Supplementary Information for

A Versatile ONN Computing Platform through Optically-Collaborative Algorithms

Yuanjia Wang[1], Xin Dong[1], Pujing Cheng[1,3], Yi Zhou[1,2,*], and Kenneth K. Y. Wong[1,4,*]

[1]Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong, China

[2]College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha, China

[3]Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, China

[4]Advanced Biomedical Instrumentation Centre, Hong Kong Science Park, Shatin, New Territories, Hong Kong, China

[*] Corresponding author

Email: yizhou1508@163.com, kywong@eee.hku.hk

S1. Comparison of training progresses with BONN and conventional ENNs

In our BONN framework, the ONN models are trained on-site instead of simulations and transferring. A comparison of training progress between the conventional ENNs and the BONN is illustrated as Fig. S1 (a).

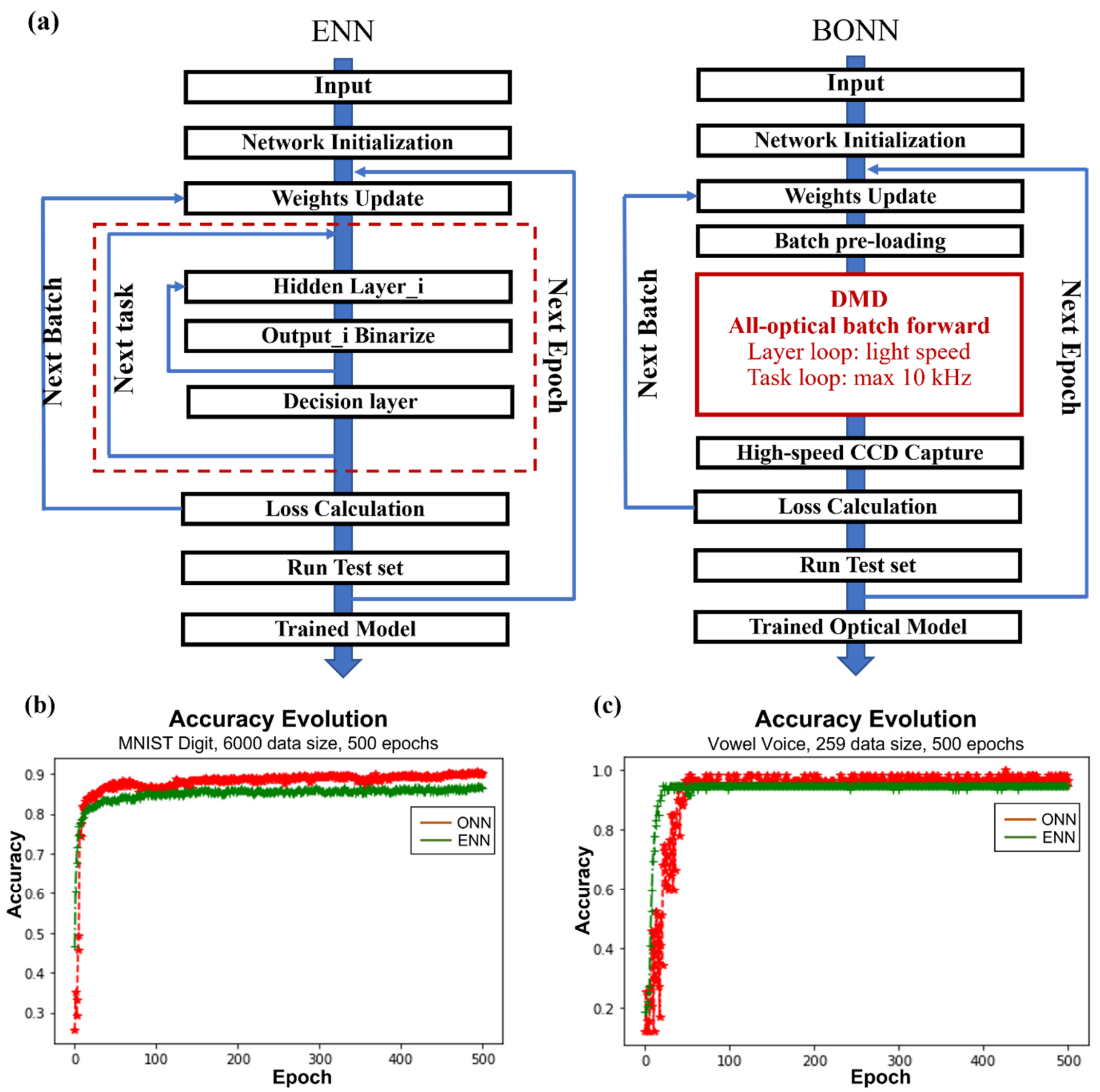

Fig. S1. Training progress comparison between conventional ENNs and BONN framework. (a) Flow charts of the training progresses, the training of BONN is realized directly with physically implemented systems, without transferring from simulations results. (b) Accuracy evolution comparison on MNIST digit dataset, maximum accuracy: ENN 89.33%, BONN 91.50%. (c) Accuracy evolution comparison on Vowel voice dataset, maximum accuracy: ENN 94.92%, BONN 100%.

In the training progress of conventional ENNs, the overall dataset is divided into one training-set and one testing-set. The training-set is further divided into several batches based on the specified batch size, and the network parameters are updated after each batch. When training the network, there are 4 loops in total: **1)** the forward computing path of each task through defined network layers; **2)** the conductions of all tasks in each batch; **3)** the parameter updates with all batches in each epoch; and **4)** the network progressing through epochs. In our evaluations, the dataset size is 6000 with 0.8 train/test ratio, 50 batch size, and 500 epochs. The loss function is cross-entropy and the optimizer is stochastic gradient descent (SGD).

In the training progress of BONN, by conducting loop **1)** and **2)** on the optical system, the majority of the forward computing loads are offloaded from electronic devices. Considering the training progress in each batch, the overall optical forwarding floating-point operations (FLOPs) of the batch is (3-layer FC, 264×320 layer-pixels):

$$n_{batch} \times n_{layer}(2C_{in}C_{out} + C_{activation})$$

$$= 50 \times (3 \times (2 \times (264 \times 320)^2 + 264 \times 320) \approx 2.14\, TFLOPs \quad (1)$$

In each task, the decision layer remains on electronic devices, and the FLOPs of which in each batch is calculated to be:

$$n_{batch} \times 2C_{in}C_{out} = 50 \times \left(2 \times (264 \times 320 \times 10)\right) = 84.48\, MFLOPs \quad (2)$$

Therefore, 99.997% of the overall forwarding path computing are offloaded onto the optical system. Meanwhile, it's worthwhile noting that each layer loop is conducted at the speed of light, and the speed of each batch loop is only determined by the frame rate of the DMD (maximum 10 kHz). With all batch tasks preloaded onto the DMD, no further optical-electronic interaction during each batch is required.

For back-propagation (BP) progress, although the backward-forward FLOPs ratio is 1:1 for FCNNs, it is not further required when using trained models, hence the influences in applications can be considered to be minor.

The accuracy evolution when training the models for MNIST digit and Vowel voice are shown as Fig. S1 (b) and (c). It can be observed that while experiencing similar evolution paths, the overall performance of BONN is better than conventional same-structured ENNs throughout the training progresses.

S2. Experimental BONN system implementation

The physically implemented BONN system for experimental evaluations is presented as Fig. S2, with detailed descriptions included in Section 2.2 of the manuscript.

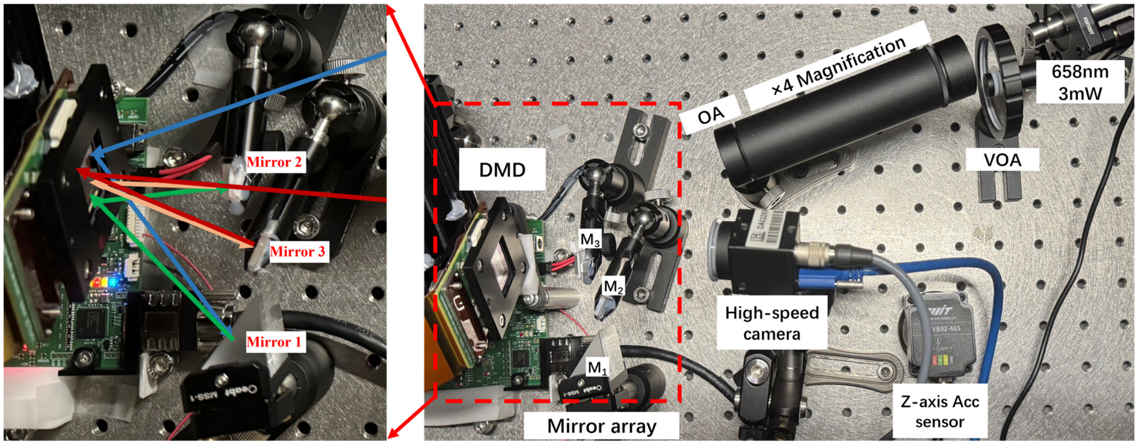

Fig. S2. The physical BONN system implementation for evaluations, VOA: variable optical attenuator, OA: optical aperture, L: lens, M: mirror, In: input layer, h: hidden layer.

In the experimental setup, the core part is the realization of 4 reflections with the mirror array. To guarantee the precise reflection areas on the DMD, calibrations are required for precise forwarding paths. With the self-developed programmable calibration tool, the calibration progress is illustrated as Fig. S3.

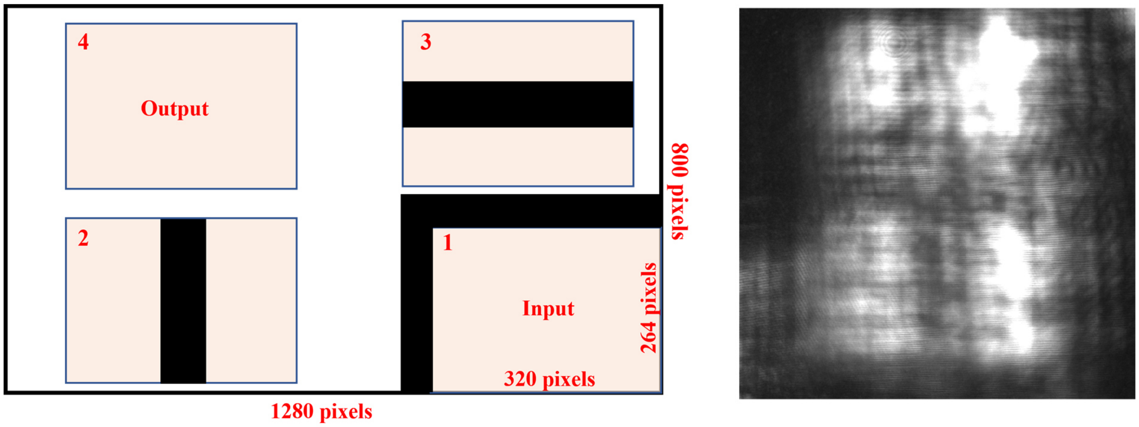

Fig. S3. Calibration of the DMD reflection path and an example of the correlated pattern.

First, in order to secure the pattern shape, the blank input pattern area is placed at one corner of the DMD illumination area, and a surrounding mask is further placed to ensure no additional interferences are input to the network. Next, by placing one vertical mask and one horizontal mask in the second and third reflection, the positions of each reflection can be observed with a cross on the final captured output from the fourth area. After careful adjustments of area boundaries, an example of the calibrated pattern is achieved, in which the effects of diffractive power-law nonlinearity can also be observed. Once calibrated, it has been proved that no additional calibration actions are required unless systemic damage is caused. The robustness of the system in long-term tasks and under environmental impacts have also been evaluated to be reliable, which is included in Section 2.2.3 in the manuscript.

Additionally, an error and exception handling mechanism are also implemented in the system. For most of the unexpected errors occurred during task conductions, such as data format exception, out of memory, and device not reachable, etc., the BONN computing platform will handle the errors and reset the system without any manual operations, further enabling our BONN as an independent and effective versatile computing platform.

S3. Dataflow of parallel tasks with GPT-oriented task scheduler

As mentioned in the manuscript Section 2.2.2, our BONN-driven versatile GPT-oriented image task scheduler supports multiple parallel task options for higher network capacity and conduction speed. Specifically, for the evaluation between binarization algorithms on MNIST fashion, the hybrid mode (parallel on electronic devices, serial on BONN due to limitations of equipment) is selected for both massively-parallel potential demonstration and environmental influences cancellation. The overall dataflow chart of the hybrid mode is presented as Fig. S4.

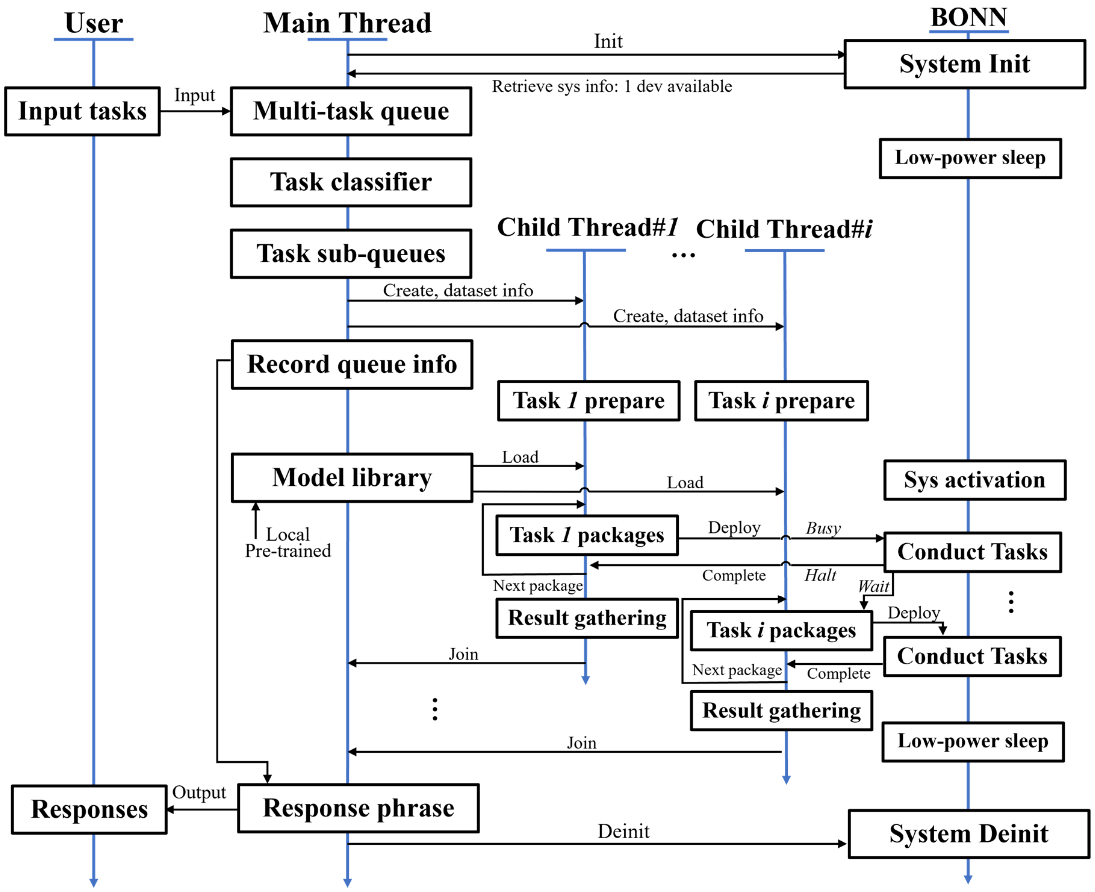

Fig. S4. The dataflow chart of the GPT-oriented image task scheduler on hybrid mode. The character `User` denotes to the operations of the application users, the character `Main Thread` denotes to the system application front-end interface, and `BONN` is our implemented optical system as black-box computing platform.

When initializing the system, an available resource list will be created based on the device information retrieved from BONN hardware (1 BONN available in this case), which will be referred when determining the system operation mode. After all hardware initializations are finished, the task scheduler is ready to use.

When used for imaging tasks with trained models, only image inputs from the users are required without any additional information (if the image is a digit or fashion object, etc.), and the images can be from any pre-trained datasets or in any queue orders. A first-stage task classifier is applied to the input queue, dividing and forming sub-task queues based on different datasets. The original information of the tasks will be stored locally for later queue reconstruct and response phrase.

Next, based on the resource availability, parallel conduction modes are decided, which is hybrid mode in this case with sufficient electronic resource and single BONN platform. For each sub-task queue, a child-thread is created for potentially time-consuming task data preparations. With parallel operations at this stage, the system speed can be increased. Based on different datasets, pre-trained models are retrieved from the local model library, and task packages are constructed to be deployed on BONN.

When BONN is `Halt` without tasks, child-threads can deploy tasks to BONN for conductions, while marking the statues of BONN to `Busy`. The BONN will respond `Wait` to other child-thread deploy requests until current task complete and mark its status back to `Halt`. While BONN starts to conduct other tasks, task-finished child-threads can start to gather results and join back to the main-thread, enabling over-laps between tasks and further reduce time-consumptions.

After all tasks are completed, the main thread can then phrase a summary based on the conduction results and recorded queue information and notify users the final response.

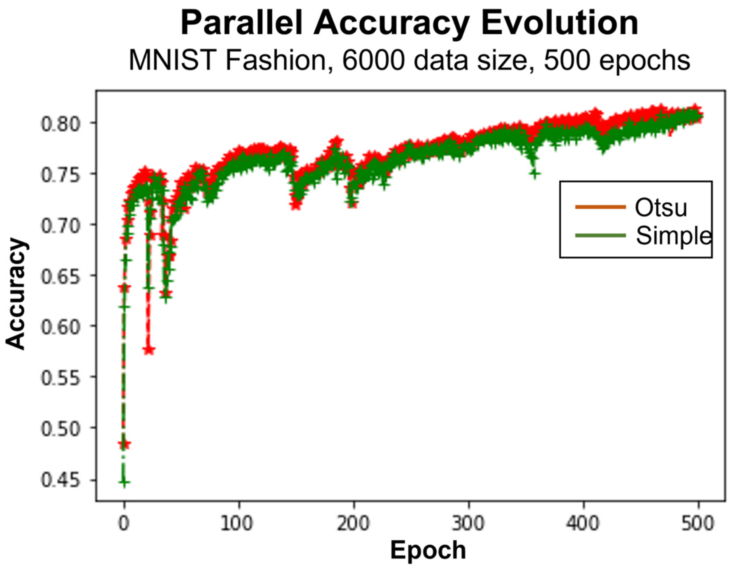

Fig. S5. Binarization algorithm comparison on MNIST fashion.

For multi-model training tasks, the overall dataflow is similar, with the parameter update of each model conducted by turns. In this way, the environmental influences can be unified and focusing the effectiveness of algorithms when comparing. With Otsu algorithm and simple-margin algorithm, an accuracy evolution comparison of binarization algorithms on MNIST fashion is shown as Fig. S5. It can be observed that a constantly higher accuracy is achieved using Otsu algorithm, further supporting the point in the manuscript. Meanwhile, although with environmental interferences (all evaluation results of the system are obtained at ordinary laboratory environments), the overall progression of the accuracy is steady and a potential tendency of increased accuracy can also be expected after 500 epochs, further indicating the reliability of BONN.

S4. Overfitting evaluation under poor PSNR condition

In section 2.2.3 of the manuscript, both noise robustness and long-term task reliability of BONN are verified. Another risk under poor PSNR condition is overfitting, with the model attempts to fit noise elements rather than original objects. Here, accuracy-loss consistence is also evaluated under 10 dB SNR, with results presented in Fig. S6.

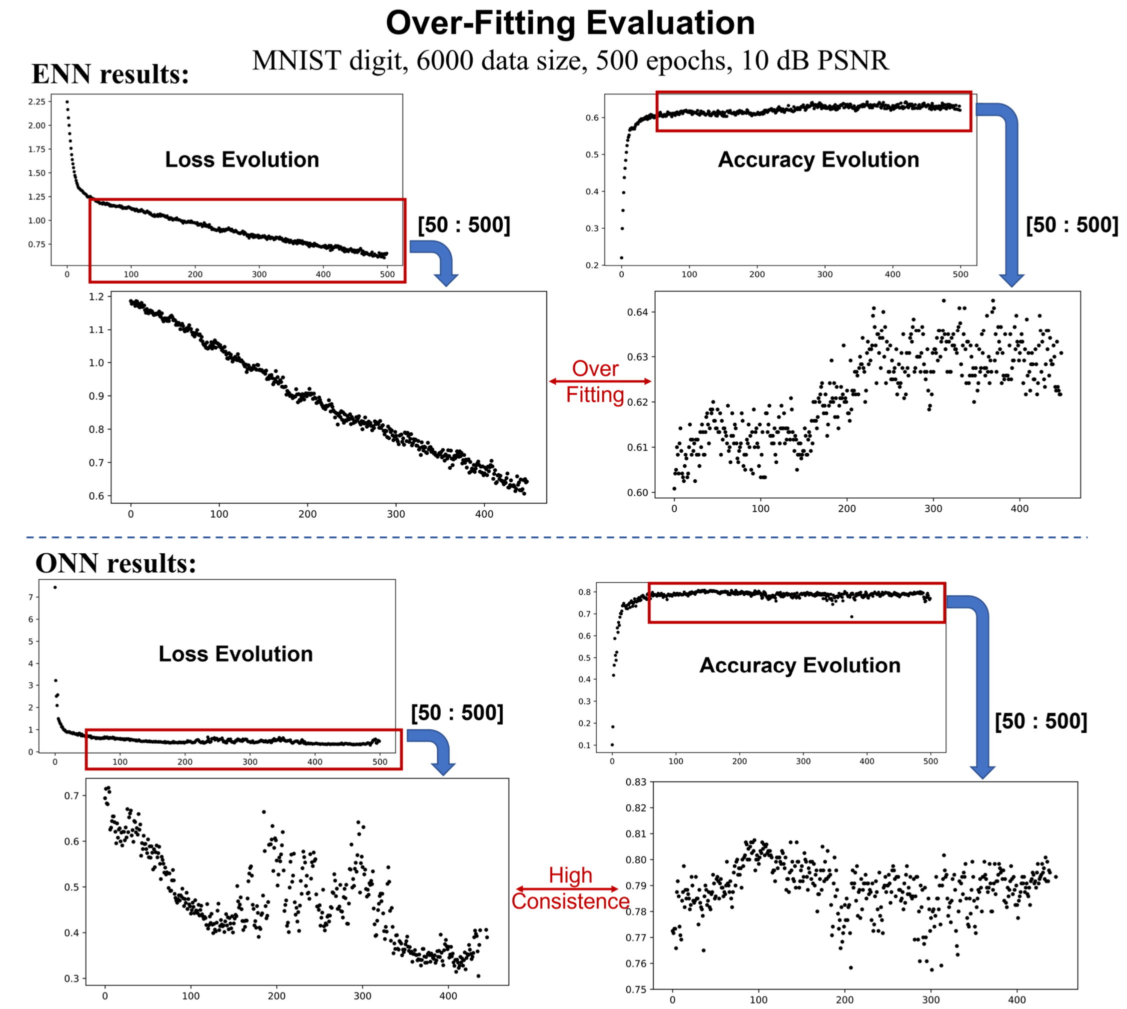

Fig. S6. A comparison of fine-tuning stages between same-structured ENN and BONN under 10 dB PSNR condition.

It can be observed that, at the fine-tuning stage of ENN (after 50 epochs), although the loss (cross-entropy) keeps decreasing, the accuracy has not been improved along with the dropping of loss, indicating overfitting as epoch progresses. On the other hand, a high consistency is observed for BONN results despite of fine-tuning fluctuations, and both higher accuracy and lower loss are achieved compared with ENN results. This further indicates that, apart from enhanced robustness, our BONN also demonstrates potential for higher overfitting resistance.