How a Bit Becomes a Story: Semantic Steering via Differentiable Fault Injection

Anonymous CVPR submission

Paper ID 18118

Abstract

*Hard-to-detect hardware bit flips, from either malicious circuitry or bugs, have already been shown to make transformers vulnerable in non-generative tasks. This work, for the first time, investigates how low-level, bitwise perturbations (fault injection) to the weights of a large language model (LLM) used for image captioning can influence the semantic meaning of its generated descriptions while preserving grammatical structure. While prior fault analysis methods have shown that flipping a few bits can crash classifiers or degrade accuracy, these approaches overlook the semantic and linguistic dimensions of generative systems. In image captioning models, a single flipped bit might subtly alter how visual features map to words, shifting the entire narrative an AI tells about the world. We hypothesize that such semantic drifts are not random but differentiably estimable. That is, the model's own gradients can predict which bits, if perturbed, will most strongly influence meaning while leaving syntax and fluency intact. We design a differentiable fault analysis framework, **BLADE** (**B**it-level Fau**L**t **A**nalysis via **D**ifferentiable **E**stimation), that uses gradient-based sensitivity estimation to locate semantically critical bits and then refines their selection through a caption-level semantic-fluency objective. Our goal is not merely to corrupt captions, but to understand how meaning itself is encoded, distributed, and alterable at the bit level, revealing that even imperceptible low-level changes can steer the high-level semantics of generative vision-language models. It also opens pathways for robustness testing, adversarial defense, and explainable AI, by exposing how structured bit-level faults can reshape a model's semantic output.*

1. Introduction

Transformer-based models deployed on physical accelerators (GPUs, FPGAs, ASICs) are vulnerable to hardware Trojans and Rowhammer-style faults that flip weight bits; prior work shows that even a few such flips can cause severe performance degradation. Multi-modal transformers for vision-language tasks (e.g., captioning, visual question answering, cross-modal retrieval) are similarly exposed, and existing attacks (e.g., PBS [26], AttentionBreaker [7]) typically drive models to catastrophic failure and obviously garbage outputs. However, this lack of subtlety makes such attacks easy to detect and trigger system avoidance or repair, limiting their practical impact. We instead propose **BLADE** (**B**it-level Fau**L**t **A**nalysis via **D**ifferentiable **E**stimation), a semantic-drift–prioritizing attack that steers the model toward semantically wrong outputs while preserving tight structural and syntactic integrity of the output.

BLADE exposes bit-level vulnerabilities in quantized vision-language transformers by identifying a *small* set of weight bit flips that maximizes a semantic-driven objective $\mathcal{J}(c) = d_{\text{SBERT}}(y^*, c) - \lambda \log \text{PPL}_{\text{distil}}(c)$, thereby driving *semantic drift* while preserving fluency. The attack operates on deployed int8-quantized captioning transformers, where it uses gradient-based saliency with a first-order Taylor approximation to rank individual bit flips by their impact. Candidate flips are then validated with finite differences against a joint objective that simultaneously increases semantic distance from a reference caption (via SBERT embeddings) while preserving fluency and surface form (via an external LM perplexity term), with beam re-decoding after each accepted flip to stabilize the adversarial caption. To evaluate attack quality, we introduce a GPT-based Semantic Drift Calculator (SDC). We implement BLADE as a highly parameterized framework and evaluate it across three captioning models (BLIP2-OPT-6.7B, BLIP2-OPT-2.7B, blip-image-captioning-base) and two datasets (COCO, Flickr8k). Evaluation results shows that BLADE outperforms bit-flip techniques such as PBS [26] (2.4x higher ASR), AttentionBreaker [7] (2.37x higher ASR), and Random (1.6x higher ASR) while maintaining ensuring high structural and syntactical scores. In summary we make the following contributions:

1. Introduce **BLADE**, a bit-flip attack on quantized image-captioning transformers that induces controlled semantic drift in generated descriptions while preserving grammaticality and surface structure.
2. Formulate a joint semantic-drift/fluency objective and a first-order, gradient-guided bit selection procedure with

Taylor-based sensitivity and finite-difference check.
3. Implement **BLADE** as a configurable framework and evaluate it, qualitatively and with a GPT-assisted Attack Success Rating, on multiple image-captioning models over Flickr8k and COCO.

2. Background and Related Works

Inducing Bit-flips with Physical/Hardware Attacks: Modern platforms are vulnerable to bit-level faults from both natural and adversarial sources: radiation can flip DRAM/SRAM cells [2], Rowhammer corrupts nearby addresses [15], hardware Trojans (malicious circuitry) can flip signals when triggered [6, 14, 25], and voltage/clock glitching induces deterministic bit-flips in commodity hardware [41, 42]. Neural network weights stored in memory inherit these vulnerabilities, as targeted bit corruptions can severely degrade or redirect predictions [26, 29]. In this work we do not perform physical attacks; instead, we flip bits in the quantized weights to emulate hardware faults, enabling controlled analysis of (i) robustness to weight corruption, (ii) semantic degradation pathways, and (iii) the influence of localized access on caption generation.

Threat Model: We consider a white-box threat model in which the adversary has full access to the model parameters. The attacker can inspect and directly modify the quantized representations of selected weight tensors by flipping individual bits. No data poisoning, input perturbation, or retraining is allowed. Only weight-level manipulations are performed. The adversary is constrained by (i) a *global bit budget BLADE_budget* limiting the total number of flipped bits, and (ii) a *per-element flip cap* $K_{\max}$ restricting how many bits of any single weight element can be changed. Formally, for the set of flipped bits $\mathcal{F} \subseteq \{(j,b)\}$, $\sum_{(j,b)\in\mathcal{F}} 1 \leq$ *BLADE_budget*, $\{b : (j,b) \in \mathcal{F}\}| \leq K_{\max} \quad \forall j$. This models a resource-bounded adversary who can selectively corrupt a limited subset of model bits within specified layers. While our threat model assumes white-box access and the ability to flip specific quantized bits, **this level of precision is standard in fault-attack research**. Real-world mechanisms like Rowhammer and undervolting increasingly allow attackers to bias faults toward particular memory regions, meaning our analysis provides an upper bound on the semantic damage such faults could induce. Thus, the **goal is not to claim perfect adversarial control, but to reveal where multimodal models are structurally brittle** so defenses can target these regions.

Related Work: Bit-flip attacks pose a major threat to DNN reliability and security [5, 13, 24, 35], particularly in vision models [3, 27, 40]. Early work showed that altering only a few stored weight bits can collapse performance [10]. Rakin et al. introduce Progressive Bit Search (PBS), a gradient-based bit-flip attack on quantized networks [26], while Das et al. propose AttentionBreaker, a three-phase bit-flip attack for LLMs that locates sensitive layers and uses evolutionary search to maximize loss under tight bit budgets [7]. Beyond deployment-only search, Dong et al. [9] co-train a benign model and a bit-close malicious neighbor so that a single post-deployment flip (e.g., via DeepHammer [39]) can trigger failure, using an ℓ_p-Box ADMM-inspired relaxation [37]. Wang et al. [36] propose Flip-S for quantized ViTs, targeting scale factors rather than individual weights and enforcing Rowhammer-style flip spacing [8, 11, 15, 33, 34]. Prior work has shown that Rowhammer can precisely induce single-bit flips [9, 19, 29, 30, 39], but reliably steering multiple specific bits remains challenging. Li et al. propose ONEFLIP [21], a fp32 attack that flips a single exponent bit and learns a tiny trigger to embed a robust backdoor, and SOLEFLIP [20], a one-bit backdoor for quantized models; both achieve near-100% ASR with negligible clean-accuracy loss. Guo et al. [12] introduce SBFA, a single-bit attack on BF16/INT8 LLMs that constrains perturbed weights to in-range values and ranks bits via a gradient-aware ImpactScore. Chen et al. [4] instead flip bits in the compiled executable (.text) using profiled "superbits," achieving random-guess accuracy with ≈1–2 flips even for quantized networks. For multimodal captioning, Aafaq et al. [1] propose a gray-box GAN-based attack that perturbs encoder features so decoder captions drift toward a target, modifying inputs rather than model weights. Most of the above methods do not preserve generation quality after corrupting model weights: attacks on classifiers or executables emphasize accuracy collapse [4, 9, 36], and inference-time backdoors prioritize targeted misclassification over output fluency [20, 21]. By contrast, **BLADE** is explicitly designed to corrupt behavior while preserving naturalness, enforcing fluency constraints so that text remains coherent and human-like despite induced failures.

3. BLADE Attack Design

Next, we describe the methodology of the BLADE attack.

3.1. Design Motivation

While prior work on bit-flip attacks has convincingly shown that flipping weights in quantized deep neural networks can rapidly degrade performance [7, 26], we observe a key gap: most studies focus on catastrophic failure or misclassifications, rather than controlled semantic drift while maintaining fluency or structure. Our attack is designed to realistically and precisely evaluate how small, localized changes to deployed models can produce meaningful semantic failures. The attack becomes realistic because semantic changes may avoid detection, and a small number of bit flips might suffice to shift meaning without big visible errors. This work will be helpful in providing insights for building defense and hardening the models. We have experimented with the quantized model as quantized models have become a norm in the deployment pipeline [23] and are more robust than

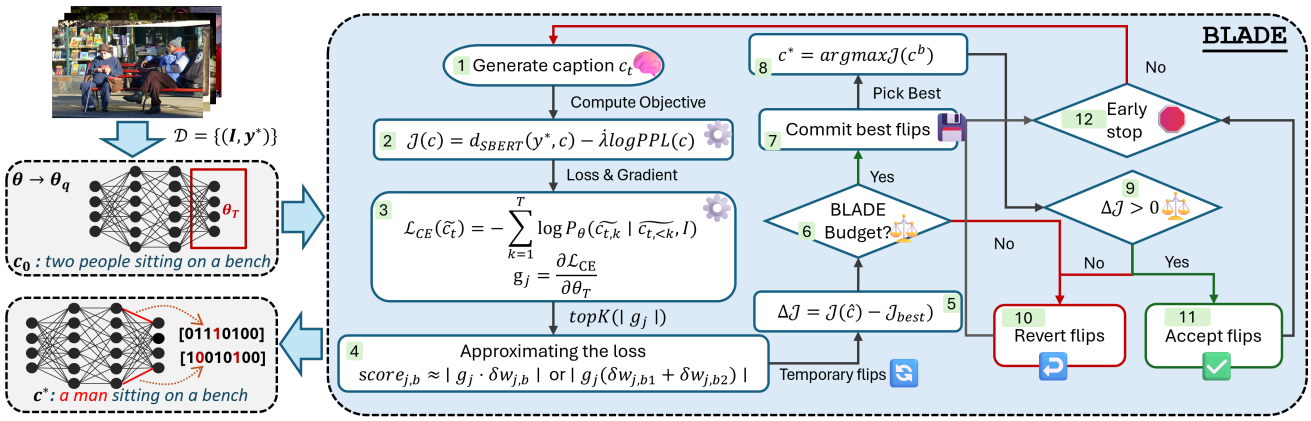

Figure 1. An high-level algorithmic overview of the BLADE methodology.

full precision models [26, 28].

3.2. Design Details

We propose the BLADE framework (see Figure 1, Algorithm 1) that exposes bit-level vulnerabilities in quantized captioning models by finding a small number of bit flips that cause semantic drift in generated captions while keeping fluency and surface structure relatively intact. For each image I, we fix a reference caption y^* from the dataset. The unperturbed captioner generates a baseline caption c_0 for I, and we denote by $\tilde{c}_t$, the current caption at optimization step t which is used as the target for teacher forcing to compute gradients, while semantic drift is always measured relative to the y^*. Given a fixed reference caption y^*, we define caption objective $\mathcal{J}$:

$$\mathcal{J}(c) = d_{\text{SBERT}}(y^*, c) - \lambda \log \text{PPL}_{\text{distil}}(c), \qquad (1)$$

Where $d_{\text{SBERT}}(y^*, c) = 1 - \cos\big(\phi(y^*), \phi(c)\big)$, with $\phi(\cdot)$ the SBERT embedding, $\text{PPL}_{\text{distil}}$ denotes external perplexity computed by an LLM, and λ weights fluency versus semantic fidelity. A larger $\mathcal{J}$ means "more drifted but still readable". Keeping y^* fixed prevents the direction of progress from wobbling across steps. The first caption generated by the unperturbed model serves as the baseline caption c_0, and its score $\mathcal{J}(c_0)$ initializes the best objective value $\mathcal{J}_{\text{best}}$ for subsequent comparison. The model is quantized to signed int8 (θ_q) prior to the attack. For each row (or tensor) of real-valued weights w, a symmetric scale s and integer representation q are computed as:

$$s = \frac{\max(|w|)}{127}, \qquad q = \text{round}(w/s), \qquad w \approx s \cdot q. \quad (2)$$

A bit flip at position b in the stored two's-complement value q_j produces an integer change $\Delta q_{j,b}$ and a perturbation:

$$\Delta w_{j,b} = s_j \cdot \Delta q_{j,b} \qquad (3)$$

After quantization, the adversary specifies a subset of target layers T to attack or can attack the entire model, forming the set of parameters $\theta_T \subseteq \theta$. Gradients and bit-level operations are restricted to θ_T, typically corresponding to the top decoder blocks and language modeling head. For the selected parameters θ_T, we compute a teacher-forced cross-entropy loss on the current caption:

$$\mathcal{L}_{\text{CE}}(\tilde{c}_t) = -\sum_{k=1}^{N} \log P_\theta(\tilde{c}_{t,k} \mid \tilde{c}_{t,<k}, I), \qquad (4)$$

where N is the number of tokens. Then compute the gradients $g_j = \partial \mathcal{L}_{\text{CE}} / \partial w_j$ for $w_j \in \theta_T$. These gradients serve as first-order sensitivity measures for each weight perturbation. Using the quantized perturbations from Eq. 3, a first-order Taylor approximation estimates the immediate change in loss by flipping the b bit of weight w over the topK gradient values as selected by the adversary:

$$\widehat{\Delta \mathcal{L}}_{j,b} \approx g_j \cdot \Delta w_{j,b} \qquad (5)$$

Each candidate flip is then ranked by its absolute score:

$$\text{score}_{j,b} = |g_j \cdot \Delta w_{j,b}| \qquad (6)$$

This first-order ranking is similar to Rakin et al.[26], who also prioritize bit flips based on gradient saliency. It is also inspired by LeCun's work [16], which motivates Taylor-based sensitivity analysis, but we restrict to the first-order term because second-order calculation would be costly and may have a negligible effect. The approximation is followed by a finite-difference (FD) evaluation to verify the true effect of each flip on the objective.

Finite-Difference Validation and Beam Evaluation : Top-ranked candidates from Eq. 6 are validated via finite differences: we temporarily apply each candidate flip (or small bundle of flips respecting K_{max}), propagate it through the quantized representation, and regenerate captions $\hat{c}$. For each trial modification, we compute the true objective $\mathcal{J}$ and record its improvement ($\Delta\mathcal{J} = \mathcal{J}(\hat{c}) - \mathcal{J}_{\text{best}}$) Only flips yielding positive $\Delta\mathcal{J}$ and satisfying the budget requirements are considered for provisional commitment.

After applying a set of candidate flips, we perform a beam re-decode producing captions $\{c^{(b)}\}_{b=1}^{B_{\text{beam}}}$ and select the best one:

$$c^\star = \arg\max_b \mathcal{J}(c^{(b)}) \tag{7}$$

If $\mathcal{J}(c^\star) > \mathcal{J}_{\text{best}}$, the flips are finalized; otherwise, they are reverted. This iterative process continues until an early-stop criterion is met i.e. we have already reached sufficiently different yet readable adversarial caption or the $BLADE\_budget$ is exhausted. Per-weight counters ensure that no element exceeds $K_{\max}$ flips.

Algorithm 1 BLADE Attack

Require: $I, y^*, \mathcal{M}_\theta, BLADE\_budget, K_{\max}, \mathcal{J}(\cdot)$
Ensure: Flipped model $\mathcal{M}_\theta$ and final caption $c^\star$

$\{Q, s\} \leftarrow$ QUANTIZE_TARGET_PARAMS$(\mathcal{M}_\theta)$
$c_0 \leftarrow$ GENERATE_CAPTION$(\mathcal{M}_\theta, I)$
$\mathcal{J}_{\text{best}} \leftarrow \mathcal{J}(c_0)$
for step $= 1 \rightarrow$ max_steps **do**
  $\tilde{c}_t \leftarrow$ GENERATE_CAPTION$(\mathcal{M}_\theta, I)$
  $\mathcal{L}_{\text{CE}} \leftarrow$ TEACHER_FORCING_LOSS$(M_\theta, I, \tilde{c}_t)$
  $g_j \leftarrow$ GRADIENTS_ON_TARGETS(M_θ)
  $\text{score}_{j,b} \leftarrow$ TAYLOR_SCORE$(g_j, \Delta w_{j,b})$
  $\mathcal{C} \leftarrow$ SELECT_CANDS$(\text{score}, BLADE_budget)$
  $\mathcal{C}_{\text{FD}} \leftarrow$ FD_VALIDATE$(\mathcal{C}, \mathcal{J})$
  APPLY_INPLACE_FLIPS$(\mathcal{M}_\theta, \mathcal{C}_{\text{FD}})$
  $\{c^{(b)}\} \leftarrow$ BEAM_DECODE$(\mathcal{M}_\theta, I)$
  $c^\star \leftarrow \arg\max_b \mathcal{J}(c^{(b)})$
  if $\mathcal{J}(c^\star) > \mathcal{J}_{best}$ **then** $\mathcal{J}_{\text{best}} \leftarrow \mathcal{J}(c^\star)$
  else $revert\_flips(\mathcal{M}_\theta, \mathcal{C}_{\text{FD}})$
  end if
end for
return $(\mathcal{M}_\theta, c^\star)$

3.3. Attack Success Rating

We introduce the GPT model-assisted Semantic Drift Calculator (SDC) that quantifies *semantic drift* between image-caption pairs under adversarial perturbations. Note that BLADE internally optimizes $\mathcal{J}(c)$ relative to the dataset reference caption y^*, while in the SDC, given an image I, base caption c_0, and an adversarial caption c^*, the SDC measures how far c^* semantically diverges from the visual truth in I while remaining grammatically and syntactically valid with respect to c_0. For each triplet, the GPT based judge first produces a single-sentence neutral description c_{img} that summarizes the visible scene and then assigns scalar scores in $[0, 100]$ that reflect how faithfully c_0 and c^* describe I, how closely c^* preserves the surface form of c_0, and how strongly c^* misdirects semantics with respect to the image. Faithfulness scores F_0 and F^* measure alignment of c_0 and c^* with c_{img}; structure preservation SP measures the similarity of surface form between c_0 and c^*; semantic misdirection M measures the degree to which c^* contradicts or misleadingly alters core objects, attributes, relations, or negations *relative to the image* summarized by c_{img}; subtlety Sb reflects how hard the edit is to notice at a glance; and risk R reflects potential real-world harm if the adversarial statement were believed. The syntax quality S is scored separately on $[0, 100]$ to see if the c^* is well syntactically correct and is used for gating.

Table 1. Experiment hyperparameters and ASR scoring.

BLADE Configuration	
Global bit budget	$BLADE\_budget = 20$
Per-weight limit	$K_{\max} = 2$
Top-K gradient weights	5000
FD validation candidates	100
Beam size	$B_{\text{beam}} = 3$
Max steps per image	10
Objective weight	$\lambda = 0.005$ (`lambda_ppl=0.005`)
Early stop thresholds	$\tau_{sbert} = 0.4$, $\tau_{distillppl}$=300.0
Random seed	42
ASR Scoring Configuration	
Judge model	GPT-4o-mini (image+text), temperature $= 0$
Structure threshold	$\tau_{SP} = 50$
Syntax minimum	$S_{\min} = 70$
No-op rule	$c^* = c_0 \Rightarrow \text{ASR} = 0$

In our implementation, ASR is a weighted combination of (i) how much faithfulness drops from F_0 to F^*, (ii) the image-centric misdirection M, and (iii) a subtlety term that is upweighted by the judged risk R, so that high-risk but barely noticeable edits are scored more harshly. $\text{ASR} = 0.5\max(0, F_0 - F^*) + 0.3\,M + 0.2\,S_w$, where $S_w = (Sb(50 + R)/100)$, so that an attack is rewarded only when it moves away from visual truth (decreasing faithfulness) and increases image-centric misdirection while remaining subtle. Attacks that clearly move captions *toward* the visual truth (substantial increase in faithfulness) therefore receive little or no credit, while edits that move *away* from c_{img} and increase misdirection can obtain a high score provided they remain subtle and well formed. The image-centric role of c_{img} is crucial. Misdirection is scored strictly with respect to what is *visually* present. Consequently, subtle edits that move c^* *toward* c_{img} (for example, adding "brown" when the image shows two brown bears) reduce misdirection and are judged as failed attacks, whereas equally small edits that move c^* *away* from c_{img} (for example, changing species, attributes, or relations that are not supported by the image) increase misdirection and can yield a high ASR provided the structure and syntax gates are satisfied. We used two layers of gating. First, a syntax–structure gate suppresses the score whenever the adversarial caption becomes ungrammatical U, structurally broken S, or fails basic well-formedness requirements. Second, a set of semantic sanity gates zero the score whenever the caption exhibits lexical nonsense N, selectional anomalies A, referential failures R, or internal contradictions C.

Table 2. Comparing BLADE with other fault injection techniques for the Flickr8k dataset (Target Model = BLIP2-OPT-2.7B).

	Metrics	F1	F2	F3	F4	F5	F10	F20	F40	F60	F80	F100
Random	**M**	12.00	13.50	14.35	12.60	13.40	13.45	14.25	13.55	12.60	11.55	11.80
	F*	83.15	80.80	81.10	82.00	82.55	80.55	80.75	81.25	82.20	82.25	81.90
	SP	95.00	94.75	94.50	95.00	95.25	95.00	95.00	94.50	94.75	95.25	94.25
	Sb	92.25	92.10	90.55	92.45	92.85	92.15	92.25	91.85	91.90	92.20	92.05
	R	18.60	18.10	18.50	18.50	17.40	16.50	18.30	17.30	18.10	17.40	16.60
	S	97.30	97.40	97.40	97.35	97.45	96.95	97.40	97.20	97.10	97.45	97.55
	ASR	10.79	10.93	10.6	10.63	11.06	10.66	11.07	10.81	10.91	10.77	10.3
PBS[26]	**M**	55.90	48.35	76.70	74.60	79.00	66.50	62.20	61.45	61.95	59.00	63.85
	F*	42.05	48.15	22.25	24.10	19.20	29.20	31.80	31.50	31.15	33.50	33.00
	SP	49.50	50.75	25.10	29.00	21.75	34.50	38.25	36.85	35.35	37.75	37.00
	Sb	47.25	48.15	23.65	27.60	20.95	33.05	35.15	35.10	32.95	36.10	35.10
	R	21.60	25.60	26.20	23.40	23.00	27.10	27.60	25.60	27.40	25.80	27.20
	S	91.40	91.65	91.80	97.25	88.20	92.15	94.30	96.20	94.15	96.00	95.75
	ASR	6.99	8.08	4.20	6.56	2.96	6.86	9.43	8.13	8.16	8.51	7.74
Attention Breaker[7]	**M**	12.60	11.45	8.50	11.35	10.80	10.00	9.75	13.30	10.45	11.35	10.00
	F*	77.65	79.40	79.95	79.45	79.10	81.00	80.60	79.60	81.15	77.95	80.25
	SP	95.15	96.00	97.50	96.40	97.25	96.75	98.00	95.50	98.25	95.65	96.25
	Sb	92.65	95.10	96.25	94.25	96.20	95.80	95.90	95.25	95.75	93.85	95.20
	R	17.20	16.20	14.50	15.70	14.60	14.80	15.40	15.60	16.20	16.60	15.90
	S	98.00	98.35	98.60	98.25	98.45	98.45	98.55	98.05	98.75	98.45	98.80
	ASR	7.33	4.89	4.28	5.18	4.51	4.61	6.73	5.60	6.41	5.13	4.49
BLADE (Ours)	**M**	18.35	16.60	18.35	17.80	18.30	18.80	18.45	18.20	16.15	16.60	17.20
	F*	76.25	77.95	77.10	78.10	77.75	76.15	77.70	77.35	78.85	78.30	78.35
	SP	93.00	93.50	95.25	95.40	93.90	93.25	92.40	92.50	92.75	94.50	93.25
	Sb	87.75	88.65	88.30	88.60	87.90	87.30	86.40	87.50	87.75	88.15	88.10
	R	17.30	16.40	15.90	17.40	16.70	19.20	17.70	17.50	17.40	16.60	16.10
	S	96.20	96.25	96.95	96.65	96.30	95.70	96.10	96.30	96.05	97.50	96.70
	ASR	**17.38**	**16.88**	**18.07**	**17.31**	**18.12**	**19.02**	**17.89**	**16.88**	**14.96**	**15.59**	**15.68**

These checks ensure that the attack does not exploit obviously invalid or meaningless edits. In addition, exact non-edits $c_0 == c^*$ are also treated as automatic failures. Only captions that pass all gates are eligible for a non-zero ASR. Finally, we report corpus-level effectiveness as the mean ASR across the evaluation set, $\overline{\text{ASR}} = \frac{1}{N}\sum_{i=1}^{N} \text{ASR}_i$.

4. Experimental Results

All results are obtained using NVIDIA A100 GPU. More results and source code are in the supplementary materials.

4.1. Experimental Setup

Datasets : For each image, we restore a fresh unperturbed checkpoint, produce a baseline caption, run the iterative BLADE bit-flip search with a fixed perturbation budget, and log all intermediate states. Experiments are conducted on two standard captioning benchmarks: Flickr8k (via `jxie/flickr8k` [38]) and COCO Caption 2017 (via `lmms-lab/COCO-Caption2017` [22]). From each dataset, we select the top 100 test images (Flickr8k) or top 100 validation images (COCO). For Flickr8k we take $caption0$ as the reference y^*, and for COCO we use the first annotated answer (answer[0]) as y^*.

Models and Scorers : We evaluate BLADE against models of increasing scale: 223M params `blip-image-captioning-base` ($\mathcal{M}_1$) [17], 2.7B params `blip2-opt-2.7b` ($\mathcal{M}_2$) and 6.7B params `blip2-opt-6.7b` ($\mathcal{M}_3$) [18]. Semantic similarity is computed using the sentence-transformer `all-MiniLM-L6-v2` [31] Sentence-BERT encoder, while `DistilGPT2` [32] provides an external perplexity measure for fluency control. We additionally report the internal model perplexity of the captioner for completeness. The hyperparameters are provided in Table 1.

Runtime Analysis: On average for BLIP2-opt-6.7b, runtime per image (sec) was: {**BLADE**: 321, **Random**: 6, **AttnB**: 832, **PBS**: 242}. GPT-Scoring takes about 4.8 sec/img.

Evaluation Metrics : For each attacked sample, we record baseline and final values of $\mathcal{J}$, d_{SBERT}, $\text{PPL}_{\text{distil}}$, internal BLIP perplexity, and standard captioning metrics (BLEU, ROUGE-L, CIDEr, METEOR). We also log the number and locations of flipped bits, per-step $\Delta\mathcal{J}$, total bits used, and average runtime. Based on these score the early stopping condition are met if $d_{\text{SBERT}} \geq \tau_{sbert}$ and $\text{PPL}_{\text{distil}} \leq \tau_{distillppl}$. Attack performance is reported using the Attack Success Rate (ASR), computed via our GPT-4o-mini based Semantic Drift Calculator (SDC). The SDC outputs the components M (misdirection), SP (structure preservation), Sb (subtlety), R (risk), and S (syntax), which together determine the final attack success rate as ASR.

Scoring ASR : All evaluations were performed using the SDC introduced in Section 3.3. This scoring framework was applied uniformly across all adversarial experiments, and its final *ASR* value is reported. For every image–caption pair, we used the `GPT-4o-mini` API (image–text mode, temperature 0) to obtain both the neutral image summary c_{img} and the scalar judgments $(F_0, F^*, SP, M, Sb, R, S)$. Table 1 summarizes the optimization and scoring hyperparameters. The lower block lists the fixed constants used for ASR computation, including the threshold values, guard conditions, and weighting coefficients.

Table 3. Comparing BLADE with other fault injection techniques for the Coco dataset (Target Model = BLIP2-OPT-6.7B).

	Metrics	F1	F2	F3	F4	F5	F10	F20	F40	F60	F80	F100
Random	M	13.75	13.25	13.25	12.90	13.00	13.40	13.10	12.45	14.45	14.70	13.85
	F*	84.10	84.20	84.75	84.80	84.30	84.10	84.00	84.55	83.60	83.65	84.35
	SP	93.00	93.00	91.55	93.25	92.15	92.50	92.40	92.15	92.25	91.75	91.90
	Sb	89.40	89.35	89.20	90.15	89.00	89.55	89.25	89.50	90.15	88.90	89.30
	R	15.90	15.40	14.80	14.70	14.60	14.50	15.40	14.80	15.40	14.60	15.40
	S	96.25	96.30	96.05	96.55	96.20	96.20	95.90	96.15	95.90	95.75	95.95
	ASR	13.16	12.14	12.68	12.85	13.14	12.15	13.20	12.71	13.69	12.47	13.32
PBS[26]	M	50.04	57.18	73.15	74.45	77.85	79.60	78.40	77.90	79.05	77.75	77.90
	F*	44.90	37.75	25.10	23.65	20.65	18.50	19.80	20.10	19.40	20.65	20.75
	SP	53.75	45.85	33.85	31.90	27.00	24.75	26.10	25.85	24.85	26.20	26.60
	Sb	52.60	42.95	31.50	29.70	25.30	23.55	23.70	24.10	23.05	23.85	24.90
	R	19.10	21.30	20.50	22.40	21.00	25.20	19.00	21.00	21.20	24.80	21.60
	S	92.60	94.70	92.60	94.05	93.65	90.15	90.15	90.35	87.75	90.15	89.80
	ASR	14.97	13.91	12.64	13.13	10.33	8.75	8.74	7.93	8.29	8.37	9.24
Attention Breaker[7]	M	6.90	4.45	5.00	7.00	6.45	6.85	4.20	7.80	4.15	5.95	5.95
	F*	86.60	86.50	89.10	87.65	88.60	86.35	90.35	87.85	89.20	84.95	90.05
	SP	97.75	97.50	97.75	97.50	97.50	97.50	98.75	97.40	98.00	98.50	97.25
	Sb	96.65	96.35	95.85	95.05	96.65	95.75	96.85	96.25	96.20	97.65	95.15
	R	12.70	12.40	12.20	12.40	12.70	12.10	12.10	12.40	12.00	12.60	12.00
	S	99.00	99.10	98.85	98.55	98.45	98.45	99.20	98.45	99.10	99.45	98.20
	ASR	3.52	3.84	5.43	6.17	5.04	4.90	3.67	4.33	4.31	4.11	5.05
BLADE (Ours)	M	13.75	13.65	13.50	14.80	14.30	15.00	15.30	14.30	17.05	16.90	16.10
	F*	84.50	84.25	84.65	84.25	83.55	82.55	82.45	84.55	80.95	81.25	82.45
	SP	93.00	93.65	92.65	92.75	93.00	92.40	92.50	92.75	92.25	91.50	91.55
	Sb	88.25	88.30	88.00	88.05	87.70	86.50	87.90	88.50	86.10	85.90	85.90
	R	14.50	13.60	15.10	14.50	15.10	16.00	15.40	15.50	16.10	17.40	16.40
	S	96.60	96.90	96.60	96.20	96.15	96.10	96.55	96.20	95.90	95.65	95.65
	ASR	**17.43**	**17.37**	**17.37**	**17.64**	**18.04**	**18.83**	**19.21**	**16.92**	**19.97**	**20.03**	**19.09**

4.2. Adapting Baseline Methods for Fairness

To compare BLADE with AttentionBreaker [7], we reuse its three-stage pipeline (layer ranking, candidate selection, evolutionary search) but optimize over cross-entropy loss. Layers are ranked by hybrid sensitivity (gradient–weight product). For candidate sets, we progressively sample the top 0.001%, 0.01%, 0.1%, and 1% most sensitive weights, flipping bits until the per-sample loss increases by 0.5. We then run evolutionary optimization over subsets, scoring by loss increase per flip, applying mutation and crossover for 50 generations, and selecting the best solution under the same flip budget and batch size (1) as BLADE. For PBS and Random [26], we adopt the original hyperparameter ($top_k = 10$) but use the same cross-entropy loss, batch size (1), and flip budget as BLADE for fairness.

4.3. Analyzing the Core Results

The Table 2 & 3 compares the BLADE against Random, Progressive Bit Search (PBS), and AttentionBreaker (AttnB). Evaluation of all the methods was carried out under identical flip budgets. Table attributes Fk represent flipping exactly k bits in the quantized model weights (e.g., $F1$ is a single–bit flip, $F2$ flips two bits, and so on). Table 2 reports the results of attacking $\mathcal{M}_2$ on 100 Flickr8k test images and the Table 3 reports the results of attacking $\mathcal{M}_3$ on 100 coco validation images, where performance is averaged under bit-flip budgets Fk with $k \in \{1, 2, 3, 4, 5, 10, 20, 40, 60, 80, 100\}$. All results reported in the paper are obtained by applying the bit-flip attack exclusively to the weights of the final two cross-attention layers of the model. Table 2, 3, and Fig. 2 shows a similar pattern, confirming that the trends are consistent across models/datasets. Hence we expect BLADE to generalize well across settings (more details in the supplementary doc).

Table 4. Percent of cases where $c_0 == c^*$

Models	Dataset	Random	PBS	AttnB	BLADE
Blip2-opt-6.7b	flickr8k	68.55	10.18	76.18	31.36
	coco	48.00	9.36	76.09	18.36
Blip2-opt-2.7b	flickr8k	53.00	9.27	76.45	26.55
	coco	54.09	5.91	72.91	34.82
Blip-image-captioning-base	flickr8k	43.27	4.64	86.36	33.73
	coco	31.00	3.18	81.91	34.18

4.3.1. Cases with no Semantic Shift

Here we consider the cases in which the adversarial caption is the same as the base caption (i.e., $c^* = c_0$). Random flips preserve the baseline caption in **50%** of cases; AttentionBreaker in **78%** of cases (highly conservative); PBS in only **7%** (highly destructive); and BLADE in about **30%** cases (balanced) (see Table 4). This demonstrates the expected qualitative behavior of PBS, where it changes captions aggressively but often produces visually detectable degenerate statements; Random flips and AttentionBreaker does not consistently alter statements. BLADE alters the statements iff a change satisfies the semantic drift objective. Some of the apparent "no-change" cases arise not because BLADE fails, but because the c_0 itself already satisfies the objective, and the algorithm is designed to avoid unnecessary perturbations when caption is already semantically misleading.

4.3.2. Deeper Dive into Specific Metrics

Figure 2 visualizes key evaluation dimensions, semantic misdirection M, attack success rate ASR, and structure preservation SP. These plots offer a clearer depiction

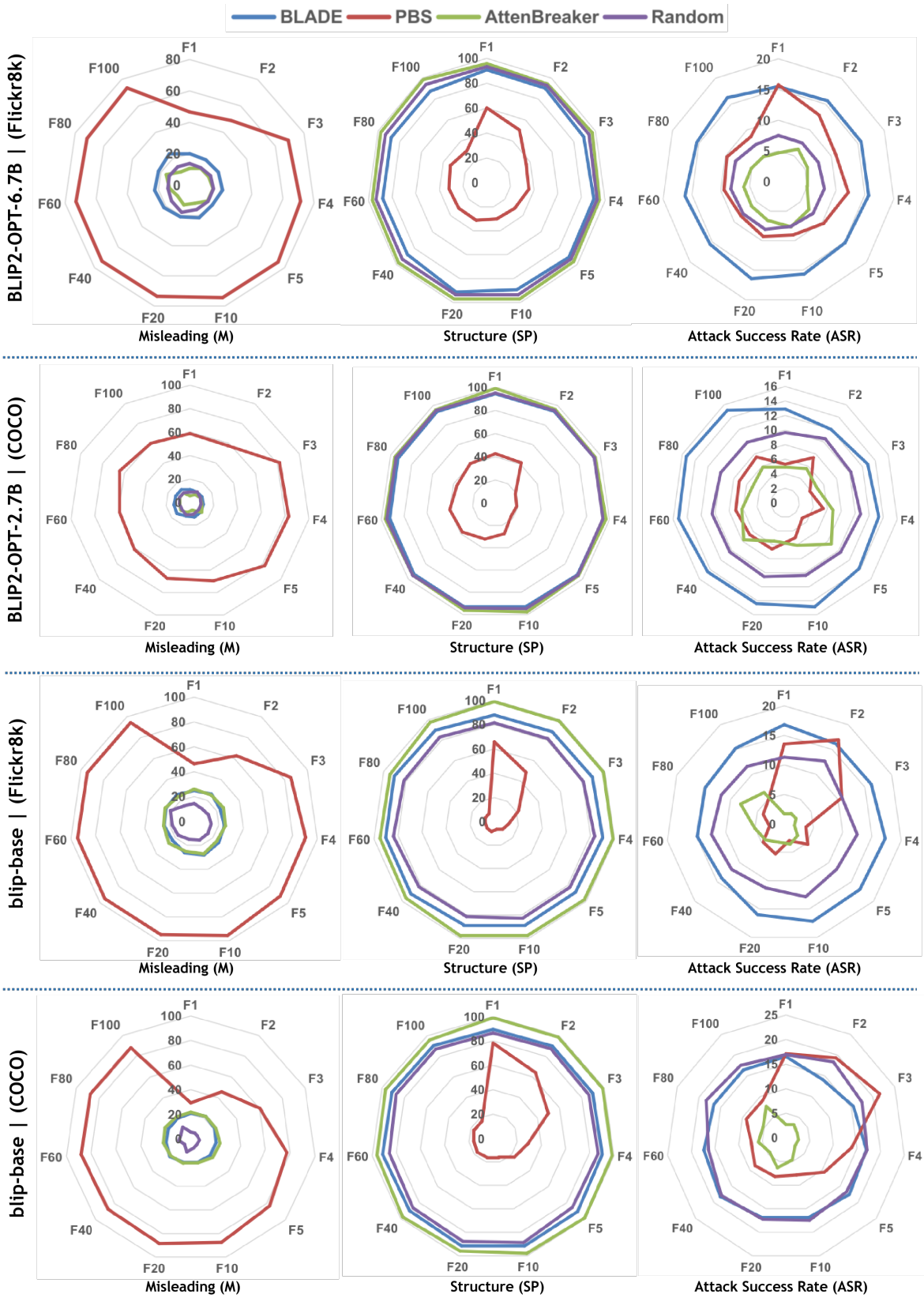

Figure 2. Visualized additional results for different dataset/model combinations.

Figure 3. Qualitative bit-flip results for BLADE and other techniques.

of how each attack regime behaves across bit budgets. BLADE sets a clear target: *maximize semantic drift while maintaining structural and linguistic validity*. ASR values depict that BLADE performs the best. Although other methods may exceed BLADE in isolated metrics, they do not align with the adversarial objective we pursue.

Overall ASR: BLADE achieves the **highest ASR across all flip budgets**, demonstrating the most effective combination of semantic drift and linguistic consistency. AttentionBreaker shows the lowest ASR, indicating that it does not reliably induce semantic misalignment. Random and PBS yield intermediate ASR values.

Structural Preservation SP and Syntax Quality S: Unlike PBS, whose SP values *rapidly decay* due to severe sentence collapse, BLADE maintains **stable and consistently high** SP across all flip budgets. Together with uniformly strong syntax quality S, this indicates that BLADE preserves similarity to c_0 (base caption) while having grammatically correct statement. Higher SP values are desirable; however, an SP score of 100 implies no semantic drift, which consequently yields zero ASR.

Misleadingness M: PBS achieves the **highest misleadingness**, but its captions often contain repetitive or nonsensical patterns, which makes them trivially detectable. Therefore, metrics like subtlety (Sb) and risk (R) are used to prevent degenerate high M captions from being counted as successful attacks. BLADE consistently yields the **second-highest misleadingness** with high S, producing meaningfully different c^* from the visual content. Random and AttentionBreaker have lower M values.

Adverserial Faithfulness F^*: Lower values are desirable because they indicate distance from ground-truth semantics. PBS achieves the lowest F^* (expected from destructive behavior). BLADE ranks **third-lowest**, indicating some semantic deviation, though not as aggressively as PBS or, in some cases, as Random. However, this moderate deviation in F^* is desirable over overly dramatic deviations. Additionally, whenever F^* exceeds F_0, the faithfulness term contributes zero to the ASR, ensuring that captions that become *more* faithful to the image are not rewarded.

4.3.3. Qualitative Analysis

Figure 3 illustrates that the BLADE produces small, grammatically valid edits that subtly shift the semantics away from the visual truth, while competing methods either fail to change the caption or collapse into incoherent text or sometimes provide a better explanation of the image. In the **first example**, the baseline correctly identifies "two people sitting on a bench," whereas BLADE changes it to "a man sitting on a bench.", preserving the fluency and structure while injecting a misleading singular person narrative. In contrast, PBS produces long, noisy outputs, and both Random and AttentionBreaker fail to induce any semantic drift. In the **second example**, BLADE changes "wall" to a plausible but incorrect "ledge", and in the **third example** it inserts "in his lap" to the caption changing the meaning. Across all three, competing methods either leave the caption unchanged or drift into obviously irrelevant text, whereas BLADE consistently achieves subtle, image-inconsistent misdirection.

5. Conclusion

BLADE exposes bit-level vulnerabilities in quantized captioners by searching for a *small* set of weight bit flips that maximizes a caption-level objective $\mathcal{J}(c) = d_{\mathrm{SBERT}}(y^*, c) - \lambda \log \mathrm{PPL}_{\mathrm{distil}}(c)$, thereby driving *semantic drift* while preserving fluency. By showing how tiny **hardware faults can shift caption** meaning and by developing test toolkits, this work will **help industry harden vision-language understanding services and edge deployments** on custom ASIC (e.g. Trainium, Inferentia) and IoT devices to **improve reliability, safety, and user trust**. This work showcases how semantic drift is differentiably estimable and model gradients can identify which individual bits, if flipped, most strongly alter semantics under fluency constraints. Our experimental results show that BLADE outperforms traditional bit-flip attacks (e.g. AttentionBreaker, PBS) for semantic drift objectives, creating a pathway for a significant amount of future research.

References

[1] Nayyer Aafaq, Naveed Akhtar, Wei Liu, Mubarak Shah, and Ajmal Mian. Language model agnostic gray-box adversarial attack on image captioning. *IEEE Transactions on Information Forensics and Security*, 18:626–638, 2022. 2

[2] R. Baumann. Soft errors in advanced computer systems. *Design & Test of Computers, IEEE*, 22:258 – 266, 2005. 2

[3] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Proflip: Targeted trojan attack with progressive bit flips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7718–7727, 2021. 2

[4] Yanzuo Chen, Zhibo Liu, Yuanyuan Yuan, Sihang Hu, Tianxiang Li, and Shuai Wang. Compiled models, built-in exploits: Uncovering pervasive bit-flip attack surfaces in dnn executables. *arXiv preprint arXiv:2309.06223*, 2023. 2

[5] Yanzuo Chen, Yuanyuan Yuan, Zhibo Liu, Sihang Hu, Tianxiang Li, and Shuai Wang. Bitshield: Defending against bit-flip attacks on dnn executables. *computing*, 2:47, 2025. 2

[6] Jonathan Cruz, Pravin Gaikwad, Abhishek Anil Nair, Prabuddha Chakraborty, and Swarup Bhunia. A machine learning based automatic hardware trojan attack space exploration and benchmarking framework. In *AsianHOST*. IEEE, 2022. 2

[7] Sanjay Das, Swastik Bhattacharya, Souvik Kundu, Shamik Kundu, Anand Menon, Arnab Raha, and Kanad Basu. Genbfa: An evolutionary optimization approach to bit-flip attacks on llms. *arXiv preprint arXiv:2411.13757*, 2024. 1, 2, 5, 6

[8] Finn De Ridder, Pietro Frigo, Emanuele Vannacci, Herbert Bos, Cristiano Giuffrida, and Kaveh Razavi. Smash: Synchronized many-sided rowhammer attacks from javascript. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1001–1018, 2021. 2

[9] Jianshuo Dong, Han Qiu, Yiming Li, Tianwei Zhang, Yuanjie Li, Zeqi Lai, Chao Zhang, and Shu-Tao Xia. One-bit flip is all you need: When bit-flip attack meets model training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4688–4698, 2023. 2

[10] Ido Galil, Moshe Kimhi, and Ran El-Yaniv. No data, no optimization: A lightweight method to disrupt neural networks with sign-flips. *arXiv preprint arXiv:2502.07408*, 2025. 2

[11] Daniel Gruss, Moritz Lipp, Michael Schwarz, Daniel Genkin, Jonas Juffinger, Sioli O'Connell, Wolfgang Schoechl, and Yuval Yarom. Another flip in the wall of rowhammer defenses. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 245–261. IEEE, 2018. 2

[12] Jingkai Guo, Chaitali Chakrabarti, and Deliang Fan. Sbfa: Single sneaky bit flip attack to break large language models. *arXiv preprint arXiv:2509.21843*, 2025. 2

[13] Zhezhi He, Adnan Siraj Rakin, Jingtao Li, Chaitali Chakrabarti, and Deliang Fan. Defending and harnessing the bit-flip based adversarial weight attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14095–14103, 2020. 2

[14] Tamzidul Hoque, Jonathan Cruz, Prabuddha Chakraborty, and Swarup Bhunia. Hardware ip trust validation: Learn (the untrustworthy), and verify. In *2018 IEEE International Test Conference (ITC)*, pages 1–10. IEEE, 2018. 2

[15] Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu. Flipping bits in memory without accessing them: An experimental study of dram disturbance errors. *ACM SIGARCH Computer Architecture News*, 42(3):361–372, 2014. 2

[16] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989. 3

[17] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. 5

[18] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 19730–19742, 2023. 5

[19] Shaofeng Li, Xinyu Wang, Minhui Xue, Haojin Zhu, Zhi Zhang, Yansong Gao, Wen Wu, and Xuemin Sherman Shen. Yes,one-bit-flip matters! universal dnn model inference depletion with runtime code fault injection. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 1315–1330, 2024. 2

[20] Xiang Li, Lannan Luo, and Qiang Zeng. Backdoor attacks on neural networks via one-bit flip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4328–4338, 2025. 2

[21] Xiang Li, Ying Meng, Junming Chen, Lannan Luo, and Qiang Zeng. Rowhammer-based trojan injection: One bit flip is sufficient for backdooring dnns. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 6319–6337, 2025. 2

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 5

[23] Kai Liu, Qian Zheng, Kaiwen Tao, Zhiteng Li, Haotong Qin, Wenbo Li, Yong Guo, Xianglong Liu, Linghe Kong, Guihai Chen, Yulun Zhang, and Xiaokang Yang. Low-bit model quantization for deep neural networks: A survey, 2025. 2

[24] Qi Liu, Jieming Yin, Wujie Wen, Chengmo Yang, and Shi Sha. Neuropots: Realtime proactive defense against bit-flip attacks in neural networks. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 6347–6364, 2023. 2

[25] Tanzim Mahfuz, Pravin Gaikwad, Tasneem Suha, Swarup Bhunia, and Prabuddha Chakraborty. Salty: Explainable artificial intelligence guided structural analysis for hardware trojan detection. In *2025 IEEE 43rd VLSI Test Symposium (VTS)*, pages 1–7. IEEE, 2025. 2

[26] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. Bit-flip attack: Crushing neural network with progressive bit search. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1211–1220, 2019. 1, 2, 3, 5, 6

[27] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. Tbt: Targeted neural network attack with bit trojan. In *Proceedings*

of the IEEE/CVF conference on computer vision and pattern recognition, pages 13198–13207, 2020. 2

[28] Adnan Siraj Rakin, Zhezhi He, Jingtao Li, Fan Yao, Chaitali Chakrabarti, and Deliang Fan. T-bfa: Targeted bit-flip adversarial weight attack. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7928–7939, 2021. 3

[29] Adnan Siraj Rakin, Yukui Luo, Xiaolin Xu, and Deliang Fan. Deep-dup: An adversarial weight duplication attack framework to crush deep neural network in multi-tenant fpga. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1919–1936, 2021. 2

[30] Adnan Siraj Rakin, Md Hafizul Islam Chowdhuryy, Fan Yao, and Deliang Fan. Deepsteal: Advanced model extractions leveraging efficient weight stealing in memories. In *2022 IEEE symposium on security and privacy (SP)*, pages 1157–1174. IEEE, 2022. 2

[31] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019. 5

[32] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. 5

[33] Andrei Tatar, Radhesh Krishnan Konoth, Elias Athanasopoulos, Cristiano Giuffrida, Herbert Bos, and Kaveh Razavi. Throwhammer: Rowhammer attacks over the network and defenses. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pages 213–226, 2018. 2

[34] Victor Van Der Veen, Yanick Fratantonio, Martina Lindorfer, Daniel Gruss, Clémentine Maurice, Giovanni Vigna, Herbert Bos, Kaveh Razavi, and Cristiano Giuffrida. Drammer: Deterministic rowhammer attacks on mobile platforms. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1675–1689, 2016. 2

[35] Jialai Wang, Ziyuan Zhang, Meiqi Wang, Han Qiu, Tianwei Zhang, Qi Li, Zongpeng Li, Tao Wei, and Chao Zhang. Aegis: Mitigating targeted bit-flip attacks against deep neural networks. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2329–2346, 2023. 2

[36] Jialai Wang, Yuxiao Wu, Weiye Xu, Yating Huang, Chao Zhang, Zongpeng Li, Mingwei Xu, and Zhenkai Liang. Your scale factors are my weapon: Targeted bit-flip attacks on vision transformers via scale factor manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20103–20112, 2025. 2

[37] Baoyuan Wu and Bernard Ghanem. ℓ_p-box admm: A versatile framework for integer programming. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1695–1708, 2018. 2

[38] Jie Xu. Flickr8k dataset. https://huggingface.co/datasets/jxie/flickr8k, 2023. Accessed: 2025-11-13. 5

[39] Fan Yao, Adnan Siraj Rakin, and Deliang Fan. Deephammer: Depleting the intelligence of deep neural networks through targeted chain of bit flips. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1463–1480, 2020. 2

[40] Xuan Zhou, Souvik Kundu, Dake Chen, Jie Huang, and Peter Beerel. What makes vision transformers robust towards bit-flip attack? In *International Conference on Pattern Recognition*, pages 424–438. Springer, 2024. 2

[41] Loic Zussa, Amine Dehbaoui, Karim Tobich, Jean-Max Dutertre, Philippe Maurine, Ludovic Guillaume-Sage, Jessy Clediere, and Assia Tria. Efficiency of a glitch detector against electromagnetic fault injection. In *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6. IEEE, 2014. 2

[42] Loïc Zussa, Jean-Max Dutertre, Jessy Clédière, and Bruno Robisson. Analysis of the fault injection mechanism related to negative and positive power supply glitches using an on-chip voltmeter. In *2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 130–135. IEEE, 2014. 2