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

Zishen Wan[1], Nandhini Chandramoorthy[2], Karthik Swaminathan[2], Pin-Yu Chen[2], Vijay Janapa Reddi[3], Arijit Raychowdhury[1]

[1]Georgia Institute of Technology [2]IBM Research [3]Harvard University

Outline

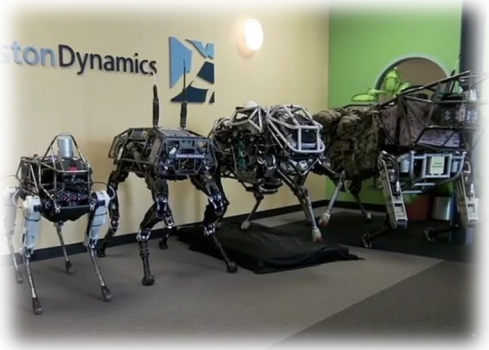
- Motivation – Why we need efficient-yet-resilient autonomous system?
- What is Autonomous Machine System
  - Physical model-based and end-to-end learning (RL)-based paradigm
- BERRY Framework
  - Drone system characterization
  - Bit-error-robustness improvement
- Evaluations
  - Various environments, models, drone platforms, chip error patterns

Motivation

Reliability

Goal: Improve operational resiliency
(Temporal/Spatial Redundancy)

Autonomous Machines



Performance

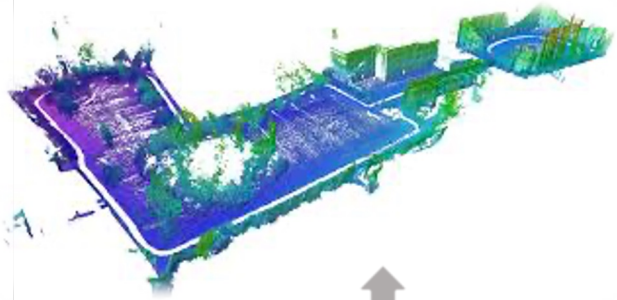
Goal: Improve mission quality
(Autonomy Algorithms)

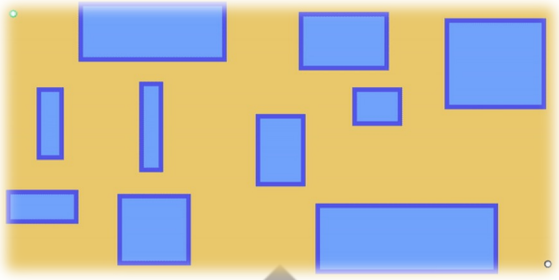
Performance-Efficiency-Reliability
Co-Optimization

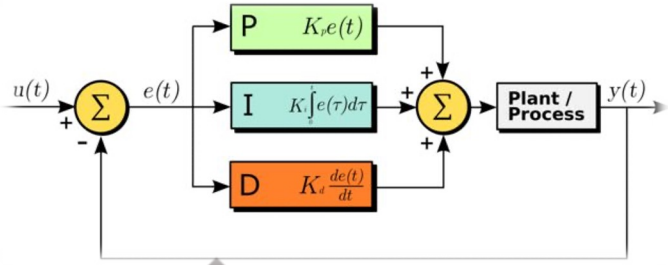
Efficiency

Goal: Improve compute efficiency
(Software Optimization,
Hardware Architecture)

What is Autonomous Machine System?







Sensing

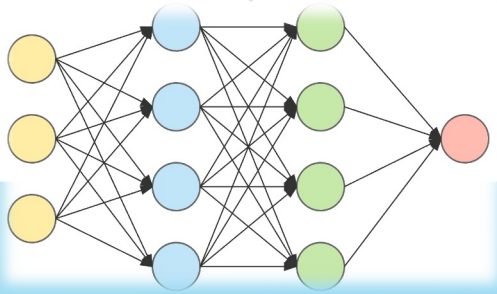
Perception

Planning

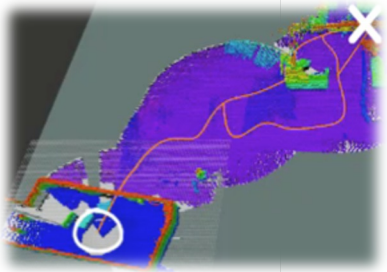
Control

Actuation

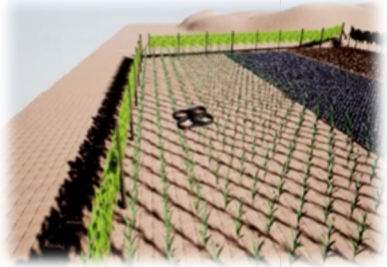
End-to-End Learning
(Reinforcement Learning)



Challenge 1: Resource Constraints

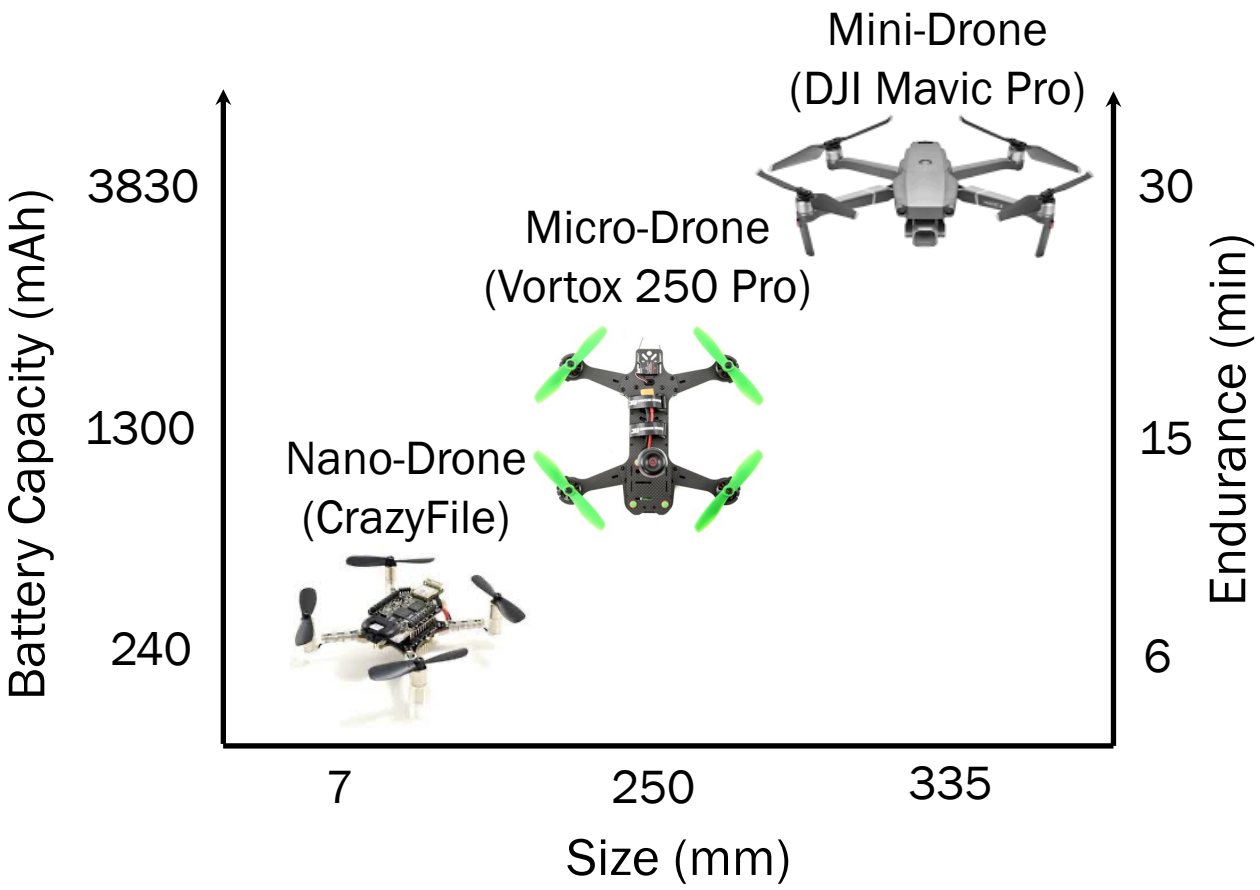


Package Delivery



Scanning

Search and Rescue



Drones are size, weight, and power (SWaP) constrained

Energy-Efficient Autonomous System

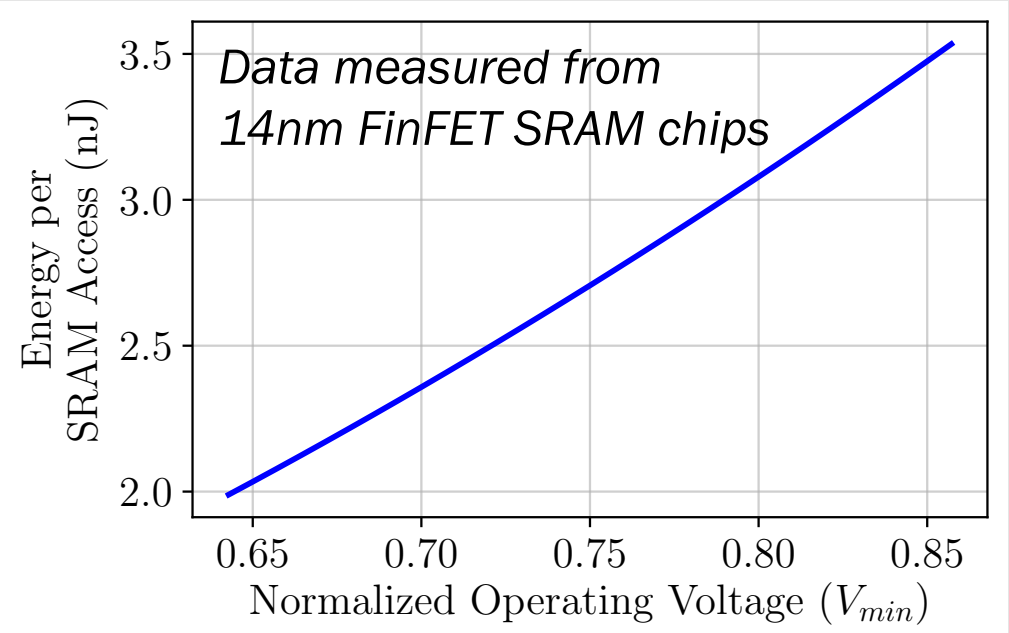
Software Optimization
(e.g., quantization, sparsity)

Hardware Optimization
(e.g., optimized dataflow, specialized compute unit)

Lower processor operating voltage

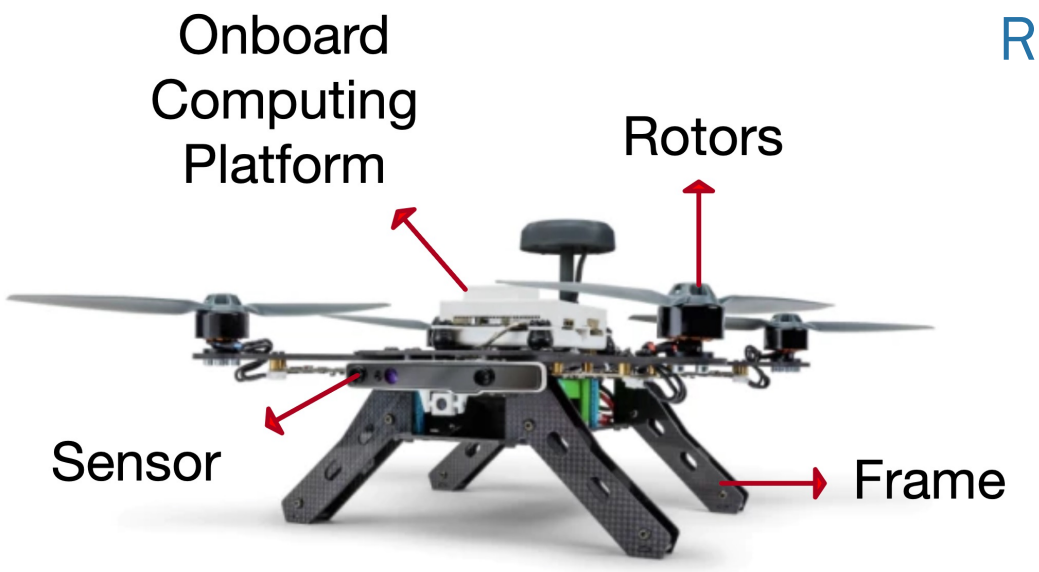
Energy ∝ Voltage ^{2}

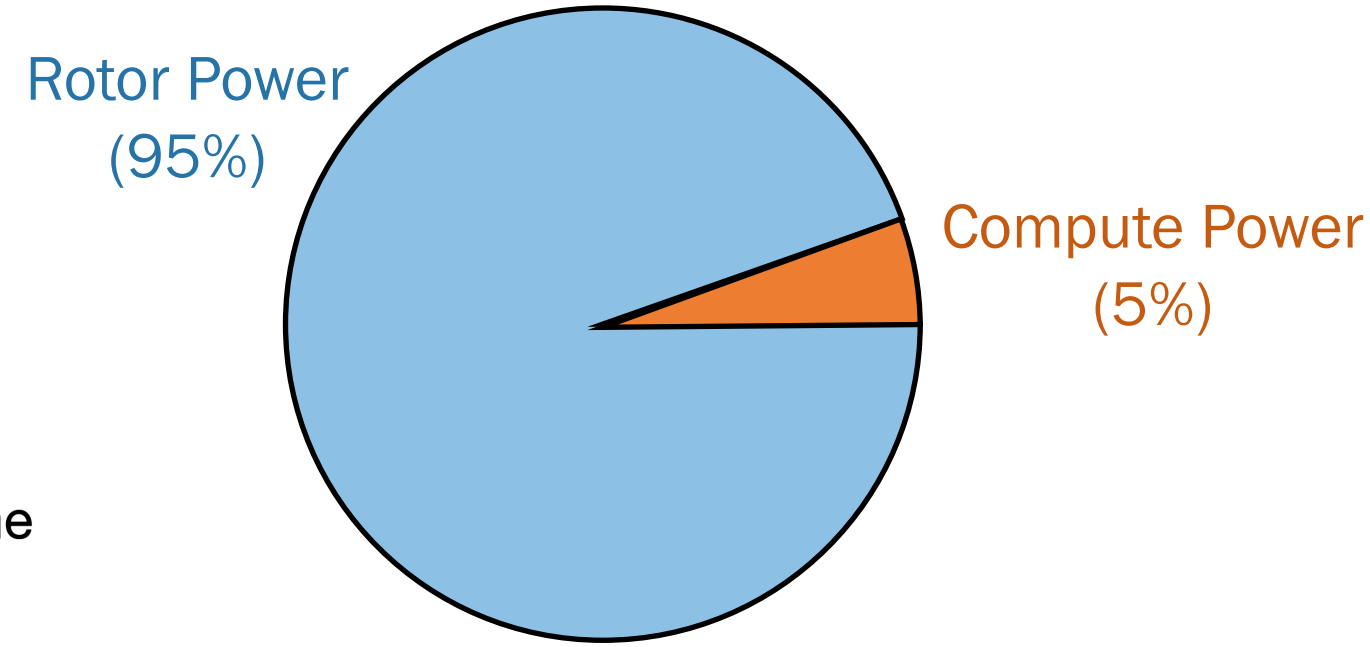
SRAM Access Energy vs. Operating Voltage



Lower operating voltage quadratically reduces energy

Challenge 2: Compute-Physics Correlation





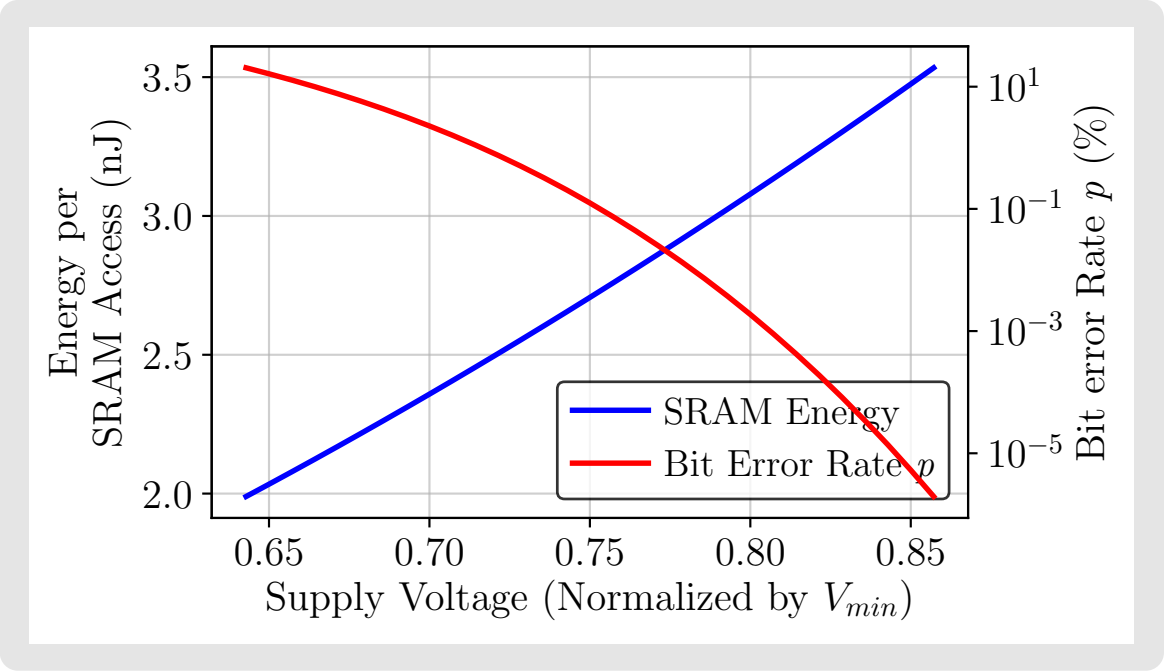
Power breakdown measured from 3DR Solo drone

Compute power is only a small fraction of total drone power
-> ***Will optimize compute bring system energy-savings?***

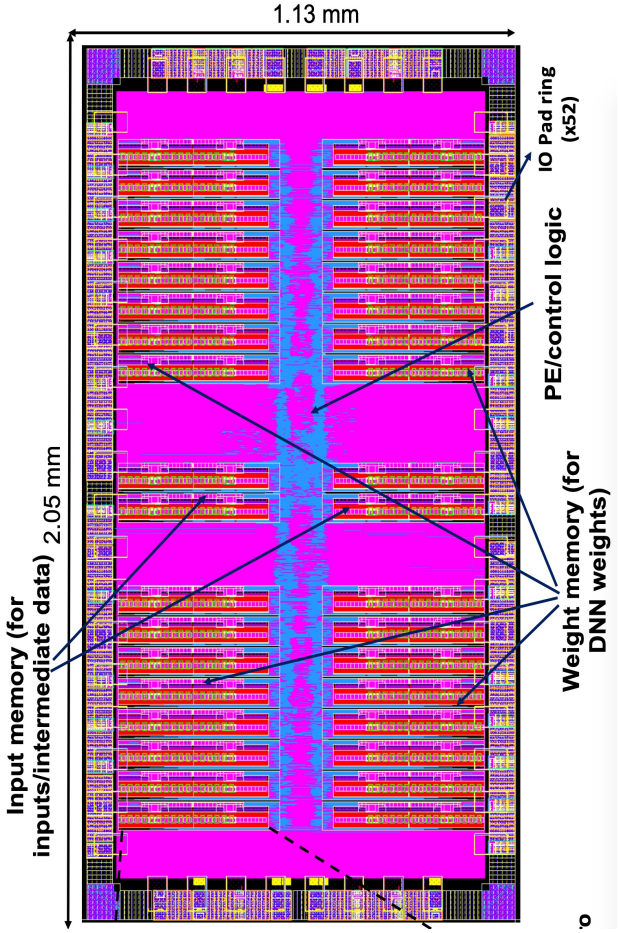
Challenge 3: Low Voltage Induces Faults

Technology	14nm
Chip Dimension	2.05 mm x 1.13 mm
Memory Capacity	128 KB weight, 16 KB input
Frequency/ Voltage	330 MHz for V_{dd}=0.8V

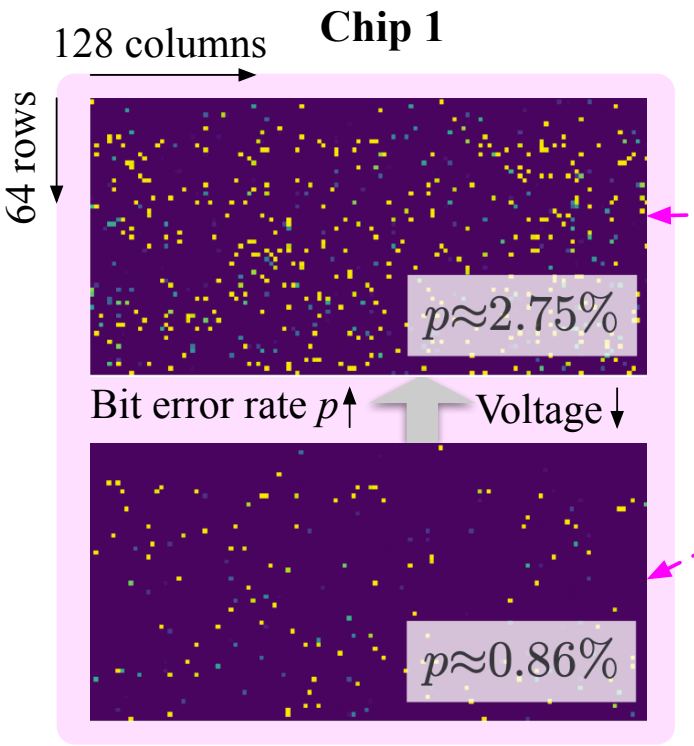
SRAM Access Energy / Bit Error Rate vs. Operating Voltage

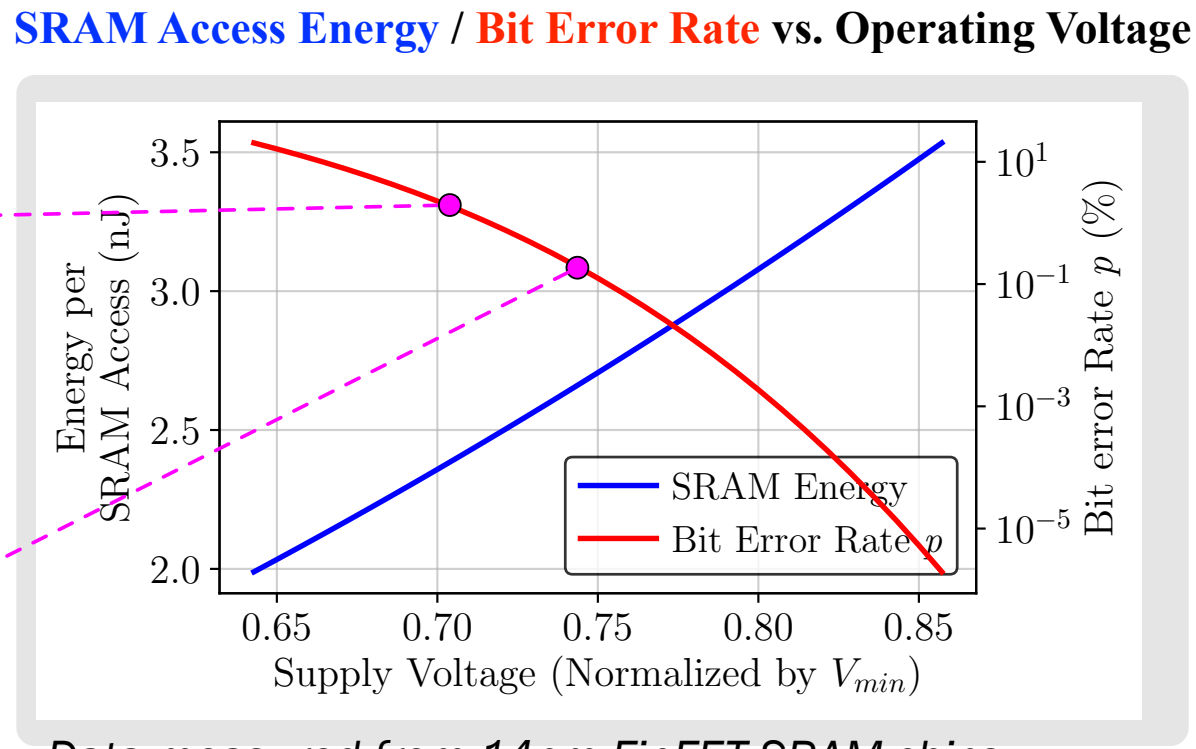


Data measured from 14nm FinFET SRAM chips



Challenge 3: Low Voltage Induces Faults





Data measured from 14nm FinFET SRAM chips

Fault Characterization

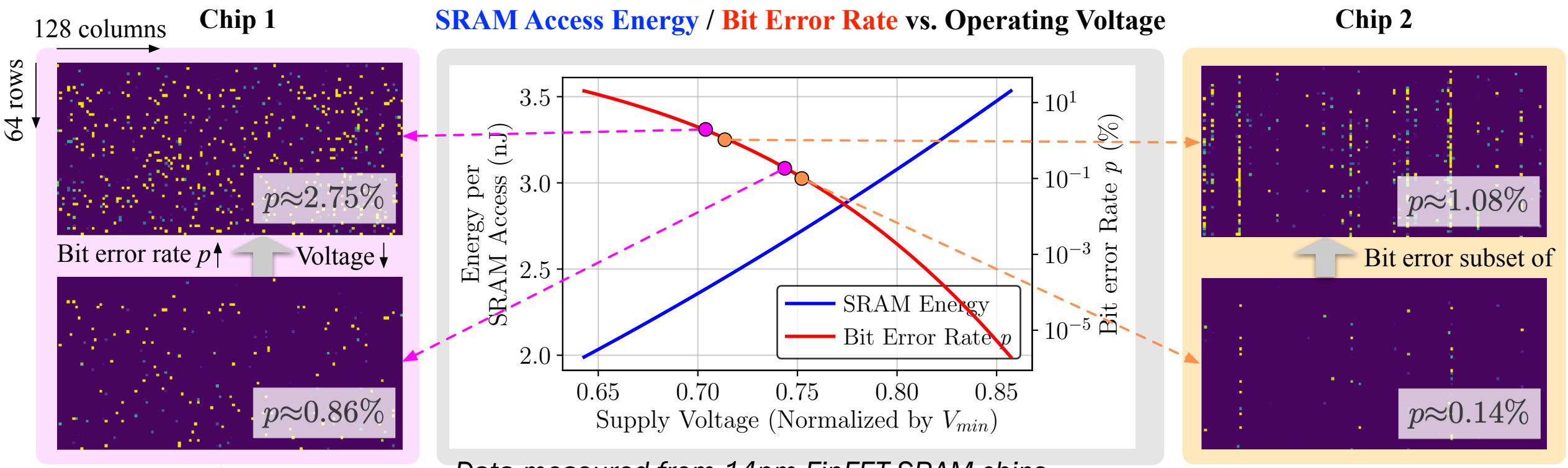
Persistent: For a fixed memory array, bit errors are persistent across voltage.

Inclusive: errors at low *BER(p)* are subset of high *BER(p).*

Independent: Across chips and arrays, error locations are random and independent.

Operating below rated voltage range results in memory bit errors, negatively impacting safety

Challenge 3: Low Voltage Induces Faults



Data measured from 14nm FinFET SRAM chips

Operating below rated voltage range results in memory bit errors, negatively impacting safety

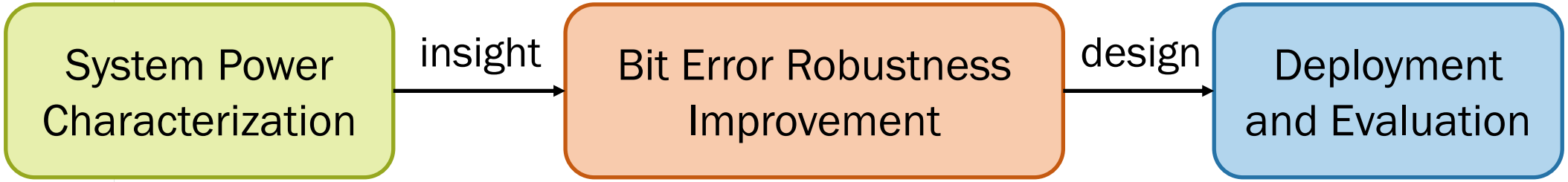
BERRY

~~How to~~ enable aggressive energy-savings yet computationally-resilience for autonomous system under **low-voltage operation**?

(performance-efficiency-resilience co-optimization)

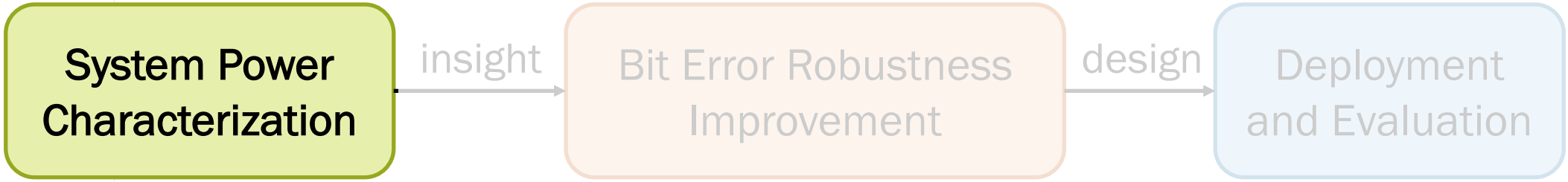
BERRY Framework

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

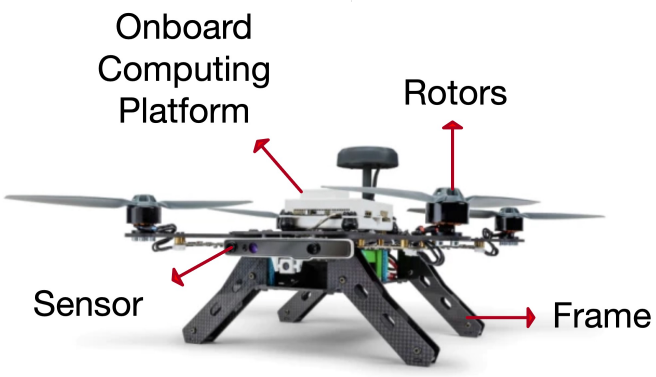


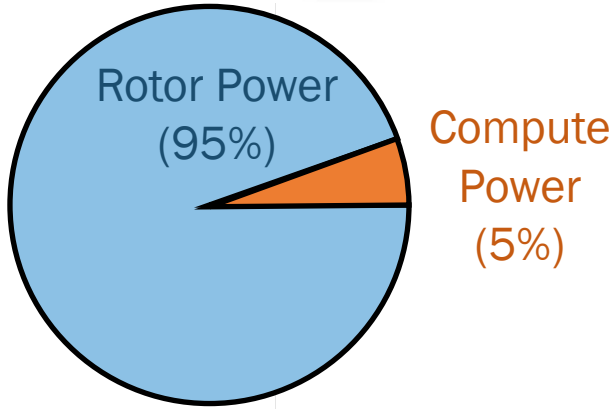
BERRY Framework

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

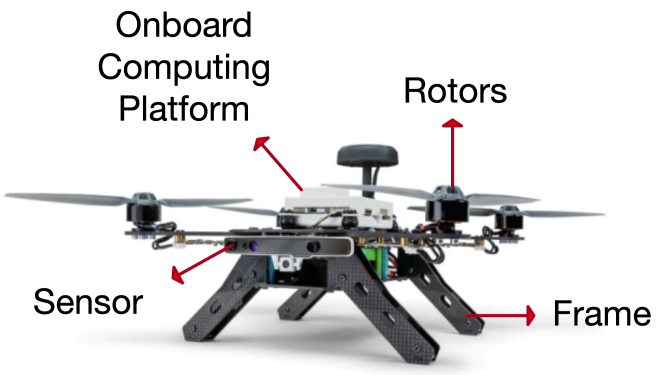


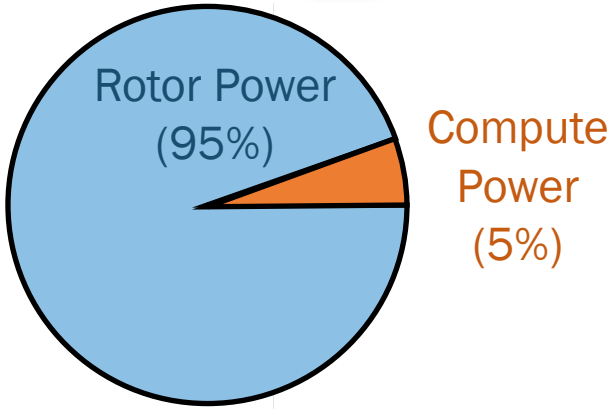
Small Compute Power, Large System Impact!



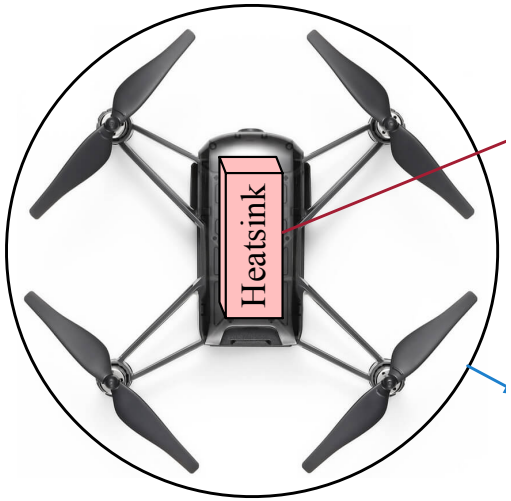


Small Compute Power, Large System Impact!



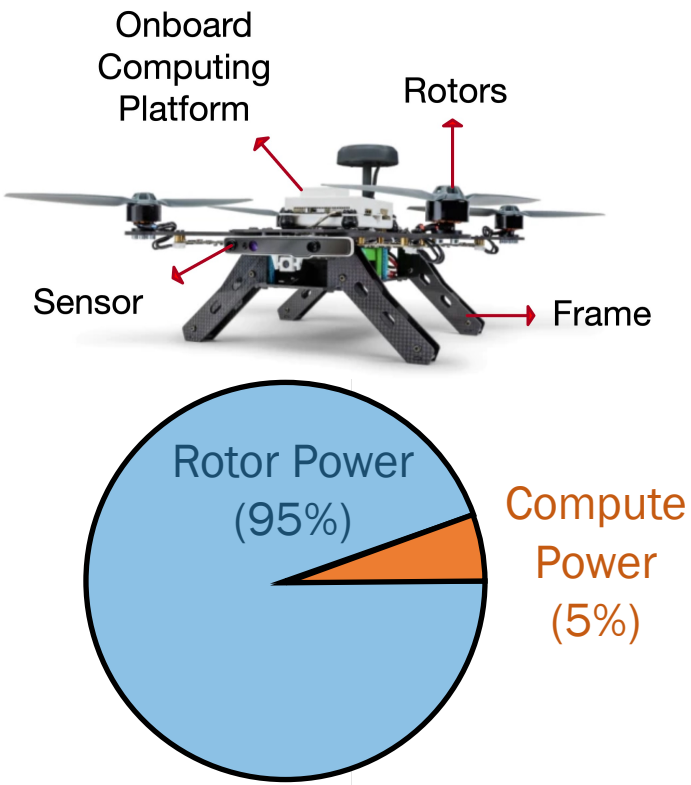


Low-Voltage Operation
(Supply Voltage↓)

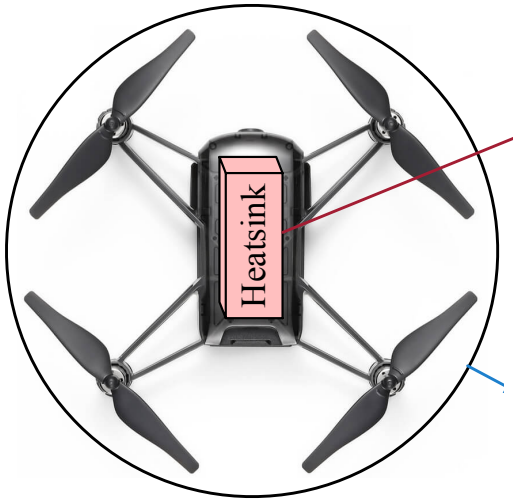


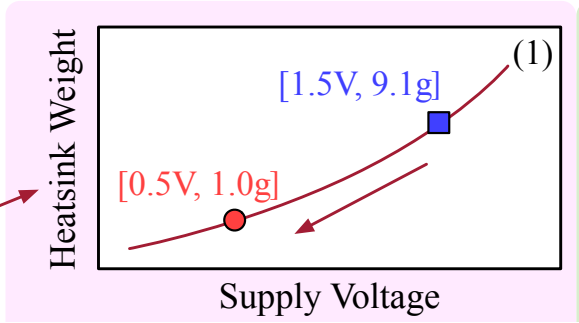
Low-voltage operation

Small Compute Power, Large System Impact!



Low-Voltage Operation (Supply Voltage↓) → Voltage-Physics Relationship (Payload↓, Flight Velocity↑)

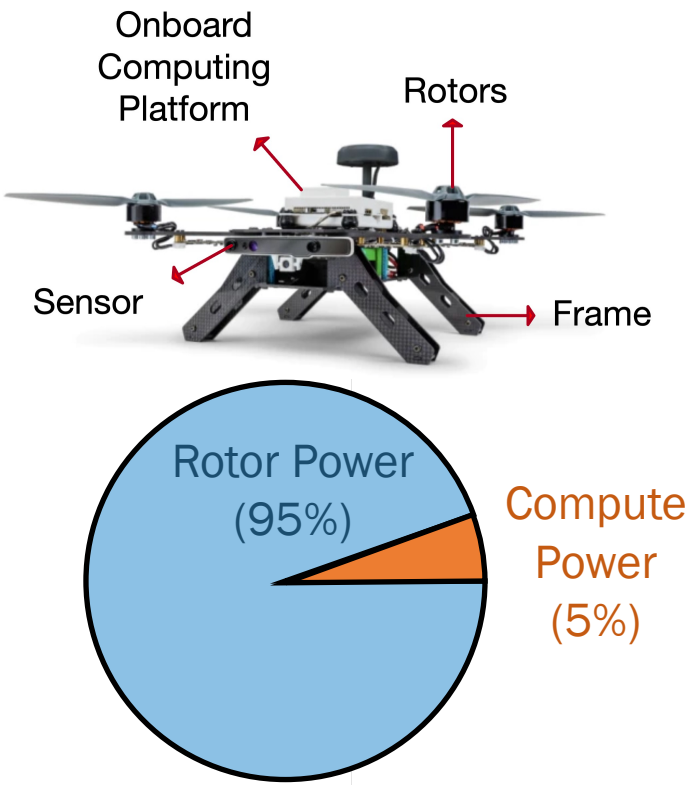




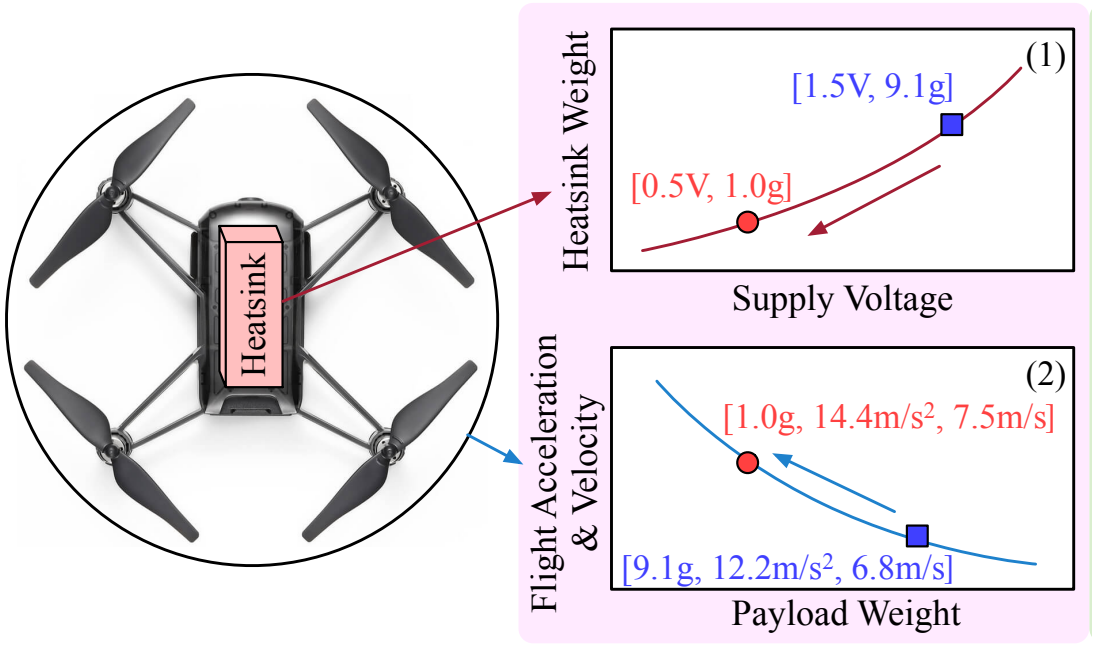
Low-voltage operation → **Peak temperature↓, heatsink size and weight↓**
(HotSpot analysis[1] + heatsink modeling[2])

[1] Hotspot 6.0: Validation, acceleration and extension
[2] Celsia Heatsink Size Simulator

Small Compute Power, Large System Impact!



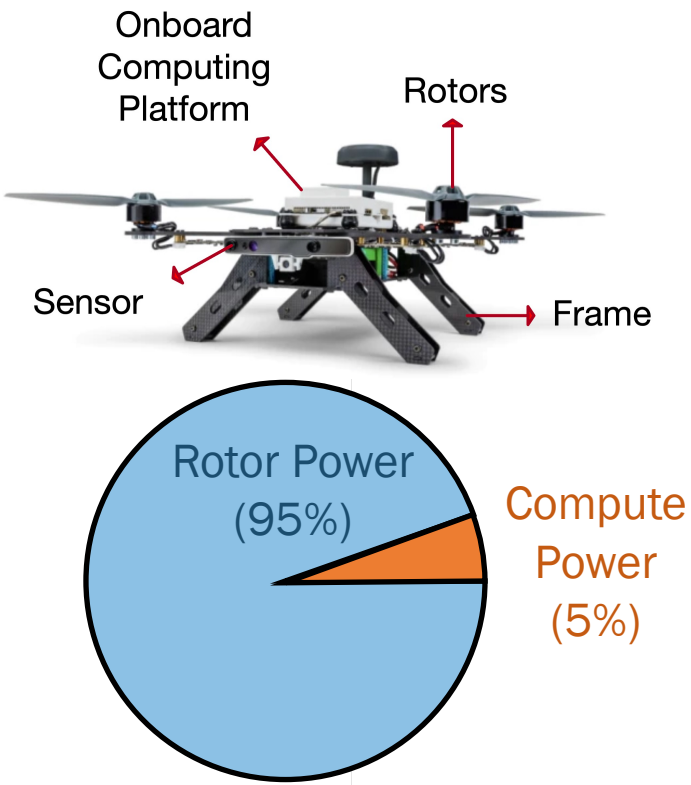
Low-Voltage Operation (Supply Voltage↓) → Voltage-Physics Relationship (Payload↓, Flight Velocity↑)



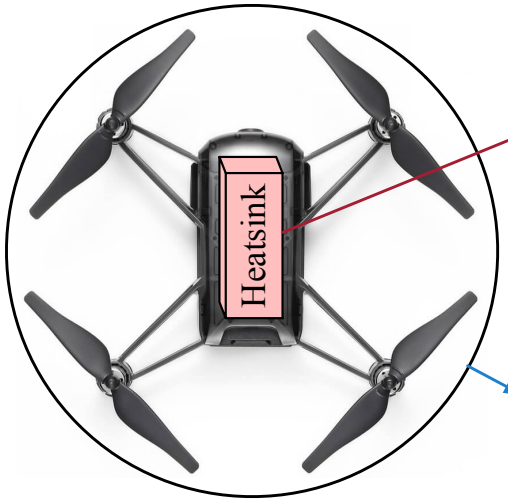
Low-voltage operation → payload weight↓

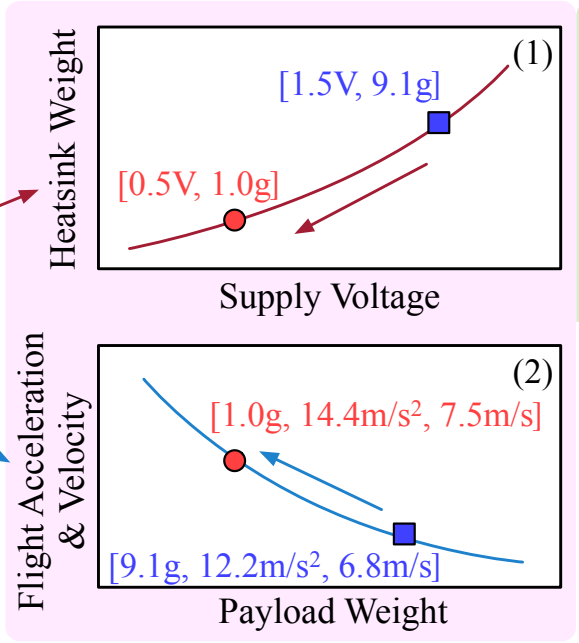
→ **safe velocity↑**

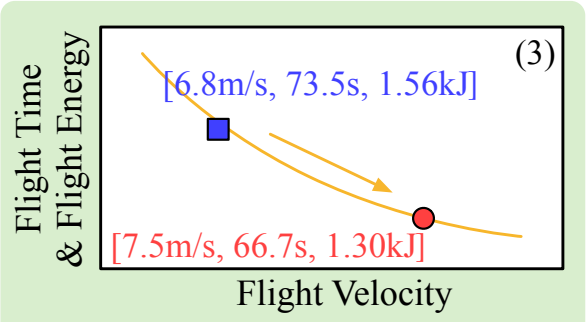
Small Compute Power, Large System Impact!



Low-Voltage Operation (Supply Voltage↓) → Voltage-Physics Relationship (Payload↓, Flight Velocity↑) → Mission Performance (Flight Time↓, Energy↓, #Missions↑)

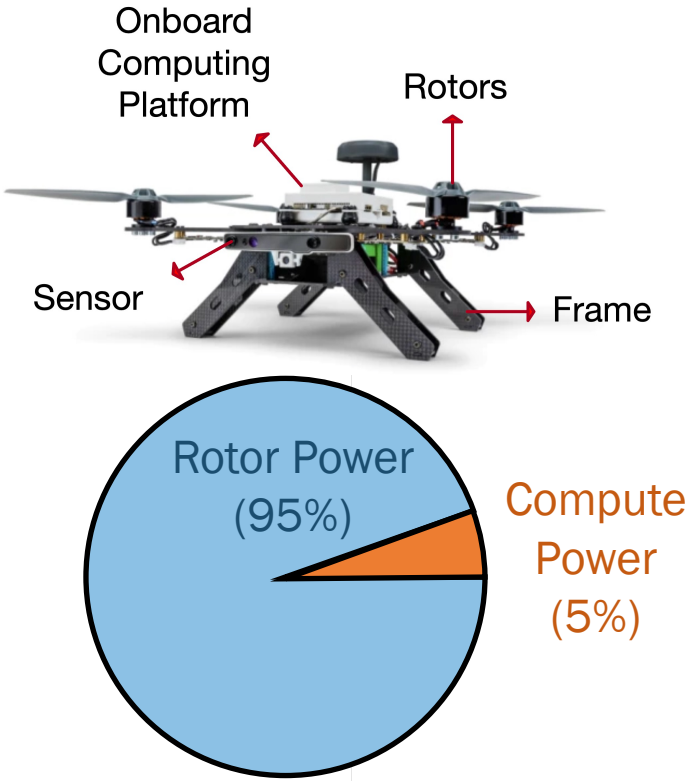




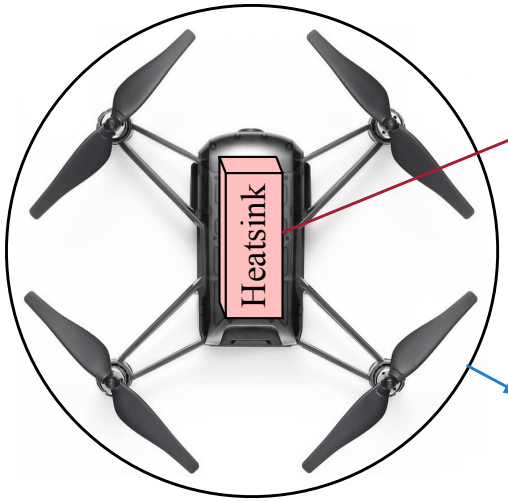


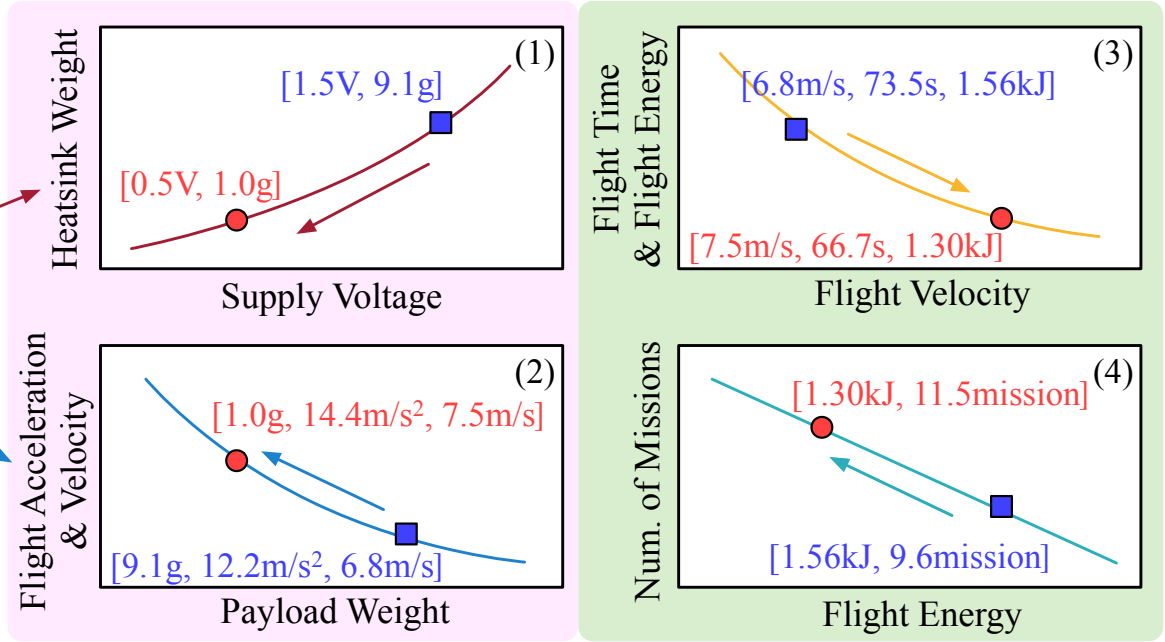
Low-voltage operation → payload weight↓ → safe velocity↑ → **flight time & energy↓**

Small Compute Power, Large System Impact!



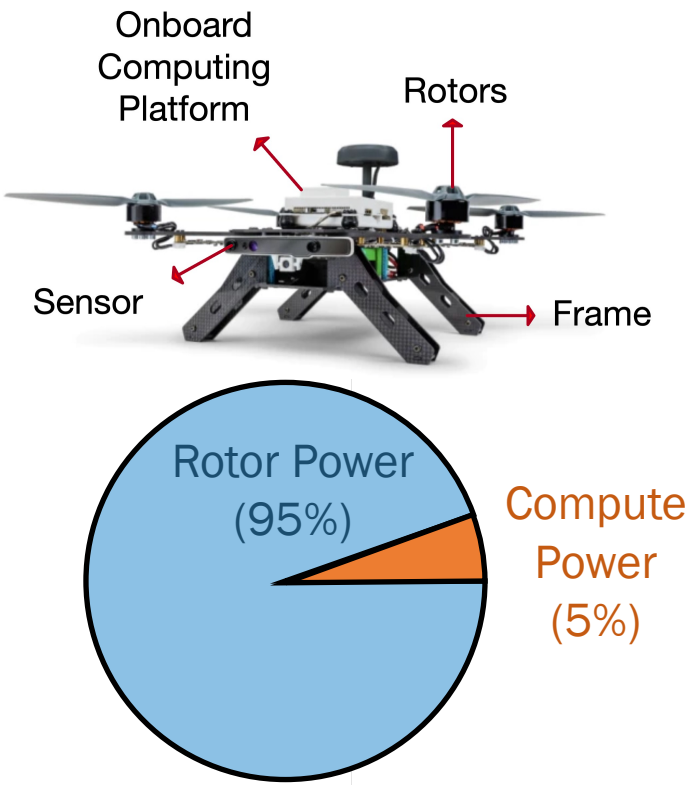
Low-Voltage Operation (Supply Voltage↓) → Voltage-Physics Relationship (Payload↓, Flight Velocity↑) → Mission Performance (Flight Time↓, Energy↓, #Missions↑)

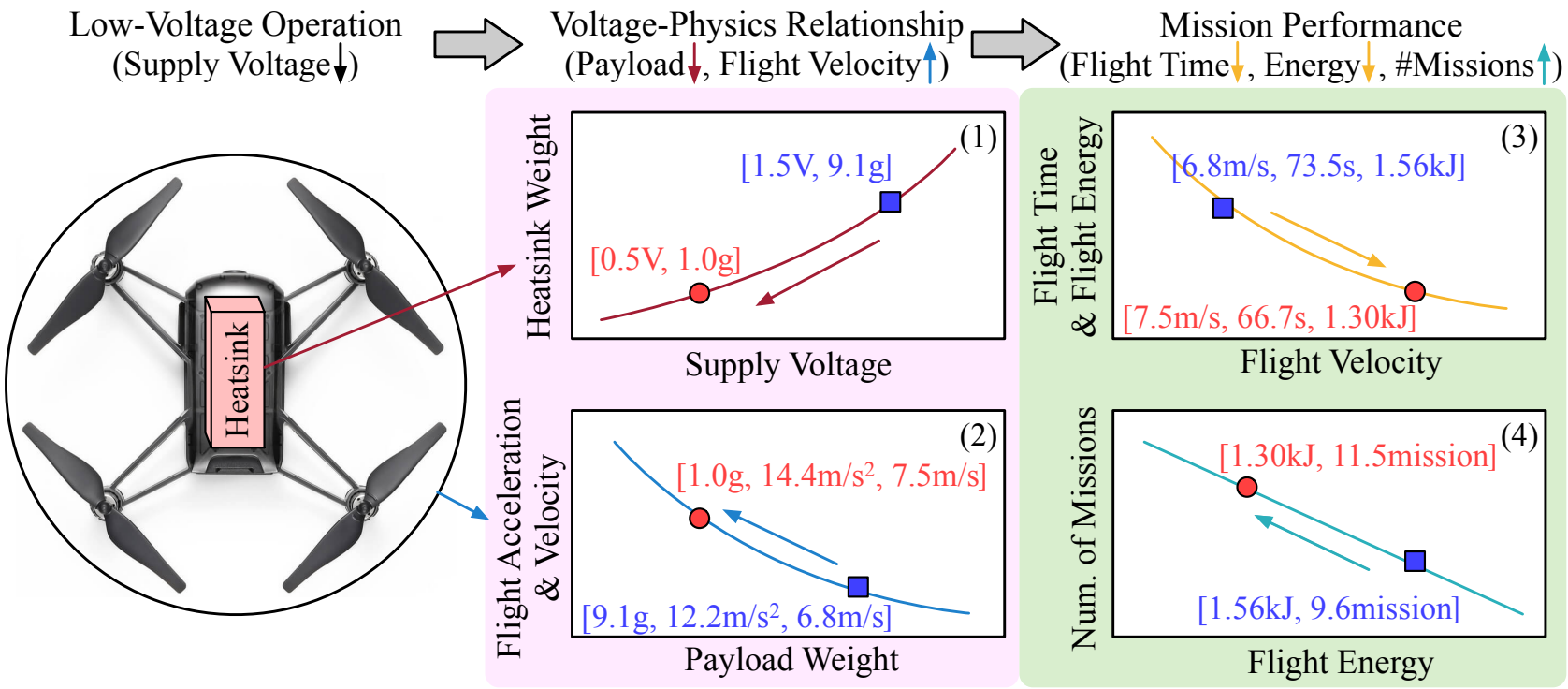




Low-voltage operation → payload weight↓ → safe velocity↑ → flight time & energy↓ num of missions↑

Small Compute Power, Large System Impact!

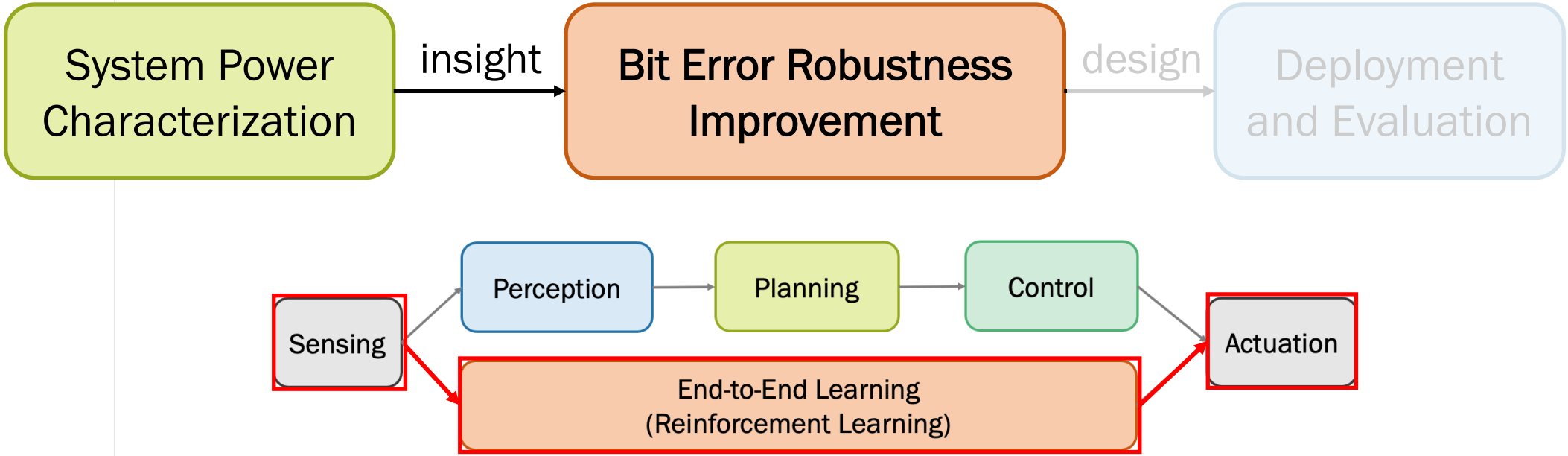




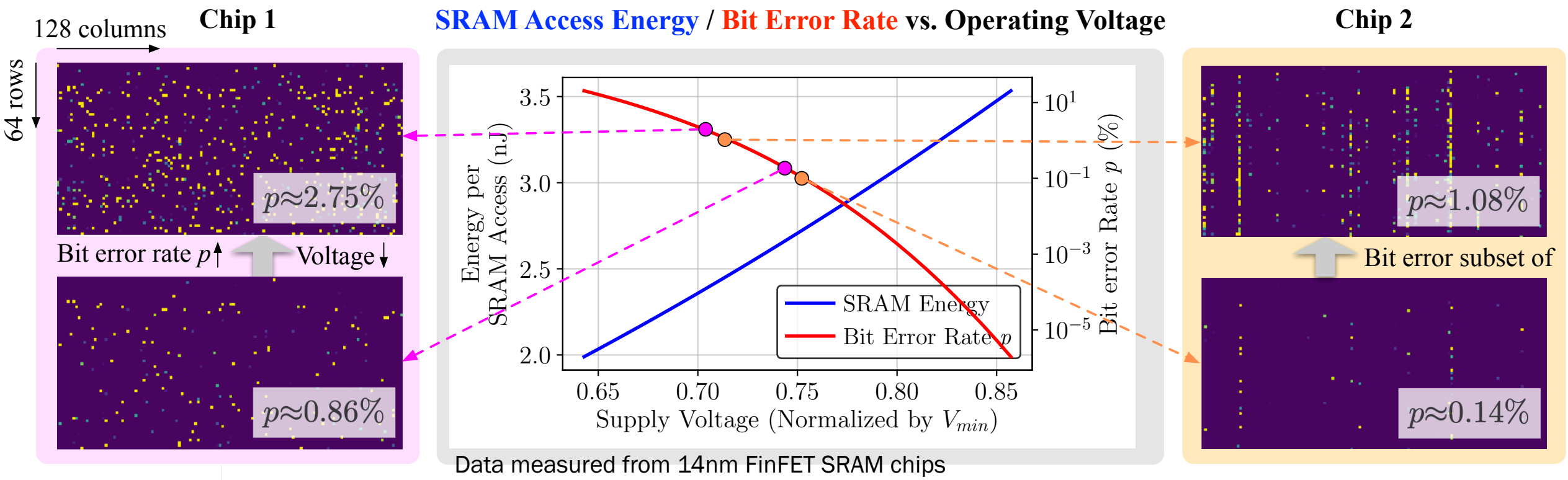
Compute power has huge impacts on end-to-end autonomous system mission energy

BERRY Framework

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



Challenge: Low Voltage Induces Faults



Data measured from 14nm FinFET SRAM chips

Operating below rated voltage range results in memory bit errors, negatively impacting safety

BERRY Framework

- **Design Principle**: Cross-layer robust learning framework, integrates *algorithm-level* error-aware training with *hardware-level* bit failures, aiming to optimize *system-level* robustness/efficiency/performance under low operating voltage.

BERRY Framework

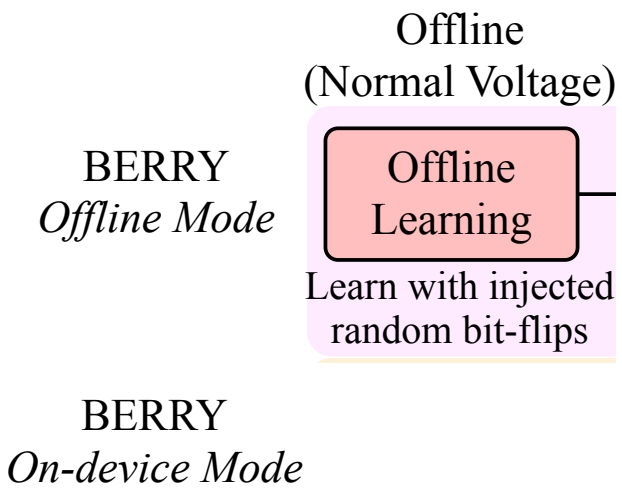
- **Design Principle**: Cross-layer robust learning framework, integrates *algorithm-level* error-aware training with *hardware-level* bit failures, aiming to optimize *system-level* robustness/efficiency/performance under low operating voltage.

BERRY
Offline Mode

BERRY
On-device Mode

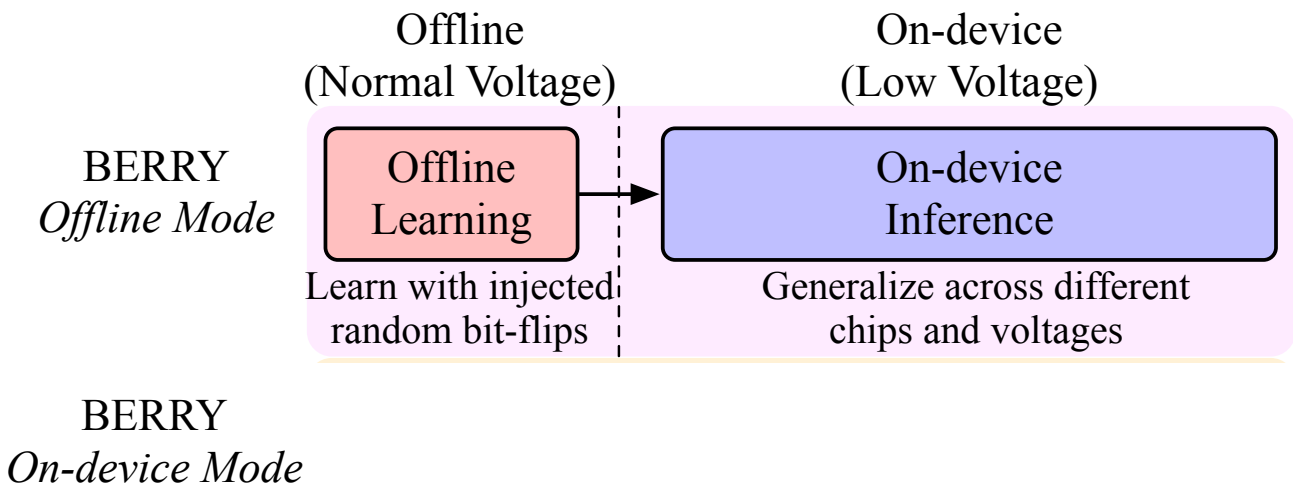
BERRY Framework

- **Design Principle**: Cross-layer robust learning framework, integrates *algorithm-level* error-aware training with *hardware-level* bit failures, aiming to optimize *system-level* robustness/efficiency/performance under low operating voltage.



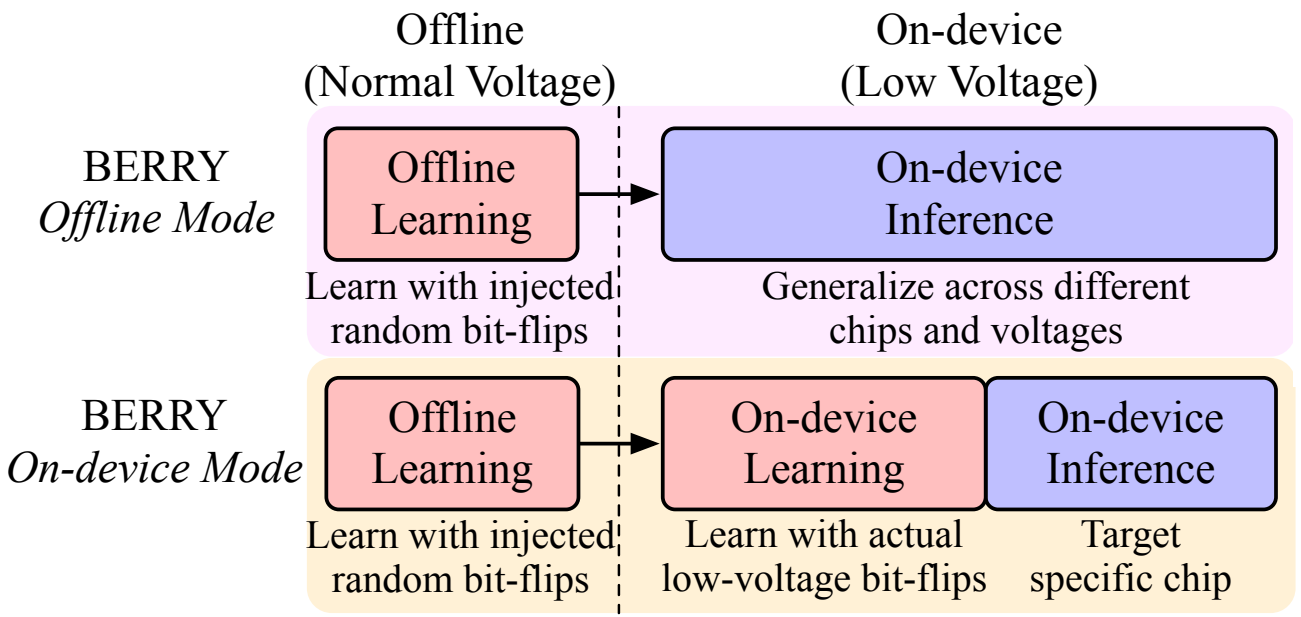
BERRY Framework

- **Design Principle**: Cross-layer robust learning framework, integrates *algorithm-level* error-aware training with *hardware-level* bit failures, aiming to optimize *system-level* robustness/efficiency/performance under low operating voltage.



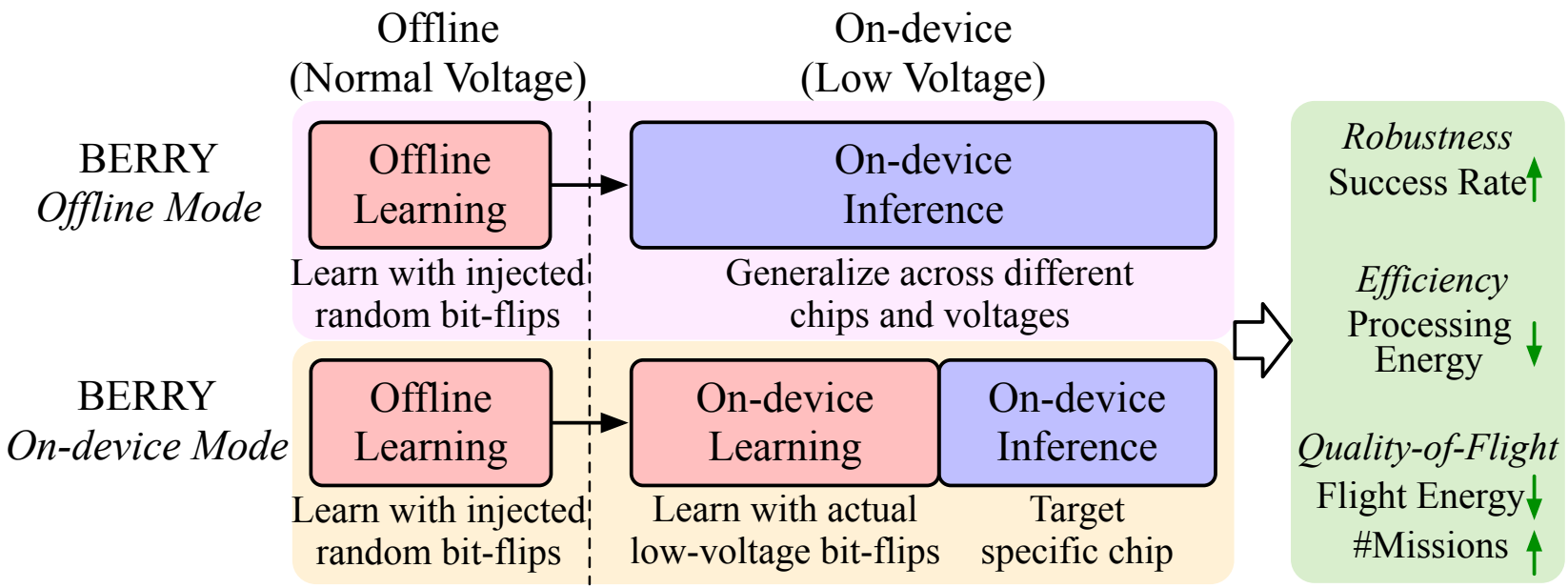
BERRY Framework

- **Design Principle**: Cross-layer robust learning framework, integrates *algorithm-level* error-aware training with *hardware-level* bit failures, aiming to optimize *system-level* robustness/efficiency/performance under low operating voltage.



BERRY Framework

- **Design Principle**: Cross-layer robust learning framework, integrates *algorithm-level* error-aware training with *hardware-level* bit failures, aiming to optimize *system-level* robustness/efficiency/performance under low operating voltage.



BERRY Framework

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

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

BERRY Framework

Start: Initialize policy

Goal: learn an action-value policy

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

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

BERRY Framework

Start: Initialize policy

Learning with experience replay buffer

Goal: learn an action-value policy

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

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

BERRY Framework

Start: Initialize policy

Learning with experience replay buffer

Goal: learn a robust action-value policy

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

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

BERRY Framework

Start: Initialize policy

Learning with experience replay buffer

Error injection

Goal: learn a robust action-value policy

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

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

BERRY Framework

Start: Initialize policy

Learning with experience replay buffer

Error injection

Loss and gradient on perturbed weights

Goal: learn a robust action-value policy

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

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

BERRY Framework

Start: Initialize policy

Learning with experience replay buffer

Error injection

Loss and gradient on perturbed weights

Gradient update (clean & perturbed)

Goal: learn a robust action-value policy

*Please refer to paper for algorithm details

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

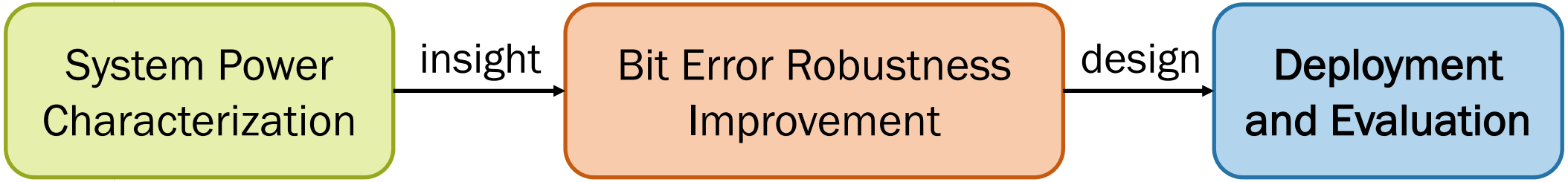
procedure BERRY(p)
  Initialize action-value function Q with weight θ
  Initialize target action-value function $\hat{Q}$ with weight $\theta^- = \theta$
  for *episode* $e = 1$ *to* E **do**
    for *time step* $t = 1$ *to* T **do**
      Given state s_t, take action a_t based on Q (ϵ-greedy)
     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

BERRY Framework Highlights

Previous Error-Aware Training	BERRY
Profiled errors injected during offline training, resulting in robust model during inference	BERRY targets learning (both offline and on-device), meaning that learning can occur on low-voltage devices with bit errors affecting parameters
Supervised learning tasks (e.g., object classification)	BERRY targets reinforcement learning
Model robustness	BERRY tackles complex relationship between low-voltage compute and cyber-physical autonomous systems

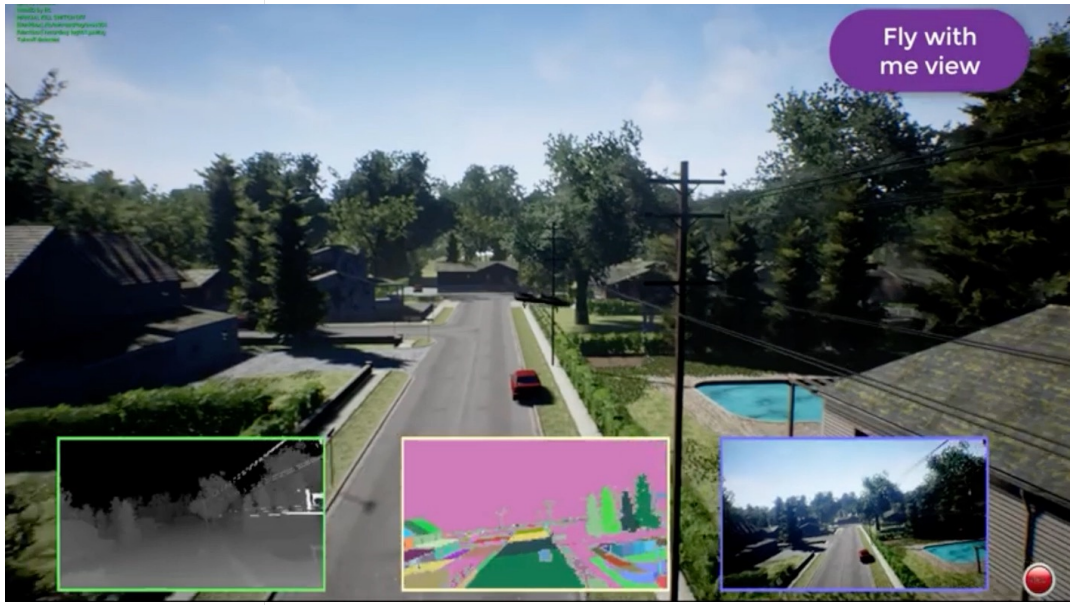
BERRY Framework

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



Experimental Setup

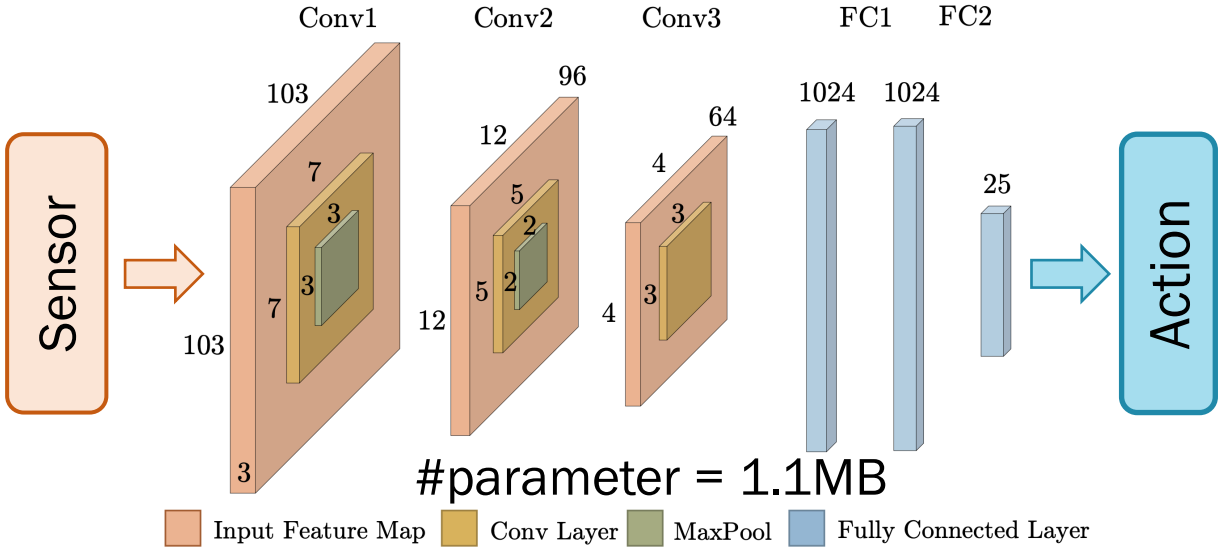
- Simulation Platform



Unreal Engine + AirSim

(3D realistic environments) (Drone dynamics)

- Task
  - point-to-point autonomous navigation (e.g., package delivery)
- Policy Architecture



#parameter = 1.1MB

Trained with Deep-Q-Network (DQN)

Experimental Setup

- Drone Platforms
  - Bitcraze Crazyflie

Nano-Drone
27g takeoff weight
15g max payload
250mAh battery

  - DJI Tello

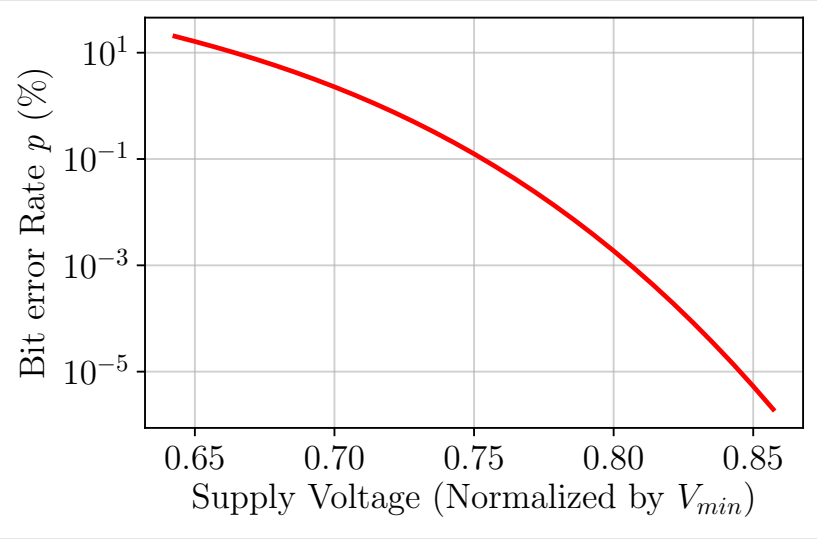
Micro-Drone
80g takeoff weight
70g max payload
1100mAh battery

- Evaluation Metrics
  - Compute-level: Processing energy
  - System-level:
    - Flight success rate ↑
    - Flight time ↓
    - Flight energy ↓
    - Number of completed missions ↑

All reported results are averaged from 500 different fault maps

Robustness Improvement

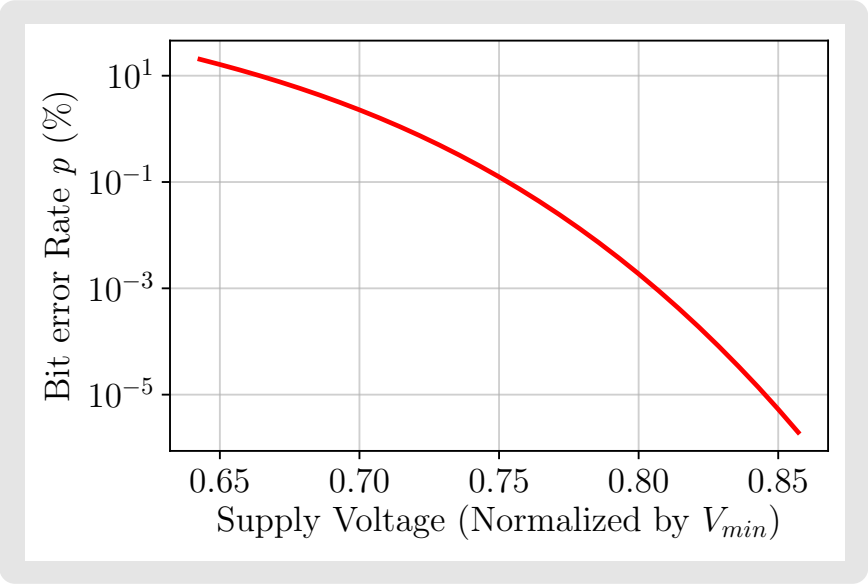
Bit Error Rate vs. Voltage



Voltage ↓ ⇨ Bit error rate ↑

Robustness Improvement

Bit Error Rate vs. Voltage



Voltage ↓ ⇨ Bit error rate ↑

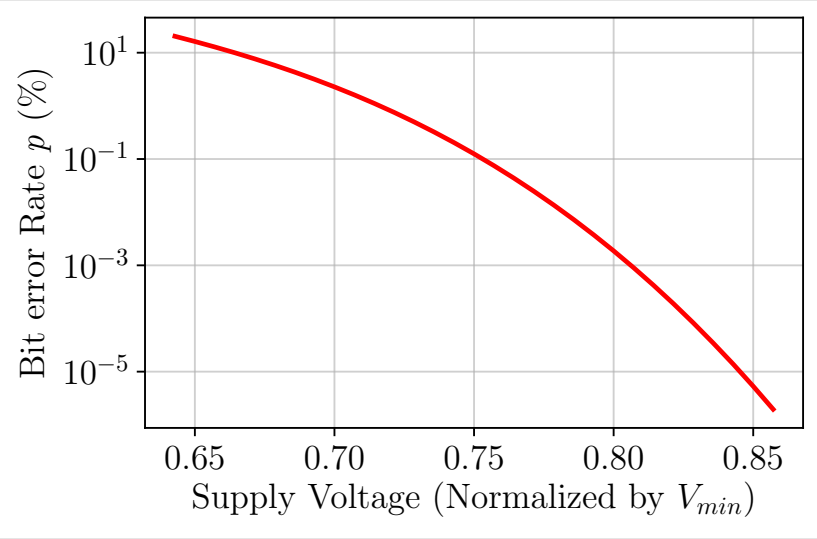
Flight Success Rate under Various Bit Error Rates p (Low Voltages)

Autonomy Scheme	Success Rate	Bit-Error Success Rate			
		p=0.01%	p=0.05%	p=0.1%	p=0.5%
Classical	88.4%	84.0%	78.2%	69.2%	48.6%

(Success Rate: percentage of successful flight of all flight trails)

Robustness Improvement

Bit Error Rate vs. Voltage



Voltage ↓ ⇨ Bit error rate ↑

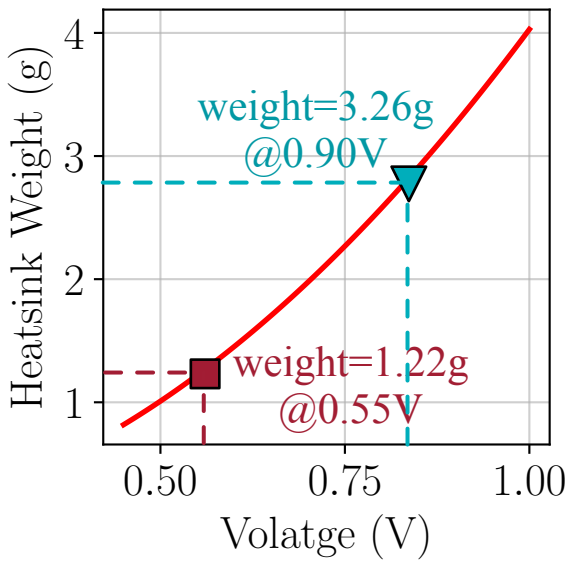
Flight Success Rate under Various Bit Error Rates p (Low Voltages)

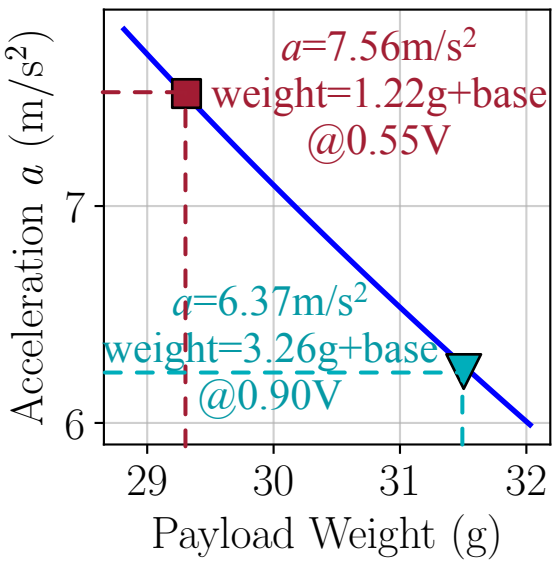
Autonomy Scheme	Success Rate	Bit-Error Success Rate			
		p=0.01%	p=0.05%	p=0.1%	p=0.5%
Classical	88.4%	84.0%	78.2%	69.2%	48.6%
BERRY	**88.8%**	**88.6%**	**86.6%**	**84.4%**	**79.8%**

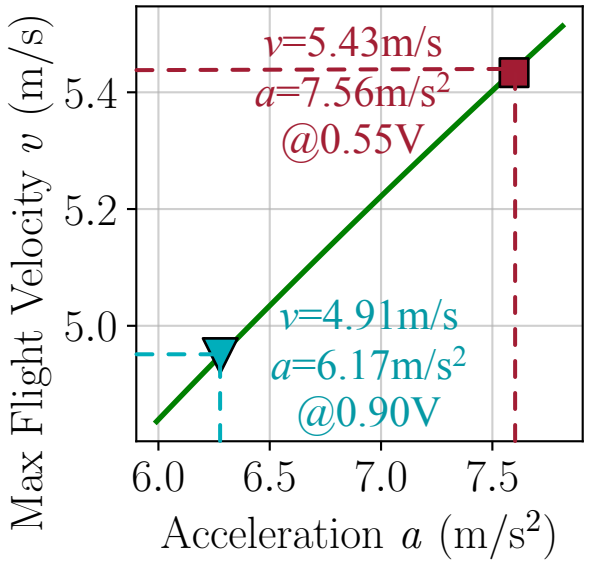
(Success Rate: percentage of successful flight of all flight trails)

BERRY improves mission success rate under bit failures

Mission Efficiency Improvement







(Characterized from Bitcraze Crazyfile drone configuration)

Voltage ↓ ⇨ Heatsink (payload) weight ↓ ⇨ Safe flight velocity ↑

Lowering operating voltage can bring benefits to cyber-physical drones (e.g., safe velocity)

Mission Efficiency Improvement

Low-Voltage Operation	
Voltage (V)	Bit Error Rate p (%)
1	0
$0.86V_{min}$	1.96×10^{-6}
$0.84V_{min}$	1.38×10^{-5}
$0.83V_{min}$	8.23×10^{-5}
$0.81V_{min}$	4.22×10^{-4}
$0.80V_{min}$	1.87×10^{-3}
$0.79V_{min}$	7.25×10^{-3}
$\mathbf{0.77}V_{min}$	$\mathbf{2.47\times10^{-2}}$
$0.76V_{min}$	7.49×10^{-2}
$0.74V_{min}$	2.03×10^{-1}
$0.73V_{min}$	4.98×10^{-1}
$0.71V_{min}$	1.11
$0.68V_{min}$	5.80
$0.64V_{min}$	20.36

Bit errors increase

Mission Efficiency Improvement

Low-Voltage Operation	
Voltage (V)	Bit Error Rate p (%)
1	0
$0.86V_{min}$	1.96×10^{-6}
$0.84V_{min}$	1.38×10^{-5}
$0.83V_{min}$	8.23×10^{-5}
$0.81V_{min}$	4.22×10^{-4}
$0.80V_{min}$	1.87×10^{-3}
$0.79V_{min}$	7.25×10^{-3}
$\mathbf{0.77V_{min}}$	$\mathbf{2.47\times10^{-2}}$
$0.76V_{min}$	7.49×10^{-2}
$0.74V_{min}$	2.03×10^{-1}
$0.73V_{min}$	4.98×10^{-1}
$0.71V_{min}$	1.11
$0.68V_{min}$	5.80
$0.64V_{min}$	20.36

Bit errors increase

Mission Efficiency Improvement

Low-Voltage Operation		Processing
Voltage (V)	Bit Error Rate p (%)	Energy Savings (%)
1	0	-
$0.86V_{min}$	1.96×10^{-6}	$2.77\times$
$0.84V_{min}$	1.38×10^{-5}	$2.87\times$
$0.83V_{min}$	8.23×10^{-5}	$2.97\times$
$0.81V_{min}$	4.22×10^{-4}	$3.07\times$
$0.80V_{min}$	1.87×10^{-3}	$3.18\times$
$0.79V_{min}$	7.25×10^{-3}	$3.30\times$
$0.77V_{min}$	**2.47×10^{-2}**	**$3.43\times$**
$0.76V_{min}$	7.49×10^{-2}	$3.55\times$
$0.74V_{min}$	2.03×10^{-1}	$3.69\times$
$0.73V_{min}$	4.98×10^{-1}	$3.84\times$
$0.71V_{min}$	1.11	$3.99\times$
$0.68V_{min}$	5.80	$4.42\times$
$0.64V_{min}$	20.36	$4.93\times$

Bit errors increase

Save processing energy

Mission Efficiency Improvement

Low-Voltage Operation		Processing	Robustness
Voltage (V)	Bit Error Rate p (%)	Energy Savings (%)	Success Rate (%)
1	0	-	88.4
$0.86V_{min}$	1.96×10^{-6}	2.77×	88.0
$0.84V_{min}$	1.38×10^{-5}	2.87×	89.2
$0.83V_{min}$	8.23×10^{-5}	2.97×	89.0
$0.81V_{min}$	4.22×10^{-4}	3.07×	88.8
$0.80V_{min}$	1.87×10^{-3}	3.18×	88.6
$0.79V_{min}$	7.25×10^{-3}	3.30×	88.6
$0.77V_{min}$	**2.47×10^{-2}**	**3.43×**	**88.4**
$0.76V_{min}$	7.49×10^{-2}	3.55×	86.2
$0.74V_{min}$	2.03×10^{-1}	3.69×	83.4
$0.73V_{min}$	4.98×10^{-1}	3.84×	79.0
$0.71V_{min}$	1.11	3.99×	74.4
$0.68V_{min}$	5.80	4.42×	63.2
$0.64V_{min}$	20.36	4.93×	50.4

Bit errors increase

Save processing energy

Maintain robustness under faults

Mission Efficiency Improvement

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)
1	0	-	88.4	14.89
$0.86V_{min}$	1.96×10^{-6}	2.77×	88.0	14.93
$0.84V_{min}$	1.38×10^{-5}	2.87×	89.2	14.86
$0.83V_{min}$	8.23×10^{-5}	2.97×	89.0	14.91
$0.81V_{min}$	4.22×10^{-4}	3.07×	88.8	14.96
$0.80V_{min}$	1.87×10^{-3}	3.18×	88.6	14.94
$0.79V_{min}$	7.25×10^{-3}	3.30×	88.6	14.94
$0.77V_{min}$	**2.47×10^{-2}**	**3.43×**	**88.4**	**14.91**
$0.76V_{min}$	7.49×10^{-2}	3.55×	86.2	15.71
$0.74V_{min}$	2.03×10^{-1}	3.69×	83.4	16.58
$0.73V_{min}$	4.98×10^{-1}	3.84×	79.0	18.03
$0.71V_{min}$	1.11	3.99×	74.4	19.46
$0.68V_{min}$	5.80	4.42×	63.2	21.84
$0.64V_{min}$	20.36	4.93×	50.4	24.52

Bit errors increase

Save processing energy

Maintain robustness under faults

Maintain trajectory distance under faults

Mission Efficiency Improvement

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)
1	0	-	88.4	14.89	6.81
$0.86V_{min}$	1.96×10^{-6}	2.77×	88.0	14.93	6.51
$0.84V_{min}$	1.38×10^{-5}	2.87×	89.2	14.86	6.48
$0.83V_{min}$	8.23×10^{-5}	2.97×	89.0	14.91	6.46
$0.81V_{min}$	4.22×10^{-4}	3.07×	88.8	14.96	6.45
$0.80V_{min}$	1.87×10^{-3}	3.18×	88.6	14.94	6.42
$0.79V_{min}$	7.25×10^{-3}	3.30×	88.6	14.94	6.39
$0.77V_{min}$	**2.47×10^{-2}**	**3.43×**	**88.4**	**14.91**	**6.35**
$0.76V_{min}$	7.49×10^{-2}	3.55×	86.2	15.71	6.67
$0.74V_{min}$	2.03×10^{-1}	3.69×	83.4	16.58	7.03
$0.73V_{min}$	4.98×10^{-1}	3.84×	79.0	18.03	7.61
$0.71V_{min}$	1.11	3.99×	74.4	19.46	8.18
$0.68V_{min}$	5.80	4.42×	63.2	21.84	9.09
$0.64V_{min}$	20.36	4.93×	50.4	24.52	10.11

BERRY improves mission efficiency (flight time↓,

Mission Efficiency Improvement

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
1	0	-	88.4	14.89	6.81	53.19	-
$0.86V_{min}$	1.96×10^{-6}	2.77×	88.0	14.93	6.51	47.23	-11.21%
$0.84V_{min}$	1.38×10^{-5}	2.87×	89.2	14.86	6.48	46.66	-12.28%
$0.83V_{min}$	8.23×10^{-5}	2.97×	89.0	14.91	6.46	46.41	-12.73%
$0.81V_{min}$	4.22×10^{-4}	3.07×	88.8	14.96	6.45	46.22	-13.11%
$0.80V_{min}$	1.87×10^{-3}	3.18×	88.6	14.94	6.42	45.80	-13.90%
$0.79V_{min}$	7.25×10^{-3}	3.30×	88.6	14.94	6.39	45.38	-14.67%
$0.77V_{min}$	**2.47×10^{-2}**	**3.43×**	**88.4**	**14.91**	**6.35**	**44.88**	**-15.62%**
$0.76V_{min}$	7.49×10^{-2}	3.55×	86.2	15.71	6.67	46.90	-11.82%
$0.74V_{min}$	2.03×10^{-1}	3.69×	83.4	16.58	7.03	49.14	-7.61%
$0.73V_{min}$	4.98×10^{-1}	3.84×	79.0	18.03	7.61	52.98	-0.39%
$0.71V_{min}$	1.11	3.99×	74.4	19.46	8.18	56.62	-6.45%
$0.68V_{min}$	5.80	4.42×	63.2	21.84	9.09	61.96	+16.49%
$0.64V_{min}$	20.36	4.93×	50.4	24.52	10.11	67.83	+27.53%

BERRY improves mission efficiency (flight time ↓, flight energy ↓, (15.62%)

Mission Efficiency Improvement

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×	88.0	14.93	6.51	47.23	-11.21%	62.05	+12.12%
$0.84V_{min}$	1.38×10^{-5}	2.87×	89.2	14.86	6.48	46.66	-12.28%	63.66	+15.03%
$0.83V_{min}$	8.23×10^{-5}	2.97×	89.0	14.91	6.46	46.41	-12.73%	63.85	+15.37%
$0.81V_{min}$	4.22×10^{-4}	3.07×	88.8	14.96	6.45	46.22	-13.11%	63.98	+15.61%
$0.80V_{min}$	1.87×10^{-3}	3.18×	88.6	14.94	6.42	45.80	-13.90%	64.42	+16.40%
$0.79V_{min}$	7.25×10^{-3}	3.30×	88.6	14.94	6.39	45.38	-14.67%	65.01	+17.46%
$0.77V_{min}$	**2.47×10^{-2}**	**3.43×**	**88.4**	**14.91**	**6.35**	**44.88**	**-15.62%**	**65.59**	**+18.51%**
$0.76V_{min}$	7.49×10^{-2}	3.55×	86.2	15.71	6.67	46.90	-11.82%	61.20	+10.58%
$0.74V_{min}$	2.03×10^{-1}	3.69×	83.4	16.58	7.03	49.14	-7.61%	56.52	+2.12%
$0.73V_{min}$	4.98×10^{-1}	3.84×	79.0	18.03	7.61	52.98	-0.39%	49.66	-10.27%
$0.71V_{min}$	1.11	3.99×	74.4	19.46	8.18	56.62	-6.45%	43.75	-20.95%
$0.68V_{min}$	5.80	4.42×	63.2	21.84	9.09	61.96	+16.49%	33.96	-38.64%
$0.64V_{min}$	20.36	4.93×	50.4	24.52	10.11	67.83	+27.53%	24.74	-55.30%

BERRY improves mission efficiency (flight time ↓, flight energy ↓ (15.62%), number of completed missions ↑ (18.51%))

Effectiveness Across Environments

Environments	Sparse Obstacle	Medium Obstacle	Dense Obstacle
Operating Voltage			
Flight Energy Savings			
Num. of Success Mission Increase			

Effectiveness Across Environments

Environments	Sparse Obstacle	Medium Obstacle	Dense Obstacle
Operating Voltage	0.76 V_{min}	0.77 V_{min}	0.80 V_{min}
Flight Energy Savings			
Num. of Success Mission Increase			

Effectiveness Across Environments

Environments	Sparse Obstacle	Medium Obstacle	Dense Obstacle
Operating Voltage	0.76 V_{min}	0.77 V_{min}	0.80 V_{min}
Flight Energy Savings	14.0%	15.6%	15.6%
Num. of Success Mission Increase			

Effectiveness Across Environments

Environments	Sparse Obstacle	Medium Obstacle	Dense Obstacle
Operating Voltage	0.76 V_{min}	0.77 V_{min}	0.80 V_{min}
Flight Energy Savings	14.0%	15.6%	15.6%
Num. of Success Mission Increase	17.0%	18.6%	17.9%

BERRY is adaptive across environments, and consistently improves robustness and efficiency

Effectiveness Across Drone Platforms

Bitcraze Crazyflie

DJI Tello

Drone Type	Network Policy	Rotor Power	Compute Power	BERRY Flight Energy Reduction	BERRY #Missions Increase
Crazyflie	3*Conv + 2*FC	93.5%	6.5%	15.62%	18.51%
DJI Tello	3*Conv + 2*FC	97.2%	2.8%	9.91%	9.96%

Effectiveness Across Drone Platforms

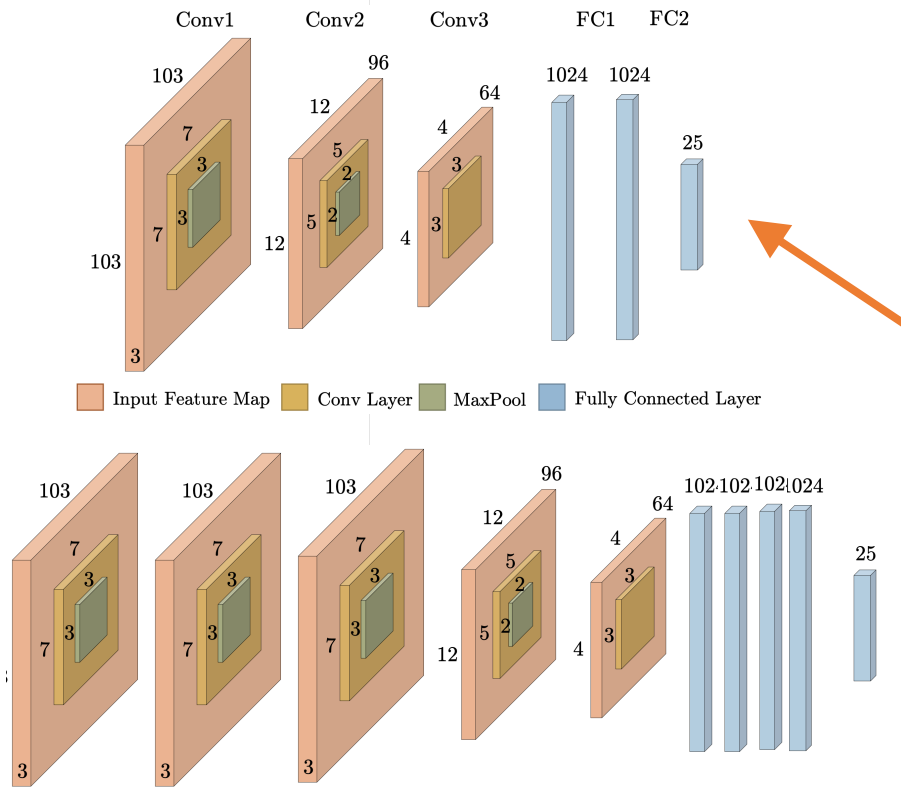
Bitcraze Crazyflie

DJI Tello

Drone Type	Network Policy	Rotor Power	Compute Power	BERRY Flight Energy Reduction	BERRY #Missions Increase
Crazyflie	3*Conv + 2*FC	93.5%	6.5%	15.62%	18.51%
DJI Tello	3*Conv + 2*FC	97.2%	2.8%	9.91%	9.96%

BERRY is adaptive across drone types, and consistently improves robustness and efficiency

Effectiveness Across Model Architectures

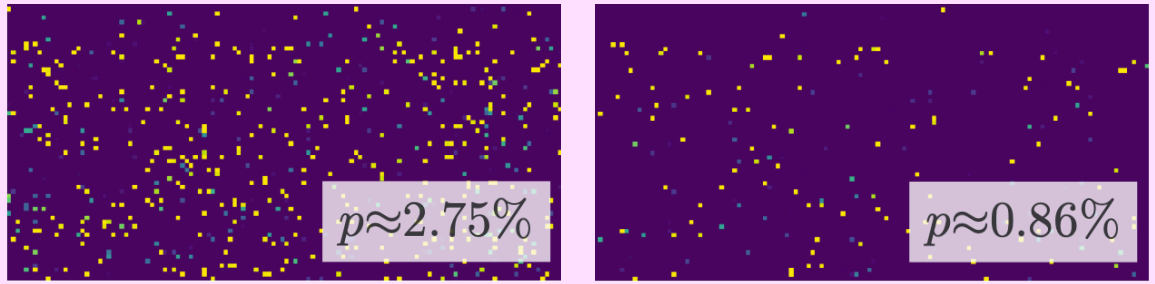


Drone Type	Network Policy	Rotor Power	Compute Power	BERRY Flight Energy Reduction	BERRY #Missions Increase
Crazyflie	3*Conv + 2*FC	93.5%	6.5%	15.62%	18.51%
DJI Tello	3*Conv + 2*FC	97.2%	2.8%	9.91%	9.96%
DJI Tello	5*Conv + 4*FC	94.9%	4.1%	13.12%	14.38%

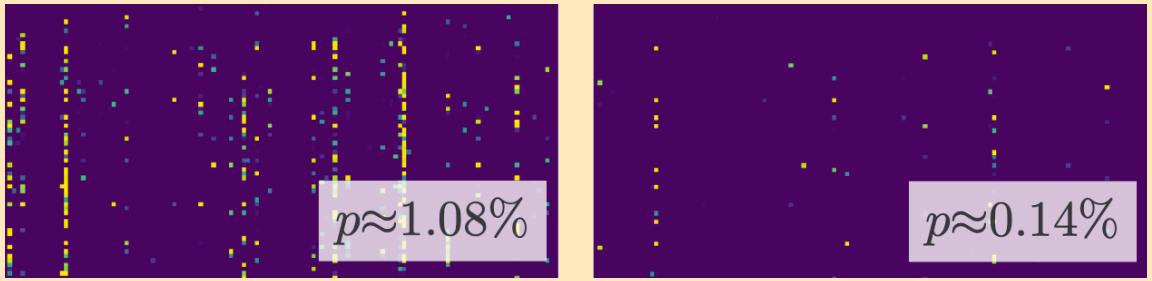
BERRY is adaptive across models, and consistently improves robustness and efficiency

Effectiveness Across Profiled Bit Errors

Chip 1



Chip 2



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	

BERRY is adaptive across bit error patterns, and consistently improves robustness and efficiency

On-Device Error-Aware Robust Learning

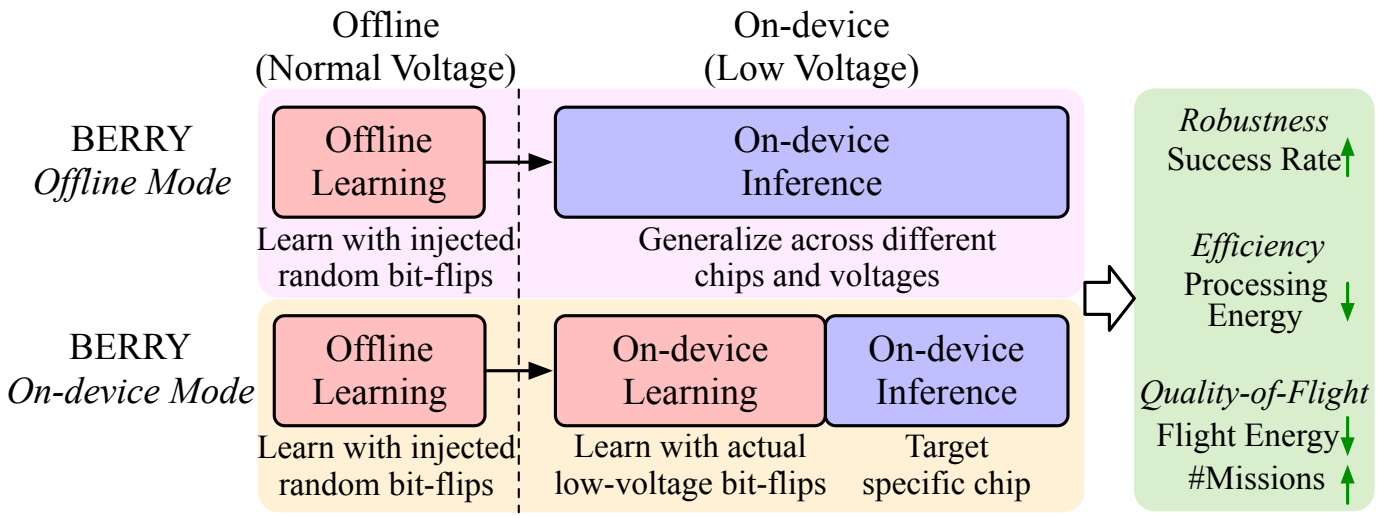
Low-Voltage Operation				Operating Efficiency	Robustness	Quality-of-Flight	
Num. of Learning Steps		Operating Voltage	Learning Energy (J)	Energy Savings	Success Rate (%)	Flight Energy (J)	Num. of Missions*
On-Device BERRY	4000	$0.77V_{min}$	1849	3.43×	84.6	264.2	48.19
		$0.70V_{min}$	1807	4.16×	82.4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

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

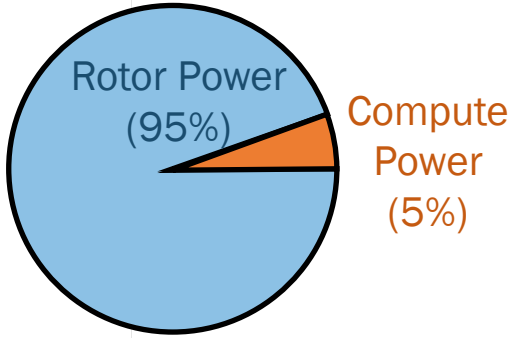
BERRY on-device robust learning consumes on-the-fly learning energy, but can enable lower operating voltage, improved robustness and higher mission efficiency.

*Please refer to paper for evaluation result details

Summary



System Power Characterization → insight → Bit Error Robustness Improvement → design → Deployment and Evaluation



Compute power: small portion, big impact!

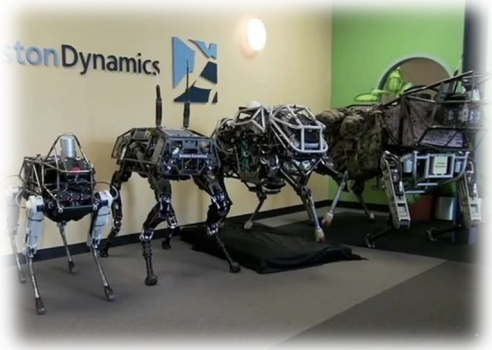
Aggressive energy-saving yet computational-resilient

Motivation

Reliability

Goal: Improve operational resiliency
(Temporal/Spatial Redundancy)

Autonomous Machines



Performance

Goal: Improve mission quality
(Autonomy Algorithms)

Performance-Efficiency-Reliability
Co-Optimization

Efficiency

Goal: Improve compute efficiency
(Software Optimization,
Hardware Architecture)

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

Zishen Wan[1], Nandhini Chandramoorthy[2], Karthik Swaminathan[2], Pin-Yu Chen[2], Vijay Janapa Reddi[3], Arijit Raychowdhury[1]

zishenwan@gatech.edu

[1]Georgia Institute of Technology [2]IBM Research [3]Harvard University

THANK YOU!