

Navigating the Unknown: Uncertainty-Aware Compute-in-Memory Autonomy of Edge Robotics

Nastaran Darabi, Priyesh Shukla, Dinithi Jayasuriya, Divake Kumar, Alex C. Stutts, and Amit Ranjan Trivedi
AEON Lab, University of Illinois Chicago (UIC), Chicago, IL, Email: amitrt@uic.edu

Abstract—This paper addresses the challenging problem of energy-efficient and uncertainty-aware pose estimation in insect-scale drones, which is crucial for tasks such as surveillance in constricted spaces and for enabling non-intrusive spatial intelligence in smart homes. Since tiny drones operate in highly dynamic environments, where factors like lighting and human movement impact their predictive accuracy, it is crucial to deploy uncertainty-aware prediction algorithms that can account for environmental variations and express not only the prediction but also *confidence in the prediction*. We address both of these challenges with Compute-in-Memory (CIM) which has become a pivotal technology for deep learning acceleration at the edge. While traditional CIM techniques are promising for energy-efficient deep learning, to bring in the robustness of uncertainty-aware predictions at the edge, we introduce a suite of novel techniques: *First*, we discuss CIM-based acceleration of Bayesian filtering methods uniquely by leveraging the Gaussian-like switching current of CMOS inverters along with co-design of kernel functions to operate with extreme parallelism and with extreme energy efficiency. *Secondly*, we discuss the CIM-based acceleration of variational inference of deep learning models through probabilistic processing while unfolding iterative computations of the method with a compute reuse strategy to significantly minimize the workload. *Overall*, our co-design methodologies demonstrate the potential of CIM to improve the processing efficiency of uncertainty-aware algorithms by orders of magnitude, thereby enabling edge robotics to access the robustness of sophisticated prediction frameworks within their extremely stringent area/power resources.

Index Terms—Compute-in-memory, Edge robotics, Autonomous Navigation, Visual Odometry, Particle Filtering

I. INTRODUCTION

At the core of the self-navigation of autonomous drones and robots lies the fundamental task of continuously ascertaining their position and orientation, often referred to as their pose estimation [1], [2]. This task is indispensable for various autonomy tasks such as path planning and obstacle avoidance. Furthermore, pose estimation must remain reliable and resilient even under dynamic changes to the vehicle's operational environment. For instance, in indoor settings, the environment is subjected to ongoing alterations, including the movement of people and fluctuations in lighting conditions which significantly impact the accuracy and effectiveness of drone localization [3]. Moreover, in Fig. 1, computing uncertainties also arise due to non-idealities (such as process variability) and noise in the underlying hardware. For robotics applications where perception-action loops are tightly coupled, these various sources of uncertainties can intricately interact

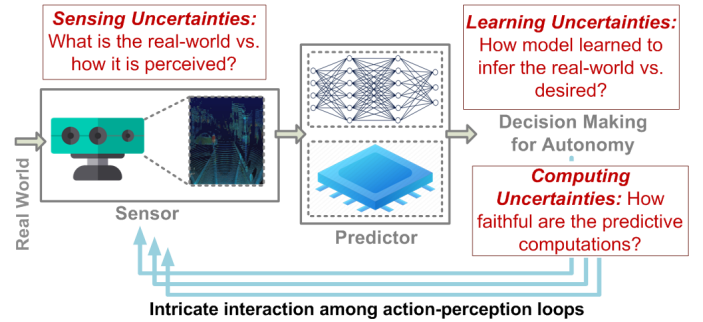


Fig. 1. Various sources of uncertainties arise in typical autonomous systems such as the inability to perceive the application domain reliably, discrepancies in the trained and applied environment, and sources of non-idealities such as noise and process variability impacting the reliability of processing.

and can dramatically degrade the quality of predictions. Therefore, techniques that not only provide real-time pose estimation but also account for various uncertainties in their predictions are imperative for risk-aware decision-making [4], [5].

Simultaneously, there is an ongoing imperative to reduce the computational and storage requirements of autonomous operations while maintaining predictive robustness in emerging applications, particularly for *autonomous insect-scale drones*. Remarkable progress has been achieved in the field of drone miniaturization, as evidenced by various studies [6], [7], resulting in drones that weigh less than a gram and are smaller than a penny! However, as these drones shrink in size, their payload capacity is inherently limited, often constrained to accommodate only a tiny battery. Consequently, it becomes crucial to minimize the power consumption associated with on-board processing tasks. Although leveraging cloud resources for offloading on-board processing is an option, many environments present challenges with limited or no connectivity. Furthermore, the continuous transmission of high-resolution camera data to the cloud can also incur significant costs.

Compute-in-memory (CIM) has emerged as a promising technology for advancing ultra-low-power deep learning by integrating memory and processing in the same module, thereby significantly reducing the energy and time required for data movement between separate memory and processor units, which is a major bottleneck for deep learning computations in conventional computing architectures. Moreover, CIM-based deep learning can also minimize necessary processing steps by *leveraging physics for computing*. For example, based on charge or current-domain representation of operands, additions of weight-input product terms can be realized simply on a

This work was supported in part by COGNISENSE, one of seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, and NSF CAREER Award #2046435.

wire by leveraging Kirchoff's law [8], [9]. While CIM has revolutionized deterministic deep learning, its application in probabilistic and Bayesian inference remains largely untapped. As edge intelligence increasingly proliferates to scenarios demanding risk-averse, real-time predictions—such as in autonomous vehicles—the need for not just resource efficiency but also heightened robustness becomes paramount.

In this article, we explore the significant potential of extending CIM to Bayesian and probabilistic inference, applicable to both deep learning and conventional signal processing algorithms where various sources of predictive uncertainties can be systematically accounted for. At the heart of our various proposals is a co-design methodology, which synergistically optimizes both the underlying hardware and computational models. This approach is crucial in maximizing the efficacy of the overall computing substrate, thereby unlocking new possibilities for intelligent, robust computing.

Our work makes the following key contributions:

- We discuss CIM processing of particle filtering for drone localization, which employs a probabilistic approach to accurately determine a drone's position and orientation by continuously updating a set of hypotheses about its state. While the workload of particle filtering can be significant for complex and extensive environments, we demonstrate how the Gaussian-like switching current of CMOS inverters, along with a co-design of map models on CIM substrate can be leveraged to remarkably minimize the resources required for large-scale particle filtering.
- We also discuss the CIM-based acceleration of Bayesian Deep Learning (BDL). In particular, exploring a CIM-based implementation of a variational inference procedure for BDL, called MC-Dropout, we demonstrate the unique opportunities for CIM to considerably minimize the workload of the Monte Carlo-based inference procedure. The discussed CIM modules not only store and operate on weight matrices, but also emulate sampling-based iterations of BDL by leveraging noise sources. Furthermore, our CIM-based BDL processing leverages opportunities such as compute reuse and optimal sample ordering, to significantly minimize the resource demands of BDL—bringing its requirements close to the energy/resource budget of edge devices.

In essence, the paper compiles a range of insights derived from our previous research [4], [10]–[13] and underscores the significance of co-designing CIM with Bayesian algorithms as a versatile framework capable of substantially augmenting their implementation efficiency. Sec. II discusses CIM-based particle filtering for ultra-low-power drone localization by employing an iterative, reasoning-based prediction procedure. Sec. III presents CIM-based acceleration of MC-Dropout and characterization on visual odometry (VO). Sec. IV concludes.

II. COMPUTE-IN-MEMORY PARTICLE FILTERING-BASED DRONE LOCALIZATION

A. Overview of Particle Filtering-based Localization

By applying Bayes' rule, the posterior probability of the drone's pose (\mathbf{x}) is given as $P(\mathbf{x}|\mathbf{z}) = \frac{P(\mathbf{z}|\mathbf{x})P(\mathbf{x})}{P(\mathbf{z})}$. Here,

$P(\mathbf{z}|\mathbf{x})$ represents the measurement likelihood, and $P(\mathbf{x})$ represents the prior probability of the pose. The Maximum Likelihood Estimate (MLE) of the drone's pose seeks to maximize the likelihood of the current measurement, i.e., $\mathbf{x}_{\text{MLE}} = \text{argmax } P(\mathbf{z}|\mathbf{x})$. On the other hand, the Maximum A Posteriori Estimate (MAP) of the pose aims to maximize the posterior probability of the pose, i.e., $\mathbf{x}_{\text{MAP}} = \text{argmax } P(\mathbf{z}|\mathbf{x})P(\mathbf{x})$.

When the drone has complete uncertainty about its pose during global localization, MAP and MLE estimates are equivalent. However, MAP estimation provides a systematic way to incorporate prior beliefs when there is partial pose knowledge. In probabilistic localization, the drone can systematically reduce estimation uncertainties by recursively integrating more measurements using a recursive Bayes rule:

$$\overline{bel}(\mathbf{x}_t) = \sum_{\mathbf{x}_{t-1}} P(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1}) bel(\mathbf{x}_{t-1}) \quad (1a)$$

$$bel(\mathbf{x}_t) = \eta P(\mathbf{z}_t|\mathbf{x}_t) \overline{bel}(\mathbf{x}_t) \quad (1b)$$

In Eq. (1a), $\overline{bel}(\mathbf{x}_t)$ represents the drone's updated pose belief at iteration t , and $bel(\mathbf{x}_{t-1})$ is the pose belief from the previous iteration $t-1$. \mathbf{u}_t denotes the control input applied to the drone, and $P(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1})$ is the probabilistic motion model, which leverages the Markov assumption, implying that the belief of \mathbf{x}_t depends only on the previous state's belief and the most recent control input \mathbf{u}_t .

Particle Filtering, as described in [14], offers a practical way to implement Bayesian filtering when dealing with arbitrary likelihood and belief function profiles through a Monte Carlo approach. Instead of relying on analytically defined likelihood and belief functions, this method involves considering a collection of hypotheses, often referred to as particles, that are sampled based on the underlying density functions. Each particle is assigned a weight, which is updated based on measurements using the equations mentioned earlier. As measurements are acquired, particles with low likelihood values are systematically removed. To account for various sources of uncertainty, such as the drone's motion control, a new set of particles is regenerated from the current set, which represents various uncertainties such as uncertainties of motion control.

B. Posterior Density Estimation with an Inverter Array

A commonly used method for obtaining $P(\mathbf{z}|\mathbf{x})$ involves the utilization of a Gaussian mixture model (GMM) to represent the 3D map [15]. By employing 3D scanning devices like the Microsoft Kinect, point-cloud data of operating domains is captured and fitted with a GMM. To accurately capture the intricacies of the 3D map, it is essential to employ an adequate number of mixture components within the GMM [16].

In a typical assessment of pose localization, the computation of the likelihood, denoted as $p_{\text{GMM}}(\mathbf{z}_{\text{proj}}(\mathbf{x}, \mathbf{z}); \Theta)$, involves numerous non-zero depth pixels and a substantial number of mixture components within the GMM model. Consequently, the computational burden associated with calculating $P(\mathbf{z}|\mathbf{x})$ becomes quite substantial. Additionally, considering that drones must continually determine their position during flight, there are strict timing constraints placed on the evaluation of $P(\mathbf{z}|\mathbf{x})$. In

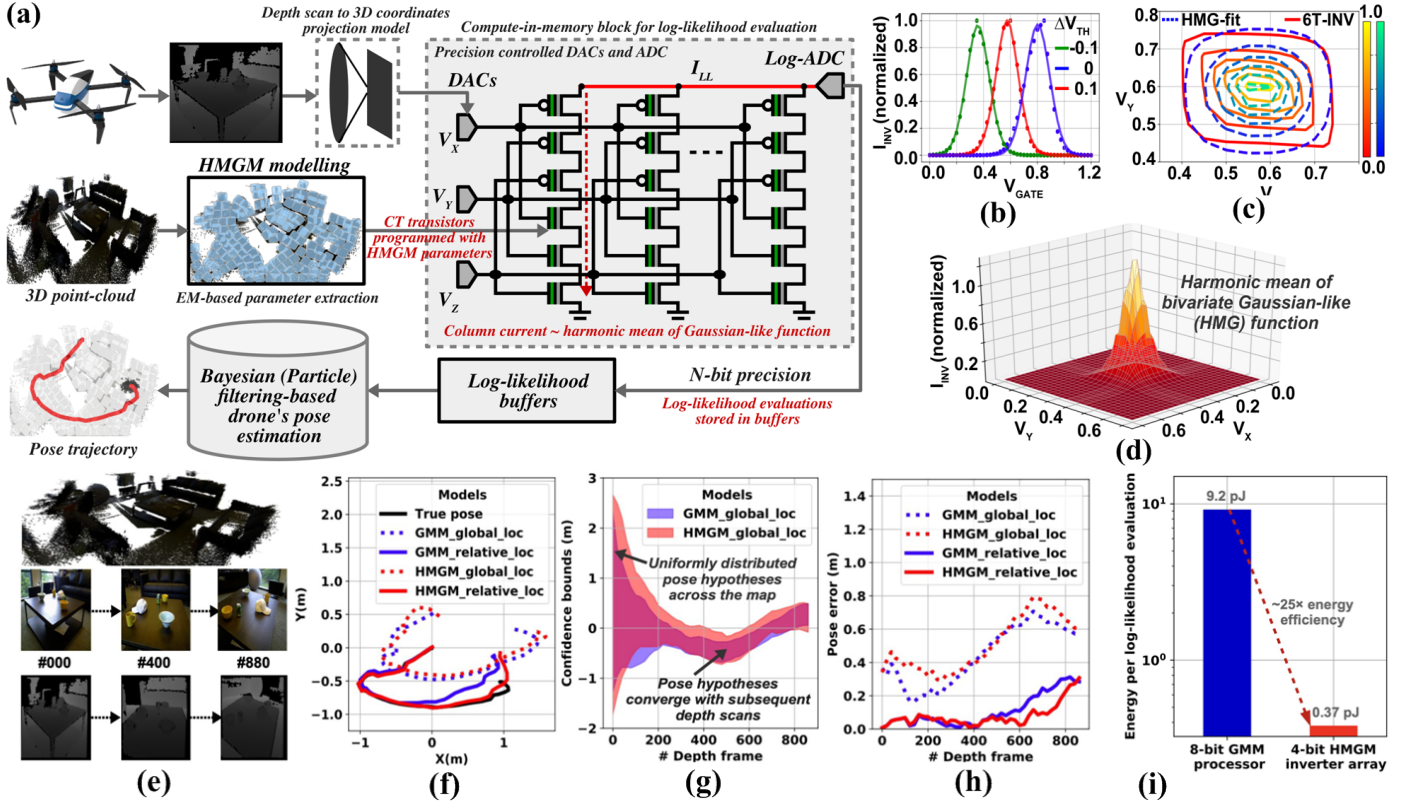


Fig. 2. **Compute-in-memory-based ultra-low-power Monte-Carlo localization framework for insect-scale drones.** (a) CIM architecture employs columns of 6-T inverters to leverage their switching current for likelihood evaluations. (b) Gaussian-like switching current of 6-T inverters. (c) Contour and (d) surface plot of switching current at varying input voltages showing rectilinear tails of the implemented function whereas a Gaussian function has elliptical tails. On RGB-D dataset (e), (f)–(h) show the comparison of localization steps between the proposed co-designed HMGM-based approach vs. conventional GMM-based approach. (i) Comparison of energy between an 8-bit digital processor and 4-bit HMGM inverter array-based CIM. [Reproduced from our prior work [10] under IEEE’s open-source license agreement.]

the following, we discuss how CIM can significantly minimize these processing overheads by exploiting a novel co-design approach.

Consider the design of a six-transistor inverter controlled by three input voltages: V_X , V_Y , and V_Z , as illustrated in Fig. 2(a), shown within the highlighted box. The transistors within the inverter incorporate a floating gate mechanism for programming their threshold voltage in a non-volatile manner, such as using charge trap transistor mechanisms [17]. The programmability of the inverter array is achieved by adjusting the charge density within the floating gate. In Fig. 2(b), the current flowing through the series-connected transistors within the inverter, denoted as I_{INV} , effectively emulates a Gaussian-like function. The plot illustrates the behavior of I_{INV} when one of the input voltages, namely V_X , V_Y , or V_Z , is varied while keeping the other fixed.

Figs. 2(c, d) depicts contour and surface plots of I_{INV} as two gate voltages vary simultaneously. The overall current of a multi-input inverter can be approximated as $1/(1/I_{INV,1} + 1/I_{INV,2} + 1/I_{INV,3})$, which is equivalent to the harmonic mean of the constituent inverters’ currents. When controlling the inverter’s current with multiple input voltages, the characteristics of I_{INV} are more accurately represented by a Harmonic Mean of Gaussian-like (HMG) function, where each constituent 1D function is influenced by input voltages V_X , V_Y ,

and V_Z . Notably, compared to implementing and evaluating a multivariate Gaussian function using a digital datapath, the approach presented in Fig. 2(a) is significantly simpler. In a digital design, evaluating a multivariate Gaussian requires digital components such as multipliers, adders, and lookup tables for exponential or log-ADD functions. In contrast, the HMG function is realized with just six transistors and can be evaluated by applying analog voltages V_X , V_Y , and V_Z .

C. Compute-in-Memory Particle Filtering

As discussed in our prior work [10], based on the current pose belief, the scan z of N non-zero depth map pixels $\{z_1, z_2, \dots, z_N\}$ is projected to 3D via the camera’s projection model which determines the inputs V_X , V_Y , and V_Z applied at the inverter array in Fig. 2(a). Evaluating the likelihood of projections thereby determines the likelihood of the pose-belief, i.e., how likely the current belief on the drone’s pose is based on the measurements and flying domain’s map model. In a typical evaluation of pose localization, the likelihood is computed for hundreds of non-zero depth pixels and hundreds of mixture functions in the GMM model; therefore, has an excessive workload. Meanwhile, using the above methodology where map models are learned using a mixture of HMGs and implemented on a floating gate inverter array, the likelihood of

each projection can be readily implemented. The total current from the parallel inverters effectively emulates the likelihood of measurements applied at the gates. This net current is then converted into the digital domain using a logarithmic Analog-to-Digital Converter for post-processing.

Using the simplified co-design approach, Figs. 2(e-h) compares the prediction accuracy of drone localization with our HMG mixtures (HMGM)-based approach against the conventional GMM-based model on the RGB-D Scenes Dataset v2 [18]. Notably, the co-designed approach achieves a matching accuracy to the conventional approach. Although CIM-based implementation necessitates data converters, ADC and DAC, the overhead associated with these components amortizes efficiently when applied in a larger-scale processing architecture consisting of numerous parallel columns. By leveraging these unique characteristics of our CIM hardware and co-design methodologies, in our prior work [10] and Fig. 2(i), the total energy consumption (with 500 inverter columns emulating 100 mixture components) for the likelihood estimation was estimated to be 374 fJ (at 45 nm CMOS process technology), which is $25\times$ lower than an 8-bit GMM processor.

III. COMPUTE-IN-MEMORY UNCERTAINTY-EXPRESSIVE BAYESIAN VISUAL ODOMETRY

A. Overview of Visual Odometry (VO)

In Visual Odometry (VO), the camera's ego motion is determined by analyzing changes in visual data captured during its operation. Traditional techniques relied on recognizing and tracking distinctive visual features like corners or edges in the camera's field of view. Recent advancements in VO involve the utilization of deep neural networks (DNN) to directly learn the relationship between the camera's visual input and its ego-motion from empirical data. PoseNet [1] pioneered the training of convolutional neural networks for end-to-end tracking of monocular camera ego-motion. PoseLSTM [19] leveraged long-short-term memories (LSTM) to enhance ego-motion accuracy compared to frame-based approaches such as PoseNet. DeepVO [20] showcased the power of recurrent convolutional neural networks (RCNNs) in capturing features and modeling sequential dependencies between successive camera frames. UnDeepVO [21] introduced unsupervised deep learning technique for recovering absolute scale. VLocNet [22], [23] presented a multitask monocular model that leverages parameter sharing and auxiliary learning for VO.

B. Bayesian and Variational Inference of Neural Networks

While most deep learning-based frameworks for VO result in a point prediction of the vehicle's pose and cannot express their prediction confidence, to improve the predictive robustness, a Bayesian approach to deep learning's inference is gaining attention which can determine both the point estimate of pose as well as confidence on the prediction. Notably, in contrast to classical neural networks, where network weights and predictions are deterministic point estimates, Bayesian neural networks (BNNs) represent network weights as a statistical distribution, ushering in two key advantages over their classical

counterparts. *Firstly*, BNNs rest on a more solid theoretical foundation by considering a distribution of weights. Learning from data is inherently an ill-posed problem and may yield an infinite number of possible weight solutions. Classical neural networks rely on a single-point estimate for weights, overlooking other potential weight combinations. In contrast, BNNs explore numerous weight configurations, leading to more robust predictions. *Secondly*, unlike classical neural networks, BNNs can express the confidence of their predictions. BNN predictions take the form of a probabilistic distribution, with higher-order moments like variance providing insight into the prediction confidence.

However, the predictive robustness of BNNs also demands an overwhelming number of computations when compared to a classical neural network. A practical implementation of BNN operates by sampling many weights from the posterior weight distribution. At each sample, the network output is computed and weighed against the sample probability. The cumulative output is obtained only after a sufficiently large number of samples ($\sim 100 - 1000$) are extracted and the network output at each sample is computed. Since the number of weights in even a moderate-sized network ranges from hundreds to tens of thousands, both sampling in the high-dimensional weight space and the output extraction of each sample is computationally expensive. Hence, without speeding up the calculations, BNN remains inaccessible for real-time applications.

To minimize the complexity of Bayesian inference, a variational approximation of the posterior can be utilized. In Variational Inference (VI), a family of approximate probability density functions, denoted as \mathcal{F} , is considered. Within this family, a specific member $q^*(w)$ is sought that minimizes the Kullback-Leibler (KL) divergence from the true posterior distribution. Subsequently, the posterior density $P(w | D)$ is approximated using this optimal member $q^*(w)$. Previous studies [24] have discussed algorithms for obtaining $q^*(w)$. However, a critical requirement is to select a flexible family \mathcal{F} that allows $q^*(w)$ to closely approximate the true posterior.

C. Compute-in-Memory Monte-Carlo Dropout-based

Monte Carlo Dropout (MC-Dropout), an adaptation of variational inference, was introduced in [25] and is adopted here for CIM-based acceleration. In a dropout (DO) layer, input and output neurons are randomly excluded in each iteration, following a Bernoulli distribution. Employing MC-dropout layers with a dropout probability of 0.5 has been demonstrated to effectively capture model uncertainties for robust inference across various applications [25], [26]. Additionally, in [27], authors have devised learning methods to determine optimal dropout probabilities from training data. Predictions in this MC-Dropout-based inference procedure involve averaging model outputs for regression tasks and majority voting for classification tasks. Model confidence is extracted by estimating the variance of each iteration's output.

While CIM has shown tremendous benefits for high-dimensional scalar products of weight matrices and input vectors, for probabilistic inference procedures such as MC-Dropout, additional capabilities are needed. In Fig. 3(a), to

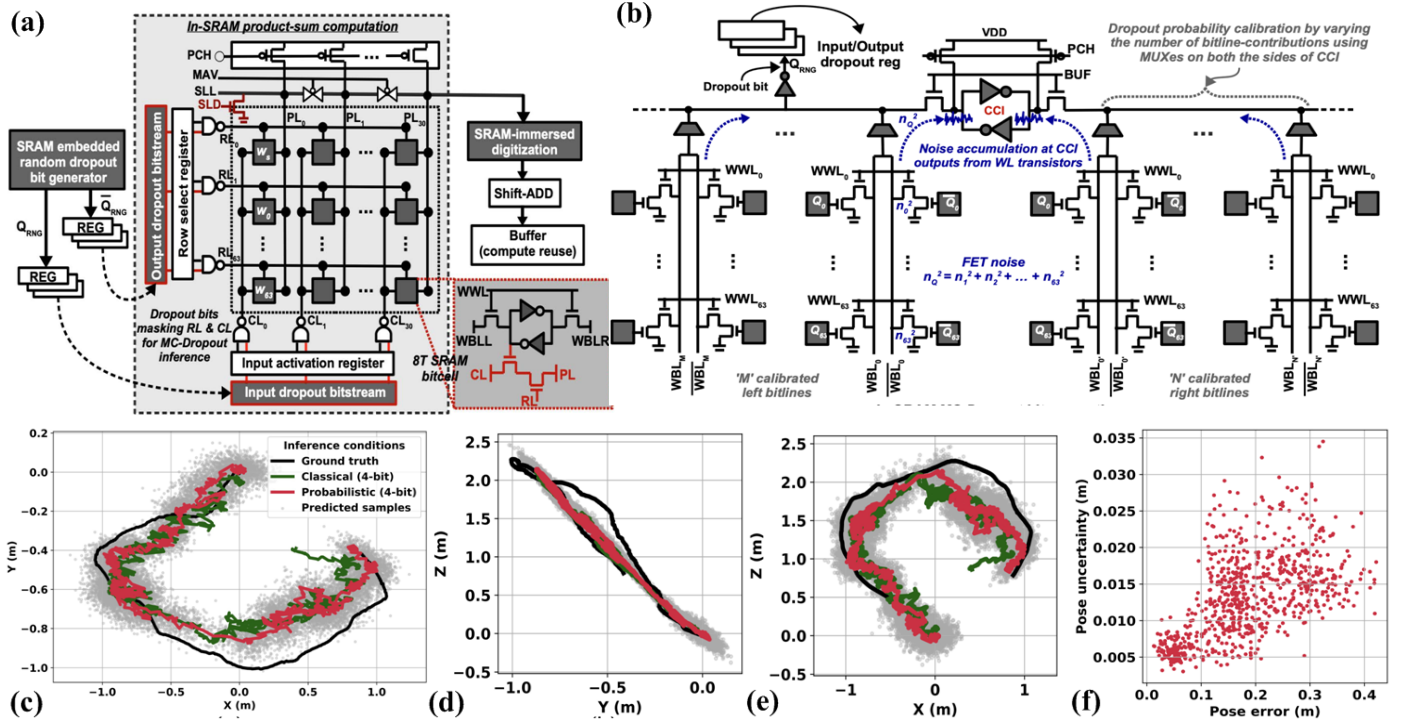


Fig. 3. **Compute-in-memory-based acceleration of variational inference of neural networks:** (a) An SRAM-based CIM macro combines storage and Bayesian inference (BI) functionalities. The inset figure illustrates an 8T SRAM cell featuring both storage and product ports. The CIM incorporates a randomly generated dropout bit generator, enabling MC-Dropout inference. (b) CIM-embedded dropout bit generator. MC-Dropout for indoor (RGB-D dataset) drone localization and trajectory estimation in (c) X-Y, (d) Y-Z, and (e) X-Z position coordinates. We also compare these results with deterministic network configurations under various inference conditions. (f) The correlation between uncertainty (variance) and the error in the pose estimates. [Reproduced from [11] with IEEE’s permission]

facilitate random input dropouts, the inputs to CL peripherals undergo an AND operation with a dropout bitstream. Similarly, for random output dropouts, row activations are masked by ANDing RL signals with an output dropout bitstream. Consequently, in MC-Dropout, an extra step is required to generate dropout bits for each applied input vector, making the high-speed generation of these dropout bit vectors a critical overhead for CIM-based MC-Dropout [11].

Fig. 3(b) illustrates the proposed SRAM-embedded random number generator (RNG), which takes advantage of SRAM’s write parasitics for SRAM-immersed generation of dropout bits. During inference, write wordlines (WWL) to a CIM macro are deactivated. Consequently, along a column, each write port injects leakage and noise current into the bit line. Despite variations in leakage current from each port, denoted as $I_{leak,ij}$, due to threshold voltage (V_{TH}) mismatches, a notable phenomenon occurs. The cumulative leakage current across parallel ports reduces the overall sensitivity of the net leakage current at the bit lines. In other words, $\sum_i I_{leak,ij}$ exhibits lower sensitivity to V_{TH} mismatches. Each write port also contributes noise current, $I_{noi,ij}$, to the bit line. Since the noise current from each port varies independently, the net noise current is amplified.

We exploit this filtering of process-induced mismatches and magnification of noise sources at the bit lines to implement a within-SRAM cross-coupled inverter (CCI)-based random number generator (RNG). In Fig. 3(b), an equal number of

SRAM columns are connected to both ends of the CCI. Both bit lines (BL and $\bar{B}L$) of a column are connected to the same end to cancel out the effect of column data. Both ends of the CCI are precharged using a precharge (PCH) signal and then discharged using column-wise leakage currents for half a clock cycle at the clock transition, pull-down transistors are activated using a delayed PCH to generate the dropout bit. For calibration, the CCI generates a fixed number of output random bits serially, allowing for the estimation of its bias.

Moreover, since two consecutive iterations of MC-Dropout share some input/output neurons, this can allow for a substantial reduction in computational effort by iteratively calculating the sum of products in each iteration using $P_i = P_{i-1} + W \times I_i^A - W \times I_i^D$. Here, I_i^A represents the input neurons that are activated in the current iteration but were inactive in the preceding one, while I_i^D signifies the input neurons that were active in the preceding iteration but are inactive in the current one. Employing this strategy enables the system to conduct only the new calculations while efficiently reutilizing the computations that overlap between iterations. In our prior work [11], we further discussed how the processing flow order can be optimized to even more substantially reduce the necessary processing.

D. Uncertainty Expressive Visual Odometry (VO)

Figs. 3(c-e) demonstrates the performance of the above framework for VO while utilizing the RGB-D Scenes Dataset v2 [18]. The comparison between ground truth and estimated

pose trajectories highlights that even with very low precision, probabilistic inference can accurately track the ground truth. Fig. 3(f) shows a scatter plot correlating pose error with variance (uncertainty) in probabilistic inference where there is a discernible correlation between error and predictive uncertainty. This indicates that, in contrast to classical (deterministic) inference, CIM's probabilistic inference can signal potential mispredictions by exhibiting high predictive uncertainties. In our prior work [11], we also characterized the energy efficiency of the above framework while operating at a clock frequency of 1 GHz and a voltage of 0.85V, and designing the system at the 16 nm CMOS node. The developed framework was benchmarked to operate at 3.04 TOPS/W (at 4-bit precision) and around 2 TOPS/W (at 6 bits) for 30 MC-Dropout network iterations. Such high efficiency throughout of Monte Carlo-based Bayesian framework demonstrates the potential of predictive uncertainty-aware robust inference at the edge.

IV. CONCLUSIONS AND FUTURE WORKS

We have presented ultra-low-power CIM-based prediction frameworks that effectively express uncertainty, such as CIM-based particle filtering for drone localization. Additionally, we explored CIM-based processing of variational inference in deep learning models, harnessing intrinsic shot noise for robust probabilistic processing. Central to both frameworks is our co-design methodologies that adapts the traditional computing models to CIM's operating and physical constraints for significant simplicity and parallelism in processing while leveraging analog mode representations to minimize the workload to begin with. However, a notable challenge in these uncertainty-expressive prediction frameworks is their reliance on Monte Carlo methods, which can be complex and resource-intensive. To address this, future research could investigate Monte Carlo-free uncertainty-aware frameworks like conformal inference [12], [28], [29] and evidential learning [30] suited for the edge.

REFERENCES

- [1] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.
- [2] N. Darabi, S. Tayebati, S. Ravi, T. Tulabandhula, A. R. Trivedi *et al.*, "Starnet: Sensor trustworthiness and anomaly recognition via approximated likelihood regret for robust edge autonomy," *arXiv preprint arXiv:2309.11006*, 2023.
- [3] P. Skrzypczyński, "Mobile robot localization: Where we are and what are the challenges?" *Automation 2017: Innovations in Automation, Robotics and Measurement Techniques 1*, pp. 249–267, 2017.
- [4] P. Shukla, S. Sureshkumar, A. C. Stutts, S. Ravi, T. Tulabandhula, and A. R. Trivedi, "Robust monocular localization of drones by adapting domain maps to depth prediction inaccuracies," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [5] D. Parente, N. Darabi, A. C. Stutts, T. Tulabandhula, and A. R. Trivedi, "Conformalized multimodal uncertainty regression and reasoning," *arXiv preprint arXiv:2309.11018*, 2023.
- [6] Y.-H. Hsiao, S. Kim, Z. Ren, and Y. Chen, "Heading control of a long-endurance insect-scale aerial robot powered by soft artificial muscles," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3376–3382.
- [7] M. L. Afakh, H. Sato, and N. Takesue, "A study towards a flapping robot maintaining attitude during gliding," *International Journal on Advanced Science, Engineering & Information Technology*, vol. 13, no. 2, 2023.
- [8] S. Nasrin, M. B. Hashem, N. Darabi, B. Parpillon, F. Fahim, W. Gomes, and A. R. Trivedi, "Memory-immersed collaborative digitization for area-efficient compute-in-memory deep learning," in *2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2023, pp. 1–5.
- [9] S. Yu, H. Jiang, S. Huang, X. Peng, and A. Lu, "Compute-in-memory chips for deep learning: Recent trends and prospects," *IEEE circuits and systems magazine*, vol. 21, no. 3, pp. 31–56, 2021.
- [10] P. Shukla, A. Muralidhar, N. Iliev, T. Tulabandhula, S. B. Fuller, and A. R. Trivedi, "Ultralow-power localization of insect-scale drones: Interplay of probabilistic filtering and compute-in-memory," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 1, pp. 68–80, 2021.
- [11] P. Shukla, S. Nasrin, N. Darabi, W. Gomes, and A. R. Trivedi, "Mc-cim: Compute-in-memory with monte-carlo dropouts for bayesian edge intelligence," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 2, pp. 884–896, 2022.
- [12] A. C. Stutts, D. Erricolo, T. Tulabandhula, and A. R. Trivedi, "Lightweight, uncertainty-aware conformalized visual odometry," *arXiv preprint arXiv:2303.02207*, 2023.
- [13] N. Darabi, M. B. Hashem, H. Pan, A. Cetin, W. Gomes, and A. R. Trivedi, "Adc/dac-free analog acceleration of deep neural networks with frequency transformation," *arXiv preprint arXiv:2309.01771*, 2023.
- [14] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE signal processing magazine*, vol. 20, no. 5, pp. 19–38, 2003.
- [15] D. A. Reynolds *et al.*, "Gaussian mixture models," *Encyclopedia of biometrics*, vol. 741, no. 659–663, 2009.
- [16] B. Eckart, K. Kim, and J. Kautz, "Fast and accurate point cloud registration using trees of gaussian mixtures," *arXiv preprint arXiv:1807.02587*, 2018.
- [17] X. Gu, Z. Wan, and S. S. Iyer, "Charge-trap transistors for cmos-only analog memory," *IEEE Transactions on Electron Devices*, vol. 66, no. 10, pp. 4183–4187, 2019.
- [18] K. Lai, L. Bo, and D. Fox, "Unsupervised feature learning for 3d scene labeling," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3050–3057.
- [19] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-Based Localization Using LSTMs for Structured Feature Correlation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 627–637.
- [20] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2043–2050.
- [21] R. Li, S. Wang, Z. Long, and D. Gu, "UnDeepVO: Monocular Visual Odometry Through Unsupervised Deep Learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [22] A. Valada, N. Radwan, and W. Burgard, "Deep Auxiliary Learning for Visual Localization and Odometry," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [23] N. Radwan, A. Valada, and W. Burgard, "VLNet++: Deep Multitask Learning for Semantic Visual Localization and Odometry," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4407–4414, 2018.
- [24] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.
- [25] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.
- [26] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *2016 IEEE international conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4762–4769.
- [27] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," *arXiv preprint arXiv:1705.07832*, 2017.
- [28] I. Gibbs and E. Candes, "Adaptive conformal inference under distribution shift," *Advances in Neural Information Processing Systems*, vol. 34, pp. 1660–1672, 2021.
- [29] A. C. Stutts, D. Erricolo, S. Ravi, T. Tulabandhula, and A. R. Trivedi, "Mutual information-calibrated conformal feature fusion for uncertainty-aware multimodal 3d object detection at the edge," *arXiv preprint arXiv:2309.09593*, 2023.
- [30] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," *Advances in neural information processing systems*, vol. 31, 2018.