VPP: The Vulnerability-Proportional Protection Paradigm Towards Reliable Autonomous Machines

Zishen Wan[1*], Yiming Gan[2*], Bo Yu[3], Arijit Raychowdhury[1], Yuhao Zhu[2]
[1]Georgia Institute of Technology, Atlanta, GA [2]University of Rochester, Rochester, NY [3]PerceptIn, Fremont, CA

1 INTRODUCTION

The next ubiquitous computing platform, after personal computers and smartphones, is likely one of the autonomous natures, such as drones, robots, and self-driving cars, which have moved from mere lab concepts to permeating almost every aspect of our society [16, 20, 25]. Behind the proliferation of autonomous machines is the critical need to ensure reliability [7, 22–24]. Almost every vendor, be it in the software, hardware, or systems segment, has to conform to functional safety standards when shipping products for automotives.

Today's resiliency solutions to autonomous machines, however, all make fundamental trade-offs between resiliency and cost, which manifests as high overhead in performance, energy, and chip area. For instance, hardware modular redundancy provides high safety but more than doubles the area and energy cost [1]. The reason is that today's solutions are of the "one-size-fits-all" nature: they use the same protection scheme throughout the entire computing stack of autonomous machines. As a result, they have to accommodate the least robust component, leading to a high protection overhead.

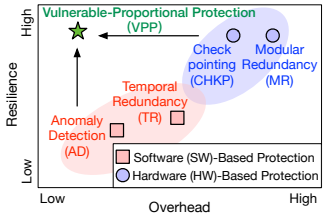
The insight of this paper is that for a resiliency solution to provide high protection coverage while introducing little cost, we must exploit the *inherent* robustness variations in the domain-specific autonomous machine computing. In particular, we show that the different autonomous machine kernels differ significantly in their inherent robustness and performance. Building on top of that, we propose a *Vulnerable-Proportional Protection (VPP)* design paradigm, in which the protection budget, be it spatially (e.g., modular redundancy) or temporally (e.g., re-execution), should be inversely proportional to the inherent robustness of a task in the autonomous machine system. In stark contrast to the existing "one-size-fits-all" strategy, VPP wisely allocates the protection budget, thus achieving the same protection coverage with little overhead, which provides a blueprint design paradigm towards reliable autonomous machines.

2 DESIGN LANDSCAPE OF RESILIENT AUTONOMOUS MACHINES

Different protection techniques exhibit distinct performance, efficiency, and resilience impact on autonomous machines. We compare four representative software and hardware protection techniques and reveal that conventional "one-size-fits-all" approaches are limited by the tradeoff in overhead and resilience improvement (Fig.1).

Various sources of errors can affect autonomous machines, including soft errors, adversarial attacks, and software bugs [24, 29]. We consider hardware bit-flips (i.e., soft error) in this paper. The exacerbating impact of soft errors has been recently emphasized by industrial studies [5, 9], where radiations and temperature change can result in random bit flip in compute units and memory cells.

*Equal Contributions.



Metric	SW-Based		HW-Based		VPP
	AD	TR	MR	CHKP	(Ours)
Latency	✗	✗	✔	✗	✔
Energy	✔	✗	✗	✔	✔
Cost	✔	✔	✗	✗	✔
Resilience	✗	✗	✔	✔	✔

Fig. 1: Design landscape of different software and hardware-based protection techniques for resilient autonomous machines. Our proposed *Vulnerable-Proportional Protection (VPP)* design paradigm co-optimizes performance, energy efficiency, and resilience.

Software-based protection scheme usually exhibits advantages in lower cost and power overhead, but suffer from compute latency overhead and non-completely recovery from faults. For example, anomaly detection [7, 10] can identify abnormal behaviors but may incur high latency overhead due to re-execution and cannot fully mitigate fault impact due to false-positive detection in corner cases. Temporal redundancy [11] executes the code more than once, which can alleviate the threat of silent data corruption but typically incurs large overhead due to the redundant sequential executions.

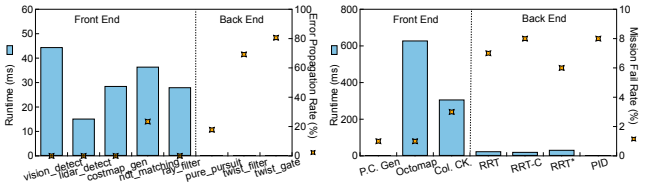
Hardware-based protection scheme usually exhibits advantages in error mitigation but incurs large power overheads and extra cost. For example, modular redundancy [1, 4] makes copies of the processing logic and is effective in fault detection with negligible latency impact while incurring considerable energy and silicon cost. Checkpointing [3] periodically stores a fault-free copy of the processor state with a recovery mechanism under failure, but it brings large runtime overhead due to the store and retrieve procedure that may violate the real-time nature of autonomous machines (Sec.5).

Conventional "one-size-fits-all" hardware or software-based protection is limited by the fundamental trade-off between overhead and resilience in the design space of resilient autonomous machines. As shown in Fig.1, VPP overcomes this trade-off by concurrently optimizing performance-efficiency-resilience and pushes the landscape frontier to the top-left by leveraging the insight from inherent autonomous machine robustness and performance variations.

3 SYSTEM CHARACTERIZATION STUDY

This section characterizes the performance and resilience of different modules in autonomous machines. Autonomous machine computing differentiates from traditional systems in dataflow, software pipeline, compute substrate, and underlying architecture [18]. Evaluated on representative autonomous vehicles (Autoware [19]) and drones (MAVBench [2]), we reveal the inherent robustness and performance variations of autonomous machine systems (Fig.2).

The front-end of autonomous machines (sensing, perception, localization) usually has higher resilience but with higher latency



(a) Vehicle Characterization (b) Drone Characterization

Fig. 2: The performance and resilience trade-offs of autonomous vehicles (Autoware) and drones (MAVBench). X-axis represents algorithm nodes. The front-end exhibits high runtime and high resilience, while the back-end exhibits low runtime and low resilience.

and energy consumption. Front-end modules deal with sensor data and provide semantic results, which is inherently computationally intensive and contributes to the largest latency in autonomous machines [6, 8, 15, 21, 26]. Front-end modules are resilient to errors due to redundant surrounding information, multiple sensor fusion, and inherent robustness of deep learning models in perception tasks.

The back-end of autonomous machines (planning, decision-making, control) is more vulnerable to errors but with lower latency. Back-end modules plan the paths and general control commands, which are computationally light [2, 17, 27]. However, back-end modules directly control the actuators, so faults have a higher probability of propagating the system and influencing the agent's behavior.

4 VPP DESIGN METHODOLOGY

In this section, leveraging the insights of inherent resilience and latency variations of front-end and back-end kernels (Sec.3), we propose an adaptive and cost-effective protection design paradigm for autonomous machine systems, achieving high operation resilience and safety with negligible latency and energy overheads.

Design Paradigm. The key principle of our adaptive protection is *Vulnerable-Proportional Protection (VPP)* - the protection budget is allocated proportionally to the inherent resilience of autonomous machine kernels, with more protection efforts on vulnerable kernels while less on robust kernels. Specifically, we propose to apply software-based protection on front-end kernels and hardware-based protection on back-end kernels (Fig.3), inspired by the insights that the front-end is resilient while the back-end is vulnerable.

Front-end - Software-Based Protection. We use anomaly detection in autonomous machine front-end sensing-perception-localization kernels. We leverage three insights: First, autonomous agents typically generate outputs with strong temporal consistency, thus errors manifested as outliers that break the consistency can be easily detected. Second, front-end kernels have inherent error-masking and error-attenuation capabilities through redundant information and operations such as low-pass filtering and operator union. Third, front-end kernels exhibit rare false positive detection with anomaly detection technique, thus significantly reducing the node re-execution overhead and protection failure cases.

Back-end - Hardware-Based Protection. We use modular redundancy and checkpointing in autonomous machine back-end planning-control kernels. We leverage three insights: First, back-end kernels are very critical to errors, motivating us to strengthen protection with hardware-based methods. Second, the back-end

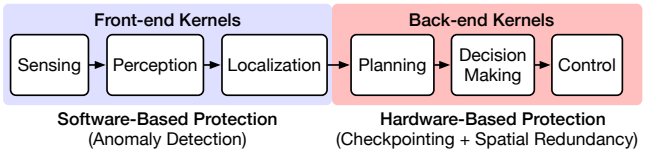


Fig. 3: Adaptive fault protection design paradigm, Vulnerable-Proportional Protection (VPP), with software-based technique for front-end kernels and hardware-based technique for back-end kernels, based on the energy proportional protection design principle.

kernels are extremely lightweight, thus the overhead of redundancy would be small. Third, more false positive detection cases are from the back-end in software protection, resulting in potential protection failure and the need to strengthen from hardware protection.

Particularly, we propose a selective redundancy and checkpointing approach. We only make redundancy copy for the core running back-end modules. In Robot Operating System, we periodically queue the message and the faulty node can directly re-execute before the node communication if faults are detected. This checkpointing method eliminates the large latency overhead from conventional architectural checkpointing/restore methods that violate the real-time nature of autonomous machines.

5 EVALUATION RESULTS

In this section, we evaluate VPP on autonomous machines, and demonstrate that VPP can achieve improved resilience with lower latency and energy overhead, compared with conventional "one-size-fits-all" techniques. Tab.1evaluates the end-to-end Autoware autonomous vehicle system from our concrete vehicle testbed [28].

Table 1: Comparison of proposed protection design VPP with various software and hardware protection schemes, evaluated on end-to-end autonomous driving (AD) performance, efficiency, and resilience.

Protection Scheme		Performance		Power and Operation Time			Resilience
		Latency (*ms*)	Object Dist. (*m*)	AD Power (*W*)*	AD Energy Change (%)	Driving Time (hour)	Error Propag. Rate (%)
No Protection		164	5.00	175	–	7.74	46.5
SW	**AD**	245	5.47	175	+33.14	7.20	24.2
	TR	347	6.05	175	+75.24	6.62	11.7
HW	**MR**	164	5.00	473	+170.29	5.59	0
	CHKP	610	7.56	324	+91.52	6.42	0
VPP (Ours)		173	5.05	175	+4.09	7.67	0

* The vehicle power without autonomous driving (AD) system is 600 W.

The proposed protection design VPP improves autonomous machine resilience. VPP greatly reduces the error propagation rate to 0 by leveraging the inherent error-masking capabilities of front-end kernels and strengthening back-end kernel resilience by hardware-based selected modular redundancy and checkpointing scheme. This level of resilience can satisfy ASIL-D safety criteria.

The proposed protection design VPP incurs low overhead. VPP reduces latency, energy, and system performance overhead by taking advantage of low cost and false-positive detection rate in front-end and low compute latency in back-end of autonomous machines. VPP only brings 5.49% more latency and 4.09% more energy consumption. By contrast, conventional "one-size-fits-all" techniques are limited by tradeoffs in resilience and overhead with reduced driving time duration and worse object avoidance distance.

We have similar observations on drones where VPP generalizes well to small-scale drone system, based on the MAVBench

testbed [2] and drone characterization model [12–14], with improved resilience and negligible overhead. By contrast, the large overhead from conventional "one-size-fits-all" protection techniques result in severe performance degradation in resource-constrained drone systems.

6 CONCLUSION

Reliability and safety are critical for autonomous machines. For the first time, we systematically analyze the design landscape of resilient autonomous machines and reveal the inherent performance-resilience variations. We propose an adaptive protection paradigm VPP and demonstrate its cost-effectiveness on autonomous vehicle and drone systems. We envision the observations and designs will further spur innovations in intelligent swarms and resilient domain-specific solutions for autonomous machine computing.

ACKNOWLEDGEMENT

We thank Shaoshan Liu (from PerceptIn) for his technical support. ZW and AR were supported by CoCoSys, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

REFERENCES

[1] P. Bannon, G. Venkataramanan, D. D. Sarma, and E. Talpes. 2019. Computer and redundancy solution for the full self-driving computer. In *2019 IEEE Hot Chips 31 Symposium (HCS)*. IEEE Computer Society, 1–22.

[2] B. Boroujerdian, H. Genc, S. Krishnan, W. Cui, A. Faust, and V. Reddi. 2018. Mavbench: Micro aerial vehicle benchmarking. In *2018 51st annual IEEE/ACM international symposium on microarchitecture (MICRO)*. IEEE, 894–907.

[3] A. Bourge, O. Muller, and F. Rousseau. 2015. Automatic high-level hardware checkpoint selection for reconfigurable systems. In *2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 155–158.

[4] A. B. de Oliveira, G. S. Rodrigues, F. L. Kastensmidt, N. Added, E. L. Macchione, V. A. Aguiar, N. H. Medina, and M. A. Silveira. 2018. Lockstep dual-core ARM A9: Implementation and resilience analysis under heavy ion-induced soft errors. *IEEE Transactions on Nuclear Science* 65, 8 (2018), 1783–1790.

[5] H. D. Dixit, S. Pendharkar, M. Beadon, C. Mason, T. Chakravarthy, B. Muthiah, and S. Sankar. 2021. Silent data corruptions at scale. *arXiv preprint arXiv:2102.11245* (2021).

[6] Y. Gan, Y. Bo, B. Tian, L. Xu, W. Hu, S. Liu, Q. Liu, Y. Zhang, J. Tang, and Y. Zhu. 2021. Eudoxus: Characterizing and accelerating localization in autonomous machines industry track paper. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 827–840.

[7] Y. Gan, P. Whatmough, J. Leng, B. Yu, S. Liu, and Y. Zhu. 2022. Braum: Analyzing and protecting autonomous machine software stack. In *2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 85–96.

[8] T. Gao, Z. Wan, Y. Zhang, B. Yu, Y. Zhang, S. Liu, and A. Raychowdhury. 2021. IELAS: An ELAS-based energy-efficient accelerator for real-time stereo matching on FPGA platform. In *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 1–4.

[9] P. H. Hochschild, P. Turner, J. C. Mogul, R. Govindaraju, P. Ranganathan, D. E. Culler, and A. Vahdat. 2021. Cores that don't count. In *Proceedings of the Workshop on Hot Topics in Operating Systems*. 9–16.

[10] Y.-S. Hsiao, Z. Wan, T. Jia, R. Ghosal, A. Mahmoud, A. Raychowdhury, D. Brooks, G.-Y. Wei, and V. J. Reddi. 2023. 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)*. IEEE, 1–6.

[11] B. C. Hu, L. Marsso, K. Czarnecki, and M. Chechik. 2022. What to Check: Systematic Selection of Transformations for Analyzing Reliability of Machine Vision Components. In *2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 49–60.

[12] S. Krishnan, Z. Wan, K. Bhardwaj, A. Faust, and V. J. Reddi. 2022. 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)*. IEEE.

[13] S. Krishnan, Z. Wan, K. Bhardwaj, P. Whatmough, A. Faust, S. Neuman, G.-Y. Wei, D. Brooks, and V. J. Reddi. 2022. Automatic Domain-Specific SoC Design for Autonomous Unmanned Aerial Vehicles. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 300–317.

[14] S. Krishnan, Z. Wan, K. Bhardwaj, P. Whatmough, A. Faust, G.-Y. Wei, D. Brooks, and V. J. Reddi. 2020. The sky is not the limit: A visual performance model for cyber-physical co-design in autonomous machines. *IEEE Computer Architecture Letters (CAL)* 19, 1 (2020), 38–42.

[15] Q. Liu, Z. Wan, B. Yu, W. Liu, S. Liu, and A. Raychowdhury. 2022. An energy-efficient and runtime-reconfigurable fpga-based accelerator for robotic localization systems. In *2022 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 01–02.

[16] S. Liu and J.-L. Gaudiot. 2022. Rise of the autonomous machines. *Computer* 55, 1 (2022), 64–73.

[17] S. Liu, Z. Wan, B. Yu, and Y. Wang. 2021. Robotic computing on fpgas. *Synthesis Lectures on Computer Architecture* 16, 1 (2021), 1–218.

[18] S. Liu, Y. Zhu, B. Yu, J.-L. Gaudiot, and G. R. Gao. 2021. Dataflow accelerator architecture for autonomous machine computing. *arXiv preprint arXiv:2109.07047* (2021).

[19] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, et al. 2020. Lgsvl simulator: A high fidelity simulator for autonomous driving. In *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)*. IEEE, 1–6.

[20] S. Sudhakar, V. Sze, and S. Karaman. 2022. Data Centers on Wheels: Emissions From Computing Onboard Autonomous Vehicles. *IEEE Micro* 43, 1 (2022), 29–39.

[21] A. Suleiman, Z. Zhang, L. Carlone, S. Karaman, and V. Sze. 2019. Navion: A 2-mw fully integrated real-time visual-inertial odometry accelerator for autonomous navigation of nano drones. *IEEE Journal of Solid-State Circuits* 54, 4 (2019), 1106–1119.

[22] Z. Wan, A. Anwar, Y.-S. Hsiao, T. Jia, V. J. Reddi, and A. Raychowdhury. 2021. Analyzing and improving fault tolerance of learning-based navigation systems. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 841–846.

[23] Z. Wan, A. Anwar, A. Mahmoud, T. Jia, Y.-S. Hsiao, V. J. Reddi, and A. Raychowdhury. 2022. Frl-fi: Transient fault analysis for federated reinforcement learning-based navigation systems. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 430–435.

[24] Z. Wan, K. Swaminathan, P.-Y. Chen, N. Chandramoorthy, and A. Raychowdhury. 2022. Analyzing and Improving Resilience and Robustness of Autonomous Systems. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 1–9.

[25] Z. Wan, B. Yu, T. Y. Li, J. Tang, Y. Zhu, Y. Wang, A. Raychowdhury, and S. Liu. 2021. A survey of fpga-based robotic computing. *IEEE Circuits and Systems Magazine* 21, 2 (2021), 48–74.

[26] Z. Wan, Y. Zhang, A. Raychowdhury, B. Yu, Y. Zhang, and S. Liu. 2021. An energy-efficient quad-camera visual system for autonomous machines on fpga platform. In *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 1–4.

[27] W. Xiao, N. Mehdipour, A. Collin, A. Y. Bin-Nun, E. Frazzoli, R. D. Tebbens, and C. Belta. 2021. Rule-based optimal control for autonomous driving. In *Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems*. 143–154.

[28] B. Yu, W. Hu, L. Xu, J. Tang, S. Liu, and Y. Zhu. 2020. Building the computing system for autonomous micromobility vehicles: Design constraints and architectural optimizations. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 1067–1081.

[29] J. J. Zhang, K. Liu, F. Khalid, M. A. Hanif, S. Rehman, T. Theocharides, A. Artussi, M. Shafique, and S. Garg. 2019. Building robust machine learning systems: Current progress, research challenges, and opportunities. In *Proceedings of the 56th Annual Design Automation Conference (DAC)*. 1–4.