

BERRY: Bit Error Robustness for Energy-Efficient Reinforcement Learning-Based Autonomous Systems

Zishen Wan¹, Nandhini Chandramoorthy², Karthik Swaminathan², Pin-Yu Chen²,
Vijay Janapa Reddi³, and Arijit Raychowdhury¹

¹Georgia Institute of Technology ²IBM Research ³Harvard University

Abstract—Autonomous systems, such as Unmanned Aerial Vehicles (UAVs), are expected to run complex reinforcement learning (RL) models to execute fully autonomous position-navigation-time tasks within stringent onboard weight and power constraints. We observe that reducing onboard operating voltage can benefit the energy efficiency of both the computation and flight mission, however, it can also result in on-chip bit failures that are detrimental to mission safety and performance. To this end, we propose BERRY, a robust learning framework to improve bit error robustness and energy efficiency for RL-enabled autonomous systems. BERRY supports robust learning, both offline and on-board the UAV, and for the first time, demonstrates the practicality of robust low-voltage operation on UAVs that leads to high energy savings in both *compute-level operation* and *system-level quality-of-flight*. We perform extensive experiments on 72 autonomous navigation scenarios and demonstrate that BERRY generalizes well across environments, UAVs, autonomy policies, operating voltages and fault patterns, and consistently improves robustness, efficiency and mission performance, achieving up to 15.62% reduction in flight energy, 18.51% increase in the number of successful missions, and 3.43× processing energy reduction.

I. INTRODUCTION

Autonomous systems are becoming prevalent for Position-Navigation-Timing (PNT) applications. To achieve fully autonomous PNT, state-of-the-art unmanned aerial vehicles (UAVs) are expected to run complex reinforcement learning (RL) models on-board with little-to-no offloading computation support [1]–[3]. However, these safety-critical UAVs usually have Size, Weight, and Power (SWaP) constraints, hence it becomes imperative to deliver energy-efficient computation without compromising on robustness and safety [4], [5]. To satisfy these constraints, processors on-board the UAV are usually designed with techniques proposed for energy-efficient AI accelerators, such as quantization [6], specialized compute units, and optimized dataflows [7], [8].

In contrast to other embedded and mobile applications, the processing power is only a small fraction of the total UAV power, with the majority being used for flight motion. However, a small reduction in processing power would enable it to be re-targeted towards increasing the flight speed, thus resulting in significant energy savings on account of reduced flight time [9], [10]. Given the quadratic relation between energy and operating voltage, lowering the supply voltage of the onboard processor is a powerful means of energy-efficient computing within prescribed SWaP budgets.

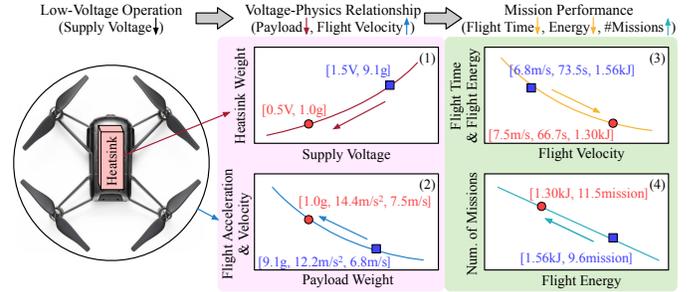


Fig. 1: Relation between supply voltage, payload weight, velocity, flight time, flight energy and the number of missions observed in DJI Tello UAV. (1) Lowering the supply voltage of the onboard processing unit helps reduce peak temperature, and correspondingly, the heatsink weight. (2) A reduced payload can significantly improve acceleration and velocity, thus (3) reducing flight time and energy, (4) making the UAV complete more missions on a single battery charge.

Fig. 1 shows the relationship between factors affecting UAV navigation. Scaling the processor supply voltage reduces its peak temperature, which would then reduce the required size and weight of the heatsink. This reduction in the payload translates to a further reduction in overall flight time and energy. A further advantage is that voltage scaling methods are complementary to and can be applied in conjunction with other energy-saving techniques described above.

However, scaling down the voltage towards near-threshold ranges can have adverse implications on the overall reliability of the UAV. Operating below rated voltage ranges can result in memory bit errors [11]–[16] and logic timing errors [17]. While [11], [13] incur overheads of error detection and mitigation through circuit-microarchitecture methods, other works present methods to generate DNN inference models that are robust to bit errors using error-aware training [18]–[20]. Although modern-day processors are equipped with RAS-enhancing features such as parity, ECC and redundancy, they are primarily targeted toward mitigating transient errors and are ineffective against low-voltage induced errors.

In this paper, we propose BERRY, a robust reinforcement learning framework for autonomous systems. BERRY applies error-aware training to optimize system *robustness*, thus boosting the processing *efficiency* and improving mission-level *performance* under *low operating voltage*. BERRY supports both offline and on-device learning paradigms to enable robust and

efficient low-voltage operation. BERRY generalizes across devices and voltages, and improves robustness and mission performance across UAVs, environments, models, and tasks.

This paper, therefore, makes the following contributions:

- We systematically analyze the impact of lowering processor voltage on autonomous systems, and determine the relation between voltage, robustness, and mission performance.
- We propose BERRY, a robust learning framework for RL-based autonomous systems, with both offline and on-device learning support. This is the *first work* to focus on *robust learning* for low-voltage operation on UAVs, achieving high *task robustness*, *processing efficiency*, and *mission performance*.
- We evaluate BERRY on 72 UAV deployment scenarios and show that BERRY generalizes across UAVs, environments, voltages, and bit error patterns. BERRY achieves up to 15.62% energy savings, 18.51% increase in successful missions with 3.43× processing energy reduction on navigation.

II. BACKGROUND AND RELATED WORK

A. Reinforcement Learning (RL)-Based Autonomous Systems

In RL-based autonomous systems, the agent learns a policy by interacting with the environment to achieve defined goals. The learning procedure is modeled by a Markov Decision Process (MDP) as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where \mathcal{S} is the state space, \mathcal{A} is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the MDP transition probabilities, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward. At each interaction i , the agent observes the tuple $\mathcal{D}_i = (s_i, a_i, s_{i+1}, r_i)$, where $s_i, s_{i+1} \in \mathcal{S}$ is the current and next state, $a_i \in \mathcal{A}$ is the action taken at step i , and $r_i = \mathcal{R}(s_i, a_i)$ is the obtained reward.

We aim to learn an optimal policy π^* given the observed \mathcal{D}_i that can maximize the reward, i.e., $\pi^*(s) = \operatorname{argmax}_a Q^*(s, a)$, with the function $Q^* : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. We use Q-learning where the Bellman backup operator is used to update the Q function:

$$Q^\pi(s_i, a_i) \leftarrow \left[\mathcal{R}(s_i, a_i) + \gamma \max_{a'} Q^\pi(s_{i+1}, a') \right] \quad (1)$$

The above policy converges to an optimal π^* (deterministic) under the Bellman backup operator. We use Deep Q-Network (DQN) approximation $f_\theta : \mathcal{S} \rightarrow \mathcal{A}$ to estimate the Q function parameterized by weights θ . This neural network learns an updated mapping from $s \rightarrow Q(s, \cdot)$ using back-propagation by minimizing the loss between the predicted and Bellman updated target Q -values (Eq. 1). Prior works have proved that DQN performs well in UAV autonomous systems [1], [2].

B. Low-Voltage Induced Bit Errors

Lowering operating voltages towards near-threshold ranges exacerbates bit cell variations. This manifests as an exponential increase in bit errors, affecting accelerator memories in which weights are stored and updated. Fig. 2 shows this relationship for an exemplary 14nm FinFET SRAM chip fabricated in [13] and a sample spatial distribution of bit errors in a segment

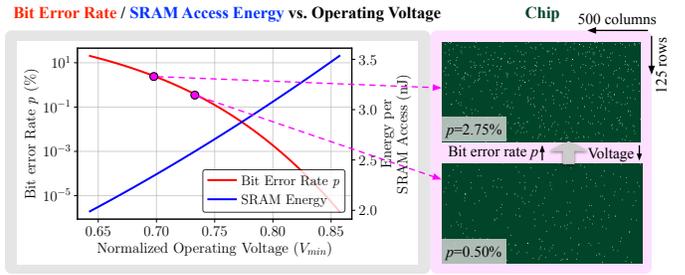


Fig. 2: Low-Voltage Operation, Energy and Bit Errors. SRAM bit error rate increases and access energy drops with decreasing supply voltage for a 14nm FinFET SRAM chip fabricated in [13]. The voltage (x-axis) is normalized to V_{min} , the lowest measured voltage at which there are no bit errors. Reproduced on the right is a random spatial error pattern in a cross-section of the memory array from [13], [19].

of the tested memory arrays. At a given voltage, these bit flips are persistent across multiple reads and write to the same location. The locations are random and independent of each other across different chips and arrays [12], [18], [19]. It needs to be noted that these are not transient errors, so computational redundancy in space and time cannot mitigate them. Similarly, standard ECC may not correct all observed errors since there could be multiple faulty bits per memory word.

C. Bit Error-Aware Training

Rather than incurring the cost of hardware-based error mitigation [11], [13], [21], some works [14], [18], [19] advocate improving robustness by generating a model resilient to bit errors. Profiled bit errors are injected during offline training on error-free hardware, resulting in a robust model during inference on low-voltage devices. BERRY proposes the generation of robust models but tackles entirely different challenges: (i) BERRY targets learning (both offline and on-device) as opposed to inference, meaning that *learning can occur on low-voltage devices* with bit errors affecting parameters. (ii) Prior works are restricted to supervised object classification while BERRY focuses on *reinforcement learning*, leading to a new error-aware training framework. (iii) BERRY tackles the complex relationship between low-voltage operation and cyber-physical autonomous systems aiming to *improve both compute-level and system-level efficiency while ensuring robustness*.

D. Resilience Characterization in UAVs

Several prior works have attempted to characterize the resilience of applications running on-board UAVs. [22]–[24] analyze the transient error impact on the resilience of learning-based and control-based UAV navigation systems. [25], [26] characterize the reliability and robustness of RL algorithms in terms of their inter-run variability, while [27] proposes techniques to train robust RL models in the presence of adversarial perturbations and interferences. In contrast, BERRY tackles the problem of bit errors due to low voltages while running RL models on UAVs, and supports both offline and on-device robust learning to mitigate fault effects.

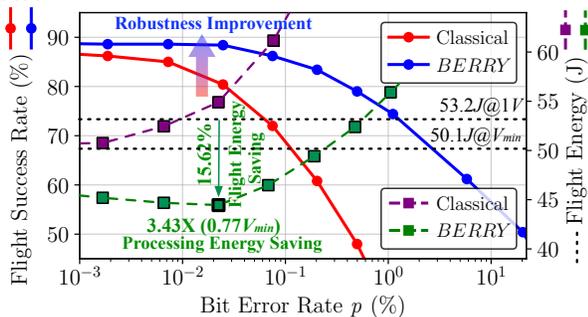


Fig. 3: Robustness to Bit Errors and Flight Energy Savings. Low-voltage-induced bit errors degrade flight success rate, while BERRY improves robustness significantly. Robustness to higher bit error rates allows more energy-efficient operation (Fig. 2). Compute correlates with physics in UAV (Fig. 1), and the robust operation brings 15.62% flight energy savings and 18.51% more missions for navigation tasks.

III. LOW VOLTAGE FAULT IMPACT CHARACTERIZATION

This section explores how low-voltage operation and its induced bit errors impact the robustness, efficiency, and mission performance of autonomous navigation systems.

We evaluate the impact of low-voltage operation on an example autonomous system where the UAV aims to navigate from the start position to the goal position in the shortest time without colliding into obstacles (Sec. V-A). Task success rate, processing energy, and quality-of-flight (flight time, flight energy, endurance) are evaluated during the mission.

Task success rate: As in Fig. 3, we observe that the low-voltage induced bit errors greatly degrade autonomous system robustness. The increasing number of bit errors gradually pollutes error-free learned policy and results in the UAV taking the wrong actions to collide with obstacles. The task success rate drops to $<85\%$ when bit error rate $p > 0.01\%$ ($\sim 0.8V_{min}$).

Flight time: The corrupted flight actions due to bit errors can directly lead to path detours, resulting in longer trajectory distance and extended flight time for a given task.

Flight energy: The increased flight time consequently increases total flight energy despite processing energy savings from low-voltage operation (Fig. 3). This is because $\sim 95\%$ flight energy is consumed by rotors that closely correlate with the flight time.

Endurance: The increased single-mission flight energy and duration further reduces the total number of missions that the UAV can successfully complete on a single charge before its battery depletes.

To achieve the robustness of an autonomous system under low voltages and improve its processing efficiency and quality-of-flight, we propose BERRY, using principles of error-aware training. Fig. 3 highlights the key results of BERRY on the same UAV navigation system: with a drop in success rate of $<1\%$, $3.43\times$ processing energy savings, 15.62% single-mission flight energy savings, and 18.51% more number of missions are achieved (compared with normal $1V$ operation), with simply lowering supply voltage to $0.77V_{min}$.

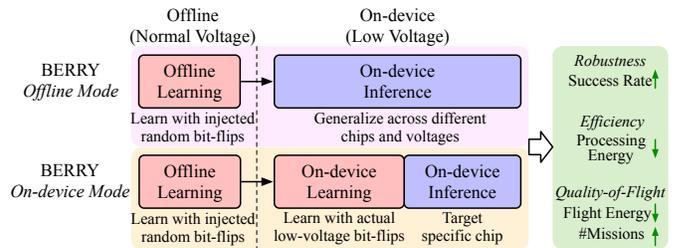


Fig. 4: BERRY with Offline and On-Device Robust Learning Modes. BERRY offline learning with random bit-flips yields generalizable robustness and quality-of-flight improvements across voltages and chips. BERRY on-device learning with specific error patterns enables even lower voltage with higher robustness and efficiency.

IV. BERRY ROBUST LEARNING FRAMEWORK

This section presents BERRY robust reinforcement learning framework, supporting both offline and on-device learning paradigms for robustness and efficiency improvements (Fig. 4). For systems without on-device learning capabilities, BERRY conducts learning offline with injected bit-flips at nominal voltage, and then deploys the robust policy on low-voltage SWaP-constrained UAV systems. For systems with on-device learning capabilities [1], BERRY enables the UAV to learn a robust policy directly on the low-voltage device incurring errors, and then perform the navigation task on the same hardware. Combining learning and navigation on the same device ensures improved success rates and higher flight energy savings tailored to the specific chip. In both cases, we perform a round of offline learning with injected bit errors, which provides generalized robustness across devices.

Algo. 1 describes the main component of BERRY, the robust learning methodology for RL-based autonomous systems. The objective is to learn a robust state action-value function of policy $Q(\theta)$. In standard Deep-Q-Learning, an evaluation network predicts the Q-value at each step, and the target network computes the Bellman temporal difference (Eq. 1), with both having the same architectures. Lines 2-3 initialize both these networks. For each step t of an episode, a minibatch of state-action-reward inputs is chosen from the experience replay buffer (line 10), and the evaluation and target networks compute the predicted Q-value and the temporal difference target y_j (line 12), respectively. The loss function is the expected squared difference between the two, and gradient $\Delta^{(t)}$ with respect to parameters θ is computed (line 13). While the Q function network parameters θ are updated every step, the target network parameters θ^- are updated every C steps to keep the training stable by copying over θ to θ^- (line 21).

Fault injection: In BERRY, both the Q-function and target networks are injected with bit errors. A random distribution of bit error locations is generated using Voltage-BER curves from existing memory characterization (Sec. II-B). Bit flips from 1-to-0 and 0-to-1 are both injected at a given voltage/bit error rate, affecting both weights and activations that are stored in the on-chip SRAM. Line 15 refers to error injection following per-layer 8-bit quantization with rounding in θ and θ^- , to obtain perturbed $\hat{\theta}$ and $\hat{\theta}^-$, respectively.

Algorithm 1 BERRY Robust Error-Aware Training Framework for Reinforcement Learning-Based Autonomous Systems

```

1: procedure BERRY( $p$ )
2:   Initialize action-value function  $Q$  with weight  $\theta$ 
3:   Initialize target action-value function  $\hat{Q}$  with weight  $\theta^- = \theta$ 
4:   for episode  $e = 1$  to  $E$  do
5:     for time step  $t = 1$  to  $T$  do
6:       Given state  $s_t$ , take action  $a_t$  based on  $Q$  ( $\epsilon$ -greedy)
7:       Obtain reward  $r_t$  and reach new state  $s_{t+1}$ 
8:       Store transition  $(s_t, a_t, r_t, s_{t+1})$  in  $D$ 
9:       // Experience replay
10:      Sample a mini-batch  $\{(s_j, a_j, r_j, s_{j+1})\}_{b=1}^B$  from  $D$ 
11:      // Clean training pass
12:      Set  $y_j = r_j + \gamma \max_{a'} Q(s_{j+1}, a'; \theta^{-(t)})$ 
13:       $\Delta^{(t)} = \nabla_{\theta} \sum_{b=1}^B (Q(s_j, a_j; \theta^{(t)}) - y_j)^2$ 
14:      // Perturbed training pass, inject bit errors at rate  $p$ 
15:       $\tilde{\theta}^{(t)} = BErr_p(\theta^{(t)})$ 
16:      Set  $\tilde{y}_j = (r_j + \gamma \max_{a'} Q(s_{j+1}, a'; \tilde{\theta}^{-(t)}))$ 
17:       $\tilde{\Delta}^{(t)} = \nabla_{\theta} \sum_{b=1}^B (Q(s_j, a_j; \tilde{\theta}^{(t)}) - \tilde{y}_j)^2$ 
18:      // Average gradients and update w.r.t  $\theta$ 
19:       $\theta^{(t+1)} = \theta^{(t)} - \alpha(\Delta^{(t)} + \tilde{\Delta}^{(t)})$ 
20:      // Periodic update of target network
21:      Every  $C$  steps reset  $\hat{Q} = Q$ , i.e., set  $\theta^- = \theta$ 
22: Output: Bit-error robust action-value function  $Q(\theta)$ 

```

Gradient update: Q and y_j are computed with perturbed parameters $\tilde{\theta}$ and $\tilde{\theta}^-$ (line 16). The new loss, termed perturbed loss, is computed along with the gradient $\tilde{\Delta}^{(t)}$ (line 17). Robust learning in BERRY is designed to work in both error-free and faulty hardware with voltage scaling without having to retrain the model. Therefore, the parameter update (line 19) uses an average of the perturbed and unperturbed gradients in stochastic gradient descent.

V. BERRY EVALUATION

This section evaluates BERRY on UAV autonomous navigation systems with 72 different scenarios. We demonstrate that BERRY consistently improves *task robustness*, *processing and mission-level efficiency* for both offline and on-device learning, and *generalize* well across various environments, UAVs, models, and error patterns measured from real chips.

A. Experimental Setup

Simulation Platform. We use open-source UAV simulation infrastructures [1], [2] to evaluate BERRY framework. The infrastructure is powered by Unreal Engine to simulate the environments, AirSim to capture UAV’s dynamics and kinematics, and RL algorithms to generate flight commands in real time.

Task and Policy Model. We adopt the autonomous navigation task (e.g., package delivery), where the UAV is initialized at a start location and navigates across the environment to reach the destination without colliding with obstacles. We use a perception-based probabilistic action space \mathcal{A} with 25 actions, and C3F2 neural network policy [22] with 1.1MB parameters. We assume the underlying systolic array-based architecture with on-chip SRAM, and integrate the SCALE-Sim [28] and Accelergy [29] simulators with our energy plugin (Fig. 2) to evaluate processing performance and energy. Along

TABLE I: Robustness Improvement. Average success rates under various bit error rates p . BERRY improves autonomous navigation task robustness under bit failures compared to classical RL policy.

Autonomy Schemes	Error-Free SuccRate (%)	Bit-Error SuccRate (%)				
		$p=0.01\%$	$p=0.05\%$	$p=0.1\%$	$p=0.5\%$	$p=1\%$
Classical	88.4	84.0	78.2	69.2	48.6	33
BERRY	88.8	88.6	86.6	84.4	79.2	74.8

with voltage, we also scale the frequency based on measured results on a deep-learning accelerator reported in [30].

UAV platforms. We use a Crazyflie nano UAV [31], with 27g takeoff weight, 15g max payload, 250mAh battery capacity, and 7min max flight time. In Sec. V-D, we configure another micro UAV DJI Tello [32] with 80g takeoff weight, 1100mAh battery and 13min max flight time for evaluation.

Evaluation Metrics. We evaluate both compute-level efficiency (processing energy) and mission-level metrics (success rate, flight time, flight energy, number of missions). Success rate is the percentage of successful trials, flight time and flight energy are the single-mission time and energy that are required for UAV to reach its goal, and the number of missions denotes the total missions that the UAV can complete on a battery charge. For each case, we evaluate 500 different fault maps and report the average quantity for all metrics.

B. Robustness and Efficiency Improvements

Robustness Improvement. Tab. I shows the navigation task success rate with different bit error rates p under different voltages (Fig. 2). Both classical DQN training policy and BERRY can reach >88% success rate in error-free missions. However, on lowering supply voltage, the classical trained policy is vulnerable to induced bit failures, while BERRY can still achieve ~80% success rate under $p = 0.5\%$ ($\sim 0.72V_{min}$). The success rate is comparable to other reported autonomous navigation task success rates for similar difficulty levels [2].

Processing Efficiency Improvement and System Implication. Lower voltage brings quadratic energy-saving benefits. As in Tab. II, compared with 1V normal operation [33], lowering voltage to $0.77V_{min}$ achieves $3.43\times$ energy efficiency. As validated on real UAVs, the processing efficiency improvement further brings benefits to the cyber-physical UAV system [34], [35]. The lower energy with lower TDP (thermal design power) requires a smaller heatsink with its reduced weight [36] (Fig. 6a), which then yields increased motion acceleration (Fig. 6b). With higher acceleration, the UAV becomes more agile when facing obstacles and can have higher safe flight velocity [34], [35] (Fig. 6c). When we lower the voltage from $1.28V_{min}$ to $0.79V_{min}$, the operating energy reduces $2.67\times$, with the required heatsink weight reducing from 3.26g to 1.22g, making the UAV achieve higher acceleration ($6.37m/s^2$ to $7.56m/s^2$) and safe flight velocity ($4.91m/s$ to $5.43m/s$), further benefiting autonomous system mission-level efficiency.

Mission Efficiency Improvement. Tab. II shows the autonomous system mission-level performance (i.e., flight distance, time, energy, and the number of missions) with BERRY under low voltages. The improved robustness benefits the mission success rate maintained at ~88% and the path distance

TABLE II: Operating and System Efficiency Improvement. Operating energy, task success rate, and system-level quality-of-flight under low voltages. BERRY improves operating energy efficiency due to quadratic effect of lowering voltage. BERRY improves system mission-level efficiency with reduced flight time, flight energy and more number of missions owing to low-voltage operation and improved robustness.

Low-Voltage Operation		Processing	Robustness	Autonomous System Mission-Level Quality-of-Flight					
Voltage (V)	Bit Error Rate p (%)	Energy Savings (%)	Success Rate (%)	Flight Distance (m)	Flight Time (s)	Flight Energy E_{flight} (J)	E_{flight} Savings	Num. of Missions $N_{mission}$	$N_{mission}$ Improvements
1	0	-	88.4	14.89	6.81	53.19	-	55.35	-
$0.86V_{min}$	1.96×10^{-6}	$2.77 \times$	88.0	14.93	6.51	47.23	-11.21%	62.05	+12.12%
$0.84V_{min}$	1.38×10^{-5}	$2.87 \times$	89.2	14.86	6.48	46.66	-12.28%	63.66	+15.03%
$0.83V_{min}$	8.23×10^{-5}	$2.97 \times$	89.0	14.91	6.46	46.41	-12.73%	63.85	+15.37%
$0.81V_{min}$	4.22×10^{-4}	$3.07 \times$	88.8	14.96	6.45	46.22	-13.11%	63.98	+15.61%
$0.80V_{min}$	1.87×10^{-3}	$3.18 \times$	88.6	14.94	6.42	45.80	-13.90%	64.42	+16.40%
$0.79V_{min}$	7.25×10^{-3}	$3.30 \times$	88.6	14.94	6.39	45.38	-14.67%	65.01	+17.46%
$0.77V_{min}$	2.47×10^{-2}	$3.43 \times$	88.4	14.91	6.35	44.88	-15.62%	65.59	+18.51%
$0.76V_{min}$	7.49×10^{-2}	$3.55 \times$	86.2	15.71	6.67	46.90	-11.82%	61.20	+10.58%
$0.74V_{min}$	2.03×10^{-1}	$3.69 \times$	83.4	16.58	7.03	49.14	-7.61%	56.52	+2.12%
$0.73V_{min}$	4.98×10^{-1}	$3.84 \times$	79.0	18.03	7.61	52.98	-0.39%	49.66	-10.27%
$0.71V_{min}$	1.11	$3.99 \times$	74.4	19.46	8.18	56.62	-6.45%	43.75	-20.95%
$0.68V_{min}$	5.80	$4.42 \times$	63.2	21.84	9.09	61.96	+16.49%	33.96	-38.64%
$0.64V_{min}$	20.36	$4.93 \times$	50.4	24.52	10.11	67.83	+27.53%	24.74	-55.30%

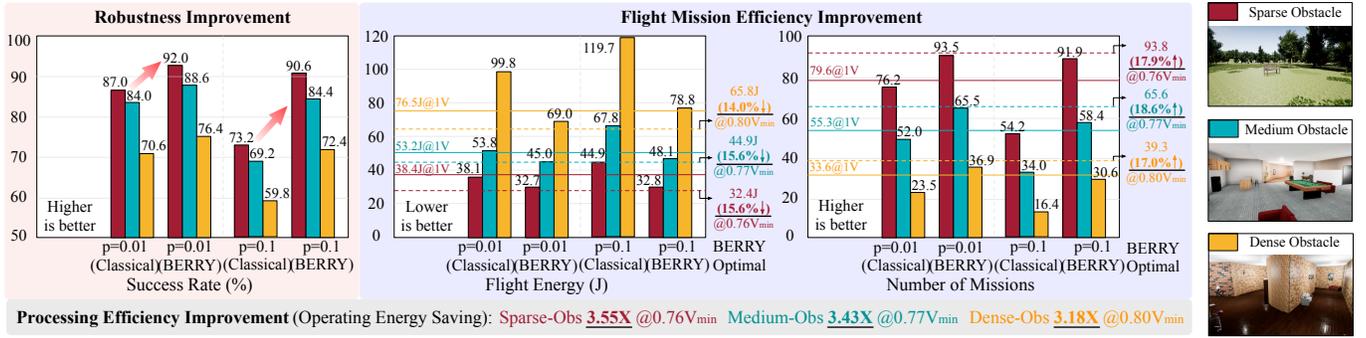


Fig. 5: Effectiveness across Different Environments. BERRY is evaluated in three environments with different obstacle densities. BERRY consistently improves task robustness and mission efficiency (reduced single-mission flight energy and increased number of missions). BERRY is adaptive to various environments, and enables lower-voltage operations in sparse ($0.76V_{min}$) than complex environments ($0.80V_{min}$).

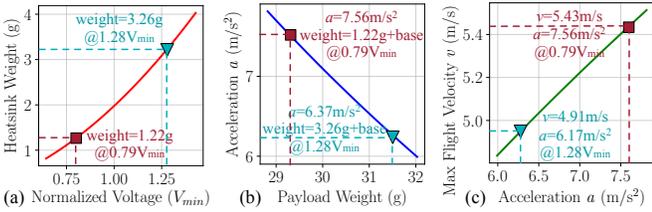


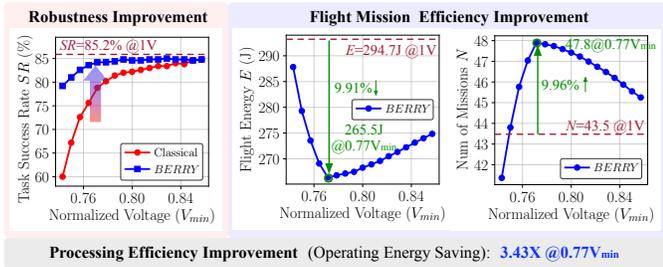
Fig. 6: Low Operating Voltage Brings System Benefits. (a) Lower operating voltage \rightarrow lower energy and thermal design power (TDP) \rightarrow require smaller heatsink size with reduced weight. (b) Lower payload weight \rightarrow higher motion acceleration. (c) Higher acceleration \rightarrow more agile when facing obstacle \rightarrow higher safe flight velocity. With higher velocity, the UAV can finish a mission faster (lower flight time), thus consuming less flight energy for a single mission. This enables more missions under a single battery charge.

at $\sim 15m$ under $0.77V_{min}$. The flight distance then increases due to path detours induced by sub-optimal actions. The flight time drops to $6.35s$ when lowering voltage to $0.77V_{min}$ due to higher flight velocity (Fig. 6) for the same flight distance. Similarly, the flight energy reduces from $53.19J$ to $44.88J$ owing to the shortened flight time and the reduced power. The number of completed missions (N) under a battery charge (E) closely correlates to the success rate (SR) and single-mission

energy (E') as $N = SR \times E/E'$. The number of missions increases from 55 to 65 owing to the reduced flight energy and increased success rate. Overall, BERRY enables lower voltage operation for robust and efficient autonomous systems. At $0.77V_{min}$ with 0.025% p , BERRY achieves 15.62% (10.95%) less flight energy, 18.51% (11.99%) more missions with $3.43 \times$ ($2.04 \times$) operating energy savings compared to $1V$ (V_{min}).

C. Environment Evaluation

Effectiveness across Different Environments. Fig. 5 evaluates BERRY on three environments with different obstacle densities, namely sparse (outdoor), medium (indoor) and dense (indoor) obstacle environments. Compared with classical DQN policy, it is well observed that BERRY improves the success rate and mission efficiency, with $3.55 \times$, $3.43 \times$, $3.18 \times$ operating energy savings, 15.6%, 15.6%, 14.0% single-mission flight energy reduction, and 17.9%, 18.6%, 17.0% more number of missions for sparse, medium, dense obstacle environments, respectively (numbers underlined in Fig. 5). Compared within three environments, BERRY enables lower operating voltage in sparse obstacle ($0.76V_{min}$) than dense obstacle ($0.80V_{min}$), this is because a more challenging environment brings more complex trajectories for UAVs to follow, which is more critical



UAV Type	Network Policy	Rotor Power	Compute Power	BERRY Flight Energy ↓	BERRY #Missions ↑
Crazyflie	C3F2	93.5%	6.5%	15.62%	18.51%
DJI Tello	C3F2	97.2%	2.8%	9.91%	9.96%
DJI Tello	C5F4	94.9%	4.1%	13.12%	14.38%

Fig. 7: Effectiveness across Different UAVs and Models. BERRY is evaluated on two UAVs and two models (top: DJI Tello with C3F2), and consistently improves robustness and efficiency. Higher processing power ratio makes BERRY bring more mission improvements.

TABLE III: Effectiveness across Different Profiled Bit Errors. BERRY is evaluated on different profiled chips including column-aligned error distributions. BERRY generalizes well across chips and voltages, with robustness and efficiency improvements.

Chips and Error Rates p (%)	SuccRate SR (%)	Flight Energy E (J)
Chip 1 (random pattern)	$p=0.16$ $p=0.74$	$p=0.16$ $p=0.74$
BERRY $p=0.5$	$SR=84.0$ $SR=77.2$	$E=48.46$ $E=54.63$
Chip 2 (column-aligned)	$p=0.067$ $p=0.32$	$p=0.067$ $p=0.32$
BERRY $p=0.5$	$SR=86.0$ $SR=81.8$	$E=46.98$ $E=51.27$
Baseline $p=0 @ 1V$	$SR=88.4$	$E=53.19$

to bit errors. BERRY is adaptive across environments and consistently achieves improved robustness and system efficiency.

D. UAV Platform and Policy Architecture Evaluation

Effectiveness across UAV Platforms. In Fig. 7, we evaluate BERRY on another UAV platform, DJI Tello (Sec. V-A), with the same C3F2 autonomy policy. DJI Tello has a larger frame size and takeoff weight than Crazyflie, thus the rotor power accounts for a higher ratio of total power (97.2%). Even with a smaller compute power ratio (2.8%), BERRY still consistently improves success rate under various low voltage levels, and achieves 9.91% lower flight energy and 9.96% more missions at $0.77V_{min}$ with $3.43\times$ processing efficiency.

Effectiveness across Model Architecture. In Fig. 7, we also evaluate BERRY on another autonomy policy architecture C5F4 (5 Conv and 4 FC layers) on DJI Tello UAV. C5F4 architecture has $1.98\times$ parameters than C3F2 and accounts for 4.1% of total power. BERRY saves up to 13.12% flight energy usage and increases the number of missions by 14.38%. Higher compute power attributes enable BERRY to bring more system-level benefits, and BERRY consistently improves task robustness and efficiency across various UAVs and models.

E. Bit Error Pattern Evaluation

Effectiveness across Profiled Bit Errors. In Tab. III, we evaluate BERRY on two different profiled bit error patterns from test chips in [13], [19] (Fig. 2), one showing a random spatial error pattern and another showing a column-aligned pattern with a bias towards 0-to-1 flips. BERRY generalizes

TABLE IV: On-Device Error-Aware Robust Learning. Learning the bit errors directly on low-voltage device enables lower operating voltage and improved robustness, resulting in more flight energy savings, while with the cost of on-the-fly learning energy consumption.

Low-Voltage Operation		Operating Efficiency	Robustness	Quality-of-Flight			
Num. of Learning Steps	Operating Voltage	Learning Energy (J)	Energy Savings	Success Rate (%)	Flight Energy (J)	Num. of Missions*	
On-Device	4000	$0.77V_{min}$	1849	$3.43\times$	84.6	264.2	48.19
	6000	$0.70V_{min}$	1807	$4.16\times$	82.4	266.5	46.52
BERRY	4000	$0.77V_{min}$	2775	$3.43\times$	85.0	260.9	49.03
	6000	$0.70V_{min}$	2711	$4.16\times$	84.8	255.1	50.01
Offline BERRY	$0.77V_{min}$	-	$3.43\times$	84.4	265.5	47.84	
	$0.70V_{min}$	-	$4.16\times$	63.8	375.6	25.56	
Baseline	1V	-	$1\times$	85.2	294.7	43.50	

* Does not include on-device learning flight energy, evaluated for missions after learning.

well to both lower and higher bit error rates than trained on, showing robustness and mission efficiency improvements.

F. On-Device Error-Aware Robust Learning

Effectiveness of On-Device Robust Learning at Lower Voltage and Better Quality-of-Flight. On-device fine-tuning is needed in some scenarios for UAV adapting to environments [1]. BERRY framework supports on-device robust learning where the UAV can learn the bit errors directly at low-voltage chips (Sec. IV). While on-device learning consumes on-the-fly energy, compared to offline BERRY, the UAV can enable lower operating voltage and improved robustness due to the same fault pattern in learning and inference. The achieved lower voltage can save more flight energy usage for further tasks. As in Tab. IV, with 6k on-device training steps, the Tello UAV achieves robust fly under $0.70V_{min}$, resulting in 13.41% less flight energy with $4.16\times$ less operating energy than 1V operation, and 3.89% less flight energy than offline BERRY. Since not all UAVs support on-device training, with the inherent tradeoff between learning-consumed energy and model efficiency, BERRY framework provides the flexibility for offline or on-device robust learning based on scenarios.

VI. CONCLUSION

BERRY is a promising robust learning framework unlocking practical low-voltage operation advantages on RL-enabled autonomous systems. BERRY relies on the systematic discovery of relationship between voltage and mission performance, and supports both offline and on-device error-aware learning. We have demonstrated that BERRY consistently improves task robustness, operating efficiency, and mission performance and achieves up to 15.62% energy savings, 18.51% increase in successful missions with $3.43\times$ processing energy reduction across environments, UAVs, and autonomy policies. We anticipate BERRY framework being useful in exploring robust and efficient low-voltage operations in other autonomous systems.

ACKNOWLEDGEMENTS

We would like to thank Srivatsan Krishnan from Harvard for helpful discussions. This work was supported by CoCoSys, one of the seven centers in JUMP2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA; IARPA MicroE4AI program; and CRNCH PhD Fellowship.

REFERENCES

- [1] A. Anwar and A. Raychowdhury, "Autonomous navigation via deep reinforcement learning for resource constraint edge nodes using transfer learning," *IEEE Access*, vol. 8, pp. 26549–26560, 2020.
- [2] S. Krishnan, B. Boroujerdian, W. Fu, A. Faust, and V. J. Reddi, "Air learning: a deep reinforcement learning gym for autonomous aerial robot visual navigation," *Machine Learning*, vol. 110, pp. 2501–2540, 2021.
- [3] B. P. Duisterhof, S. Krishnan, J. J. Cruz, C. R. Banbury, W. Fu, A. Faust, G. C. de Croon, and V. J. Reddi, "Tiny robot learning (tinyrl) for source seeking on a nano quadcopter," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7242–7248, IEEE, 2021.
- [4] Z. Wan, K. Swaminathan, P.-Y. Chen, N. Chandramoorthy, and A. Raychowdhury, "Analyzing and improving resilience and robustness of autonomous systems," in *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–9, 2022.
- [5] Z. Wan, Y. Gan, B. Yu, S. Liu, A. Raychowdhury, and Y. Zhu, "Vpp: The vulnerability-proportional protection paradigm towards reliable autonomous machines," in *Proceedings of the 5th International Workshop on Domain Specific System Architecture (DOSSA)*, pp. 1–6, 2023.
- [6] S. Krishnan, M. Lam, S. Chitlangia, Z. Wan, G. Barth-Maron, A. Faust, and V. J. Reddi, "Quarl: Quantization for fast and environmentally sustainable reinforcement learning," in *Transactions on Machine Learning Research (TMLR)*, 2022.
- [7] A. Suleiman, Z. Zhang, L. Carlone, S. Karaman, and V. Sze, "Navion: A 2-mw fully integrated real-time visual-inertial odometry accelerator for autonomous navigation of nano drones," *IEEE Journal of Solid-State Circuits (JSSC)*, vol. 54, no. 4, pp. 1106–1119, 2019.
- [8] Z. Wan, A. S. Lele, and A. Raychowdhury, "Circuit and system technologies for energy-efficient edge robotics," in *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 275–280, IEEE, 2022.
- [9] B. Boroujerdian, H. Genc, S. Krishnan, W. Cui, A. Faust, and V. Reddi, "Mavbench: Micro aerial vehicle benchmarking," in *2018 51st annual IEEE/ACM international symposium on microarchitecture (MICRO)*, pp. 894–907, IEEE, 2018.
- [10] S. Krishnan, Z. Wan, K. Bhardwaj, P. Whatmough, A. Faust, S. Neuman, G.-Y. Wei, D. Brooks, and V. J. Reddi, "Automatic domain-specific soc design for autonomous unmanned aerial vehicles," in *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 300–317, IEEE, 2022.
- [11] B. Reagen, P. Whatmough, R. Adolf, S. Rama, H. Lee, S. K. Lee, J. M. Hernández-Lobato, G.-Y. Wei, and D. Brooks, "Minerva: Enabling low-power, highly-accurate deep neural network accelerators," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 267–278, 2016.
- [12] S. Ganapathy, J. Kalamatianos, K. Kasprak, and S. Raasch, "On characterizing near-threshold sram failures in finfet technology," in *Proceedings of the 54th Annual Design Automation Conference (DAC)*, pp. 1–6, 2017.
- [13] N. Chandramoorthy, K. Swaminathan, M. Cochet, A. Paidimarri, S. Eldridge, R. V. Joshi, M. M. Ziegler, A. Buyuktosunoglu, and P. Bose, "Resilient low voltage accelerators for high energy efficiency," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 147–158, IEEE, 2019.
- [14] S. Koppula, L. Orosa, A. G. Yağlıkcı, R. Azizi, T. Shahroodi, K. Kanellopoulos, and O. Mutlu, "Eden: Enabling energy-efficient, high-performance deep neural network inference using approximate dram," in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 166–181, 2019.
- [15] B. Salami, E. B. Onural, I. E. Yuksel, F. Koc, O. Ergin, A. C. Kestelman, O. Unsal, H. Sarbazi-Azad, and O. Mutlu, "An experimental study of reduced-voltage operation in modern fpgas for neural network acceleration," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 138–149, IEEE, 2020.
- [16] S. S. N. Larimi, B. Salami, O. S. Unsal, A. C. Kestelman, H. Sarbazi-Azad, and O. Mutlu, "Understanding power consumption and reliability of high-bandwidth memory with voltage undervolting," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 517–522, IEEE, 2021.
- [17] J. Zhang, K. Rangineni, Z. Ghodsi, and S. Garg, "Thundervolt: enabling aggressive voltage undervolting and timing error resilience for energy efficient deep learning accelerators," in *Proceedings of the 55th Annual Design Automation Conference (DAC)*, pp. 1–6, 2018.
- [18] S. Kim, P. Howe, T. Moreau, A. Alaghi, L. Ceze, and V. Sathé, "Matic: Learning around errors for efficient low-voltage neural network accelerators," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, IEEE, 2018.
- [19] D. Stutz, N. Chandramoorthy, M. Hein, and B. Schiele, "Bit error robustness for energy-efficient dnn accelerators," *Proceedings of Machine Learning and Systems (MLSys)*, vol. 3, pp. 569–598, 2021.
- [20] D. Stutz, N. Chandramoorthy, M. Hein, and B. Schiele, "Random and adversarial bit error robustness: Energy-efficient and secure dnn accelerators," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 45, no. 3, pp. 3632–3647, 2022.
- [21] P. N. Whatmough, S. K. Lee, H. Lee, S. Rama, D. Brooks, and G.-Y. Wei, "14.3 a 28nm soc with a 1.2 ghz 568nj/prediction sparse deep-neural-network engine with 0.1 timing error rate tolerance for iot applications," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 242–243, IEEE, 2017.
- [22] Z. Wan, A. Anwar, Y.-S. Hsiao, T. Jia, V. J. Reddi, and A. Raychowdhury, "Analyzing and improving fault tolerance of learning-based navigation systems," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pp. 841–846, IEEE, 2021.
- [23] Z. Wan, A. Anwar, A. Mahmoud, T. Jia, Y.-S. Hsiao, V. J. Reddi, and A. Raychowdhury, "Frl-fi: Transient fault analysis for federated reinforcement learning-based navigation systems," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 430–435, IEEE, 2022.
- [24] Y.-S. Hsiao, Z. Wan, T. Jia, R. Ghosal, A. Mahmoud, A. Raychowdhury, D. Brooks, G.-Y. Wei, and V. J. Reddi, "Mavfi: An end-to-end fault analysis framework with anomaly detection and recovery for micro aerial vehicles," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, IEEE, 2023.
- [25] S. C. Chan, S. Fishman, J. Canny, A. Korattikara, and S. Guadarrama, "Measuring the reliability of reinforcement learning algorithms," *International Conference on Learning Representations (ICLR)*, 2020.
- [26] H. Zhang, H. Chen, D. Boning, and C.-J. Hsieh, "Robust reinforcement learning on state observations with learned optimal adversary," in *International Conference on Learning Representations (ICLR)*, 2021.
- [27] C.-H. H. Yang, L.-T. D. Hung, Y. Ouyang, and P.-Y. Chen, "Training a resilient q-network against observational interference," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 8814–8822, 2022.
- [28] A. Samajdar, J. M. Joseph, Y. Zhu, P. Whatmough, M. Mattina, and T. Krishna, "A systematic methodology for characterizing scalability of dnn accelerators using scale-sim," in *2020 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 58–68, IEEE, 2020.
- [29] Y. N. Wu, J. S. Emer, and V. Sze, "Accelergy: An architecture-level energy estimation methodology for accelerator designs," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–8, IEEE, 2019.
- [30] T. Jia, P. Mantovani, M. C. Dos Santos, D. Giri, J. Zuckerman, E. J. Loscalzo, M. Cochet, K. Swaminathan, G. Tombesi, J. J. Zhang, et al., "A 12nm agile-designed soc for swarm-based perception with heterogeneous ip blocks, a reconfigurable memory hierarchy, and an 800mhz multi-plane noc," in *ESSCIRC 2022-IEEE 48th European Solid State Circuits Conference (ESSCIRC)*, pp. 269–272, IEEE, 2022.
- [31] "Bitcraze crazyflie 2.1." <https://www.bitcraze.io/products/crazyflie-2-1/>.
- [32] "Dji ryze tech tello quadcopter." <https://www.ryzerobotics.com/tello>.
- [33] D. Palossi, A. Loquercio, F. Conti, E. Flamand, D. Scaramuzza, and L. Benini, "A 64-mw dnn-based visual navigation engine for autonomous nano-drones," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8357–8371, 2019.
- [34] S. Krishnan, Z. Wan, K. Bhardwaj, P. Whatmough, A. Faust, G.-Y. Wei, D. Brooks, and V. J. Reddi, "The sky is not the limit: A visual performance model for cyber-physical co-design in autonomous machines," *IEEE Computer Architecture Letters*, vol. 19, no. 1, pp. 38–42, 2020.
- [35] S. Krishnan, Z. Wan, K. Bhardwaj, N. Jadhav, A. Faust, and V. J. Reddi, "Roofline model for uavs: A bottleneck analysis tool for onboard compute characterization of autonomous unmanned aerial vehicles," in *2022 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 162–174, IEEE, 2022.
- [36] "Heat sink size calculator." <https://celsiainc.com/resources/calculators/heat-sink-size-calculator/>.