TT-SEAL: TTD-Aware Selective Encryption for Adversarially-Robust and Low-Latency Edge AI

Kyeongpil Min[†], Sangmin Jeon[†], Jae-Jin Lee[‡], and Woojoo Lee[†]
[†]Department of Intelligent Semiconductor Engineering, Chung-Ang University, Seoul, Korea
[‡] AI Edge SoC Research Section, Electronics and Telecommunications Research Institute, Daejeon, Korea

Abstract

Cloud–edge AI must jointly satisfy model compression and security under tight device budgets. While Tensor-Train Decomposition (TTD) shrinks on-device models, prior selective-encryption studies largely assume dense weights, leaving its practicality under TTD compression unclear. We present *TT-SEAL*, a selective-encryption framework for TT-decomposed networks. TT-SEAL ranks TT cores with a sensitivity-based importance metric, calibrates a one-time robustness threshold, and uses a value-DP optimizer to encrypt the minimum set of critical cores with AES. Under TTD-aware, transfer-based threat models—and on an FPGA-prototyped edge processor—TT-SEAL matches the robustness of full (black-box) encryption while encrypting as little as 4.89–15.92% of parameters across ResNet-18, MobileNetV2, and VGG-16, and drives the share of AES decryption in end-to-end latency to low single digits (e.g., 58% → 2.76% on ResNet-18), enabling secure, low-latency edge AI.

Keywords

model compression, tensor-train decomposition, selective encryption, adversarial robustness, edge AI, FPGA prototyping

1 Introduction

Cloud-edge collaborative AI leverages cloud resources to train large-scale neural networks while enabling low-latency inference on edge devices [1, 2]. This architecture has accelerated the deployment of intelligent services such as speech recognition, video analytics, and IoT applications by flexibly distributing workloads between cloud and edge. However, frequent model updates require transmitting large parameter sets, and edge devices—operating under tight storage, computation, and energy budgets—face persistent challenges in efficiently storing and synchronizing models [3, 4].

To address these challenges, numerous model compression techniques have been studied, including quantization [5], pruning [6], knowledge distillation [7], and low-rank or tensor factorization [8]. Among these, Tensor-Train Decomposition (TTD) [9] has shown particular promise by decomposing a high-dimensional weight tensor into a sequence of low-dimensional cores. TTD preserves the structural properties of layers while reducing redundancy, yielding models with much lower parameter counts and storage complexity. As a result, it reduces model size, communication volume, and memory-access cost during inference, thereby enabling efficient deployment under edge constraints [10–12].

Beyond efficiency, however, cloud–edge collaborative AI raises significant *security* concerns. Unlike conventional cloud-only models where parameters remain protected in datacenters, distributed deployment requires storing parameters in off-chip DRAM and transmitting them over external buses on edge devices, exposing them to physical attacks [13, 14]. Compromised weights can lead to IP theft, reverse engineering [15], model cloning [16], and adversarial attacks [17, 18], ultimately threatening the reliability of IoT devices [19, 20] and autonomous systems [21]. Notably, when applying adversarial transferability attacks [22, 23] to TTD-compressed models, we observe that despite reduced representational capacity and constrained decision boundaries, substitute models can still approximate the original model's behavior closely. This finding indicates that TTD compression alone does not mitigate adversarial transferability risks, and adversarial threats remain a pressing security concern.

A straightforward defense is to encrypt all model parameters before storing or transmitting them. While full encryption provides strong protection, it imposes significant latency and energy overhead because every inference requires heavy decryption at the edge. To reduce this overhead, *selective encryption* has been explored. Instead of protecting all weights, selective encryption targets only a subset of parameters, approximating the robustness of full encryption at a fraction of the cost [24–26]. Examples include L1-norm–based selection of convolutional kernel rows [24], probabilistic masking [25], encrypting only a portion of gradient updates in federated learning [26], or combining partial functional encryption with adversarial training [27]. These works collectively show that encrypting only part of a dense model can maintain robustness against inference attacks while greatly reducing cost.

However, nearly all prior approaches assume *dense* weight representations. Directly applying such dense-oriented selective encryption to TTD-compressed models is problematic. In dense networks, redundancy allows partial encryption (e.g., 50% of weights) to achieve robustness close to full encryption. In contrast, TTD removes much of this redundancy, concentrating information into fewer parameters; leaving even a small portion unencrypted can still expose critical structure. This fundamental difference limits the effectiveness of existing methods and calls for a TTD-aware selective encryption scheme.

In this work, we present **TT-SEAL**—*Selective Encryption for Adversarially-Robust and Low-Latency*—to our knowledge, the first selective encryption framework explicitly tailored to *TTD*-compressed neural networks. TT-SEAL leverages the structural properties of TT-format weights to achieve strong protection while minimizing encryption overhead. Concretely, TT-SEAL introduces (i) a core-wise importance metric that evaluates the security impact of each TT-core, (ii) a data-driven threshold calibration that aligns protection strength with a robustness target, and (iii) a minimal-cost selection algorithm that efficiently identifies the smallest set

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by MSIT (No. RS-2024-00345668), and in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) No. RS-2025-02263706, Development of an analog-digital mixed ultra-low-power neuromorphic edge SoC).
Kyeongpil Min and Sangmin Jeon contributed equally to this work.
Woojoo Lee is the corresponding author.

DAC '26, Long Beach, CA, USA

ACM ISBN 979-8-4007-2254-7/2026/07
https://doi.org/10.1145/3770743.3804048

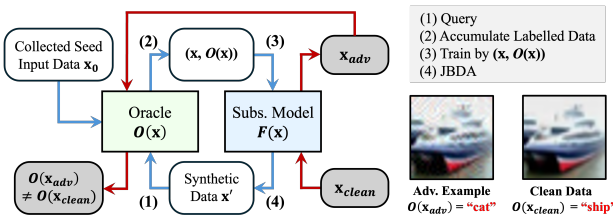

Figure 1: Transfer-based adversarial attack using JBDA. Blue lines: the attacker queries oracle $O(x)$ with clean inputs, trains a substitute $F(\mathbf{x})$, and augments data $\mathbf{x}'$ near decision boundaries. Red lines: F generates adversarial examples $\mathbf{x}_{adv}$ that transfer to O.

of TT-cores to encrypt using dynamic programming-based optimization. Through extensive experiments, we demonstrate that TT-SEAL significantly reduces decryption overhead while maintaining robustness comparable to full encryption, thereby enabling secure and low-latency inference on edge devices.

The contributions of this work are as follows:

- **TTD-tailored selective encryption.** We introduce TT-SEAL, the first selective encryption method specifically designed for TTD-compressed models, exploiting TT-core structure for strong protection with reduced overhead.
- **Core-level metric and optimization.** We define a sensitivity-based importance metric, calibrate a robustness threshold, and formulate an optimization procedure to identify the minimal set of TT-cores to encrypt.
- **Prototype-based validation.** On an FPGA-based edge AI processor, TT-SEAL achieves robustness comparable to full encryption while encrypting as little as 4.89% of parameters (ResNet-18), with similar trends observed on VGG-16 and MobileNetV2. Moreover, the decryption share in end-to-end inference is reduced from 58% to 2.76%, making secure edge inference feasible.

2 System Context and TTD-Aware Threat Model

2.1 TTD Compression for Deployed DNNs

TTD represents a d-dimensional weight tensor $\mathcal{W}$ as the product of d low-dimensional core tensors G_k [9]:

$$\mathcal{W}(i_1, i_2, \cdots, i_d) \approx G_1[i_1]\, G_2[i_2] \cdots G_d[i_d], \quad \mathcal{W} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}, \quad G_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}, \tag{1}$$

where n_k denotes the size of the k-th mode and r_k the TT-rank with $r_0 = r_d = 1$. This TT-format reduces parameter count and storage cost while preserving the layer structure.

Applied to neural networks, TTD compresses convolutional, recurrent, and transformer models with minimum accuracy loss. For example, a convolutional layer has a four-dimensional weight tensor $\mathcal{W}_{conv} \in \mathbb{R}^{C_{out} \times C_{in} \times K_h \times K_w}$, where C_{in} and C_{out} denote the number of input and output channels, and K_h and K_w represent the kernel height and width, respectively. The storage complexity of $\mathcal{W}_{conv}$ is $O(C_{out} C_{in} K_h K_w)$, while its TT-format counterpart requires only $O(r^2(C_{out} + C_{in} + K_h + K_w))$. This reduction alleviates memory footprint, reduces transmission volume, and lowers inference cost, which is particularly beneficial for edge devices [28–30].

2.2 TTD-Aware Threat Model for Edge AI

System setting. An edge SoC integrates compute units (CPUs or accelerator cores), system interconnect, on-chip memory (registers, caches, eSRAM), and off-chip DRAM. Large neural network weights cannot fit on-chip and are stored in compressed form in DRAM. During inference, these parameters are fetched over the DDR bus, decompressed on-chip, and then loaded into compute units for execution.

Trust boundary. The on-chip region is physically embedded and thus difficult to probe directly, whereas off-chip DRAM and memory buses are external and exposed to adversaries. We therefore assume model parameters in DRAM are always encrypted as a baseline protection. Nevertheless, outputs (e.g., labels, logits, or scores) remain observable through normal I/O interfaces and can be collected by an attacker.

Adversary capability. We assume (i) encrypted parameters in DRAM can be observed by the attacker, and (ii) the attacker can repeatedly query the deployed model (oracle O) to collect input-output pairs $(\mathbf{x}, O(\mathbf{x}))$. Even when all parameters remain encrypted (black-box setting), such queries allow the attacker to train a substitute model F that approximates O.

Attack procedure. Figure 1 illustrates the process: in the *blue path*, the attacker queries the oracle $O(x)$ with clean inputs, trains a substitute $F(x)$, and augments data near decision boundaries using Jacobian-based Dataset Augmentation (JBDA) [22]. In the *red path*, the trained F generates adversarial examples $\mathbf{x}_{adv}$ that transfer to O. A common generator is the Fast Gradient Sign Method (FGSM) [17, 31]:

$$\mathbf{x}_{adv} = \mathbf{x} + \epsilon \cdot \text{sign}(\nabla_{\mathbf{x}} L(\mathbf{x}, y_{\text{true}})), \tag{2}$$

where y_{true} is its ground-truth label, $L(\cdot, \cdot)$ is the loss (e.g., cross-entropy), $\nabla_{\mathbf{x}}$ is the gradient w.r.t. the input, and $\text{sign}(\cdot)$ is applied element-wise. Larger ϵ values typically raise the chance of misclassification but also make perturbations more visible, whereas smaller values keep inputs visually closer to the original but reduce attack success. Both non-targeted (any misclassification) and targeted (forcing a specific label) variants are considered realistic threats in this work.

Implication for compressed models. The effectiveness of such transfer-based attacks depends critically on how well F approximates O: the closer the approximation, the more likely adversarial examples crafted on F will transfer to O. In practice, full secrecy of weights constrains F, but even partial parameter exposure can substantially improve its accuracy and thus boost attack transferability [23]. Our experiments confirm that TTD compression does not eliminate this risk: on CIFAR-10, substitute models trained in a white-box setting achieved accuracies of 92.29% (ResNet-18 [32]), 89.5% (MobileNetV2 [33]), and 90.73% (VGG-16 [34]). Despite parameter reduction and redundancy removal, TTD models still leak sufficient structural information for substitutes to approximate decision boundaries, leaving them vulnerable to transfer-based adversarial attacks.

2.3 Limits of Selective Encryption in TT Models

Selective encryption seeks to reduce the overhead of full encryption by protecting only a subset of parameters while preserving robustness. This approach is attractive for edge deployment where full encryption incurs excessive latency and power.

Prior work has explored several strategies: kernel-row selection based on L1 norm (showing 50% suffices for dense CNNs) [24], probabilistic masking (8% encrypted) [25], selective gradient encryption in federated learning (up to 4.15× communication savings) [26], and partial functional encryption combined with adversarial training [27]. These results collectively show that encrypting only part of a model can be both feasible and effective—yet nearly all assume dense weight representations.

To examine the TT setting, we applied [24] to both dense and TTD-compressed ResNet-18 on CIFAR-10. Figure 2 plots substitute

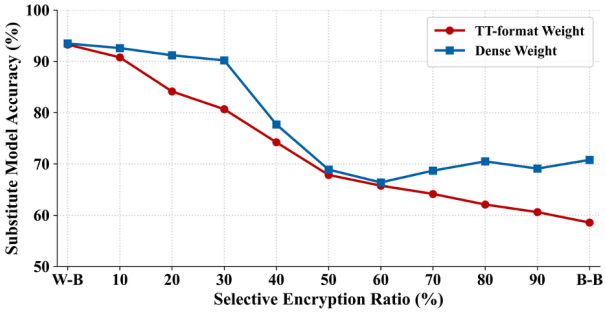

Figure 2: Substitute-model accuracy vs. selective encryption ratio in ResNet-18 with dense and TTD-compressed weights.

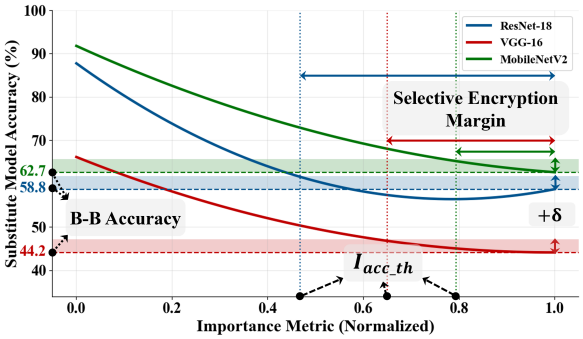

Figure 3: Relationship between I_{acc} and substitute-model accuracy for TTD-compressed ResNet-18, MobileNetV2 and VGG-16.

Algorithm 1 Computation of I_{acc} for TT-cores

Input: TTD-compressed model M, validation set $\mathcal{D}_{val}$, # of probes N_v
For each core $G_{l,k}$, initialize accumulators for gradient norms.
for mini-batch $\mathcal{B} \subset \mathcal{D}_{val}$ **do**
  Forward pass to obtain y and loss L.
  Backward pass on L to accumulate $\|\partial L/\partial G_{l,k}\|_F^2$.
  for $t = 1$ to N_v **do**
    Sample $\mathbf{v} \sim \mathcal{R}$ and compute gradient of $s(\mathbf{v}) = \mathbf{v}^\top y$.
    Accumulate $\|\partial s(\mathbf{v})/\partial G_{l,k}\|_F^2$.
  end for
end for
Normalize by $\mu_l = \text{mean}_k \|G_{l,k}\|_F$ per layer.
Return $I_{acc}(G_{l,k})$ for all cores.

accuracy versus encryption ratio. Using the unencrypted model as the *white-box* (W-B) reference and the fully encrypted model as the *black-box* (B-B) baseline: the dense model (blue) reached B-B robustness with only ∼50% encrypted weights, whereas the TTD model (red) required encrypting > 90% yet still failed to match B-B.

This discrepancy stems from information density. Dense models contain redundancy, so unencrypted portions often hold less critical information. TT-format removes redundancy during decomposition; remaining parameters are more concentrated and thus reveal more if left unencrypted. Consequently, directly applying dense-oriented selective encryption to TT-format models is ineffective. These observations motivate a selective encryption scheme specifically tailored to the TT structure, which we develop in Section 3.

3 TT-SEAL: Proposed Method

This section presents TT-SEAL, a selective encryption framework tailored to TTD-compressed neural networks. As discussed in Section 2.3, TT-format weights differ from dense representations in two key ways: redundancy is largely removed by decomposition, and information is structurally distributed across sequential TT-cores as defined in (1). These properties weaken dense-oriented selection rules, but they also imply that encrypting only a subset of cores can significantly obfuscate the original weights. TT-SEAL leverages this structure through three components: a core-wise importance metric, a data-driven security threshold, and a minimal-cost selection algorithm.

Problem Setup and Security Target

Let $\{G_{l,k}\}$ denote the set of TT-cores across all layers l, with k indexing the cores in layer l, and let $size(G_{l,k})$ be the number of parameters in core $G_{l,k}$. Following the threat model in Section 2.2, the *security target* is to identify the smallest set $S \subseteq \{G_{l,k}\}$ to encrypt such that the accuracy of a substitute model does not exceed the fully encrypted black-box baseline by more than δ (we use $\delta = 3\%$).

Let A_{BB} denote the baseline accuracy under full black-box encryption. TT-SEAL enforces the constraint

$$A_{\text{sub}}(S) \;\le\; A_{\text{BB}} + \delta, \tag{3}$$

where $A_{\text{sub}}(S)$ is the accuracy of a substitute model given encrypted set S, while minimizing the total encryption cost

$$C_{\text{enc}} = \sum_{G \in S} size(G). \tag{4}$$

Core-wise Importance Metric

To score how critical each core is to accuracy and decision boundaries, we define the **Importance Metric** I_{acc}:

$$I_{acc}(G_{l,k}) = \frac{1}{\mu_l} \sqrt{\mathbb{E}_{\mathbf{x} \in \mathcal{D}_{val}} \left[\left\| \frac{\partial L}{\partial G_{l,k}} \right\|_F^2 + \left\| \frac{\partial y}{\partial G_{l,k}} \right\|_F^2 \right]}, \tag{5}$$

where L is the training loss, y the model output (post-softmax vector), $\|\cdot\|_F$ the Frobenius norm, $\mathcal{D}_{val}$ a held-out set, and μ_l the average Frobenius norm of all cores in layer l (for cross-layer normalization). Intuitively, I_{acc} aggregates first-order sensitivities of both output and loss w.r.t. a core; a larger value implies that encrypting this core is more likely to suppress substitute accuracy.

a) Practical estimation.: The term $\partial L/\partial G$ is obtained from a standard backward pass. The Jacobian norm $\|\partial y/\partial G\|_F$ can be estimated efficiently via a Hutchinson-type trick:

$$\left\| \frac{\partial y}{\partial G} \right\|_F^2 \;=\; \mathbb{E}_{\mathbf{v} \sim \mathcal{R}} \left[\left\| \frac{\partial (\mathbf{v}^\top y)}{\partial G} \right\|_F^2 \right], \tag{6}$$

where $\mathcal{R}$ is a Rademacher or standard normal distribution over the output dimension. In practice, one computes gradients of the scalar $\mathbf{v}^\top y$ for N_v random vectors ($N_v \in [1, 4]$ typically suffices), averages their squared Frobenius norms across a small validation subset, and plugs the estimate into (5). The full computation procedure is summarized in Algorithm 1.

b) Normalization and robustness.: We use $\mu_l = \text{mean}_k \|G_{l,k}\|_F$ for scale invariance across layers; median or percentile-based options are also viable to dampen outliers. Mini-batch estimates over $\mathcal{D}_{val}$ are averaged to mitigate stochasticity.

c) Empirical monotonicity.: As illustrated in Figure 3, encrypting cores in ascending I_{acc} order yields a monotone decrease in substitute accuracy that approaches the B-B baseline. Occasional local rebounds may appear due to overlapping information across TT-cores and stochastic factors in substitute training (e.g., mini-batch sampling, random initialization, SGD noise, core-combination effects), but the overall trend decreases and converges to the B-B robustness level.

Algorithm 2 I_{acc_th} Calibration via binary search over prefixes

Input: Sorted cores $\{G_i\}$ by ascending I_{acc}, B-B robustness A_{BB}, tolerance δ
Output: I_{acc_th}
$lo \leftarrow 0$, $hi \leftarrow n$
while $lo < hi$ **do**
  $m \leftarrow \lfloor (lo + hi)/2 \rfloor$; encrypt $\{G_1, \ldots, G_m\}$
  Evaluate substitute accuracy $A_{\text{sub}}(m)$ under Section 2.2
  if $A_{\text{sub}}(m) \le A_{\text{BB}} + \delta$ **then** $hi \leftarrow m$
  else $lo \leftarrow m+1$
end while
$I_{acc\_th} \leftarrow \sum_{i=1}^{lo} I_{acc}(G_i)$
return I_{acc_th}

Threshold Calibration (I_{acc_th})

I_{acc_th} is calibrated once per model/dataset by searching the prefix of cores sorted in ascending I_{acc}. The smallest prefix whose cumulative importance reduces substitute accuracy to within $A_{\text{BB}} + \delta$ is identified, and its cumulative I_{acc} is set as the threshold. The full procedure is summarized in Algorithm 2. This one-time calibration aligns the empirical robustness target with a model-intrinsic threshold, decoupling subsequent selection from per-layer sizes.

Minimal Encryption Set: Optimization Formulation

Core sizes differ, so the same threshold may lead to different encryption ratios. We therefore minimize total encrypted parameters subject to meeting the calibrated robustness:

$$\textbf{Minimize} \quad C_{enc} = \sum_{l,k} size(G_{l,k}) \cdot x_{l,k}, \quad x_{l,k} \in \{0,1\}, \qquad (7)$$

$$\textbf{subject to} \quad \sum_{l,k} I_{acc}(G_{l,k}) \cdot x_{l,k} \;\ge\; I_{acc\_th}.$$

This is a 0–1 knapsack-type problem with "cost" $w_i = size(G_i)$ and "value" $v_i = I_{acc}(G_i)$. Our goal is to meet a value threshold with minimum cost.

Solution Method

The minimal encryption set problem in (7) is solved using a *Value-DP* formulation. Algorithm 3 details the procedure: values $v_i = I_{acc}(G_i)$ are scaled and integerized as $\tilde{v}_i$, and a dynamic program is used to minimize the encryption cost subject to I_{acc_th}. We define $\tilde{V}_{\max} = \sum_i \tilde{v}_i$, and maintain a DP table

$$dp[i][v] \;=\; \min_{\text{subsets of first } i \text{ cores with total value } =v} \text{cost}.$$

The recurrence is

$$dp[i][v] = \min\big(dp[i-1][v],\; dp[i-1][v-\tilde{v}_i] + w_i\big), \qquad (8)$$

with $dp[0][0] = 0$ and $dp[0][v > 0] = +\infty$. The optimal cost is

$$C_{enc\_opt} \;=\; \min_{v \ge \tilde{V}_{th}} dp[n][v],$$

where $\tilde{V}_{th}$ is the integerized counterpart of I_{acc_th}. The time and space complexity of the algorithm are both $O(n \cdot \tilde{V}_{\max})$, as it maintains the full DP table with backtracking. A finer scaling yields larger $\tilde{V}_{\max}$ (higher precision, higher cost), while coarser scaling reduces both.

In summary, TT-SEAL (i) quantifies core criticality via a scale-normalized, efficiently estimable I_{acc}, (ii) calibrates I_{acc_th} from the desired robustness target, and (iii) selects a *minimal* encryption set through an optimization framework. As shown in Section 4, TT-SEAL achieves robustness comparable to the black-box baseline with only a fraction of the parameters encrypted.

4 Experimental Work

In this section, we implement and evaluate TT-SEAL on a prototype FPGA-based edge AI processor. We first describe the hardware platform and experimental setup, including the RTL-designed processor,

Algorithm 3 Value-DP for minimal encryption set

Input: $\{(w_i, v_i)\}$ with $w_i = size(G_i)$ and $v_i = I_{acc}(G_i)$, threshold I_{acc_th}
Output: Minimal cost C_{enc_opt} and selected core set S
Integerize $v_i \to \tilde{v}_i$ by fixed scaling; compute $\tilde{V}_{\max} = \sum_i \tilde{v}_i$ and $\tilde{V}_{th}$
Initialize $dp[0][0] = 0$ and $dp[0][v > 0] = +\infty$; `parent[i][v]`$\leftarrow$none
for $i = 1$ to n **do**
  (Init) For all $v \in [0, \tilde{V}_{\max}]$, set $dp[i][v] \leftarrow dp[i-1][v]$
  for $v = \tilde{V}_{\max}$ down to $\tilde{v}_i$ **do**
    if $dp[i-1][v-\tilde{v}_i] + w_i < dp[i][v]$ **then**
      $dp[i][v] \leftarrow dp[i-1][v-\tilde{v}_i] + w_i$; `parent[i][v]`$\leftarrow 1$
  end for
end for
$C_{enc\_opt} \leftarrow \min_{v \ge \tilde{V}_{th}} dp[n][v]$; $v^\star \leftarrow \arg\min_{v \ge \tilde{V}_{th}} dp[n][v]$
(Backtrack) $S \leftarrow \emptyset$, $i \leftarrow n$, $v \leftarrow v^\star$
while $i \ge 1$ **do**
  if `parent[i][v]` $= 1$ **then** $S \leftarrow S \cup \{G_i\}$; $v \leftarrow v - \tilde{v}_i$
  $i \leftarrow i - 1$
end while
return C_{enc_opt}, S

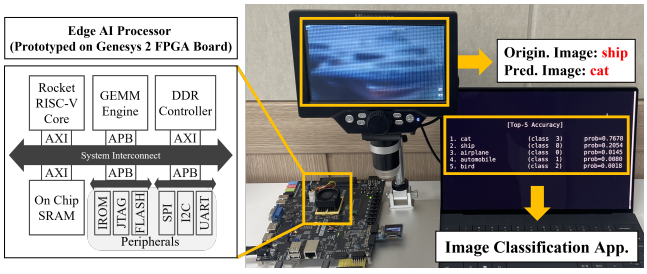

Figure 4: Image-classification demo running on the FPGA-prototyped processor, illustrating a transfer-based misclassification: an original *ship* is predicted as a *cat* under the configured threat model.

Table 1: FPGA resource utilization of the prototype edge AI processor.

IP Block	LUTs	FFs	IP Block	LUTs	FFs
Rocket RISC-V Core	15,040	9,880	Peripherals (incl. DMA)	5,043	10,370
SRAM	166	323	System Interconnect	9,745	17,375
DDR Controller	7,950	7,571	GEMM Engine	84,150	32,939

FPGA prototyping, and CIFAR-10-based adversarial attack generation. We then assess the robustness of TT-SEAL-encrypted models under the TTD-aware threat model, measuring transferability of adversarial examples across architectures and attack types. Finally, we quantify the system-level benefits of TT-SEAL by analyzing encryption cost and decryption overhead in end-to-end inference, demonstrating that selective encryption preserves B-B robustness level while enabling real-time execution on edge hardware.

4.1 Prototype Processor and Evaluation Setup

All experiments were performed on our RTL-designed and FPGA-prototyped edge AI processor, developed using RISC-V eXpress (RVX), a widely adopted EDA tool for RISC-V processor development [35–39]. The processor is based on a single-core RISC-V Rocket [40]. To support the high-bandwidth and low-latency data movement required by AI workloads, on-chip SRAMs and peripherals are connected to the core through AXI interfaces, and the system interconnect adopts a lightweight μNoC. The processor was prototyped on a Genesys2 Kintex-7 FPGA board [41] and operates at 100 MHz. The overall architecture diagram and FPGA prototype photograph are shown in Figure 4, while Table 1 summarizes the FPGA resource utilization of the prototype processor.

For application-level evaluation, we used the Darknet framework [42], which offers a lightweight C-based implementation of

Table 2: Transfer ratios (%) across ResNet-18, MobileNetV2, and VGG-16 for different perturbation strengths ϵ and attack types. Results are shown under three encryption levels: W-B (white-box), TH (I_{acc_th}), and B-B (black-box). The comparison highlights how TT-SEAL suppresses transferability to the B-B robustness level.

Models	ResNet-18												MobileNetV2												VGG-16											
ϵ	ϵ = 0/255			ϵ = 4/255			ϵ = 8/255			ϵ = 16/255			ϵ = 0/255			ϵ = 4/255			ϵ = 8/255			ϵ = 16/255			ϵ = 0/255			ϵ = 4/255			ϵ = 8/255			ϵ = 16/255		
Att. \ I_{acc}	W-B	TH	B-B	W-B	TH	B-B	W-B	TH	B-B	W-B	TH	B-B	W-B	TH	B-B	W-B	TH	B-B	W-B	TH	B-B	W-B	TH	B-B	W-B	TH	B-B	W-B	TH	B-B	W-B	TH	B-B	W-B	TH	B-B
NT	6.51	6.51	6.51	87.21	9.32	9.1	98.8	13.82	13.81	99.97	34.88	34.79	7.88	7.88	7.88	59.35	10.77	10.97	89.43	14.69	14.8	98.71	23.69	24.23	7.85	7.85	7.85	10.57	8.38	8.29	15.02	9.34	8.64	26.49	11.66	10.59
RD	0.59	0.71	0.59	49.88	0.95	1.06	84.82	1.8	1.87	96.22	4.95	3.45	0.97	0.85	0.97	20.51	1.39	1.44	53.23	2.06	1.93	81.6	3.82	3.51	0.82	0.94	0.96	1.38	0.83	0.89	2.43	1.09	1.06	4.46	1.34	1.21
SM	0.0	0.0	0.0	86.18	3.03	3.15	98.66	6.61	6.42	99.71	15.17	14.15	0.0	0.0	0.0	53.24	4.01	4.12	82.11	6.32	6.69	95.38	10.72	10.98	0.0	0.0	0.0	4.38	1.53	1.25	8.31	2.9	1.85	14.01	4.51	4.32
LL	0.0	0.0	0.0	18.63	0.0	0.0	67.36	0.02	0.01	92.1	0.25	0.39	0.0	0.0	0.0	6.41	0.0	0.0	40.57	0.01	0.01	75.31	0.7	0.56	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.53	0.02	0.0

Figure 5: Visualization of adversarial examples generated from the substitute of the TT-SEAL-encrypted model. As ϵ increases, stronger perturbations are progressively added to the input image.

CNNs suitable for FPGA prototyping. We evaluated TTD-compressed variants of ResNet-18, MobileNetV2, and VGG-16. To instantiate the threat model, of the CIFAR-10 training images, 45,000 were used to train the original (oracle) model, and the remaining 5,000 served as the attacker's initial seed data for the substitute model. The attacker trains the substitute via JBDA: beginning with the 5,000 seeds, iterative dataset augmentation expands the labeled set up to 45,000; each augmentation iteration is followed by 10 training epochs with learning rate 0.01 and SGD as the optimizer. Adversarial examples are generated using I-FGSM (Iterative-FGSM) based on (2), with 15 iterations each; in total, 10,000 adversarial examples are used for evaluation.

For encryption, we adopt the standard AES scheme. TT-SEAL is applied to the convolution layers of each model, while the network input (first convolution layer) and output (final fully connected layer) are always fully encrypted. Accordingly, the *white-box* setting means the attacker can access all convolution-layer weights; even in this case, the first and last layers remain encrypted without exception. If the attacker obtains a selectively encrypted model produced by TT-SEAL, the exposed parameters are used as-is for learning the substitute, and the encrypted remainder is randomly initialized to construct the substitute model. As a consequence of this setup, Figure 4 shows a demo in which an image of a ship is misclassified as a cat on the FPGA prototype under our configured threat model.

4.2 Evaluation

4.2.1 Adversarial Robustness. Under the threat model in Section 2.2, the accuracy of the substitute model that generates adversarial examples is a key determinant of transfer-attack success. We evaluate how effectively TT-SEAL suppresses the substitute's accuracy and thereby blocks transfer to the oracle O (original model).

Adversarial attacks are categorized as *non-targeted* and *targeted*. Non-targeted attacks induce arbitrary misclassification relative to the ground-truth label, whereas targeted attacks force the oracle to predict an attacker-specified target label. We define the transferability from the substitute F to the oracle O as

$$T_{F\to O} = \begin{cases} \frac{1}{N}\sum_{i=1}^{N}\left[O(\mathbf{x}_{adv,i}) \neq y_i\right], & \textit{(non-target)} \\ \frac{1}{N}\sum_{i=1}^{N}\left[O(\mathbf{x}_{adv,i}) = y_{t,i}\right], & \textit{(target)} \end{cases} \tag{9}$$

where $[\cdot]$ denotes the Iverson bracket (1 if the condition holds, 0 otherwise), $y_{t,i}$ is the target label for sample i, and N is the number of evaluation samples. Larger T indicates stronger transfer (more successful attacks); smaller T indicates transfer is blocked and the oracle maintains correct behavior.

We first calibrate the selective-encryption threshold I_{acc_th} for each model as described in Figure 3; the thresholds are 0.44 for ResNet-18, 0.79 for MobileNetV2, and 0.65 for VGG-16. Using substitutes trained at these I_{acc_th} points, we generate adversarial examples. Figure 5 visualizes examples for varying ϵ: small ϵ yields imperceptible perturbations, whereas larger ϵ produces visibly distorted inputs. We evaluate up to $\epsilon = 16/255$, which is widely regarded as the practical upper bound on CIFAR-10, beyond which perturbations become visually obvious.

Table 2 reports the transfer ratio (%) computed from T across ϵ, attack types, and encryption levels: *White-box* (W-B), threshold (I_{acc_th}; TH), and *Black-box* (B-B). The attack types include non-target (NT), random target (RD), second most likely target (SM), and least-likely target (LL).

First, the W-B transfer ratios show that as ϵ increases, transfer rises steeply beyond 90%; for ResNet-18, NT and SM attacks approach 100%. This confirms that, under the worst-case assumption in which the attacker knows internal parameters and structure, our threat model and setup yield strong attacks even on TTD-compressed models.

In contrast, TT-SEAL at the TH level effectively suppresses transfer to the B-B level even at large perturbations such as $\epsilon = 16/255$. For example, in the ResNet-18 model at $\epsilon = 8/255$ with SM attacks, the absolute W-B vs. TH gap reaches 92.1%. Across cases where the W-B transfer exceeds 50%, TH achieves on average a relative reduction exceeding 90%, demonstrating that the proposed method remains effective even under severe conditions.

Moreover, the effect is consistent across models and attack types. For ResNet-18, MobileNetV2, and VGG-16 alike, TH results closely match B-B, indicating broad applicability irrespective of network structure. Notably, TH suppresses transfer to the B-B level not only for RD and LL (typically easier to defend) but also for higher-success NT and SM attacks. The absolute TH–B-B difference is at most 1.5% across all cases and averages 0.2%, showing that TT-SEAL-based selective encryption maintains adversarial robustness at the B-B level across diverse conditions.

To further stress the setting and to verify that I_{acc_th} acts as a threshold for selective encryption, we analyze non-target transfer at $\epsilon = 32/255$ while sweeping I_{acc}, as shown in Figure 6. Although the magnitude of change varies by model, all show a sharp increase

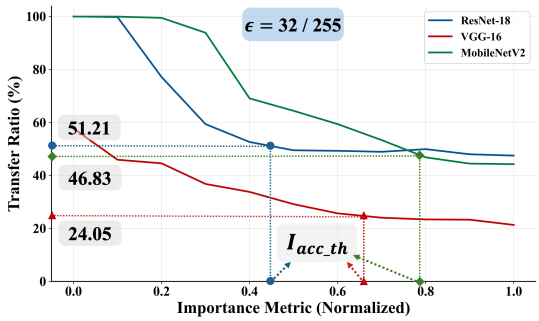

Figure 6: Effect of selective encryption under non-targeted attacks (ϵ = 32/255). Across ResNet-18, VGG-16, and MobileNetV2, the transfer ratio remains suppressed until I_{acc} falls below I_{acc_th}, demonstrating its effectiveness as a robustness threshold.

in transfer once the selective-encryption ratio drops below I_{acc_th}. This indicates that minimal selective encryption at or above I_{acc_th} is sufficient to suppress transfer, providing adversarial robustness that converges to the B-B level.

To verify that TT-SEAL generalizes beyond I-FGSM, we evaluate seven transfer-attack generators compatible with our threat model and operating on the substitute model: PGD [43], DI^2-FGSM [44], MI-FGSM [45], M-DI^2-FGSM [44], SI-NI-FGSM [46], VMI-FGSM [47], and LGV [48]. Figure 7 summarizes non-targeted transfer ratios at $\epsilon = 16/255$ for ResNet-18, MobileNetV2, and VGG-16. Despite differences in absolute transferability across generators, all attacks are suppressed to the B-B level at the importance threshold $I_{\text{acc\_th}}$, with a sharp rise in transferability observed only below this point. These results indicate that $I_{\text{acc\_th}}$ generalizes across attack generators and architectures, further validating the robustness of TT-SEAL.

4.2.2 TT-SEAL Performance Analysis via Optimization. We now quantify performance gains obtained at the optimized selective-encryption ratio found by TT-SEAL. Table 3 summarizes, for each model (ResNet-18, MobileNetV2, VGG-16), the number of encrypted parameters and the decryption-time ratios at $I_{acc} \in \{0.2, 0.4, 0.6, 0.8\}$ and at I_{acc_th}. Here, $t_{TT-SEAL}$ and t_{BB} denote the decryption time with TT-SEAL (optimized selective encryption) and with full decryption, respectively; t_{Inf} denotes the end-to-end inference time including AES decryption, TTD decoding, and the forward pass.

Because I_{acc} is not directly determined by core size, a core's importance does not necessarily correlate with its parameter count. Empirically, applying TT-SEAL's optimization tends to produce a generally monotonic trend: as I_{acc} increases and the search becomes more selective, the number of encrypted parameters increases gradually. This indicates that I_{acc}-based selection can modulate encryption cost without relying on raw core size.

Across models, the total parameter counts are 1.83M (ResNet-18), 447k (MobileNetV2), and 1.51M (VGG-16). At I_{acc_th}, the encrypted parameters are about 89k (4.89%), 71k (15.92%), and 98k (6.46%),

Table 3: Selective-encryption parameters and decryption times for each model under I_{acc} intervals and at I_{acc_th}.

I_{acc}		0.2	0.4	0.6	0.8	I_{acc_th}
ResNet-18	Encrypted Parameters	21,028	72,208	134,604	465,068	89,685
	Encryption Ratio (%)	1.14	3.94	7.35	25.41	4.89
	$t_{TT-SEAL}$ / t_{BB} (%)	1.13	4.16	7.23	25.7	4.87
	$t_{TT-SEAL}$ / t_{Inf} (%)	0.66	2.37	4.04	13.02	2.76
MobileNetV2	Encrypted Parameters	15,317	19,143	29,794	74,706	71,268
	Encryption Ratio (%)	3.42	4.29	6.66	16.7	15.92
	$t_{TT-SEAL}$ / t_{BB} (%)	3.39	4.52	6.62	16.13	15.66
	$t_{TT-SEAL}$ / t_{Inf} (%)	1.39	1.85	2.70	6.32	6.15
VGG-16	Encrypted Parameters	9,699	29,364	65,052	297,705	97,513
	Encryption Ratio (%)	0.64	1.94	4.31	19.72	6.46
	$t_{TT-SEAL}$ / t_{BB} (%)	0.63	2.05	4.28	19.06	6.35
	$t_{TT-SEAL}$ / t_{Inf} (%)	0.25	0.79	1.64	6.92	2.42

respectively. ResNet-18 and VGG-16 require only ∼5–6% to secure robustness, whereas MobileNetV2—due to its smaller parameter budget and deeper, more fragmented TT-cores—distributes importance more uniformly and therefore selects more cores, yet still needs only about one-sixth of parameters to reach B-B level.

In terms of decryption time, reducing the encrypted parameter count shrinks $t_{TT-SEAL}$ to as little as 4.87% of t_{BB}. Moreover, the decryption share of B-B end-to-end inference time-58% (ResNet-18), 41.8% (MobileNetV2), and 39% (VGG-16)-drops to 2.76%, 6.15%, and 2.42% under TT-SEAL, respectively. By driving AES decryption overhead down to the low single digits, TT-SEAL preserves security while enabling real-time inference on edge devices.

5 Conclusion

This paper presented *TT-SEAL*, a TTD-aware selective encryption framework for secure and efficient edge AI. TT-SEAL exploits the structure of TT-format weights to protect only a minimal set of critical cores: it scores cores with a scale-normalized importance metric that aggregates loss and output sensitivities, calibrates a data-driven threshold aligned with a robustness target, and selects a minimum-cost set of TT cores to encrypt through an optimization procedure. Under a compression-aware threat model and on an FPGA-prototyped edge AI processor, TT-SEAL suppresses adversarial transfer while encrypting as little as 4.89% of parameters (ResNet-18; 6.46% for VGG-16 and 15.92% for MobileNetV2). Consequently, the share of AES decryption in end-to-end inference drops from double digits to the low single digits across our models (e.g., 58% → 2.76% on ResNet-18; 41.8% → 6.15% on MobileNetV2; 39% → 2.42% on VGG-16), confirming that TT-SEAL preserves security while satisfying edge latency constraints. Taken together, these results demonstrate that compression and security can be jointly achieved: by aligning protection with the TT structure, TT-SEAL delivers robustness comparable to full encryption with markedly lower decryption overhead, enabling practical deployment of TTD-compressed models at the edge.

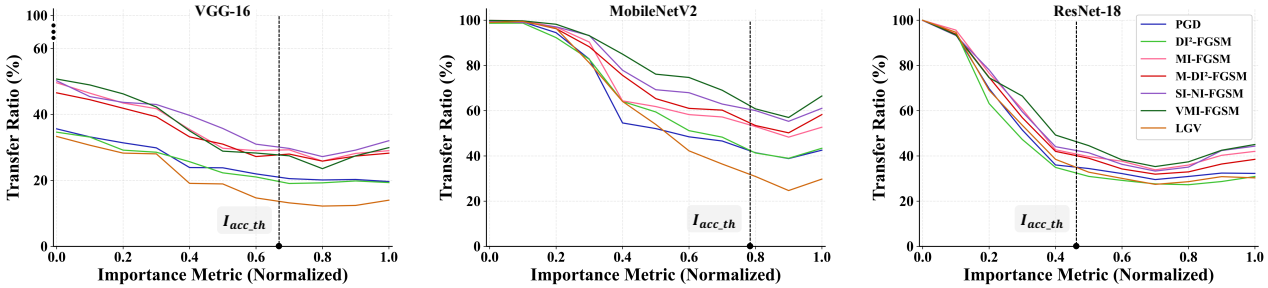

Figure 7: Transfer ratios across models for diverse adversarial example generation methods under the non-targeted attack with ϵ = 16/255.

References

[1] Liang Wang, Kai Lu, Nan Zhang, Xiaoyang Qu, Jianzong Wang, Jiguang Wan, Guokuan Li, and Jing Xiao. Shoggoth: Towards efficient edge-cloud collaborative real-time video inference via adaptive online learning. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2023.

[2] Lingling Zhang, Zhenxiong Zhou, Bo Yi, Jing Wang, Chien-Ming Chen, and Chunyang Shi. Edge-cloud framework for vehicle-road cooperative traffic signal control in augmented internet of things. *IEEE Internet of Things Journal*, 2024.

[3] Md Maruf Hossain Shuvo, Syed Kamrul Islam, Jianlin Cheng, and Bashir I Morshed. Efficient acceleration of deep learning inference on resource-constrained edge devices: A review. *Proceedings of the IEEE*, 111(1):42–91, 2022.

[4] Zhehui Wang, Tao Luo, Rick Siow Mong Goh, and Joey Tianyi Zhou. EDCompress: Energy-aware model compression for dataflows. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):208–220, 2022.

[5] Xuan Shen, Weize Ma, Jing Liu, Changdi Yang, Rui Ding, Quanyi Wang, Henghui Ding, Wei Niu, Yanzhi Wang, Pu Zhao, et al. QuartDepth: Post-training quantization for real-time depth estimation on the edge. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11448–11460, 2025.

[6] Yang He and Lingao Xiao. Structured pruning for deep convolutional neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(5):2900–2919, 2023.

[7] Thanaphon Suwannaphong, Ferdian Jovan, Ian Craddock, and Ryan McConville. Optimising tinyml with quantization and distillation of transformer and mamba models for indoor localisation on edge devices. *Scientific Reports*, 15(1):10081, 2025.

[8] Xingyi Liu and Keshab K Parhi. Tensor decomposition for model reduction in neural networks: A review [feature]. *IEEE Circuits and Systems Magazine*, 23(2):8–28, 2023.

[9] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[10] Hyunseok Kwak, Kyeongwon Lee, Kyeongpil Min, Chaebin Jung, and Woojoo Lee. TT-Edge: A hardware-software co-design for energy-efficient tensor-train decomposition on edge AI. *arXiv preprint arXiv:2511.13738*, 2025.

[11] Haoyan Dong, Hai-Bao Chen, Jingjing Chang, Yixin Yang, Ziyang Gao, Zhigang Ji, Runsheng Wang, and Ru Huang. Duqtta: Dual quantized tensor-train adaptation with decoupling magnitude-direction for efficient fine-tuning of llms. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, pages 1–7. IEEE, 2025.

[12] Hyunseok Kwak, Sangmin Jeon, Kyeongwon Lee, and Woojoo Lee. Optimizing tensor-train decomposition for efficient edge AI: Accelerated decoding via gemm and reshape minimization. *AIMS Mathematics*, 10(7):15755–15784, 2025.

[13] Keke Gai, Yaoling Ding, An Wang, Liehuang Zhu, Kim-Kwang Raymond Choo, Qi Zhang, and Zhuping Wang. Attacking the edge-of-things: A physical attack perspective. *IEEE Internet of Things Journal*, 9(7):5240–5253, 2021.

[14] Cheng Wang, Zenghui Yuan, Pan Zhou, Zichuan Xu, Ruixuan Li, and Dapeng Oliver Wu. The security and privacy of mobile-edge computing: An artificial intelligence perspective. *IEEE Internet of Things Journal*, 10(24):22008–22032, 2023.

[15] Weizhe Hua, Zhiru Zhang, and G Edward Suh. Reverse engineering convolutional neural networks through side-channel information leaks. In *Proceedings of the 55th Annual Design Automation Conference*, pages 1–6, 2018.

[16] Sanjay Kariyappa and Moinuddin K Qureshi. Defending against model stealing attacks with adaptive misinformation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–778, 2020.

[17] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, page 99–112. Chapman and Hall/CRC, 2018.

[18] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019.

[19] Zhida Bao, Yun Lin, Sicheng Zhang, Zixin Li, and Shiwen Mao. Threat of adversarial attacks on DL-based IoT device identification. *IEEE Internet of Things Journal*, 9(11):9012–9024, 2021.

[20] Sunder Ali Khowaja, Ik Hyun Lee, Kapal Dev, Muhammad Aslam Jarwar, and Nawab Muhammad Faseeh Qureshi. Get your foes fooled: Proximal gradient split learning for defense against model inversion attacks on iomt data. *IEEE Transactions on Network Science and Engineering*, 10(5):2607–2616, 2022.

[21] Adnan Qayyum, Muhammad Usama, Junaid Qadir, and Ala Al-Fuqaha. Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward. *IEEE Communications Surveys & Tutorials*, 22(2):998–1026, 2020.

[22] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

[23] Ian Goodfellow, Patrick McDaniel, and Nicolas Papernot. Making machine learning robust against adversarial inputs. *Communications of the ACM*, 61(7):56–66, 2018.

[24] Pengfei Zuo, Yu Hua, Ling Liang, Xinfeng Xie, Xing Hu, and Yuan Xie. Sealing neural network models in encrypted deep learning accelerators. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 1255–1260. IEEE, 2021.

[25] Jinyu Tian, Jiantao Zhou, and Jia Duan. Probabilistic selective encryption of convolutional neural networks for hierarchical services. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2205–2214, June 2021.

[26] Chenghao Hu and Baochun Li. Maskcrypt: Federated learning with selective homomorphic encryption. *IEEE Transactions on Dependable and Secure Computing*, 22(1):221–233, 2024.

[27] Théo Ryffel, David Pointcheval, Francis Bach, Edouard Dufour-Sans, and Romain Gay. Partially encrypted deep learning using functional encryption. *Advances in Neural Information Processing Systems*, 32, 2019.

[28] Donghyun Lee, Ruokai Yin, Youngeun Kim, Abhishek Moitra, Yuhang Li, and Priyadarshini Panda. TT-SNN: tensor train decomposition for efficient spiking neural network training. In *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6. IEEE, 2024.

[29] Zikun Wei, Tingting Wang, Chenchen Ding, Bohan Wang, Ziyi Guan, Hantao Huang, and Hao Yu. FMTT: Fused multi-head transformer with tensor-compression for 3d point clouds detection on edge devices. In *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6. IEEE, 2024.

[30] Kang Han and Wei Xiang. Multiscale tensor decomposition and rendering equation encoding for view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4232–4241, 2023.

[31] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[33] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[35] Kyuseung Han, Sukho Lee, Kwang-Il Oh, Younghwan Bae, Hyeonguk Jang, Jae-Jin Lee, Woojoo Lee, and Massoud Pedram. Developing TEI-aware ultralow-power soc platforms for iot end nodes. *IEEE Internet of Things Journal*, 8(6):4642–4656, 2021.

[36] Jina Park, Kyuseung Han, Eunjin Choi, Jae-Jin Lee, Kyeongwon Lee, Woojoo Lee, and Massoud Pedram. Designing low-power RISC-V multicore processors with a shared lightweight floating point unit for IoT endnodes. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 71(9):4106–4119, 2024.

[37] Kyeongwon Lee, Sangmin Jeon, Kangju Lee, Woojoo Lee, and Massoud Pedram. Radar-PIM: Developing IoT processors utilizing Processing-in-Memory architecture for ultrawideband-radar-based respiration detection. *IEEE Internet of Things Journal*, 12(1):515–530, 2025.

[38] Sangmin Jeon, Kangju Lee, Kyeongwon Lee, and Woojoo Lee. HH-PIM: Dynamic optimization of power and performance with heterogeneous-hybrid PIM for edge AI devices. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, pages 1–7, 2025.

[39] Jongin Choi, Jina Park, Jae-Jin Lee, Massoud Pedram, and Woojoo Lee. Asap-fe: Energy-efficient feature extraction enabling multi-channel keyword spotting on edge processors. In *2025 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 1–7, 2025.

[40] SiFIVE. https://github.com/chipsalliance/rocket-chip, Accessed 01 Apr. 2026.

[41] Xilinx. https://digilent.com/shop/genesys-2-kintex-7-fpga-development-board, Accessed 01 Apr. 2026.

[42] Joseph Redmon. Darknet: Open source neural networks in c. https://pjreddie.com/darknet/, 2013–2016.

[43] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[44] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2730–2739, 2019.

[45] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[46] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. *arXiv preprint arXiv:1908.06281*, 2019.

[47] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1924–1933, 2021.

[48] Martin Gubri, Maxime Cordy, Mike Papadakis, Yves Le Traon, and Koushik Sen. Lgv: Boosting adversarial example transferability from large geometric vicinity. In *European Conference on Computer Vision*, pages 603–618. Springer, 2022.