POLITECNICO DI TORINO
Repository ISTITUZIONALE

Dynamic Workload Selection for Ranked Fault Injection via Sequential Bernoulli Discovery Modeling

Original
Dynamic Workload Selection for Ranked Fault Injection via Sequential Bernoulli Discovery Modeling / Bellarmino, N., Bosio, A., Cantoro, R.. - (In corso di stampa). (IEEE 35th Asian Test Symposium (ATS 2026) Kaohsiung, Taiwan 1st - 3rd December, 2026).

Availability:
This version is available at: 11583/3016156 since: 2026-09-30T11:30:59Z

Publisher:
IEEE

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©9999 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

29 November 2026

Dynamic Workload Selection for Ranked Fault Injection via Sequential Bernoulli Discovery Modeling

Nicolò Bellarmino*, Alberto Bosio†, and Riccardo Cantoro*

*CAD Research Group,
Politecnico di Torino,
Italy
{name.surname}@polito.it

† Centrale Lyon, INSA Lyon, CNRS
UCBL, CPE Lyon, INL,
UMR5270, Ecully, France.
alberto.bosio@ec-lyon.fr

***Abstract*—Fault Injection (FI) is the standard methodology for assessing neural-network resilience to hardware faults, but its cost remains high because each injected fault must be exercised over many input samples. Recent workload-ranking approaches reduce this cost by identifying the smallest set of input samples that maximizes fault detection. Despite their efficiency, these techniques remain static in terms of workload size. However, it has been noticed that the number of unique detected faults decreases as long as we apply input samples, leading to the possibility of dynamically adjusting the workload size. This paper proposes the dynamic adaptation of workloads by modeling the FI campaign as a sequential Bernoulli discovery process. For each processed input sample, a binary event indicates whether at least one previously unseen fault has been detected. Dynamic estimation of such discovery probability is then used as an early stop criterion for the FI campaign. Experiments on five network–dataset pairs, including CIFAR-10, CIFAR-100, MNIST, ImageNet, and a quantized CNN, show that dynamic estimation saves 77.8% of the workload while preserving 100% mean fault coverage.**

***Index Terms*—Fault Injection, Neural Networks, Reliability, Hardware Faults, Test Compaction, Fault Tolerance, Machine Learning, Difficulty Sampling, Rational Sampling**

I. Introduction

Convolutional Neural Networks (CNNs) are increasingly deployed in safety-critical domains such as autonomous driving, industrial automation, and healthcare, where inference reliability is a fundamental requirement [1], [2], [3], [4]. In these scenarios, transient or permanent hardware faults may corrupt the network execution and alter its predictions, potentially leading to unsafe behavior. Fault Injection (FI) is the standard methodology to assess CNN resilience under such conditions: by perturbing the hardware or model state and observing the resulting inference behavior, FI campaigns identify *critical faults*, i.e., faults that cause misclassification or unacceptable output degradation [2]. However, FI remains computationally expensive. A typical campaign involves thousands of injected faults, each exercised over a large input workload, resulting in millions or even hundreds of millions of fault–input evaluations.

Recent works have shown that this cost can be reduced by acting on the input workload rather than only on the fault space [5], [6]. In particular, test samples can be ranked according to difficulty or uncertainty metrics computed from the network output, so that inputs more likely to expose fault-induced vulnerabilities are processed first. Metrics such as Cross Entropy, Gini, EL2N, and Mirrored EL2N have been used to build informative workloads that expose critical faults earlier than random ordering. These approaches show that critical-fault discovery is often concentrated in a small prefix of the ranked workload.

However, ranked workload selection still leaves a key question open: *when should the FI campaign stop?* Existing workload-ranking methods mainly define which samples should be processed first or static evaluate fixed truncation points a posteriori from complete detection curves [6], [7]. They do not provide any criterion to decide dynamically, during the campaign, whether the expected benefit of processing additional samples is still worth the cost. This distinction is important because a practical FI campaign unfolds sequentially: after each processed input, the campaign either discovers new critical faults or it does not. Therefore, ranked FI should not only be treated as a static ordering problem, but also as a dynamic decision problem.

This paper addresses this missing decision layer by modeling ranked FI as a *sequential Bernoulli discovery process*. Given an ordered workload, each processed sample induces a binary event indicating whether at least one previously unseen critical fault has been discovered. Dynamic estimation of this discovery probability are then used to stop the campaign when the residual expected discovery becomes sufficiently small. The proposed formulation turns ranked FI from a static workload-reduction heuristic into an adaptive dynamic campaign with explicit workload–coverage trade-offs.

We instantiate this framework through three representative stop policies. These policies are evaluated on informative ranked workloads generated by Gini and Mirrored EL2N, and

XXX-x-XXXX-XXXX-X/XX/$31.00 ©2026 IEEE

against random ordering as a negative control.

The central empirical finding is that adaptive dynamic stopping is effective when the workload ordering is informative. Ranked workloads produce front-loaded discovery curves, allowing stopping policies to save large fractions of the workload while preserving high critical-fault coverage. Under random ordering, this advantage is substantially reduced: stopping either occurs much later or incurs larger coverage loss. This confirms that online stopping complements workload ranking; it does not replace it.

The main contributions of this work are as follows:

- we formulate ranked CNN fault injection as a sequential Bernoulli discovery process, enabling dynamic reasoning on critical-fault discovery dynamics;
- we instantiate the framework with three representative dynamic stopping policies: an aggressive local policy, a balanced forgetting-based policy, and a cumulative Beta-Bernoulli baseline;
- we show across representative CNN workloads that adaptive stopping is effective on informative ranked orderings but substantially weaker under random ordering, demonstrating that the gain comes from the coupling between ranking quality and dynamic stopping.

II. Background and Related Work

FI is the standard methodology for evaluating the resilience of neural networks under hardware faults. In an FI campaign, faults are injected into the hardware or model representation, and the faulty inference is compared against the fault-free behavior to classify faults as masked, non-critical, or critical [2]. Since each injected fault is typically exercised over a large input workload, FI remains computationally expensive even when the fault space is sampled statistically.

Prior work has reduced FI cost along two main dimensions. The first is the *fault space*: Statistical Fault Injection (SFI) selects representative subsets of faults while preserving confidence and error guarantees [8], [9]. The second is the *input space*: instead of evaluating every injected fault over the full test set, recent studies select or rank input samples that are more likely to expose fault-induced vulnerabilities [5], [6], [10], [11], [12], [13]. This direction is motivated by the input-dependent nature of fault observability: many faults remain masked for most stimuli and become critical only for specific activation patterns.

Only a limited number of works explicitly analyze which input samples are most informative for FI campaigns. Existing approaches typically use difficulty or uncertainty metrics derived from the network output, such as Cross Entropy, DeepGini, EL2N, or Mirrored EL2N, to prioritize samples that are more likely to reveal critical faults [6], [7], [10], [13]. These methods show that ranked workloads can expose most critical faults much earlier than random ordering, turning input selection into an effective workload-reduction strategy.

However, these methods remain essentially *static*. They determine which samples should be processed first, or evaluate fixed truncation points a posteriori from complete detection curves, but they do not decide during the campaign when the residual discovery potential is no longer worth the cost. This is the gap addressed in this work. We keep the ranking layer provided by prior workload-selection methods and add an online decision layer on top of it, modeling ranked FI as a sequential Bernoulli discovery process and stopping adaptively from the observed discovery history.

III. Sequential Bernoulli Modeling for Online FI

A. Sequential Discovery Process

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ denote the workload used during an FI campaign. We assume that samples are processed sequentially according to a given ordering, which may be induced by an informative ranking metric or by random ordering. For a fixed ordering, let C_t be the cumulative number of *distinct critical faults* discovered after processing the first t samples. Since faults are counted only once, C_t is monotone non-decreasing.

We define the incremental number of newly discovered critical faults as

$$Y_t = C_t - C_{t-1}, \tag{1}$$

with $C_0 = 0$. The count process is then converted into a binary discovery event:

$$Z_t = \mathbf{1}(Y_t > 0), \tag{2}$$

where $Z_t = 1$ if sample x_t discovers at least one previously unseen critical fault, and $Z_t = 0$ otherwise. The sequence $\{Z_t\}_{t=1}^{N}$ defines the sequential Bernoulli discovery process associated with the FI campaign.

The objective of dynamic stopping is to estimate, at each step t, whether additional samples are still likely to discover new critical faults. The campaign is stopped when the estimated residual discovery potential becomes sufficiently small.

B. Online Estimation of Discovery Probability

Each discovery event is modeled as

$$Z_t \sim \text{Bernoulli}(p_t), \tag{3}$$

where p_t is the probability that the next processed sample discovers at least one new critical fault. Since ranked FI campaigns are generally front-loaded and non-stationary, p_t is estimated online rather than assumed constant.

We consider three estimators, corresponding to the three policies evaluated in this paper.

Cumulative Beta-Bernoulli. The cumulative baseline starts from a Beta prior and updates its posterior using all past observations:

$$\alpha_t = \alpha_0 + \sum_{i=1}^{t-1} Z_i, \qquad \beta_t = \beta_0 + \sum_{i=1}^{t-1} (1 - Z_i), \tag{4}$$

with posterior mean

$$\hat{p}_t^{\text{mean}} = \frac{\alpha_t}{\alpha_t + \beta_t}. \tag{5}$$

This estimator provides a simple online Bayesian baseline.

Forgetting Beta-Bernoulli. To make the estimator more responsive to changes in the discovery process, we introduce a forgetting factor $\lambda \in (0, 1]$:

$$\alpha_t = \lambda\alpha_{t-1} + Z_{t-1}, \qquad \beta_t = \lambda\beta_{t-1} + (1 - Z_{t-1}). \quad (6)$$

The posterior mean is still computed as in (5), but recent observations receive more weight than older ones. This is useful when fault discovery is concentrated early and the discovery probability decreases along the ranked workload.

Sliding-window estimation. As a local alternative, we estimate the discovery probability over the most recent observations:

$$\hat{p}_t^{\text{mean}} = \frac{1}{|\mathcal{W}_t|} \sum_{i \in \mathcal{W}_t} Z_i, \quad (7)$$

where $\mathcal{W}_t$ contains the indices of the last w observed samples, or fewer if less than w samples have been processed. This estimator reacts quickly to local changes and is used to define an aggressive stopping policy.

For all estimators, the mean estimate is paired with an upper confidence bound:

$$\hat{p}_t^{\text{upper}} = \min\left(1, \hat{p}_t^{\text{mean}} + z_\gamma \hat{\sigma}_t\right), \quad (8)$$

where z_γ is the confidence multiplier and $\hat{\sigma}_t$ is the estimated standard deviation of the current probability estimate. This upper bound is used to make stopping decisions conservative with respect to estimation uncertainty.

C. Stopping Rules and Reported Policies

Let

$$R_t = N - t \quad (9)$$

be the number of samples remaining after step t. We use two stopping criteria.

The first is an instantaneous threshold rule:

$$\hat{p}_t^{\text{upper}} < \delta, \quad (10)$$

which stops when even the upper estimate of the next-step discovery probability is small.

The second is a horizon-aware rule:

$$\hat{p}_t^{\text{upper}} R_t < \varepsilon, \quad (11)$$

which stops when the upper estimate of the expected number of future discovery events falls below a threshold.

Combining the estimators and stopping rules above defines a family of online FI policies. In this work, we use this family to instantiate three representative operating points: a local high-compression policy, a forgetting-based coverage-preserving policy, and a cumulative Bayesian baseline.

Policy parameters are selected on a disjoint development set and then frozen before final evaluation. This prevents a posteriori tuning on the reported case studies and makes the comparison representative of a deployable online FI workflow.

IV. Experimental Setup

A. Evaluation Cases and Fault Model

The proposed dynamic FI framework was evaluated on five representative network–dataset pairs selected to cover different task complexities, architectures, and numerical representations: ResNet18 on CIFAR-10, HCGNet-A2 on CIFAR-100, ConvNeXt-Tiny on ImageNet, ResNet18 on MNIST, and a post-training quantized LeNet5 on MNIST [14], [15], [16], [17]. The evaluation set therefore includes standard image-classification workloads, a large-scale ImageNet case, and a low-precision edge-oriented network.

For MNIST, CIFAR-10, and CIFAR-100, the workload coincides with the full test set of 10,000 images [18], [19]. For ConvNeXt-Tiny, we use a subset of ImageNet workload of 4,000 validation images, which provides a large-scale setting while keeping the FI campaign computationally manageable.

Faults were injected at the bit level into network weights following an SFI methodology [8], [9]. Each injection flips a single randomly selected bit under the single-fault assumption. A fault is labeled as *critical* if it causes at least one misclassification over the considered workload. Distinct critical faults therefore correspond to unique fault sites that become observable under at least one processed sample. For each network, a total of about 16,640 random faults are injected across all layers to estimate the resiliency metrics with an error margin $e = 0.01$ and confidence level of 0.99 ([8], [9]).

B. Workload Orderings

The dynamic stopping framework operates on top of a fixed workload ordering. We consider two informative ranked orderings, Gini and Mirrored EL2N, and one random ordering used as a negative control [5], [6], [20]. Gini and Mirrored EL2N represent difficulty-based rankings that prioritize samples expected to expose fault-induced vulnerabilities early.

For each network–ordering pair, the FI campaign produces a cumulative detection curve $C(t)$, where $C(t)$ is the number of distinct critical faults discovered after processing the first t samples. From this curve, we derive the Bernoulli discovery sequence introduced in Section III and apply the dynamic stopping policies causally, using only observations available up to the current step.

Results on random ordering are the mean over 5 runs. Figure 1 shows the main steps of the framework, from metric computation to dynamic stopping procedure.

C. Policy Development and Final Evaluation

Policy selection was performed only on informative ranked workloads, using Gini and Mirrored EL2N metrics. Random ordering was excluded from policy selection and used only as a negative control in the final evaluation.

The candidate policy grid combines the estimators and stopping rules described in Section III. For the sliding-window estimator, we considered window sizes $w \in \{0.01N, 0.02N\}$, where N is the workload size, and thresholds $\delta \in \{0.05, 0.02, 0.01\}$. For the forgetting Beta-Bernoulli estimator, we considered forgetting factors $\lambda \in \{0.990, 0.995, 0.999\}$

Fig. 1: Diagram showing the steps of the proposed adaptive FI framework

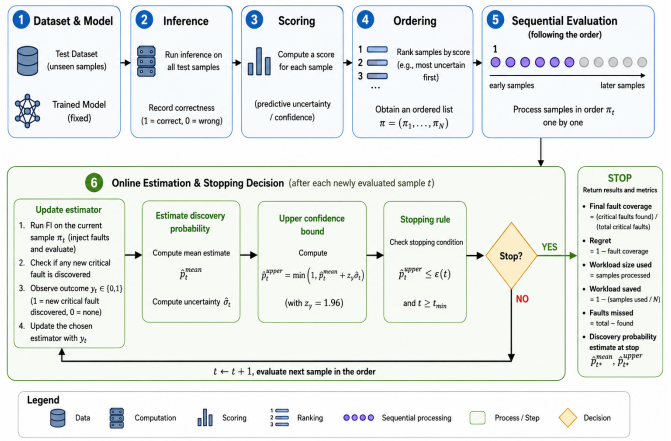


TABLE I: Reported dynamic FI policies and corresponding ranked development-set statistics. The configurations are frozen before final evaluation.

Policy	Selected configuration	Dev.Set mean saved	Dev.Set mean coverage	Dev.Set cases with coverage $\geq 99\%$
Aggressive	SW, 1%N, $\hat{p}_t^{\text{upper}} < 0.05$	97.4%	95.31%	65.0%
Balanced	BBF, $\lambda = 0.990$, $\hat{p}_t^{\text{upper}} R_t < 5.0$	81.6%	99.72%	90.0%
Cumulative	BBCUM, $\hat{p}_t^{\text{upper}} < 0.05$	86.7%	99.53%	90.0%

and horizon thresholds $\varepsilon \in \{5.0, 2.0, 1.0\}$. For the cumulative Beta-Bernoulli baseline, we considered thresholds $\delta \in \{0.05, 0.02, 0.01\}$. All policies used the upper confidence estimate $\hat{p}_t^{\text{upper}}$ with $z = 1.96$, and stopping was disabled before processing at least $\max(20, \lceil 0.005N \rceil)$ samples.

Each candidate configuration was evaluated over the development traces and summarized by mean workload saved, mean coverage at stop, and the fraction of cases reaching at least 99% coverage. The final policies were then selected according to their intended operating role. The *Aggressive* policy was selected among sliding-window candidates as the configuration with the largest mean workload saved subject to mean development coverage of at least 95%. The *Balanced* policy was selected among forgetting Beta-Bernoulli candidates as the configuration with the largest mean workload saved subject to at least 90% of development cases reaching 99% coverage. The *Cumulative* baseline was selected among cumulative Beta-Bernoulli candidates using the same 90% of cases above 99% coverage constraint.

This selection procedure yields the three fixed policies reported in Table I. The selected configurations are then applied unchanged to all five final network–dataset pairs and to all three workload orderings.

D. Evaluation Measures

Stopping quality is evaluated through the number of samples processed before stopping, the fraction of workload saved, the coverage at stop, the regret, and the fraction of cases reaching at least 99% of the final critical-fault coverage. Coverage is computed with respect to the final set of distinct critical faults discovered by the complete campaign under the same ordering. Regret is defined as one minus the coverage at stop.

TABLE II: Policy performance separated by workload ordering group. Aggregates are computed over 3 reported policies, 5 networks, 3 workload orderings, and 15 network–ordering cases.

Ordering	Policy	Cases	Mean saved	Mean coverage	Mean regret	Cases with coverage $\geq 99\%$
Ranked	Aggressive	10	96.1%	93.88%	6.12%	60.0%
	Balanced	10	77.8%	100.00%	0.00%	100.0%
	Cumulative	10	79.4%	98.57%	1.43%	90.0%
Random	Aggressive	5	91.9%	58.11%	41.89%	0.0%
	Balanced	5	19.6%	96.20%	3.80%	60.0%
	Cumulative	5	46.5%	89.09%	10.91%	40.0%
Ranked + Random	Aggressive	15	94.7%	81.96%	18.04%	40.0%
	Balanced	15	58.4%	98.73%	1.27%	86.7%
	Cumulative	15	68.4%	95.41%	4.59%	73.3%

Importantly, the development set contains only CIFAR-10 and MNIST workloads: CIFAR-100 and ImageNet are used exclusively in the final evaluation. This separation makes the final HCGNet/CIFAR-100 and ConvNeXt/ImageNet results a test of whether the selected dynamic stopping policies transfer beyond the datasets used for policy selection.

V. Results

A. Policy Selection on the Development Set

Table I reports the three FI policies selected on the ranked development set. The policies cover three different operating points. *Aggressive* uses local sliding-window evidence and maximizes early stopping, reaching 97.4% mean workload saved on the development set, but with lower coverage robustness. *Balanced* uses forgetting-based Beta-Bernoulli estimation with horizon-aware stopping and provides the intended default trade-off, saving 81.6% of the workload while preserving 99.72% mean coverage. *Cumulative* is the baseline: it uses all past observations with equal weight and therefore tests whether forgetting or local adaptation is actually useful.

These policies are not re-tuned on the final case studies. The purpose of the final evaluation is therefore not to find the best stopping point a posteriori, but to assess whether fixed policies can generalize to unseen networks, datasets, and workload orderings.

B. Ranked Orderings Enable Dynamic Stopping

Table II summarizes the main result by separating informative ranked workloads from random ordering. On ranked workloads, *Balanced* saves 77.8% of the workload while preserving 100.00% mean critical-fault coverage, reaching at least 99% coverage in all cases. This confirms that ranked FI campaigns can be stopped dynamically when the workload ordering concentrates fault discovery early.

The contrast with random ordering is substantial. Under random ordering, the same *Balanced* policy saves only 19.6% of the workload and its mean coverage drops to 96.20%. *Aggressive* still saves most of the workload under random ordering, but mean coverage falls to 58.11%, showing that early stopping without front-loaded discovery is unsafe. *Cumulative* also degrades under random ordering, with 89.09% mean coverage despite saving 46.5% of the workload.

TABLE III: Average performance of 3 reported FI policies for informative ranked workloads, aggregated over 5 networks and 2 ranking metrics.

Metric	Policy	Mean saved	Mean coverage	Mean regret
Gini	Aggressive	95.7%	93.65%	6.35%
	Balanced	81.1%	100.00%	0.00%
	Cumulative	77.5%	100.00%	0.00%
Mirrored EL2N	Aggressive	96.6%	94.12%	5.88%
	Balanced	74.5%	100.00%	0.00%
	Cumulative	81.3%	97.14%	2.86%

This split is central. Dynamic stopping does not create efficiency by itself: it exploits the temporal structure induced by informative rankings. When that structure is absent, the same stopping rules either wait much longer or lose coverage.

It is worth noting that this behavior is not limited to datasets used during policy selection. The development set contains only CIFAR-10 and MNIST workloads, whereas HCGNet/CIFAR-100 and ConvNeXt/ImageNet appear only in the final evaluation, as mentioned in Section IV-D. The fact that the balanced policy still reaches 100.00% coverage on these cases under Gini ordering, while saving 73.7% and 66.0% of the workload respectively, indicates that the selected stopping rule transfers to more complex datasets and architectures.

C. Policy Trade-off on Ranked Workloads

The ranked results also clarify the role of each policy. *Aggressive* defines the high-compression regime. Across the ranked workloads, it saves 96.1% of the workload on average, but mean coverage is 93.88%, and only 60.0% of cases reach at least 99% coverage. This policy is useful to expose the maximum possible reduction, but it is not robust enough as a default FI stopping strategy.

Balanced provides the strongest coverage-preserving operating point. It saves a large fraction of the workload while preserving full mean coverage on ranked orderings. *Cumulative* is competitive in some ranked cases, but its aggregate behavior is less robust than *Balanced*: it saves 79.4% of the workload, but coverage drops to 98.57%, and only 90.0% of ranked cases reach at least 99% coverage. This supports the use of forgetting-based estimation as the default policy under front-loaded, non-stationary discovery dynamics.

D. Metric-Level and Case-Study Results

Table III shows that the conclusion is not tied to a single ranking metric. With Gini ordering, *Balanced* saves 81.1% of the workload with 100.00% mean coverage. With Mirrored EL2N ordering, it saves 74.5% while again preserving 100.00% mean coverage. The *Aggressive* policy saves more than 95% for both ranked metrics, but loses coverage in both cases. The *Cumulative* baseline reaches full mean coverage on Gini, but drops to 97.14% on Mirrored EL2N, confirming that its behavior is less stable across ranked discovery patterns.

Table IV provides the per-network breakdown for the Gini workload. The table shows that *Balanced* reaches 100.00% coverage in all five evaluated networks while saving between 66.0% and 90.8% of the workload. The strongest savings occur on ResNet18/MNIST and ResNet18/CIFAR-10, while the ConvNeXt/ImageNet case remains more demanding. This is expected: larger and more complex workloads can retain a longer discovery tail, making overly aggressive stopping less reliable.

TABLE IV: Case-study results for the Gini workload over 5 evaluated networks and 3 reported policies, grouped by dataset.

Network	Policy	Samples used	Workload saved	Coverage at stop
CIFAR10				
ResNet18	Aggressive	174	98.3%	100.00%
	Balanced	1121	88.8%	100.00%
	Cumulative	1019	89.8%	100.00%
CIFAR100				
HCGNet	Aggressive	916	90.8%	72.93%
	Balanced	2630	73.7%	100.00%
	Cumulative	3663	63.4%	100.00%
ImageNet				
ConvNeXt	Aggressive	242	94.0%	95.43%
	Balanced	1359	66.0%	100.00%
	Cumulative	1294	67.7%	100.00%
MNIST				
ResNet18	Aggressive	106	98.9%	100.00%
	Balanced	925	90.8%	100.00%
	Cumulative	269	97.3%	100.00%
LeNet5 Quantized	Aggressive	372	96.3%	99.89%
	Balanced	1394	86.1%	100.00%
	Cumulative	3058	69.4%	100.00%

The difference between aggressive compression and robust stopping is visible on HCGNet/CIFAR-100. Under Gini ordering, *Aggressive* saves 90.8% of the workload but captures only 72.93% of the final critical-fault set, whereas *Balanced* saves 73.7% and preserves 100.00% coverage. This illustrates why the proposed default policy must account not only for the current discovery probability, but also for the remaining campaign horizon.

E. Detection Curves Explain the Mechanism

Figure 2 explains the mechanism behind the aggregate results. Under informative ranked orderings, the cumulative discovery curve rises sharply at the beginning of the campaign and then approaches saturation. This front-loaded behavior gives dynamic stopping policies a reliable signal: after the early discovery burst, the observed probability of finding new critical faults decreases, and the campaign can be stopped with limited regret.

Under random ordering, discoveries are less concentrated in the early workload prefix. As a result, the same stopping logic becomes less effective. Stopping early risks missing a substantial fraction of critical faults, while preserving coverage requires processing a much larger portion of the workload. This confirms that adaptive stopping and workload ranking are coupled components: ranking determines where critical faults are likely to appear, while the Bernoulli stopping layer determines when further processing is no longer cost-effective.

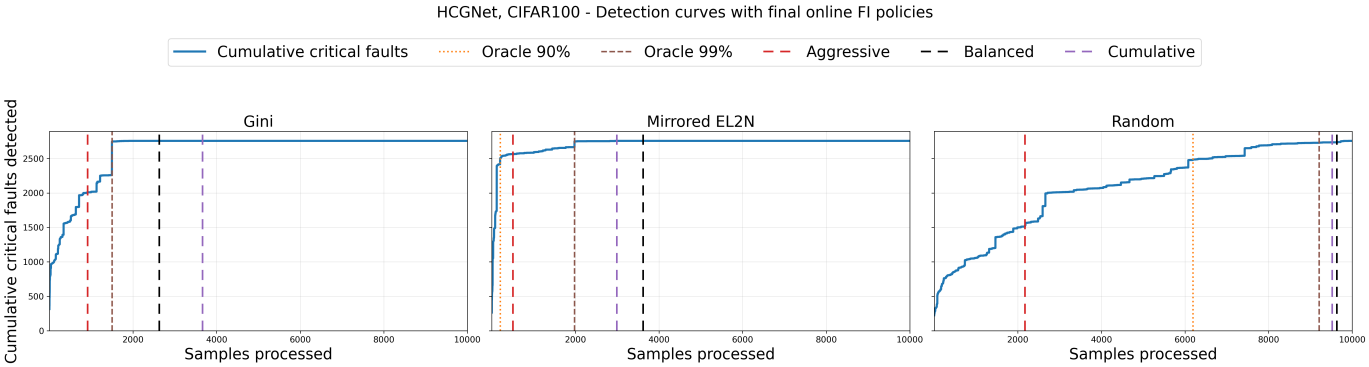


Fig. 2: Detection curves for HCGNet/CIFAR-100 under Gini, Mirrored EL2N, and random workload ordering. Informative rankings concentrate critical-fault discovery early, making dynamic stopping substantially more effective than under random ordering.

VI. Conclusion

This paper presented an adaptive stopping framework for ranked fault injection campaigns. We modeled an FI campaign as a sequential Bernoulli discovery process, where each processed input either discovers at least one previously unseen critical fault or it does not. This formulation enables lightweight dynamic estimation of discovery probability and stopping decisions based only on the campaign history.

The results show that dynamic stopping is effective when the workload ordering is informative. On ranked workloads generated by Gini and Mirrored EL2N, the balanced forgetting-based policy saves 77.8% of the workload while preserving 100.00% mean critical-fault coverage. The aggressive sliding-window policy exposes the high-compression regime, but with lower robustness, while the cumulative Beta-Bernoulli policy provides a useful baseline without forgetting or local adaptation.

The comparison against random ordering confirms the main mechanism behind the approach. Adaptive stopping does not replace ranking: it exploits the front-loaded discovery dynamics induced by informative orderings. When this structure is absent, the stopping advantage is substantially reduced, and early stopping can incur significant coverage loss. Ranked FI and dynamic stopping should therefore be viewed as coupled components of the same adaptive reliability-assessment workflow.

Future work will investigate richer discovery models that account for the number of newly discovered faults, adaptive policies that adjust their aggressiveness online, and broader validation across additional fault models, hardware targets, and compressed neural networks.

Acknowledgement

This work has been funded by the ANR France 2030 AdaptING project “ANR-23-PEIA-0009”.

References

[1] G. D. Natale et al., *Cross-Layer Reliability of Computing Systems*. Jan. 2020.

[2] A. Ruospo et al., “A survey on deep learning resilience assessment methodologies,” *Computer*, 2023.

[3] N. Bellarmino et al., “Covid-19 detection from exhaled breath,” *Scientific Reports*, 2024.

[4] N. Bellarmino et al., “Investigating on Gradient Regularization for Testing Neural Networks,” in *LOD 2024 : 10th International Conference on machine Learning, Optimization and Data science*, Riva de Sole, Italy, Sep. 2024.

[5] N. Bellarmino et al., “Opra: Optimizing resiliency assessment for deep neural networks,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2025.

[6] N. Bellarmino et al., “Dear-cnn: Data-efficient assessment of resiliency in convolutional neural networks,” in *2025 IEEE 28th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, 2025.

[7] S. A. El-Sayed et al., “Compact functional testing for neuromorphic computing circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023.

[8] A. Ruospo et al., “Assessing convolutional neural networks reliability through statistical fault injections,” in *2023 Design, Automation and Test in Europe Conference I& Exhibition (DATE)*, 2023.

[9] R. Leveugle et al., “Statistical fault injection: Quantified error and confidence,” in *2009 Design, Automation & Test in Europe Conference & Exhibition*, 2009.

[10] M. Barbareschi et al., “Investigating the resilience source of classification systems for approximate computing techniques,” *IEEE Transactions on Emerging Topics in Computing*, 2024.

[11] F. Meng et al., “Exploring image selection for self-testing in neural network accelerators,” in *2022 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2022.

[12] F. Meng et al., “A self-test framework for detecting fault-induced accuracy drop in neural network accelerators,” in *2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2021.

[13] N. Bellarmino et al., “M-EL2N: Data-Efficient Fault Injection Through Sample Difficulty Analysis,” in *IEEE 27th Latin American Test Symposium (LATS 2026)*, IEEE, Ed., Florianópolis, Brazil: IEEE, Mar. 2026.

[14] K. He et al., “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[15] C. Yang et al., *Gated convolutional networks with hybrid connectivity for image classification*, https://github.com/winycg/HCGNet, 2020.

[16] Y. LeCun et al., “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, 1998.

[17] R. Leveugle et al., “Edge AI Trustworthiness: Revisiting Bit-Flip Impacts and Criticality Conditions,” *Electronics*, 2025.

[18] Y. LeCun et al., *The mnist database of handwritten digits*, http://yann.lecun.com/exdb/mnist/, Accessed on April 11, 2024, 1998.

[19] A. Krizhevsky, *Cifar-10 dataset*, https://www.cs.toronto.edu/~kriz/cifar.html, Accessed on April 11, 2024, 2009.

[20] Y. Feng et al., *Deepgini: Prioritizing massive tests to enhance the robustness of deep neural networks*, 2020.