15(a) shows the RMSE of predicted EoL for each charging policy, based on the first cycle only, when all the cells with that charging policy were used as testing sets, i.e. they were not included in training sets. It can be observed that most of the points are in dark blue color (low RMSE) in Fig. S 15(a). Points with colors other than dark blue typically accumulated in the region of *C*2 < 4. This implied that lower currents applied at the second stage of charge process may be less harmful to cells, leading to less aging features in the first few cycles for Full DNN to inference. Next, Fig. S 15(b) shows the reduction of RMSE for each charging policy when the first 100 cycles were considered. It is interesting that Fig. S 15(a) and (b) look almost the same, indicating that cells with certain charging policies which were not well captured by Full DNN with fewer input cycles, had significant improvement as long as large enough number of cycle used as input.



Fig. S 15: (a) RMSE of predicted EoL for each charging policy, based on the first cycle only, when all the cells with that charging policy were used as testing sets. (b) The reduction of RMSE for each charging policy when the first 100 cycles were considered. (c) The illustration of the charging policy adopted in Severson et al.

**The learning progress of the current DNNs**

Here we take Discharge DNN as an example to illustrate the learning progress of the DNNs on the current battery problems. The loss (RMSE) of Discharge DNN corresponding to training epochs and first a cycles as input data was visualized as a 3-dimensional landscape in Fig. S 16. At the early stage of the training, the loss for the cases with more input cycles decreased more significantly compared to the cases with fewer input cycles. After 50 training epochs, the loss for cases with more than the first 80 input cycles converged at the minimum. This indicated that cases with more input cycles dominated the weights and biases of the DNN. Then, the optimizers of the DNN were able to learn about the cases with fewer input cycles more efficiently due to the assistance of those existed parameters predetermined by the cases with more input cycles.

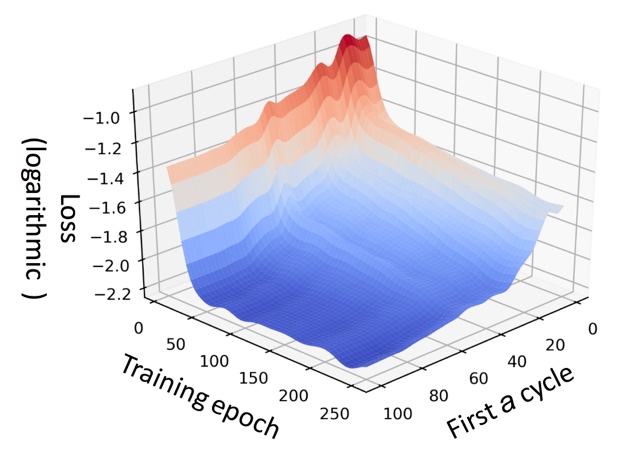


Fig. S 16: The 3-dimensional landscape of the loss of Discharge DNN corresponding to training epochs and first a cycles as input data.

We also used the relevance score of each feature corresponding to the first *a* input cycles obtained by Deep Taylor decomposition (DTD) to visualize the learning progress of the two DNNs about each feature, as shown in Fig. S 17. For the illustration purpose, features of similar types, such as features related to capacity and temperature, were averaged and merged into one curve. Note that the area under these curves represented the information about the corresponding features learned by the DNN. It can be observed that, for both the two DNNs, a significant amount of information has been learned before the first 40 input cycles and the data after the first 60 or 80 input cycles were less influential to the predicted results.

It is of interest noting that the curve for Discharge DNN indicated that the predicted results were still greatly affected by the data around the 80th cycle. This is because of the uncertainty introduced by the unlearned charging policies, and apparently, the features of these policies were very difficult to be extracted by Discharge DNN with the discharge input data only. Full DNN resolved this problem by considering both the charging and discharge input data. The sufficient amount of data assisted Full DNN to extract features.

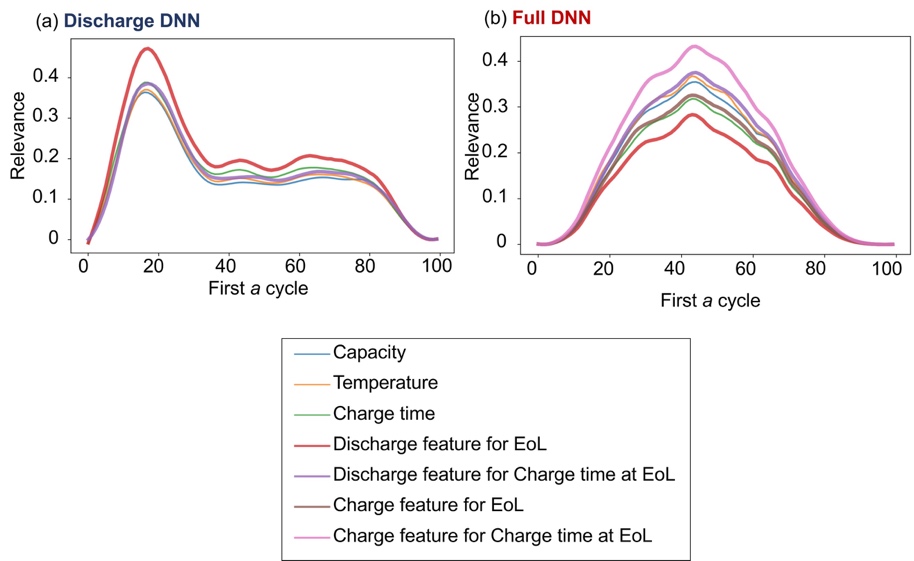


Fig. S 17: The DTD relevance score of each feature corresponding to the first a input cycles for (a) Discharge and (b) Full DNN. A significant amount of information has been learned by all the two DNNs before the first 40 input cycles and the data after the first 60 or 80 input cycles were less influential to the predicted results.

The DTD relevance score of each feature was used again to demonstrate the flexibility and the robustness of EoL prediction of our DNNs. Since network in network (NiN) structure was adopted in Predictors in all the current DNNs, the sequence length of the input data can be varied by using the last padding technique. Here, input data with a length of 60, 80, and 100 (the present setting described in the main texts) cycles for the previous version of Discharge DNN was taken as an example here. The bounded DTD approach was applied to interpret the behavior of the DNN and the corresponding relevance scores of human-picked and data-driven features with different transparency are shown in Fig. S 18. It can be observed that the shapes of the curves are all similar, showing great robustness of the learning progress of the current DNN even with fewer input cycles. More importantly, the peaks of all the three cases are in the range of the 20th and 40th cycles. This demonstrates that the power of the current DNN in terms of serving as an early predictor of EoL of batteries.

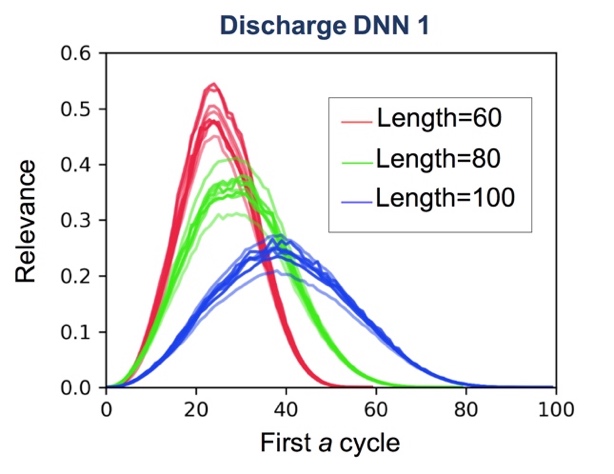


Fig. S 18: The DTD relevance score of the previous version of Discharge DNN with an input data length of 60, 80, and 100 cycles, showing flexibility and the robustness of EoL prediction.

**Five-fold cross-validation tests of the current DNNs**

The reproducibility and reliability of the two DNNs were examined by five-fold cross-validation tests. Five different training sets were used to train the three DNNs. Fig. S 19 shows the range of RMSE of EoL vs. the number of consecutive input cycles for all the five cases in blue; the average of the curves was marked by a dashed line. It can be observed that the fluctuation of RMSE occurred in both DNNs in the early stage. This was because batteries were typically at a relatively healthy state initially, showing almost no aging features in the curves. Thus, it is reasonable that DNNs may produce relatively inaccurate results based on fewer cycles in the early stage. Another reason was that the quality of measurement in each cycle is more influential to the performance of prediction when fewer input cycles were considered. These issues can be solved by either feeding more data of cycles to the DNNs or providing data of the batteries with early occurrence of aging where extreme charging/discharge policies were applied.

As the number of input cycle *a* becomes greater than around 30, all DNNs reached their converged values of RMSE. Wherein, the converged RMSE value of Discharge DNN was less than it of Full DNN. This is reasonable due to the different nature of the two DNNs. As mentioned above, the training datasets of Discharge DNN missed the data of some randomly selected batteries (testing strategy <1>); those of Full DNN missed the data of certain batteries with some randomly selected charging policies (testing strategy <2>). Thus, Discharge DNN and Full DNN were forced to have the ability to extrapolate for unknown batteries and charging policies, respectively. Thus, the prediction of Full DNN encountered a much higher complexity than Discharge DNN. Fortunately, Fig. S 19 shows that both of DNNs achieved high robustness as the input cycle number increased.

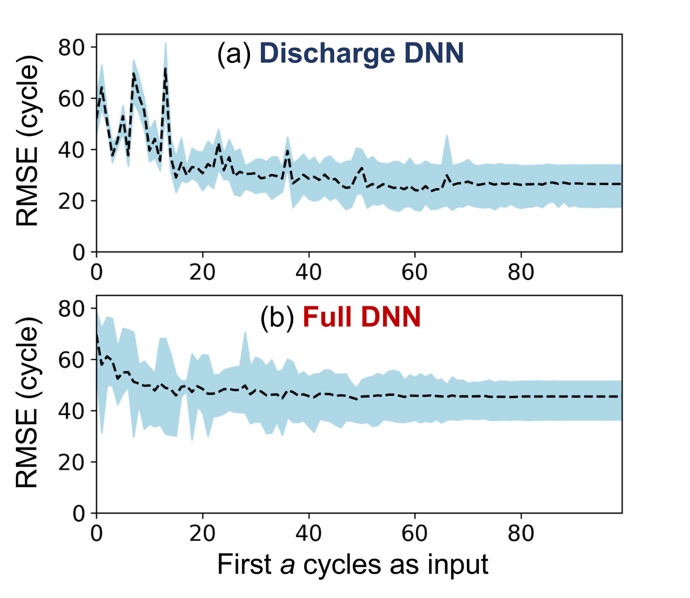


Fig. S 19: The range of RMSE of EoL vs. the first *a* cycles as input data for the five-fold cross-validation tests colored in blue. Where the average of the five curves marked with the dashed line.

**Data screening process for the dataset provided by Attia et al.**

We have implemented a program and screen out the unusable data of batteries both manually and automatically in the dataset provided by Attia et al., based on four types of reason. First, the experiment log provided in the dataset package stated that batteries in channel 5, 12, 45, 46, 48 are problematic. Second, after the data was visualized, we found that curves of some batteries are greatly shifted between different cycles. For example, current curves of batteries in channel 9, 17, 31, 34 and voltage curves of batteries in channel 9, 17, 30, 34, 39 are greatly shifted, where one of the cases were shown in Fig. S 20(a)(b). In addition, Fig. S 20 (b) also shows there was an unusual voltage drop at the early stage of the measurement step in one of the cycles, marked by a green circle. Third, the patterns of the measured curves of certain batteries are weird and different from the most of the batteries in the dataset. For example, temperature curves of batteries in channel 14, 15, 16, 17, 30, 34, 44 are messy, where one of the cases were shown in Fig. S 20(c). Forth, an unreasonable negative value recorded in time log at a random measurement step in the 31st to 35th cycle during the charging or discharging process in all the batteries. This systematic problem was not recoded in their experiment log. Thus, we did not inference these batteries as they did not pass our data screening program.

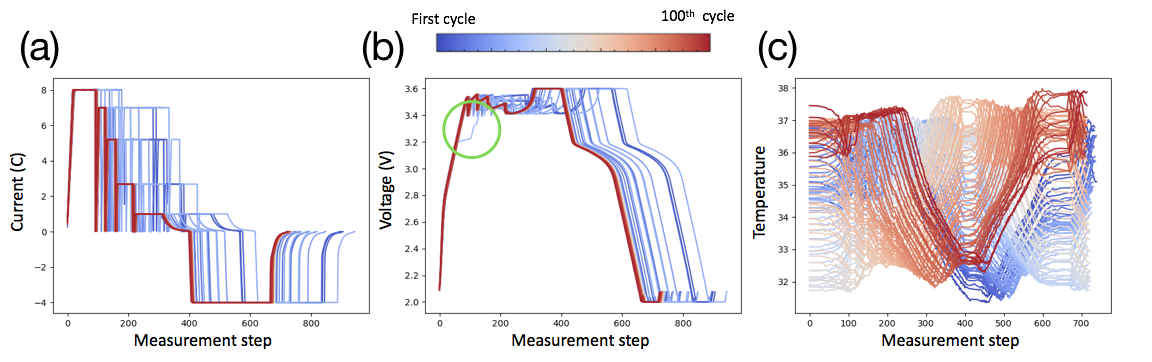


Fig. S 20: One of the examples that (a)(b) the measured curves are greatly shifted between different cycles, and (c) the patterns of the curves in the battery are different from the most of the batteries in the dataset.

We also have applied Full DNN to inference EoL of 34 batteries in Attia et al. These 34 batteries were chosen because their data of the first cycle only can pass our data screening program, while the quality of the remaining cycles may not have acceptable quality for the inference of EoL. The predicted values EoL vs. EoL is shown in Fig. S 21(a). It can be observed that our Full DNN successfully predicted EoL of the 34 batteries with very low mean-absolute-percentage-error of 9.6%, based on the data of the first cycle () only.

We further colored these points based on the categories defined in Attia et al. in Fig. S 21(b). It can be observed that the EoL prediction of the batteries in the category of CLO TOP 3 was with relatively higher error. We believe this is because, during the charging process of batteries of CLO TOP 3, the current curves had deep gaps between steps in the given charging policies (marked by green circles in Fig. S 22(a)), and the gaps were not severe (Fig. S 22(b)) in the other batteries marked by blue and green dots in Fig. S 21(b).

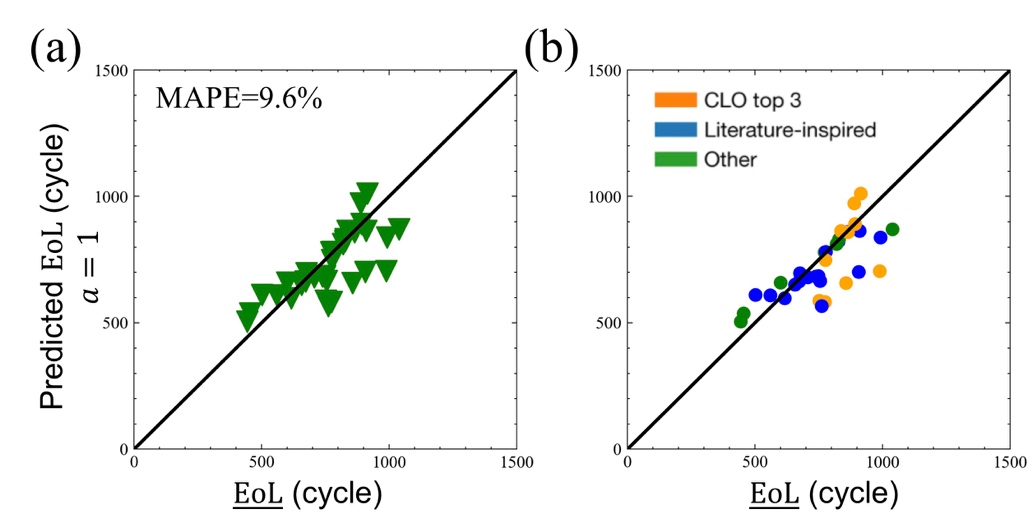


Fig. S 21: The predicted values EoL vs. EoL plot with MAPE of about 9.6%. Where Full DNN predicted 34 batteries provided by Attia et al. based on the data of the first cycle only (). (b) Data points are colored based on the charging protocol specified in Attia et al., where the EoL prediction of the batteries in the category of CLO TOP 3 was with relatively higher error.

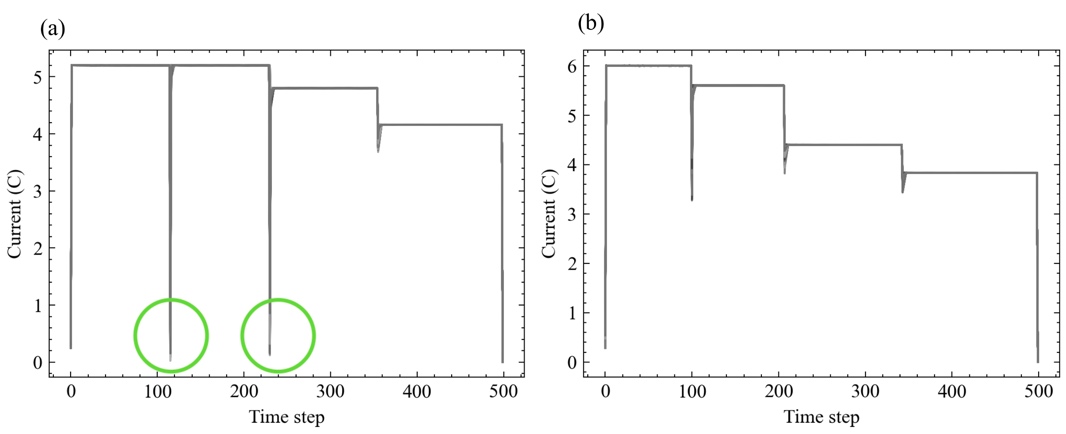


Fig. S 22: Typical examples of current curves corresponding to the charging policies (a) “CLO TOP3” and (b) “Literature inspired” and “Other” provided by Attia et al.

**Attribution of the current DNNs**

A backward propagation method, layer-wise relevance propagation (LRP), is used to analyze Full DNN and determine the relevance between EoL (the output layer) and all data points in the discharge curves of voltage in each cycle (the input layer). We took four batteries with the fourth and second shortest and longest EoL as examples without losing generality. Their EoL were 335, 429, 1284, and 1638 cycles. Relevance scores determined by LRP were plotted on the curves from the 1st to the 100th cycles, as shown in Fig. S 23. It can be observed that features which lead a short EoL for batteries (colored in red) can be discovered in many regions in the two bad batteries by our DNN. Similarly, features which lead a long EoL for batteries (colored in green) can also be discovered in the two good batteries. This indicates that our DNN can capture the key features to the prediction of EoL.

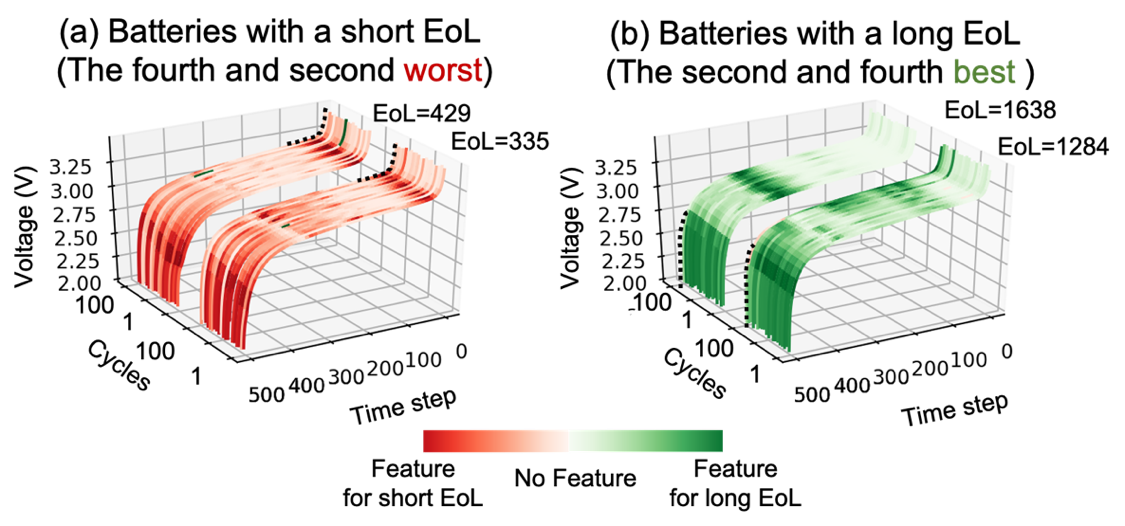


Fig. S 23: Relevance scores determined by LRP plotted on voltage curves in discharging process of batteries with a (a) short and (b) long EoL. Where green and red colors indicate features for leading to a long and short EoL, respectively, while white indicates no feature for EoL were found. Segments marked by dashed lines will be shown in Fig. S 24.

It is of interest to correlate between the value of voltage and EoL. Two segments of the voltage curves were chosen marked by dashed lines in Fig. S 23. Then, the voltage curves at that segment of all the batteries in the dataset and their corresponding EoL were plotted in Fig. S 24. It can be observed that the results show that the higher voltage at these segments has strong correlation with longer EoL[1]. This section demonstrated DNN and LRP can provide in-depth insight to design batteries or discharge conditions.

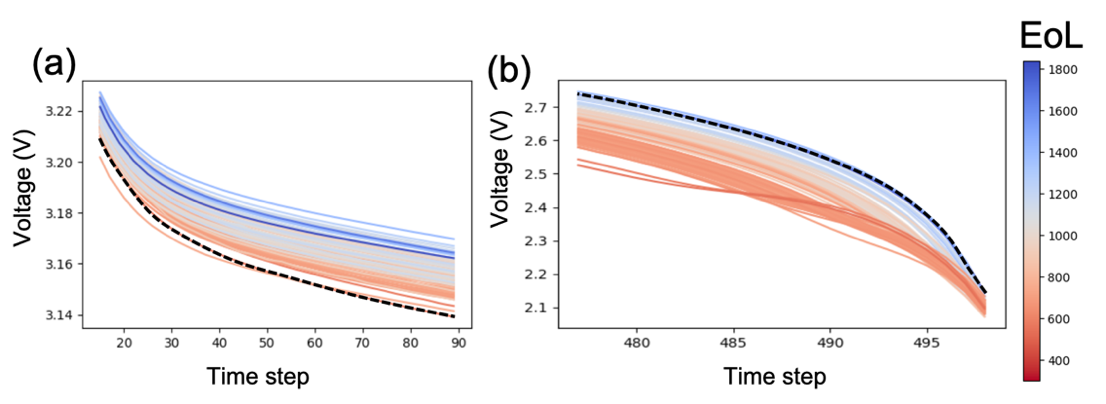


Fig. S 24: EoL and selected segments of voltage curves in discharging process for all batteries in the dataset at the 100th cycle, showing that higher voltage within these segments having longer EoL. Here, dash lines in (a)(b) are corresponding to those shown in Fig. S 23(a)(b), respectively.

**Full range of** **RMSE RUL and *s* and the amount of sampling in training set**

Testing RMSE diagrams for RUL and in the full range by Full RUL DNN are shown in Fig. S 25. It can be observed that most of regions of RMSE of RUL were less than 40. It is also true for RMSE of when . However, RMSE of dramatically increased around the 800th, 900th, and 1000th—1500th cycles. This is because the significant reduction of amount of sampling of testing strategy <3> corresponding to the batteries with cycles in training dataset. Fig. S 26 shows that the number of batteries can be used for the given value of *s*. The amount of convolutional sampling throughout the entire dataset is in total was 53027, while that with was 9166, i.e. 17.3% of the entire dataset. Thus, insufficient information can be learned by Full RUL DNN.



Fig. S 25: Testing RMSE of predicted (a) RUL and (b) *s*, which can be regarded as the current battery age, in full range from cycle 1--1500 by Full RUL DNN.



Fig. S 26: The number of batteries can be used for the given value of *s*, where the area represents the number of training sample.

**The specification of the current DNNs**

The specifications of the Discharge DNN, Full DNN and RUL DNN were shown in Table S 2…Table S 4, respectively. Where the DNNs with an input data length of 60, 80, and 100 cycles for testing purposes mentioned in Fig. S 18 were also listed. The numbers were estimated by TensorFlow (v2.1.0) and Keras (v2.3.1). It can be observed that the number of parameters in Discharge DNN were around a million for the prediction of single value properties and 4 million for time-series curves. The computation requirements, referred to as the number of floating-point operation (FLOPs), were less than 40 million. Feature selectors (i.e. Dimension reduction in the manuscript) in Full DNN required more the number of parameters (around 3 million) and the three “student networks” in DML required less than a million parameters. It can be concluded that the current DNNs are very efficient and effective. Our DNNs can be run in real-time in typical hand-held devices

Table S 2: The specification of the Discharge DNN.

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Objective** | **Parameters (M)** | **FLOPs (M)** |
| Feature selector | EoL/RUL (Discharge) | 1.3 | 4.1 |
| Charge time at EoL (Discharge) | 0.7 | 6.3 |
| Predictor | EoL & Charge time at EoL (60 cycle) | 1.3 | 12.6 |
| EoL & Charge time at EoL (80 cycle) | 1.3 | 25.2 |
| EoL & Charge time at EoL (100 cycle) | 1.3 | 37.8 |
|  | 4.2 | 32.3 |

Table S 3: The specification of the Full DNN.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component** | **Objective** | | **Parameters(M)** | **FLOPs(M)** |
| Feature selector | EoL (Discharge) | | 3.1 | 11.7 |
| Charge time at EoL (Discharge) | | 3.4 | 20.6 |
| EoL (Charge) | | 2.1 | 29.5 |
| Charge time at EoL (Charge) | | 0.8 | 32.9 |
| Predictor | Integrated  DML Predictor | Student 1 | 0.1 |  |
| Student 2 | 0.2 |
| Student 3 | 0.4 |
| Total | 0.8 | 6 |
|  | | 1.3 | 21.2 |

Table S 4: The specification of the Full RUL DNN.

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Objective** | **Parameters(M)** | **FLOPs(M)** |
| Feature selector | Discharging part (1&2&5) | 5.9 | 33.5 |
| Charging part (3&4&6) | 5.9 | 23.8 |
| Predictor | RUL & () | 2.7 | 37.9 |

**The data of each cycle of the battery contains sufficient amount of information and the power of proposed last padding technique**

In our work, the proposed DNNs determined and take the advantage of lower-fidelity version of the target, i.e. our data-driven features, to show higher sensitivity to the variation of the principal components/axes of the reduced latent feature space of APR18650M1A batteries than human-picked features. This reduces the complexity degree of space, and thus, we can extract high influence factors from one single cycle.

Here we proposed two statements to show the first cycle can provide sufficient information for the prediction of EoL. Firstly, we found that sufficiently high precision (resolution) of measurement can provide many degrees of freedom to describe battery status comprehensively. It allows the “battery genome” embedded in time-series properties in a single cycle. Secondly, our data augmentation (last padding technique) is the key approach to enable a great flexibility of *a*, and to let the same value of targets (such as EoL) being described by different lengths of meaningful input data during the training process, leading to an efficient learning for the DNN. Each of the statements will be demonstrated by a simple toy testing model in the follows.

For the first statement, we adopted the model of natural language process to demonstrate how the battery behavior can be encoded in the curves with sufficiently high precision of measurement. The algorithm is shown in Algorithm R 1. Here the K-means[2] Clustering method is used to group each timestep *k* in standardized capacity, voltage, current, and temperature curves, , of battery *i* during the *j*-th discharge half-cycle by the given number of cluster *N* into a sequence . Where *N* serves as the number of available vocabularies of the current version of battery language, and can also be regarded as the level of precision (resolution) of the data measurement. Since there is no physical meaning between the clusters in , we next applied positional encoding method[3][4], a typical nature language process (NLP) used with transformers to tokenize group ID for each time step based on its frequency and position of occurrence, giving . These tokenized sequences, i.e. "battery language," were then fed into ALBERT[5], which is the complex unsupervised neural network consisting of 223 million parameters trained by dataset of multi-languages.

Algorithm R 1: Testing model 1 to obtain the results in Fig. S 27(a).

|  |
| --- |
| **Require**: Battery ; cycle number; time step *.*  **Require**: : Standardized Capacity, Voltage, Current, Temperature.  **Require**: : Vocabulary size.  with vocabulary size .    **Return** |

Next, Manhattan distance is used to measure the difference between the activation values outputted by ALBERT based on different *N* corresponding to the first and the 100th cycles. Fig. S 27(a) shows that, the resulting Manhattan distance (normalized with the reference of the case with *N=*100) increased with the vocabulary size *N*. This indicated that it is possible that the tokenized sequences of the two cycles possessing the features of aging effect of the considered battery can be successfully distinguished by ALBERT with sufficiently high level of precision of the data measurement.

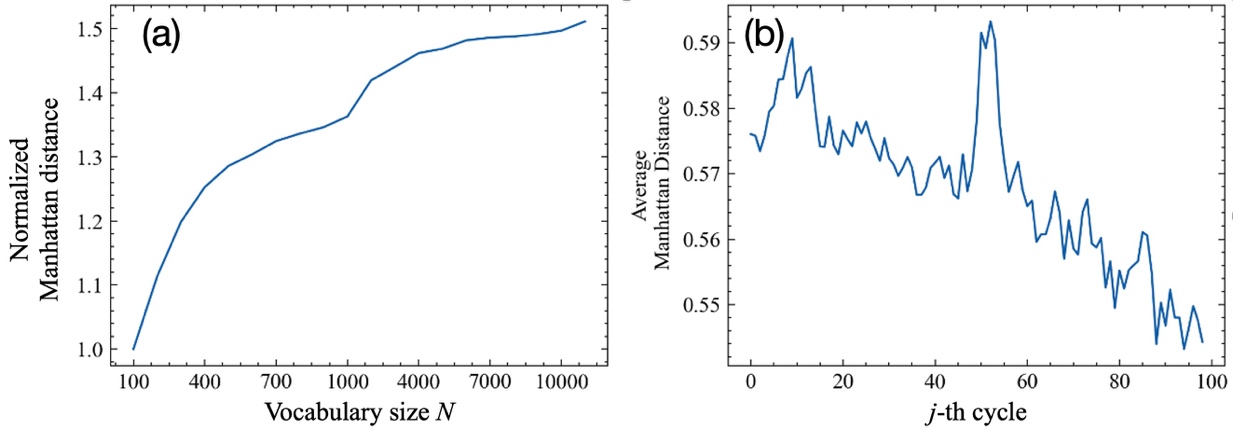


Fig. S 27: (a) Normalized Manhattan distance vs. vocabulary size *N*. (b) The averaged Manhattan distance throughout different tested number of vocabulary size *N* between the tokenized sequences of the *j*-th () and the 100th cycles.

Next, we illustrate the averaged Manhattan distance throughout different tested number of vocabulary size *N* between the tokenized sequences of the *j*-th () and the 100th cycles by

where is the symbol of average. The results are shown in Fig. S 27(b). It can be observed that ALBERT seems to find a one-to-one mapping value corresponding to the distinct cycle number, and the relationship between each cycle can also be revealed. Thus, it implies that it is possible to accurately predict the future properties of batteries based on single cycle by a properly designed DNN. It is also interesting that there is an unusual peak around the 50th to 55th cycles, where anomaly temperature fluctuation occurred during the measurement as mentioned in Severson et al., showing that the mapping values can successfully describe the behavior of the batteries.

For the second statement, we will demonstrate that our last padding technique is the key approach to make the inference based on single cycle possible. In the literature, most of the training approaches is end-to-end, i.e. a set of specific feature correlate to a set of target value. Thus, in the case of battery EoL prediction, it becomes extremely difficult when only single cycle is considered. Different from the end-to-end approach, “multi-ends-to-end” is used in the current work, i.e. multiple sets of feature correlate to the same set of target value, where the multiple sets can be augmentable.

We trained the Discharge DNN with *a*=1, *a*=2, *a*=3, … *a*=100 all together for each selected battery. Where the cases of *a*<100 adopt the last-padding technique to repeat the information of the last cycle to fully fill the fixed-width of 100 data block. For example, the training input of the case of *a*=1 contains the features of cycle 1 repeated 100 times; the training input of the case of *a*=2 contains the features of cycle 1 and those of cycle 2 repeated 99 times. Note that these 100 input blocks correlate with the same EoL of that selected battery, i.e. multi-ends-to-end approach.

It is proved by Severson et al. that the input block of *a*=100 contains sufficient information to accurately predict EoL, and of course, this task is easier than it of *a*=99. Thus, input of *a*=100 can contribute more adequate gradients for modifying the weights of the DNN during the training process. Based on the guidance provided by the input of *a*=100, the DNN then become more experienced to find the correlation between the input of *a*=99 and the same EoL, and so on so forth. With this properly designed training strategy, the DNN can gradually learn how to predict EoL by the input of *a*=1.

The above-mentioned learning process is very similar to it of U-Net. During the training, the DNN temps to start its update of weights based on the input with the highest correlation to the targets, then based on those with lower correlation (i.e. the more difficult task with the less cycle information). Thus, when the DNN performed the inference for the difficult tasks, it can either be based on the parameters obtained from the learning process of the easier tasks, or be based on its own independent gateway to make the prediction for the target.

To compare the effect between the traditional end-to-end and our multi-ends-to-end approaches, we applied both approaches to replace the regression procedure done by gradient descent method on a problem of linear regression. We first generated 1000 lines in *x*, *y* space which are in the form of , where *w* and *b* are uniformly randomly chosen in the range . Each of these lines was described by five points where and the corresponding *y*. Then, gradient descent method was applied to fit each line with the same initial guess of based on the five points. We recorded (), where , and () is expected to be the result closest to the answer (). Fig. S 28 shows the error of *w* and *b* are dramatically altered between the iterations, showing that this task is not as easy as we expected if gradient descent method is used. After the data preparation, we adopted three models, (1) ANN with end-to-end approach, (2) ANN with our approach, (3) CNN with our approach, to learn how to predict () based on (). Where the three models have similar complexity of network with the number of parameters around 105.



Fig. S 28: The error of *w* and *b* of 1000 lines in the *j*-th iteration. Where the black dots show the path of the convergence of one of the lines, showing the error dramatically altered between the iterations

The first model directly used all the sets of and () as input and output to train the network. Where the structure of [2,128,128,2] unit of dense lays was used, where batch normalization[6] was applied to each layer. The second model adopted the structure of [20,128,128,2] and was trained by the proposed last padding technique. Where 20 are 10 for *w* and 10 for *b*. For example, if only the first step is used, then they will be repeated for another 9 times to fill the input matrix; if nine steps are used, and then will be repeated once. Note that although we used all the ten steps to train our second model, its inference of () during the testing is still based on the first step The third model was very similar to the second one and adopted convolution layers (CNN). The performance of the three models on the testing sets is shown in Fig. S 29. It can be observed our approach, Model (2) and (3), enables excellent convergence on both the progress and results. Also, the relationship between each time step in the entire time series can be well captured by convolution layers in Model (3).

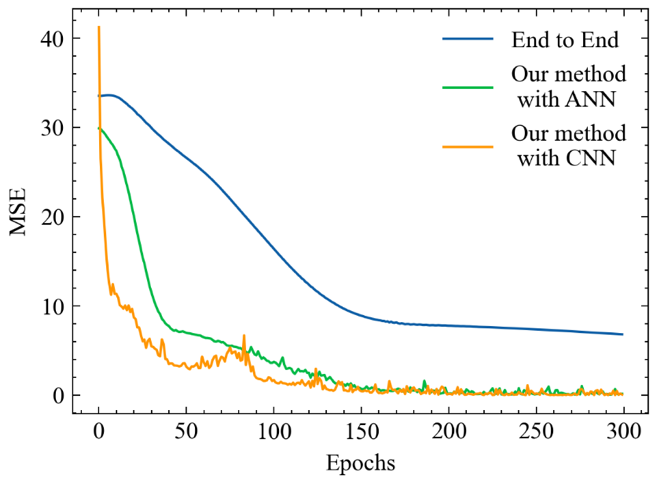


Fig. S 29: The performance of the three models on the testing sets. Note that performance of models evaluated by

In order to demonstrate the behavior of learning, we plot the gradient summation contributed by the input matrix containing the information of in each training batch, as shown in Fig. S 30. It can be observed that the input matrices containing more meaningful information (i.e. greater *j*) contributed more and those containing less meaningful information (i.e. less *j*) in the early stage of training (before around the 100th training batch). However, in the later stage of the training, the model has completely learned the features provided by the input matrices with greater *j*, and started to focus on the learning about the input matrices with less *j*. With the two demos mentioned above, we show that (1) the data of each cycle of the battery contains sufficient amount of information to be distinguished, provided that the level of precision/resolution of the measurement is sufficiently high; (2) the proposed last padding technique enables an efficient and effective learning and make the most of the dataset compared with the traditional end-to-end approach.

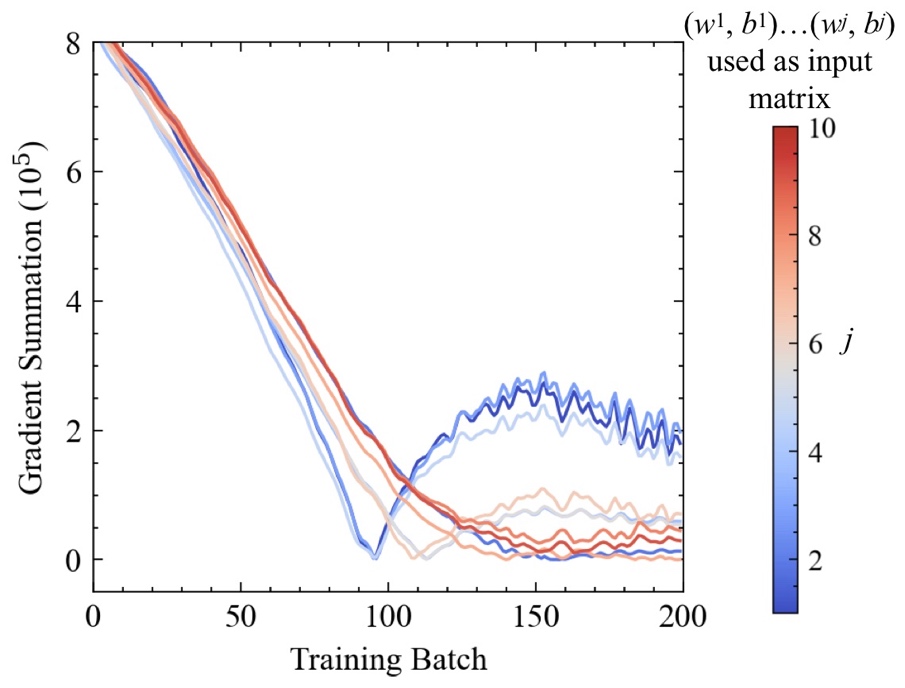


Fig. S 30: The gradient summation contributed by the *j*-th input matrix in each training batch.

**Reference in Supplementary**

[1] XiongR. Battery management algorithm for electric vehicles. 2019. https://doi.org/10.1007/978-981-15-0248-4.

[2] ArthurD, VassilvitskiiS. K-means++: The advantages of careful seeding. Proc. Annu. ACM-SIAM Symp. Discret. Algorithms, vol. 07-09-January-2007, 2007.

[3] VaswaniA, ShazeerN, ParmarN, UszkoreitJ, JonesL, GomezAN, et al. Attention is all you need. Adv. Neural Inf. Process. Syst., vol. 2017- December, 2017.

[4] DevlinJ, ChangMW, LeeK, ToutanovaK. BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., vol. 1, 2019.

[5] LanZ, ChenM, GoodmanS, GimpelK, SharmaP, SoricutR. Albert: A lite bert for self-supervised learning of language representations. ArXiv 2019.

[6] IoffeS, SzegedyC. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 32nd Int. Conf. Mach. Learn. ICML 2015, vol. 1, 2015.