面向神经网络推理的网络引导乘加单元自动综合

Zhouyang Wu、Hongyang Pan、Chuyu Wang、Ji Zhuang、Keren Zhu*、Xin Zhao 和 Xuan Zeng*复旦大学，中国上海

arXiv:2611.31986v1 [cs.AR] 24 Nov 2026

摘要——在神经推理加速器中，乘加运算（MAC）数据通路占据了大部分电路面积，并决定了关键路径延迟；算子级数值格式则同时决定硬件开销和量化误差。然而，硬件成本与端到端网络精度之间并不存在简单直接的映射关系：逻辑综合会以高度非线性的方式转换数据通路，而数值误差则会在各层间传播并相互耦合。为解决这一脱节问题，我们提出了 NAMSI，这是一种面向神经网络推理的自动化网络感知 MAC 综合框架。NAMSI 使用格式–分布表示（FD）表征异构数据格式，并利用拓扑感知精度与硬件成本建模（TaH）——通过实际综合后的面积–延迟乘积（ADP）测量进行校准——联合估计相互耦合的全网络精度和电路成本。此外，我们引入可逆格搜索（RLS），以便在指数级扩大的设计空间中高效搜索。在 ImageNet-1K 上对五种神经网络进行的评估表明，在训练后量化（PTQ）条件下，与 FP32 相比，NAMSI 可将综合后 MAC ADP 降低54.2%–60.4%，同时保留至少99% 的全精度 Top-1 准确率，在硬件效率和精度保持方面均优于基线方法。

索引词—**算术电路综合、软硬件协同设计、神经网络推理**

一、引言

低精度乘累加（MAC）计算是高效神经网络推理的核心，现有研究主要沿着两条路径推进。一条路径通过训练、校准和重构 [1]–[7], 以及受资源约束的混合精度分配 [8]–[12], ，为现有硬件优化量化，同时将底层算术电路视为固定不变。另一条路径则开发低精度数值表示 [13]–[21], 并配合专用数据通路、推理内核和乘加架构 [22]–[25] ，以提升硬件和系统效率。然而，这两条路径都未将端到端网络训练后量化（PTQ）的

精度与 MAC 综合成本纳入同一个优化循环。

这种解耦使显著的精度-成本收益未能实现，因为算子数据格式共同决定了软件侧的跨层数值误差，以及硬件侧综合后的数据通路结构。首先，由于不同网络和层之间的算子张量分布与工作负载占比差异显著，量化策略很难直接迁移到不同网络。如图1(a)所示，

*通讯作者：Keren Zhu（krzhu@fudan.edu.cn）和 Xuan Zeng（xzeng@fudan.edu.cn）。

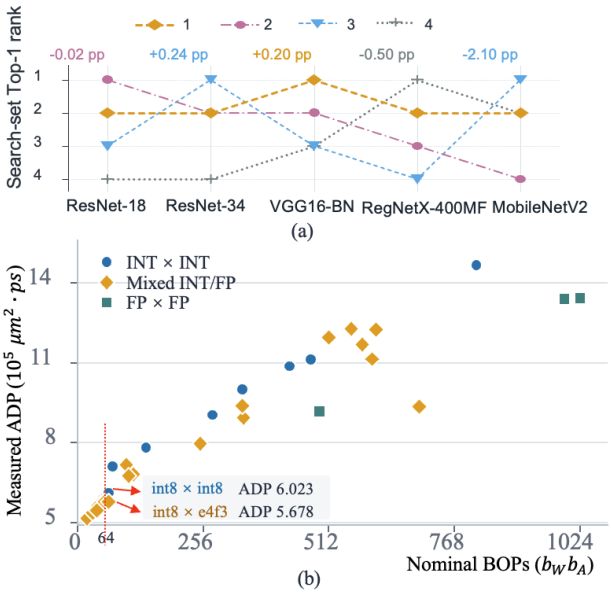


图1. (a) 不同网络中各量化方案的 Top-1 准确率排名。百分点（pp）表示同一搜索集上排名最高的方案与对应 FP32 Top-1 基线之间的差值。(b) 统一时钟周期下，实测的综合后 MAC ADP 与标称BOP数（$b_W b_A$）的关系。

不同量化策略的相对精度排名会因网络而异。因此，需要采用网络感知方法，根据网络特定特征自适应地选择量化方案。其次，MAC综合成本无法通过简单的代理模型可靠地估算，因为标称位宽无法准确反映异构整数和自定义浮点乘加运算的真实硬件成本。如图1(b)所示，具有相同位操作数（$b_W b_A$）的格式对，在综合后硬件成本上可能出现非线性差异，这是由结构差异造成的。由于相同的异构格式在不同网络拓扑中的精度排名各异，且逻辑综合后呈现非线性的面积-延迟权衡，因此必须在单一闭环框架中协同优化网络级格式分配与底层MAC电路综合。

在训练后量化下实现这种综合在环定制面临两项根本挑战。**C1.** 全网络 PTQ 评估和门级MAC综合会产生很高的运行时开销，而跨层误差耦合使得无法将指数级设计空间分解为彼此独立的

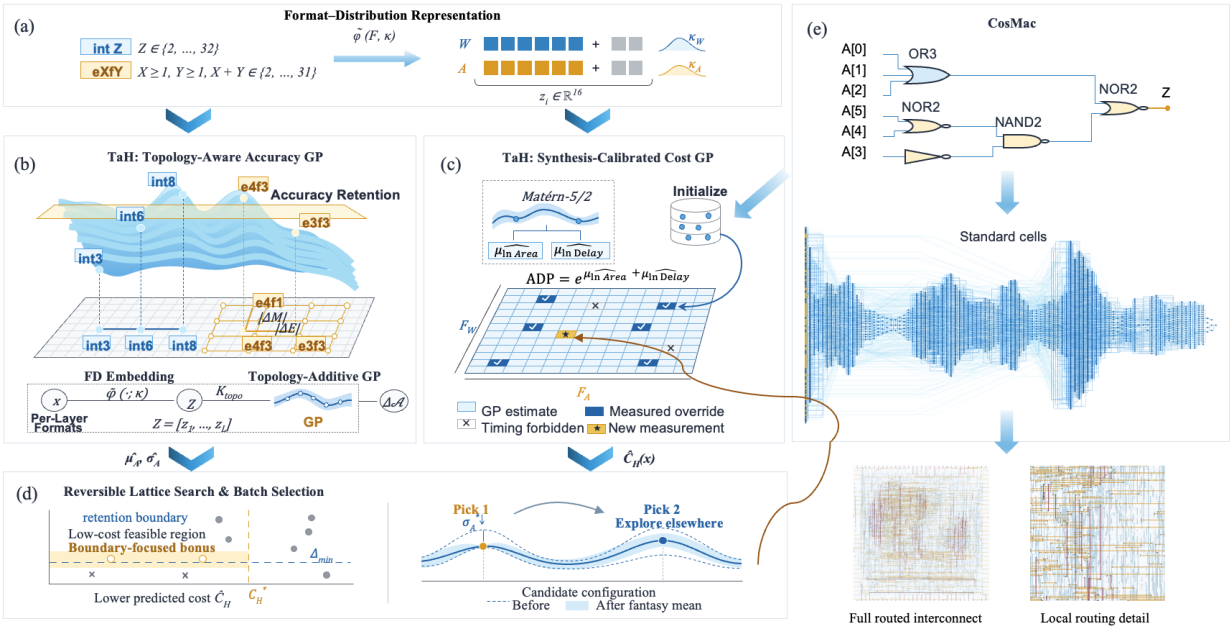


图2。NAMSI概览。

算子优化。**C2.** 异构格式在这一离散空间中不存在统一的位宽排序，因此会导致陡峭的精度悬崖 [13] 和非单调的硬件成本景观。

为应对这些挑战，我们作出以下贡献：

- 我们开发了NAMSI，这是一种端到端的网络感知乘加运算综合框架，将参数化异构乘加运算综合集成到精度约束下的全网络训练后量化优化中。
- 我们提出了拓扑感知精度与硬件成本模型（TaH），将计算图稀疏高斯过程（GP）与综合校准成本高斯过程相结合，以联合预测跨层耦合精度和综合后ADP。
- 我们设计了格式–分布表示（FD），将格式属性与张量统计量相结合，以刻画精度悬崖，并采用可逆格搜索（RLS）高效探索无序且非单调的格式空间。
- 我们在五个ImageNet-1K骨干网络上，结合7纳米综合和布局布线后的乘加运算实现对NAMSI进行了评估；在冻结的训练后量化条件下，在保持 $\geq$ 99%精度的同时，相较于FP32实现了54.2%–60.4%的MAC ADP降低。

II. 预备知识

A. 数据格式

数值数据格式规定了有限位序列如何表示一个数。其位宽决定了可用编码的数量，而编码规则则决定了可表示值的范围和间隔。整数格式提供间隔均匀的数值，而浮点格式类似于科学计数法，使用符号位、用于设定数量级的指数，以及用于细化数值的尾数。在位宽固定时，为指数分配更多位数会扩大动态范围，而为尾数分配更多位数则会提高精度。

B. 问题表述

我们将网络感知MAC综合表述为一个约束优化问题

$$\begin{aligned} \min_{x\in\mathcal{X}} \quad & C_H(x) \\ \text{s.t.} \quad & \tau\mathcal{A}_{\mathrm{FP}} - \mathcal{A}(x) \le 0, \\ & x \in \Omega_{\mathrm{num}}, \\ & x_i \in \Omega_{\mathrm{hw}}, \quad i = 1, \ldots, L. \end{aligned} \tag{1}$$

其中，$x = (x_1, \ldots, x_L)$指定了 L 个 MAC 运算符的权重–激活格式对 $x_i = (F_{W,\,i},\ F_{A,\,i})$，其中 $\mathcal{X} = (\mathcal{F}^2)^L$。目标 $C_H(x) = \sum_{i=1}^{L} w_i\,\mathrm{Area}(x_i)\mathrm{Delay}(x_i)$是工作负载加权的综合后 MAC ADP，其中 w_i是运算符 i在 MAC 总计数中的占比。精度约束要求量化精度 $\mathcal{A}(x)$在相同评估划分上至少保留全精度精度的 τ $\mathcal{A}_{\mathrm{FP}}$ 。集合 Ω_{num} 和 Ω_{hw} 分别确保数值有效性，以及在目标时钟周期下实现时序收敛。

III. 方法

A. 框架概述

NAMSI 在闭环中协同优化算子格式和 MAC 电路（图2和算法1）。它在精度、数值有效性和时序约束下，最小化工作负载加权的综合后 MAC 面积延迟积。离线初始化会分析

算法1 NAMSI 闭环格式与乘加运算综合

要求： 格式域 $\mathcal{F}$、种子观测 $\mathcal{D}_{\text{init}}$、训练后量化预算 B_{tot}、批大小 B。
确保： 最佳经验证的可行分配 x^{*}。

分析张量并提取图 G。
初始化硬件缓存 $\mathcal{H}$ 和格状态 Λ。
$\mathcal{D} \leftarrow \mathcal{D}_{\text{init}}$，$\mathcal{U} \leftarrow \{x : (x, \Delta\mathcal{A}) \in \mathcal{D}\}$，$n \leftarrow 0$
当 $n < B_{\text{tot}}$ **时**
  $x^{*} \leftarrow$ 最佳可行解$(\mathcal{D}, \mathcal{H})$
  在 $\mathcal{D}$ 上拟合精度高斯过程，在 $\mathcal{H}$ 上拟合成本GP。
  $\mathcal{P} \leftarrow$ 提出$(\mathcal{D}, \Lambda, G, \mathcal{H}) \setminus \mathcal{U}$
  如果 $\mathcal{P} = \varnothing$，**则**
    跳出
  结束if
  通过 (7) 选择最多 $\min\{B, B_{\text{tot}} - n, |\mathcal{P}|\}$ 个候选项用于 $\mathcal{S}$，并 Kriging Believer。
  对于 $x \in \mathcal{S}$，**执行**
    $\mathcal{U} \leftarrow \mathcal{U} \cup \{x\}$
    验证 x 中未见过的配对，并更新 $\mathcal{H}$。
    如果所有配对均通过硬件验证**则**
      $(\mathcal{A}(x)\ v) \leftarrow$ 训练后量化(x)
      $n \leftarrow n + 1$
      如果 $v =$有效 **则**
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x, \mathcal{A}(x) - \mathcal{A}_{\text{FP}})\}$
      结束如果
    结束如果
    根据验证和训练后量化结果更新 Λ。
  结束 for 循环
结束 while 循环
返回 最佳可行解$(\mathcal{D}, \mathcal{H})$

校准数据上的逐张量统计量，并提取 MAC 图 G。FD 将每种操作数格式及其张量分布进行编码。随后，TaH 预测全网络精度和硬件成本。接着，RLS 提出候选分配并选择一个评估批次 $\mathcal{S}$。在进行全网络 PTQ 之前，CosMac 会对 $\mathcal{S}$ 中未见过的格式对进行综合和验证。新的测量结果会更新代理模型、硬件缓存 $\mathcal{H}$ 和搜索状态 Λ。当 PTQ 预算 B_{tot} 耗尽后，NAMSI 返回成本最低的经验证的可行分配。

B. 通过 *CosMac* 进行参数化 *MAC* 综合

对于每个格式对 $x_i = (F_{W,\,i},\ F_{A,\,i})$，CosMac 都会为采用 FP32 累加的融合 MAC 生成专用的 SystemVerilog RTL。格式参数用于配置操作数解码、乘法、累加器对齐、归一化和舍入。对于每个格式对，CosMac 会评估多种算术结构和流水线配置，并保留成本最低的有效设计。Synopsys Design Compiler 对 RTL 进行综合，并根据 T_{clk} 检查时序。候选设计只有在通过门级功能验证以及建立/保持时间检查后，才会进入 Ω_{hw} 。

硬件缓存 $\mathcal{H}$ 会存储每个已评估格式对的实测成本 $c_{\text{MAC}}(x_i)$、=面积(x_i)、x_i延迟以及验证状态。缓存结果可在不同算子和搜索轮次之间复用，因此综合仅针对 $\mathcal{S}$中尚未见过的格式对运行。

C. 格式–分布表示

FD 用六个数值属性表示每种格式 $F \in \mathcal{F}$：

$$\phi(F) = \left[k(F), m_{\text{eff}}(F), e(F), b(F), r(F), q(F)\right]. \quad (2)$$

其中，$k \in \{0, 1\}$ 表示整数或浮点数。对于有符号整数，术语 m_{eff} 等于 $Z - 1$，而对于 eXfY，则等于 Y。属性 e 和 b 分别表示指数位宽和总位宽，因此对于 int$Z$，$e = 0$，$b = Z$；对于 eXfY，则为 $e = X$，$b = 1 + X + Y$。在采用标准指数偏置并支持次正规数的情况下，r =log₂(最大有限值/最小正值)用于衡量对数动态范围，而 $q = -\overline{m}_{\text{eff}}$ 用于衡量对数分辨率。

每个张量 T 都通过其对数峰值因子进行概括：

$$\kappa(T) = \log_2\left(\frac{\|T\|_{\infty} + \varepsilon}{\sqrt{\mathbb{E}[T^2]} + \varepsilon}\right), \quad (3)$$

其中，$\varepsilon > 0$ 可确保数值稳定性。较大的 $\kappa(T)$ 表示相对于均方根幅值而言更强的离群值。在峰值缩放量化下，离群值会根据 $m_{\text{eff}}(F) - \kappa(T)$ 稀释整个张量主体中的均匀整数分辨率。窄格式还会产生范围跨度缺失 $\max(0,\ \kappa(T) - r(F))$，而 $\kappa(T)k(F)$ 则捕获浮点尾部适应。对 $\phi(F)$ 及这些交互项进行有界非线性缩放，可得到操作数嵌入 $\tilde{\phi}(F, \kappa) \in \mathbb{R}^8$。将权重嵌入与激活嵌入拼接后得到

$$\begin{aligned} z_i(x) &= \tilde{\phi}(F_{W,i}, \kappa(W_i)) \,\|\, \tilde{\phi}(F_{A,i}, \kappa(A_i)), \\ Z(x) &= z_1(x) \,\|\, \cdots \,\|\, z_L(x), \end{aligned} \quad (4)$$

其中，$\|$ 表示拼接，$z_i(x) \in \mathbb{R}^{16}$。这种表示方式在保留离群值统计的同时，将动态范围与主体精度分离开来。

D. 拓扑感知的精度与硬件成本建模

TaH 将格式对硬件成本模型与基于图的精度模型相结合。利用 $\mathcal{H}$，两个高斯过程根据 $\phi(F_{W,\,i}) \,\|\, \phi(F_{A,\,i})$ 预测面积和延迟。对于未见过的格式对，预测成本为

$$\begin{aligned} \ln \hat{c}_{\text{MAC}}(x_i) &= \hat{\mu}_{\ln \text{Area}}(x_i) + \hat{\mu}_{\ln \text{Delay}}(x_i), \\ \hat{C}_H(x) &= \sum_{i=1}^{L} w_i\, \hat{c}_{\text{MAC}}(x_i). \end{aligned} \quad (5)$$

$\mathcal{H}$ 中的格式对直接使用 $c_{\text{MAC}}(x_i)$，而失败的格式对会在评分前被拒绝。

在精度建模中，执行追踪会在 MAC 运算符上构建稀疏无向图 $G = (\mathcal{V},\ \mathcal{E})$。边连接数据流路径上最近的 MAC 祖先，以及在残差加法处汇合的分支。每次训练后量化评估都会先在校准集划分上校准量化范围，再测量精度 $\mathcal{A}(x)$。未发生溢出、非有限值或零范围坍缩的运行会标记为有效 $(x \in \Omega_{\text{num}})$。精度高斯过程在 $\mathcal{D}$ 中使用按样本量缩放的噪声方差，对有效记录进行训练，并针对各划分的 FP32 基线对相对变化 $\Delta\mathcal{A}(x) = \mathcal{A}(x) - \mathcal{A}_{\text{FP}}$ 进行建模。其协方差由节点项和边项共同构成：

$$\begin{aligned} K_{\text{topo}}(x, x') = {} & \frac{\sigma_V^2\, g_V(Z, Z')}{L} \sum_{i=1}^{L} k_V(z_i, z_i') \\ & + \frac{\sigma_E^2\, g_E(Z, Z')}{|\mathcal{E}|} \sum_{(i,j) \in \mathcal{E}} k_E(z_i \,\|\, z_j, z_i' \,\|\, z_j'), \end{aligned} \quad (6)$$

其中 $Z = Z(x)$、 $Z' = Z(x')$、 $z_i = z_i(x)$ 和 $z'_i = z_i(x')$。参数 σ_V^2 和 σ_E^2 控制节点振幅和边振幅；当 $\mathcal{E} = \varnothing$ 时，不计入边项求和。这种结构能够捕捉局部敏感性和相邻层耦合，而无需使用稠密的全对项。TaH 从高斯过程后验中报告预测精度 $\mu_A(x) = \mathcal{A}_{\mathrm{FP}} + \mathbb{E}[\Delta\mathcal{A}(x) \mid \mathcal{D}]$ 和标准差$\sigma_A(x)$。

E. 可逆格搜索与批量选择

RLS 使用族内容量顺序进行探索。状态映射 Λ 跟踪每个操作数的格式状态（活跃、休眠或禁止），同时记录全局禁止的时序违规对。单算子数值验证失败时，会禁止有问题的格式及其较低格锥。局部精度大幅下降或成本占优的格式会进入休眠状态。当 G 中的相邻算子改变格式时，RLS 会以概率方式探测休眠边界格式，并重新激活通过代理筛选和验证的格式。

RLS 保留一个由成本最低的可行分配组成的小型精英池，并在每轮生成一个去重候选池 $\mathcal{P}$，其范围位于 $\mathcal{U}$ 之外。算子按照与 $\sqrt{w_i}$成正比的概率进行采样。提案结合四种操作：族内格点步进、由 κ引导的相近位宽整数–浮点切换、在 $\mathcal{E}$中的边之间进行耦合精度重新分配，以及对高工作负载算子进行联合压缩。

令 C_H^* 为迄今测得的最低可行成本。预测相对成本降幅为 $I_{\mathrm{cost}}(x) = \max(0,\ 1 - \hat{C}_H(x)/C_H^*)$。可行性概率为 $P_{\mathrm{feas}}(x) = \Phi((\mu_A(x) - \tau\mathcal{A}_{\mathrm{FP}})/\max(\sigma_A(x),\ \varepsilon))$，其中 Φ 为标准正态累积分布函数。每个候选项的评分为

$$\alpha(x) = I_{\mathrm{cost}}(x) P_{\mathrm{feas}}(x) + \lambda\, \mathbb{I}[\hat{C}_H(x) < C_H^*] \times \frac{\sigma_A(x)}{\mathcal{A}_{\mathrm{FP}}} \exp\left[-\frac{(\mu_A(x) - \tau\mathcal{A}_{\mathrm{FP}})^2}{2\sigma_A^2(x) + \varepsilon}\right], \tag{7}$$

其中， $\mathbb{I}[\cdot]$ 是指示函数， $\lambda > 0$ 用于加大对精度边界附近区域的探索权重。Kriging Believer [26]采用贪心策略，为批次 $\mathcal{S}$ 最多选择 $\min\{B, B_{\mathrm{tot}} - n, |\mathcal{P}|\}$ 个候选项。每选出一个候选项后，都会根据该候选项的预测均值 $\mu_A(x)$ 对临时精度高斯过程进行条件化，以抑制冗余选择。这里， B 是最大批大小， n 统计在线训练后量化运行次数。每个 $x \in \mathcal{S}$ 在训练后量化前都要经过 CosMac验证，且只有训练后量化运行会消耗B_{tot}。有效的训练后量化运行会更新 $\mathcal{D}$ ，即使其低于 $\tau\mathcal{A}_{\mathrm{FP}}$；实测成本则会更新 $\mathcal{H}$。NAMSI 返回实测成本最低的可行分配。

IV. 实验评估

A. 实验设置

所有实验均在 Linux 服务器上使用 PyTorch 2.5.1 和 CUDA 12.1 运行。对于 CosMac生成的MAC RTL，逻辑综合和物理实现分别使用 Synopsys Design Compiler 和 Synopsys IC Compiler II (ICC2) 完成，基于 7纳米ASAP7 工艺设计套件 [27]，时间为 $T_{\mathrm{clk}} = 2,\ 000$ ps。

我们在 ImageNet-1K [28] 上评估了五个网络：ResNet-18、ResNet-34 [29], VGG16-BN [30],RegNetX-400MF [31], 以及 MobileNetV2 [32]。所有权重均保持冻结，不进行微调，并以 FP32 Top-1 准确率（$\mathcal{A}_{\mathrm{FP}}$）作为参考，采用统一的 $\tau = 0.99$ 保留阈值。我们抽取两个互不重叠的训练子集，分别用于校准（C，4，000 张图像）和在线搜索评估（S，1，000–5，000 张图像），并确保官方 50，000 张图像验证集（V）在最终的一次性评估之前严格不被使用。

我们将 NAMSI 与高通 AIMET-AMP（v1.35.0） [33], 进行比较；后者在相同的冻结训练后量化协议下，搜索 $\{\mathrm{int2}, \ldots, \mathrm{int8}\}$ 范围内相互独立的权重和激活格式，并通过相同的 CosMac 综合流程测量硬件 ADP。我们还设置了两项消融研究，分别验证代理引导的可逆格点搜索相较于逐层顺序贪心选择（Greedy-Sensitivity）的有效性，以及通过禁用图边来检验跨层拓扑耦合的必要性（NAMSI 去除边）。在所有实验中，搜索均以 256 张图像的离线单算子锚点初始化，并在 $B_{\mathrm{tot}} = 100$ 的在线评估预算内进行优化。

B. 端到端准确率与硬件比较

图 3(a) 汇总了各网络在锁定的 V-拆分上的精度以及综合后硬件成本，其中硬件指标以综合后的 FP32 乘加运算成本为基准进行归一化，以便跨网络对齐。在所评估的网络中，与 FP32 相比，NAMSI 将乘加运算 ADP 降低了 **54.2%–60.4%**，且其 ADP 低于或接近 AIMET-AMP。AIMET-AMP 会造成显著的精度下降，而 NAMSI 在所有测试网络中始终保留至少**99%** 的全精度 FP32 基线精度。对于紧凑型分组卷积网络 RegNetX-400MF，由于通道间冗余有限，AIMET-AMP 的相对精度下降尤为明显。相比之下，NAMSI 会选择性地为对瓶颈敏感的层分配具有较宽动态范围的异构格式；与 FP32 相比，这使 ADP 降低了 54.2%，同时将相对精度损失控制在 0.75% 以内，并保持严格可行。

图 3(b) 展示了布线后的乘加实现，从物理层面证明异构数据通路能够有效降低名义位宽代理指标未能捕捉到的乘法器相关硅面积开销。总体而言，这些端到端和布线后结果证实了 NAMSI 的有效性，表明其相较于传统混合精度分配具有明显的精度–硬件优势。

为解释这些端到端增益背后的机制，下面我们将通过逐层异构格式分配以及张量级格式–分布匹配，进一步分析 NAMSI 的决策。

1）逐层异构格式分配：在不同网络中识别出的逐层异构格式分配方案，揭示了 NAMSI 如何将算子级决策转化为端到端的精度–ADP 增益。如图 4 所示，NAMSI 并未采用统一位宽，也没有将所有算子限制为单一格式族，而是为具有显著激活

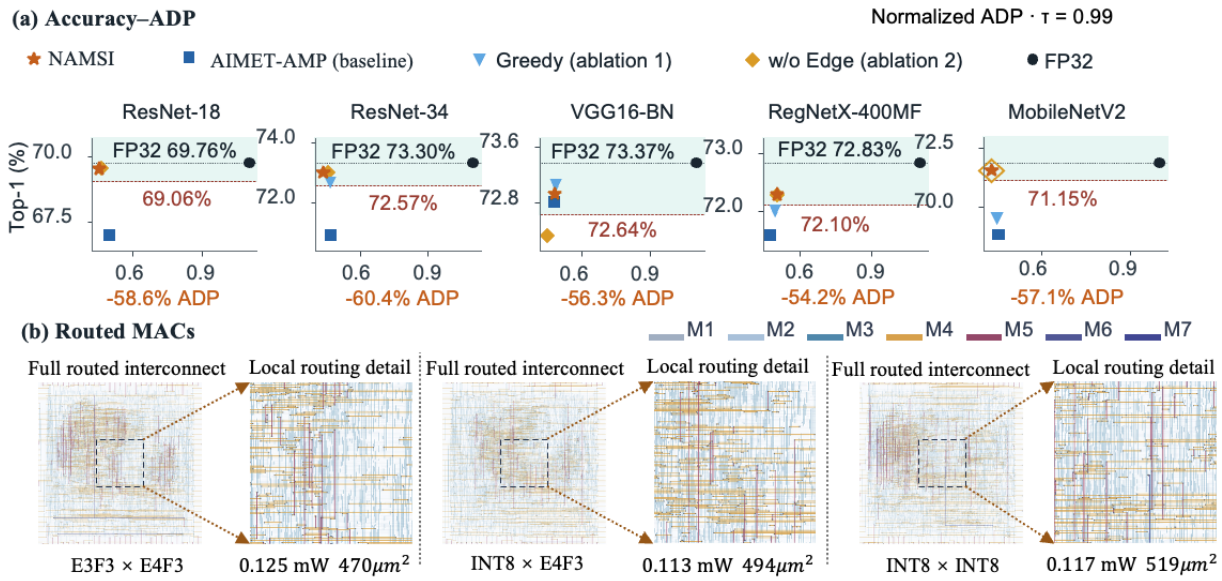


图3. 不同网络和布线后的乘加布局之间的端到端比较。(a) 锁定验证（V）中，Top-1 准确率与 FP32归一化的工作负载加权乘加面积-延迟-功耗积的关系。(b) ICC2 完成布局布线的乘加布局，展示完整的布线后互连和局部布线细节。

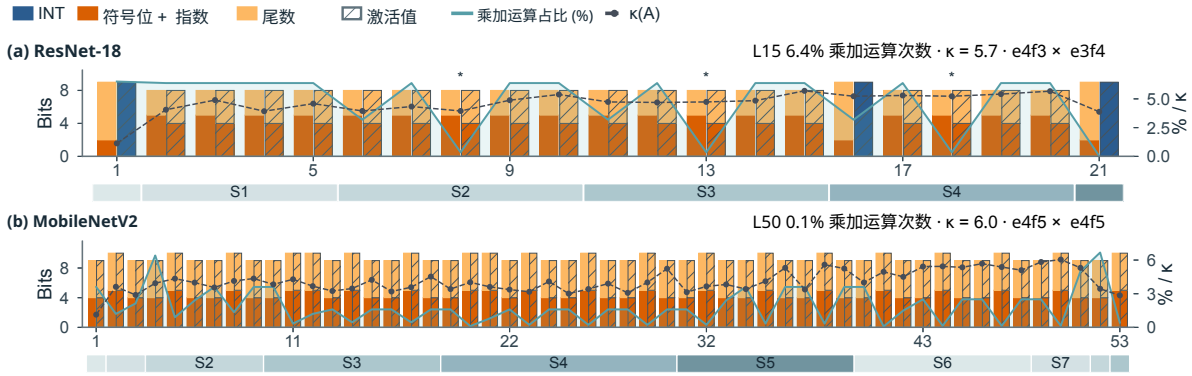


图4. NAMSI 发现的逐层权重（实心）和激活值（阴影）格式分配，并叠加逐层乘加占比（w_i）和激活值对数峰值因子（κ（A_i））。浮点条形图将总位宽分解为 $1 + X + Y$，其中 X 和 Y 分别为指数和尾数位宽。

异常值的计算密集型层分配紧凑型定制浮点数据通路，并为对精度至关重要的瓶颈保留更高精度格式。

对于 ResNet-18（图 4(a)），NAMSI 为三个敏感瓶颈分配 9 位混合格式（e1f7 ×int9）：输入激活值较为紧凑的干卷积（$\kappa(A_1) = 1.1$）、Stage-4 入口层（3.2% 乘加运算次数）以及最终分类器（0.03% 乘加运算次数），从而在受控硬件成本下保持特征和输出精度。相反，对于其余 18 个主干卷积和投影卷积，这些层占据主要计算量（占总乘加运算次数的 90.3%，单层最高达 6.4%），并表现出明显的激活异常值（$\kappa(A_i) \in [3.9, 5.7]$，在 L15 达到峰值 $\kappa = 5.7$），NAMSI 将操作数压缩为紧凑的 8 位自定义浮点数对（e4f3 ×e3f4），以吸收重尾特征，同时最大化全网络 ADP 节省。这些结果表明，有效的异构格式分配必须综合考虑每个算子的工作负载贡献和张量异常值统计。NAMSI 利用这两类信号进行自适应调整

每层的格式族和精度，在局部精度敏感性与全网络硬件效率之间取得平衡。

2) 格式–分布机制：图 5 对比了含大量离群值的 ResNet-18 激活值（$\kappa = 4.68$）与分布紧凑的 RegNetX-400MF 权重张量（$\kappa = 2.54$），揭示了这些格式选择在不同张量分布情形下的数值机制。图中展示了整数和浮点量化网格如何与各自的幅值分布（上图）对齐，以及在 4–8 位格式中，相对 L_2 量化误差如何变化（下图）。对于 ResNet-18（图 5(a1)），大多数数值都紧密集中在零附近。全范围 int8 将统一量化级别低效地分配给样本稀疏的尾部，而百分位裁剪 int8 则会因截断裁剪区域中的极端值而产生严重误差。相比之下，选定的 e3f4 格式使用三位指数覆盖长尾，并用四位尾数提高零附近的分辨率，从而实现最低的相对 L_2

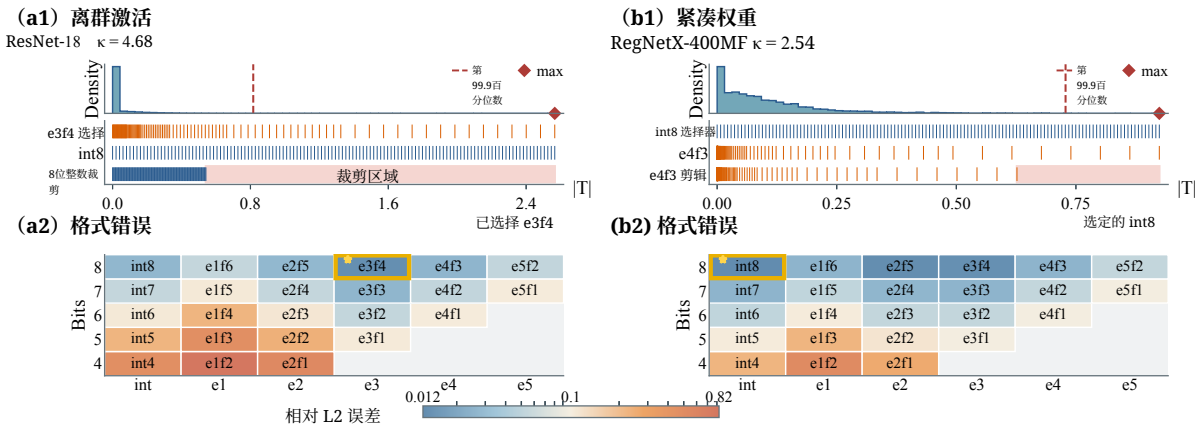


图5. 两个代表性张量上的微观格式–分布匹配。(a1)–(a2) ResNet-18 中的异常值密集型激活（$\kappa = 4.68$），比较跨越4–8位数的可表示量化网格和二维相对 L_2 误差。(b1)–(b2) RegNetX-400MF 中的紧凑型权重张量（$\kappa = 2.54$）。

误差，在所评估的 8位格式中表现最佳（图 5(a2)）。

对于紧凑的 RegNetX-400MF 权重张量（图5(b1)），其幅值平滑衰减，不存在明显的离群值。指数域更宽的格式会牺牲有效精度，并在密集分布的中间范围内留下更粗的量化间隙；相比之下，均匀int8能够密集覆盖张量支集，并实现最低误差（图5(b2)）。总体而言，这些结果从微观数值层面解释了 NAMSI 的算子级格式选择。通过将格式族及指数–尾数分配与每个张量的异常值特征相匹配，NAMSI能够避免重尾分布中的截断误差，同时在紧凑分布中保持分辨率，而无需分配不必要的指数位。

C. 消融研究

为分别评估 RLS 和 TaH 的贡献，我们在相同的评估预算和固定的 V-split 协议下，将 NAMSI 与 Greedy-Sensitivity 以及 NAMSI w/oEdge 进行比较。

1) 消融实验 *1::* Greedy-Sensitivity 根据独立的单算子敏感度按顺序压缩各层。如图3(a)所示，这种短视策略在紧凑型网络上违反了 $\tau = 0.99$精度约束：MobileNetV2 的精度比 NAMSI 低 2.00 个百分点；同时，在 ResNet-34 等更深的骨干网络上，其 MAC ADP 也高出 7.1%。这些结果表明，在严格的精度约束下，全局代理模型引导的 RLS 对于协调各层的格式选择至关重要。

2）消融实验 *2*：在 NAMSI 去除边中禁用图边（$\sigma_E = 0$ in (6)）会忽略跨层误差耦合。由于未对序贯误差累积进行建模，NAMSI 去除边会对 VGG16-BN 进行过度压缩，导致其违反 $\tau = 0.99$ 精度保持阈值，如图 3(a) 所示。由于无法捕捉跨分支交互，其在 ResNet-18 和 ResNet-34 上的 MAC ADP 也分别高出 2.0% 和 4.9%。这些结果表明，TaH 对于捕捉层间误差传播、降低硬件成本并同时保持可行性至关重要。

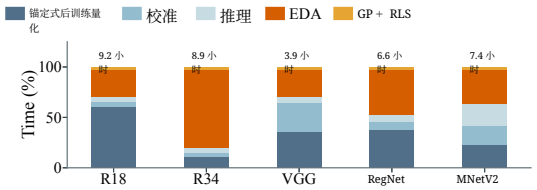


图6. 各评估网络的端到端搜索时间及归一化五阶段运行时间构成。

D. 搜索效率与运行时间分解

在每轮搜索中，RLS 会生成 ∼600 候选配置，其中包含许多此前未经综合的格式对。NAMSI 不会在提案评分期间调用门级综合，而是通过成本GP评估未见过的格式对，并使用 $O(L)$加权求和（5），在毫秒内为每个 L层候选方案评分。只有当所选采集批次中出现新的格式对时，才会触发精确综合；测量结果会缓存在 $\mathcal{H}$ 中，以便立即复用到各层和后续轮次中。这种双速评估显著降低了综合在环优化的计算开销。

图 6 进一步将端到端运行时间细分为五个阶段：离线单算子锚点 PTQ（锚定式后训练量化）、在线 PTQ 校准（校准）与搜索切分推理（推理）、按需调用 Design Compiler 进行综合（EDA），以及结合 RLS 采集的拓扑感知 GP 拟合（GP + RLS）。在所有评估的网络中，GP + RLS 占总运行时间的 $< 3\%$。其余运行时间主要用于必要的 PTQ 评估以及对新遇到的异构格式对进行精确综合。值得注意的是，ResNet-34 的 EDA 运行时间占比明显更高，因为其搜索遇到了更多此前未见的格式对，导致综合缓存命中率较低，且需要更频繁地按需调用 Design Compiler。

V. 结论

我们提出了 NAMSI，这是一种网络感知自动化 MAC 综合框架，可在 PTQ 下联合优化算子级异构数据格式与由 CosMac 综合的算术数据通路。NAMSI 集成了格式–分布表示、用于精度与硬件成本建模的双高斯过程拓扑感知模型，以及可逆格搜索；它能够遍历无序格式空间，同时仅对未见过的格式对进行门级综合。在 7 纳米 ASAP7 工艺下，针对 ImageNet-1K 上的五种网络，NAMSI 在 3.9–9.2 小时内将综合后 MAC 面积延迟积较 FP32 降低 54.2%–60.4%，同时保留 $\geq$ 99% 的 Top-1 精度，证明了联合优化网络及其硬件实现的重要性。

参考文献

[1] B. Jacob、S. Kligys、B. Chen、M. Zhu、M. Tang、A. Howard、H. Adam 和 D. Kalenichenko，“面向高效纯整数算术推理的神经网络量化与训练”，载于*IEEE* 计算机视觉与模式识别会议（*CVPR*）论文集，2018年，第2704–2713页。[2] S. K. Esser、J. L. McKinstry、D. Bablani、R. Appuswamy 和 D. S. Modha，“学习步长量化”，载于国际学习表征会议（*ICLR*），2020年。[3] M. Nagel、R. A. Amjad、M. Van Baalen、C. Louizos 和 T. Blankevoort，“向上还是向下？训练后量化的自适应舍入”，载于第*37*届国际机器学习会议（*ICML*）论文集，第119卷，2020年，第7197–7206页。[4] Y. Li、R. Gong、X. Tan、Y. Yang、P. Hu、Q. Zhang、F. Yu、W. Wang 和 S. Gu，“BRECQ：通过块重建突破训练后量化的极限”，载于国际学习表征会议（*ICLR*），2021年。[5] E. Frantar、S. Ashkboos、T. Hoefler 和 D. Alistarh，“OPTQ：生成式预训练变换器的精准训练后量化”，载于国际学习表征会议（*ICLR*），2023年。[6] G. Xiao、J. Lin、M. Seznec、H. Wu、J. Demouth 和 S. Han，“SmoothQuant：大型语言模型的精准高效训练后量化”，载于第*40*届国际机器学习会议（*ICML*）论文集，丛书：机器学习研究论文集，第202卷，2023年，第38 087–38 099页。[7] J. Lin、J. Tang、H. Tang、S. Yang、W.-M. Chen、W.-C. Wang、G. Xiao、X. Dang、C. Gan 和 S. Han，“AWQ：激活值感知权重量化，用于设备端大语言模型压缩与加速”，载于机器学习与系统论文集（*MLSys*），第6卷，2024年。[8] K. Wang、Z. Liu、Y. Lin、J. Lin 和 S. Han，“HAQ：采用混合精度的硬件感知自动化量化”，载于*IEEE/CVF* 计算机视觉与模式识别会议（*CVPR*）论文集，2019年，第8604–8612页。[9] Z. Yao、Z. Dong、Z. Zheng、A. Gholami、J. Yu、E. Tan、L. Wang、Q. Huang、Y. Wang、M. W. Mahoney 和 K. Keutzer，“HAWQ-V3：二进制幂神经网络量化”，载于第*38*届国际机器学习会议（*ICML*）论文集，第139卷，2021年，第11 875–11 886页。[10] Z. Cai 和 N. Vasconcelos，“重新思考混合精度神经网络的可微搜索”，载于*IEEE/CVF* 计算机视觉与模式识别会议（*CVPR*）论文集，2020年，第2349–2358页。[11] J. Dotzel、G. Wu、A. Li、M. Umar、Y. Ni、M. S. Abdelfattah、Z. Zhang、L. Cheng、M. G. Dixon、N. P. Jouppi、Q. V. Le 和 S. Li，“FLIQS：一次性混合精度浮点与整数化搜索”，载于第三届国际自动化机器学习会议论文集，丛书：机器学习研究论文集，第256卷，2024年，第6/1–26页。[12] Y. Zhang、L. Zhao、S. Cao、S. Zhang、W. Wang、T. Cao、F. Yang、M. Yang、S. Zhang 和 N. Xu，“整数还是浮点数？大型语言模型低比特量化的新视角”，载于*IEEE* 多媒体与博览会国际会议（*ICME*）论文集，2024年，第1–6页。[13] A. Kuzmin、M. Van Baalen、Y. Ren、M. Nagel、J. Peters 和 T. Blankevoort，“FP8 量化：指数的力量”，发表于神经信息处理系统进展（*NeurIPS*），第 35 卷，2022年，第 14 651–14 662 页。[14] S.-y. Liu、Z. Liu、X. Huang、P. Dong 和 K.-T. Cheng，“LLM-FP4：4 位浮点量化变换器”，发表于*2023*年自然语言处理实证方法会议论文集（*EMNLP*），2023年，第 592–605 页。[15] Y. Li、X. Dong 和 W. Wang，“加性二次幂量化：一种用于神经网络的高效非均匀离散化方法”，发表于国际学习表征会议（*ICLR*），2020年。[16] J. Dotzel、Y. Chen、B. Kotb、S. Prasad、G. Wu、S. Li、M. S. Abdelfattah 和 Z. Zhang，“向学生学习：运用 t分布探索适用于大语言模型的准确高效格式”，发表于第 *41* 届国际机器学习会议论文集（*ICML*），《机器学习研究论文集》，第 235 卷，2024年，第 11 573–11 591 页。[17] T. Tambe、E.-Y. Yang、Z. Wan、Y. Deng、V. J. Reddi、A. Rush、D. Brooks 和 G.-Y. Wei，“面向弹性深度学习推理的自适应浮点编码算法—硬件协同设计”，发表于第 *57* 届 *ACM/IEEE*设计自动化会议论文集（*DAC*），2020年，第 1–6 页。[18] B. Darvish Rouhani、D. Lo、R. Zhao、M. Liu、J. Fowers、K. Ovtcharov、A. Vinogradsky、S. Massengill、L. Yang、R. Bittner、A. Forin、H. Zhu、T. Na、P. Patel、S. Che、L. C. Koppaka、X. Song、S. Som、K. Das、S. Tiwary、S. Reinhardt、S. Lanka、E. Chung 和 D. Burger，“借助微软浮点在云规模下突破窄精度推理的极限”，发表于神经信息处理系统进展（*NeurIPS*），第 33 卷，2020年，第 10 271–10 281 页。[19] B. Darvish Rouhani、R. Zhao、V. Elango、R. Shafipour、M. Hall、M. Mesmakhosroshahi、A. More、L. Melnick、M. Golub、G. Varatkar、L. Shao、G. Kolhe、D. Melts、J. Klar、R. L’Heureux、M. Perry、D. Burger、E. Chung、Z. S. Deng、S. Naghshineh、J. Park 和 M. Naumov，“借助共享微指数，微小的移位也能产生显著效果”，发表于第 *50* 届计算机体系结构年度国际研讨会论文集（*ISCA*），2023年。[20] C. Guo、C. Zhang、J. Leng、Z. Liu、F. Yang、Y. Liu、M. Guo 和 Y. Zhu，“ANT：利用自适应数值数据类型进行低位宽深度神经网络量化”，发表于第 *55* 届 *IEEE/ACM*微体系结构国际研讨会论文集（*MICRO*），2022年，第 1414–1433 页。[21] W. Jang 和 T. Tambe，“BlockDialect：面向节能大语言模型推理的分块细粒度混合格式量化”，发表于第 *42* 届国际机器学习会议论文集（*ICML*），《机器学习研究论文集》，第 267 卷，2025年，第 26 925–26 945 页。[22] P. Judd、J. Albericio、T. Hetherington、T. M. Aamodt 和 A. Moshovos，“Stripes：位串行深度神经网络计算”，发表于第 *49* 届 *IEEE/ACM*微体系结构年度国际研讨会论文集（*MICRO*），2016年，第 1–12 页。[23] H. Sharma、J. Park、N. Suda、L. Lai、B. Chau、J. K. Kim、V. Chandra 和 H. Esmaeilzadeh，“位融合：用于加速深度神经网络的位级动态可组合架构”，发表于*ACM/IEEE*第 *45* 届计算机体系结构年度国际研讨会论文集（*ISCA*），2018年，第 764–775 页。[24] H. Xia、Z. Zheng、X. Wu、S. Chen、Z. Yao、S. Youn、A. Bakhtiari、M. Wyatt、D. Zhuang、Z. Zhou、O. Ruwase、Y. He 和 S. L. Song，“Quant-LLM：在现代GPU上通过以 FP6 为核心的算法—系统协同设计加速大型语言模型服务”，发表于*2024*年 *USENIX*年度技术会议论文集（*USENIX ATC*），2024年，第 699–713 页。[25] B. Crafton、X. Peng、X. Sun、A. S. Lele、B. Zhang、W.-S. Khwa 和 K. Akarvardar，“为大语言模型推理寻找低精度数据格式与乘加运算架构的帕累托前沿”，发表于第 *62* 届 *ACM/IEEE*设计自动化会议论文集（*DAC*），2025年，第 1–7 页。[26] D. Ginsbourger、R. Le Riche 和 L. Carraro，“克里金法非常适合并行化优化”，载于昂贵优化问题中的计算智能，《自适应、学习与优化》系列，Y. Tenne 和 C.-K. Goh 编，施普林格，2010年，第131–162 页。[27] L. T. Clark、V. Vashishtha、L. Shifren、A. Gujja、S. Sinha、B. Cline、C. Ramamurthy 和 G. Yeric，“ASAP7：一种 7纳米 FinFET 预测性工艺设计套件”，载于《微电子学杂志》，第 53 卷，第 105–115 页，2016年。[28] J. Deng、W. Dong、R. Socher、L.-J. Li、K. Li 和 L. Fei-Fei，“ImageNet：大型分层图像数据库”，发表于*IEEE*论文集

计算机视觉与模式识别会议（*CVPR*），2009年，第 248–255 页。[29] K. He、X. Zhang、S. Ren 和 J. Sun，“用于图像识别的深度残差学习”，发表于*IEEE*计算机视觉与模式识别会议论文集（*CVPR*），2016年，第 770–778 页。[30] K. Simonyan 和 A. Zisserman，“用于大规模图像识别的超深卷积网络”，发表于国际学习表征会议（*ICLR*），2015年。[31] I. Radosavovic、R. P. Kosaraju、R. Girshick、K. He 和 P. Doll´ar，“网络设计空间的设计”，发表于*IEEE/CVF*计算机视觉与模式识别会议论文集（*CVPR*），2020年，第 10 428–10 436 页。[32] M. Sandler、A. Howard、M. Zhu、A. Zhmoginov 和 L.-C. Chen，“MobileNetV2：倒置残差与线性瓶颈”，发表于*IEEE/CVF*计算机视觉与模式识别会议论文集（*CVPR*），2018年，第 4510–4520 页。[33] S. Siddegowda、M. Fournarakis、M. Nagel、T. Blankevoort、C. Patel 和 A. Khobare，“使用人工智能模型效率工具包（AIMET）进行神经网络量化”，载于*arXiv*预印本 *arXiv:2201.08442*，2022年。