A APPENDIX

A.1 PARAMETERS AND CONSTRAINTS

Type	Index	Hardware Parameters	Valid Range	Meaning
PE	H1	PE mesh-X	Factors of # PEs	Decide the arrangement of the 2-D PE array.
	H2	PE mesh-Y	Factors of # PEs	
Local buffer	H3	Input entries in Local buffer	0 to # local buffer entries	Decide the partition of local buffer. The partition leads to sub-buffers with inflexible sizes. This is useful as the latency to access each smaller sub-buffer decreases.
	H4	weights entries in Local buffer	0 to # local buffer entries	
	H5	outputs entries in Local buffer	0 to # local buffer entries	
Global buffer	H6	Global buffer instances	Factors of #PEs	Determine the arrangement of global buffer, and its connection between global buffer and per PE's local buffer (Local buffer of PEs along the X-axis shares the instances of global buffer along the X-axis).
	H7	Global buffer mesh-X	Factors of PE-mesh-X	
	H8	Global buffer mesh-Y	Factors of PE-mesh-Y	
	H9	Global buffer block size	Factors of 16	Determines the width of a global buffer entry
	H10	Global buffer cluster size	Factors of 16	Determines of the number of wider structures where multiple entries are ganged into
Dataflow	H11	Dataflow option of filter width	1, 2	Options that determine the size of filter width in PE's local buffer
	H12	Dataflow option of filter height	1, 2	Options that determine the size of filter height in PE's local buffer

Figure 6: Hardware parameters.

Type	Hardware Constraints
PE	PE mesh-X (H1) * PE mesh-Y (H2) = # PEs
Local buffer	The sum of local sub-buffers (H3, H4, H5) does not exceed buffer size
Global buffer	Global buffer mesh-X (H7) * global buffer mesh-Y (H8) = # Global buffer instances (9)
Local buffer & global buffer (unknown)	A valid software mapping exists depending mainly on local buffer partition (H3, H4, H5) and global buffer arrangement (H6, H7, H8)

Figure 7: Hardware constraints.

Type	Index	Software Parameters	Valid Range	Meaning
Loop blocking and degree of parallelism	S1	Blocking factors of R	Factors of R	Determines the size (parallelism) of each type of data (inputs, weights and outputs) in each storage layer (except those that are in the hardware dataflow).
	S2	Blocking factors of S	Factors of S	
	S3	Blocking factors of P	Factors of P	
	S4	Blocking factors of Q	Factors of Q	
	S5	Blocking factors of C	Factors of C	
	S6	Blocking factors of K	Factors of K	
Loop reorder	S7	Loop order in local buffer	Permutations of non-1 factors	Affects the reuse of each type of data (inputs, weights and outputs) in each storage layer.
	S8	Loop order in global buffer	Permutations of non-1 factors	
	S9	Loop order in DRAM	Permutations of non-1 factors	

Figure 8: Software parameters.

B HYPERPARAMTERS FOR BO

In Figure 10 we report the hyperparamters for BO.

Type	Software Constraints
Loop blocking and degree of parallelism	Product of all blocking factors of R (S1) equals R of the target neural layer
	Product of all blocking factors of S (S2) equals S of the target neural layer
	Product of all blocking factors of P (S3) equals P of the target neural layer
	Product of all blocking factors of Q (S4) equals Q of the target neural layer
	Product of all blocking factors of C (S5) equals C of the target neural layer
	Product of all blocking factors of K (S6) equals K of the target neural layer
Buffer capacity (local)	Inputs/weights/outputs sizes (S1-S6) cannot exceed corresponding local sub-buffer capacity
Buffer capacity (global)	Size of all types of data (S1-S6) does not exceed global buffer capacity
Parallelism	Product of blocking factors in global buffer X-axis (S1-S6) cannot exceed # PEs in X-axis
	Product of blocking factors in global buffer (S1-S6) cannot exceed total # PEs

Figure 9: Software constraints.

number of independent trials	5 (HW), 10 (SW)
number of random data points	50 (HW), 150 (SW)
number of warmup data points	5 (HW), 30 (SW)
number of samples for EI	1000
lambda for LCB	1.0

Figure 10: Hyperparamters for BO.

C Neural Model Specifications.

In Figure 11 and Figure 12 we report the specifications of neural models benchmarked in this paper.

D Paramterization of 2D Convolution

Listing 14 gives the seven-level nested loop that comprises a 2D convolution.

Figure 17 shows a design point for the CONV4 layer of ResNet. The architecture components are again the same as in the 1D example, but since the memory footprint is significantly larger, the PE can no longer capture all data reuse, so the Global Buffer must store large portions of the inputs and outputs.

E Example Parameter Vector

Below are example vectors of hardware and software parameters our BO optimizes.

F Additional results

F.1 Software optimization

In Figure 18 we show more examples of the software optimization over multiple layers of the different architectures. Our Bayesian optimization formulation consistently outperforms the baselines (Chen et al., 2018).

F.2 Ablations

In Figure 19 we compare different surrogate models and acquisition functions for Bayesian optimization of the software mapping. We found Gaussian processes with LCB to consistently outperform other alternatives.

Model	Layers	Specifications
ResNet	ResNet-K1	Filter size: 3×3 Output size: 56×56 # input channel: 64 # output channel: 64 Stride: 2
	ResNet-K2	Filter size: 3×3 Output size: 28×28 # input channel: 128 # output channel: 128 Stride: 1
	ResNet-K3	Filter size: 3×3 Output size: 14×14 # input channel: 256 # output channel: 256 Stride: 1
	ResNet-K4	Filter size: 3×3 Output size: 7×7 # input channel: 512 # output channel: 512 Stride: 1
DQN	DQN-K1	Filter size: 8×8 Output size: 20×20 # input channel: 4 # output channel: 16 Stride: 4
	DQN-K2	Filter size: 4×4 Output size: 9×9 # input channel: 16 # output channel: 32 Stride: 2

Figure 11: Specifications of ResNet (ResNet-18) (He et al., 2016) and DQN (Mnih et al., 2013)

In Figure 20 we investigate the robustness of LCB for software optimization using different values of λ. We found that $\lambda = 0.1$ tends to be too greedy, but that above $\lambda = 0.5$, LCB tends to be fairly robust.

Model	Layers	Specifications
MLP	MLP-K1	d_{in}: 512 d_{out}: 512
	MLP-K2	d_{in}: 64 d_{out}: 1024
Transformer	Transformer-K1	d_{model} = 512 d_v = 32 d_k = 32 h = 16
	Transformer-K2	d_{model} = 512 d_v = 64 d_k = 64 h = 8
	Transformer-K3	d_{model} = 512 d_v = 128 d_k = 128 h = 4
	Transformer-K4	d_{model} = 512 d_v = 512 d_k = 512 h = 1

Figure 12: Specifications of MLP and Transformer (Vaswani et al., 2017)

Model	Feature name	Description
Hardware	mesh_x_ratio	The ratio of PE array and global buffer along x-axis
	mesh_y_ratio	The ratio of PE array and global buffer along y-axis
Software	input_buffer_usage	input data size / input (local) buffer size
	weight_buffer_usage	weight data size / input (local) buffer size
	output_buffer_usage	output data size / input (local) buffer size
	global_buffer_usage	all data size / global buffer size
	parallelism_ratio_x	used parallelism / available parallelism in the x-axis of global buffer
	parallelism_ratio_y	used parallelism / available parallelism in the y-axis of global buffer

Figure 13: Extra features used by the hardware and software BO optimizers.

```
for n in [0:N)
  for k in [0:K)
    for r in [0:R)
      for s in [0:S)
        for p in [0:P)
          for q in [0:Q)
            for c in [0:C)
              outputs[n][k][q][p] += weights[k][c][s][r] *
                                     inputs[n][c][q+s][p+r]
```

Figure 14: Computing a 2D convolution with a seven-level nested loop.

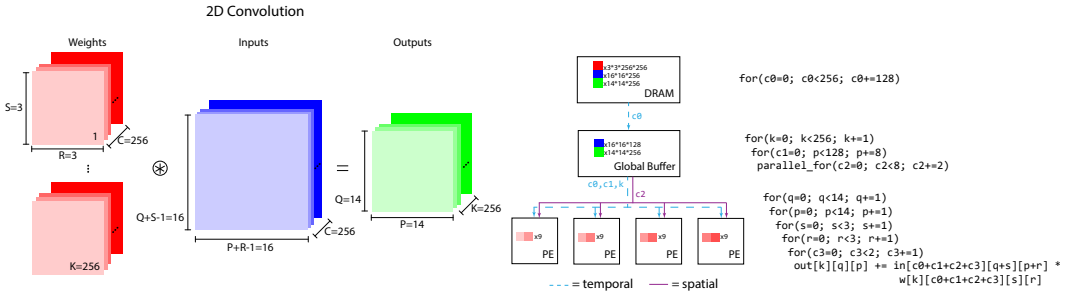


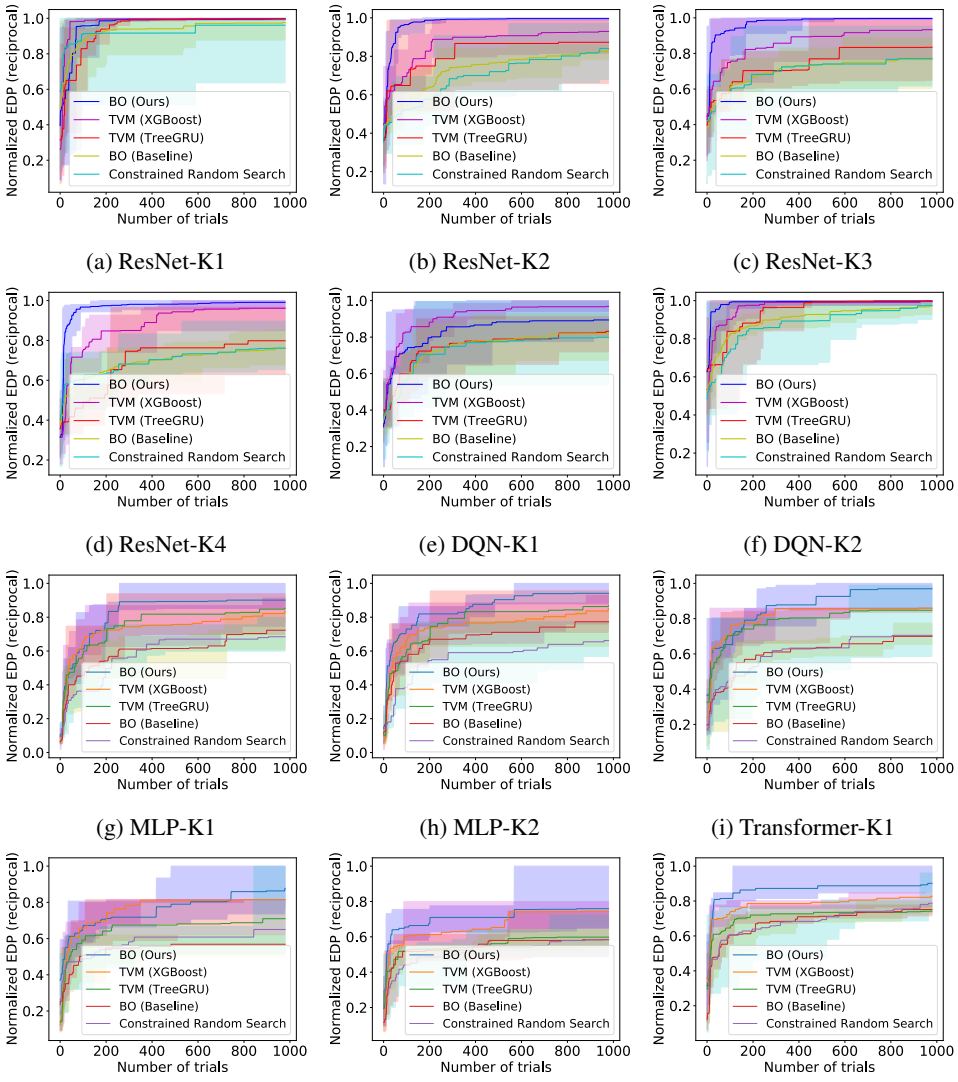
Figure 15: An architecture computing the CONV4 layer of ResNet.

Index	Type	Range of Values
1	int	Factors of 256
2	int	Factors of 256
3	int	0-220 (total local buffer size)
4	int	0-220 (total local buffer size)
5	int	0-220 (total local buffer size)
6	int	Factors of 168
7	int	Factors of H1
8	int	Factors of H2
9	int	Factors of 16
10	int	Factors of 16
11	categorical	0, 1
12	categorical	0, 1

Figure 16: An example vector of hardware parameters. Please refer to Figure 6 for more detailed descriptions.

Index	Type	Range of Values
1-2	int	Factors of 3
3-4	int	Factors of 3
5-6	int	Factors of 28
7-9	int	Factors of 28
10-12	int	Factors of 128
13-17	int	Factors of 128
18	categorical	0-1
19	categorical	0-5
20	categorical	0-1
21	categorical	0-1
22	categorical	0-23

Figure 17: An example vector of software parameters (with ResNet-K2). Please refer to Figure 8 for more detailed descriptions. In this example, parameters 1-17 correspond row-wise to S1-S6 respectively, parameters 18-20 correspond to S7, and parameters 21 and 22 correspond to S8 and S9 respectively.



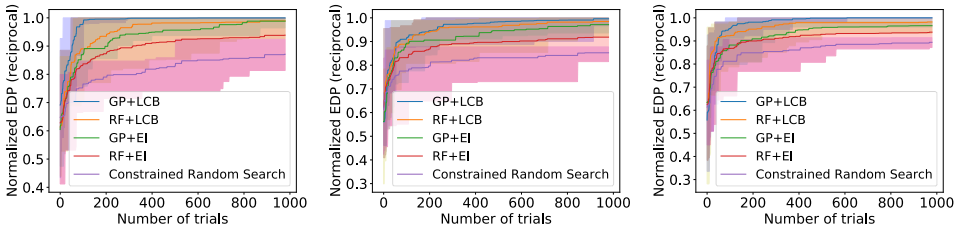
(a) ResNet-K1 (b) ResNet-K2 (c) ResNet-K3

(d) ResNet-K4 (e) DQN-K1 (f) DQN-K2

(g) MLP-K1 (h) MLP-K2 (i) Transformer-K1

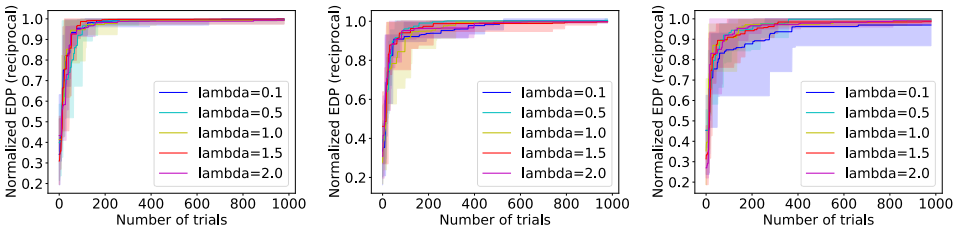
(j) Transformer-K2 (k) Transformer-K3 (l) Transformer-K4

Figure 18: Software mapping optimization on ResNet, DQN, MLP, and Transformer. The Y-axis shows the reciprocal of energy-delay product (EDP) (normalized against the best EDP value). Higher is better.



(a) ResNet-K2 (b) ResNet-K3 (c) ResNet-K4

Figure 19: GP with different surrogate models and acquisition functions.



(a) ResNet-K2 (b) ResNet-K3 (c) ResNet-K4

Figure 20: LCB acquisition function with different lambda values.