nature electronics

ARTICLES

https://doi.org/10.1038/s41928-020-00515-3

Check for updates

Illusion of large on-chip memory by networked computing chips for neural network inference

Robert M. Radway[1]✉, Andrew Bartolo[2], Paul C. Jolly[1], Zainab F. Khan[1], Binh Q. Le[1,3], Pulkit Tandon[1], Tony F. Wu[1,4], Yunfeng Xin[1], Elisa Vianello[5], Pascal Vivet[5], Etienne Nowak[5], H.-S. Philip Wong[1], Mohamed M. Sabry Aly[6], Edith Beigne[4], Mary Wootters[1,2] and Subhasish Mitra[1,2]

Hardware for deep neural network (DNN) inference often suffers from insufficient on-chip memory, thus requiring accesses to separate memory-only chips. Such off-chip memory accesses incur considerable costs in terms of energy and execution time. Fitting entire DNNs in on-chip memory is challenging due, in particular, to the physical size of the technology. Here, we report a DNN inference system—termed Illusion—that consists of networked computing chips, each of which contains a certain minimal amount of local on-chip memory and mechanisms for quick wakeup and shutdown. An eight-chip Illusion system hardware achieves energy and execution times within 3.5% and 2.5%, respectively, of an ideal single chip with no off-chip memory. Illusion is flexible and configurable, achieving near-ideal energy and execution times for a wide variety of DNN types and sizes. Our approach is tailored for on-chip non-volatile memory with resilience to permanent write failures, but is applicable to several memory technologies. Detailed simulations also show that our hardware results could be scaled to 64-chip Illusion systems.

Despite decades of technological advances, it remains difficult to create integrated circuits with large amounts of on-chip memory—memory that is densely connected to processing elements (PEs) on the same chip. Instead, systems typically rely on off-chip memory that is physically separate from the computing chips, and accessing this memory contributes to, and often dominates, the overall energy and execution time[1,2]. This remains a key bottleneck for deep neural network (DNN) inference hardware, particularly as data and model sizes continue to grow (despite the use of sparsity and quantization techniques)[3–9]. On-chip memory capacity is therefore a limiting factor in the energy, execution time and combined energy-delay product (EDP; product of energy and execution time) of today's DNN hardware[10–15]. Moreover, applications often require the use of several DNNs[7,16], further compounding the on-chip memory challenge.

This challenge, known as the 'memory wall', is critical, regardless of the computing architecture. Indeed, embedded microcontrollers, multicore processors, graphics processing units, field-programmable gate arrays (FPGAs) and domain-specific accelerators (including in-memory computing) all face on-chip memory challenges when used for DNNs[1,2,17–25]. Specialized architectures that maximize on-chip data reuse[10,12–14], massive wafer-scale chips[11], dense memory technologies and multiple-bits-per-cell storage[3,17,26–28] and chip stacking methods attempt to address this memory challenge. However, DNNs continue to require higher memory capacity with higher bandwidths and lower energies[2,13,14,27,29].

A hypothetical ideal chip (Fig. 1a) for fast, energy-efficient DNN inference requires that the DNN fit entirely in a large on-chip memory (volatile or non-volatile) that is dense, low-energy and accessible at high bandwidth. Such a chip gives the PEs fast and low-energy access to the data needed for inference, and only DNN inputs and final outputs are communicated externally. The memory wall is thus minimized, leading to both energy and execution time benefits. However, owing to the limitations of current technology (predominantly due to size constraints), this approach cannot be used for existing state-of-the-art DNNs (which can approach trillions of parameters). As DNN model sizes continue to grow, it becomes increasingly difficult to realize an ideal chip, in spite of memory technology advances[6].

In this Article we report a system, which we call Illusion, that consists of a network of multiple computing chips, each with a certain minimal amount of local on-chip memory and mechanisms for quick wakeup and shutdown (that is, the system contains no separate memory-only chips; Fig. 1b). For DNN inference tasks, Illusion performs like an ideal chip, with near-ideal energy, execution time and EDP. In hardware, we demonstrate an Illusion system consisting of eight computing chips, and the energy, execution time and EDP of this eight-chip Illusion system are measured to be within 1.035×, 1.025× and 1.06×, respectively, of the values of the ideal chip (which correspondingly contains eight times more memory than the individual chips used in the demonstration). We demonstrate Illusion's effectiveness for convolutional neural networks (CNNs), long–short-term memory (LSTM) and dynamic deep neural nets (D2NNs). Illusion does not require modifications to the DNNs themselves and provides configurability and flexibility in achieving near-ideal energy, delay and EDP for a wide range of DNN sizes, from those fitting in the memory of a single chip to those requiring the use of the entire memory capacity across all chips in the system.

Our approach can use several on-chip memory technologies, including non-volatile memories (NVMs) such as resistive RAM (RRAM), which allow quick wakeup and shutdown of the individual computing chips through fine-grained temporal power gating. Although volatile static RAM (SRAM) can also be used (where on-chip computing elements are power-gated and volatile SRAM contents are held in retention mode separately), our approach is tailored to on-chip NVM through resilience techniques that

[1]Department of Electrical Engineering, Stanford University, Stanford, CA, USA. [2]Department of Computer Science, Stanford University, Stanford, CA, USA. [3]Department of Electrical Engineering, San Jose State University, San Jose, CA, USA. [4]Facebook, Menlo Park, CA, USA. [5]CEA, LETI, Grenoble, France. [6]School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore. ✉e-mail: radway@stanford.edu

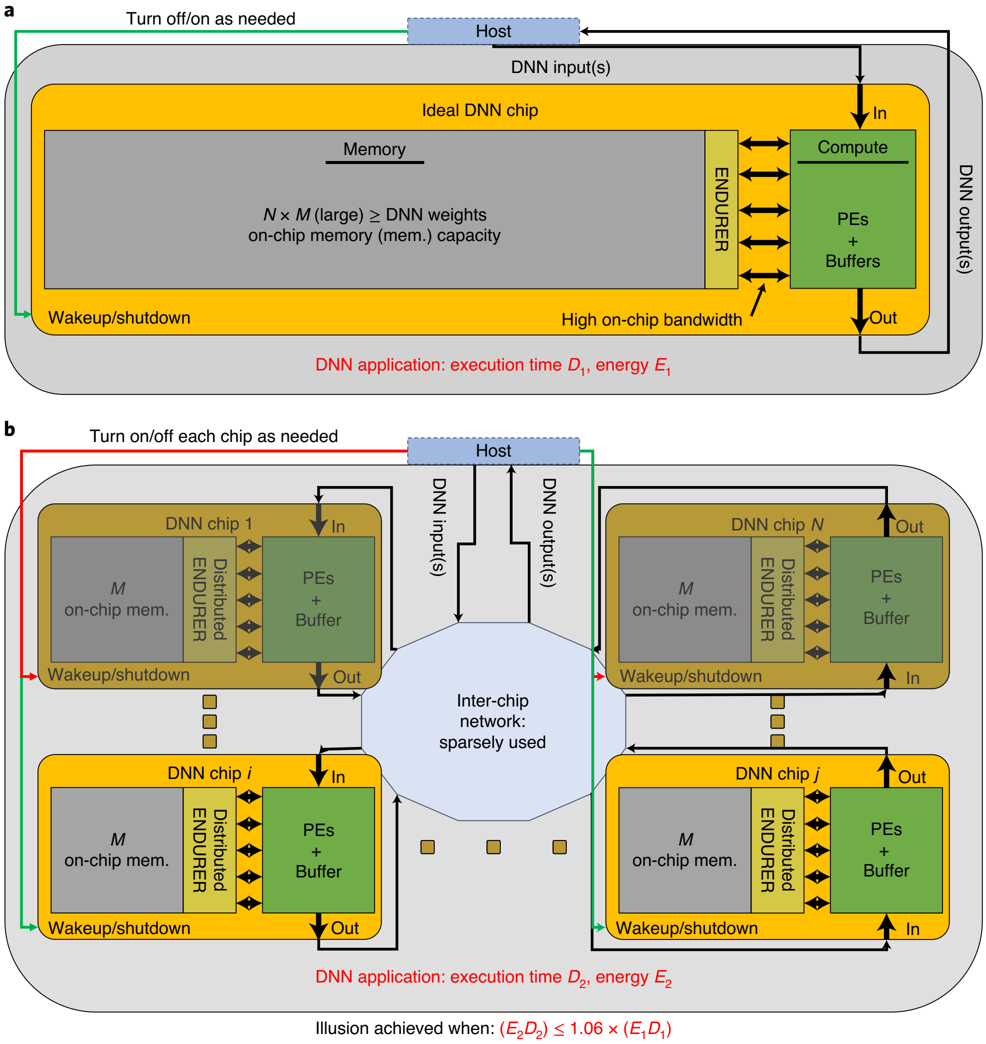


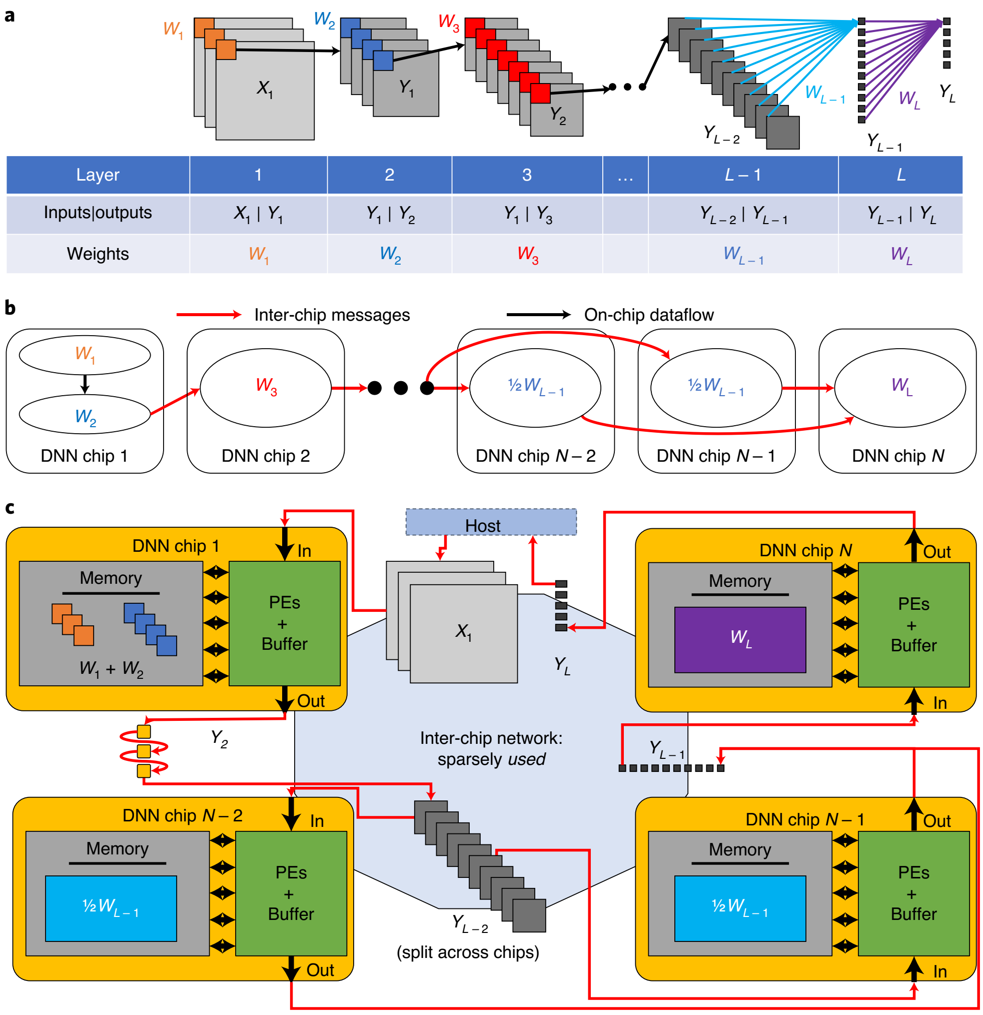
Fig. 1 | An ideal chip and our Illusion system with nearly identical performance. a, A hypothetical ideal chip for fast, yet energy-efficient DNN inference. This requires that the DNN entirely fits in a large on-chip memory accessible by PEs at high bandwidth and low energy. Nearly no off-chip accesses are needed, providing major energy, execution time and energy–delay product (EDP) benefits compared with traditional systems that require off-chip memory. ENDURER[2] provides write endurance resilience if needed by the on-chip memory technology. For a given DNN inference, an ideal chip will have execution time D_1 and consume energy E_1, with an EDP of E_1D_1. **b**, Our Illusion system consists of a network of multiple DNN chips on an inter-chip network, each with a certain minimal amount (labelled *M*) of local on-chip memory and mechanisms for quick wakeup and shutdown. Each chip can access its on-chip memory at high bandwidth and low energy. With appropriate DNN mapping and scheduling to the Illusion system (Figs. 2 and 3), a DNN inference will have execution time D_2 and consume energy E_2, with EDP E_2D_2. Our Illusion system achieves a performance nearly identical (for example, we demonstrate $(E_2D_2) \leq 1.06 \times (E_1D_1)$) to that of the ideal chip, which in total is *N* times larger than the individual Illusion component chips. Like the ideal chip, our Distributed ENDURER provides multi-chip write endurance resilience if needed by the on-chip memory technology. Our inter-chip network is depicted as a ring for simplicity.

overcome technology-specific issues such as the permanent write failures associated with some NVM technologies. (We outline how Illusion is distinct from existing parallelization techniques targeting multi-chip systems in Supplementary Section 2.)

Using detailed simulations of memory- and compute-intensive DNNs (up to gigabytes of weights and thousands of PEs), we show that our hardware results scale for large-scale Illusion systems (up to 64 chips). We also derive additional insights through analytical models for (conservative) estimates of energy, execution time and EDP for Illusion systems. These models are critical to understanding the interplay between on-chip memory capacity and inter-chip network efficiency in an Illusion system. This is particularly important for major technology trends such as advanced 2.5-dimensional (2.5D) integration of chiplets[23,30–32] and ultra-dense (for example, monolithic) 3D integration[1,2,33], which amplify the effectiveness of Illusion.

Illusion system overview

Our hardware chips monolithically and heterogeneously integrate RRAM on commercial silicon complementary metal–oxide–semiconductor (CMOS) technology. We have chosen RRAM because it is a dense, low-energy/latency, on-chip memory technology with demonstrated benefits for applications such as the Internet of Things edge DNN inference[2,27,29]. RRAM allows fine-grained temporal power gating[27] (where compute and memory can be turned on and off rapidly), which is attractive for edge applications and is a key component of our approach. RRAM also provides multiple bits per cell capabilities for further increased density[27]. Compared to other



Layer	1	2	3	…	$L-1$	L
Inputs|outputs	$X_1 \mid Y_1$	$Y_1 \mid Y_2$	$Y_1 \mid Y_3$		$Y_{L-2} \mid Y_{L-1}$	$Y_{L-1} \mid Y_L$
Weights	W_1	W_2	W_3		W_{L-1}	W_L

Fig. 2 | DNN mapping onto our Illusion system for sparse inter-chip messages. a, An example DNN input to the Illusion mapping algorithm. Each layer has input, output and weight tensors as listed. The sum of the weights must fit into the total Illusion system weight memory capacity (that is, $N \times M$ from Fig. 1). **b**, Illusion mapping algorithm example output depicting the weight assignment and corresponding inter-chip data flow. When the Illusion system has sufficient memory capacity per chip, inter-chip messages will be sufficiently small to preserve near-ideal chip energy, execution time and EDP. **c**, Another representation of **b**, depicting the weight assignment onto Illusion's component chips' memories and the input, output and intermediate activation tensor data communicated as messages on the inter-chip network.

NVM-based (such as Flash) computing chips, Illusion's component chips enable DNN inference with 11 times lower energy and a similar execution time[27] (chip details are provided in the Methods).

Illusion consists of three interdependent components (Figs. 2 and 3): the Illusion mapping algorithm, the Illusion scheduling algorithm and the Distributed ENDURER technique. For the Illusion mapping algorithm, we map DNN weights to the component chips in the system during compile time, while ensuring sparse inter-chip messages. For the Illusion scheduling algorithm, from the mapping algorithm output we schedule inter-chip communication and fine-grained (quick) wakeup and shutdown for each chip (that is, create a system schedule). Finally, the Distributed ENDURER technique is designed to overcome RRAM write endurance challenges: that is, the limited number of set-(writing a '1')–reset-(writing a '0') cycles a memory cell can undergo before permanent write failure (stuck at the '1' or '0' state).

Illusion mapping and scheduling

The first step for Illusion system inference is mapping the DNN onto the Illusion system hardware. At compile time, the Illusion mapping algorithm is invoked. Algorithm inputs include the number of chips in the Illusion system, per-chip memory capacity to store weights and the DNN architecture (input, output and weight tensor dimensions and bit width per DNN layer). A key input is the max message count, labelled MM. This message count limit can be computed from the desired energy, execution time or EDP (for example, we desire an Illusion energy that is ≤1.05× that of the ideal chip). The inter-chip network efficiency determines how many messages (that is, MM bytes) equate to 5% of the ideal chip inference energy. Supplementary Section 4 provides more details on derivation based on the inter-chip network characteristics. The Illusion mapping algorithm we provide in Supplementary Fig. 3 is based on two mapping heuristics. We have also formulated a binary integer linear program (BILP, Supplementary Section 1) that produces provably optimal mappings. However, our heuristic-based Illusion mapping algorithm is highly scalable for large Illusion systems.

Our algorithm maps each DNN weight to be stored uniquely on one chip in the Illusion system. Once a weight has been mapped, associated computations are correspondingly assigned to the same chip. We assume that the chips have sufficient on-chip buffer to store layer inputs and activations for each DNN layer being computed. This is to avoid activation write-back to the weight

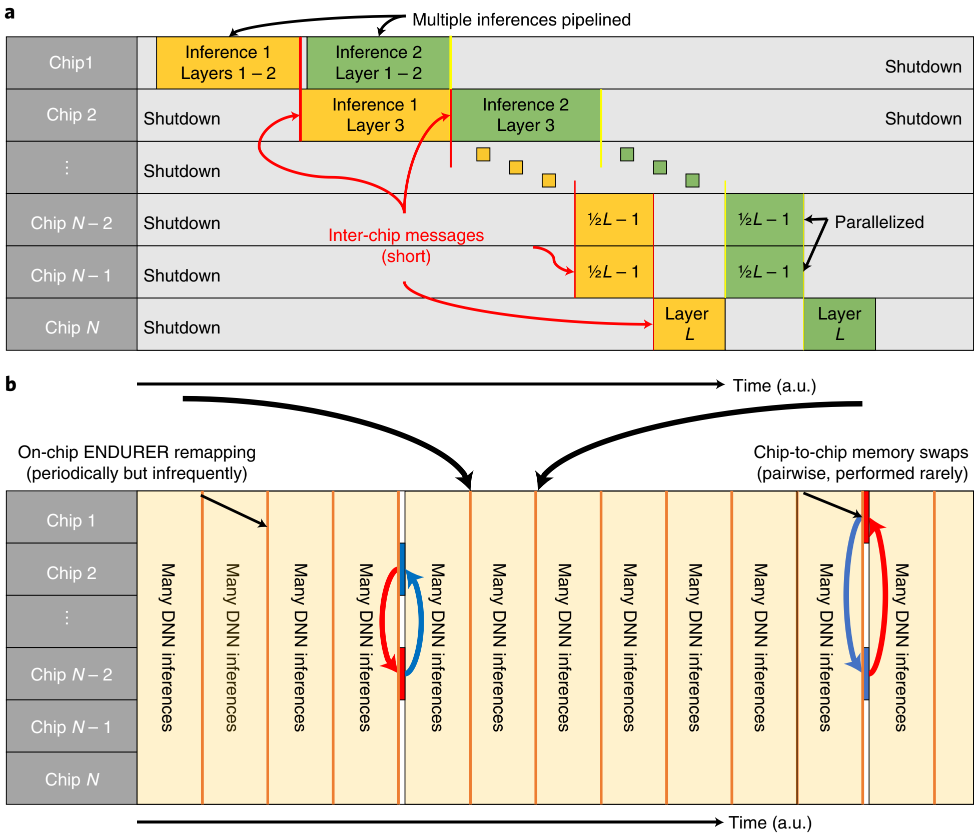


Fig. 3 | Inference scheduling with quick wakeup and shutdown and Distributed ENDURER. a, An Illusion system schedule, corresponding to the example in Fig. 2, that utilizes the quick wakeup and shutdown (implemented via fine-grained temporal power gating (FGTPG)) of each chip in the Illusion system for energy efficiency. The mapping in Fig. 2 generates sparse inter-chip communication, resulting in minimal additional overhead during inference execution. As mapped, layers can be parallelized across chips for a single inference and multiple inferences pipelined to improve execution time and throughput beyond a serialized execution. FGTPG is used both in the single inference case and within the multi-inference pipeline to eliminate idle energy during pipeline stage stalls. **b**, Distributed ENDURER provides write endurance resilience if needed by the on-chip memory technology. Normally, DNN inferences are performed as described in **a**; periodically (in time, but infrequently) the on-chip ENDURER[2] primitives perform a remapping procedure on the local on-chip RRAM to distribute write wear evenly across all words in the on-chip memory[2]. A Distributor primitive (details are provided in Supplementary Section 5) evaluates write wear across the component chips in the Illusion system. If needed, it performs a chip-to-chip memory swap between the most written (red) and least written (blue) chips' memories. By design, we can ensure this occurs only rarely throughout the lifetime of the Illusion system.

memory (desirable for non-volatile weight memory with limited write endurance; Supplementary Section 5). For large activations (and weight memory technology with sufficient write endurance), the algorithm can be modified to provision memory capacity for activations in addition to weights (versus weights only, as described).

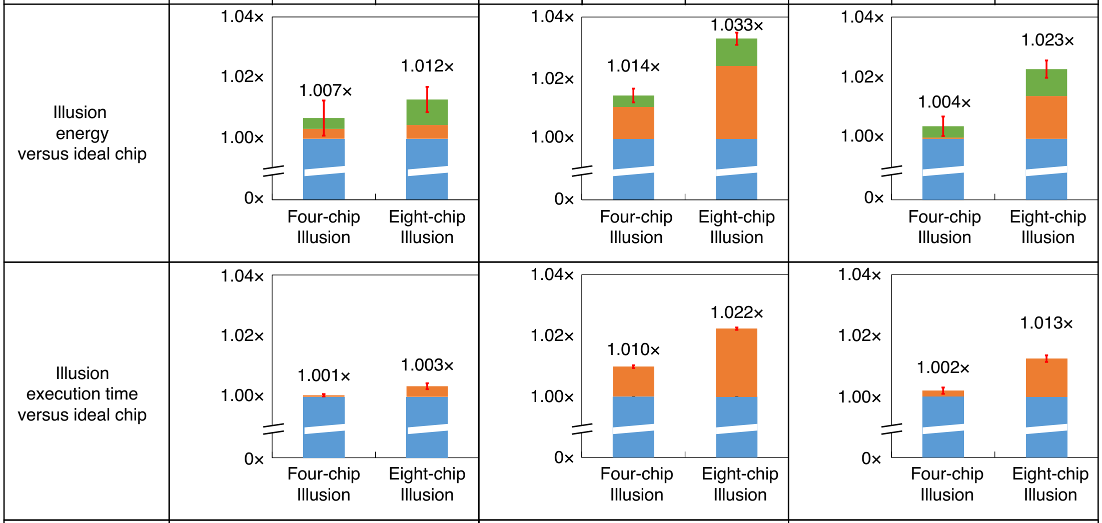
Two heuristics guide our mapping to minimize inter-chip messages at runtime (and satisfy the message limit, MM). First, we map sequential DNN layers to the same chip if possible (Supplementary Fig. 3). Second, if a layer does not fit in the remaining on-chip capacity, we partition its weights along the dimension that results in the fewest inter-chip messages (DNN layer type-specific; Supplementary Fig. 3 presents fully connected layers, while other types are given in Supplementary Section 1). These heuristics yield an algorithm that is applicable to any DNN. Additional mapping optimizations can utilize excess system capacity to further reduce messages (for example, by allowing duplicated weights). Our BILP enables such use and can help explore such trade-offs.

The Illusion scheduling algorithm (given in Supplementary Fig. 3) is then applied to the mapping output (Figs. 2 and 3). The algorithm determines (for each chip in the Illusion system) the timing of inter-chip messages, chip wakeup, on-chip computation and chip shutdown, as required to properly perform inference for the mapped DNN (wakeup and shutdown details are provided in the Methods). Our fine-grained wakeup and shutdown mechanisms ensure the desired near-ideal energy (because idle energy is avoided). For NVM Illusion systems, both compute and memory can be power-gated. Illusion systems with volatile memory can power-gate just the compute and put memory into retention mode. The on-chip scheduling heuristic maintains the same on-chip data flow (for example, systolic row stationary, weight stationary and so on) as the ideal chip, even when a layer is partitioned by our mapping algorithm. This keeps the overall computation energy similar for the ideal chip and the Illusion system.

Our scheduling algorithm provides two additional optimizations to further reduce execution time and increase throughput. Our mappings often split a layer inter-chip (for an example see Fig. 3a). This allows concurrent computation of the layer partitions (model parallelism). Similarly, the mappings naturally form an inter-chip pipeline (pipelined parallelism), whereby each chip can concurrently compute its portion of the computation for different inferences. Our scheduling capitalizes on both parallelization to improve the execution time and pipelining to increase the throughput (for example, model pipeline parallelism). Combining these, Illusion thus achieves high throughput for a given mapping. If the inter-chip

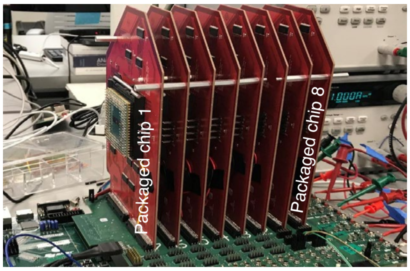
a

	SVHN CNN			KWS LSTM			D2NN		
Configuration	Ideal chip	Four-chip Illusion	Eight-chip Illusion	Ideal chip	Four-chip Illusion	Eight-chip Illusion	Ideal chip	Four-chip Illusion	Eight-chip Illusion
Energy (mJ)	23.78	23.95	24.05	18.06	18.32	18.66	2.51	2.52	2.57
Execution time (s)	32.93	32.94	33.04	25.13	25.37	25.69	3.50	3.51	3.55



	SVHN CNN	KWS LSTM	D2NN
EDP versus ideal chip	<1.02×	<1.06×	<1.04×
Legend	Compute + on-chip memory	Inter-chip messages	Chip shutdown mode

b



c

	Eight-chip Illusion serialized	Eight-chip Illusion parallelized	Eight-chip Illusion pipelined
KWS LSTM inferences	Single	Single	Multiple
Energy (mJ)	18.66	18.66	18.66
Execution time (s)	25.69	3.77	25.69
Throughput (inf. per s)	0.039	0.26	0.30

Fig. 4 | Illusion system performance summary. a, A breakdown of the measured energy and execution times for the ideal chip, our four-chip Illusion system and our eight-chip Illusion system across three DNNs (details are provided in the Methods). The four-chip and eight-chip Illusion system energy is within 1.035×, execution time within 1.025× and EDP within 1.06× those of the ideal chip across all DNNs. To account for chip-to-chip performance variations, all values reported are the mean value across 64 measurement samples, with 95% confidence intervals (error bars) for the relative performance as shown. The KWS LSTM performs a single inference on a sequence of 40 inputs, and the prediction is performed on the last input. The D2NN has two data paths: high accuracy (H) and low energy (L). An inference only executes one path depending on the input data. The energy and execution time reported are for the average path executed. In shutdown mode, a small on-chip scheduler idles, consuming energy. The time to enter/exit shutdown mode is <0.1 ms. **b**, Photograph of the hardware test set-up (see Methods for details). **c**, The Illusion scheduling algorithm can provide additional execution time reduction through parallelization and increased multiple-inference throughput via pipelining. These values are computed from the schedules measured in **a**.

network or the power management scheme limits Illusion operation to a single computing chip at any given time, our scheduling algorithm allows parallelization and/or pipelining to be turned off. The two-step mapping and scheduling algorithms can be merged using heuristics that jointly optimize both inter-chip messages and throughput. Alternatively, an updated BILP with an updated cost function can be used.

Illusion hardware performance

We measured the energy and execution time of our Illusion system hardware for various DNN types: a large CNN (whose weights require the total memory across all eight chips), a large LSTM and a large D2NN. The Illusion system hardware (Fig. 4b), measurement methodology and DNN details are described in the Methods (including Illusion system measurements for a smaller DNN whose weights fit on a single chip). These DNNs represent workloads in computer vision (for example, CNNs for object recognition such as on the Street View House Numbers (SVHN) dataset[34]) and natural language processing (for example, a LSTM for Keyword Spotting (KWS)[35]). The D2NN[36] executes either a high-energy/high-accuracy or a low-energy/low-accuracy path, depending on the input data, to exploit trade-offs between energy and accuracy, thus highlighting the configurability and flexibility of Illusion. Each DNN was analysed for an ideal chip, a four-chip Illusion system and an eight-chip Illusion system. The ideal chip and four-chip system are measured using one and four of the hardware chips, respectively. Weights are mapped to the chips as if they have full capacity (for example, all weights to one chip for the ideal chip), but are overlapped in the

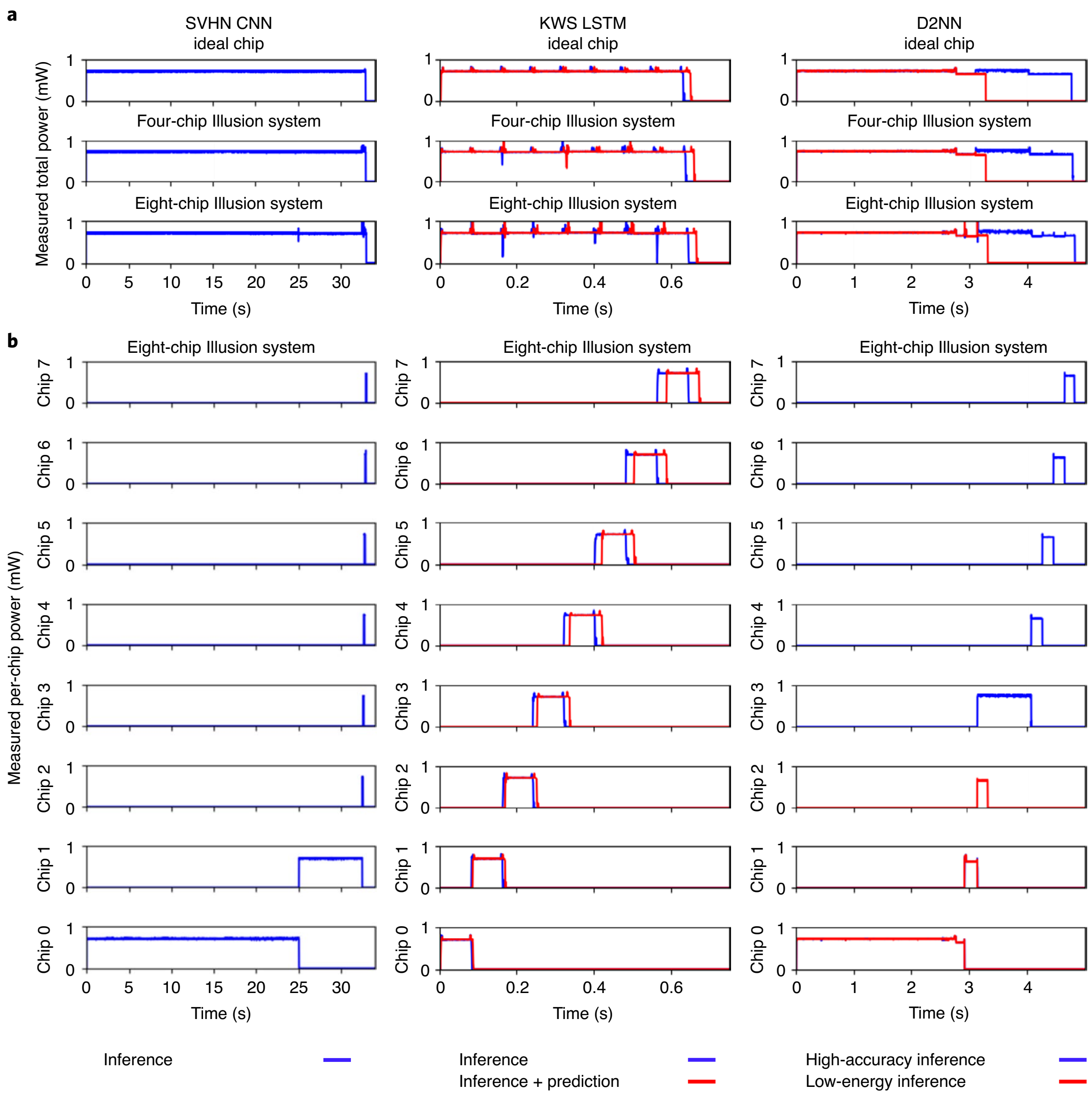


Fig. 5 | Measured ideal chip and Illusion system total power and per-chip power. a, Comparison of the total measured power during inference for the three DNNs summarized in Fig. 4 with the Ideal chip, the four-chip Illusion system and the eight-chip Illusion system. The KWS LSTM performs a single inference on a sequence of 40 inputs. The prediction is performed on the last input; here we separate and show the first and last inputs. The D2NN has two data paths, high accuracy (H) and low energy (L). An inference only executes one path. We show both paths. **b**, A per-chip breakdown of the measured power for the eight-chip Illusion system. As a result of the quick wakeup and shutdown, chips are in very low power shutdown mode when not actively computing.

physical address space using a software-defined data structure. This yields similar instruction sequences and memory accesses (and therefore system execution time) as if the full capacity were present (further details are provided in the Methods). All systems—ideal, four-chip and eight-chip Illusion—have the same wakeup and shutdown behaviour to provide an apples-to-apples comparison. Figures 4 and 5 provide key results, with additional data and a discussion on the Illusion mappings and schedules provided in Supplementary Section 1.

Across all DNNs (Fig. 4a), our eight-chip Illusion system EDPs are within 1.06× of the ideal chip EDPs, and the energy and execution time are within 1.035× and 1.025×, respectively. As we go from an eight-chip to a four-chip Illusion system (that is, each chip in the four-chip Illusion system with twice the on-chip RRAM capacity versus the eight-chip Illusion system), our Illusion improves to below 1.02× of the ideal chip EDP. These results also agree with our detailed simulations (Supplementary Section 3) and analytical models (Supplementary Section 4). The corresponding message sizes are very small compared to the DNN size—the key to Illusion (Supplementary Table 1). Illusion thus maintains the large benefits of an ideal chip versus a traditional system with off-chip memory. As expected, the inter-chip message counts increase as we go from a four-chip to an eight-chip Illusion system. This raises the following question: what minimum memory capacity per chip is required to achieve near-ideal energy, execution time and/or EDP? We answer this question below.

Our Illusion scheduling algorithm increases inference throughput beyond a desired single-inference Illusion in two ways: concurrent execution of a DNN layer split inter-chip by our mapping (that is, model parallelism) and concurrent execution of multiple inputs through the Illusion system (that is, pipelined parallelism), when combined yielding model pipeline parallel execution. In Fig. 5, we focus our measurements on the conservative use case—a single inference with serial execution of split layers—to show that both energy and execution time (and therefore EDP) are near the ideal chip values (for example, within 1.06× EDP). This ensures that we do not obfuscate the per-inference energy and execution time results by increasing the multiple inference throughput (for example, via pipelining). Instead, we achieve near-ideal energy, execution time and EDP and thus our Illusion system has created the illusion of an ideal chip with large on-chip memory. With parallelism and pipelining, our scheduling can provide an up to 7.6× increase in throughput (Fig. 4c) versus a single-input inference. Supplementary Section 1 discusses the heuristics used in our mapping and scheduling algorithms and their impact on throughput.

In addition to our hardware demonstrations, we also simulate a variety of well-known DNNs for larger-scale Illusion systems: CNNs based on ResNet-50[37], VGG-Net[38] and AlexNet[39] (on the ImageNet Dataset[40]) and an LSTM language model[41] (on the One Billion Word Benchmark[42]). Our simulations use an end-to-end framework[2] that has been calibrated/validated using hardware data (for example, a multicore processor or DNN accelerator). Our results demonstrate (as detailed in Supplementary Section 3) strong agreement with our hardware results (for DNNs run on Illusion hardware). Our Illusion systems for large-scale DNN inference provide <1.1× EDP versus an ideal chip, which is up to 44× (depending on the DNN) better than a comparable baseline with off-chip memory (DRAM). Our scheduling algorithm provides additional throughput benefits (up to 5.5× on an eight-chip Illusion system) while maintaining near-ideal energy. Illusion is effective across a sweep of design points (memory capacity per chip, inter-chip network characteristics), demonstrating its broad applicability. In Supplementary Section 3 we provide a brief discussion on the Illusion system's PE count versus that of the ideal chip for various inference scenarios.

Next, we analytically modelled (Supplementary Section 4) the Illusion system to derive estimates for inter-chip messages (based on the DNN type, size, average activation size and the memory capacity per chip). The estimates are conservative (that is, our mapping algorithm generates mappings with fewer inter-chip messages versus our model) for the systems analysed using hardware measurements and simulations (Supplementary Section 4 provides a comparison). Using these estimates and the inter-chip network characteristics (for example, bandwidth and energy per byte), we calculate the degree (for example, 1.1×) of near-ideal energy, execution time and EDP achievable.

As Fig. 6a shows, for a desired degree of near-ideal EDP, each DNN has a different minimum-memory-per-chip point (or, equivalently, a maximum number of chips in the Illusion system). We derived the critical insight that Illusion systems can be characterized by the sizing ratio: the ratio of DNN model size (that is, the total memory capacity required to store the DNN weights) to memory capacity per chip in the Illusion system. This sizing ratio is similar across DNN types studied in this Article and depends on the inter-chip network characteristics. For a desired near-ideal performance (for example, 1.1× ideal chip EDP), the sizing ratio bounds the maximum number of chips allowed in an Illusion system (Fig. 6b). Our analytical model and the sizing ratio help derive useful guidelines for Illusion system design, especially in the context of emerging technology trends. Advanced inter-chip networks, for example, with 2.5D chiplet integration[23,30–32,43], result in larger sizing ratios (that is, more and smaller chips can now support the same near-ideal EDP). This is critical in scaling DNN sizes on Illusion systems. For a given inter-chip network, larger DNNs demand increased on-chip memory integration (to preserve the sizing ratio), which has profound implications for memory technologies (efficient and dense NVM (including multiple bits per cell storage) and dense integration with logic, for example through ultra-dense (for example, monolithic) 3D integration[1,2,33]). With improvements in compute energy efficiency—for example, through energy-efficient logic devices[44] or SRAM/NVM-based in-memory computing[20,21,45–50]—further on-chip memory integration and/or better inter-chip networks are required to preserve Illusion's near-ideal performance.

A multi-chip write endurance resilience technique

Emerging NVM technologies such as RRAM promise energy efficiency, high density and dense (for example, monolithic) 3D integration. However, Illusion must overcome the write endurance challenges of RRAM because of Illusion's frequent chip wakeup and shutdown, and some DNNs have persistent states (for example, the LSTM 'cell' and 'hidden' states) to maintain through shutdown in the Illusion system. This requires sending the state to the system host (additional message cost) or writing the state to the on-chip NVM (that is, checkpointing). Similarly, for Illusion system flexibility, if a DNN layer's inputs and activations do not fit fully into the on-chip SRAM buffer, inference might require some activations to be written to the on-chip NVM.

We present Distributed ENDURER, a new multi-chip write endurance resilience technique enabling the use of Illusion in emerging memory technologies with write endurance limitations. Distributed ENDURER increases the Illusion system lifetime from months to 10 years, while maintaining its energy, execution time and EDP near those of the ideal chip. Distributed ENDURER is inspired by our previous work—single-chip ENDURER[2]—which filters and redistributes writes evenly across all NVM words, achieving a 10-year single-chip lifetime[27]. Single-chip ENDURER cannot be used directly for Illusion, because our mapping and scheduling algorithms rely on a specific mapping; applying single-chip ENDURER across the entire Illusion system randomizes this mapping and drastically increases inter-chip messages. Moreover, applying single-chip ENDURER locally on each chip may not be sufficient, because writes may not occur evenly across all chips.

Distributed ENDURER consists of two hardware primitives, a new Distributor primitive and our existing single-chip ENDURER primitive[2], implemented on each chip of the Illusion system. Single-chip ENDURER filters writes to frequently written words via a write buffer, and distributes writes intra-chip to the NVM through a local remap procedure that periodically (in time) remaps the NVM address space (intra-chip) by an offset (constant random number for each entire period). Our new Distributor counts the total number of on-chip NVM words written. The counter values from all chips are broadcast via inter-chip messages at the end of every ENDURER remap period. When the counter values indicate imbalance (detailed in Supplementary Section 5), the Distributor uses the inter-chip network to perform a chip-to-chip memory swap, swapping out the entire NVM contents of the most-written chip with that of the least-written.

Combining these two primitives, we have demonstrated 10 years of continuous inference across SVHN CNN, KWS LSTM and D2NN workloads on our Illusion system hardware. Distributed ENDURER incurs minimal additional energy and execution time impact—less than 0.1% extra—resulting from chip-level ENDURER operations and inter-chip messages generated by chip-to-chip swaps and write counter broadcasts. Without Distributed ENDURER, endurance-induced permanent write failures considerably degrade inference within a year (Fig. 6c). Supplementary Section 5 provides bounds on the Distributed ENDURER lifetime, the required chip-to-chip swaps and the testing methodology.

Conclusions

Our Illusion system overcomes one of the key challenges facing DNN systems—the need for large on-chip memory capacity accessible at high bandwidth in an energy-efficient manner. Hardware results demonstrate the effectiveness and practicality of our Illusion

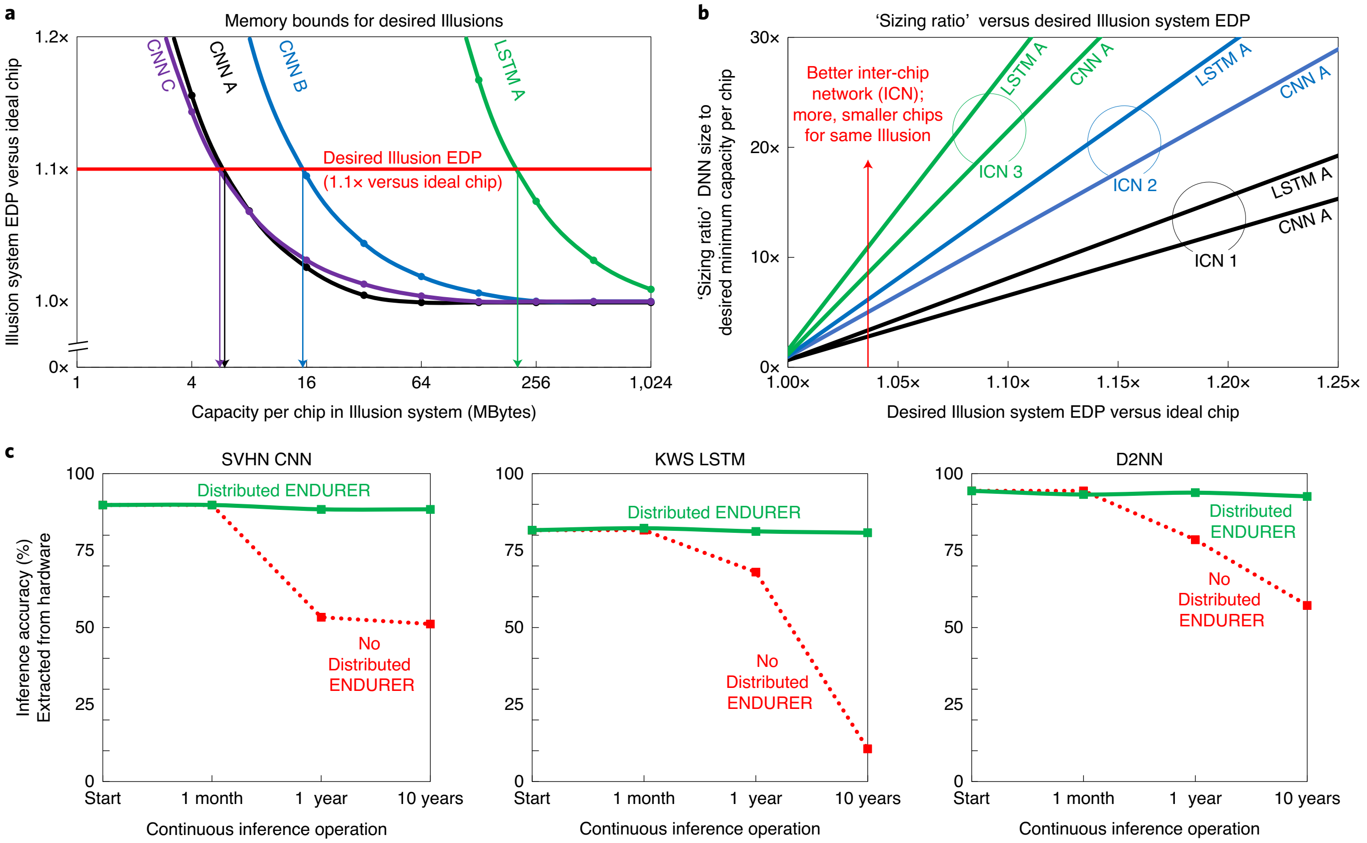


Fig. 6 | Illusion's minimum capacity per chip and Distributed ENDURER performance. a, Given a desired Illusion system performance (for example, 1.1× ideal chip EDP, as shown), our analytical model can be used to determine the minimum capacity per chip needed to achieve that performance or better. Supplementary Section 3 provides the DNN specifications. The derivation of the conservative analytical model is provided in Supplementary Section 4. The inter-chip network is assumed to provide 32 GBytes s^{-1} chip-to-chip bandwidth with a message energy of 256 pJ per byte. Drop-down arrows indicate these minima for the DNNs shown. **b**, The 'sizing ratio' is the ratio of the DNN weight capacity to the minimum capacity per chip found in **a**. Equivalently, this is a conservative maximum number of chips with which the desired Illusion EDP can be achieved. For LSTM A and CNN A (Supplementary Section 1), we compute these sizing ratios for three different inter-chip network (ICN) assumptions. ICN1 is the same as in **a**. ICN2 has 64 GBytes s^{-1} chip-to-chip bandwidth with a message energy of 128 pJ per byte. ICN3 has 128 GBytes s^{-1} chip-to-chip bandwidth with a message energy of 64 pJ per byte. As the ICN improves from ICN1 to ICN3, our sizing ratio increases and thus more, smaller chips achieve the same Illusion system EDP (shown here relative to the ideal chip). **c**, Ten-year continuous inference is demonstrated using Distributed ENDURER for the SVHN CNN, KWS LSTM and D2NN. Without Distributed ENDURER, all DNNs suffer inference accuracy degradation within a year due to limited write endurance. Supplementary Section 5 provides the measurement methods.

system, which achieves energy, execution time and EDP within 3.5%, 2.5% and 6%, respectively, of the values for a single ideal chip when used in several DNN inference applications. Illusion also offers a scalable path for DNN hardware advances in the future. In particular, its effectiveness can be amplified with emerging technologies, such as 2.5D chiplet integration, dense NVM and new logic devices, in-memory computing and ultra-dense (for example, monolithic) 3D integration of logic and memory. At the same time, Illusion can play a critical role in guiding the progress of these technologies themselves. Beyond the DNNs explored in this work, our approach is also applicable for emerging DNN workloads such as deep learning recommendation models and transformers. Beyond inference, Illusion-based approaches for DNN training are also possible with modifications to our mapping and scheduling algorithms.

Methods

Hardware demonstration of Illusion. Our Illusion demonstration hardware is presented in Supplementary Fig. 1. Eight identical chips are used in the Illusion system. Each chip[27] monolithically and heterogeneously integrates two technologies: RRAM on top of commercial 130-nm silicon CMOS. Ideal chips and Illusion systems both require a host (Fig. 1). The host is responsible for wakeup (and shutdown) of each DNN chip, sending DNN inputs to the first chip and receiving final outputs via the inter-chip network. An FPGA serves as the host for our hardware demonstration.

The inter-chip network is realized as a bus, with the FPGA serving as the bus host. Each chip sends/receives messages via its peripheral port to the FPGA host, and first-in–first-out (FIFOs) buffers implemented on the FPGA are used as network buffers. Each chip has wakeup and reset external interrupts. However, in our demonstration system, only one chip can be active at a time (that is, wakeup asserted, actively computing and not in reset), otherwise there is contention for the inter-chip bus. Inter-chip messages contain a single destination header. No other control is required as the Illusion mapping specifies the message length, and the DNN software on-chip is compiled with input/output message lengths. The network operates as follows. With an active chip, the FPGA buffers incoming messages, waits for the active chip to safely shut down, reads the message destination header, transfers the messages to an output FIFO, and wakes up the destination chip, which then consumes the messages. After software completion, chips self-shutdown by flagging the on-chip scheduler.

We drop the FPGA's energy in our measurements. As the host, it is active during the entire inference for both the ideal chip and Illusion systems. Furthermore, the FPGA buffering and intermediation are only required because of a lack of externalized interrupts (our chips were not initially designed to communicate chip-to-chip); otherwise, we could directly send messages chip-to-chip with an external interrupt scheme to determine bus control. The chip-side power needed to drive and read the bus is measured through the chip's power rails. The on-chip scheduler is critical to implement the fine-grained chip power gating specified by the Illusion schedule.

Applications under test. To demonstrate the functionality of the Illusion approach, we used four different DNNs: a small (~4 kByte) CNN and a large (~32 kByte) high–low topology D2NN[36] trained on the MNIST dataset[51], a large (~32 kByte) CNN trained on the SVHN[34] dataset and a large (~32 kByte) LSTM trained on Google's KWS dataset[52] with 10 mel-frequency cepstrum[35] input features extracted. These datasets are common use cases for small DNNs. Using a range of DNN types and sizes allows us to demonstrate Illusion's generality. Supplementary Table 2 summarizes the networks used. Note that quantization is not required for Illusion; rather, we do so to ensure fast operation on our limited hardware (an integer 16-bit MAC unit). Our Illusion approach is the same regardless of data bit width.

Data pre-processing. All images in the SVHN and MNIST datasets were normalized to pixel values in [−1,1] and quantized to signed 8 bits with 4 bits of fractional precision. The KWS spotting audio dataset was pre-processed using a mel-frequency cepstrum (MFCC)[35] model to extract 10 features per input. These features were also quantized to values in [−1,1], to signed 8 bits with 7 bits of fractional precision.

Quantized training. To train quantized weights that would translate well to hardware, we used the low-precision simulation package QPyTorch[53]. QPyTorch provides a low-precision optimizer and handles low-precision weight, gradient, momentum and error accumulation updates. This package allows us to perform training and inference using fixed-point quantized weights and activations. During training, the forward pass is fixed-point-quantized to 8-bit weights and 16-bit accumulation, while the backward pass (that is, the gradients, momenta and errors) is left in floating point.

Native C implementation and Illusion system implementations. QPyTorch uses floating-point operations in the back-end (with results forced to fixed-point values). There are slight differences between this approach and a native C implementation (in particular with the accumulation and rounding). To imitate inference on our hardware, we tested the final accuracy by implementing the quantized network in C and compiling natively (Supplementary Table 2 provides details of inference accuracy). This code was then modified to properly use our test hardware (for example, linking data to the right memory segments, ensuring the correct use of the multiplier on-chip) and the message passing code was added. We ran this compiled code[54] on a cycle-accurate chip register-transfer level simulation to estimate the execution time as a check of our measurement results.

Illusion system and ideal chip execution time and energy measurement. We measured the voltage across a 1% shunt resistor for the supplies of each chip. The resistor nominals were chosen to provide voltages within the 0.1-V range of the 0.5 kSample s^{-1} 12-bit analog-to-digital converter (a LabJack T7Pro; the KWS LSTM was measured at 1 kSample s^{-1}). The shunt resistors were measured independently to account for variations from nominal to accurately determine the current. The measured power alternated between distinct active and shutdown modes; thresholding these modes was used to determine the execution time. Three supplies (V_{DD} for the digital logic, V_{DDSA} for the RRAM sense amplifier and V_{CC} for the RRAM controller) were measured independently, time-multiplexed. The bias current for the RRAM sense amplifier (generated off-chip) was not power-gated by the on-chip scheduler, and we removed this bias power during the shutdown mode. The eight-chip Illusion system was measured chip by chip, time-multiplexed (for each of the three channels). Communication energy was accounted for by the increased chip power draw. Measurements continued for the total Illusion runtime to measure the shutdown mode power of the chips (due to cell leakage and the scheduler). Our hardware has chip-to-chip variations in performance. To reduce the impact of this variation we measured each mapped DNN segment for each of the eight chips in our hardware (that is, weights for Illusion system 'chip 0' can be mapped to any one of physical chips 0, 1, 2 and so on). We sampled 64 possible permutations of these configurations and computed the mean energy and execution time, along with the 95% confidence interval as reported in Fig. 4.

For the SVHN CNN, KWS LSTM and D2NN, we mimicked our ideal chip and four-chip Illusion systems on one and four of our hardware chips, respectively. The physical measurement techniques were the same as described above. The weight mapping to the chips assumed the full capacity was available (for example, 32 kBytes for the ideal chip on one chip). This resulted in more weights than can be compiled into the physical memory (4 kBytes capacity). These excess weights were overlapped (addresses modulo 4 kBytes) in the same physical address space using a software-defined data structure. Owing to the simple instruction set in our hardware, this mimicked the same instruction execution and memory access patterns as an ideal chip or four-chip Illusion system.

Our corresponding energy measurements are optimistic for our ideal chip and four-chip Illusion systems for two reasons. First, larger RRAM on-chip requires additional idle power during computation. Second, per read, additional logic would be activated, yielding more expensive memory accesses. As we use smaller hardware chips in our energy measurements, we are most optimistic for the ideal chip and optimistic for the four-chip Illusion system, with hardware-accurate measurements for the eight-chip Illusion system. The Illusion system EDP values we achieve (relative to the ideal chip measurement) are thus conservative estimates.

Our small CNN (inference on MNIST; results are provided in Supplementary Fig. 2) requires none of this treatment, as it fits on each on-chip memory for all the systems (for the four-chip and eight-chip Illusions, we map as if we had only 1 kByte or 0.5 kBytes of RRAM per chip). As we measure on physically larger hardware chips, the four-chip Illusion system is pessimistic and the eight-chip Illusion system is even more pessimistic (for the same two reasons as above). The Illusion system EDPs we achieve (relative to the ideal chip measurement) are thus conservative estimates. For the three DNNs discussed above and the small MNIST CNN, the results are consistent across scales, confirming that our measurement techniques for the four-chip Illusion system and ideal chip are valid on the large DNNs, as one chip is already an ideal chip for the small CNN. In addition, by using exactly the same hardware for workloads requiring one chip (MNIST CNN) up to eight chips (SVHN CNN, KWS LSTM, D2NN), we show that our Illusion systems are configurable and flexible. We achieve near-ideal EDP, regardless of the number of chips used by the DNN in the Illusion system.

Data availability

The data that support the findings of this work are available at https://github.com/robust-systems-group/illusion_system.

Code availability

The code that supports the findings of this work is available at https://github.com/robust-systems-group/illusion_system.

Received: 17 May 2020; Accepted: 18 November 2020;
Published online: 11 January 2021

References

1. Aly, M. M. S. et al. Energy-efficient abundant-data computing: the N3XT 1,000. *Computer* **48**, 24–33 (2015).
2. Aly, M. M. S. et al. The N3XT approach to energy-efficient abundant-data computing. *Proc. IEEE* **107**, 19–48 (2019).
3. Donato, M. et al. On-chip deep neural network storage with multi-level eNVM. In *Proc. 55th Design Automation Conference* (*DAC*) https://doi.org/10.1145/3195970.3196083 (IEEE, 2018).
4. Li, H., Bhargava, M., Whatmough, P. N. & Wong, H.-S. P. On-chip memory technology design space explorations for mobile deep neural network accelerators. In *Proc. 56th Design Automation Conference* (*DAC*) https://doi.org/10.1145/3316781.3317874 (IEEE, 2019).
5. Hestness, J. et al. Deep learning scaling is predictable, empirically. Preprint at https://arxiv.org/abs/1712.00409 (2017).
6. Xu, X. et al. Scaling for edge inference of deep neural networks. *Nat. Electron.* **1**, 216–222 (2018).
7. Wu, C. J. et al. Machine learning at Facebook: understanding inference at the edge. In *Proc. International Symposium on High Performance Computer Architecture* (*HPCA*) 331–344 https://doi.org/10.1109/HPCA.2019.00048(IEEE, 2019).
8. Sun, G., Zhao, J., Poremba, M., Xu, C. & Xie, Y. Memory that never forgets: emerging nonvolatile memory and the implication for architecture design. *Natl Sci. Rev.* **5**, 577–592 (2018).
9. Yarotsky, D. Error bounds for approximations with deep ReLU networks. *Neural Netw.* **94**, 103–114 (2017).
10. Jouppi, N. P. et al. In-datacenter performance analysis of a tensor processing unit. In *Proc. International Symposium on Computer Architecture* (*ISCA*) 1–12 (ACM, 2017).
11. Lie, S. Wafer-scale deep learning (Hot Chips 2019 Presentation) https://www.hotchips.org/hc31/HC31_1.13_Cerebras.SeanLie.v02.pdf (Cerebras, 2019).
12. Chen, Y. H., Emer, J. & Sze, V. Eyeriss: a spatial architecture for energy-efficient dataflow for convolutional neural networks. In *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)* https://doi.org/10.1109/ISCA.2016.40 (2017).
13. Gao, M., Pu, J., Yang, X., Horowitz, M. & Kozyrakis, C. TETRIS: scalable and efficient neural network acceleration with 3D memory. In *Proc. 22nd International Conference on Architectural Support for Programming Languages and Operating Systems* (*ASPLOS*) 751–764 (ACM, 2017).
14. Gao, M., Yang, X., Pu, J., Horowitz, M. & Kozyrakis, C. Tangram: optimized coarse-grained dataflow for scalable NN accelerators. In *Proc. 24th International Conference on Architectural Support for Programming Languages and Operating Systems* (*ASPLOS*) 807–820 (ACM, 2019).
15. Yang, X. et al. Interstellar: using Halide's scheduling language to analyze DNN accelerators. In *Proc. 25th International Conference on Architectural Support for Programming Languages and Operating Systems* (*ASPLOS*) 369–383 (ACM, 2020).
16. Rabii, S. et al. Computational directions for augmented reality systems. In *VLSI Symposium Circuits* 102–106 (IEEE, 2019).

17. Wong, H.-S. P. & Salahuddin, S. Memory leads the way to better computing. *Nat. Nanotechnol.* **10**, 191–194 (2015).
18. Jung, M. et al. Driving into the memory wall: the role of memory for advanced driver assistance systems and autonomous driving. In *Proc. International Symposium on Memory Systems* https://doi.org/10.1145/3240302.3240322 (ACM, 2018).
19. Dazzi, M. et al. 5 Parallel Prism: a topology for pipelined implementations of convolutional neural networks using computational memory. Preprint at https://arxiv.org/abs/1906.03474 (2019).
20. Song, L., Qian, X., Li, H. & Chen, Y. PipeLayer: a pipelined ReRAM-based accelerator for deep learning. In *Proc. International Symposium on High-Performance Computer Architecture* (*HPCA*) 541–552 (IEEE, 2017).
21. Ankit, A. et al. PUMA: a programmable ultra-efficient memristor-based accelerator for machine learning inference. In *Proc. International Conference on Architectural Support for Programming Languages and Operating Systems* (*ASPLOS*) https://doi.org/10.1145/3297858.3304049 (ACM, 2019).
22. Narayanan, D. et al. PipeDream: generalized pipeline parallelism for DNN training. In *ACM Symposium on Operating Systems Principles* https://doi.org/10.1145/3341301.3359646 (SOSP, 2019).
23. Shao, Y. S. et al. Simba: scaling deep-learning inference with multi-chip-module-based architecture. In *Proc. Annual International Symposium on Microarchitecture*, *MICRO* 14–27 (IEEE, 2019).
24. Wei, X., Liang, Y. & Cong, J. Overcoming data transfer bottlenecks in FPGA-based DNN accelerators via layer conscious memory management. In *Proc. 56th Annual Design Automation Conference* https://doi.org/10.1145/3316781.3317875 (ACM, 2019).
25. Huang, Y. et al. GPipe: efficient training of giant neural networks using pipeline parallelism. In *Advances in Neural Information Processing Systems* (*NeurIPS*) 32 (NIPS, 2019).
26. Le, B. Q. et al. Resistive RAM with multiple bits per cell: array-level demonstration of 3 bits per cell. *IEEE Trans. Electron Devices* **66**, 641–646 (2019).
27. Wu, T. F. et al. 14.3-A 43-pJ/cycle non-volatile microcontroller with 4.7-µs shutdown/wake-up integrating 2.3-bit/cell resistive RAM and resilience techniques. In *Proc. IEEE International Solid-State Circuits Conference* (*ISSCC*) 226–228 (IEEE, 2019).
28. Hsieh, E. R. et al. High-density multiple bits-per-cell 1T4R RRAM array with gradual SET/RESET and its effectiveness for deep learning. In *Proc. International Electron Devices Meeting* (*IEDM*) https://doi.org/10.1109/IEDM19573.2019.8993514 (IEEE, 2019).
29. Chen, A. A review of emerging non-volatile memory (NVM) technologies and applications. *Solid State Electron.* **125**, 25–38 (2016).
30. Naffziger, S., Lepak, K., Paraschou, M. & Subramony, M. AMD chiplet architecture for high-performance server and desktop products. In *Proc. IEEE International Solid-State Circuits Conference* (*ISSCC*) 44–45 (IEEE, 2020).
31. Vivet, P. et al. A 220GOPS 96-core processor with 6 chiplets 3D-stacked on an active interposer offering 0.6-ns/mm latency, 3-Tb/s/mm^2 inter-chiplet interconnects and 156-mW/mm^2 @ 82%-peak-dfficiency DC–DC converters. In *Proc. IEEE International Solid-State Circuits Conference* (*ISSCC*) 46–48 (IEEE, 2020).
32. Greenhill, D. et al. A 14-nm 1-GHz FPGA with 2.5D transceiver integration. In *Proc. IEEE International Solid-State Circuits Conference* (*ISSCC*) 54–55 (IEEE, 2017).
33. Shulaker, M. M. et al. Three-dimensional integration of nanotechnologies for computing and data storage on a single chip. *Nature* **547**, 74–78 (2017).
34. Netzer, Y. & Wang, T. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning* http://ufldl.stanford.edu/housenumbers (NIPS, 2011).
35. Zhang, Y., Suda, N., Lai, L. & Chandra, V. Hello Edge: keyword spotting on microcontrollers. Preprint at https://arxiv.org/abs/1711.07128 (2017).
36. Liu, L. & Deng, J. Dynamic deep neural networks: optimizing accuracy-efficiency trade-offs by selective execution. In *Proc. 32nd AAAI Conference on Artifical Intelligence* 3675–3682 (AAAI, 2018).
37. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 770–778 (IEEE, 2016).
38. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings* (ICLR, 2015).
39. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Neural Inf. Process. Syst.* https://doi.org/10.1145/3065386 (2012).
40. Deng, J. et al. ImageNet: a large-scale hierarchical image database. In *Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition* 248–255 (IEEE, 2009).
41. Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N. & Wu, Y. Exploring the limits of language modeling. Preprint at https://arxiv.org/abs/1602.02410 (2016).
42. Chelba, C. et al. One billion word benchmark for measuring progress in statistical language modeling. In *Proc. Annual Conference of the International Speech Communication Association*, *INTERSPEECH* 2635–2639 (International Speech and Communication Association, 2014).
43. Turner, W. J. et al. Ground-referenced signaling for intra-chip and short-reach chip-to-chip interconnects. In *Proc. 2018 IEEE Custom Integrated Circuits Conference*, *CICC 2018* https://doi.org/10.1109/CICC.2018.8357077 (IEEE, 2018).
44. Hills, G. et al. Understanding energy efficiency benefits of carbon nanotube field-effect transistors for digital VLSI. *IEEE Trans. Nanotechnol.* **17**, 1259–1269 (2018).
45. Le Gallo, M. et al. Mixed-precision in-memory computing. *Nat. Electron.* **1**, 246–253 (2018).
46. Dong, Q. et al. A 351TOPS/W and 372.4GOPS compute-in-memory SRAM macro in 7-nm FinFET CMOS for machine-learning applications. In *Proc. IEEE International Solid-State Circuits Conference* (*ISSCC*) 242–244 (IEEE, 2020).
47. Shafiee, A. et al. ISAAC: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars. In *Proc. 43rd Annual International Symposium on Computer Architecture* (*ISCA*) 14–26 (IEEE, 2016).
48. Qiao, X., Cao, X., Yang, H., Song, L. & Li, H. AtomLayer: a universal ReRAM-based CNN accelerator with atomic layer computation. In *2018 55th ACM/ESDA/IEEE Design Automation Conference* (*DAC*) https://doi.org/10.1109/DAC.2018.8465832 (IEEE, 2018).
49. Guo, R. et al. A 5.1-pJ/neuron 127.3-us/inference RNN-based speech recognition processor using 16 computing-in-memory SRAM macros in 65-nm CMOS. In *Proc. 2019 IEEE Symposium on VLSI Circuits* C120–C121 (IEEE, 2019).
50. Wan, W. et al. A 74 TMACS/W CMOS-RRAM neurosynaptic core with dynamically reconfigurable dataflow and in-situ transposable weights for probabilistic graphical models. In *Proc. IEEE International Solid-State Circuits Conference* (*ISSCC*) 498–500 (IEEE, 2020).
51. LeCun, Y., Cortes, C. & Burges, C. J. C. *MNIST Handwritten Digit Database* (2010); http://yann.lecun.com/exdb/mnist/
52. Warden, P. Speech commands: a dataset for limited-vocabulary speech recognition. Preprint at https://arxiv.org/abs/1804.03209 (2018).
53. Zhang, T., Lin, Z., Yang, G. & De Sa, C. QPyTorch: a low-precision arithmetic simulation framework. Preprint at https://arxiv.org/abs/1910.04540 (2019).
54. MSP430-GCC-OPENSOURCE GCC – Open Source Compiler for MSP Microcontrollers (Texas Instruments, accessed 5 August 2020); https://www.ti.com/tool/MSP430-GCC-OPENSOURCE

Acknowledgements

We acknowledge the Defense Advanced Research Projects Agency (DARPA) 3DSoC programme, the NSF/NRI/GRC E2CDA programme, Intel Corporation, CEA-LETI and the Stanford SystemX Alliance. M.M.S.A. is supported in part by the Singapore AME programmatic fund titled Hardware-Software Co-optimization for Deep Learning (project no. A1892b0026). We would also like to acknowledge S. Taheri and the Stanford Prototyping Facility for assistance with the design, test and debugging of the test harness printed circuit boards.

Author contributions

R.M.R. developed the Illusion approach, the system architectural design and the Illusion scheduling and mapping algorithms, and performed all measurements. P.C.J. led DNN implementation and training. R.M.R. and P.T. developed the BILP. T.F.W. and B.Q.L. designed the test chips, under the guidance of E.V., P.V., E.N., E.B. and H.-S.P.W. The test harness was developed by R.M.R. and T.F.W. Y.X., A.B. and R.M.R. performed Illusion system simulations under the guidance of M.M.S.A. The modelling of Illusion was performed by Z.F.K. and R.M.R. Distributed ENDURER was developed by Z.F.K., who performed analysis and simulations with M.M.S.A., with M.W. providing guidance. S.M. was in charge, advised and led on all aspects of the project.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41928-020-00515-3.

Correspondence and requests for materials should be addressed to R.M.R.

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

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