arXiv:2603.03383v1 [cs.DC] 3 Mar 2026

通过推测解码加速 OpenPangu 在 NPU 上的推理

Yuntao Dai、Jing Wu、Hang Gu、
Teng Wang中国科学技术大学软件工程学院

摘要

为缓解大语言模型（LLM）在**NPU** 硬件上推理时遇到的内存墙瓶颈，并解决国产基础设施对主流推测解码算法原生支持不足的问题，本研究提出了一种面向 OpenPangu-7B 的端到端推测推理加速方案。我们通过设计轻量级多头预测架构和静态树形注意力验证机制，对 Medusa 算法进行了适配与增强。实证评估表明，在短序列生成任务中，我们的方法在**NPU** 上实现了**1.35x**的端到端加速比。此外，我们还深入分析了处理长序列时影响加速收益的内存带宽限制，为未来面向硬件的优化提供了重要参考。代码见：https://github.com/wujing215/OpenPangu7B-with-Medusa。

关键词：大语言模型、推测解码、**NPU**

1 引言

基于 Transformer 的大语言模型（LLM）的出现 Vaswani 等人 [2017],Radford等人 [2019], Brown等人 [2020]彻底改变了人工智能的发展格局，使自然语言理解与生成能力取得了前所未有的进步。然而，这些通常拥有数十亿参数的模型 Touvron等人 [2023],Chowdhery等人 [2023], 在部署时面临一个根本性的硬件瓶颈：计算能力与内存带宽之间的不匹配 Mutlu等人 [2015]。标准的自回归解码会按顺序生成词元；每生成一个词元，都必须将整个模型权重从高带宽内存（HBM）传输到芯片上的计算单元。这种受内存带宽限制的特性导致算术强度极低 Williams 等人 [2009], ，使强大的矩阵计算单元——例如**NPU**中的张量核心——在推理周期的大部分时间里都处于闲置状态。为缓解这一“内存墙”问题，推测推理应运而生，成为一种颇具前景的新范式。通过利用较轻量的代理计算草拟多个候选词元，再由目标模型并行验证，推测解码有效地将受内存带宽限制的顺序操作转变为受计算能力限制的并行操作 Leviathan 等人 [2023], Chen 等人 [2023], Kim等人 [2023]。在这些方法中，Medusa Cai 等人 [2024] 代表了最先进水平：它在冻结的主干网络上附加轻量级多层感知机（MLP）注意力头，从而无需维护单独的草稿模型，即可实现分块生成。尽管 Medusa 在英伟达图形处理器平台上效果显著，但将这一范式移植到新兴的国产**NPU** 硬件生态系统中仍面临不小的挑战。首先，软硬件之间存在明显错配。当前实现高度依赖动态控制流和针对 CUDA 的优化 Pope 等人 [2023]，这些都与**NPU** 软件栈不兼容。其次，也是更关键的一点，推测解码的算法特性与 NPU 的执行模型之间存在冲突。Medusa算法本质上需要根据实时预测概率进行动态树构建

和验证 Miao 等人 [2024]。相比之下，**NPU**架构针对静态图执行进行了优化 Liao 等人 [2021], , 在运行时之前就固定内存地址和计算图，以最大限度地提高吞吐量。直接实现动态树注意力逻辑通常会频繁触发图重新编译，并产生严重的同步开销 Aminabadi 等人 [2022], , 往往会抵消任何潜在的加速效果。为应对这些挑战，本文提出了一种面向 OpenPangu-7B Chen 等人 [2025]的端到端加速方案，弥合先进推测算法与国产硬件特性之间的差距。我们的主要贡献总结如下：

- 我们在**NPU** PyTorch 生态系统中首次全面实现了 Medusa 推测解码框架，并系统地解决了软件栈中的算子级不兼容问题。
- 我们提出了一种硬件友好的静态树构建机制，并搭配零拷贝检索策略，从而解决动态推测验证与 NPU 静态执行模型之间的冲突。
- 我们在**NPU**平台上进行了广泛的实证基准测试，展示了 1.35×的端到端加速比，并对内存带宽限制进行了关键分析 Dao 等人。 [2022], Dao [2023]。

2 背景

2.1 自回归生成中的内存墙

大语言模型推理的性能取决于工作负载相对于硬件运算强度能力的运算强度 Williams 等人 [2009]。在标准自回归解码阶段，生成单个 token x_t 需要基于上下文 $x_{<t}$ 对模型参数 θ 进行一次完整的前向传播。该操作的算术强度（OPS/Byte）受批量大小限制。对于批量大小较小的延迟敏感型应用，算术强度会显著低于机器平衡点 Pope 等人 [2023], Kwon 等人 [2023]。**NPU** 处理器采用大量并行计算单元（ Cube Cores ），专为稠密矩阵乘法优化。然而，延迟主要由从 HBM 获取权重所需的内存访问时间决定。这一现象表明，除非从根本上改变算法范式，否则单纯提高计算能力（每秒浮点运算次数）只会让推理延迟的改善逐渐趋于有限 Shazeer [2019]。

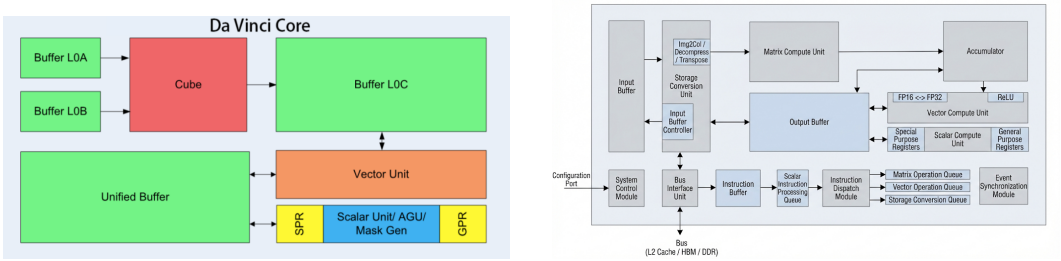
2.2 推测解码范式的演进

推测解码旨在将内存访问频率与令牌生成速率解耦。早期研究 Leviathan 等人 [2023],Chen 等人 [2023] 提出了“草稿模型”概念，即使用较小的辅助模型生成候选序列。尽管理论上合理，草稿模型方法在实际部署中仍面临障碍：在内存中管理两个不同的模型 Spector 和 Re [2023], 会使服务基础设施更加复杂，同时还需要实现分词器对齐。为解决这些问题，人们提出了前瞻解码 Fu 等人 [2024]等“无模型”方法，利用 n 元语法匹配。此外，以 Medusa Cai 等人 [2024] 和 EAGLE Li 等人 [2024], 为代表的“自推测”范式，将草稿生成整合进主模型。Medusa 通过添加辅助头，以极低的开销实现了较高的接受率。然而，其基于树的验证在面向静态图的架构上仍是瓶颈 Miao 等人 [2024]。

2.3 NPU 的架构限制

与支持灵活线程调度的 GPU 不同，**NPU** 处理器侧重于通过静态执行模型实现高吞吐量 Liao 等人 [2021]。如图1所示，其底层架构依赖专门的计算单元层级（Cube 和 Vector 核心）以及显式的内存管理（L0/L1 缓冲区）。**NPU** 软件栈会在离线编译阶段执行最优的算子融合和内存分配。这种**静态形状** 机制假定输入张量形状和计算图拓扑保持不变。推测解码算法涉及条件分支：每一步接受的词元数量以及注意力掩码的形状都会发生变化。在**NPU**上，这种动态性会触发昂贵的即时（JIT）重新编译，或

CPU 回退，造成难以接受的延迟。**NPU** 软件栈执行最优操作 r



(a) 核心架构 (b) 数据流与计算单元

图1：**NPU 处理器的微架构**。(a) 核心将矩阵（Cube）运算与向量运算分开，因此需要分别进行缓冲区管理（L0A/L0B/L0C）。(b) 详细的数据流显示，指令分发和内存传输采用了高度流水线化的设计。这种硬件设计非常适合**静态形状**执行，即数据传输路径在编译时确定，这与 GPU 的动态调度灵活性形成鲜明对比。

并基于这一架构在离线编译阶段执行融合和内存分配。这种称为**静态形状**的机制假定输入张量形状和计算图拓扑在执行期间保持不变。顾名思义，推测解码算法涉及条件分支：每一步接受的词元数量和注意力掩码的形状都会根据验证结果发生变化。在英伟达 GPU 上，这通过动态内核启动来处理。然而，在**NPU**上，这种动态性会迫使运行时回退到中央处理器上的标量执行，或触发昂贵的即时（JIT）重新编译，二者都会带来难以承受的延迟惩罚。因此，要在**NPU** 上实现高效的推测解码，就必须从动态逻辑转向静态的基于张量的表达形式，这正是本研究的核心技术贡献。

3 方法

3.1 多头预测架构

为使 OpenPangu-7B 主干网络在保持架构一致性的同时具备未来预测能力，我们采用非侵入式适配策略。具体而言，我们在骨干模型最终 Transformer 层之后立即附加 K个并行解码头，并将其命名为 Medusa 头。在架构上，每个头被实例化为配备残差连接的多层感知机（MLP），用于将当前时间戳 t 的隐藏状态直接投影为未来词元的概率分布。这些辅助头的训练与冻结的主干网络解耦，以保持原始模型的生成质量。优化目标被表述为所有头上交叉熵损失的加权和。令h_t 表示第 t步的最后隐藏状态， $p_k(h_t)$表示第 k个头对 $t+k+1$ 处词元所预测的概率分布。总损失函数 $\mathcal{L}$ 定义为：

$$\mathcal{L} = \sum_{k=1}^{K} \lambda_k \cdot \mathrm{CE}(p_k(h_t), x_{t+k+1}) \tag{1}$$

其中 λ_k 作为衰减系数。该加权方案反映了预测更远词元时不确定性的增加，从而稳定多头训练过程的收敛。

3.2 NPU 静态实现

在 **NPU** 上部署推测解码的主要障碍，在于硬件偏好静态形状执行。树拓扑的动态变化通常会导致频繁重新编译计算图，从而抵消潜在的加速效果。为解决这一问题，我们提出一种完全静态的实现范式，以消除动态控制流。

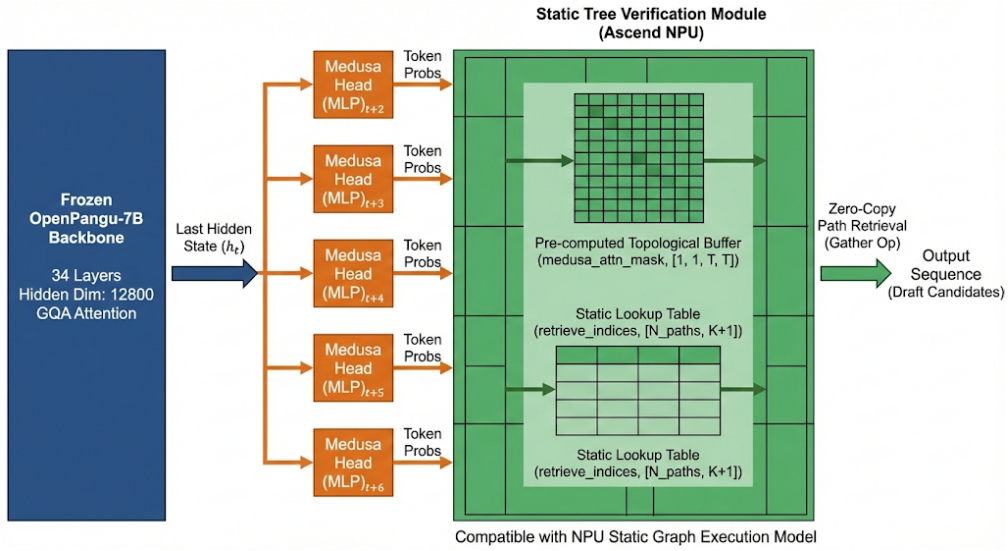


图 2：**适配 NPU 的 OpenPangu 与 Medusa 架构概览**。该系统包括：(a) 冻结的 OpenPangu 主干网络，以及用于分块式词元预测的轻量级 MLP 头；(b) 静态树验证模块。后者使用预先计算的拓扑缓冲区（medusa_attn_mask）实现零拷贝路径检索，确保与 NPU 的静态图执行模型兼容。

树拓扑张量化。 我们不在运行时动态构建验证树，而是预先离线计算其拓扑结构，并将其映射为静态张量。我们定义一个维度为 $[1, 1, T, T]$ 的静态可见性矩阵 medusa_attn_mask，用于编码树中所有候选路径的父子依赖关系。同时，我们构建一个 tree_indices 张量，在展平后的树节点与其在键值缓存中的对应位置之间建立确定性映射。在初始化阶段将这些不变张量加载到 NPU 内存中后，无论推测验证结果如何，计算图都能保持不变，从而使编译器能够执行积极的算子融合。

零拷贝检索策略。传统实现通常依赖中央处理器端的指针操作来重构已接受词元序列，这一过程会产生显著的同步开销，并中断 NPU 执行流水线。我们引入零拷贝检索机制以绕过这一瓶颈。我们构建了一个大小为$[N_{paths}, K+1]$的静态查找表 retrieve_indices，其中存储了每条可能候选路径对应的精确内存偏移量。在验证阶段，根据接受准则确定最优路径索引后，模型利用片上 Gather 算子，通过查找表直接从候选缓冲区提取最终词元序列。该设计使所有数据移动都在高带宽 NPU 内存中完成，实现零拷贝路径重构，并最大限度地利用静态执行图。

4 实验与分析

4.1 实验设置

硬件环境。 实验评估是在配备**NPU** 处理器的高性能计算集群上进行的（处理器运行固件，并搭载于鲲鹏 920 中央处理器上）。为提供跨架构基线，我们还使用 CUDA 11.8 在 NVIDIA RTX A6000 图形处理器（Ampere 架构）上进行了对比基准测试。 **研究模型：OpenPangu-7B**。所有实验的研究模型均为**openPangu-Embedded-7B-V1.1**，这是一款在**NPU** 生态系统上从头训练的高效大语言模型（LLM），Chen 等人 [2025]。该模型使用 25 万亿个词元进行预训练，并具备独特的“快慢思考”自适应能力。与通用的基于 Llama 的架构不同，OpenPangu-7B 专门针对

国产硬件特性进行了优化。模型的详细架构参数汇总于表1。 **实现细节**。 在训练辅助 Medusa 头时，我们采用了

Table 1: **openPangu-Embedded-7B-V1.1 的架构规格**。（非嵌入参数 -eter约占 7B）。

属性	**配置**
架构	稠密
参数（非嵌入）	7B
层数	34
隐藏维度	12,800
注意力机制	分组查询注意力（GQA）
注意力头数量	Q 为 32，KV 为 8
词汇表大小	153k
上下文长度（原生）	32k
预训练词元	25T

使用 AdamW 优化器，学习率为 $1e-3$，全局批次大小为 64。所有推理测试均采用 Float16 精度。**评估指标**。为定量评估所提方案的效率，我们定义了三个主要指标。首先，**端到端加速比**表示与标准自回归解码相比，本方法在实际运行时间上的改善。其次，**接受率（AC）**衡量每个解码步骤中平均验证并接受的词元数量。最后，**开销比**量化推测采样和树验证过程所引入的额外计算成本。这些指标之间的关系形式化如下：

$$\text{Speedup} = \frac{\text{AC}}{\text{Overhead}} \tag{2}$$

$$\text{Overhead} = \frac{\text{Time}_{speculative}}{\text{Time}_{autoregressive}} \tag{3}$$

Where $\text{Time}_{speculative}$ 表示单个 Medusa 解码步骤的延迟，而 $\text{Time}_{autoregressive}$ 表示标准单token生成步骤的延迟。

4.2 自蒸馏数据规模的影响

预测头与骨干模型之间的对齐对于获得较高的接受率至关重要。为研究 Medusa 头对数据的依赖性，我们使用公开数据集（ShareGPT）和自定义自蒸馏数据集训练了多种配置。自蒸馏数据集的构建方式是：使用 ShareGPT 中的查询提示 OpenPangu-7B 主干网络，并收集其输出 logits 作为软标签。此过程确保注意力头能够学习宿主模型特定的概率分布。我们将训练数据规模从 2k 扩大到 50k 个样本，并评估预测token 的 Top-1 准确率。表 2 总结了不同训练配置下的性能。最初仅依赖公开 ShareGPT 数据集（2k 个样本）的实验效果不佳，第一头的 Top-1 准确率仅为 62.40%，这表明通用对话数据与模型的特定行为之间存在分布偏移。引入自蒸馏

表 2：使用不同训练数据集的 Medusa 头性能比较

数据集大小（样本）	**数量** 注意力头	**训练使用自蒸馏?**	**保留特殊标记?**	**Top-1 准确率 (头 1)**	**Top-1 准确率 (头 2)**	**训练时间 (小时)**
2k	3	No	No	62.40%	42.00%	7.45
10k	5	Yes	No	67.80%	49.30%	63.00
50k	**5**	**Yes**	**Yes**	**74.60%**	**54.10%**	**65.70**

（10k 个样本）将准确率提升至 67.80%，但对齐仍不完美。深入分析发现，初始蒸馏过程无意中筛除了 OpenPangu 的特殊控制标记（例如，用于标记内部“思考”状态或起止边界的词元）。因此，注意力头未能学会主干网络的结构格式规范。在最终配置（50k 个样本）中，我们在数据生成和训练阶段明确保留了这些特殊标记。这项优化证明至关重要，使头 1 的 Top-1 准确率提升至 **74.60%**，头 2 的 Top-1 准确率提升至 **54.10%**。这一结果表明，对于 OpenPangu 这类专用大语言模型，预测

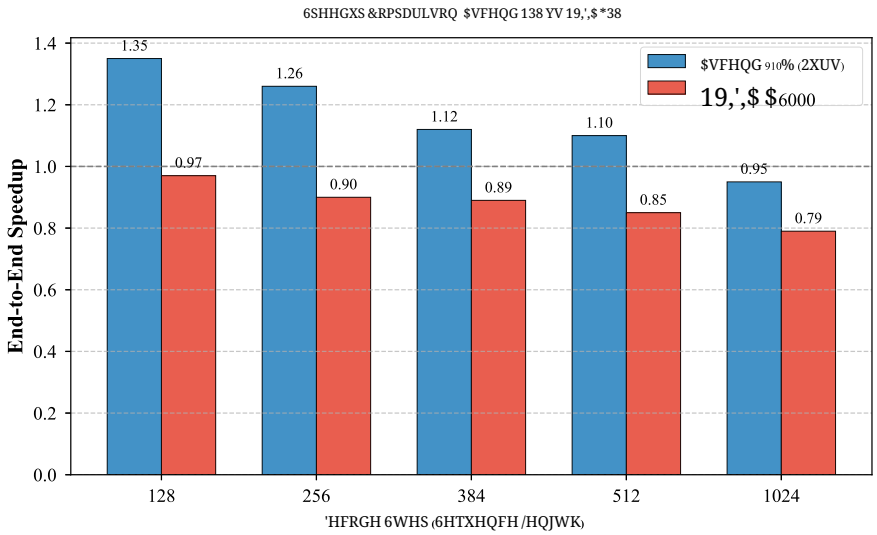
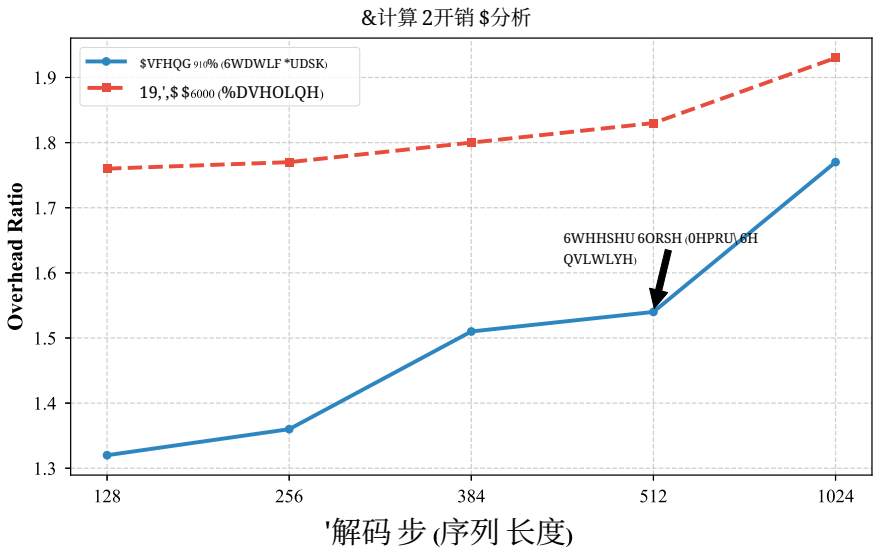


图3：**端到端加速比比较（NPU 与英伟达 A6000）**。所提出的方法在短序列（$L = 128$）上通过**NPU** 实现了最高 **1.35**×的加速比，显著优于未经优化的 GPU 基线。然而，随着序列长度增加，NPU 上的加速比呈下降趋势，最终在 $L = 1024$ 处越过盈亏平衡点。

头必须经过训练，不仅要模仿语义内容，还要准确复现宿主模型在词元级格式上的特征。



'解码 步 (序列 长度)

图4：**计算开销分析**。该图展示了推测解码时间与标准自回归时间的比值。与图形处理器（红色虚线）相比，**NPU**（蓝线）呈现出更陡的斜率，表明其对内存访问模式更加敏感。在 NPU 的长上下文场景中，这种开销的非线性增长是限制加速比的主要因素。

4.3 NPU 深入性能分析

我们进一步分析了系统在不同序列长度下的动态性能特性。在**NPU** 平台上，我们的方法在短序列中展现出稳定的加速比**1.35**×

（解码步128）。这一性能优势归功于静态图中矩阵运算的高效并行化，同时开销比保持在较低水平（1.32）。然而，随着序列长度增加，性能呈现出明显的变化趋势。接受率相对稳定（仅从1.78略微下降至1.65），但计算开销呈非线性增长。如图4所示，当序列长度为1024时，**NPU** 上的开销达到1.77。这一现象凸显了硬件固有的内存墙限制。随着键值缓存线性增长，注意力机制的内存访问模式变得愈发稀疏，且对带宽的需求不断增加。与GPU基线相比，**NPU**虽然计算能力强大，但对这些受内存限制的操作更为敏感。因此，在长上下文场景中，加速比增益会逐渐减小，这表明未来的优化应优先关注内存访问模式，例如键值缓存压缩或分块策略，而非单纯优化计算。

5 结论

本文提出了一种适用于**NPU**硬件的 OpenPangu-7B 端到端加速方案。通过引入**静态树构建** 和**零拷贝检索**，我们解决了动态推测解码与 NPU 静态图执行之间的不兼容问题。结果显示，在 **NPU** 上实现了**1.35 倍加速比**。未来的工作将聚焦于长上下文场景下的算子融合和内存优化。

参考文献

Reza Yazdani Aminabadi、Samyam Rajbhandari、Ammar Ahmad Awan、Cheng Li、Du Li、Elton Zheng、Olatunji Ruwase、Shaden Smith、Minjia Zhang、Jeff Rasley 等。Deepspeed-inference：以前所未有的规模实现 Transformer 模型的高效推理。第 1–15 页，2022 年。

Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared D Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell 等。语言模型是少样本学习者，2020 年。

Tianle Cai、Yuhong Li、Zhengyang Geng、Hongwu Peng、Jason D Lee、Deming Chen 和 Tri Dao。Medusa：采用多个解码头的简单大语言模型推理加速框架。*arXiv* 预印本 *arXiv:2401.10774*，2024年。

Charlie Chen、Sebastian Borgeaud、Geoffrey Irving、Jean-Baptiste Lespiau、Laurent Sifre 和 John Jumper。使用推测采样加速大语言模型解码。*arXiv* 预印本 *arXiv:2302.01318*，2023。

Hanting Chen、Yasheng Wang、Kai Han、Dong Li、Lin Li、Zhenni Bi、Jinpeng Li、Haoyu Wang、Fei Mi、Mingjian Zhu 等。Pangu embedded：具备元认知能力的高效双系统大语言模型推理器。*arXiv* 预印本 *arXiv:2505.22375*，2025年。

Aakanksha Chowdhery、Sharan Narang、Jacob Devlin、Maarten Bosma、Gaurav Mishra、Adam Roberts、Paul Barham、Hyung Won Chung、Charles Sutton、Sebastian Gehrmann 等。Palm：利用路径扩展语言建模。机器学习研究期刊，24(240)：1–113，2023。

Tri Dao。Flashattention-2：通过更好的并行性和工作划分实现更快的注意力。*arXiv* 预印本 *arXiv:2307.08691*，2023。

Tri Dao、Dan Fu、Stefano Ermon、Atri Rudra 和 Christopher Ré。Flashattention：具备 IO 感知能力的快速、节省内存的精确注意力。神经信息处理系统进展，35：16344–16359，2022。

Yichao Fu、Peter Bailis、Ion Stoica 和 Hao Zhang。使用前瞻解码打破大语言模型推理的顺序依赖。*arXiv* 预印本 *arXiv:2402.02057*，2024年。

Sehoon Kim、Karttikeya Mangalam、Suhong Moon、Jitendra Malik、Michael W Mahoney、Amir Gholami 和 Kurt Keutzer。使用大小解码器进行推测解码。神经信息处理系统进展，36:39236–39256，2023。

Woosuk Kwon、Zhuohan Li、Siyuan Zhuang、Ying Sheng、Lianmin Zheng、Cody Hao Yu、Joseph Gonzalez、Hao Zhang 和 Ion Stoica。使用 PagedAttention 高效管理大语言模型服务的内存。发表于第*29*届操作系统原理研讨会论文集，第611–626页，2023。

Yaniv Leviathan、Matan Kalman 和 Yossi Matias。通过推测解码加速 Transformer 推理。发表于国际机器学习大会，第 19274–19286 页。PMLR，2023。

Yuhui Li、Fangyun Wei、Chao Zhang 和 Hongyang Zhang。EAGLE：推测采样需要重新思考特征不确定性。*arXiv* 预印本 *arXiv:2401.15077*，2024年。

Heng Liao、Jiajin Tu、Jing Xia、Hu Liu、Xiping Zhou、Honghui Yuan 和 Yuxing Hu。昇腾：面向普及型深度神经网络计算的可扩展统一架构：工业赛道论文。发表于*2021*年*IEEE*高性能计算机体系结构国际研讨会（*HPCA*），第 789–801 页。IEEE，2021年。

Xupeng Miao、Gabriele Oliaro、Zhihao Zhang、Xinhao Cheng、Zeyu Wang、Zhengxin Zhang、Rae Ying Yee Wong、Alan Zhu、Lijie Yang、Xiaoxiang Shi 等。SpecInfer：通过基于树的推测推理和验证加速大语言模型服务。发表于第*29*届*ACM*编程语言与操作系统架构支持国际会议论文集，第*3*卷，第 932–949 页，2024年。

Onur Mutlu、Justin Meza和Lavanya Subramanian。主存系统：挑战与机遇。2015年。

Reiner Pope、Sholto Douglas、Aakanksha Chowdhery、Jacob Devlin、James Bradbury、Jonathan Heek、Kefan Xiao、Shivani Agrawal和Jeff Dean。高效扩展Transformer推理。机器学习与系统会议论文集，5:606–624，2023。

Alec Radford、Jeffrey Wu、Rewon Child、David Luan、Dario Amodei、Ilya Sutskever等。语言模型是无监督多任务学习器。*OpenAI*博客，1(8):9，2019年。

Noam Shazeer。快速Transformer解码：只需一个写入头。*arXiv*预印本 *arXiv:1911.02150*，2019年。

Benjamin Spector和Chris Re。利用分阶段推测解码加速大语言模型推理。*arXiv*预印本 *arXiv:2308.04623*，2023。

Hugo Touvron、Louis Martin、Kevin Stone、Peter Albert、Amjad Almahairi、Yasmine Babaei、Nikolay Bashlykov、Soumya Batra、Prajjwal Bhargava、Shruti Bhosale 等。《Llama 2：开放基础模型与微调聊天模型》。*arXiv* 预印本 *arXiv:2307.09288*，2023。

Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。《Attention Is All You Need》。神经信息处理系统进展，第 30 卷，2017年。

Samuel Williams、Andrew Waterman 和 David Patterson。《Roofline：一种适用于多核架构的直观性能模型》。*ACM* 通讯，52(4):65–76，2009年。