When Space Flips a Bit: Recovering Transformer Inference from Radiation-Induced Faults

Anonymous authors
Paper under double-blind review

Abstract

As transformer models move into space CNBC (2025); Booz Allen Hamilton (2024), radiation-induced faults threaten inference reliability by silently corrupting hidden states. Existing defenses offer limited fault coverage, risk suppressing legitimate rare computations, or require hardware modifications that are impractical for commercial off-the-shelf accelerators. We introduce **RIFT** (Recovery via Inference-time Fault Tolerance), a five-stage pipeline that detects and corrects non-repeating transient corruption without retraining, architectural changes, or a clean reference model. RIFT screens hidden states using spectral, temporal, and cross-layer residuals, then re-executes flagged layers to distinguish corruption from legitimate novelty. It recovers corrupted states through uncertainty-aware spectral correction and gated consistency fusion, and adjusts output distributions according to estimated fault severity. Across language, vision-language, and vision-language-action transformers subjected to single-event upsets, multi-bit upsets, single-event transients, and bus corruption, RIFT approaches the task quality of triple modular redundancy. Under single-event upsets, it achieves WikiText-2 perplexity of 6.21 versus 6.15 and LIBERO task success of 76.8% versus 78.1%, with 5.5% clean-run overhead on LLaMA-3-8B and 95.7–97.1% novelty preservation across the three evaluated models. We further evaluate RIFT on a compact decoder running aboard a satellite in low Earth orbit DPhi Space (2026). Across 400 controlled fault injections, RIFT achieves 91.5% fault detection and 91.7% recovered-token agreement with fault-free execution, with runtime overheads of 8.4% during clean execution and 12.6% under injected faults. This orbital experiment demonstrates recovery from controlled faults on hardware operating in space, while recovery from naturally occurring radiation-induced faults remains unverified. Together, these results support inference-time fault recovery that preserves legitimate model behavior across transformer workloads and extends to orbital computing.

1 Introduction

Large transformer models have begun moving from terrestrial datacenters into low Earth orbit (LEO). Starcloud's Starcloud-1 satellite launched with an NVIDIA H100 in November 2025 and subsequently demonstrated Google's Gemma running in space (CNBC, 2025), and Google's Project Suncatcher proposes a larger-scale system of TPU-equipped satellites linked by free-space optics (Google Research, 2025). Such systems expose commercial off-the-shelf (COTS) accelerators to radiation-induced faults whose incidence depends on the device, orbit, and shielding (Barth et al., 2003; Baumann, 2005; Stassinopoulos & Raymond, 1988). We consider four fault regimes: single-event upsets (SEUs) alter stored bits in weights, activations, or KV-cache entries (Binder et al., 1975; Dodd & Massengill, 2003); multi-bit upsets (MBUs) affect multiple cells and can exceed SEC–DED ECC when several affected bits share a codeword (Quinn et al., 2005; Reed et al., 1997); single-event transients (SETs) disturb logic and propagate into computational results (Buchner et al., 1997; Ferlet-Cavrois et al., 2013); and bus or interconnect corruption alters data between computation and consumption, escaping checks confined to the producing operation. These regimes describe error mechanisms or locations rather than mutually exclusive physical phenomena. When such errors es-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

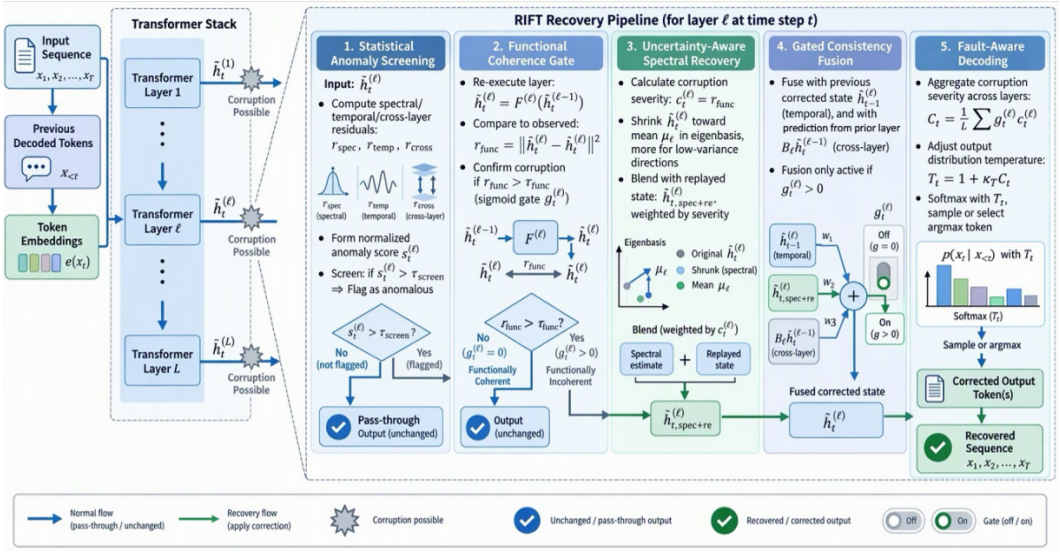


Figure 1: **The RIFT pipeline.** Stages 1–4 run at each layer ℓ and decoding step t: statistical screening flags suspicious hidden states; a functional coherence gate re-executes flagged layers to separate transient corruption from legitimate novelty; uncertainty-aware spectral recovery shrinks confirmed-corrupt states and blends them with the replay; and gated fusion adds temporal and cross-layer structure only when corruption is confirmed. Stage 5 adjusts the output temperature according to aggregate corruption severity. Unflagged and coherent states pass through unchanged.

cape detection, they cause silent data corruption (Dixit et al., 2021; Hochschild et al., 2021) that can alter generated tokens, visual answers, or robot actions. Triple modular redundancy (TMR) masks these errors but roughly triples compute, a cost that power- and mass-constrained orbital payloads can rarely afford.

Cheaper defenses trade coverage or fidelity for cost. Algorithm-based fault tolerance (ABFT) checks specific operations such as attention (Liang et al., 2025; Dai et al., 2025) or relies on custom detection circuits (Xie et al., 2025); activation-range restriction (Sha et al., 2024) is cheap but can suppress legitimate outliers; and other approaches require memory-protection machinery (Zheng et al., 2025) or architectural changes (Wang et al., 2024) (§2). The open question is whether software alone can approach TMR-level task quality at a small fraction of its cost, across language, vision–language, and vision–language–action (VLA) inference, without retraining and without discarding statistically rare but correct computation.

We answer this with **RIFT** (*Recovery via Inference-time Fault Tolerance*), a five-stage pipeline operating on unmodified pretrained weights (1). RIFT first screens hidden states using spectral, temporal, and cross-layer residuals, then re-executes flagged layers and applies a thresholded functional coherence gate. The gate assumes approximately deterministic clean execution and transient faults that occur independently across repeated executions; under these assumptions, disagreement between a layer output and its replay is evidence of corruption that statistical novelty alone does not provide, while reproducible unusual computation passes unchanged. Confirmed-corrupt states undergo severity-dependent spectral shrinkage blended with the replay output, followed by gated temporal and cross-layer fusion, and a final temperature adjustment reflects confirmed corruption while leaving greedy token selection unchanged. Replay is not presumed fault-free, and faults that persist across both executions fall outside the gate's guarantees.

Our contributions are: **(i)** RIFT, a software-only recovery pipeline that approaches TMR task quality at a small fraction of its cost: on LLaMA-3-8B, Qwen2.5-VL-7B, and OpenVLA-7B under SEU, MBU, SET, and bus-corruption injections, it reaches WikiText-2 perplexity of 6.21 versus 6.15 for TMR and LIBERO success of 76.8 % versus 78.1 % under SEU, with 5.5 % clean-run overhead on LLaMA-3-8B compared with about 200 % for TMR; **(ii)** a controlled evaluation on orbital hardware, running RIFT on an NVIDIA Jetson Orin NX aboard a commercial LEO satellite, where it detects

$91.5\,\%$ of 400 injected faults and restores $91.7\,\%$ token agreement with fault-free execution at $8.4\,\%$ clean-run overhead; and **(iii)** a coherence-gated detector that separates statistical screening from replay-based confirmation, evaluated for both fault detection and preservation of legitimate novelty (95.7–$97.1\,\%$ across the three models).

2 Related Work

Resilience of neural networks to hardware faults. Algorithm-based fault tolerance for matrix operations (Huang & Abraham, 1984) introduced the checksum-based detection that underlies most later work. Large-scale resilience studies of DNNs (Reagen et al., 2018; Li et al., 2017) showed that fault sensitivity varies by orders of magnitude across models, layers, and bit positions, motivating alternatives cheaper than redundancy. Recent studies extend this analysis to large language models, characterizing bit-flip vulnerability across components (Chen et al., 2025), end-to-end inference resilience (Sun et al., 2025), and instruction-level GPU soft errors (Chai et al., 2025). Fleet-scale reports of silent data corruption (Dixit et al., 2021; Hochschild et al., 2021) confirm that such errors occur in production hardware even on the ground.

Range restriction. Clipping activations to an admissible interval costs almost nothing at runtime: FT-ClipAct (Hoang et al., 2020) and Ranger (Chen et al., 2021) target CNNs, and Global Clipper (Sha et al., 2024) targets transformer self-attention. Because the admissible range is fixed, these methods also suppress legitimately rare activations, the failure mode RIFT's coherence gate is designed to avoid.

Checksums, co-design, and protection frameworks. Transformer-specific ABFT extends Huang–Abraham checksums to attention, as in ATTNChecker (Liang et al., 2025) and FT-Transformer (Dai et al., 2025), while ApproxABFT (Xue et al., 2023) relaxes exact checks to tolerate benign numerical deviation. ReaLM (Xie et al., 2025) combines statistical ABFT with custom detection circuits. SAVE (Zheng et al., 2025) protects model inference against GPU memory bit flips in software, and NAPER (Rajagede et al., 2025) targets real-time, resource-constrained DNNs. These methods protect the operations and data paths they monitor; RIFT instead monitors hidden states directly and uses replay disagreement to decide when to correct.

Computing in orbit. RedNet (Wang et al., 2024) argues for application-aware radiation tolerance in orbital computing and avoids explicit detection through a multi-exit architectural change; it is the only baseline in our evaluation validated on flight hardware. RIFT requires no architectural change, and we evaluate it with controlled fault injection on a GPU operating in orbit.

3 Method

This section presents RIFT, a software-only recovery pipeline that runs alongside standard transformer inference on unmodified pretrained weights. The key idea is to use disagreement under re-execution as evidence of corruption, assuming approximately deterministic clean computation and independent transient faults across executions. Screening identifies candidate states, and the coherence gate determines whether recovery is applied. States that pass the coherence check are left unchanged. We begin by formalizing the inference setting and the corruption model.

Let f_θ be a transformer with L layers. At autoregressive step t,

$$y_t \sim p_\theta(\cdot \mid x, y_{<t}), \qquad h_t^{(\ell)} = F^{(\ell)}\Big(h_t^{(\ell-1)};\, \theta^{(\ell)}\Big), \qquad z_t = W_{\text{head}}\, h_t^{(L)} + b, \tag{1}$$

where $h_t^{(0)}$ is the token embedding and $F^{(\ell)} \colon \mathbb{R}^d \to \mathbb{R}^d$ denotes the ℓ-th layer map with the decoding context and attention-cache inputs implicit. Under radiation-induced faults (§1), an observed layer output is perturbed as $\tilde{h}_t^{(\ell)} = h_t^{(\ell)} + e_t^{(\ell)}$, where $e_t^{(\ell)}$ represents the resulting hidden-state error. During protected inference, the clean layer output is defined relative to the corrected input $\hat{h}_t^{(\ell-1)}$. The additive notation describes output corruption without implying that all underlying faults are independent or recoverable; corrupted weights or cache entries that persist across executions fall outside the replay guarantee.

The central difficulty is correcting such corruption *without* suppressing the model's legitimate but statistically rare computations. A hidden state deviating from running statistics may signal a hardware fault, or it may signal a surprising token, a sharp semantic shift, or reasoning through an uncommon subspace. Treating every statistical outlier as a fault can therefore discard correct novel inference. RIFT addresses this tension by correcting only states classified as functionally incoherent:

$$\hat{h}_t^{(\ell)} = \mathcal{R}_t^{(\ell)}\left(\tilde{h}_t^{(\ell)},\ \hat{h}_{t-1}^{(\ell)},\ \hat{h}_t^{(\ell-1)},\ F^{(\ell)}\right), \tag{2}$$

where $\mathcal{R}_t^{(\ell)}$ integrates the observed state $\tilde{h}_t^{(\ell)}$, the previous corrected state $\hat{h}_{t-1}^{(\ell)}$, the corrected input from below $\hat{h}_t^{(\ell-1)}$, and the layer function $F^{(\ell)}$ itself, with running statistics implicit. RIFT comprises five stages (1); Stages 1–4 operate at each layer, with replay and recovery applied selectively, and Stage 5 operates after the final layer. Theoretical analysis is in A2, computational cost in A3, and the full algorithm in A10.

Stage 1 (Statistical Anomaly Screening): cheaply flag hidden states whose geometry departs from recently observed behaviour, without committing to any correction. For each layer ℓ, RIFT maintains exponentially weighted running estimates $\mu_\ell \leftarrow (1-\eta_\mu)\mu_\ell + \eta_\mu \hat{h}_{t-1}^{(\ell)}$ and $M_\ell \leftarrow (1-\eta_M)M_\ell + \eta_M \hat{h}_{t-1}^{(\ell)}\hat{h}_{t-1}^{(\ell)\top}$. These expressions describe the statistics available for step t after the preceding corrected state has been incorporated once. With matched averaging rates and consistent initialization, they yield the regularized covariance $\hat{\Sigma}_\ell = M_\ell - \mu_\ell\mu_\ell^\top + \epsilon I$, where $\epsilon > 0$, with eigendecomposition $\hat{\Sigma}_\ell = U_\ell \Lambda_\ell U_\ell^\top$. Here $U_{\ell,k} \in \mathbb{R}^{d\times k}$ collects the top-k eigenvectors. A ridge-regularized cross-layer predictor

$$B_\ell = \arg\min_B \sum_{\tau<t} \omega_{t,\tau} \left\| \hat{h}_\tau^{(\ell)} - B\,\hat{h}_\tau^{(\ell-1)} \right\|_2^2 + \lambda_B \|B\|_F^2, \tag{3}$$

updated online via recursive least squares, estimates the mapping from layer $\ell-1$ to ℓ. RIFT computes three complementary residuals:

$$r_{\mathrm{spec},t}^{(\ell)} = \left\| (I - U_{\ell,k}U_{\ell,k}^\top)\,(\tilde{h}_t^{(\ell)} - \mu_\ell) \right\|_2^2, \tag{4}$$

$$r_{\mathrm{temp},t}^{(\ell)} = \left\| \tilde{h}_t^{(\ell)} - \hat{h}_{t-1}^{(\ell)} \right\|_2^2, \tag{5}$$

$$r_{\mathrm{cross},t}^{(\ell)} = \left\| \tilde{h}_t^{(\ell)} - B_\ell\,\hat{h}_t^{(\ell-1)} \right\|_2^2. \tag{6}$$

The spectral residual equation 4 measures energy outside the leading covariance subspace; the temporal residual equation 5 measures changes from the preceding corrected state; and the cross-layer residual equation 6 measures disagreement with the prediction from the preceding layer. These residuals provide complementary views of corruption, but none uniquely identifies a fault type, and each can also increase during legitimate computation. Each residual is normalized by a positive running robust scale, $\bar{r}_{\bullet,t}^{(\ell)} = r_{\bullet,t}^{(\ell)}/\hat{\sigma}_\ell^\bullet$, estimated using an exponentially weighted median absolute deviation. The normalized residuals form the score $s_t^{(\ell)} = a_{\mathrm{spec}}\bar{r}_{\mathrm{spec},t}^{(\ell)} + a_{\mathrm{temp}}\bar{r}_{\mathrm{temp},t}^{(\ell)} + a_{\mathrm{cross}}\bar{r}_{\mathrm{cross},t}^{(\ell)}$ with fixed weights $a_\bullet \geq 0$. A state is flagged when $s_t^{(\ell)} > \tau_{\mathrm{screen}}$; unflagged states pass through unmodified ($\hat{h}_t^{(\ell)} = \tilde{h}_t^{(\ell)}$). No correction occurs at this stage.

Stage 2 (Functional Coherence Gate): assess corruption through replay disagreement under the stated clean-execution and fault-independence assumptions. Statistical screening alone cannot determine whether a flagged state arose from hardware corruption or from genuinely novel computation. Outside burn-in, the gate re-executes each flagged layer on the same corrected input, with the same decoding context and attention-cache inputs,

$$\check{h}_t^{(\ell)} = F^{(\ell)}\big(\hat{h}_t^{(\ell-1)};\,\theta^{(\ell)}\big), \qquad r_{\mathrm{func},t}^{(\ell)} = \big\|\tilde{h}_t^{(\ell)} - \check{h}_t^{(\ell)}\big\|_2^2. \tag{7}$$

When both executions are clean, their discrepancy is bounded by the numerical reproducibility assumption in A2.1, regardless of statistical novelty. If the original output is corrupted and replay is clean, a sufficiently large perturbation produces an above-threshold discrepancy, as formalized in 1. Persistent or correlated corruption that affects both executions can remain undetected (5, A1). With

$\bar{r}^{(\ell)}_{\mathrm{func},t} = r^{(\ell)}_{\mathrm{func},t}/\hat{\sigma}^{\mathrm{func}}_{\ell}$ and $\hat{\sigma}^{\mathrm{func}}_{\ell} > 0$, the gate is

$$g_t^{(\ell)} = \begin{cases} 0, & \bar{r}^{(\ell)}_{\mathrm{func},t} \le \tau_{\mathrm{func}}, \\ \sigma\Big(\beta\big(\bar{r}^{(\ell)}_{\mathrm{func},t} - \tau_{\mathrm{func}}\big)\Big), & \bar{r}^{(\ell)}_{\mathrm{func},t} > \tau_{\mathrm{func}}. \end{cases} \tag{8}$$

Here σ is the logistic sigmoid, $\tau_{\mathrm{func}} \ge 0$ is the confirmation threshold, and $\beta > 0$ controls sharpness. Exact preservation of clean states requires their normalized replay discrepancies to remain at or below τ_{func}. States below or at the threshold receive $g = 0$ and remain unchanged; above it, $g > 0.5$ and approaches one as the residual increases. The corrected state is then

$$\hat{h}_t^{(\ell)} = (1 - g_t^{(\ell)})\, \tilde{h}_t^{(\ell)} \;+\; g_t^{(\ell)}\, \hat{h}^{(\ell)}_{t,\mathrm{fused}}, \tag{9}$$

where $\hat{h}^{(\ell)}_{t,\mathrm{fused}}$ is produced by Stages 3–4 only when $g_t^{(\ell)} > 0$. The threshold enforces exact pass-through for coherent states, while the sigmoid retains graded recovery weights above the threshold.

Stage 3 (Uncertainty-Aware Spectral Recovery): shrink confirmed-corrupt states toward the running mean, with stronger attenuation along low-variance directions. For states with $g_t^{(\ell)} > 0$, RIFT defines corruption severity from the functional residual, $c_t^{(\ell)} = \bar{r}^{(\ell)}_{\mathrm{func},t}$. This measures disagreement with the layer's re-executed computation rather than statistical unusualness alone. Recovery operates in the eigenbasis of $\hat{\Sigma}_\ell$ with severity-dependent strength $\gamma_t^{(\ell)} = \gamma_{\min} + \kappa_c\, c_t^{(\ell)}$, where $\gamma_{\min} > 0$ and $\kappa_c \ge 0$:

$$\hat{h}^{(\ell)}_{t,\mathrm{spec}} = \mu_\ell + \sum_{i=1}^{d} \frac{\lambda_{\ell,i}}{\lambda_{\ell,i} + \gamma_t^{(\ell)}}\; u_{\ell,i}\, u_{\ell,i}^{\top}\big(\tilde{h}_t^{(\ell)} - \mu_\ell\big). \tag{10}$$

For nonnegative covariance eigenvalues, the shrinkage factor $\lambda_i/(\lambda_i + \gamma)$ attenuates low-variance directions more strongly than high-variance directions. For fixed eigenvalues and $\kappa_c > 0$, higher severity increases attenuation in every positive-variance direction. This provides covariance-based regularization without assuming that SEU/MBU perturbations occur exclusively in low-variance directions; the theoretical analysis in A2.2 applies under its stated noise and signal assumptions. The re-executed state $\check{h}_t^{(\ell)}$ from equation 7 provides an additional anchor:

$$\hat{h}^{(\ell)}_{t,\mathrm{spec+re}} = \Big(1 - \alpha^{(\ell)}_{\mathrm{re},t}\Big)\, \hat{h}^{(\ell)}_{t,\mathrm{spec}} + \alpha^{(\ell)}_{\mathrm{re},t} \check{h}_t^{(\ell)}, \qquad \alpha^{(\ell)}_{\mathrm{re},t} = \frac{\bar{r}^{(\ell)}_{\mathrm{func},t}}{\bar{r}^{(\ell)}_{\mathrm{func},t} + \tau_{\mathrm{re}}}, \tag{11}$$

where $\tau_{\mathrm{re}} > 0$. When the normalized replay residual is small relative to τ_{re}, the spectral estimate dominates; as their ratio increases, the blend places greater weight on replay. Replay is not guaranteed to be clean: the spectral contribution regularizes the blend but does not guarantee recovery when replay is also corrupted, particularly when $\alpha^{(\ell)}_{\mathrm{re},t}$ is close to one.

Stage 4 (Gated Consistency Fusion): incorporate temporal and cross-layer structure only when the coherence gate confirms corruption. Temporal continuity and cross-layer predictability provide additional recovery anchors, but applying them indiscriminately can suppress legitimate semantic transitions or nonlinear transformations. RIFT multiplies both fusion coefficients by $g_t^{(\ell)}$ from equation 8, so that fusion vanishes exactly when $g_t^{(\ell)} = 0$. The temporal coefficient blends the recovered state with the previous corrected state:

$$\alpha^{(\ell)}_{\mathrm{temp},t} = g_t^{(\ell)} \cdot \frac{\bar{r}^{(\ell)}_{\mathrm{temp},t}}{\bar{r}^{(\ell)}_{\mathrm{temp},t} + \tau_{\mathrm{temp}}}, \qquad \hat{h}^{(\ell)}_{t,\mathrm{temp}} = \Big(1 - \alpha^{(\ell)}_{\mathrm{temp},t}\Big)\, \hat{h}^{(\ell)}_{t,\mathrm{spec+re}} + \alpha^{(\ell)}_{\mathrm{temp},t}\, \hat{h}^{(\ell)}_{t-1}. \tag{12}$$

The cross-layer coefficient then blends this estimate with the prediction from the preceding corrected layer:

$$\alpha^{(\ell)}_{\mathrm{cross},t} = g_t^{(\ell)} \cdot \frac{\bar{r}^{(\ell)}_{\mathrm{cross},t}}{\bar{r}^{(\ell)}_{\mathrm{cross},t} + \tau_{\mathrm{cross}}}, \qquad \hat{h}^{(\ell)}_{t,\mathrm{fused}} = \Big(1 - \alpha^{(\ell)}_{\mathrm{cross},t}\Big)\, \hat{h}^{(\ell)}_{t,\mathrm{temp}} + \alpha^{(\ell)}_{\mathrm{cross},t}\, B_\ell\, \hat{h}_t^{(\ell-1)}. \tag{13}$$

Here $\tau_{\mathrm{temp}}, \tau_{\mathrm{cross}} > 0$. The resulting $\hat{h}^{(\ell)}_{t,\mathrm{fused}}$ enters equation 9. Statistical novelty alone cannot activate these fusion steps: they require a positive coherence gate. Preservation of fault-free states

therefore depends on the gate's assumptions and threshold, rather than constituting an unconditional guarantee. After computing $\hat{h}_t^{(\ell)}$, the running statistics (μ_ℓ, M_ℓ) and predictor B_ℓ are updated once using this final state for subsequent steps. Successful recovery limits the propagation of corruption into these estimates, although residual errors can still affect subsequent updates.

Stage 5 (Fault-Aware Decoding): adjust predictive confidence in response to confirmed hidden-state corruption. RIFT aggregates replay-based corruption severity across layers as $C_t = \frac{1}{L}\sum_\ell g_t^{(\ell)}\, c_t^{(\ell)}$, with $g_t^{(\ell)} = c_t^{(\ell)} = 0$ for layers where corruption is not confirmed. The temperature $T_t = 1 + \kappa_T C_t$, with $\kappa_T \geq 0$, softens the distribution as this severity increases:

$$p_t^\star(v) = \frac{\exp(\hat{z}_{t,v}/T_t)}{\sum_{v'}\exp(\hat{z}_{t,v'}/T_t)}, \qquad \hat{y}_t = \arg\max_v\ p_t^\star(v), \tag{14}$$

where $\hat{z}_t = W_{\text{head}}\,\hat{h}_t^{(L)} + b$. For finite logits, increasing temperature reduces distributional concentration but does not guarantee improved likelihood or correctness. Positive temperature scaling preserves the argmax and therefore leaves greedy token selection unchanged; its effects concern likelihood-based evaluation using $p_t^\star$. The severity estimate reflects upstream replay discrepancies and does not detect faults confined to the projection head. When no corruption is confirmed, $C_t = 0$ and $T_t = 1$, recovering the unscaled output distribution.

4 Experiments

We evaluate RIFT across three model categories—language-only, vision-language, and vision-language-action—under four physically motivated fault regimes. The experiments answer four questions: **(Q1)** does RIFT preserve task quality and semantic outputs under transient faults better than existing methods, **(Q2)** can it distinguish hardware corruption from legitimately rare computation, **(Q3)** does it generalise across fault geometries (SEU, MBU, SET, bus corruption), and **(Q4)** what is its computational cost. Extended ablations appear in A5 and sensitivity analyses in A6.

Setup: We evaluate **LLaMA-3-8B** (Grattafiori et al., 2024) on **WikiText-2** (Merity et al., 2016) (perplexity ↓), **Qwen2.5-VL-7B** (Bai et al., 2025) on **MMBench** (Liu et al., 2024) (top-1 accuracy ↑), and **OpenVLA-7B** (Kim et al., 2024) on **LIBERO** (Liu et al., 2023) (success rate ↑, 500 episodes per suite). We compare against implementations of six fault-tolerance methods: **ReaLM** (Xie et al., 2025), **FT-Transformer** (Dai et al., 2025), **SAVE** (Zheng et al., 2025), **ATTNChecker** (Liang et al., 2025), **Global Clipper** (Sha et al., 2024), and **RedNet** (Wang et al., 2024). **No Protection** provides an unprotected reference, while **TMR** (triple modular redundancy, approximately 200 % overhead) provides the redundancy reference. The evaluated adaptations and reimplementations are described in A4.2.

Faults are injected at inference time into hidden states or inter-layer paths. SEUs introduce bit flips in bfloat16 representations at BER values from 10^{-6} to 10^{-3}; MBUs flip contiguous $C{\in}\{2,4,8\}$-bit clusters with $N_{\text{mbu}}{\in}\{1,3,5\}$ events per layer-step; SETs add $\mathcal{N}(0,(\alpha\sigma_\ell)^2)$ noise to a randomly selected 10 % of the dimensions of the $F^{(\ell)}$ output for one step, with $\alpha{\in}\{1,5,10\}$; and bus corruption overwrites a fraction $\rho{\in}\{5,10,25\}$ % of $h_t^{(\ell)}$ on the path to layer $\ell{+}1$. These software-controlled regimes are motivated by radiation-induced error mechanisms. Main-text results are means over three independent runs, with variability given in the captions; full protocol details are in A4.1.

Implementation Details: RIFT runs as a drop-in PyTorch wrapper requiring no changes to weights, architecture, or tokenizer. The 7–8 B trio is evaluated on a single A100-80GB, while the LLaMA-3.2-3B and LLaMA-3-70B scaling experiments (A5.4) run on H100-80GB hardware—one GPU for 3 B and two with tensor parallelism for 70 B. We use gate threshold $\tau_{\text{func}} = 10^{-4}$, gate sharpness $\beta = 50$, and spectral subspace dimension $k = \lfloor d/4 \rfloor$; performance is relatively stable over $\tau_{\text{func}} \in [10^{-5}, 10^{-3}]$, $\beta \in [20, 100]$, and $k/d \in [0.15, 0.5]$. A 128-token warmup prefix initializes the running statistics and calibrates the screening threshold. Full hyperparameters, baseline reimplementations, and the TMR protocol are in A4.2.

4.1 Main Results

Q1: Task quality and semantic preservation. 1 reports task quality across all four fault types at moderate severity (SEU BER= 10^{-4}; MBU $C{=}4, N{=}3$; SET $\alpha{=}5$; bus $\rho{=}10\,\%$). RIFT obtains

Table 1: Task performance under four fault types at moderate severity. **LLaMA-3-8B**: perplexity on WikiText-2 (↓; clean=6.14). **Qwen2.5-VL-7B**: accuracy on MMBench (↑; clean=72.8 %). **OpenVLA-7B**: success rate on LIBERO (↑; clean=78.4 %). Best result excluding TMR in **bold**; TMR is the redundancy reference. Run-to-run variability is discussed in A6.1; per-suite LIBERO results appear in A5.8, and output-level semantic preservation in A5.9. Means over three independent runs; STD ≤ 0.5 in the displayed units.

	LLaMA-3-8B (PPL ↓)				**Qwen2.5-VL-7B** (Acc. ↑)				**OpenVLA-7B** (SR ↑)			
Method	SEU	MBU	SET	Bus	SEU	MBU	SET	Bus	SEU	MBU	SET	Bus
No Protection	24.7	41.3	18.9	35.6	38.2	24.6	45.1	29.7	31.2	18.5	40.3	22.8
ReaLM	6.43	12.8	8.71	14.2	61.4	49.3	58.7	47.1	58.6	42.1	54.9	39.8
FT-Trans.	6.51	10.6	7.24	13.8	60.8	51.7	59.2	46.3	57.3	44.6	56.2	38.5
SAVE	6.38	11.4	9.53	12.1	62.1	48.5	52.4	50.6	60.2	40.8	48.7	43.2
ATTNChecker	7.12	14.3	7.96	16.9	58.3	44.2	57.3	41.9	55.1	38.7	53.6	35.1
Gl. Clipper	7.84	9.17	10.2	11.6	59.7	53.8	48.1	52.1	56.8	46.2	44.9	44.7
RedNet	8.93	13.1	11.7	15.4	55.2	46.1	50.8	43.4	53.4	41.3	49.2	40.6
TMR	6.15	6.16	6.15	6.14	72.6	72.5	72.7	72.6	78.1	77.9	78.2	78.0
RIFT (ours)	**6.21**	**6.38**	**6.24**	**6.31**	**71.9**	**70.4**	**72.1**	**70.8**	**76.8**	**74.1**	**77.2**	**74.9**

the best result among the evaluated methods excluding TMR in all twelve model×fault cells, with a maximum relative deviation from the clean baseline of approximately $\mathbf{5.5}\,\%$. The evaluated baselines exhibit different sensitivities across fault types: ReaLM and SAVE incur larger degradation under MBU and bus corruption than under SEU, while Global Clipper performs comparatively well under MBU but less well under SET. On the VLA task, RIFT substantially improves success rates across all four fault types; the per-suite LIBERO results in A5.8 further examine performance across manipulation tasks. Beyond aggregate task quality, paired clean and faulted runs assess output preservation. RIFT achieves BLEU-4 $= 0.964$ on LLaMA-3-8B, top-1 answer match $= 95.8\,\%$ on Qwen2.5-VL-7B, and step-wise action match $= 94.1\,\%$ on OpenVLA-7B. The corresponding gaps to TMR are 0.025 BLEU, 3.6 percentage points, and 4.6 percentage points, respectively (A5.9). Recovery also extends across the LLaMA-3 family from 3 B to 70 B parameters, with RIFT/TMR perplexity ratios of $1.009/1.010/1.020$ at the three scales (A5.4).

Q2: Novelty preservation. 2 reports two complementary metrics. Fault detection rate (FDR) measures the fraction of injected SEU faults detected at BER$= 10^{-4}$, while novelty preservation rate (NPR $=1-\mathrm{FPR}$) is evaluated separately on fault-free, distributionally unusual sequences and measures the fraction with no false detection. For RIFT, a false positive occurs when the coherence gate satisfies $g > 0$ at any layer-step; statistical screening or re-execution alone does not count (A4.3).

In this three-model evaluation, RIFT achieves FDR$\geq 95.9\,\%$ and NPR$\geq 95.7\,\%$. Its gate uses replay consistency to distinguish reproducible unusual computation from transient discrepancies exceeding the confirmation threshold. Global Clipper preserves 68.5–76.4 % of the novelty sequences, indicating a larger false-positive rate on this benchmark. The approximately 100 % NPR entries for ReaLM and ATTNChecker are construction-based expectations for the arithmetic-invariant detector variants, rather than measured novelty results. Such expectations require fault-free computation to satisfy their checks and do not establish coverage of every fault type. NPR is unavailable for FT-Transformer and SAVE, so no novelty-preservation ranking against these methods is claimed.

Q3: Generalisation across fault geometries. 3 sweeps fault severity on LLaMA-3-8B. The perplexity gap between RIFT and the strongest baseline at each fault type and severity widens with severity. RIFT's spectral shrinkage equation 10 and replay blend equation 11 vary with the functional residual: weaker discrepancies favor spectral recovery, while stronger discrepancies increase the weight assigned to the replayed state. Outside burn-in, replay is performed for every state flagged by screening, regardless of whether recovery is subsequently applied.

At high MBU severity, the best baseline (Global Clipper) reaches 34.7 perplexity, compared with RIFT's 7.14, a $4.9\times$ perplexity ratio. Under high-severity SET, RIFT achieves 6.52, compared with 14.8 for the best baseline. The continuous BER stress test in A6.3 places RIFT's collapse threshold at approximately 1.05×10^{-2}, about $13\times$ ReaLM's threshold of 8×10^{-4}, where collapse

Table 2: Fault detection under SEU (BER= 10^{-4}) and novelty preservation under fault-free execution. FDR: fault detection rate (↑). NPR: novelty preservation rate $= 1 - \mathrm{FPR}$ (↑). Both metrics are percentages. Novelty is evaluated on 500 deliberately unusual sequences per model (A4.3). †: construction-based expectation for arithmetic-invariant detector variants, assuming fault-free computation satisfies their checks; not a measured NPR result. *n/r*: NPR unavailable; no value is imputed. Measured entries are means over three independent runs; $\mathrm{STD} \leq 0.3$ percentage points.

	LLaMA-3-8B		**Qwen2.5-VL-7B**		**OpenVLA-7B**	
Method	FDR	NPR	FDR	NPR	FDR	NPR
ReaLM	94.2	$\approx 100^{\dagger}$	91.8	$\approx 100^{\dagger}$	90.5	$\approx 100^{\dagger}$
FT-Trans.	96.7	*n/r*	94.1	*n/r*	93.8	*n/r*
SAVE	97.1	*n/r*	95.3	*n/r*	94.6	*n/r*
ATTNChecker	91.5	$\approx 100^{\dagger}$	88.9	$\approx 100^{\dagger}$	87.2	$\approx 100^{\dagger}$
Gl. Clipper	88.3	76.4	85.7	71.2	83.1	68.5
RedNet	82.6	81.3	79.4	78.1	77.8	75.9
RIFT (ours)	**98.4**	**97.1**	**96.8**	**96.3**	**95.9**	**95.7**

Table 3: RIFT performance on LLaMA-3-8B (WikiText-2 perplexity ↓) across fault types at low, moderate, and high severity. Clean baseline: 6.14. "Best baseline" is selected per fault type and severity, excluding TMR. BER stress test in A6.3; simultaneous-fault test in A5.3; sustained-burst test in A6.4. Means over three independent runs; $\mathrm{STD} \leq 0.06$.

Severity	**SEU**	**MBU**	**SET**	**Bus**
Low (SEU: BER= 10^{-6}; MBU: C=2, N=1; SET: α=1; Bus: ρ=5%)				
No Protection	6.31	6.58	6.27	6.72
Best baseline	6.17	6.24	6.18	6.29
RIFT	**6.15**	**6.16**	**6.15**	**6.17**
Moderate (SEU: BER= 10^{-4}; MBU: C=4, N=3; SET: α=5; Bus: ρ=10%)				
No Protection	24.7	41.3	18.9	35.6
Best baseline	6.38	9.17	7.24	11.6
RIFT	**6.21**	**6.38**	**6.24**	**6.31**
High (SEU: BER= 10^{-3}; MBU: C=8, N=5; SET: α=10; Bus: ρ=25%)				
No Protection	187.4	493.2	82.6	312.7
Best baseline	11.3	34.7	14.8	41.2
RIFT	**6.47**	**7.14**	**6.52**	**6.93**

is defined as exceeding twice clean perplexity. Under simultaneous SEU+SET+bus faults (A5.3), RIFT's perplexity rises from 6.21 to 6.83 (about 10 % relative to SEU alone), compared with 503 % for the strongest baseline. The sustained-burst test (A6.4) further shows RIFT remaining below twice clean perplexity through the full tested range of 100 consecutive faulted decoding steps.

Q4: Computational overhead. On LLaMA-3-8B, RIFT incurs 5.5 % wall-clock overhead on clean inputs, rising to 8.2 % under SEU faults at BER= 10^{-4}. Screening and statistics updates run at every layer-step, while replay is restricted to states flagged by screening outside burn-in. RIFT's clean-run overhead is below the reported clean-run overheads of SAVE (8.8 %), FT-Transformer (13.9 %), and TMR (201 %) in this comparison. The per-method timing breakdown and further overhead analyses are provided in A5.6. Appendix A9 additionally evaluates recovery on larger language-only and vision-language models under the same four transient fault regimes.

4.2 Ablations and Sensitivity Tests

Component ablation on LLaMA-3-8B under MBU identifies the contributions of individual stages. Without the coherence gate (Stage 2), novelty preservation is 71.4 % in the ablation evaluation, supporting the importance of replay-based confirmation. Removing spectral recovery (Stage 3) raises perplexity from 6.38 to 7.93, the largest increase among individual component ablations;

Table 4: Controlled fault recovery on hardware operating in low Earth orbit. Token agreement is measured against fault-free execution. Each fault type includes 100 injected events. All values are percentages. Means over three independent runs; STD ≤ 0.4.

Metric	SEU	MBU	SET	Bus
Agreement, unprotected	68.7	52.4	70.9	57.8
Agreement, RIFT	**93.4**	**90.1**	**92.8**	**90.6**
Fault detection, RIFT	93.0	91.0	92.0	90.0

removing all recovery stages raises it to 11.6. Cross-modal injection on Qwen2.5-VL-7B yields a vision-only versus language-only accuracy gap of 0.7 percentage points for RIFT, compared with 3.6 for SAVE and 3.2 for Global Clipper. Across LLaMA-3-family models from 3 B to 70 B, clean-run wall-clock overhead stays between $5.4\,\%$ and $5.7\,\%$. Full ablation and sensitivity results appear in A5 and A6.

4.3 Controlled On-Orbit Validation

We deployed RIFT on a commercial on-orbit compute platform in LEO (provider and platform names omitted for anonymous review), using a Jetson Orin NX 16 GB node and a six-layer decoder with 0.31M int8 parameters and FP16 activations. Across 64 prompts and 2,048 generated tokens, we performed 400 controlled fault injections, equally divided among SEU, MBU, SET, and bus corruption. RIFT achieved 91.7% aggregate recovered-token agreement, 91.5% fault detection, and 93.8% sequence-level novelty preservation, with 8.4% clean and 12.6% faulted runtime overhead. Agreement improved by 21.9–37.7 percentage points across fault types (4). Appendix A7 gives the full orbital protocol and telemetry, together with additional experiments on fault placement, structured perturbations, replay persistence, domain shift, and long-context stability.

5 Limitations

RIFT's guarantees hold only under its stated assumptions: approximately deterministic clean execution and transient faults that occur independently across the original and replayed computation. Independence does not follow from transience alone. Corrupted weights, inputs, or KV-cache entries that persist across both executions can go undetected, and a corrupted replay can disagree with a clean output (A1). Cumulative total-ionizing-dose effects (Oldham & McLean, 2003; Schwank et al., 2008), single-event latchup, and destructive effects such as gate rupture and burnout (Johnston, 1996; Sexton, 2003) require hardware protection beyond hidden-state replay. All faults in our evaluation, including those on orbit, were injected in software at the tensor level; we have not validated the mapping from particle strikes to these error models through beam testing or naturally occurring radiation events, and the on-orbit experiment used a compact model and limited workload (A4.4).

6 Conclusion

RIFT combines replay-based corruption detection with uncertainty-aware spectral shrinkage and coherence-gated fusion. Across LLaMA-3-8B, Qwen2.5-VL-7B, and OpenVLA-7B under four injected fault types, it approaches TMR task quality with $5.5\,\%$ clean-run overhead on LLaMA-3-8B, versus about $200\,\%$ for TMR, and without retraining, architectural changes, or a clean reference model. Controlled injections on a Jetson Orin NX in low Earth orbit show that the same recovery runs on orbital hardware. Testing RIFT against naturally occurring radiation strikes is the main next step.

Ethics Statement

This work studies inference-time fault detection and recovery in transformer models, with the goal of improving reliability under transient corruption. The evaluation uses publicly available models and

benchmarks, including WikiText-2, MMBench, LIBERO, and LongBench-E, and does not involve human-subject experiments. The robotic manipulation experiments are conducted in simulation. All evaluated faults, including those in the low Earth orbit experiment, are injected through software; the orbital results demonstrate controlled fault recovery on operating orbital hardware, rather than validation against naturally occurring radiation strikes. RIFT is intended as a complementary software safeguard and does not replace hardware protection or safety qualification for deployment in critical systems.

REPRODUCIBILITY STATEMENT

RIFT operates on pretrained models without retraining or architectural modification. Section 3 describes the five-stage pipeline, and Appendix A10 provides the inference procedure in pseudocode. Appendices A4.1 and A4.2 describe the models, benchmarks, fault-injection settings, hyperparameters, calibration procedure, and baseline implementations. Appendix A4.3 specifies the novelty-preservation benchmark, while Appendices A5 and A6 report component ablations, additional evaluations, and sensitivity analyses. Appendix A2 states the assumptions and proofs underlying the theoretical results, and Appendix A3 analyzes computational and memory costs. Appendix A7 describes the controlled on-orbit experiment and additional robustness studies. We will publicly release our code, configuration files, and evaluation scripts upon acceptance.

THE USE OF LARGE LANGUAGE MODELS

Large language models were used as general-purpose assistants to polish the writing, support coding, including debugging and refining experiment and plotting scripts, and assist with literature search. All research ideas, methods, analyses, and conclusions are the authors' own. The authors reviewed and verified all LLM-assisted text, code, and retrieved references and take full responsibility for the content of this paper. The language, vision-language, and vision-language-action models evaluated as research subjects are described in Section 4 and the corresponding appendices.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Janet L Barth, CS Dyer, and EG Stassinopoulos. Space, atmospheric, and terrestrial radiation environments. *IEEE Transactions on nuclear science*, 50(3):466–482, 2003.

Robert C Baumann. Radiation-induced soft errors in advanced semiconductor technologies. *IEEE Transactions on Device and materials reliability*, 5(3):305–316, 2005.

D Binder, E C Smith, and A B Holman. Satellite anomalies from galactic cosmic rays. *IEEE Trans. Nucl. Sci., v. NS-22, no. 6, pp. 2675-2680*, 12 1975. ISSN ISSN IETNA. doi: 10.1109/TNS.1975.4328188. URL https://www.osti.gov/biblio/4094577.

Booz Allen Hamilton. Large language models in space and beyond. Booz Allen Insights, 2024. URL https://www.boozallen.com/insights/space/large-language-models-in-space-and-beyond.html.

S. Buchner, M. Baze, D. Brown, D. McMorrow, and J. Melinger. Comparison of error rates in combinational and sequential logic. *IEEE Transactions on Nuclear Science*, 44(6):2209–2216, 1997. doi: 10.1109/23.659037.

Duo Chai, Zizhen Liu, Shuhuai Wang, Songwei Pei, Cheng Liu, Huawei Li, and Shangguang Wang. Analysis of llm vulnerability to gpu soft errors: An instruction-level fault injection study, 2025. URL https://arxiv.org/abs/2601.19912.

Yuhang Chen, Zhen Tan, Ajay Kumar Jaiswal, Huaizhi Qu, Xinyu Zhao, Qi Lin, Yu Cheng, Andrew Kwong, Zhichao Cao, and Tianlong Chen. Bit-flip error resilience in LLMs: A comprehensive analysis and defense framework. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10414–10424, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.528. URL https://aclanthology.org/2025.emnlp-main.528/.

Zitao Chen, Guanpeng Li, and Karthik Pattabiraman. A low-cost fault corrector for deep neural networks through range restriction. In *2021 51st Annual IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN)*, pp. 1–13. IEEE, 2021. URL https://arxiv.org/abs/2003.13874.

Suhwan Choi, Yunsung Lee, Yubeen Park, Chris Dongjoo Kim, Ranjay Krishna, Dieter Fox, and Youngjae Yu. vla-eval: A unified evaluation harness for vision-language-action models, 2026. URL https://arxiv.org/abs/2603.13966.

CNBC. Nvidia-backed starcloud trains first AI model in space, plans orbital data centers. CNBC, December 2025. URL https://www.cnbc.com/2025/12/10/nvidia-backed-starcloud-trains-first-ai-model-in-space-orbital-data-centers.html. Accessed: 2026-04-27.

Huangliang Dai, Shixun Wu, Jiajun Huang, Zizhe Jian, Yue Zhu, Haiyang Hu, and Zizhong Chen. Ft-transformer: Resilient and reliable transformer with end-to-end fault tolerant attention, 2025. URL https://arxiv.org/abs/2504.02211.

Harish Dattatraya Dixit, Sneha Pendharkar, Matt Beadon, Chris Mason, Tejasvi Chakravarthy, Bharath Muthiah, and Sriram Sankar. Silent data corruptions at scale, 2021. URL https://arxiv.org/abs/2102.11245.

Paul E Dodd and Lloyd W Massengill. Basic mechanisms and modeling of single-event upset in digital microelectronics. *IEEE Transactions on nuclear Science*, 50(3):583–602, 2003.

DPhi Space. A satellite just described Earth with an LLM: DPhi Space opens Clustergate-2, an on-orbit compute platform for developers. DPhi Space News, April 2026. URL https://www.dphispace.com/news/clustergate-2-llm-on-orbit. Published April 29, 2026. Accessed September 11, 2026.

Véronique Ferlet-Cavrois, Lloyd W. Massengill, and Pascale Gouker. Single event transients in digital cmos—a review. *IEEE Transactions on Nuclear Science*, 60:1767–1790, 2013. URL https://api.semanticscholar.org/CorpusID:38254057.

Google Research. Exploring a space-based, scalable AI infrastructure system design. Google Research Blog, November 2025. URL https://research.google/blog/exploring-a-space-based-scalable-ai-infrastructure-system-design/. Project Suncatcher; companion preprint: "Towards a future space-based, highly scalable AI infrastructure system design".

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Le-Ha Hoang, Muhammad Hanif, and Muhammad Shafique. Ft-clipact: Resilience analysis of deep neural networks and improving their fault tolerance using clipped activation. pp. 1241–1246, 03 2020. doi: 10.23919/DATE48585.2020.9116571.

Peter H. Hochschild, Paul Turner, Jeffrey C. Mogul, Rama Govindaraju, Parthasarathy Ranganathan, David E. Culler, and Amin Vahdat. Cores that don't count. In *Proceedings of the Workshop on Hot Topics in Operating Systems (HotOS '21)*, pp. 9–16. ACM, 2021. doi: 10.1145/3458336.3465297.

Kuang-Hua Huang and Jacob A. Abraham. Algorithm-based fault tolerance for matrix operations. *IEEE Transactions on Computers*, C-33(6):518–528, June 1984. doi: 10.1109/TC.1984.1676475.

AH Johnston. The influence of vlsi technology evolution on radiation-induced latchup in space systems. *IEEE Transactions on Nuclear Science*, 43(2):505–521, 1996.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

Guanpeng Li, Siva Kumar Sastry Hari, Michael Sullivan, Timothy Tsai, Karthik Pattabiraman, Joel Emer, and Stephen W. Keckler. Understanding error propagation in deep learning neural network (DNN) accelerators and applications. In *Proc. Int. Conf. for High Performance Computing, Networking, Storage and Analysis (SC '17)*. ACM, 2017. doi: 10.1145/3126908.3126964.

Yuhang Liang, Xinyi Li, Jie Ren, Ang Li, Bo Fang, and Jieyang Chen. Attnchecker: Highly-optimized fault tolerant attention for large language model training, 2025. URL https://arxiv.org/abs/2410.11720.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023. URL https://arxiv.org/abs/2306.03310.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player? In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. URL https://arxiv.org/abs/2307.06281.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016. URL https://arxiv.org/abs/1609.07843.

Timothy R Oldham and FB McLean. Total ionizing dose effects in mos oxides and devices. *IEEE transactions on nuclear science*, 50(3):483–499, 2003.

H. Quinn, P. Graham, J. Krone, M. Caffrey, and S. Rezgui. Radiation-induced multi-bit upsets in sram-based fpgas. *IEEE Transactions on Nuclear Science*, 52(6):2455–2461, 2005. doi: 10.1109/TNS.2005.860742.

Rian Adam Rajagede, Muhammad Husni Santriaji, Muhammad Arya Fikriansyah, Hilal Hudan Nuha, Yanjie Fu, and Yan Solihin. Naper: Fault protection for real-time resource-constrained deep neural networks. In *2025 IEEE 31st International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pp. 1–8. IEEE, 2025.

Brandon Reagen, Udit Gupta, Lillian Pentecost, Paul Whatmough, Sae Kyu Lee, Niamh Mulholland, David Brooks, and Gu-Yeon Wei. Ares: A framework for quantifying the resilience of deep neural networks. In *Proc. 55th Annual Design Automation Conference (DAC '18)*, pp. 17:1–17:6. ACM, 2018. doi: 10.1145/3195970.3195997.

RA Reed, MA Carts, PW Marshall, CJ Marshall, O Musseau, PJ McNulty, DR Roth, S Buchner, J Melinger, and T Corbiere. Heavy ion and proton-induced single event multiple upset. *IEEE Transactions on Nuclear Science*, 44(6):2224–2229, 1997.

James R Schwank, Marty R Shaneyfelt, Daniel M Fleetwood, James A Felix, Paul E Dodd, Philippe Paillet, and Véronique Ferlet-Cavrois. Radiation effects in mos oxides. *IEEE Transactions on Nuclear Science*, 55(4):1833–1853, 2008.

F. W. Sexton. Destructive single-event effects in semiconductor devices and ICs. *IEEE Transactions on Nuclear Science*, 50(3):603–621, 2003. doi: 10.1109/TNS.2003.813137.

Qutub Syed Sha, Michael Paulitsch, Karthik Pattabiraman, Korbinian Hagn, Fabian Oboril, Cornelius Buerkle, Kay-Ulrich Scholl, Gereon Hinz, and Alois Knoll. Global clipper: Enhancing safety and reliability of transformer-based object detection models, 2024. URL https://arxiv.org/abs/2406.03229.

E G Stassinopoulos and J P Raymond. The space radiation environment for electronics. *Proc. IEEE; (United States)*, 76:11, 11 1988. ISSN ISSN IEEPA. doi: 10.1109/5.90113. URL https://www.osti.gov/biblio/6042035.

Yu Sun, Zachary Coalson, Shiyang Chen, Hang Liu, Zhao Zhang, Sanghyun Hong, Bo Fang, and Lishan Yang. Demystifying the resilience of large language model inference: An end-to-end perspective. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1127–1144, 2025.

Meiqi Wang, Han Qiu, Longnv Xu, Di Wang, Yuanjie Li, Tianwei Zhang, Jun Liu, and Hewu Li. A case for application-aware space radiation tolerance in orbital computing, 2024. URL https://arxiv.org/abs/2407.11853.

Tong Xie, Jiawang Zhao, Zishen Wan, Zuodong Zhang, Yuan Wang, Runsheng Wang, Ru Huang, and Meng Li. Realm: Reliable and efficient large language model inference with statistical algorithm-based fault tolerance, 2025. URL https://arxiv.org/abs/2503.24053.

Xinghua Xue, Cheng Liu, Feng Min, Tao Luo, and Yinhe Han. Approxabft: approximate algorithm-based fault tolerance for neural network processing. *arXiv preprint arXiv:2302.10469*, 2023.

Wenxin Zheng, Bin Xu, Jinyu Gu, and Haibo Chen. {SAVE}:{Software-Implemented} fault tolerance for model inference against {GPU} memory bit flips. In *2025 USENIX Annual Technical Conference (USENIX ATC 25)*, pp. 1585–1604, 2025.

APPENDIX

CONTENTS

1 Introduction **1**

2 Related Work **3**

3 Method **3**

4 Experiments **6**

4.1 Main Results 6

4.2 Ablations and Sensitivity Tests 8

4.3 Controlled On-Orbit Validation 9

5 Limitations **9**

6 Conclusion **9**

Appendix **14**

A1 Scope and Limitations of the Coherence Gate **16**

A2 Theoretical Analysis **16**

A2.1 Coherence gate correctness 16

A2.2 Minimax spectral recovery 17

A2.3 Shrinkage monotonicity 17

A2.4 Fusion stability 17

A3 Computational Cost **18**

A4 Detailed Setup and Implementation **18**

A4.1 Detailed setup 18

A4.2 Detailed implementation 19

A4.3 Construction of the novelty benchmark 19

A4.4 Limitation: beam-test and on-orbit validation 20

A5 Ablation Studies **20**

A5.1 Component ablation 20

A5.2 Cross-modal fault propagation 21

A5.3 Compound fault conditions 21

A5.4 Scaling within the LLaMA-3 family and cross-architecture transfer 21

A5.5 Memory overhead 22

A5.6 Detailed computational overhead breakdown 23

A5.7 Cross-modal wall-clock overhead 24

A5.8 Per-suite LIBERO breakdown 24

A5.9 Semantic preservation under fault recovery 25

A6 Sensitivity Analysis **25**

A6.1 Statistical reliability 26

A6.2 Threshold and hyperparameter sensitivity 26

A6.3 Failure-mode analysis: graceful degradation and collapse threshold 27

A6.4 Stress test: sustained fault bursts 27

A6.5 Replay frequency and computational overhead 28

A7 On-Orbit Deployment and Additional Robustness Experiments **29**

A7.1 Controlled Fault Recovery in Low Earth Orbit 29

A7.2 Sensitivity to Fault Placement 30

A7.3 Structured Perturbations and Replay Persistence 30

A7.4 Adaptation under Sustained Domain Shift 31

A7.5 Long-Context Statistical Stability 31

A8 Qualitative Case Study under Injected Transient Faults **32**

A9 Additional Cross-Model Generalization **32**

A10 Algorithm **33**

A1 Scope and Limitations of the Coherence Gate

The functional coherence gate targets transient corruption under assumptions of approximately deterministic clean execution and independence of faults across executions. Independence is an assumption of the analysis, not a defining property of transient faults.

Deterministic clean execution. Repeated clean evaluations of $F^{(\ell)}$ on the same input must differ in Euclidean norm by at most δ_{fp}. For a positive normalization scale $\hat{\sigma}^{\mathrm{func}}$, fault-free states pass through unchanged when $\delta_{\mathrm{fp}}^2/\hat{\sigma}^{\mathrm{func}} \leq \tau_{\mathrm{func}}$. Disabling stochastic components supports reproducibility, but floating-point discrepancies depend on the implementation and must satisfy this normalized threshold condition.

Replay reliability. Detection of corruption in the original output is guaranteed only when replay is clean and the discrepancy is sufficiently large, as specified in 1. Under independent faults with per-execution probability p_{fault}, an individual replay is corrupted with probability p_{fault}, while both executions in a fixed pair are corrupted with probability p_{fault}^2. The latter is an unconditional probability, not the probability of replay corruption given that screening triggered. A clean original execution followed by a corrupted replay can also produce a positive gate; disagreement alone does not identify which execution is faulty.

Persistent and correlated faults. Persistent faults, such as stuck bits in weight memory, can corrupt both executions identically and produce a residual below the confirmation threshold. Such faults are not reliably detected by replay disagreement and require additional protection or recovery mechanisms. Correlated transient faults likewise fall outside the independence guarantee. Residual corruption can enter the running statistics $(\mu_\ell, \hat{\Sigma}_\ell)$ through the states used for their updates, affecting subsequent screening and recovery. RIFT therefore provides conditional guarantees for the stated transient-fault model, rather than coverage of all sources of silent data corruption.

A2 Theoretical Analysis

We provide theoretical support for the coherence gate, spectral recovery, and fusion stability under explicit assumptions.

A2.1 Coherence gate correctness

[Deterministic clean execution] At inference time, repeated clean evaluations of $F^{(\ell)}$ on the same input differ in Euclidean norm by at most $\delta_{\mathrm{fp}} \geq 0$.

[Independent transient faults] Transient faults at a given layer and step are independent across executions, each occurring with probability p_{fault}.

Theorem 1 (Coherence gate correctness)**.** *Let h and h' be the clean outputs of the original and replay executions on the same corrected input. Write $\tilde{h} = h + e$ and $\check{h} = h' + e'$, with independent transient perturbations e, e'. Let $\hat{\sigma}^{\mathrm{func}} > 0$ be the current normalization scale, $r_{\mathrm{func}} = \|\tilde{h} - \check{h}\|_2^2$, and $\bar{r}_{\mathrm{func}} = r_{\mathrm{func}}/\hat{\sigma}^{\mathrm{func}}$. For the thresholded gate in equation 8, with $\tau_{\mathrm{func}} \geq 0$ and $\beta > 0$, Assumptions A2.1–A2.1 imply:*

1. ***Novelty preserved.*** *If the state is screened, $e = e' = 0$, and $\delta_{\mathrm{fp}}^2/\hat{\sigma}^{\mathrm{func}} \leq \tau_{\mathrm{func}}$, then $g = 0$ and no correction is applied, regardless of how statistically unusual the state is.*
2. ***Corruption detected.*** *If the state is screened, $e' = 0$, and $\|e\|_2 > \delta_{\mathrm{fp}} + \sqrt{\tau_{\mathrm{func}}\hat{\sigma}^{\mathrm{func}}}$, then $\bar{r}_{\mathrm{func}} > \tau_{\mathrm{func}}$ and $g > 0.5$. Moreover, $g \to 1$ as $\beta(\bar{r}_{\mathrm{func}} - \tau_{\mathrm{func}}) \to \infty$.*
3. ***Double-fault probability.*** *For a fixed pair of executions, the probability that both experience a fault is p_{fault}^2.*

Proof. **(a)** For clean executions, $r_{\mathrm{func}} = \|h - h'\|_2^2 \leq \delta_{\mathrm{fp}}^2$. The normalized residual is therefore at or below the threshold, giving $g = 0$ by equation 8. **(b)** With $e' = 0$, the reverse triangle inequality gives $\|\tilde{h} - \check{h}\|_2 \geq \|e\|_2 - \delta_{\mathrm{fp}}$. The stated condition implies $\bar{r}_{\mathrm{func}} > \tau_{\mathrm{func}}$; the sigmoid branch then

gives $g > 0.5$ and the stated limit. **(c)** Independence gives $\Pr[\text{both executions experience a fault}] = p_{\text{fault}}^2$. □

These guarantees require the stated assumptions and a positive normalization scale. Unflagged states bypass replay, and subthreshold faults can remain undetected. The double-fault probability is unconditional for a fixed pair, not conditional on screening. Detection does not itself guarantee successful recovery. Stochastic components that violate the clean-execution bound, persistent faults, and faults correlated across executions fall outside the guarantees (A1).

A2.2 Minimax spectral recovery

[Signal-plus-noise model] The observation is $\tilde{x} = x + n$, where x and n are independent, $\mathbb{E}[x] = \mathbb{E}[n] = 0$, and $\mathrm{Cov}(x) = \Sigma = U\Lambda U^\top$ with orthogonal U and $\lambda_i > 0$. The noise covariance is $\mathrm{Cov}(n) = \sigma^2 I$, with unknown $\sigma^2 \in [\sigma^2_{\min}, \sigma^2_{\max}]$ and $0 \le \sigma^2_{\min} \le \sigma^2_{\max} < \infty$.

Theorem 2 (Minimax linear shrinkage)**.** *Under Assumption A2.2, the linear estimator $\hat{x} = W^\star \tilde{x}$ minimising $\sup_{\sigma^2 \in [\sigma^2_{\min}, \sigma^2_{\max}]} \mathbb{E}\|W\tilde{x} - x\|_2^2$ is diagonal in the eigenbasis of Σ, with*

$$W^\star = U\,\mathrm{diag}(w_1^\star, \dots, w_d^\star)\,U^\top, \qquad w_i^\star = \frac{\lambda_i}{\lambda_i + \sigma^2_{\max}}.$$

Proof. Let $A = U^\top W U$. Centering and independence give

$$\mathrm{MSE}(A, \sigma^2) = \sum_i \lambda_i (A_{ii} - 1)^2 + \sum_{i \ne j} \lambda_j A_{ij}^2 + \sigma^2 \sum_{i,j} A_{ij}^2.$$

For every fixed A, this expression is nondecreasing in σ^2, so its supremum occurs at $\sigma^2_{\max}$. Every off-diagonal entry contributes $(\lambda_j + \sigma^2_{\max}) A_{ij}^2$ and is therefore zero at the optimum. Writing $A = \mathrm{diag}(w_1, \dots, w_d)$ leaves

$$\mathrm{MSE}(A, \sigma^2_{\max}) = \sum_i \left[(1 - w_i)^2 \lambda_i + w_i^2 \sigma^2_{\max} \right].$$

Differentiation yields $w_i^\star = \lambda_i / (\lambda_i + \sigma^2_{\max})$. Since $\lambda_i > 0$, the objective is strictly convex, ensuring uniqueness. □

RIFT's shrinkage weights share this functional form with γ playing the role of $\sigma^2_{\max}$. This is a structural correspondence, not a claim that the replay-based severity is a calibrated noise-variance bound or that arbitrary radiation-induced perturbations satisfy Assumption A2.2.

A2.3 Shrinkage monotonicity

Proposition 1. *For $\lambda_i > 0$, $w_i(\gamma) = \lambda_i/(\lambda_i + \gamma)$ is strictly decreasing in $\gamma > 0$; for $\lambda_i = 0$, $w_i(\gamma) = 0$ identically.*

Proof. For $\lambda_i > 0$, $dw_i/d\gamma = -\lambda_i/(\lambda_i + \gamma)^2 < 0$. For $\lambda_i = 0$ and $\gamma > 0$, the numerator is zero. □

For fixed covariance eigenvalues, $\gamma_t^{(\ell)} = \gamma_{\min} + \kappa_c c_t^{(\ell)}$ is nondecreasing in $c_t^{(\ell)}$ when $\kappa_c \ge 0$. Consequently, the shrinkage weights are non-increasing with corruption severity. The decrease is strict for positive eigenvalues when $\kappa_c > 0$; zero-eigenvalue directions retain zero weight.

A2.4 Fusion stability

Proposition 2. *Suppose $g_t^{(\ell)} \in [0, 1]$, the normalized temporal and cross-layer residuals are nonnegative, and $\tau_{\text{temp}}, \tau_{\text{cross}} > 0$. Then the corrected state $\hat{h}_t^{(\ell)}$ lies in the convex hull of $\{\tilde{h}_t^{(\ell)}, \hat{h}_{t,\text{fused}}^{(\ell)}\}$, and $\hat{h}_{t,\text{fused}}^{(\ell)}$ lies in the convex hull of $\{\hat{h}_{t,\text{spec+re}}^{(\ell)}, \hat{h}_{t-1}^{(\ell)}, B_\ell \hat{h}_t^{(\ell-1)}\}$.*

Proof. The stated conditions place the temporal and cross-layer fusion coefficients in $[0, 1]$. Equations equation 9, equation 12, and equation 13 are therefore convex combinations; nesting them preserves membership in the corresponding convex hull. □

A3 COMPUTATIONAL COST

Each layer-step follows one of three paths. **Path A (unflagged):** screening costs $O(dk)$ for the spectral projection and $O(d^2)$ for a dense cross-layer predictor, or $O(dr)$ when the predictor is represented by rank-r factors. **Path B (flagged, coherent):** adds one complete layer re-execution. Its cost includes attention and feed-forward computation and depends on context length. **Path C (flagged, incoherent):** additionally applies spectral recovery and consistency fusion. Applying a dense spectral recovery operator costs $O(d^2)$ once its eigendecomposition is available; the convex blends cost $O(d)$, excluding predictor evaluation.

Let p_{flag} be the fraction of layer-steps that trigger replay outside burn-in and $p_{\text{corrupt}} \leq p_{\text{flag}}$ the fraction confirmed corrupt. Assuming equal layer costs, the expected replay-and-recovery compute overhead relative to the original layer evaluations is

$$\Delta_{\text{cost}}^{\text{replay+recovery}} = p_{\text{flag}} + p_{\text{corrupt}} \cdot \frac{c_{\text{recovery}}}{c_{\text{layer}}}, \tag{15}$$

where c_{layer} is the cost of one layer forward pass and c_{recovery} is the additional recovery cost per confirmed-corrupt layer-step. For the reported clean-input replay fraction $p_{\text{flag}} \approx 0.05$, replay alone adds approximately $5\,\%$ to layer-forward compute, independently of the number of layers.

Screening and running-statistics updates add work beyond the replay-and-recovery estimate. The $5.5\,\%$ clean-input wall-clock overhead reported in 9 is a separate measurement; it does not follow directly from the compute estimate in equation 15. Relating the two requires the cost of replayed layers, batching, and execution timings. A6.5 reports replay frequency and measured overhead.

Memory. Storing M_ℓ and B_ℓ as dense matrices requires $2Ld^2$ elements, approximately $4.30\,\text{GB}$ in FP32 or $2.15\,\text{GB}$ in FP16 for $L = 32$ and $d = 4096$. These figures exclude the spectral basis and other state. General rank-r factors for both matrices require $4Ldr$ elements, while a separately stored top-k spectral basis adds Ldk. For $r = 64$ and $k = d/4$, these two contributions occupy approximately $0.067\,\text{GB}$ and $0.268\,\text{GB}$ in FP16, respectively, before other state and workspace. Thus approximately $0.34\,\text{GB}$ counts factors plus a stored basis, rather than factors alone. Here GB denotes 10^9 bytes. Decomposing a rank-r covariance factor costs $O(dr^2 + r^3)$; this count does not include construction of additional spectral directions. A5.5 distinguishes these storage calculations from reported implementation footprints.

A4 DETAILED SETUP AND IMPLEMENTATION

This appendix expands the model, dataset, baseline, fault-injection, and hyperparameter details that 4 compresses into single paragraphs, and then describes the construction of the novelty benchmark used for the NPR metric in 2 and the methodological limitation of software-only fault injection.

A4.1 DETAILED SETUP

Models. LLaMA-3-8B (Grattafiori et al., 2024) is an 8 B-parameter decoder-only LLM with RMSNorm, evaluated in the LLM fault-tolerance literature (Xie et al., 2025; Zheng et al., 2025). **Qwen2.5-VL-7B** (Bai et al., 2025) is a vision-language model in which faults can propagate between visual and language representations, a setting evaluated in our cross-modal experiments. **OpenVLA-7B** (Kim et al., 2024) is a 7 B-parameter vision-language-action model trained on 970 k robot demonstrations; because a single corrupted action token can cause physical harm, it represents the safety-critical deployment scenario that motivates RIFT.

Datasets and metrics. For LLaMA-3-8B we report perplexity on **WikiText-2** (Merity et al., 2016), a likelihood-based measure of predictive quality also used by ReaLM (Xie et al., 2025). For Qwen2.5-VL-7B we report top-1 accuracy on **MMBench** (Liu et al., 2024), which measures visual question-answering performance under injected faults. For OpenVLA-7B we report task success rate on **LIBERO** (Liu et al., 2023), averaged over its four suites at 500 episodes each (Choi et al., 2026).

Baselines. *Tier 1* (LLM fault-tolerance methods, all 2025): **ReaLM** (Xie et al., 2025), a statistical ABFT framework with customized detection circuits that skips recovery when errors fall below

adaptive thresholds (DAC); **FT-Transformer** (Dai et al., 2025), which fuses attention into a single kernel and applies end-to-end ABFT with strided checksums plus selective neuron-value restriction (SC); and **SAVE** (Zheng et al., 2025), a software-only system that classifies bit vulnerability, allocates critical computations to reliable GPU memory, and verifies via DMA copy-back, range checks, and integer recomputation (ATC). *Tier 2* (complementary coverage): **ATTNChecker** (Liang et al., 2025), ABFT for attention computations (PPoPP '25); **Global Clipper** (Sha et al., 2024), activation range restriction adapted for transformer self-attention; and **RedNet** (Wang et al., 2024), a multi-exit architecture exploiting uneven layer-wise SEU sensitivity, evaluated using ground tests on Chaohu-1 payload hardware and radiation-error emulation. We additionally include **No Protection** as the unprotected reference and **TMR** (triple modular redundancy) as the redundancy reference.

Fault injection protocol. Prior work (Xie et al., 2025; Zheng et al., 2025; Dai et al., 2025) evaluates bit-flip faults, including structured patterns in SAVE. Because RIFT targets four distinct fault types, we adopt a physically motivated protocol that covers all four and injects faults at inference time on hidden states $h_t^{(\ell)}$ or on inter-layer data paths. **SEUs** introduce bit flips in the bfloat16 representation of $h_t^{(\ell)}$, with bit-error rate swept across $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$ per bit per decoding step; these are controlled injection rates, not a calibrated conversion from device cross-sections to orbital event rates. **MBUs** use logical bit clusters motivated by multi-cell radiation events: we flip a contiguous cluster of $C \in \{2, 4, 8\}$ bits across 1–4 adjacent elements of $h_t^{(\ell)}$ with $N_{\mathrm{mbu}} \in \{1, 3, 5\}$ independent events per layer per step. Multi-bit errors can exceed single-error-correction capability (Quinn et al., 2005); our logical clusters test sensitivity to structured activation corruption without emulating a device-specific ECC layout. **SETs** corrupt computation rather than storage: we add Gaussian perturbations with variance $(\alpha\sigma_\ell)^2$ on selected dimensions of the output of $F^{(\ell)}$ at a single decoding step, where σ_ℓ is the running standard deviation of clean activations and $\alpha \in \{1, 5, 10\}$ controls severity; the perturbation affects a random $10\,\%$ of dimensions and is not retained as a persistent perturbation; independence across executions is an assumption of the injection model. Finally, **bus corruption** models a layer that computes correctly but whose output arrives corrupted: a fraction $\rho \in \{5\%, 10\%, 25\%\}$ of the elements of $h_t^{(\ell)}$ are overwritten with samples from $\mathcal{N}(0, (10\sigma_\ell)^2)$ on the data path to layer $\ell{+}1$, occurring after the layer computation and contributing to the cross-layer residual r_{cross}.

A4.2 Detailed implementation

Running statistics use exponential moving average rates $\eta_\mu = \eta_M = 0.05$, with the spectral screen retaining $k = \lfloor d/4 \rfloor$ eigenvectors and the cross-layer predictor using ridge $\lambda_B = 0.01$, updated by recursive least squares with forgetting factor $\omega_{t,\tau} = 0.95^{t-\tau}$. Statistics are initialised from a 128-token warmup prefix during which the screening threshold τ_{screen} is calibrated to the 95th percentile of the clean combined score $s_t^{(\ell)}$ with residual weights $a_{\mathrm{spec}} = 0.4$, $a_{\mathrm{temp}} = 0.3$, $a_{\mathrm{cross}} = 0.3$. The coherence gate uses $\tau_{\mathrm{func}} = 10^{-4}$ with sharpness $\beta = 50$ and the thresholded sigmoid in equation 8; the threshold is applied to the normalized replay residual; spectral shrinkage uses $\gamma_{\min} = 10^{-3}$ and severity scaling $\kappa_c = 1.0$; re-execution blending uses $\tau_{\mathrm{re}} = 0.1$; temporal and cross-layer fusion thresholds are $\tau_{\mathrm{temp}} = \tau_{\mathrm{cross}} = 0.5$; and fault-aware decoding uses $\kappa_T = 2.0$. For baselines, ReaLM (Xie et al., 2025) and ATTNChecker (Liang et al., 2025) are adapted to LLaMA-3-8B; the ReaLM paper links its error-injection code online[1]. FT-Transformer, SAVE, Global Clipper, and RedNet are reimplemented from their published descriptions; TMR runs inference three times with bitwise majority voting; and methods are compared under the same fault-injection settings.

A4.3 Construction of the novelty benchmark

The novelty preservation rate (NPR) reported in 2 is computed on a held-out set of 500 sequences per model designed to be statistically unusual without injected hardware faults (i.e. statistical novelty should *not* be classified as corruption). For LLaMA-3-8B on WikiText-2, sequences are drawn from three categories of ~ 167 each: (i) WikiText-2 sequences whose mean per-token negative log-likelihood under a clean LLaMA-3-8B forward pass exceeds the 95th percentile of the WikiText-2 validation distribution—i.e. genuinely surprising tokens to the model itself rather than tokens we subjectively label rare; (ii) sequences containing controlled code-switching, constructed by inter-

[1] https://github.com/2000012835xt/ReaLM-DAC

leaving English with Mandarin or Spanish at clause boundaries, where each clause is grammatical in its own language; (iii) domain-shifted sequences drawn from medical, legal, and source-code corpora not present in WikiText-2. For Qwen2.5-VL-7B and OpenVLA-7B the analogous protocols substitute (i) benchmark examples with high output negative log-likelihood, (ii) cross-modal mismatches (an image-caption pair where the caption is plausible for a related image), and (iii) out-of-distribution scenes or instructions. These mismatches test whether semantic inconsistency is incorrectly classified as hardware corruption; they are not assumed to be correctly grounded image-caption pairs.

A sequence is counted as a false positive if RIFT's coherence gate equation 8 satisfies $g_t^{(\ell)} > 0$ at any layer-step during decoding, *in the absence of any fault injection*. Under the thresholded gate, this occurs exactly when a screened state's normalized replay residual exceeds τ_{func}. Statistical screening or re-execution alone does not count as a false positive. NPR is the fraction of evaluation sequences with no such event:

$$\text{NPR} = 1 - \frac{1}{500} \sum_{i=1}^{500} \mathbf{1}\Big\{ \exists\, (\ell, t) : g_{i,t}^{(\ell)} > 0 \Big\}.$$

Calibration is performed on a disjoint 200-sequence development set drawn from the same three categories; the 500 evaluation sequences are seen only at test time. This sequence-level criterion counts even one false gate activation as a failure, regardless of subsequent recovery or output quality. For the gate in equation 8, the conditions $g > 0$ and $g > 0.5$ are equivalent and therefore yield identical NPR values.

A4.4 Limitation: beam-test and on-orbit validation

Our fault injection protocol uses software-controlled perturbations. Software injection cannot fully reproduce real radiation effects: the temporal distribution of SET pulses, the physical spatial correlation of MBU clusters, and CMOS charge-collection dynamics require device-specific characterization. The burst-stress experiment in A6.4 tests temporally concentrated injections but does not reproduce these analog processes. RedNet (Wang et al., 2024) similarly combines ground tests on payload hardware with radiation-error emulation. Our controlled on-orbit experiment demonstrates execution and recovery on hardware operating in low Earth orbit. It does not validate recovery against naturally occurring radiation strikes or establish equivalence to radiation-beam testing.

A5 Ablation Studies

This appendix isolates the contribution of individual design choices in RIFT—the five-stage pipeline (A5.1), modality boundaries (A5.2), fault-type composition (A5.3), model scale (A5.4), running-statistics memory format (A5.5), and the per-method decomposition of the wall-clock cost (A5.6, A5.7)—as well as the per-task and per-output decomposition of the LIBERO and main-results numbers (A5.8, A5.9). Unless stated otherwise, the ablation results are reported as medians over five seeds. Values explicitly reproduced from 1 retain that table's three-run mean aggregation. The implementation details of A4.2 apply unless an experiment specifies otherwise.

A5.1 Component ablation

5 isolates the contribution of each RIFT stage by removing one component at a time on LLaMA-3-8B under MBU (C=4, N_{mbu}=3).

Without the coherence gate (Stage 2), NPR is $71.4\,\%$ in this ablation evaluation, the lowest among the listed individual component ablations. Removing spectral recovery (Stage 3) increases perplexity from 6.38 to 7.93, the largest individual task-quality decrease. Under MBU, removing cross-layer fusion has a larger effect than removing temporal fusion (7.21 vs. 6.84 perplexity). Removing all recovery stages gives perplexity 11.6, while screening-only rejection yields FDR of $89.3\,\%$ and NPR of $82.7\,\%$. These comparisons support the contribution of the evaluated components without establishing independent or multiplicative effects. The Stage 5 perplexity difference concerns output probabilities; positive temperature scaling cannot change greedy token selection.

Table 5: Ablation on LLaMA-3-8B (WikiText-2 perplexity ↓) under MBU (C=4, N=3). Each row removes one RIFT component; "Screening only" reduces the entire pipeline to Stage 1 with rejection on flag. Clean: 6.14. FDR is measured under MBU; NPR is measured on fault-free novelty sequences using the corresponding configuration. The full-RIFT NPR is reported separately in 2; the dash indicates that it is not repeated here.

Configuration	PPL	FDR (%)	NPR (%)
Full RIFT	**6.38**	**97.6**	—
− Coherence gate (Stage 2)	6.72	93.1	71.4
− Spectral recovery (Stage 3)	7.93	97.2	95.9
− Temporal fusion (Stage 4)	6.84	97.4	96.5
− Cross-layer fusion (Stage 4)	7.21	97.3	96.1
− Fault-aware decoding (Stage 5)	6.54	97.5	96.7
− All recovery (Stages 3–5)	11.6	96.8	95.1
Screening only (Stage 1 → reject)	8.14	89.3	82.7

Table 6: Qwen2.5-VL-7B accuracy (%) on MMBench under SEU (BER= 10^{-4}) with faults injected selectively into the vision encoder only, the language decoder only, or both. The "Both" column reproduces the SEU column of 1.

Method	Vision only	Language only	Both
No Protection	51.3	44.7	38.2
SAVE	67.4	63.8	62.1
Gl. Clipper	65.1	61.9	59.7
RIFT (ours)	**72.1**	**71.4**	**71.9**

A5.2 Cross-modal fault propagation

6 injects SEU faults selectively into the vision encoder, the language decoder, or both modules of Qwen2.5-VL-7B at BER $= 10^{-4}$.

Across the evaluated methods, language-only injection gives lower accuracy than vision-only injection ($44.7\,\%$ vs. $51.3\,\%$ unprotected). The gaps are 3.6 percentage points for SAVE, 3.2 for Global Clipper, and 0.7 for RIFT. With faults in both modules, RIFT achieves $71.9\,\%$, close to its $72.1\,\%$ vision-only result. These measurements support recovery across the tested injection locations; they do not isolate the contribution of the cross-layer predictor.

A5.3 Compound fault conditions

The main injection protocol evaluates one fault type at a time. 2 additionally reports LLaMA-3-8B perplexity on WikiText-2 under all seven non-empty subsets of SEU, SET, and bus corruption, each at the moderate-severity setting of 1. Selected fault types are injected independently in this experiment. MBU is not included in this compound-fault study.

These results show that RIFT retains relatively low perplexity under the evaluated compound conditions. They do not establish statistical independence between residuals or a multiplicative decomposition of their contributions.

A5.4 Scaling within the LLaMA-3 family and cross-architecture transfer

To assess whether the recovery quality and overhead reported in 4 extend across model scale and architecture, we evaluate RIFT on two additional members of the LLaMA-3 family at the smaller and larger ends—**LLaMA-3.2-3B** (28 layers, d=3072) and **LLaMA-3-70B** (80 layers, d=8192)—and on a fourth model from a different family at comparable scale, **Qwen2.5-72B-Instruct** (80 layers, d=8192, GQA with 64-head attention, post-instruction-tuned weights). All four are evaluated on WikiText-2 under SEU at BER $= 10^{-4}$. The 3B model runs on a single H100-80GB; the 70B and 72B models run on 2× H100-80GB with tensor parallelism. The model and injection settings follow

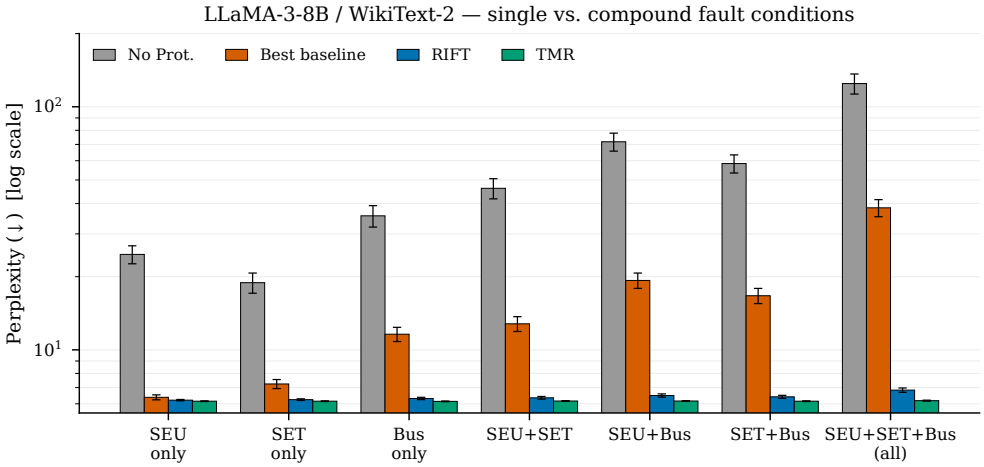


Figure 2: **LLaMA-3-8B perplexity on WikiText-2 under single and compound fault conditions.** Each compound condition activates the listed fault types simultaneously and independently at the moderate severities of 1 (SEU: BER $= 10^{-4}$; SET: $\alpha = 5$; bus: $\rho = 10\,\%$). RIFT degrades from 6.21 (SEU only) to 6.83 (all three simultaneously)—approximately a $10\,\%$ relative increase—whereas the strongest per-condition baseline degrades by $503\,\%$ over the same range (6.38 to 38.4). The comparison measures recovery under concurrent injected fault types.

Table 7: **Scaling within the LLaMA-3 family and cross-architecture transfer to Qwen2.5-72B-Instruct.** WikiText-2 perplexity (↓) under SEU (BER $= 10^{-4}$) and wall-clock overhead, median over 5 seeds. "Best baseline" is SAVE in every configuration. RIFT/TMR ratios show the relative recovery quality with respect to TMR. Qwen2.5-72B-Instruct, evaluated as a cross-family stress test at the 70 B-scale point, achieves a RIFT/TMR ratio of 1.008 and the same $5.5\,\%$ clean overhead as LLaMA-3-8B.

	WikiText-2 PPL (↓)				Overhead (%)		
Model	**Clean**	**No Prot.**	**SAVE**	**RIFT / TMR**	**Clean**	**Fault**	**RIFT/TMR**
LLaMA-3.2-3B	7.83	28.6	8.12	**7.90** / 7.83	5.7	8.4	1.009
LLaMA-3-8B	6.14	24.7	6.38	**6.21** / 6.15	5.5	8.2	1.010
LLaMA-3-70B	3.92	18.4	4.07	**4.05** / 3.97	5.4	8.0	1.020
Qwen2.5-72B-Instruct	3.74	17.9	3.96	**3.81** / 3.78	5.5	8.1	1.008

4, including the rank-r=64 low-rank factorisation of the running statistics (A5.5); the cross-layer predictor B_ℓ is initialised from a 128-token warmup prefix on each model exactly as for the main results.

7 reports task quality and wall-clock overhead for all four configurations.

RIFT's recovery quality remains within approximately $2\,\%$ relative perplexity of TMR at all three LLaMA-3 scales (RIFT/TMR ratio: $1.009/1.010/1.020$ at 3B/8B/70B). On Qwen2.5-72B-Instruct, the reported perplexities are 3.81 for RIFT and 3.78 for TMR, a ratio of 1.008. The spectral subspace and cross-layer predictor are estimated online from each model's activations. RIFT outperforms SAVE in all four configurations, although its relative advantage is not constant across scales. Clean overhead ranges from $5.4\,\%$ to $5.7\,\%$. These measurements support recovery across the evaluated scales and families; asymptotic operation counts alone do not determine their wall-clock overhead.

A5.5 Memory overhead

RIFT maintains, per layer, a running mean $\mu_\ell \in \mathbb{R}^d$, a second-moment matrix $M_\ell \in \mathbb{R}^{d\times d}$, a ridge predictor $B_\ell \in \mathbb{R}^{d\times d}$, and a top-$k$ spectral basis $U_{\ell,k} \in \mathbb{R}^{d\times k}$ for screening. Recovery in equation 10 additionally uses covariance eigenvalues and the full spectral operator. The screening

Table 8: **Reported RIFT memory accounting per model.** "Full" counts only M_ℓ and B_ℓ as dense $d\times d$ matrices. "Low-rank" (LR) gives the reported implementation footprint, including the stored spectral basis; its accounting therefore differs from the dense-matrix-only columns. "% of model" gives the reported ratio to model bfloat16 weight memory (e.g. 16 GB for LLaMA-3-8B).

Model	L	d	FP32 full	FP16 full	FP16 LR	% of model (FP16 LR)
LLaMA-3.2-3B	28	3072	2.11 GB	1.06 GB	0.18 GB	2.9 %
Qwen2.5-VL-7B	28	3584	2.88 GB	1.44 GB	0.27 GB	1.9 %
LLaMA-3-8B	32	4096	4.30 GB	2.15 GB	0.34 GB	2.1 %
OpenVLA-7B	32	4096	4.30 GB	2.15 GB	0.34 GB	2.4 %
LLaMA-3-70B	80	8192	42.9 GB	21.5 GB	2.80 GB	2.0 %

Table 9: **Computational overhead** (wall-clock latency increase %) on LLaMA-3-8B, measured over 1,000 tokens on a single A100-80GB. *Detection* is the per-step cost of monitoring (checksum, range check, screening, etc.); *correction* is the cost of recovery action when triggered. Cross-modal numbers in 3; BER dependence in 9; memory in 8.

Method	**Detection**	**Correction**	**Total (clean)**	**Total (BER$= 10^{-4}$)**
ReaLM (Xie et al., 2025)	1.4 %	0.4 %	1.8 %	4.2 %
FT-Trans. (Dai et al., 2025)	4.7 %	9.1 %	13.9 %	15.6 %
SAVE (Zheng et al., 2025)	5.1 %	3.7 %	8.8 %	12.4 %
ATTNChecker (Liang et al., 2025)	4.3 %	2.8 %	7.1 %	9.5 %
Gl. Clipper (Sha et al., 2024)	2.1 %	0.3 %	2.4 %	2.6 %
RedNet (Wang et al., 2024)	3.8 %	1.2 %	5.0 %	6.3 %
TMR	—	—	201 %	201 %
RIFT (ours)	3.6 %	1.9 %	**5.5 %**	**8.2 %**

basis alone does not specify that operator. 8 separates dense-matrix storage counts from the reported low-rank implementation footprints for five models.

The dense M_ℓ and B_ℓ matrices contain $2Ld^2$ elements. A stored spectral basis adds Ldk, and running means add Ld. General rank-r factors for both matrices contain $4Ldr$ elements. With FP16 storage, $r = 64$, and $k = d/4$, factors plus basis require approximately 0.336 GB for LLaMA-3-8B, 0.231 GB for Qwen2.5-VL-7B, and 3.020 GB for LLaMA-3-70B, excluding other state and workspace. These are storage calculations, not replacements for the reported implementation footprints in 8; those footprints are not derivable from the factor dimensions alone. For reference, 3.020 GB is approximately 2.813 GiB. The dense-only and implementation columns therefore do not establish a like-for-like total-memory reduction. A rank-r covariance decomposition supplies at most r nonzero-variance directions; handling $k > r$ screening directions and evaluating the full recovery operator require additional implementation details not specified by these memory counts.

A5.6 Detailed computational overhead breakdown

4.1 (Q4) summarises overhead in a single sentence. 9 gives the full per-method breakdown on LLaMA-3-8B, listing detection and correction costs alongside total overhead on clean inputs and under faults at BER $= 10^{-4}$.

9 reports RIFT overhead of $5.5\,\%$ on clean inputs and $8.2\,\%$ under faults, compared with $8.8\,\%$ and $12.4\,\%$ for SAVE. Global Clipper has lower overhead but lower novelty preservation and task quality (2,1). TMR's reported $201\,\%$ overhead is approximately $25\times$ RIFT's faulted overhead; this compares overhead percentages, not total inference runtimes. Replay frequency alone does not establish the wall-clock cost breakdown.

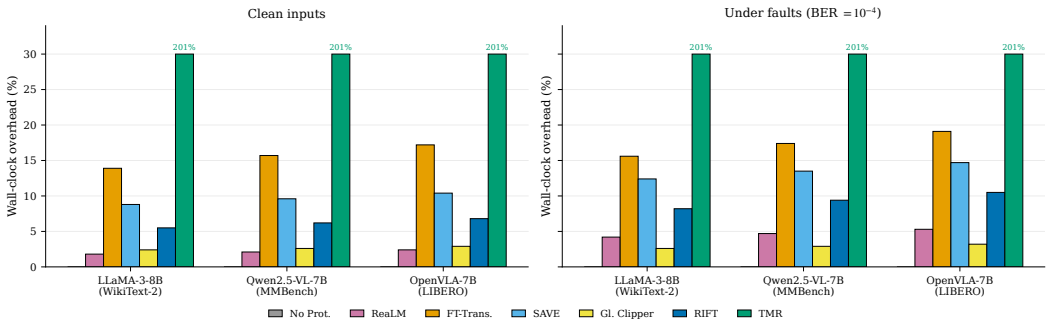


Figure 3: **Wall-clock overhead across modalities.** TMR is plotted at 30 % for legibility; its reported value is 201 % in all six cells. RIFT's overhead is 5.5 %/6.2 %/6.8 % on clean LLM/VLM/VLA respectively, rising to 8.2 %/9.4 %/10.5 % under faults. The plotted values describe the evaluated model implementations.

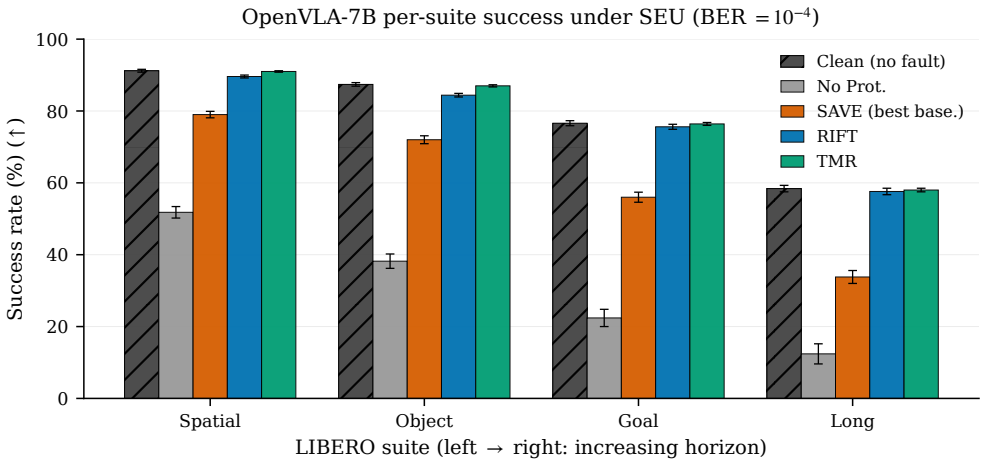


Figure 4: **Per-suite LIBERO success rate under SEU** (BER $= 10^{-4}$) on OpenVLA-7B, 500 episodes per suite, median over 5 seeds. RIFT's advantage over SAVE increases across the displayed suites—from $+10.6$ pp on Spatial, to $+12.4$ pp on Object, to $+19.6$ pp on Goal, to $+23.8$ pp on Long. RIFT closes the gap to TMR most aggressively on LIBERO-Long (57.6 % vs. 58.0 %), and trails TMR by at most 2.6 pp on every suite.

A5.7 Cross-modal wall-clock overhead

9 reports overhead only on LLaMA-3-8B. 3 extends the measurement to Qwen2.5-VL-7B and OpenVLA-7B, on both clean inputs and under faults at BER $= 10^{-4}$, on the same A100-80GB hardware.

Control-loop feasibility depends on the latency of a complete action update, including all required action tokens and runtime processing. Per-token latency alone does not determine control frequency. The overhead percentages in 3 therefore do not by themselves establish a robot control-rate guarantee.

A5.8 Per-suite LIBERO breakdown

The aggregate LIBERO success rate in 1 averages over the four evaluated LIBERO suites (Spatial, Object, Goal, Long), which differ markedly in horizon length and difficulty. 4 reports the corresponding suite-level SEU evaluation, summarized as five-seed medians rather than the three-run mean in 1.

Table 10: **Semantic preservation under SEU** (BER $= 10^{-4}$). Each cell is the similarity between paired runs with and without fault injection; the recovery method is applied only to the faulted run. Higher is better. *LLaMA-3-8B*: corpus BLEU-4 over 1,000 WikiText-2 64-token continuations. *Qwen2.5-VL-7B*: top-1 match rate over 500 MMBench questions. *OpenVLA-7B*: step-wise action match rate over 500 LIBERO episodes. "Δ to TMR" is the arithmetic mean of the three gaps to TMR after multiplying BLEU by 100. Median over 5 seeds.

Method	**LLaMA-3-8B** BLEU-4 (↑)	**Qwen2.5-VL-7B** Top-1 match % (↑)	**OpenVLA-7B** Action match % (↑)	**Δ to TMR (avg.)**
No Protection	0.487	51.3	36.2	53.6
ReaLM	0.864	83.7	79.2	15.9
FT-Trans.	0.871	82.9	78.4	16.2
SAVE	0.879	84.5	81.6	14.3
ATTNChecker	0.823	79.2	75.1	20.1
Gl. Clipper	0.832	81.4	78.9	17.8
RedNet	0.798	76.8	73.2	22.4
TMR	0.989	99.4	98.7	0.0
RIFT (ours)	**0.964**	**95.8**	**94.1**	**3.6**

The largest RIFT–SAVE gap among these suites is on LIBERO-Long (23.8 percentage points), where RIFT also has its smallest gap to TMR (0.4 percentage points). These suite-level comparisons indicate variation in recovery gains across tasks; they do not isolate horizon length from other differences between the suites.

A5.9 Semantic preservation under fault recovery

1 measures *aggregate* task quality, but two methods can achieve identical perplexity or accuracy on a benchmark while producing different outputs on individual examples. For an LLM deployed on orbit to summarise a sensor stream, or for a VLA closing a manipulation loop, output agreement with an uncorrupted run provides an additional measure of recovery quality. We refer to this as *semantic preservation* and operationalise it as the similarity between outputs of two paired runs: a clean run with no fault injection, and a faulted run with the method's recovery applied. Both runs use the same prompts, same seeds, the stated decoding setup (T=0.7 for the LLM, greedy decoding for the VLM and VLA), and the same input batches; recovery is enabled only for the faulted run. The $T = 0.7$ setting is specific to this output-comparison protocol; this experiment does not isolate the contribution of the adaptive temperature in Stage 5.

The similarity metric is chosen per modality. For LLaMA-3-8B, we compute corpus-level BLEU-4 between recovered and clean outputs over 1,000 WikiText-2 prompts with 64-token continuations. For Qwen2.5-VL-7B, we compute the top-1 answer match rate over the 500 MMBench questions used in 1. For OpenVLA-7B, we compute step-wise action match rate, where two action tokens at the same step in paired runs match if their argmax over the discretised action vocabulary agrees, averaged over all steps in 500 LIBERO episodes. 10 reports all three.

Three observations follow. First, RIFT is closer to TMR than every evaluated non-oracle baseline, exceeding them by 8.5–16.6 BLEU points on the LLM and 11.3–20.9 percentage points on the VLM and VLA. Second, aggregate quality and output similarity measure different properties: ReaLM achieves perplexity 6.43, only 0.29 above clean, but BLEU-4 of 0.864 against clean outputs. RIFT achieves BLEU-4 of 0.964; this is not an exact token-agreement percentage. Third, OpenVLA's 94.1 % is step-wise action agreement, 4.6 percentage points below TMR, rather than the fraction of entire trajectories with no divergent action.

A6 Sensitivity Analysis

This appendix probes RIFT's robustness to perturbations along the axes a careful reader is likely to question: variance across random seeds (A6.1), the screening and recovery hyperparameters introduced in 3 (A6.2), the bit-error rate at which RIFT ceases to track TMR (A6.3), the temporal

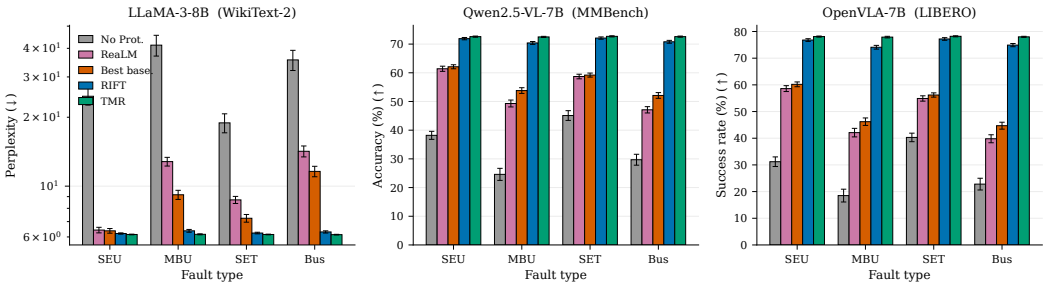


Figure 5: **Main results with $\pm 1\sigma$ error bars.** Bars show the model–fault–method settings of 1; error bars summarize between-run standard deviation.

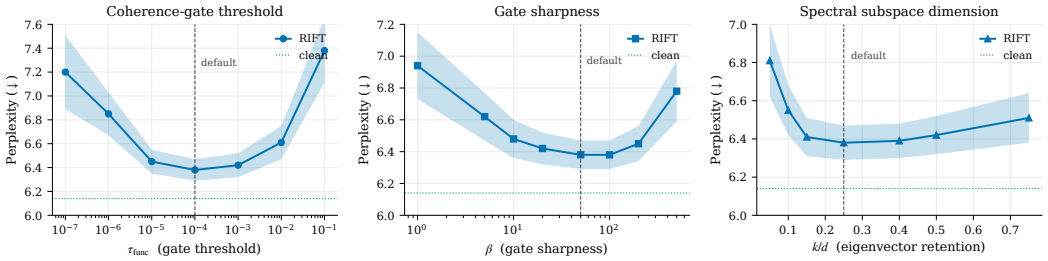


Figure 6: **Threshold and hyperparameter sensitivity on LLaMA-3-8B under MBU** ($C = 4$, $N_{\text{mbu}} = 3$). Shaded bands are $\pm 1\sigma$ over 5 seeds; vertical dashed lines mark the defaults used in 4. **(a)** The coherence-gate threshold has a two-decade plateau between $\tau_{\text{func}} = 10^{-5}$ and 10^{-3}. **(b)** The gate sharpness has a plateau between $\beta \in [20, 100]$. **(c)** The eigenvector retention is robust over $k/d \in [0.15, 0.5]$. The plotted curves summarize sensitivity within the tested parameter ranges.

correlation of the fault stream (A6.4), and the fraction of layer-steps that trigger gate firing as a function of fault rate (A6.5). Together these characterize recovery over the tested operating conditions and complement 1.

A6.1 Statistical reliability

5 presents the four-fault model–method comparisons with one-standard-deviation error bars summarizing between-run variability. §1, 2,3 report means over three independent runs; appendix figures and tables identify five-seed summaries where applicable. These are descriptive summaries of variability, not pooled hypothesis tests across perplexity, accuracy, and success rate.

A6.2 Threshold and hyperparameter sensitivity

6 sweeps three parameters on LLaMA-3-8B under MBU ($C = 4, N_{\text{mbu}} = 3$): the coherence-gate threshold τ_{func}, gate sharpness β, and spectral-subspace dimension k. The screening threshold τ_{screen} also controls replay frequency but is not swept in this figure. Other settings are held at their defaults. In the reported pilot experiments, varying an individual running-statistics rate, ridge penalty, screening weight, fusion threshold, or severity scale within an order of magnitude of its default changed WikiText-2 perplexity by less than 0.10.

The three panels show ranges over which task quality is relatively stable. Hyperparameters were chosen prior to evaluation; the test set was not used for selection. The thresholded gate preserves clean states when their normalized replay residual does not exceed τ_{func}. The figure does not contain a screening-percentile sweep and therefore does not establish robustness over screening thresholds.

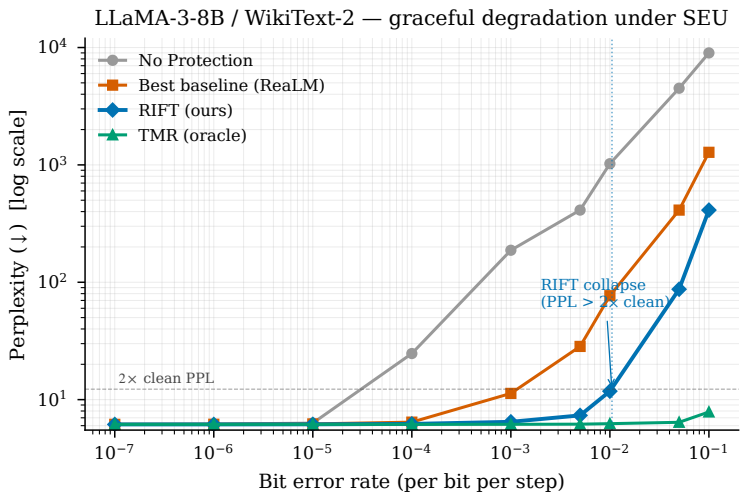


Figure 7: **Graceful degradation of RIFT under increasing SEU bit-error rate** on LLaMA-3-8B/WikiText-2 (median over 5 seeds). The dashed horizontal line marks twice the clean perplexity ($2 \times 6.14 = 12.28$), which defines the collapse criterion. RIFT remains below this threshold at the plotted BER of 10^{-2} and exceeds it at the next higher tested rate. ReaLM crosses the threshold between the plotted BER values of 10^{-3} and 5×10^{-3}. Perplexity increases sharply at higher fault rates.

A6.3 Failure-mode analysis: graceful degradation and collapse threshold

A fault-tolerance method that wins at one severity but collapses sharply outside that operating point is undesirable for orbital deployment, where the bit-error rate varies with solar activity, orbit altitude, and other mission conditions. 7 sweeps SEU bit-error rate over five orders of magnitude on LLaMA-3-8B and overlays four methods: No Protection, ReaLM, RIFT, and TMR.

RIFT tracks TMR closely up to BER $= 10^{-3}$ and exceeds $2\times$ clean perplexity above BER $\approx 10^{-2}$. Under the independent-fault model, an individual replay is clean with probability $1 - p_{\text{fault}}$ at every fault rate; p_{fault}^2 instead describes simultaneous corruption of a fixed pair of executions. At high fault rates, replay corruption and residual errors in running statistics can limit recovery. The SEU rates of 10^{-6} to 10^{-3} per bit per step in 3 are controlled injection settings. Mapping them to orbital conditions requires device cross-sections, particle spectra, shielding, and execution duration.

A6.4 Stress test: sustained fault bursts

The protocol of A4.1 assumes independent injections across decoding steps. To examine temporally concentrated faults, we hold the time-averaged BER at 10^{-4} and vary the burst length L, the number of consecutive decoding steps over which faults are concentrated. Here L denotes burst duration rather than model depth. The $L = 1$ condition serves as the independent-injection reference. Longer bursts can stress the running statistics and cross-layer predictor when residual corruption persists in the corrected states used for their updates. 8 reports the results.

Three observations are noteworthy. First, RIFT maintains relatively low perplexity under sustained bursts; its perplexity at $L = 50$ is 7.84. Second, an additional intervention indicates that running-statistics updates contribute to degradation: when we additionally freeze M_ℓ during bursts (using a two-EMA scheme with a fast detector on ∂M_ℓ), perplexity at $L = 100$ drops from 9.41 to 7.62, reducing the excess perplexity above clean. We retain the simpler single-EMA design in the main paper to avoid the additional mechanism; the two-EMA variant is reported as a robustness option for high-flux deployments. Third, TMR remains comparatively stable across the tested burst lengths; this observation depends on how injected bursts affect its replicated executions.

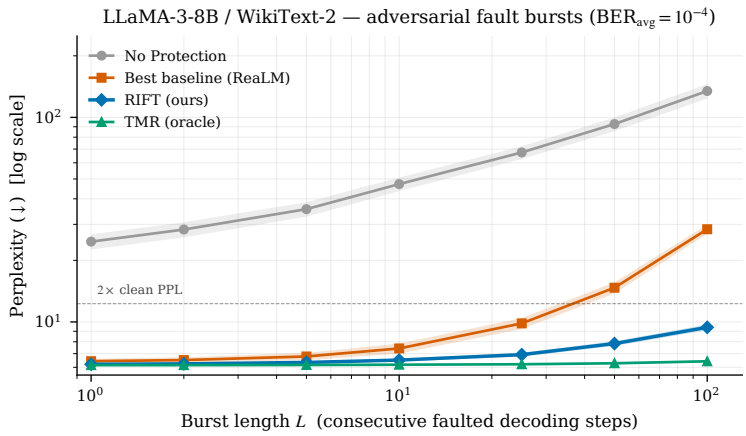


Figure 8: **LLaMA-3-8B perplexity under sustained fault bursts at fixed time-averaged BER $= 10^{-4}$.** The x-axis is burst length L, the number of consecutive decoding steps over which faults are concentrated; L=1 reproduces the independent-injection setting of 1. RIFT remains below $2\times$ clean perplexity across the full tested range (the dashed line marks $2 \times 6.14 = 12.28$; RIFT's worst value, at $L = 100$, is 9.41), whereas the strongest baseline crosses that line at $L \approx 30$ and is nearly $5\times$ above clean by $L = 100$. TMR varies little over the tested burst lengths under the redundancy and injection protocol used here.

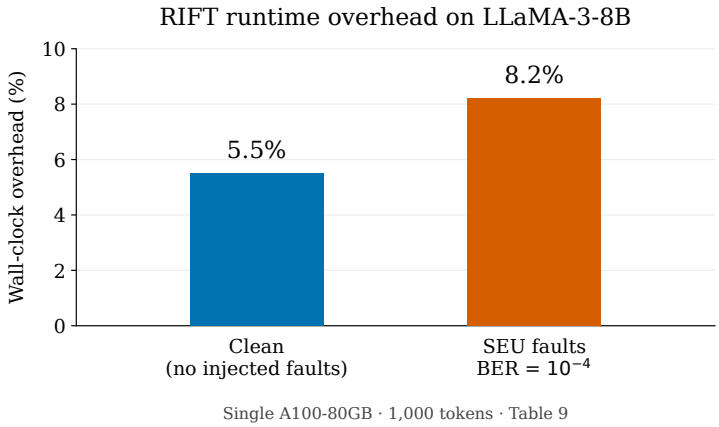


Figure 9: **Wall-clock overhead on LLaMA-3-8B.** RIFT incurs $5.5\,\%$ overhead without injected faults and $8.2\,\%$ under SEU faults at BER $= 10^{-4}$, relative to unprotected inference. Values are reproduced from Table 9, measured over 1,000 tokens on a single A100-80GB GPU.

A6.5 Replay frequency and computational overhead

9 reports the fraction of layer-steps that trigger re-execution after statistical screening, as specified in equation 7, alongside the resulting wall-clock overhead, as the SEU bit-error rate is swept over six orders of magnitude.

15 gives a layer-compute estimate: replaying approximately $5\,\%$ of layer-steps on clean inputs adds approximately $5\,\%$ layer-forward work before screening, statistics updates, and recovery. The reported $5.5\,\%$ clean wall-clock overhead is measured separately. Replay frequency, confirmed-corruption frequency, and wall-clock overhead are distinct quantities; a timing breakdown is required to relate them.

Table 11: Controlled fault recovery on a commercial LEO platform. The experiment includes 100 injected events per fault type. Token agreement is measured against fault-free execution. All values are percentages, and higher is better.

Metric	SEU	MBU	SET	Bus
Token agreement, no protection	68.7	52.4	70.9	57.8
Token agreement, RIFT	**93.4**	**90.1**	**92.8**	**90.6**
FDR, RIFT	93.0	91.0	92.0	90.0

Table 12: Aggregate recovery and operational measurements for the controlled on-orbit experiment. Clean and faulted runtime overheads are reported separately. Zero observed events refer only to the monitored experiment.

Measurement	Value
Aggregate FDR	91.5%
Sequence-level NPR	93.8%
Recovered-token agreement	91.7%
Clean / faulted gate firing	6.2% / 15.3%
Clean / faulted runtime overhead	8.4% / 12.6%
Mean / peak temperature	51.6 / 61.3 °C
Mean / peak power	14.7 / 20.2 W
CUDA Xid events / resets / throttling	0 / 0 / 0
ECC/EDAC counters	Not exposed

A7 On-Orbit Deployment and Additional Robustness Experiments

We extend the main evaluation with a controlled on-orbit deployment and four additional studies. These examine nonuniform fault placement, structured perturbations and replay persistence, sustained domain shift, and long-context statistical stability. Together, the experiments assess deployment feasibility, sensitivity to the injected error model, and the effect of online adaptation on valid computation.

A7.1 Controlled Fault Recovery in Low Earth Orbit

Platform and protocol. We deployed RIFT on a commercial on-orbit compute platform in low Earth orbit (provider and platform names omitted for anonymous review). The platform provides programmable orbital compute with an NVIDIA GPU and support for containerized applications. Our experiment used a Jetson Orin NX node with 16 GB memory and containerized PyTorch L4T/CUDA. The evaluated model was a compact six-layer decoder with four attention heads, 0.31 million int8 parameters, and FP16 activations. This deployment complements the main 7–8B-model evaluation by testing execution on orbital hardware under its operational compute, power, and thermal conditions.

The workload comprised 64 prompts, including 32 standard and 32 novelty-rich prompts, and 2,048 generated tokens. We introduced 400 controlled faults, with 100 injections for each of SEU, MBU, SET, and bus corruption. We measured fault detection rate (FDR), recovered-token agreement with fault-free execution, sequence-level novelty preservation rate (NPR), runtime overhead, and system telemetry. Table 11 reports recovery by fault type, and Table 12 summarizes aggregate and operational measurements.

Recovery performance. RIFT achieved 91.5% aggregate FDR, 93.8% sequence-level NPR, and 91.7% recovered-token agreement. Relative to unprotected inference, token agreement improved by 24.7, 37.7, 21.9, and 32.8 percentage points under SEU, MBU, SET, and bus corruption, respectively. The largest gains occurred under MBU and bus corruption, although these conditions also retained the largest residual disagreement. These results demonstrate effective recovery from controlled numerical corruption while the model executes on COTS hardware in orbit.

Table 13: Fixed-budget fault-placement control on LLaMA-3.2-3B. PPL/clean is the ratio to fault-free perplexity. Agreement denotes token agreement with clean execution.

Placement	PPL/clean ↓	FDR (%) ↑	Agreement (%) ↑
Uniform	1.041	97.2	96.1
80/20 hotspot	1.047	96.8	95.8
Attention only	1.052	96.3	95.4
MLP only	1.049	96.5	95.6
Worst 20% of layers	1.061	95.7	94.8

Runtime and scope. RIFT added 8.4% runtime overhead on clean execution and 12.6% under injected faults, with gate firing increasing from 6.2% to 15.3%. The monitored run recorded no CUDA Xid event, reset, or thermal-throttling event, and spacecraft telemetry remained nominal. Mean and peak temperatures were 51.6 and 61.3 °C, and mean and peak power were 14.7 and 20.2 W. Differences from the main GPU experiments cannot be attributed to the orbital environment alone because model size, quantization, calibration, and runtime conditions also differ.

The faults were injected in software and were not synchronized with radiation telemetry. Consequently, this experiment establishes controlled recovery and deployment feasibility on hardware physically operating in orbit. It does not constitute a beam test, flight qualification, or observation of naturally occurring radiation strikes, and it does not experimentally validate the mapping from particle interactions to the injected tensor-level errors. The compact model and limited workload further bound the scope of the deployment evidence.

A7.2 Sensitivity to Fault Placement

We tested sensitivity to nonuniform logical fault placement using LLaMA-3.2-3B on 128 WikiText-2 sequences over three seeds. Each condition used 300 moderate-severity MBU injections, with the total event budget held fixed while placement varied across uniform assignment, an 80/20 hotspot, attention-only placement, MLP-only placement, and concentration in the worst 20% of layers. The clean perplexity was 7.83.

Under worst-layer concentration, the PPL/clean ratio increased by 0.020 relative to uniform placement, while FDR and token agreement decreased by 1.5 and 1.3 percentage points. Thus, recovery remained effective across the tested logical placements. This control measures sensitivity to where faults enter model computation. It does not model physical die-level exposure, since model layers share hardware resources over time.

A7.3 Structured Perturbations and Replay Persistence

We examined dependence on the injection geometry and the non-recurrence assumption using LLaMA-3.2-3B on 64 WikiText-2 sequences over three seeds. Each condition contained 300 moderate-severity events, comprising 100 SEUs, 100 MBUs, and 100 SETs, with event count and perturbation norm controlled across comparisons. The structured mixture replaced uniform bit selection with significance-weighted SEUs, logical MBU contiguity with permutation-scrambled clusters affecting nonadjacent logical values, and diffuse SET noise with norm-matched low-rank perturbations confined to contiguous activation tiles. We additionally varied ρ_{persist}, the probability that an injected corruption remains identical during replay.

Without replay persistence, the structured mixture reduced FDR by 1.3 percentage points and token agreement by 0.9 points, while increasing PPL/clean by 0.014. Recovery therefore remained effective under these alternative numerical geometries. Persistence had a larger effect. At $\rho_{\mathrm{persist}} = 0.25$, FDR fell to 72.2% and PPL/clean rose to 1.402. Identical corruption in both executions produces little functional discrepancy, exposing the gate's intended boundary. These results quantify sensitivity to that boundary without establishing the prevalence of replay-persistent errors on a particular device. Stored-weight, cache, or register corruption that survives replay requires complementary integrity checking or restoration.

Table 14: Sensitivity to perturbation geometry and replay persistence. Agreement denotes token agreement with clean execution. Increasing ρ_{persist} tests departures from the gate's non-recurrence assumption.

Mixture	ρ_{persist}	FDR (%)	Agreement (%)	PPL/clean
Original	0	96.9	95.8	1.043
Structured	0	95.6	94.9	1.057
Structured	0.05	90.9	93.1	1.096
Structured	0.10	86.2	91.4	1.145
Structured	0.25	72.2	86.1	1.402

Table 15: Fault-free sustained domain-shift control. All entries are percentages. Initial and final screen rates measure screening activity, while false intervention measures unintended correction.

Condition	Initial screen	Final screen	False intervention	Clean agreement
In-domain	5.2	5.1	0.2	99.4
Shift, standard RIFT	13.2	5.8	0.4	99.1
Shift, frozen statistics	13.3	12.4	0.7	98.6

A7.4 Adaptation under Sustained Domain Shift

We evaluated LLaMA-3.2-3B on 30 medical, legal, and source-code streams without injected faults. Each stream began with a 128-token WikiText-2 warm-up and then switched to a continuous 1,024-token shifted regime. We compared standard EMA/RLS adaptation with statistics frozen after warm-up.

With standard adaptation, screening fell from 13.2% immediately after the shift to 5.8%, approaching the in-domain rate of 5.1%. Frozen statistics retained a 12.4% final screening rate. Standard RIFT limited false intervention to 0.4% and maintained 99.1% agreement with clean execution. These observations support adaptation to sustained distribution changes over the evaluated streams, while showing that agreement on fault-free inputs is high but not exact.

A7.5 Long-Context Statistical Stability

We assessed whether corrected-state updates cause cumulative interference using 45 English examples from Qasper, MultiFieldQA-en, and GovReport in LongBench-E (Bai et al., 2024), with five examples per task from each of three context-length strata. LLaMA-3.2-3B was evaluated under fault-free execution and an early-fault condition in which moderate SEUs were restricted to the first context quarter, followed by fault-free execution. Early-fault conditions used five fault-injection seeds, with paired deterministic fault-free executions using identical inputs. Standard corrected-state updates were compared with statistics frozen after the 128-token warm-up.

Final-half false intervention measures the percentage of fault-free layer-steps in the final context half at which the gate confirms corruption and applies a nontrivial correction. Output similarity compares the RIFT output with the corresponding clean execution without RIFT using the task-appropriate LongBench metric. Task-score retention is the achieved task score divided by its clean-reference score. Table 16 summarizes the results under fault-free and early-fault conditions, comparing adaptive and frozen statistics.

Following early corruption, standard RIFT retained 98.7% output similarity and 99.2% task performance, with 0.5% final-half false intervention. Freezing the statistics changed false intervention by at most 0.2 percentage points and similarity and score retention by at most 0.3 points. The small differences provide evidence against substantial cumulative interference from corrected-state updates over the evaluated contexts, while not implying stability for arbitrary context lengths or fault histories.

Table 16: Long-context statistics-stability control. All entries are percentages. False intervention is measured only over fault-free layer-steps in the final context half.

Condition	False intervention ↓	Output similarity ↑	Score retention ↑
Fault-free, standard	0.3	99.1	99.7
Fault-free, frozen	0.2	99.2	99.8
Early SEU, standard	0.5	98.7	99.2
Early SEU, frozen	0.3	99.0	99.5

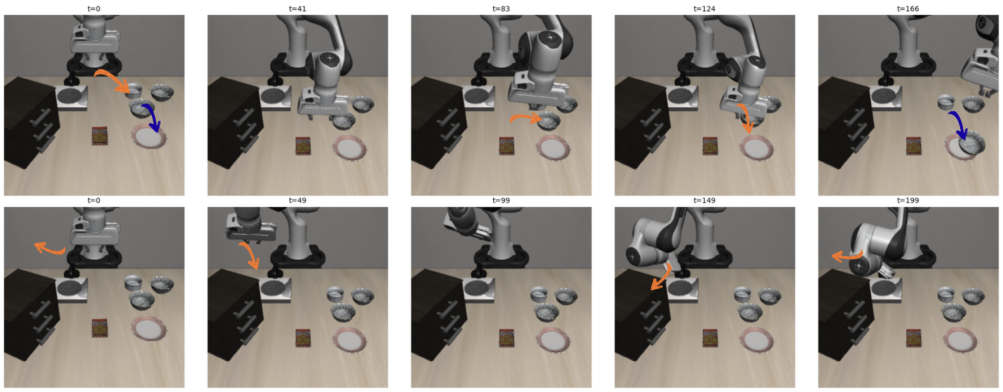


Figure 10: OpenVLA trajectories on LIBERO Task 0 under injected transient faults. Top row: with RIFT, the robot successfully places the black ball on the plate. Bottom row: without RIFT, the robot fails to complete the task.

A8 Qualitative Case Study under Injected Transient Faults

We present a qualitative OpenVLA (Kim et al., 2024) example on LIBERO (Liu et al., 2023) to complement the quantitative results in Table 1. Transient faults are injected during inference using the radiation-inspired fault model from the main experiments.

LIBERO Task 0: embodied manipulation with OpenVLA. The task instruction is: *"Pick up the black bowl between the plate and the ramekin and place it on the plate."* Figure 10 compares trajectories with and without RIFT under the same fault-injection setting. With RIFT enabled, OpenVLA completes the placement task. Without RIFT, the robotic arm exhibits erratic movement and fails to complete the task. This case illustrates the effect of inference-time fault recovery on task execution in a simulated manipulation environment.

A9 Additional Cross-Model Generalization

The main evaluation in Section 4 studies one representative model from each deployment category: language-only, vision-language, and vision-language-action. To assess recovery beyond the 7–8B model scale, we additionally evaluate two larger language-only models and two larger vision-language models under the four transient fault regimes used in the main text: SEU, MBU, SET, and bus corruption.

Additional language-only models. Table 17 reports WikiText-2 perplexity for LLaMA-3.1-70B and Qwen2.5-72B-Instruct. Across both models and all four fault types, RIFT achieves the lowest perplexity among the evaluated non-oracle methods and remains close to TMR. Its advantage over SAVE and Global Clipper is particularly pronounced under MBU and bus corruption. These results extend the recovery behavior observed in the main evaluation to larger models from two language-model families.

Table 17: Additional language-only model results on WikiText-2 under moderate-severity transient faults. Perplexity is reported; lower is better. Bold indicates the best non-oracle result among the evaluated methods.

Model	Method	Clean	SEU	MBU	SET	Bus
LLaMA-3.1-70B	No Protection	3.94	18.7	31.5	15.2	27.8
	SAVE	3.94	4.11	6.92	5.36	7.84
	Global Clipper	3.94	4.68	5.74	6.21	6.58
	TMR	3.94	3.96	3.97	3.96	3.95
	RIFT (ours)	3.94	**4.03**	**4.24**	**4.06**	**4.18**
Qwen2.5-72B-Instruct	No Protection	3.74	17.9	29.2	14.6	25.9
	SAVE	3.74	3.96	6.55	5.18	7.31
	Global Clipper	3.74	4.41	5.39	6.04	6.27
	TMR	3.74	3.78	3.79	3.78	3.77
	RIFT (ours)	3.74	**3.81**	**4.07**	**3.91**	**4.03**

Table 18: Additional vision-language model results on MMBench under moderate-severity transient faults. Top-1 accuracy (%) is reported; higher is better. Bold indicates the best non-oracle result among the evaluated methods.

Model	Method	Clean	SEU	MBU	SET	Bus
Qwen2.5-VL-32B	No Protection	78.6	44.8	31.9	51.6	36.7
	SAVE	78.6	68.4	55.3	60.7	58.2
	Global Clipper	78.6	66.9	59.1	56.8	60.4
	TMR	78.6	78.3	78.1	78.4	78.2
	RIFT (ours)	78.6	**77.2**	**75.6**	**77.5**	**75.9**
InternVL2.5-38B	No Protection	77.9	43.6	30.7	50.2	35.4
	SAVE	77.9	67.1	54.6	59.8	57.3
	Global Clipper	77.9	65.8	58.4	55.9	59.6
	TMR	77.9	77.6	77.5	77.7	77.5
	RIFT (ours)	77.9	**76.5**	**74.9**	**76.8**	**75.2**

Additional vision-language models. Table 18 reports MMBench top-1 accuracy for Qwen2.5-VL-32B and InternVL2.5-38B. RIFT achieves the highest accuracy among the evaluated non-oracle methods across all four fault types. Its largest gains over the strongest baseline occur under MBU and bus corruption. These results support the applicability of RIFT to larger vision-language models, although aggregate accuracy alone does not isolate the contribution of individual recovery stages or establish where corruption propagates across modalities.

Together, these results support cross-model generalization under the evaluated injection protocols. Across the additional models, RIFT remains close to TMR in task quality and consistently outperforms the evaluated non-oracle baselines. The findings extend the empirical evidence beyond the models used in the main evaluation, without establishing architecture-independent recovery or validation against naturally occurring radiation events.

A10 Algorithm

The complete RIFT inference procedure is given in A10. Per-layer state consists of the running mean μ_ℓ, second moment M_ℓ, ridge cross-layer predictor B_ℓ, robust scales $\hat{\sigma}_\ell^{\text{spec}}, \hat{\sigma}_\ell^{\text{temp}}, \hat{\sigma}_\ell^{\text{cross}}, \hat{\sigma}_\ell^{\text{func}}$ used to normalise the four residuals, and the previous corrected state $\hat{h}_{t-1}^{(\ell)}$ used by equation 5 and equation 12. State is initialised either from a short calibration pass on the prompt prefix or, when no calibration data is available, with conservative defaults ($\mu_\ell = 0$, $M_\ell = \epsilon I$, $B_\ell = I$, $\hat{\sigma}_\ell^{\bullet} = 1$). Without calibration, gating is disabled for the first t_{burn} steps while statistics are updated; with calibration, $t_{\text{burn}} = 0$. Following equation 8, states whose normalized replay residual does not exceed τ_{func} pass through unchanged. Above the threshold, the sigmoid supplies a graded recovery weight.

[t] RIFT Inference with Functional Coherence Gating [1] Model f_θ with L layers; prompt x; hyperparameters $\Theta = \{\tau_{\text{screen}}, \tau_{\text{func}}, \tau_{\text{re}}, \tau_{\text{temp}}, \tau_{\text{cross}}, \beta, k, \gamma_{\min}, \kappa_c, \kappa_T, t_{\text{burn}}\}$ Generated sequence $\hat{y}_{1:T}$ Initialise per-layer state $\{\mu_\ell, M_\ell, B_\ell, \hat{\sigma}_\ell^{\text{spec}}, \hat{\sigma}_\ell^{\text{temp}}, \hat{\sigma}_\ell^{\text{cross}}, \hat{\sigma}_\ell^{\text{func}}, \hat{h}_0^{(\ell)}\}_{\ell=1}^{L}$; set $t_{\text{burn}} \leftarrow 0$ if calibrated $t = 1, \ldots, T$ $\hat{h}_t^{(0)} \leftarrow \text{Embed}(x_t)$ $\ell = 1, \ldots, L$ $g_t^{(\ell)}, c_t^{(\ell)} \leftarrow 0$, 0 default contribution to Stage 5 $\tilde{h}_t^{(\ell)} \leftarrow F^{(\ell)}\big(\hat{h}_t^{(\ell-1)}; \theta^{(\ell)}\big)$ layer output (possibly corrupted) *Stage 1: Statistical screening* Compute $r_{\text{spec},t}^{(\ell)}, r_{\text{temp},t}^{(\ell)}, r_{\text{cross},t}^{(\ell)}$ via equation 4–equation 6 $\bar{r}_{\bullet,t}^{(\ell)} \leftarrow r_{\bullet,t}^{(\ell)} / \hat{\sigma}_\ell^{\bullet}$ for $\bullet \in \{\text{spec}, \text{temp}, \text{cross}\}$; $s_t^{(\ell)} \leftarrow \sum_\bullet a_\bullet \bar{r}_{\bullet,t}^{(\ell)}$ $t \le t_{\text{burn}}$ **or** $s_t^{(\ell)} \le \tau_{\text{screen}}$ $\hat{h}_t^{(\ell)} \leftarrow \tilde{h}_t^{(\ell)}$ pass through (burn-in or unflagged) **goto** line A10 *Stage 2: Thresholded soft coherence gate* $\check{h}_t^{(\ell)} \leftarrow F^{(\ell)}\big(\hat{h}_t^{(\ell-1)}; \theta^{(\ell)}\big)$ re-execute on corrected input $r_{\text{func},t}^{(\ell)} \leftarrow \big\|\tilde{h}_t^{(\ell)} - \check{h}_t^{(\ell)}\big\|_2^2$; $\bar{r}_{\text{func},t}^{(\ell)} \leftarrow r_{\text{func},t}^{(\ell)} / \hat{\sigma}_\ell^{\text{func}}$ Eq. equation 7 $\bar{r}_{\text{func},t}^{(\ell)} \le \tau_{\text{func}}$ $\hat{h}_t^{(\ell)} \leftarrow \tilde{h}_t^{(\ell)}$ coherent; $g_t^{(\ell)} = 0$ **goto** line A10 $g_t^{(\ell)} \leftarrow \sigma\big(\beta\,(\bar{r}_{\text{func},t}^{(\ell)} - \tau_{\text{func}})\big)$; $c_t^{(\ell)} \leftarrow \bar{r}_{\text{func},t}^{(\ell)}$ Eq. equation 8 *Stage 3: Spectral recovery with re-execution anchoring* $\gamma_t^{(\ell)} \leftarrow \gamma_{\min} + \kappa_c\, c_t^{(\ell)}$ Compute $\hat{h}_{t,\text{spec}}^{(\ell)}$ via equation 10; $\hat{h}_{t,\text{spec+re}}^{(\ell)}$ via equation 11 *Stage 4: Gated consistency fusion* Compute $\hat{h}_{t,\text{temp}}^{(\ell)}$ via equation 12; $\hat{h}_{t,\text{fused}}^{(\ell)}$ via equation 13 $\hat{h}_t^{(\ell)} \leftarrow (1 - g_t^{(\ell)})\,\tilde{h}_t^{(\ell)} + g_t^{(\ell)}\,\hat{h}_{t,\text{fused}}^{(\ell)}$ Eq. equation 9 *Per-step state update (corrected values)* Update μ_ℓ, M_ℓ by EMA on $\hat{h}_t^{(\ell)}$; update B_ℓ by RLS on $(\hat{h}_t^{(\ell-1)}, \hat{h}_t^{(\ell)})$ Update $\hat{\sigma}_\ell^{\bullet}$ by EMA-MAD on $r_{\bullet,t}^{(\ell)}$ for each $\bullet \in \{\text{spec}, \text{temp}, \text{cross}, \text{func}\}$ whose residual was computed $\hat{h}_{t-1}^{(\ell)} \leftarrow \hat{h}_t^{(\ell)}$ cache for next step *Stage 5: Fault-aware decoding* $C_t \leftarrow \frac{1}{L}\sum_{\ell=1}^{L} g_t^{(\ell)}\, c_t^{(\ell)}$; $T_t \leftarrow 1 + \kappa_T\, C_t$ $\hat{z}_t \leftarrow W_{\text{head}}\,\hat{h}_t^{(L)} + b$ $\hat{y}_t \leftarrow \arg\max_v \ \text{softmax}(\hat{z}_t / T_t)$ Eq. equation 14 $\hat{y}_{1:T}$