XXXX

基于OVSF编码的边缘侧DNN低成本容错方法

靳松[1,2,3]，武炳硕[1,2,3]

（1. 华北电力大学 电子与通信工程系，河北 保定 071003；
2. 华北电力大学 河北省电力物联网技术重点试验室，河北 保定 071003；
3. 华北电力大学 电力物联智慧化技术河北省工程研究中心，河北 保定 071003）

摘　要：针对资源受限边缘设备上神经元级容错方法阈值存储开销过大的问题，报告提出了一种基于正交可变扩频因子（Orthogonal Variable Spreading Factor，OVSF）基码的神经元级阈值压缩方法。该方法将激活阈值表示为少量OVSF正交基码的线性组合，仅需存储稀疏系数及其索引，并在推理阶段通过在线重构机制恢复完整阈值，从而在不引入显著计算开销的前提下显著降低存储与传输需求。实验结果表明，在AlexNet与VGG16等典型模型上，该方法在阈值保真度保持在94%以上且容错性能与未压缩FitAct方案基本一致的前提下，将阈值存储开销降低50%–90%。进一步的FPGA(Field Programmable Gate Array)边缘平台实验验证了该方法在带宽受限场景下的可实现性与效率优势。该工作为在边缘设备上部署高效、低开销的容错神经网络提供了一种可行方案。

关键词：深度神经网络；容错；正交可变扩频因子；模型压缩；硬件瞬态故障

中图分类号：TP393

文献标志码：A

doi: 10.11959/j.issn.1000–0801.

A Resource-Efficient Fault Tolerance Method for Edge-Side DNNs Based on OVSF Coding

JIN Song[1,2,3], WU Bingshuo[1,2,3]

1. Department of Electronic and Communication Engineering, North China Electric Power University, Baoding 071003, Hebei, China
2. Hebei Key Laboratory of Power Internet of Things Technology, North China Electric Power University, Baoding 071003, Hebei, China
3. Hebei Engineering Research Center of Intelligent Technology for Power Internet of Things，North China Electric Power University, Baoding 071003, Hebei, China

Abstract: To address the excessive storage overhead of neuron-level fault-tolerance methods on resource-constrained edge devices, this paper proposes a neuron-level threshold compression method based on Orthogonal Variable Spread-

收稿日期：XXXX–XX–XX；修回日期：XXXX–XX–XX
通信作者：靳松，jinsong@ncepu.edu.cn
基金项目：河北省省级科技计划(No.SZX2020034);河北省自然科学基金(No.F2021502006)
Foundation Items: Hebei Provincial Science and Technology Plan (No.SZX2020034); Natural Science Foundation of Hebei Province (No.F2021502006)

ing Factor (OVSF) basis codes. In the proposed approach, activation thresholds are represented as linear combinations of a small number of orthogonal OVSF basis codes. Only sparse coefficients and their indices are stored, while full thresholds are reconstructed on-the-fly during inference through a lightweight online reconstruction mechanism. This design significantly reduces threshold storage and transmission overhead without introducing substantial computational cost. Experimental results on representative DNN models, including AlexNet and VGG16, show that when the threshold fidelity score (TFS) is maintained above 94%, the proposed method achieves fault-tolerance performance comparable to the uncompressed FitAct scheme. Under this condition, the threshold storage overhead can be reduced by 50% – 90%, depending on the selected compression ratio. Furthermore, FPGA-based edge experiments verify the feasibility and efficiency of the proposed method under bandwidth-constrained deployment scenarios. These results indicate that the proposed approach provides a practical and resource-efficient solution for deploying low-overhead fault-tolerant neural networks on edge devices.

Key words: Deep Neural Networks (DNN), Fault Tolerance, Orthogonal Variable Spreading Factor (OVSF), Model Compression, Hardware Transient Faults

1 概述

随着深度神经网络(Deep Neural Network，DNN)的广泛应用，其可靠性问题日益凸显[1][2]。在实际部署环境中，软错误（如瞬时性硬件故障引发的内存位翻转）可能导致神经网络激活值异常传播，显著降低模型推理的可靠性[3]。研究表明，在典型嵌入式系统中，软错误率可达10^{-7}至10^{-4}比特每小时，而在高海拔或辐射环境下可能升高至10^{-3}量级[4][5]。单个比特翻转可能导致激活值产生数量级的偏差，这种异常会沿着网络层级指数级放大，最终导致网络做出灾难性的误判[6]。例如，在自动驾驶场景中，传感器数据一位比特的错误可能导致车辆将行人误识别为背景，造成严重后果[1]。

为解决DNN遭受软错误时的可靠性问题，学者们提出了传统硬件冗余容错技术（如三模冗余TMR和错误检测与纠正码ECC），但因资源开销过大而难以应用于资源受限的边缘设备。TMR需3倍硬件资源，ECC需额外25-30%的存储开销，而微控制器类设备的功耗预算仅毫瓦级、存储容量常以KB计[7]。近年来出现了基于激活值约束的容错技术（如FitAct[8]），通过为每个神经元学习独立的阈值，可有效抑制由软错误引发的异常激活值传播。然而，该方法需为每个神经元存储独立阈值（单个阈值通常占用4-32位）。即使部署MobileNet等轻量模型，额外开销仍可达数百KB——远超微控制器KB级存储容量，严重限制其在资源受限设备中的应用。

已有的基于阈值约束的方法在可靠性提升与模型推理精度之间普遍存在权衡关系。全局阈值方法（如Ranger、Clip-Act）无法适配神经元间的激活值分布差异[9][10]。在DNN中，同一层内不同通道的输出激活值动态范围可能相差一个数量级以上，统一阈值忽略了这种非均匀性，往往导致在提升容错能力的同时对模型推理精度产生不利影响。具体而言，这样的阈值对激活幅值较低神经元而言过于宽松，无法有效屏蔽故障导致的激活偏差，致使错误传播；而对激活幅值较高神经元则约束过强，通过削弱其有效的表征信号损害了模型的分类性能。而采用神经元级的阈值（如FitAct）虽然能精确匹配每个神经元的激活分布，但其阈值的存储开销随网络规模线性增长，且在推理时需要频繁的内存访问，增加了能耗和延迟[8]。现有模型压缩技术（如权重量化、二值化）可降低参数规模，但直接压缩阈值参数会降低其容错精度。例如，在8位量化模型中应用8位阈值时，因阈值边界控制需更高分辨率（通常

需 16 位以上），故障抑制效果仍会下降超过 30%[11]。而传统的熵编码、霍夫曼编码等无损压缩技术压缩率有限（通常仅 2-3 倍），且解码开销大。现高压缩率、低解码成本的阈值压缩方案，已成为将阈值约束方法应用于边缘侧神经网络可靠部署的瓶颈。

因此，本文主要面向边缘计算环境下的深度神经网络推理应用场景。在该场景中，神经网络模型通常部署在资源受限的边缘设备上，仅执行推理任务而不涉及训练过程。受制于存储容量、功耗预算以及硬件可靠性等因素，网络参数（尤其是阈值或激活相关参数）在存储过程中易受到软错误（如随机比特翻转）的影响，从而导致推理精度显著下降。在保证推理精度的同时，如何在不引入过高存储和计算开销的前提下实现有效的容错保护，成为边缘侧深度神经网络部署中亟需解决的问题[12]。

针对上述挑战，本文提出一种基于正交可变扩频因子（OVSF）基码的神经元级阈值压缩框架。该框架通过结构化稀疏表示与在线容错机制的协同设计，实现了存储效率与容错性能之间的平衡。

具体而言，本文的工作体现在以下三个方面：

1. 神经元级阈值的 OVSF 编码表示：本文首先将每个神经元通道的激活阈值在数学上表示为一个高维向量，并采用其在标准基下的原始数值表示形式。基于信号分解理论，本文证明了该向量可以由一组正交的 OVSF 基码进行精确的线性组合表示。这一变换将阈值数据从原始参数表示域映射到了正交的变换域，是后续高效压缩的理论基础。本文所称的‘OVSF 变换域’，是指阈值向量经 OVSF 正交基展开后，其系数向量所在的坐标空间，即正交线性变换后的表示域。

2. 阈值数据压缩：为实现压缩，本文设计了一种迭代优化算法来选择性地保留信息。该算法通过交替执行“能量筛选”与“最小二乘拟合”，在给定的压缩率的前提下，精准地定位出对阈值重构贡献最大的少数基码及其系数。为便于描述，本文将阈值在 OVSF 变换域中各系数的幅值的平方称为对应基码的‘能量’，其大小反映了该基码对阈值重构的贡献。因此，压缩的本质是仅需存储这部分稀疏的系数向量及其对应的基码索引，而非完整的阈值数据，从而实现了存储需求与网络规模的解耦。

3. 阈值的在线重构机制：在模型推理阶段，本文设计并实现了一种轻量级的在线重构机制。该机制利用 OVSF 基码可通过哈达玛矩阵递归构造的确定性特点，避免了存储基码矩阵本身。推理模块仅需根据预存的基码索引，便可“无状态”地实时生成所需基码，并与从内存中加载的稀疏系数进行矩阵运算，高效地还原出完整的神经元阈值，以供神经网络激活层使用。

该框架通过上述设计，将通信领域的编码理论转化为一种适用于神经网络容错的、数学上可解释的压缩方案。在 CIFAR-10 数据集及 VGG16 架构上的实验结果证实了本方法的有效性：在 1×10^{-6} 的故障率下，本方法在容错性能上与未压缩的 FitAct 方案持平，但阈值存储开销从 3.2MB 锐减至 0.8MB（当 $\rho=0.25$ 时），而推理延迟仅有微量增加，完全满足边缘设备的部署要求。

2　相关工作

在面向安全关键应用的边缘计算场景下，DNN 的容错问题备受关注。现有工作主要围绕硬件冗余、算法优化与模型压缩等方向展开。

2.1　硬件冗余容错方法

传统容错技术（如双模冗余（Dual Modular Redundancy, DMR）、三模冗余）通过复制计算单元或数据实现错误检测与恢复[7]。例如，Tesla 自动驾驶系统采用 DMR 对关键计算模块进行冗余备份，但此类方法需额外消耗 100%-200% 的硬件

资源，难以在内存与算力受限的边缘设备中普及[13]。针对此问题，Mahmoud等人[14]提出了选择性节点加固方案，其核心理论是：网络中所有神经元对模型输出的影响力并非都均等，因此可以通过仅保护“最脆弱”的节点来实现较高的资源效率。该方法通常通过大规模的故障注入仿真，统计分析单个神经元或权重发生比特翻转时，对模型最终输出产生的误差大小，从而量化每个节点的“脆弱性指数”。根据该指数，仅对最脆弱的少数节点（例如排名前5%）应用硬件冗余，如双模冗余。由于脆弱性评估模型存在缺陷、故障影响随数据和状态动态变化，且在高故障率时表现不佳，该方案难以成为应对高可靠性与高故障率场景的通用方法。

2.2 基于激活值约束的容错优化

近年来，学术界提出了多种软件层面的容错方法。其中，基于激活值约束的技术逐渐成为热点方向。Clip-Act为神经网络每层的激活函数设置一个统一的阈值来阻止故障传播，从而显著提升了DNN的容错能力[10]；Ranger[9]同样利用类似的方法将关键故障产生的大偏差转化为可容忍的小偏差，在多个网络上将错误容忍度提升了3～50倍。然而，Clip-Act和Ranger都使用层级或全局统一阈值，难以适应不同神经元间的激活值分布差异，无法满足精细化容错。为了更精细地控制激活值，FitAct[8]提出了神经元级的后训练激活阈值优化方法，通过为每个神经元学习独立的激活值上下界来抑制错误传播，在多种常用模型上表现出优于Clip-Act和Ranger的容错性能。FitAct虽然有效，但需为每个神经元都存储一个阈值，内存和带宽开销较高，不利于部署在微处理器等边缘设备上。

随着边缘计算与安全关键场景对实时性及能效约束的日益严格，研究重点逐渐从传统的“高冗余纠错”转向“可承受开销的轻量化防护”。2024年，Zhou等人提出SAR(Sharpness-Aware Minimization，SAR)框架，借助SAR提升模型对权重比特翻转的内在鲁棒性，其核心在于不引入显式冗余，而是通过优化训练目标来增强容错能力[15]。同年，Xu等人面向CPU－GPU一体化边缘设备，提出DarkneTF框架，在可信执行环境（Trusted Execution Environment, TEE）内对权重完整性与关键卷积计算进行算法级验证，并以较低内存开销（0.46%－10.22%）和显著加速效果提供了可行的工程实现路径[16]。2025年，Zheng等人提出SAVE方法，通过离线筛选需“可靠内存”保护的关键数据位置，并结合推理时的快速范围校验，实现对GPU显存比特翻转的推理级容错，在仅保护少量关键数据的情况下大幅降低了可靠内存的占用[17]。在芯片/阵列计算层面，Xue等人提出ApproxABFT，将“近似计算”思想引入ABFT流程，通过阈值化选择性恢复与分块机制，减少因过度纠错带来的计算开销[18]。此外，Xue等人还提出了基于轻量图神经网络（GNN）的输入相关脆弱性预测与自适应保护策略，推动防护机制从“静态选择性保护”向“在线自适应防护”演进，在保证可靠性的同时进一步降低平均开销[19]。

从上述研究可以看出，近期工作主要沿两个方向展开：一是在训练阶段提升模型的内在鲁棒性（如SAR）；二是在运行阶段通过验证或选择性保护来降低开销（如DarkneTF、SAVE、ApproxABFT与自适应保护策略）。然而，现有方法多集中于“保护权重/显存比特”或“对计算结果进行验证”，针对本文关注的“基于OVSF编码的激活/阈值存储结构”，如何在容错能力、硬件开销与计算精度之间进行系统权衡与优化，仍缺乏完整的分析框架。因此，有必要在已有研究基础上，对该方向展开进一步针对性探索。

2.3 模型压缩与容错的权衡

模型压缩技术（如量化、剪枝）已成为神经网络在边缘设备上部署的关键手段，其目标是降

低模型的计算与存储开销，从而满足资源受限环境下的实时性与能效要求。然而，在涉及容错机制时，压缩与可靠性之间存在复杂的关联[11]。模型压缩通常优化的是网络整体的统计特性与推理效率，而基于阈值的容错机制则依赖于局部参数的精确数值边界。

在此情境下，若直接将通用的量化或有损压缩方法应用于容错阈值，可能导致精度边界的微小偏移。例如，将32位浮点阈值量化为8位整数时，阈值边界不可避免地产生量化误差，这会引发两类潜在风险：一是阈值上移，使得部分应被抑制的错误激活未被检测（故障“逃逸”）；二是阈值下移，过度抑制正常激活信号，削弱模型原有的判别能力。

相较之下，无损压缩方法（如霍夫曼编码）虽然能保持精度，但其压缩率有限，且解码开销较大，不利于在低功耗设备上进行实时推理计算。

因此，在压缩与容错之间寻找平衡点成为边缘人工智能部署的关键问题。本研究提出的基于OVSF编码的阈值压缩方法，并非直接对阈值进行数值压缩，而是通过在正交域中保留阈值信号的主要能量分量（即对重构误差贡献最大的少数基码系数），实现了高压缩率与高重构精度的兼顾。

这种方法在保持容错边界精度的同时，大幅降低了存储与带宽需求，为模型压缩与容错提供了一种可行路径。

2.4　编码理论在神经网络中的应用

正交可变扩频因子（Orthogonal Variable Spreading Factor, OVSF）码最早用于宽带码分多址（W-CDMA）通信系统，是一种可递归生成的二值正交码族[20]。其构造基于Hadamard矩阵，通过层级化的码树生成互相正交、取值为±1的基码，从而实现多通道信号的无干扰复用。OVSF码完备的正交性与可伸缩码长的数学特性，使其适合用于构建可复用的二值正交基空间。

近年来，一些研究开始将OVSF编码思想引入神经网络结构设计中。例如文献[21]提出了Deterministic Binary Filters (DBF)方法，利用一组预定义的OVSF二值正交基来表示卷积核，通过学习这些基的线性组合系数，实现了卷积层参数量的显著减少和模型压缩效果。

类似地，后续研究也尝试基于OVSF或Hadamard基对卷积滤波器进行稀疏重构，以降低模型的存储与计算开销。例如，Cetin等人提出的正交变换域感知层能够基于DCT、HT等正交变换构造卷积层的替代方案，在减少参数规模与乘法累加操作（MAC）的同时，保持甚至提升模型精度[22]；Hamdan与Cetin进一步在2025年提出HTMA-Net，将Hadamard变换层与存内计算中的乘法规避算子相结合，在精度基本不变的前提下可消除最高约52%的乘法运算[23]。在训练与量化优化方面，Kim等人在CVPR 2025中提出HOT方法，将Hadamard量化与Hadamard低秩近似有选择地应用于反向传播路径，实现最高约75%的显存节省，并带来实际的GPU加速收益[24]。

这些研究显示“正交结构/码字”已从传统通信扩频场景扩展到神经网络高效计算与压缩；但现有工作多聚焦于训练/推理效率或一般的正交变换，尚缺乏针对本文所讨论的OVSF编码约束、阈值存储与容错开销之间关系的系统化建模与优化，因此仍有进一步研究空间。

以往方法多采用固定比例截断策略进行OVSF线性组合，本研究提出了一种基于变换域系数贡献度的自适应基码选择机制：通过计算各基码对应展开系数对阈值重构的贡献占比（例如以系数幅值或平方幅值衡量），自适应保留贡献最大的少数成分，从而在近似精度可控的前提下实现高压缩率的阈值重构。该策略将基码选择由静态规则转为数据驱动的动态选择，进而提升了编码效率与重构稳定性。

3 本文方法

本节将详细介绍所提出的基于正交可变扩频因子基码的阈值压缩方法。具体来说，该方法将所有神经元的阈值分解为一组OVSF基码与对应系数的线性组合，只需存储极少量系数即可在线精确重构阈值。该方法包括三个阶段：原始模型训练、阈值提取与压缩、后训练微调与推理时阈值重构。首先，对神经网络进行训练，获得初始权重与性能；随后，按层逐通道提取激活阈值，并利用OVSF基码将其压缩为少量系数；最后，在推理阶段通过重构基码动态还原阈值，传输至激活函数中以实现容错目的。本文方法为后训练（post-training）压缩方法，OVSF系数在压缩阶段确定，在推理阶段保持固定，不参与反向传播。图1介绍了本文的方法实现容错目的的全部流程。

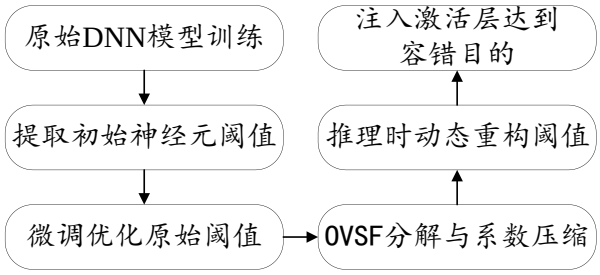

图1 基于OVSF的阈值压缩与重构整体流程

需要说明的是，OVSF编码思想本身源自通信领域，并非本文首次提出。本文的创新性不在于提出新的编码理论，而在于将OVSF正交结构引入神经元级阈值容错问题，并构建了一种面向边缘推理场景的结构化压缩与在线重构框架。

相较于已有基于正交基的参数压缩方法，本文的主要创新体现在：（1）建立了阈值在OVSF正交域中的稀疏表示模型，并从容错性能保持角度分析其可压缩性；（2）提出了结合能量筛选与最小二乘优化的迭代基码选择算法，在压缩率约束下最小化重构误差；（3）设计了支持在线轻量化重构的实现机制，使压缩与容错在边缘推理阶段协同工作。

上述创新构成了本文方法的核心贡献。

3.1 系统模型

本文考虑的系统模型为边缘计算环境下的深度神经网络推理系统。在该系统中，已完成训练的神经网络模型被部署在资源受限的边缘设备上，用于执行推理任务，例如图像分类或目标识别等典型感知类应用。由于此类应用通常对实时性和能耗具有较高要求，系统仅在部署阶段执行推理计算，不涉及训练或在线参数更新过程。

在硬件层面，网络参数（包括阈值相关参数）存储在片上SRAM(Static Random Access Memory)或外部DRAM(Dynamic Random Access Memory)中。受工艺微缩、电压波动及环境辐射等因素影响，存储单元可能发生随机软错误，表现为参数比特位翻转。本文假设计算单元本身是可靠的，从而将研究重点聚焦于存储阶段软错误对推理精度的影响。

从系统约束角度看，边缘设备通常在存储容量和计算能力方面受限，传统基于冗余复制或复杂纠错编码的容错方案会带来显著的额外存储与计算开销，难以直接应用于上述场景。因此，本文重点研究在该系统模型下，如何在不显著增加存储和计算开销的前提下，提高网络阈值参数对存储软错误的容忍能力，从而保证推理阶段的模型精度与系统可靠性。

3.2 神经元级的阈值容错

在神经元级阈值容错机制中，神经元仍采用统一的激活函数形式，但为每个神经元分配独立的激活阈值参数，以精细控制其输出上界，从而在局部层面抑制由软错误引发的异常激活值传播。公式(1)给出了这种神经元级有界ReLU激活函数（CappedReLU）[8]：

$$\mathrm{CappedReLU}(x)=\begin{cases}0 & \text{if} \quad x>\lambda_i \ , i\in[0,N]\\ x & \text{if} \quad 0<x<\lambda_i, i\in[0,N]\\ 0 & \text{if} \quad x\le 0\end{cases} \tag{1}$$

其中N表示神经元λ_i的数量，i表示第个神经元的阈值。在DNN推理中，当神经元的值大于λ_i时，便视其为潜在的软错误，直接将其屏蔽（设置为0）。这样做的原因在于，0可以作为一个安全的“故障抑制”信号，即通过将激活值归零，减少对后续层计算的影响。

3.3　OVSF基码生成机制与阈值重构

本研究采用正交可变扩频因子（OVSF）基码对神经网络的激活阈值进行变换域表示，即将阈值从原始参数空间映射到由一组正交二值基向量张成的坐标空间中。

对于模型中任一层的激活阈值张量（例如，某一卷积通道对应的二维阈值张量，形状为$H\times W$首先将其按固定顺序展平为一维向量$\boldsymbol{\theta}\in R^N$，其中$N=H\times W$。该向量作为后续OVSF正交展开与稀疏重构的基本处理单元。

在此基础上，阈值向量$\boldsymbol{\theta}$将被表示为一组OVSF基码的线性组合，其基码的生成方式及对应的阈值重构机制将在下文中分别予以说明。

3.4　OVSF基码的无状态生成

OVSF基码的生成基于哈达玛矩阵（Hadamard Matrix）的递归结构，其构造可通过西尔维斯特（Sylvester）构造法实现[25]。该方法从最小阶矩阵开始，通过递归方式生成更高阶的正交矩阵。

具体而言，阶数为$L=2^k$的哈达玛矩阵$\boldsymbol{H}_L$可由以下递推公式构造：

$$\boldsymbol{H}_1=[1],\quad \boldsymbol{H}_{2L}=\begin{bmatrix}\boldsymbol{H}_L & \boldsymbol{H}_L\\ \boldsymbol{H}_L & -\boldsymbol{H}_L\end{bmatrix} \tag{2}$$

通过上述递归构造，可以生成元素取值为±1、且行向量两两正交的哈达玛矩阵。例如，当$L=2$和$L=4$时，对应的哈达玛矩阵分别为：

$$\boldsymbol{H}_2=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix},\ \boldsymbol{H}_4=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix} \tag{3}$$

在本研究中，哈达玛矩阵$\boldsymbol{H}_L$的每一行向量被视为一条OVSF基码，用于构成阈值向量的正交表示基。由于该构造过程完全由递归规则确定，在推理阶段无需显式存储完整的基码矩阵，而可根据所需的基码长度L按需生成对应的基向量，从而实现OVSF基码的无状态生成。

3.5　基于OVSF基的阈值表示与重构

根据线性代数理论[26]，经过填充的阈值向量$\boldsymbol{\theta}'\in R^L$可以被一组正交的OVSF基向量$\{\boldsymbol{B}_i\}_{i=1}^{L}$完全表示，其形式为基向量的线性组合：

$$\boldsymbol{\theta}'=\sum_{i=1}^{L}\boldsymbol{\alpha}_i\boldsymbol{B}_i^{\mathrm{T}}=\boldsymbol{B}^{\mathrm{T}}\boldsymbol{\alpha} \tag{4}$$

其中，$\boldsymbol{B}$是以$\{\boldsymbol{B}_i\}$为行向量的$L\times L$的OVSF基码矩阵，$\boldsymbol{B}^{\mathrm{T}}=\boldsymbol{B}$。系数向量$\boldsymbol{\alpha}=[\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\ldots,\boldsymbol{\alpha}_L]^{\mathrm{T}}$是阈值向量$\boldsymbol{\theta}'$在OVSF域的表示，可以通过正交变换$\boldsymbol{\alpha}=\boldsymbol{B}\cdot\boldsymbol{\theta}'$计算得出。

在实际应用中，本方法并不存储完整的系数向量$\boldsymbol{\alpha}$，而是存储一个经过压缩的、仅包含K个最重要系数的向量及其对应的索引（具体选择方法见下一节）。在推理阶段，阈值的重构过程如下：

加载K个非零系数$\{\alpha_j\}$及其索引$\{j\}$。

根据预存的基码长度L，即时生成完整的$L\times L$OVSF基码矩阵$\boldsymbol{B}$。

利用索引$\{j\}$从$\boldsymbol{B}$中抽取对应的K个基向量，构成子矩阵$\boldsymbol{B}_{sel}$。

通过稀疏线性组合计算重构后的阈值向量：$\widehat{\boldsymbol{\theta}'}=\boldsymbol{B}_{sel}^{\mathrm{T}}\cdot\boldsymbol{\alpha}_{sel}$。

若进行了填充，则将重构向量$\widehat{\boldsymbol{\theta}'}$截断至原始长度$N$。

最终，将一维向量$\hat{\boldsymbol{\theta}}$恢复为其原始的张量形状（如$H\times W$），供后续神经网络层使用。

为了更直观地说明阈值的重构过程，下面给出一个简化示例:

假设某一卷积通道对应的阈值张量经展平后得到阈值向量

$$\boldsymbol{\theta}=\left[\theta_1,\theta_2,\theta_3,\theta_4\right]^{\mathrm{T}},$$

其长为$L=4$。根据3.2.1小节所述的西尔维斯特构造法，可生成对应的OVSF基码矩阵（哈达玛矩阵）：

$$\boldsymbol{H}_4=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}$$

在压缩阶段，阈值向量$\boldsymbol{\theta}$被展开到OVSF基上，得到系数向量

$$\boldsymbol{\alpha}=\boldsymbol{H}_4\boldsymbol{\theta}$$

假设经过能量（贡献度）筛选，仅保留第1和第3个基码对应的系数，即

$$\alpha_1=10,\quad \alpha_3=4$$

其余系数被置零。此时，压缩后仅需存储：

1.基码索引集合$\tau=\{1,3\}$

2.对应的非零系数$\{\alpha_1,\alpha_3\}$

在推理阶段，阈值的重构过程如下：

(1)根据预存的基码长度$L=4$，按需生成完整的OVSF基码矩阵H_4;

(2)根据索引集合τ,从H_4中选取对应的基向量，构成子矩阵

$$\boldsymbol{H}_\tau=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & 1 & -1 & -1\end{bmatrix}$$

(3)利用保留的系数进行线性组合，得到重构阈值向量

$$\begin{aligned}\hat{\boldsymbol{\theta}}&=\boldsymbol{\alpha}_1\cdot\boldsymbol{H}_{\tau,1}+\boldsymbol{\alpha}_3\cdot\boldsymbol{H}_{\tau,2}\\&=10\left[1,\,1,\,1,\,1\right]^{\mathrm{T}}+4\left[1,\,1,\,-1,\,-1\right]^{\mathrm{T}}\end{aligned}$$

(4)最终得到重构后的阈值向量

$$\hat{\boldsymbol{\theta}}=\left[14,\,14,\,6,\,6\right]^{\mathrm{T}}$$

该示例清晰地展示了在仅存储少量基码索引与系数的情况下，如何通过OVSF基码的在线生成与稀疏线性组合，高效重构完整的阈值向量。

这一“在线生成-重构”的机制，构成了本方法高效压缩神经元阈值的基础。

3.6 阈值压缩

在DNN的推理过程中，为每个神经元设定精确的激活阈值对于模型的容错性能至关重要。然而，直接存储这些高精度的阈值张量会带来巨大的内存开销。本文的解决方案是利用正交可变扩频因子（OVSF）基码对阈值进行变换域表示。

理论上，一个完整的阈值向量$\boldsymbol{\theta}\in R^N$可以由一个$N\times N$的OVSF矩阵$\boldsymbol{B}$和其对应的系数向量$\boldsymbol{\alpha}\in R^N$无损重构，即$\boldsymbol{\theta}=\boldsymbol{B}^{\mathrm{T}}\boldsymbol{\alpha}$。然而，若直接存储完整的系数向量$\boldsymbol{\alpha}$，其存储需求与直接存储原始阈值相当，并未能有效缓解存储压力。但DNN中的神经元阈值通常表现出高度的结构化特性，即在同一卷积通道内，阈值在空间分布和数值上具有较强相关性，并可在正交变换域中由少量基向量进行有效表示。这意味着其在OVSF变换域中的‘能量’往往集中在少数几个关键的基向量上。这里的‘能量’指的是阈值向量在OVSF基码空间中的投影值，即在正交变换后，阈值信息集中于少数几个基向量上的现象。简单来说，“能量”是对阈值向量在变换域中信息集中度的量化，反映了在压缩过程中，哪些基向量对阈值的重构贡献最大。因此，通过保留这些能量集中度高的基向量及其系数，可以在有效保持阈值信息的同时大幅减少存储需求。

本文为便于对比，采用统一压缩因子ρ对所有通道进行处理。然而，由于各卷积层及不同通道在模型中的敏感度和贡献度存在差异，在结构上可以进行自适应压缩率分配。具体而言，可以根据通道重要性指标（如敏感度分析、剪枝得分或容错影响度评估）为关键通道分配较大的ρ值（即保留更多基码），而对冗余或影响较小的通道采用较小的ρ值以获得更高压缩率。

由于OVSF压缩与重构过程在各通道之间完全独立，该自适应策略不会增加额外的结构复杂度，仅需在压缩阶段调整各通道的基码选择数量即可实现。因此，本文方法在理论上具备层间与通道间压缩率自适应调整的能力，为在不同资源约束条件下进行精细化部署提供了灵活性。

图 2 展示了通过变换与选择操作实现数据压缩的过程，通过对矩阵 □ 和向量 □ 进行选择性降维，在保留关键信息的同时降低计算复杂度。

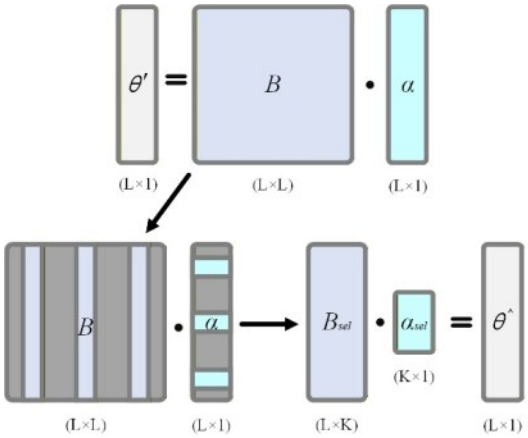

图 2　阈值压缩与重构

3.6.1　基于能量的基码选择与迭代优化算法

为了在给定压缩因子ρ的情况下，从完整的OVSF基码中选择一个最优的子集（大小为 $K=\lfloor\rho\cdot N\rfloor$）以最小化重构误差，本文设计了如算法一所示的迭代优化流程。该算法的核心思想是交替进行系数优化和基码集的再选择，从而逐步逼近最优解。算法一给出了在给定压缩因子条件下选择 OVSF 基码及其对应系数的迭代优化过程。其核心思想是在OVSF正交变换域中，通过交替优化基码子集与展开系数，寻找一组在有限基码数量约束下对阈值重构贡献最大的基向量。

在初始化阶段，算法首先将阈值向量投影到完整的OVSF基上，计算所有展开系数，并依据系数幅值（能量大小）选取贡献最大的K个基码作为初始基码集合。该步骤旨在快速获得一个合理的初始解。

在随后的迭代过程中，算法在固定当前基码集合的前提下，通过最小二乘方法重新估计对应的最优系数，以最小化当前基码子集下的重构误差；随后，根据更新后的系数重新评估各基码对阈值重构的贡献，并据此更新基码选择集合。通过在“系数优化”和“基码再选择”之间交替迭代，算法能够逐步逼近在给定基码数量约束下的最优稀疏表示。

该迭代策略在保证重构精度的同时，有效避免了单次能量截断可能带来的次优选择，使得在较高压缩率下仍能保持稳定的阈值重构性能。

3.6.2　重构质量与压缩效果

通过上述方法，DNN激活层可以用K个系数和对应的K个索引来代替存储完整的N维阈值向量。在推理时，通过加载这些稀疏系数即可重构出近似的阈值向量$\hat{\boldsymbol{\theta}}$。重构质量可以用相对误差率$\varepsilon(\rho)$来衡量：

$$\varepsilon(\rho)=\frac{\left\|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\right\|^2}{\left\|\boldsymbol{\theta}\right\|^2} \tag{5}$$

其中，$\hat{\boldsymbol{\theta}}$是由K个选定基向量重构出的阈值。此误差率直接受到压缩因子 ρ 的影响。ρ越小，内存占用减少越明显，但重构误差通常会相应增大。

实验分析表明，由于阈值能量在OVSF域的高度集中特性，即使采用较高的压缩率也能保持极高的重构精度。例如：

当压缩因子 $\rho=0.5$ 时，即使用50%的OVSF基码，重构误差率$\varepsilon(\rho)$可低于0.05，意味着保留了超过95%的原始信息；

当压缩因子进一步降低至 $\rho=0.25$ 时，误差率仍可维持在较低水平，保留约 90% 以上的信息。

这种高效的压缩能力使得在大幅降低模型存储需求的同时，能够最大限度地保留对模型推理

算法一：OVSF基码与系数选择

输入：$\boldsymbol{\theta}$：原始阈值向量，维度为N

$\boldsymbol{B}$：完整的$N\times N$ OVSF基码矩阵

K：目标选择的基码数量

$Iters$：迭代优化的总次数

输出：idx_{best}：最终选定的K个基码的索引

α_{best}：与选定基码对应的K个最优系数

流程：

1： // 1. 初始化

2： $\boldsymbol{\alpha}_{full}\leftarrow \boldsymbol{B}\cdot\boldsymbol{\theta}$

3： $idx_{current}\leftarrow SelectTopKIndices(|\boldsymbol{\alpha}_{full}|,K)$ /

4： $min\_error\leftarrow\infty$

5： $idx_{best},\boldsymbol{\alpha}_{best}\leftarrow null,null$

6： // 2. 迭代优化

7： **for** $i\leftarrow 1$ **to** $Iters$ **do**

8： $\boldsymbol{B}_{current}\leftarrow \boldsymbol{B}[idx_{current},:]$

9： // 2a. 系数优化

10： $\boldsymbol{\alpha}_{current}\leftarrow SolveLeastSquares(\boldsymbol{B}_{current}{}^{\mathrm{T}},\boldsymbol{\theta})$

11： // **2b. 误差评估与更新**

12： $\hat{\boldsymbol{\theta}}\leftarrow \boldsymbol{B}_{current}{}^{T}\cdot\hat{\boldsymbol{\alpha}}_{current}$

13： $error\leftarrow\left\|\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}\right\|_2$

14： **if** $error<min\_error$ **then**

15： $min\_error\leftarrow error$

16： $idx_{best}\leftarrow idx_{current}$

17： $\boldsymbol{\alpha}_{best}\leftarrow\boldsymbol{\alpha}_{current}$

18： **end if**

19： // **2c. 基码再选择**

20： **if** $i<Iters$ **then**

21： $\boldsymbol{\alpha}_{full}\leftarrow\boldsymbol{B}\cdot\boldsymbol{\theta}$

22： $idx_{current}\leftarrow SelectTopKIndices(\left|\boldsymbol{\alpha}_{full}\right|,K)$

23： **end if**

24： **end for**

精度至关重要的阈值信息。

对给定压缩因子ρ，保留系数数量$K_l=|\rho\times N_l|$，对应的能量保存比例为：

$$\eta(\rho)=\eta_{K_1}=\frac{\sum_{i=1}^{K_1}\boldsymbol{\alpha}_i{}^2}{\sum_{i=1}^{N_l}\boldsymbol{\alpha}_i{}^2} \tag{6}$$

该能量保存比例表示了在给定压缩因子下，保留基码对原始阈值信息的覆盖程度，为后续重构精度与容错性能的分析提供了直观解释。

3.7 OVSF基码系数矩阵的分通道独立存储

在本方法的框架中，阈值的压缩与重构是在每个卷积通道上独立进行的。这一设计旨在最大化计算效率、重构精度与系统设计的灵活性。与之相对，将多个通道甚至整个网络层的阈值合并处理的粗粒度策略，虽然在数学表示层面是可行的，但在实际应用中，该策略会导致变换维度和计算复杂度显著增加，同时混合不同通道的阈值统计特性，削弱阈值在变换域中的能量集中性，从而降低压缩效率并增加实现开销。聚合处理要求操作的哈达玛矩阵维度随处理单元数量指数级增长，同时，将统计特性各异的阈值向量混合会造成变换域中的能量分散，从根本上削弱了压缩性能。

因此，本文采用的分通道独立处理策略，其核心优势体现在以下三个方面：

1. 保持信号同质性以实现最优压缩：OVSF变换之所以能高效压缩，其数学基础在于阈值向量在正交基下呈现出明显的能量集中（sparsity in orthogonal basis）特性。单个卷积通道内的神经元通常协同学习一类特定的局部特征（如边缘、纹理），因此它们的激活阈值在数值分布上具有较强的同质性。独立处理保留了这种重要的局部结构信息，使得阈值向量在变换到OVSF域后，其能量能自然地集中在少数基向量上，从而仅需少量系数即可实现高质量重构，达到最优的压缩效果。

2. 计算可行性与资源效率：通过将大规模的压缩问题分解为大量小规模的独立子问题，规避了计算复杂度的指数增长。每个通道的阈值向量维度小且固定，对应的OVSF变换和系数求解的计算开销极低，完全在边缘设备的算力承受范围之内。这使得整个框架在保持高压缩比的同时，具备了低延迟和低功耗的特性。

3. 支持自适应的精细化压缩策略：分通道处

理赋予了本方法的框架极高的灵活性，支持根据不同通道对模型整体性能的重要性，为其分配差异化的压缩率。例如，通过网络剪枝分析或敏感度分析，可以识别出对模型分类结果至关重要的“关键通道”。对这些通道，本文采用较低的压缩率（即较高的ρ值）来保证其阈值重构的最高精度；而对于冗余或次要的通道，则可应用更高的压缩率（即较低的ρ值）以最大化存储节省资源。这种自适应的资源分配策略，使本方法能够在全局层面实现容错性能与存储开销的最优化平衡。

4　实验设计与评估

本节将介绍用于评估本文基于OVSF的阈值压缩框架的实验设置，包括所用的模型、数据集、故障注入机制，以及各个方法的对比。

4.1　实验设置

为全面评估本文所提出方法的有效性与可部署性，实验分别在通用计算平台和资源受限的边缘侧硬件平台上进行。实验中选用的网络结构与数据集用于模拟典型的边缘推理任务，其主要目的是验证本文方法在推理阶段面对存储软错误时的有效性与低开销特性。

在算法性能评估阶段，本文主要采用基于GPU的通用计算平台，对模型的分类精度、容错性能以及阈值压缩效果进行验证。具体而言，相关实验在搭载NVIDIA GeForce RTX 3060 GPU的工作站上完成，软件环境为TensorFlow 2.x。该平台具有充足的计算资源与存储带宽，能够为不同方法提供公平、稳定的对比环境。

此外，为验证所提出OVSF阈值压缩方法在实际边缘侧部署场景中的可行性与潜在优势，本文进一步在FPGA平台上实现并测试了关键推理模块，相关实验设置与结果将在第4.3.3小节中详细介绍。

故障模型：为模拟硬件瞬态故障（软错误），本文采用了在容错研究中广泛使用的权重比特翻转模型。在每次推理前，以预设的故障率（Fault Rate, FR，范围从1×10^{-7}到5×10^{-5}）在模型的卷积层和全连接层的权重参数中随机选择位置，并对其32位浮点数进行一次单比特翻转（Single-Bit Flip, SBF）。本文聚焦于权重SBF故障模型，这是已有工作中最常用且最具破坏性的基准模型，其它类型故障（如激活翻转）可视为该模型的变体。

4.2　对比方法

为确保对比的公平性，实验围绕四种方法展开：Orig（无保护的基准模型）、Clip-Act（层级统一阈值）、FitAct（神经元级微调阈值）以及OVSF（本文提出的压缩方法）。所有受保护方法的准备流程均始于一个统一的、预训练好的采用标准ReLU激活函数的模型（即Orig）。首先通过在测试集上推理，收集每个神经元激活值作为初始阈值。在此基础上，针对不同方法进行最终处理：对于FitAct，冻结网络权重，对这些统计阈值进行10轮微调以获得最优性能；对于Clip-Act，通过训练得到每层阈值的最佳值作为其统一阈值；而对于OVSF，直接将微调后的统计阈值作为原始数据，应用本文算法一进行压缩与重构。为保证结果稳定性，每种故障率配置均重复独立实验50次并报告其平均性能。

4.3　实验结果

4.3.1　压缩率对重构质量的影响分析

在本文提出的方法中，压缩因子ρ是一个关键参数，它直接决定了用于重构阈值的OVSF基码数量$K(K=\lfloor\rho\cdot N\rfloor)$，从而在模型的存储开销与重构精度之间进行权衡。为了量化这种权衡关系，对不同ρ值下的阈值重构质量进行了一系列实验。本文采用阈值保真度分数（Threshold Fidelity Score, TFS）、平均绝对误差（Mean Absolute Error, MAE）和均方根误差（Root Mean

Square Error, RMSE）作为核心评估指标。

从图 3 以直观地看出，随着 ρ 的降低（即压缩程度的提高），阈值保真度（TFS）平滑下降，而两种误差度量（MAE 和 RMSE）则相应上升。这符合预期，因为使用更少的基码必然会导致信息损失。

值得注意的是，TFS 曲线的下降斜率在 ρ 值较高时（例如从 1.0 到 0.5）相对平缓，表明在此区间内，可以用较小的精度代价换取显著的存储压缩。例如，当 ρ= 0.5 时，本文虽然仅使用了 50% 的基码，但 TFS 仍高达 96.82%，与无损重构（ρ= 1.0）时的 98.96% 相比，保真度仅下降了约 2 个百分点。

当 ρ=0.25 时，TFS 依然保持在 94.04% 的较高水平。这充分证明了本方法的有效性：由于 DNN 阈值在 OVSF 变换域的能量高度集中特性，只需保留少量关键的基码，便可精确地重构出原始阈值的大部分信息。

需要进一步说明的是，阈值保真度（Threshold Fidelity Score, TFS）用于衡量压缩后阈值与原始阈值之间的相似程度，其数值越接近 100%，表示重构阈值与原始阈值越相似。在工程意义上，TFS 并不存在统一的“绝对标准阈值”，其可接受范围取决于重构误差是否会对模型最终推理精度产生显著影响。

从实验结果可以观察到，当 ρ = 0.5 时，TFS 为 96.82%，对应的平均绝对误差和均方根误差均保持在较低水平。更重要的是，在后续的容错性能对比实验中（见图 3），该压缩率下的 OVSF 模型在不同故障率条件下的分类准确率曲线与未压缩 FitAct 模型几乎重合。这说明，尽管阈值在数值层面存在约 3% 左右的重构偏差，但该偏差并未对模型的容错能力和最终推理精度产生可观测影响。

实验还表明，当 TFS 保持在约 94% 以上时（例如 ρ = 0.25 时 TFS=94.04%），模型的容错性能仍与未压缩方案保持一致。这意味着，在本文所研究的边缘推理场景下，当阈值保真度维持在 94% – 97% 以上区间时，压缩误差对实际分类结果的影响可以忽略不计。因此，TFS 可被视为压缩质量的一个中间度量指标，而模型最终的分类精度与容错曲线则构成对压缩可接受性的最终

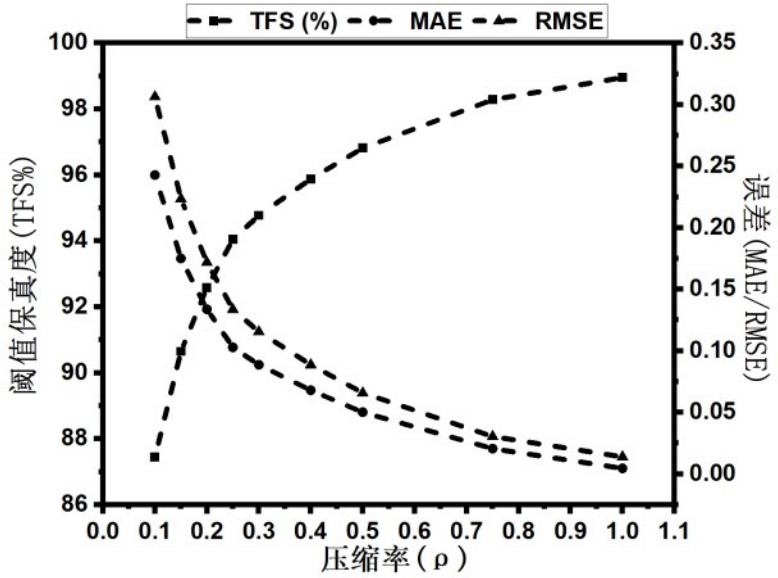

图 3　不同压缩率下的 OVSF 阈值重构质量

判据。

在本文方法中，OVSF 基码长度与对应阈值向量的维度一致，以保证正交展开的完备性。因此，不同卷积层由于通道尺寸不同，其对应的 OVSF 码长也不同。

从理论上看，压缩效果主要取决于阈值在 OVSF 变换域中的能量集中程度，而非基码长度本身。只要阈值向量在正交域中呈现出稀疏性特征，即其能量集中于少数几个基向量上，则在不同码长条件下，压缩趋势应保持一致。

实验过程中对不同层（对应不同阈值维度与码长）的统计结果表明，在各类码长条件下，TFS 随压缩因子 ρ 变化的趋势基本一致：当 $\rho \geq 0.25$ 时，均可维持较高的重构保真度区间。不同码长之间仅在具体数值上存在轻微差异，而压缩趋势和容错性能保持规律性一致。这说明图 2 所呈现的结果并非依赖于某一特定 OVSF 码长，而是源于阈值在正交变换域中的结构化稀疏特性。

综上所述，实验结果表明，研究者们可以根据具体的应用场景和资源限制，在存储开销和容错精度之间灵活选择一个合适的 ρ 值。对于大多数边缘计算场景，当 ρ= 0.25 时，所提出的方法在显著降低阈值存储开销的同时，仍能够保持较高的阈值重构质量和稳定的容错性能，且该结论在不同码长条件下具有一致性。

4.3.2　容错性能对比

本文在不同故障率下对比了四种容错方法在 VGG-16 和 AlexNet[27]模型上的性能表现。图 4 展示了在 CIFAR-10 数据集上的实验结果。

如图 4 所示，随着故障率的逐渐升高，四种模型的分类准确率均呈下降趋势。当故障率超过 1×10^{-6} 时，Orig 模型和 Clip-Act 模型的准确下降至约 10%，几乎等同于随机猜测水平。相比之下，采用神经元级阈值容错机制的 FitAct 模型和 OVSF 模型在高故障率下仍保持较强的鲁棒性，其容错性能显著优于 Clip-Act 模型。

值得注意的是，在不同压缩率配置下，OVSF 模型的性能曲线与 FitAct 模型高度重合。这表明，尽管 OVSF 方法对阈值进行了压缩处理，其在容错性能上与未压缩的 FitAct 模型几乎无差别，验证了所提压缩机制在保持模型可靠性的同时显著降低存储开销的有效性。

4.3.3　运行时间与内存空间开销

为了存储大量神经元的阈值，应用 FitReLU 的模型，额外需要 0.877Mb 的存储空间，这对资源受限的边缘设备来说是不可接受的。与此相

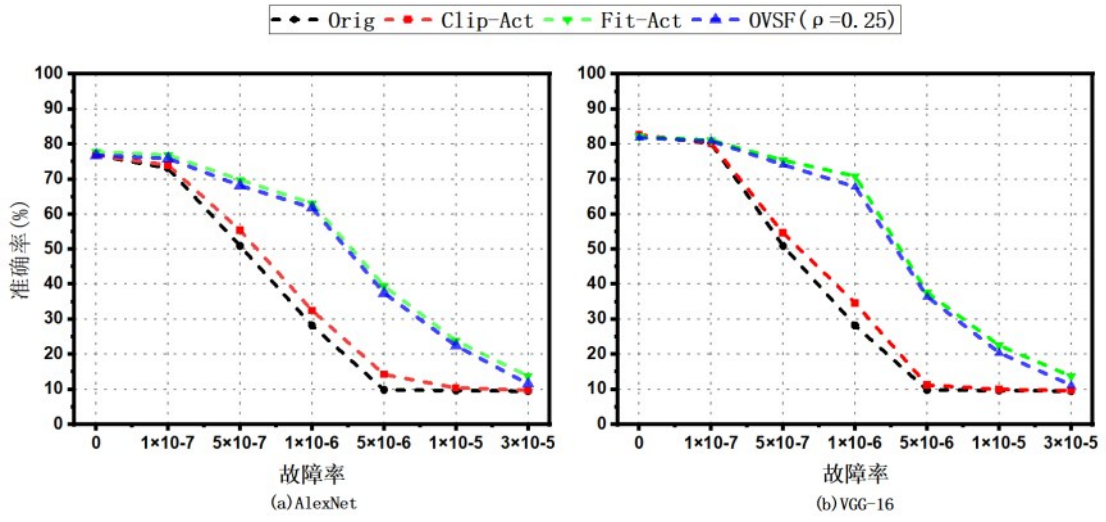

图 4　VGG-16、AlexNet 模型容错性能对比

比，采用OVSF阈值压缩的CappedReLU DNN模型，减少了约75%的存储空间占用，同时仅带来了约3.47%的精度损失与10%左右的推理时间增加。推理时间的轻微增加主要来源于OVSF方法在推理阶段引入的在线阈值重构开销。与FitAct直接从存储中加载完整阈值不同，OVSF方法需要根据预存的基码索引和稀疏系数，在推理过程中按需生成OVSF基码并执行稀疏线性组合以重构完整阈值。尽管该过程仅涉及加法和符号操作，其计算开销仍会反映在推理关键路径中。在本文所使用的GPU平台上，由于显存带宽充足，阈值存储量减少所带来的内存访问时间收益相对有限，因此总体推理时间表现为约10%的增加。需要指出的是，在片外存储访问代价更高、带宽受限的边缘设备或FPGA平台上，额外计算开销可被显著减少的参数传输时间所抵消，甚至带来整体延迟的降低，这一点也在后续硬件实验中得到了验证。

表 1　存储开销分析

	阈值存储（MB）		推理时间（ms）	
方法\模型	VGG-16	AlexNet	VGG-16	AlexNet
Orig	0	0	2.070	1.230
Fit-Act	0.877	0.810	2.319	1.396
OVSF(ρ=0.25)	0.219	0.203	2.547	1.548

4.3.4　FPGA边缘侧验证实验

为进一步评估本文方法在资源受限边缘设备上的实际部署效果，本节基于前述实验平台，在Xilinx Zynq-7020 FPGA上选取VGG16模型[19]典型卷积层实现其推理流程，并对FitAct与OVSF两种阈值容错机制在该代表性计算单元上的时延与内存带宽开销进行对比分析。该实验设置旨在于受限硬件资源条件下验证所提出阈值压缩与在线重构机制的可实现性与性能影响，其结果可为完整网络在FPGA等边缘平台上的部署提供参考。

实验重点关注阈值参数在边缘侧的传输代价与重构开销，旨在验证OVSF阈值压缩机制在真实硬件条件下对带宽利用率和整体推理延迟的影响。

实验中，FPGA上复现了OVSF阈值压缩与解码的关键步骤，并对FitAct方案的完整阈值传输过程进行了同步测量。表2表示了两种方法在不同压缩因子(ρ = 0.5, 0.25, 0.1)下的实验结果。从实验结果可见，尽管OVSF方案在阈值重构阶段引入了轻微的计算开销（约10%），但其显著减少了参数传输时间与内存带宽占用。当压缩因子为ρ = 0.25时，总推理时延与FitAct基本持平，而参数传输时间降低约70%。这一结果表明，OVSF编码机制在硬件实现中同样具备边缘侧时延与带宽优化能力，为其在低功耗设备上的部署提供了进一步验证。

表 2　不同模型在FPGA平台上的实现

指标\方法	FitAct	OVSF (ρ =0.5)	OVSF (ρ =0.25)	OVSF (ρ =0.1)
参数量	256 KB	160 KB	80 KB	32 KB
参数传输时间	0.674 ms	0.421 ms	0.211 ms	0.084 ms
计算时间	3.070 ms	3.970 ms	3.550 ms	3.290 ms
总延迟	3.744 ms	4.391 ms	3.761 ms	3.374 ms

为进一步评估所提方法的硬件友好性，本文对FPGA实现的资源利用情况进行了统计。与未压缩的FitAct方案相比，OVSF（ρ = 0.25）方案在BRAM、DSP、FF和LUT资源上均存在一定程度的增加，其中OVSF方案的BRAM使用数量由10增加至12，LUT利用率由35.8%提升至39.1%。该资源增长主要来源于在线重构模块中增加的正交展开与系数选择逻辑。需要指出的是，本文方法的设计目标在于降低阈值存储与传输开销，并增强容错能力，而非减少逻辑资源占用。在保持较低存储需求的同时，额外增加的逻辑资源开销处于可接受范围内。

从整体资源利用率来看，各类资源占用率仍处于合理区间，未对FPGA部署产生显著压力。

这表明所提出方法在实现存储压缩与容错增强的同时，具备良好的硬件可实现性与工程可行性。

5　总结

本文为解决DNN在资源受限的边缘设备上部署高精度容错机制时所面临的巨大存储开销挑战，设计并验证了一个基于正交可变扩频因子（OVSF）编码的阈值压缩框架。本文通过引入一套结合了迭代系数优化与即时重构的创新机制，在保持神经元级阈值容错性能的同时，将阈值的存储需求有效降低，打破了高可靠性与低资源占用之间的固有矛盾。

本研究的核心成果在于，证明了将通信领域的OVSF编码理论应用于神经网络阈值表示的可行性与高效性。实验结果充分证实了本方法的优势：在VGG16模型上，当压缩因子$\rho=0.25$时，本方法将阈值存储开销从FitAct所需的0.877MB减至0.219MB（压缩率达75%）。在实现如此高压缩比的同时，其容错性能在多种故障率下均与未压缩的FitAct方案持平（例如，在1×10^{-6}故障率下，准确率分别为68.86%和68.83%），且远超其他基准方法。此外，本文在FPGA上对VGG16单卷积层的测试进一步验证了OVSF方法在边缘侧的可实现性与优化潜力，其在保持容错性能的同时，显著降低了阈值传输带宽与推理延迟，展示了面向实际硬件部署的可行路径。

总体而言，本文提出的基于OVSF编码的容错方法，不仅为边缘AI的可靠性部署提供了一个实用且高效的解决方案，更在理论层面建立了通信编码与神经网络容错之间的桥梁，为推动高可靠性人工智能在自动驾驶、工业物联网等安全关键领域的广泛应用，开辟了新的技术路径。

参考文献：

[1] BOJARSKI M, DEL TESTA D, DWORAKOWSKI D, et al. End to end learning for self-driving cars[J/OL]. arXiv: 1604.07316, 2016.

[2] ESTEVA A, KUPREL B, NOVOA R A, et al. Dermatologist-level classification of skin cancer with deep neural networks[J]. Nature, 2017, 542(7639): 115-118.

[3] LI G, PATTABIRAMAN K, CHERNIARUBAN S C, et al. Understanding the vulnerability of deep neural networks to soft errors[J]. ACM Transactions on Embedded Computing Systems (TECS), 2018, 17(2): 25.

[4] MUKHERJEE S. Soft errors in modern electronic systems[C]// IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems. 2008: 3-11.

[5] BAUMANN R. Soft errors in advanced computer systems[J]. IEEE Design & Test of Computers, 2005, 22(3): 258-266.

[6] HE Y, ZHU C, SAVARIA Y, et al. Evaluating the robustness of deep neural networks against bit-flip errors[C]//2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS). 2019: 159-164. DOI: 10.1109/IOLTS.2019.8854389.

[7] KOREN I, KRISHNA C M. Fault-tolerant systems[M]. San Francisco: Morgan Kaufmann, 2020.

[8] GHAVAMI B, SADATI M, FANG Z, et al. FitAct: error resilient deep neural networks via fine-grained post-trainable activation functions[C]//2022 Design, Automation & Test in Europe Conference & Exhibition (DATE). 2022: 1239-1244. DOI: 10.23919/DATE54114.2022.9774635.

[9] CHEN Z, LI G, PATTABIRAMAN K. A low-cost fault corrector for deep neural networks through range restriction[C]//Proceedings of 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks. 2021: 1-13. DOI: 10.1109/DSN48987.2021.00018.

[10] CHEN H, LI G, KHAN H M, et al. CLIP-Act: an accurate and efficient fault tolerance scheme for deep neural networks[C]// 2019 IEEE 37th International Conference on Computer Design (ICCD). 2019: 361-369.

[11] ZHANG Z, LI G, PATTABIRAMAN K, et al. A survey on fault tolerance techniques for deep neural networks[J]. Journal of Computer Science and Technology, 2021, 36(4): 906-930.

[12] 郭宇辉,闫亚旗,付韬等.边缘算力发展态势分析[J].电信科学,2025,41(11):1-13. DOI：10.11959/j.issn.1000-0801.2025195.

[13] KOOPMAN P. Tesla's “Full Self-Driving” hardware (and other production autonomy claims) [EB/OL]. (2017-02-06) [2025-12-12].

[14] MAHMOUD A, AKRAMI M, LI G, et al. Selective node hardening for fault-tolerant deep neural networks[C]//2019 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT). 2019: 1-6.

[15] ZHOU C, DU X, YAN M, et al. SAR: Sharpness-aware minimization for enhancing DNNs' robustness against bit-flip errors [J]. Journal of Systems Architecture, 2024, 138: 103284.

[16] XU H, LIAO L, LIU X, et al. Fault-tolerant deep learning inference on CPU–GPU integrated edge devices with TEEs[J]. Future Generation Computer Systems, 2024, 161: 404-414.

[17] ZHENG W, XU B, GU J, et al. SAVE: Software-implemented fault tolerance for model inference against GPU memory bit flips[C]//Proceedings of the 2025 USENIX Annual Technical Conference (USENIX ATC). 2025.

[18] XUE X, LIU C, MIN F, et al. ApproxABFT: Approximate algorithm-based fault tolerance for neural network processing [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2025 (Early Access).

[19] XUE X, LIU C, MIN F, et al. Adaptive soft error protection for neural network processing[EB/OL]. arXiv:2407.19664v2, 2025.

[20] VITERBI A J. CDMA: Principles of spread spectrum communication[M]. Reading: Addison-Wesley, 1995.

[21] LENG J, ZHANG S, LI G, et al. OVSF-inspired compressing of convolutional neural networks[C]//Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI). 2018: 2383-2389.

[22] PAN H, HAMDAN E, ZHU X, et al. Multi-channel orthogonal transform-based perceptron layers for efficient ResNets[EB/OL]. arXiv:2303.06797v2, 2024.

[23] HAMDAN E, CETIN A E. HTMA-Net: Towards multiplication-avoiding neural networks via Hadamard transform and in-memory computing[EB/OL]. arXiv:2509.23103v2, 2025.

[24] KIM S, SHIN J, WOO S, et al. HOT: Hadamard-based optimized training[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2025.

[25] HORADAM K J. Hadamard matrices and their applications [M]. Princeton: Princeton University Press, 2007.

[26] STRANG G. Introduction to linear algebra[M]. 5th ed. Wellesley: Wellesley-Cambridge Press, 2016.

[27] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[C]//Advances in Neural Information Processing Systems 25 (NIPS 2012). 2012: 1097-1105.

[作者简介]

靳松（通讯作者），1977年生，男，副教授，现为华北电力大学硕士生导师。研究方向：硬件安全、深度学习安全。Email：jinsong@ncepu.edu.cn

武炳硕，2000年生，男，现为华北电力大学硕士研究生。研究方向：硬件安全、深度学习安全。Email：15203709360@163.com