

Fully-Binarized Distance Computation based On-device Few-Shot Learning for XR applications

Vivek Parmar¹, Sandeep Kaur Kingra¹, Syed Shakib Sarwar², Ziyun Li²,
Barbara De Salvo², Manan Suri¹

¹Indian Institute of Technology Delhi ² Meta Reality Labs Research

{vivekparmar, eez168070, manansuri}@ee.iitd.ac.in {shakib7, liziyun, barbarads}@meta.com

Abstract

Low-Power Edge-AI capabilities are essential for on-device extended reality (XR) applications to support the vision of Metaverse. A critical requirement for emerging AI applications is personalization and adaptability without requiring retraining. Few-shot learning using embedding-based computations present an attractive method for the same. However, quantization-based optimizations to map such computations are yet to be explored. In this work, we present a fully binarized distance computing (BinDC) framework to perform distance computations for few-shot learning using only accumulation and logic operations (XOR/XNOR). The proposed method leads to marginal loss in accuracy of $\approx 4\%$ (for 4-bits). This leads to savings in memory ($\approx 8\times$), energy ($\approx 2.5-3\times$), power ($\approx 2\times$) and latency ($\approx 1.1-1.5\times$) compared to a floating-point cosine distance computation when using CPU-based computations performed on an embedded platform. We further demonstrate realization utilizing RRAM (resistive random access memory) based IMC (in-memory computing) to further improve EDP (energy delay product) ($\approx 1000\times$) in comparison to the embedded CPU-based realization.

1. Introduction

Extended reality (XR), *i.e.*, virtual, augmented, and mixed reality have emerged as key enablers for future-ready edge and mobile systems with the advent of Metaverse. XR-devices can be used to enhance user-experience and enable novel capabilities for a wide variety of applications such as education, entertainment, defense, robotics, etc. With the recent innovations in the field of AI (artificial intelligence), XR applications have become more computationally intensive [6]. Current generation of portable XR devices rely on high-performance compute servers to perform bulk of the computation due to limitations of power, compute and memory capabilities of the edge-device. This approach suf-

fers from some major disadvantages such as (i) patchy and non-seamless user experiences, (ii) data transfer/network overheads, and (iii) user privacy and security concerns.

Most commonly used AI applications center around the use of deep neural networks (DNN). Current large state-of-the-art (SOTA) DNNs don't scale to edge-computing use-cases due to power-hungry floating point multiply-accumulate (MAC) operations as well as memory bottleneck due to network size ($> \text{MB}$) [5]. Furthermore, conventional DNNs demonstrate limited tolerance to variations in inputs typically observed in case of edge-AI applications. Such networks are trained once (requiring long training times and high-power computational resources) and deployed with the models updated rarely. Any update often requires re-training from scratch or fine-tuning the DNN over the entire training dataset, including the data the network was previously trained on [7]. To address this, recent focus in the AI community has shifted towards adaptable networks to be able to perform few-shot learning (FSL) *i.e.* learning with limited samples. Two key approaches are adopted for such networks: (a) Metric-based learning and (b) Meta Learning. Metric-based learning typically uses a frozen network and stores embeddings generated from feature extractor (FE) for performing classification/regression at the last layer. In case of meta-learning, fine-tuning is performed at the last layers requiring gradient computations with floating-point precision. For this study, we focus on utilizing metric-based FSL approach. While most metric-based FSL approaches utilize cosine distance computation in order to assess similarity between query vectors and the support data this again leads to requirement for high-precision computations [4, 15]. Utilizing pattern matching with low-precision computations such as Hamming Distance (HD), Euclidean Distance, etc. can lead to significant savings in energy at minor trade-offs in learning performance. Similar studies have been attempted in the past with focus on in-memory computing (IMC) [10, 12, 14, 18] however the datasets utilized by such studies were comparatively smaller with focus on implementing complete net-

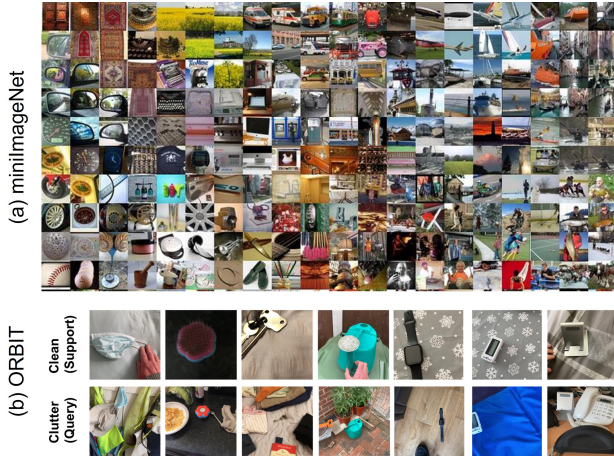


Figure 1. Example images from datasets used in the study: (a) miniImageNet [17], (b) ORBIT [4].

works at fixed precision thus limiting the scope for adaptability.

In this study, we perform detailed analysis both from algorithmic and implementation perspectives for our proposed binarized distance computing (BinDC) based FSL framework. To the best of our knowledge, this is the first work that simultaneously tries to address algorithmic aspects, implementation on embedded platforms as well as projections for IMC-based realization. Our key contributions and the novel aspects are:

1. Novel algorithm for performing BinDC utilizing floating-precision data vectors (converted to binarized representations) to perform matching for FSL applications.
2. Benchmarking of learning performance of the proposed method on miniImageNet [17] and ORBIT [13] datasets utilizing embeddings derived from prior work [4, 15].
3. Benchmarking of CPU-based implementation for the proposed method against cosine-based distance computations utilizing floating-point precision on an embedded platform to demonstrate savings in energy, latency and power.
4. Estimating performance gain in terms of latency and energy of proposed method when mapped to emerging NVM-based IMC platform as well as validation of basic operation in context of RRAM (resistive random access memory)-based IMC platform.

Table 1. Description of datasets used in the study.

Parameters	Dataset	
	miniImageNet	ORBIT
Categories	64 (train) / 20 (test)	486
Training	38400	2996 (videos)
Test	12000	826 (videos)
#Ways	5	Random
#Shots		

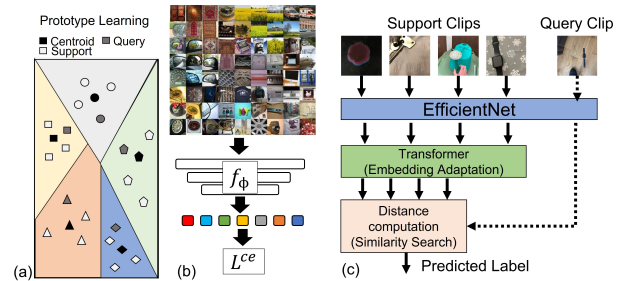


Figure 2. (a) Prototype learning representation in 2-D space. (b) Feature extraction pipeline for image classification task. (c) Feature extraction and inference pipeline used for object classification with videos.

2. Prior Art

2.1. Datasets for FSL

2.1.1 Image Classification

Image classification is the baseline task associated with computer vision applications. In context edge-AI with focus on FSL, researchers have work with a variety of datasets such as MNIST [11], OmniGlot [12], CIFAR100 [9] and miniImageNet [17]. For the current study we focus on using the miniImageNet dataset. It contains 100 classes with 600 images in each class, which are built upon the ImageNet dataset [2]. The 100 classes are divided into 64, 16, and 20 for meta-training, meta-validation, and meta-testing, respectively. Dataset samples from the same are shown in Fig. 1(a). Prototypical network architectures shown in Fig. 2(a,b) are typically utilized for this task and hence selected as the baseline for the study. To assess embeddings during more advanced techniques, we also explore the application of FEAT (few-shot embedding adaptation with Transformer) [19] for performing image classification.

2.1.2 Object Recognition from videos

As a more challenging task to perform real-world FSL, the ORBIT dataset was recently proposed [13]. It contains 3,822 videos of 486 objects collected by 67 users. Each user is asked to collect videos with the target object in isolation

which is referred as clean videos. Some videos also have the target object mixed with multiple other objects. These videos are referred as clutter videos. The goal of this to train a teachable object recognizer such that the model is personalized for each user using their clean videos. The personalized model is then evaluated on the clutter videos [13]. In the concept of meta-learning scenario, the clean videos are analogy as support set while the clutter videos are query set. To attempt this task we utilize the network architecture proposed by Li *et al.* [4], utilizing EfficientNet-BO [16] and FEAT [19] based computation pipeline as shown in Fig. 2(c).

2.2. Metric-based FSL

Prior work in the area of edge-computing for FSL has been heavily focussed on utilizing metric-based learning approaches. Metric-based methods rely on learning similarity between samples and perform classification by utilizing distance computations requiring low-computational overheads thus improving efficiency for FSL. Metric-based FSL has been realized using a variety of distance metrics such as L1-distance [11], cosine distance [17], Euclidean distance [15]. For more complex tasks innovative approaches utilizing transformer-based feature refinement combined with cosine distance computations were proposed [19]. Recent studies have successfully demonstrated utilization of IMC-based computations at low precisions (typically binary) [8, 10, 12, 14, 18]. However, most of these implementations focus on fixed-precision computations and limited-size datasets with low scope for generalization to wider datasets.

3. Proposed BinDC for FSL

Here, we propose a fully-binarized computational approach that not only reduces the storage requirement but also simplifies computational requirements by computing HD as the metric for similarity. Flow for the proposed method is summarized in Algorithm 1. The proposed technique performs normalization of both support and query data using min. and max. values from the support data. For this two methods can be adopted: (a) Norm. (normalization) method 1 where each feature channel uses the same parameters, (b) Norm. method 2 where we utilize feature channel-specific parameters for normalization. Post normalization vectors q_n and s_n are derived. Based on precision settings, thermometric encoding [1] is performed by first converting the normalized vectors to integer precision and then extracting binarized representations (with number of bits equal to original integer precision used during quantization). Although a single data point is still represented by multiple bits, it is important to note here that all the bits are independent functionally and hence can be used for single-bit (binary) computations. Feature matching is performed

Algorithm 1 Proposed BinDC method for FSL.

Require: Query vector QI, Support vector SI, Precision n, Feature Extractor f

Ensure: Match Index m

Pre-processing:

$$q_x = f(QI)$$

$$s_{min} = \min(SI)$$

$$s_{max} = \max(SI)$$

$$s_n = \frac{SI - s_{min}}{s_{max} - s_{min}}$$

$$q_n = \frac{q_x - s_{min}}{s_{max} - s_{min}}$$

$$r_n = 2^n - 1$$

$$r = \frac{2^n}{n}$$

$$s_i = \text{round}(s_n \times r_n)$$

$$q_i = \text{round}(q_n \times r_n)$$

for i=0; i<n; i=i++ **do**

$$s_x[:,i] = s_i > (r \times i + r - 1)$$

$$q_x[i] = q_i > (r \times i + r - 1)$$

end for

Similarity Search:

for k=0; k<len(SI); k=k++ **do**

$$d_x[k] = \text{popcount}(q_x \oplus s_x[k])$$

end for

$$m = \text{index}(\min(d_x))$$

by computing HD between the binarized support and query vectors with min. HD representing the perfect match. Conversely, it is possible to perform similar matching using XNOR operation in place of XOR used for computing HD in order to represent maximum response represent a match. Learning capabilities of the proposed technique as well as hardware benchmarking is explored in next sections.

4. Results and Discussion

4.1. Network Results

To validate the proposed method BinDC, we utilize two datasets described in Tab. 1. In order to assess the impact of choice of FSL model and backbone, t-SNE (t-distributed stochastic neighbor embedding) based representations are derived using samples from miniImageNet dataset as shown in Fig. 3. As can be observed, the combination of ProtoNet (Prototypical Networks) with ResNet12 represents a clean distribution which is further validated through assessment of inference accuracies presented in Tab. 2. We next analyzed the impact of proposed technique on the embedding space. Results from t-SNE analysis using multiple precisions with both normalization techniques discussed in Sec. 3 are shown in Fig. 4. Inference accuracy achieved using proposed method with two variants of FSL models and backbones across precisions utilizing both normalization techniques is summarized in Tab. 2. It can be clearly observed from the findings that ResNet12 and ProtoNet is the

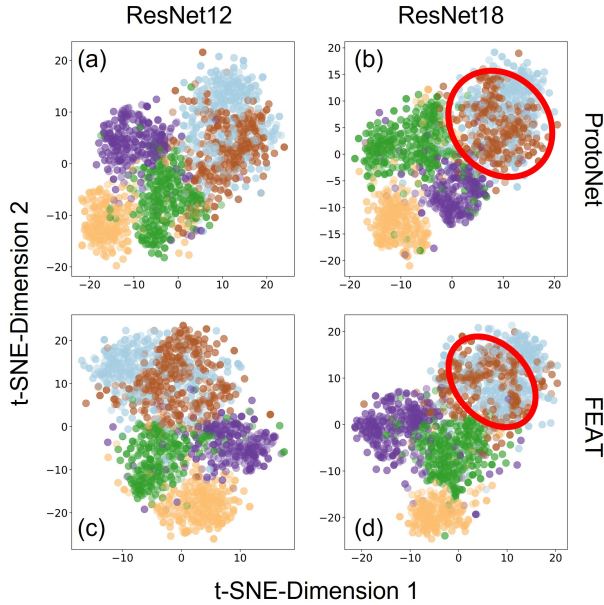


Figure 3. t-SNE based distributions of embeddings derived using combination of ProtoNet with (a) ResNet12,(b) ResNet18 backbones and FEAT with (c) ResNet12,(d) ResNet18 backbones. Red circles show class-wise confusion.

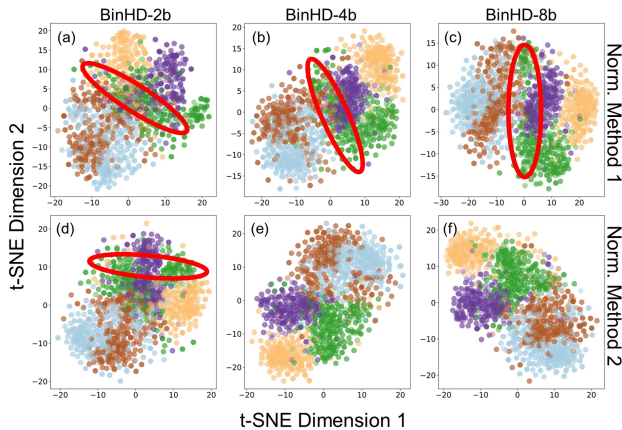


Figure 4. t-SNE based distributions of embeddings derived using proposed binarized embeddings with precisions of (a) 2-bit, (b) 4-bit, (c) 8-bit with normalization method 1. Corresponding results with normalization method 2 (d-f). Red circles show class-wise confusion.

best combination in terms of learning performance. Also Norm. method 2 provides superior accuracy even at very low precision ($> 50\%$ even with 2-bit) due to more channel-specific scaling within the embedding. Based on these findings, we then compared the learning performance across two different types of datasets as shown in Tab. 3. The proposed method achieves comparable accuracy to baseline

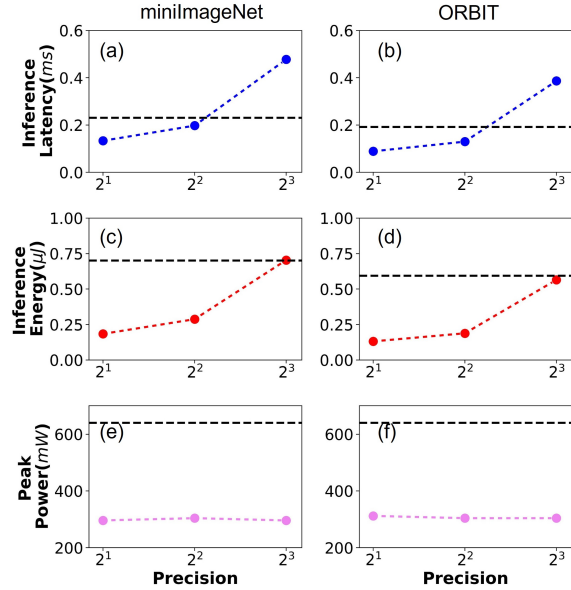


Figure 5. Measurement based results on Jetson Xavier NX platform (CPU-based) for the two workloads: (a,b) Inference latency, (c,d) Inference energy and (e,f) Peak power. Dashed black line shows the floating-point baseline utilizing cosine distance computations.

i.e. ‘cosine’ at 8-bit precision.

4.2. Benchmarking on Embedded Platforms

Based on inference results presented in Tabs. 2 and 3 it is apparent that 8-bit precision with proposed BinDC method is comparable to floating-point performance (within 2% for miniImageNet). Next, we evaluate the benefits of the proposed method from an implementation perspective on an embedded platforms. For this we utilize Jetson Xavier NX. Since logic operations are dominant in the proposed distance computation method, we make use of 4-CPU cores available on the Jetson Xavier NX platform utilizing the aarch64 instruction set. Workload for performing evaluation utilizes 10k samples from the dataset. Embeddings based on pre-trained networks in floating-point precision along with class labels are used as inputs. For cosine similarity, we utilize PyTorch framework while for BinDC we utilize a combination of NumPy with JIT (just-in-time) compilation to extract maximum efficiency in the implementation. Measurement results from the experiment are shown in Fig. 5. Latency and energy benefits (with reasonable accuracy) are apparent at 4-bit precision (see Fig. 5(a-d)). Power savings of upto $\approx 2\times$ can also be observed (see Fig. 5(e,f)) as a result of purely using integer precision and logic operations.

Table 2. Impact of normalization technique, FSL model and backbone on performance of proposed BinDC method for miniImageNet.

Distance Compute Method	Normalization Method	Bits/data point	Computation precision	Inference Accuracy (%)			
				ProtoNet		FEAT	
				Res12	Res18	Res12	Res18
Cosine		32	Float32	78.07	76.65	77.03	76.32
BinDC	1 (Global)	2	Binary	25.41	24.44	53.66	47.88
		4		68.23	55.19	41.06	22.88
		8		76.05	71.92	64.06	46.41
		16		77.56	75.37	69.84	63.05
	2 (Channel-specific)	2		58.29	50.61	58.93	54.47
		4		75.93	73.92	60.03	52.7
		8		77.14	75.34	68.45	59.28
		16		77.3	75.72	69.99	61.43

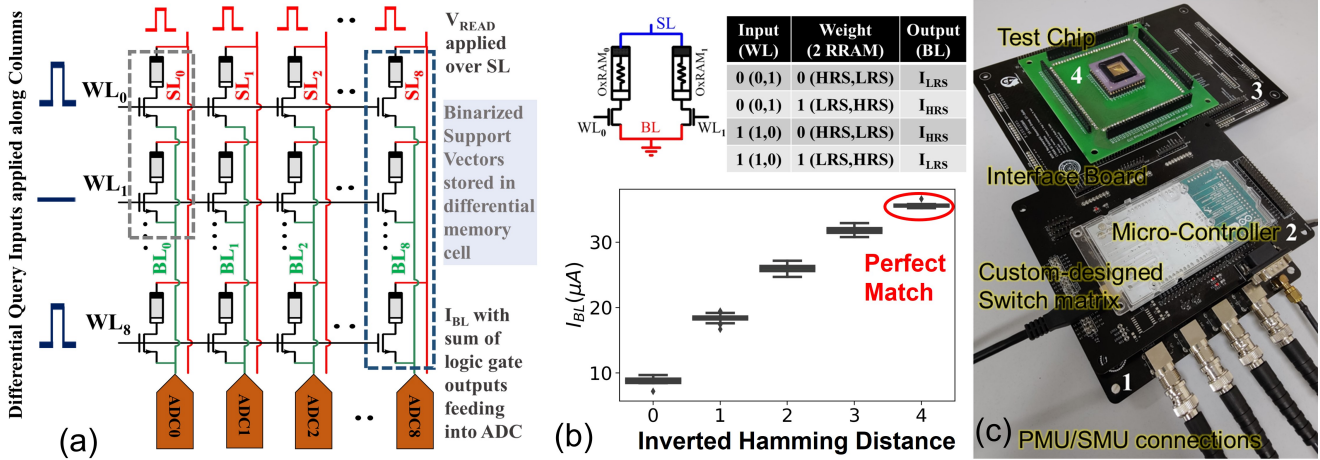


Figure 6. (a) Schematic of IMC-array showing mapping of binarized support vector bits and application of query inputs at the WL of the array. (b) I_{BL} as a function of computed HD. (c) Custom PCB used for performing experimental measurements with RRAM-based memory array for IMC applications.

Table 3. Benchmarking learning results based on proposed method against cosine distance at floating point precision for two workloads (Norm. method 2 for miniImageNet and Norm. method 1 for ORBIT).

Distance Method	Precision	Inference Accuracy (%)	
		miniImageNet	ORBIT
Cosine	Float	78.07	71.69
BinDC	2	58.29	39.18
	4	75.93	59.84
	8	76.69	65.71
	16	77.63	69.14

4.3. IMC-based optimizations

While utilizing standard digital hardware with the proposed method offers significant performance benefits, the vector dimensions used for FSL (640 for miniImageNet and 1280 for ORBIT) pose a limitation in terms of memory operations. Due to limited memory bandwidth, conventional compute architectures have energy and latency overheads [10]. IMC-based optimizations have been shown to lead to significant gains in terms of throughput and energy efficiency [18]. As a result, we perform experimental characterization of an RRAM-based IMC chip utilizing a customized PCB to evaluate benefits compared to conventional embedded hardware implementations.

The 8×8 1T-1R RRAM array used for experimental validation of IMC is shown in Fig. 6(a). Support vectors (SI) for performing FSL are stored in the form of RRAM device

Table 4. Benchmarking of performance for BinDC-based FSL with 4-bit encoding using OxRAM-based IMC implementations.

Energy Estimation Method	Device Data Reference	Technology Node (nm)	miniImageNet		ORBIT	
			Search Energy (nJ)	Inference time (μ s)	Search Energy (nJ)	Inference time (μ s)
Experiment	[10]	130	1.11	40	0.62	22.4
Simulated	[12]	40	0.67	20	0.37	11.2
	[3]	28	0.45	20	0.25	11.2

state along columns ('0' is encoded as top RRAM = HRS (high resistance state), bottom RRAM = LRS (low resistance state), while '1' is encoded as top RRAM = LRS, bottom RRAM = HRS). To realize XOR gate in hardware, a 2T-2R bitcell (see Fig. 6b) is realized by selecting two consecutive 1T-1R bitcells in the same column. Query Input vectors (QI) are applied as binary inputs ('0', '1') in a differential representation: '0' \rightarrow [0,1], and '1' \rightarrow [1,0] using a WL-decoder circuit. To perform computation, SL is charged to V_{READ} and QI is applied as input to corresponding 2T-2R bitcell. Output is obtained in the form of current flowing through corresponding BL (I_{BL}). When there is a mismatch between QI and SI for a given index, RRAM in HRS is selected and negligible current flows. In case of a match, RRAM in LRS is selected leading to higher I_{BL} . Following principles of KCL (Kirchoff's Current Law), output current of all XOR cells along a column can be integrated to compute HD between QI and SI. I_{BL} as a function of inverted HD, between a 4-bit QI vector and a 4-bit SI vector is shown in Fig. 6(c). Fig. 6(d) shows the custom experimental setup and RRAM test chip used in the study. Programming signals are applied using high speed pulse measurement unit (PMU) from semiconductor parameter analyzer (SPA). The signals from PMU channels are multiplexed and applied to different signal lines (WL,SL,BL) using the custom switch board.

Tab. 4 presents benchmarking of FSL using the proposed BinDC method with various RRAM technologies. The technology used in the current work being at an older prototype node exhibits high dissipation in terms of energy. However, using more scaled devices from literature at advanced technology nodes such as 28nm, EDP (energy delay product) savings of the order of 6300 \times were observed compared to CPU profiling (14nm) shown in Fig. 5.

5. Conclusion

In this work, we present a BinDC framework to perform distance computations for few-shot learning using only accumulation and logic operations (XOR/XNOR). The proposed method leads to marginal loss in accuracy of \approx 4% (for 4-bits). This leads to savings in memory (\approx 8 \times), energy (\approx 2.5-3 \times), power (\approx 2 \times) and latency (\approx 1.1-

1.5 \times) compared to a floating-point cosine distance computation when using CPU-based computations performed on an embedded platform. We further demonstrate realizations utilizing RRAM (resistive random access memory) based IMC (in-memory computing) to further improve EDP (\approx 1000 \times) in comparison to the embedded CPU-based realization. This can be further improved through technology scaling to achieve EDP savings of 6300 \times .

Acknowledgements

The PI Prof. Manan Suri would like to acknowledge the support of Meta Reality Labs Research.

References

- [1] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian J. Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 3
- [2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009. 2
- [3] L. Grenouillet, N. Castellani, A. Persico, V. Meli, S. Martin, O. Billoint, R. Segaud, S. Bernasconi, C. Pellissier, C. Jahan, C. Charpin-Nicolle, P. Dezest, C. Carabasse, P. Besombes, S. Ricavy, N.-P. Tran, A. Magalhaes-Lucas, A. Roman, C. Boixaderas, T. Magis, M. Bedjaoui, M. Tessaire, A. Seignard, F. Mazen, S. Landis, E. Vianello, G. Molas, F. Gaillard, J. Arcamone, and E. Nowak. 16kbit 1t1r oxram arrays embedded in 28nm fdsoi technology demonstrating low ber, high endurance, and compatibility with core logic transistors. In *IMW*, pages 1–4, 2021. 6
- [4] Li Gu, Zhixiang Chi, Huan Liu, Yuanhao Yu, and Yang Wang. Improving protonet for few-shot video object recognition: Winner of ORBIT challenge 2022. *CoRR*, abs/2210.00174, 2022. 1, 2, 3
- [5] Ramyad Hadidi, Jiashen Cao, Yilun Xie, Bahar Asgari, Tushar Krishna, and Hyesoon Kim. Characterizing the deployment of deep neural networks on commercial edge devices. In *IEEE International Symposium on Workload Char-*

- acterization, *IISWC 2019, Orlando, FL, USA, November 3-5, 2019*, pages 35–48. IEEE, 2019. [1](#)
- [6] Muhammad Huzaifa, Rishi Desai, Xutao Jiang, Joseph Ravichandran, Finn Sinclair, and Sarita V. Adve. Exploring extended reality with ILLIXR: A new playground for architecture research. *CoRR*, abs/2004.04643, 2020. [1](#)
- [7] Indhumathi Kandaswamy, Saurabh Farkya, Zachary Daniels, Gooitzen van der Wal, Aswin Raghavan, Yuzheng Zhang, Jun Hu, Michael Lomnitz, Michael Isnardi, David Zhang, and Michael Piacentino. Real-time hyper-dimensional reconfiguration at the edge using hardware accelerators. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3609–3617, 2022. [1](#)
- [8] Geethan Karunaratne, Manuel Le Gallo, Giovanni Cherubini, Luca Benini, Abbas Rahimi, and Abu Sebastian. In-memory hyperdimensional computing. *Nature Electronics*, 3(6):327–337, jun 2020. [3](#)
- [9] G. Karunaratne, M. Hersche, J. Langeneager, G. Cherubini, M. Le Gallo, U. Egger, K. Brew, S. Choi, I. Ok, C. Silvestre, N. Li, N. Saulnier, V. Chan, I. Ahsan, V. Narayanan, L. Benini, A. Sebastian, and A. Rahimi. In-memory realization of in-situ few-shot continual learning with a dynamically evolving explicit memory. In *ESSCIRC 2022- IEEE 48th European Solid State Circuits Conference (ESSCIRC)*, pages 105–108, 2022. [2](#)
- [10] Sandeep Kaur Kingra, Vivek Parmar, Deepak Verma, Alessandro Bricalli, Giuseppe Piccolboni, Gabriel Molas, Amir Regev, and Manan Suri. Fully binarized, parallel, rram-based computing primitive for in-memory similarity search. *IEEE Trans. Circuits Syst. II Express Briefs*, 70(1):46–50, 2023. [1](#), [3](#), [5](#), [6](#)
- [11] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015. [2](#), [3](#)
- [12] Haitong Li, Wei-Chen Chen, Akash Levy, Ching-Hua Wang, Hongjie Wang, Po-Han Chen, Weier Wan, Win-San Khwa, Harry Chuang, Y.-D. Chih, Meng-Fan Chang, H.-S. Philip Wong, and Priyanka Raina. Sapiens: A 64-kb rram-based non-volatile associative memory for one-shot learning and inference at the edge. *IEEE Transactions on Electron Devices*, 68(12):6637–6643, 2021. [1](#), [2](#), [3](#), [6](#)
- [13] Daniela Massiceti, Luisa M. Zintgraf, John Bronskill, Lida Theodorou, Matthew Tobias Harris, Edward Cutrell, Cecily Morrison, Katja Hofmann, and Simone Stumpf. ORBIT: A real-world few-shot dataset for teachable object recognition. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 10798–10808. IEEE, 2021. [2](#), [3](#)
- [14] Kai Ni, Xunzhao Yin, Ann Franchesca Laguna, Sidharth Joshi, Stefan Dünkler, Martin Trentzsch, Johannes Müller, Sven Beyer, Michael Niemier, Xiaobo Sharon Hu, and Suman Datta. Ferroelectric ternary content-addressable memory for one-shot learning. *Nature Electronics*, 2(11):521–529, nov 2019. [1](#), [3](#)
- [15] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. [1](#), [2](#), [3](#)
- [16] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019. [3](#)
- [17] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. [2](#), [3](#)
- [18] Tony F. Wu, Haitong Li, Ping-Chen Huang, Abbas Rahimi, Gage Hills, Bryce Hodson, William Hwang, Jan M. Rabaey, H.-S. Philip Wong, Max M. Shulaker, and Subhasish Mitra. Hyperdimensional computing exploiting carbon nanotube fets, resistive ram, and their monolithic 3d integration. *IEEE JSSC*, 53(11):3183–3196, 2018. [1](#), [3](#), [5](#)
- [19] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8805–8814, 2020. [2](#), [3](#)