第 62 卷　第 5 期
2026 年 9 月

南京大学学报(自然科学)
JOURNAL OF NANJING UNIVERSITY
(NATURAL SCIENCE)

Vol. 62, No. 5
Sept., 2026

DOI:10.13232/j.cnki.jnju.2026.05.007

基于EasyControl的DiT可控文生图模型高性能推理研究

唐承捷[1],高嘉伟[1,2*],李国源[1]

(1.山西大学计算机与信息技术学院,太原,030006;2.计算智能与中文信息处理教育部重点实验室,太原,030006)

摘　要:随着DiT (Diffusion Transformer)在可控视觉生成领域的广泛应用,引入外部控制分支(如EasyControl)带来的计算冗余与碎片化算子访存瓶颈,限制了模型在大规模工业级场景下的高效推理.针对多分支架构面临的显存带宽墙与图调度延迟问题,提出一套软硬协同的全栈式高性能推理架构.首先,在算法层面提出多分支特征切片与时空冗余自适应裁剪策略,剥离控制流中的冗余计算;其次,在算子层面深度定制全局融合算子,将高频碎片化的全局显存读写转化为片上连续计算,降低了访存开销;最后,在底层系统层面,设计面向图编译的FP8首轮校准与静态缩放冻结量化部署方案.实验表明,该全栈加速方案在跨分辨率实验中保持了较好的视觉一致性,将端到端推理吞吐量平均提升了约2.5倍,为可控视觉生成模型的推理速度优化提供了解决方案.

关键词:扩散型Transformer,推理加速,算子融合,FP8量化

中图分类号:TP183　　**文献标志码**:A

Research on high-performance inference for easycontrol-based controllable text-to-image DiT models

Tang Chengjie[1], Gao Jiawei[1,2*], Li Guoyuan[1]

(1. School of Computer and Information Technology, Shanxi University, Taiyuan, 030006, China; 2. Key Laboratory of Computational Intelligence and Chinese Information Processing, Ministry of Education, Taiyuan, 030006, China)

Abstract: With the widespread adoption of DiT (Diffusion Transformer) in controllable visual generation, the computational redundancy introduced by external control branches, such as EasyControl, and the memory-access bottlenecks caused by fragmented operator invocations have constrained efficient inference in large-scale industrial-grade scenarios. To address the challenges posed by multi-branch architectures, namely, GPU memory bandwidth limitations and graph-scheduling latency, we propose a software-hardware co-designed, full-stack high-performance inference architecture. First, at the algorithmic level, we introduce a multi-branch feature-tiling strategy combined with adaptive spatiotemporal redundancy pruning to eliminate redundant computations within the control flow. Second, at the operator level, we deeply customize global fusion operators, transforming frequent, fragmented global memory reads and writes into on-chip computation, thereby reducing memory-access overhead. Finally, at the system level, we design an FP8-based first-round calibration and static scaling quantization deployment scheme tailored for graph compilation. Experimental results demonstrate that this full-stack acceleration approach maintains strong visual consistency across resolutions, boosting end-to-end inference throughput by approximately 2.5× on average, thus providing a viable solution for optimizing inference speed in controllable visual generation models.

Keywords: diffusion transformer, inference acceleration, operator fusion, FP8 quantization

基金项目:国家自然科学基金(62472269,62072291,62576198)

收稿日期:2026－06－28

*通信联系人,E-mail:gjw@sxu.edu.cn

近年来，人工智能生成内容（Artificial Intelligence Generated Content，AIGC）在多媒体数据挖掘、图像处理与智能信息生成领域取得了快速发展，特别是以扩散型Transformer[1−2]（Diffusion Transformer，DiT）为代表的架构，凭借其较好的可扩展性和生成质量，逐渐取代传统U-Net[3]结构成为视觉生成的主流基座模型．在实际的智能应用场景中，为了满足对生成内容的精细化控制，研究者引入EasyControl[4]等控制机制，通过外加网络或适配器如LoRA（Low-Rank Adaptation[5]）等，实现对生成图像的深度、边缘或语义特征等特定条件的精准引导．这种可控文生图技术为高质量视觉数据的合成提供了基础工具．

然而，面向大规模工业级部署与联机数据处理时，基于EasyControl的可控DiT模型面临高性能推理与轻量化挑战．首先，在算法与结构层面，动态加载的控制权重（如LoRA）和针对每一步去噪过程的条件特征注入，引入大量冗余计算与控制分支开销．其次，在访存与算子层面，控制特征与主干网络的融合涉及大量碎片化的逐元素（Element-wise）操作，导致显存带宽出现瓶颈．同时，随着生成分辨率的提升，全局注意力机制的计算复杂度呈二次方增长，进一步加剧了计算负担．最后，在底层执行层面，传统的动态图调度和常规精度（如FP16/BF16）计算没有充分利用现代GPU的张量计算核心算力，且存在显著的系统调度延迟．这些问题共同导致了较高的端到端推理延迟和显存占用，较大地限制了该类模型在海量信息处理中的可伸缩性．

针对上述挑战，本文立足于大模型优化与轻量化部署，提出一套面向EasyControl的软硬协同高性能推理方法．在算法层面，提出一种低额外开销的控制权重预合并与条件特征动态跳过机制，并结合基础词元的QKV（Query，Key，Value）优化与旋转位置编码切片，从源头削减计算图的冗余度．在算子层面，针对模型推理特性设计定制化的Triton[6]融合算子，以缓解访存瓶颈．在底层编译层面，实现基于首轮校准与静态缩放冻结的FP8量化推理机制[7]，并与PyTorch[8]静态图编译结合，利用了现代GPU的低精度算力．

本文的主要贡献如下．

（1）提出算法级轻量化与重构方法．通过LoRA权重静态折叠、冗余条件计算跳过以及注意力词元切片优化，有效降低可控生成过程中的结构开销与理论计算量，实现模型推理的轻量化．

（2）构建缓解带宽瓶颈的高性能算子级协同机制．针对EasyControl的特征注入路径设计细粒度融合算子，减少全局显存读写次数，改善高分辨率生成场景下的执行效率．

（3）实现硬件感知的低精度编译与高效部署．开发具有静态缩放冻结机制的FP8线性层，并可与图编译技术兼容．

实验表明，该全栈加速方案整体保持较高的结构相似性和视觉一致性，并将端到端推理吞吐量平均达到基线的约2.5倍，为可控文生图模型的高效部署提供了解决方案．

1　相关工作

针对视觉生成模型在推理阶段计算成本高昂的问题，近年来，国内外诸多研究团队围绕词元精简、模型剪枝、特征缓存、稀疏注意力以及可控架构定制等多个维度展开了深入探索．

对于词元中存在的大量相似冗余信息，有研究提出ToMe（Token Merging）[9]的加速方法．为了解决大型视觉生成模型推理成本高昂的问题，ToMe在Transformer块的注意力和多层感知机分支之间引入轻量级的二分软匹配算法，利用自注意力机制中的键计算余弦相似度，在不需要重新训练的前提下，逐步合并模型内部包含冗余信息的相似词元．这一设计不仅避免了传统动态剪枝方法因改变词元数量而无法进行批量推理的缺陷，还能在仅损失少量精度的条件下提升吞吐量．然而，ToMe的特征合并假设基于图像空间的平滑性，直接合并词元容易削弱外部控制信号中包含的高频细粒度结构．

在扩散模型的结构剪枝方面，DiP-GO[10]框架提供一种通过少量梯度优化步骤来实现剪枝的创新方法，将模型剪枝转化为一个子网搜索过程．首先，基于标准扩散模型构建一个包含特征备份连接的超网，以捕捉相邻时间步之间的特征相似性潜力．随后，引入插件式的剪枝网络，通过精心设计的稀疏性与一致性优化损失，在少量梯度更

新步骤内精准识别并剔除冗余计算块. 虽然该方法能在免重训的前提下实现显著提速,但其优化局限于模型深度维度的块级粗粒度剪枝,缺乏针对不同输入区域特征变化的自适应感知能力.

为了在时间维度上提高对重要词元的处理效率,TokenCache[11]方法引入基于词元级别的细粒度时序特征复用框架,其核心模块缓存预测器通过动态评估多步去噪过程中的词元重要性得分,在词元、网络块与时间步三个维度上实现了自适应的缓存调度. 通过跳过低重要性词元的当前计算并直接复用其历史特征,可降低自注意力的计算开销. 然而,该机制依赖于有损的近似特征缓存,不仅引入额外的预测网络参数与计算开销,且在时间步中不断累积的局部误差会影响高保真多模态任务的像素级对齐精度.

对于稀疏注意力优化,SpargeAttn[12]的免训练通用稀疏加速方法设计了两阶段在线滤波机制. 首先,通过计算 Query 和 Key 块内的自相似性对词元进行选择性压缩,快速预测注意力图的稀疏区块;其次,在 GPU 线程束级别,利用在线 Softmax 局部与全局最大值的差异动态剔除趋零的冗余计算. 结合 Hilbert 曲线重排与 8 比特量化,该全栈式设计可为各类大模型提供加速,但 SpargeAttn 优化仅作用于标准的自注意力计算流内部,无法覆盖较大的残差连接与 FFN 分支.

针对系统性的控制型文生图推理效率,OminiControl2[13]旨在解决 DiT 在处理长条件输入和多条件生成时面临的计算效率瓶颈. 该框架通过紧凑词元表示减少参与计算的条件词元数量,并通过条件特征复用在首步计算后全程静态缓存条件特征. 为了解决静态缓存与动态去噪的不匹配问题,模型在训练阶段引入非对称注意力掩码以确保特征一致性,但该方法失去了免训练部署的灵活性,且在复杂的多种控制条件组合下,特征压缩与静态复用策略牺牲了生成保真度,指标出现一定退化.

综上,无论是词元精简、时序缓存还是注意力稀疏化,现有加速策略大多针对基础主干网络内部的特定计算流进行局部算法优化. 但在面对引入外部控制网络或动态参数微调的复杂可控扩散模型时,这些方法均有一定的短板:没有针对控制分支,有效识别并剔除冗余的条件注入计算;没有良好解决控制特征频繁注入主干引发的大量碎片化逐元素操作及动态挂载权重带来的底层调度延迟,使得模型面临显存带宽墙的限制. 同时,在底层硬件部署层面,尽管 8 位浮点数量化正逐渐成为大模型推理的新标准,但现有的动态缩放量化方案在面对带有复杂控制流的生成架构时,往往难以在底层算子级别实现高效融合与低延迟调度,限制了端到端推理优化带来的加速上限. 表明仅停留在算法结构的裁剪已难以满足推理速度的需求,凸显了从计算图重构、全局算子融合到底层低精度高效部署进行全栈优化的必要性.

2 问题分析

为了明确可控文生图推理中的主要性能瓶颈,首先对 768×1024 分辨率、25 步采样和 FA4 基线配置下的端到端推理过程进行阶段耗时剖析. 结果显示,DiT 去噪阶段占端到端耗时的 96.27%,条件缓存预热占 1.63%,VAE 解码占 1.05%,其余文本嵌入、控制图编码、图像预处理、后处理与调度器更新等阶段合计约占 1%. 表明基于 EasyControl 的可控 DiT 推理瓶颈主要集中在 DiT/Transformer 主体计算,有必要进一步从算法架构的系统局限、碎片化算子访存与下发瓶颈两个方面对其性能瓶颈进行分析.

2.1 算法架构的系统局限 在可控生成架构的设计上,EasyControl 摒弃传统 ControlNet[14-15]引入较大额外分支的做法,转而在扩散模型内部引入轻量级的条件注入模块与独立计算的条件分支. 此外,通过引入因果注意力机制,该框架在初始时间步对条件特征的键值对进行了一次性计算与持续缓存,避免了条件特征的重复提取. 然而,深入剖析其底层推理计算图可以发现,这种多分支协同的架构在动态推理阶段仍存在不可忽视的算法级冗余,主要体现在两个方面.

(1)动态 LoRA 注入的矩阵乘法算力损耗. 为了保证预训练文本与噪声特征空间的完整性,EasyControl 采用并行计算策略,仅对条件分支的输入表示 Z_c 施加低秩矩阵适配. 在标准的推理流程中,条件分支的 Query 投影特征被定义为标准线性变换与 LoRA 变换的叠加:

$$Q_c = W_Q Z_c + B_Q A_Q Z_c$$

其中，W_Q为预训练共享权重，A_Q与B_Q为特定控制条件的低秩矩阵．尽管这种设计在参数量上极为高效，但在实际的动态推理阶段，由于引入额外的低秩矩阵连乘操作，模型需要在运行时实时分配计算单元以执行额外的矩阵乘法．由于生成任务往往需要同时加载多种控制条件（如边缘、深度等），散落在各个计算块中的动态LoRA计算会线性叠加，产生不可忽视的算力与调度开销．

（2）注意力计算的时空冗余与KV缓存的局限性．EasyControl的另一项核心优化是基于因果掩码实现的条件KV缓存，其虽然免去了条件特征Z_c自身的重复前向传播，但在多步去噪循环中，主干网络（噪声与文本分支）的Query依然需要在每一层、每个时间步与较大的缓存条件KV进行全量的注意力交叉计算．从扩散模型的去噪动力学来看，模型对外部控制信号的依赖呈现显著的时序与空间衰减特性．在早期去噪阶段，模型主要依赖控制信号构建宏观的低频布局．在后期去噪阶段，当图像的高频纹理与微观结构已在隐空间中确立时，继续强制主干网络在每个注意力头中与全量的高频控制特征进行注意力计算，不仅无法带来生成质量的实质性提升，反而会引发算力浪费．这种持续全量注意力机制，没有有效利用扩散过程中的时序冗余，构成了当前可控DiT推理加速的一大算法级瓶颈．

2.2　碎片化算子访存与下发瓶颈　DiT模型凭借其较大的参数量与密集的注意力机制，在宏观上常被视为计算密集型模型．然而，在引入EasyControl等复杂的可控生成架构后，系统在微观执行层面的性能瓶颈发生了显著转移．为了实现多条件信号的和谐共存，EasyControl在自注意力计算前执行了密集的张量拼接操作，这种多分支特征的制备与融合，产生了大量的碎片访存．

（1）碎片化算子的高带宽显存读写．在构建上述拼接特征Q, K, V以及注入控制信号的过程中，模型需要频繁调用层归一化、张量切片、张量拼接、旋转位置编码应用以及各种偏置加法等操作．在原生PyTorch等主流深度学习框架的动态图执行机制下，每个逐元素或归约操作都会触发一次独立的GPU内核下发．对于现代GPU架构，张量计算单元的运算速度远超显存的物理读取速度．上述大量碎片化算子的存在，迫使较大的高维特征张量在片外高带宽显存与片上高速缓存之间进行极其频繁的反复搬运．在处理高分辨率图像带来的超长词元序列时，这种数据传输的延迟占据了绝大部分的执行时间，导致张量计算单元处于等待状态．

（2）FlashAttention[16]优化盲区与算子融合的迫切性．尽管当前的大模型推理系统已广泛集成FlashAttention技术以缓解注意力机制的二次方访存复杂度$\mathrm{Softmax}\left(QK^{\top}/\sqrt{d_k}+M\right)V$，但根据阿姆达尔定律[17]，当核心注意力计算被极度加速后，其外围的“数据制备”与“特征路由”操作便成为新的性能短板．FlashAttention的加速域仅局限于标准的注意力计算图内部，难以覆盖EasyControl架构中为了实现多分支条件注入而散落在网络各处的层归一化、LoRA乘加以及位置编码等算子，意味着单纯依赖现有的标准化加速库，已无法完全利用硬件算力．为了打破这一访存壁垒，需要突破标准算子边界，在寄存器或SRAM级别对这些碎片化操作进行全局重构与算子融合，以缓解中间变量的全局内存读写开销．

3　主要方法

本节提出一种涵盖底层算子特征切片与融合、算法级时空冗余自适应裁剪以及基于首轮校准的FP8静态缩放量化部署方案的全栈加速框架，旨在缓解可控DiT模型在多分支推理中的访存带宽墙与图调度瓶颈．

3.1　预备知识　基座模型Flux[18]是基于扩散模型DiT的生成架构，其通过双流与单流注意力块联合处理潜空间图像词元z_t与文本词元c．在线性映射阶段，各模态特征通过线性层投影为查询、键、值张量，并在自注意力计算时物理拼接为统一序列：

$$Q = \left[Q_{\mathrm{img}};\ Q_{\mathrm{txt}}\right]$$

$$K = \left[K_{\mathrm{img}};\ K_{\mathrm{txt}}\right]$$

$$V = \left[V_{\mathrm{img}};\ V_{\mathrm{txt}}\right]$$

此外，为了支持任意分辨率的图像生成，模型在注意力计算前对查询和键施加旋转位置编码

(RoPE)[19]. 在较大的序列拼接架构下,包含QKV投影与FFN在内的线性映射构成了该模型最主要的算力开销.

为了实现可控DiT生成,EasyControl将外部控制信号编码为条件词元z_{cond},并通过低秩适配器将其注入预训练的投影层. 其投影计算的数学形式定义为:

$$y = Wx + \alpha \cdot W_{\text{up}} W_{\text{down}} x$$

其中,W为冻结的基础权重,W_{up}与W_{down}为条件分支专用的低秩矩阵. 为了加速推理过程,采用两阶段推理缓存策略. 在初始去噪步时,模型执行完整计算并持久化记录条件词元的键值对$(K_{\text{cond}}, V_{\text{cond}})$;在后续去噪步中,条件词元跳过QKV的动态计算,直接以缓存形式拼接入全局序列参与注意力交互:

$$K = \left[K_{\text{img}};\ K_{\text{txt}};\ K_{\text{cond}}\right]$$

$$V = \left[V_{\text{img}};\ V_{\text{txt}};\ V_{\text{cond}}\right]$$

8位浮点数通过缩减位宽降低激活显存占用并提升张量计算吞吐量,是新一代模型部署的标准格式. 在FP8量化计算中,将低精度张量还原为高精度特征的反量化过程定义为:

$$X = X_{\text{FP8}} \times S_X$$

其中,S_X为该张量的缩放因子.

3.2 多分支特征的全局算子融合与切片 在传统的受控生成推理范式中,图像分支h_{img}与条件分支h_{cond}通常在进入线性层前被物理拼接(即$\text{FFN}\left(\left[h_{\text{img}};\ h_{\text{cond}}\right]\right)$). 这种批量拼接不仅造成序列长度的膨胀,还带来冗余矩阵乘法计算,其频繁的内存拷贝操作还物理阻断了底层的算子融合链路. 为了缓解这一设计带来的开销,本文提出一种计算流解耦与多分支独立切片策略,将图像分支与条件分支的执行流进行逻辑解耦,使其分别作为独立的张量通过相同的线性层权重进行处理:

$$y_{\text{img}} = \text{FFN}\left(h_{\text{img}}\right)$$

$$y_{\text{cond}} = \text{FFN}\left(h_{\text{cond}}\right)$$

解耦后,每条分支的内部计算拓扑在内存访问上恢复了连续性,为后续执行端到端的内核融合提供基础. 分支的分离使条件特征的处理在计算图层面成为一个正交的独立模块,为后续(3.3)引入时空冗余裁剪机制奠定结构基础. 系统可直接在控制流中旁路条件分支,尽量不影响主图像分支的计算连贯性.

同样,在单流块中,该策略将原有的"拼接-投影-拆分"流程$\text{proj_out}\left(\left[h_{\text{img}};\ h_{\text{cond}}\right]\right)$重构为并行的独立映射路径$\text{proj_out}\left(h_{\text{img}}\right)$与$\text{proj_out}\left(h_{\text{cond}}\right)$.

通过分支切片恢复计算的连续性后,基于Triton深度,定制针对Transformer块的全局融合算子. 该策略抛弃了框架原生的标准算子调用方式,转而通过在片上高速缓存级别重构计算流,将连续的碎片化逐元素操作合并为单次执行.

以双流块为例,FFN的原始计算路径包含高度碎片化的独立内核执行流,在引入定制化算子融合后,将上述路径的内存读写重构为三个深度融合的执行流. 在此重构范式下,密集的GEMM操作交由底层高性能计算库(如cuBLAS)执行,所有外围的特征路由与规范化操作从8个独立的算子被压缩至3个定制的Triton融合算子. 融合算子将激活张量从全局显存加载至片上高速缓存后,直接在内部连续完成计算,待数值稳定后一次性写回. 这一底层重构显著减少了中间临时张量的HBM读写开销,降低了CPU侧的内核下发调度延迟,如图1所示.

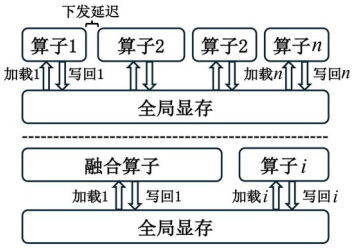


图1 碎片化算子多次访问全局内存,算子间存在下发延迟
Fig. 1 Fragmented kernels incur repeated global memory accesses and inter-kernel launch overhead

3.3 时空冗余自适应裁剪 如前所述,EasyControl在预热阶段将条件词元的键值对缓存至bank_kv,同时将其注意力输出缓存. 在后续的去噪迭代阶段,条件词元的信息直接以缓存读取的形式参与全局注意力交互. 然而,在原始架构的物理实现中,条件分支在去噪阶段依然会执行完整的前向计算链路:

$$h_{\text{cond}}^{l+1}=h_{\text{cond}}^{l}+\text{Gate}\cdot\text{FFN}\left(\text{Norm}\left(h_{\text{cond}}^{l}+\text{Gate}\cdot a_{\text{cond}}^{l}\right)\right)$$

其中，a_{cond}^{l}直接采自缓存，而非实时计算获得．本文从信息流拓扑的角度对该计算链路的必要性进行深度的系统级剖析．在去噪阶段，条件分支的特征演化链路是一个自包含的闭环，其内部状态的更新不向主图像分支传递额外有效信息，因而不显著影响最终图像分支的计算结果．在第l层注意力处理器的去噪路径中，主图像分支与条件分支的跨模态交互主要局限于两个物理接口．(1)键值对供给．图像查询(Query)对条件信息的注意力聚合通过$K_{\text{cond}}, V_{\text{cond}} \leftarrow \text{bank_kv}^{l}$实现，该张量在预热阶段已被持久化固化，不受当前步条件分支隐状态h_{cond}^{l}演化的影响．(2)注意力输出替代．条件词元自身的注意力计算结果$a_{\text{cond}}^{l} \leftarrow \text{bank_attn}^{l}$同样直接从静态缓存中被读取，由此可知，条件分支隐状态h_{cond}^{l}在当前层的更新，其主要作用是作为$l+1$层条件分支的输入h_{cond}^{l+1}．而在第$l+1$层的注意力交互中，条件分支对图像分支的信息供给被静态的bank_kv^{l+1}与bank_attn^{l+1}替代，条件分支的隐层演化对图像生成流不再显式参与主图像分支的动态更新．

基于上述信息流冗余分析，结合扩散模型在去噪后期的低频固化特性，本文提出一种两级渐进裁剪策略．

在去噪阶段的注意力处理器中，系统停止对多分支拼接张量执行统一的线性映射，转而仅在基底图像词元上计算QKV投影：

$$Q=W_Q\cdot h_{\text{img}}$$

$$K=\left[W_K\cdot h_{\text{img}};\ \text{bank_kv}_K\right]$$

$$V=\left[W_V\cdot h_{\text{img}};\ \text{bank_kv}_V\right]$$

条件词元的KV直接由静态缓存拼接提供，绕过了较大的线性投影层重计算，削减了约25%的QKV投影计算量．同时，为了解决切片造成的特征维度不对齐问题，引入旋转位置编码(RoPE)维度对齐．此时，查询张量Q仅包含图像词元(长度为S_{img})，而键张量K包含图像与缓存的条件词元(总长度为$S_{\text{img}}+S_{\text{cond}}$)，传统的全量位置编码将触发维度不匹配．因此，系统对RoPE的相位矩阵θ执行自适应维度切片：

$$Q_{\text{rot}}=\text{RoPE}\left(Q,\theta_{[:S_{\text{img}}]}\right)$$

$$K_{\text{rot}}=\text{RoPE}\left(K,\theta_{[:S_{\text{img}}+S_{\text{cond}}]}\right)$$

在条件分支全链路裁剪上，进一步在系统调度层减少条件分支在每个DiT块中的大部分动态计算．在预热阶段($t=0$)收集完所有必需的静态缓存后，系统在流水线的去噪阶段($t\geqslant 1$)直接向DiT控制流下发指令，各执行块在检测到条件输入后在计算图中自动旁路条件相关的计算分支．

3.4　面向图编译的低精度量化部署　以NVIDIA Hopper架构为代表的新一代GPU引入对8位浮点数张量计算核心的原生支持，其理论算力峰值(1513 TFLOPS)可达BF16精度的两倍．为了释放这一硬件能力，本文设计面向编译器的特定量化模块FP8Linear，以兼容式替换DiT主干中的标准线性层．

(1)数据格式与权重量化．采用E4M3浮点格式(4位指数，3位尾数)，其最大可表示值为$V_{\max}=448$，范围覆盖$[-448, 448]$．在模型加载阶段，系统对权重矩阵$W\in\mathbb{R}^{D_{\text{out}}\times D_{\text{in}}}$执行一次性静态量化，通过计算全局绝对最大值获取缩放因子s_W，进而将权重转换为FP8格式：

$$s_W=\frac{\|W\|_{\infty}}{V_{\max}}$$

$$W_{\text{fp8}}=\text{Cast}_{\text{E4M3}}\left(\text{Clamp}\left(\frac{W}{s_W},-V_{\max},V_{\max}\right)\right)$$

量化后的权重以FP8格式驻留显存，直接将模型参数的访存带宽占用削减一半．

(2)激活量化与底层GEMM执行．对于前向传播中的动态输入激活$x\in\mathbb{R}^{M\times K}$，标准的Per-tensor量化流程同样需要计算缩放因子并执行类型转换．在完成操作数准备后，底层调用torch._scaled_mm接口执行带缩放的FP8矩阵乘法，$y=s_x\cdot s_W\cdot\left(x_{\text{fp8}}\otimes W_{\text{fp8}}^{\mathrm{T}}\right)$，其中，$\otimes$表示硬件原生的FP8张量核心乘法，其输出结果在片上寄存器被直接累加至BF16高精度，从而在加速计算的同时降低了输出阶段精度溢出和数值损耗的风险．

(3)选择性替换策略．不是所有的线性投影都适合极低精度量化，为了保护EasyControl条件注入模块中低秩适配器的敏锐控制能力，采用选择性替换策略，仅对输入与输出维度均不小于

256的全连接层执行FP8Linear替换,从而在加速比与生成保真度之间取得了较好平衡.

(4)首轮校准与静态缩放冻结. 尽管FP8矩阵乘法本身极其高效,但上述针对激活张量x的标准动态量化流程会引入额外的运行时调度开销. Nsight Systems的底层分析表明,标准动态量化流程依赖的三步额外操作($\|x\|_\infty$归约计算、缩放除法、类型转换)在默认模式下会被发射为独立的Triton内核,意味着每个FP8Linear层会额外产生三个内核调用,在单步去噪循环中会产生约1500次额外的内核启动开销. 此类强数据依赖会增加张量核心等待时间,并可能影响计算图捕获的稳定性. 为了缓解运行时的归约内核与图编译冲突,设计基于离线校准的静态缩放冻结机制,其核心部署流水线分三个严格依序的阶段.

离线校准. 以待生成图像的目标配置执行一次完整推理,并记录每个FP8Linear层在各个时间步前向传播时的激活绝对最大值$a_{\max}^{(t)}$.

冻结操作. 系统在首轮校准的基础上提取全局最大激活值,并引入安全裕度超参数γ,根据首轮校准阶段记录的激活峰值计算静态缩放因子:

$$s_x^{\text{static}}=\frac{\gamma\cdot\max_t a_{\max}^{(t)}}{V_{\max}},\gamma=1.1$$

其中,γ为安全裕度超参数,可吸收不同提示词、随机种子和控制条件带来的激活分布轻微波动,降低FP8量化过程中发生数值截断的风险. 冻结操作完成后,所有前向传播直接硬编码读取预计算的s_x^{static},从运行时计算图中移除极大值归约操作.

图融合编译. 在确认所有首轮校准得到的静态缩放因子被冻结为静态常数后,启动torch.compile执行全局图捕获. 此时,由于极值归约的动态性已被清除,编译器能够将剩余的缩放除法与FP8转换操作作为Element-wise算子,融合进其相邻的Triton内核中.

4 实验评估

通过详尽的端到端基准测试与细粒度消融分析,验证本文提出的全栈加速框架在显著降低推理延迟以及维持生成质量方面的系统级有效性.

4.1 实验配置与测试指标 配备NVIDIA Tesla H100 PCIe GPU的服务器平台,具体实验环境配置如表1所示. 以端到端推理时间为为性能测试指标,*FID*(Fréchet Inception Distance),*PSNR*(Peak Signal-to-Noise Ratio),*SSIM*(Structural Similarity Index Measure)为生成一致性指标,对比基线为FlashAttention,数据集包括EasyControl Samples,MS-COCO2017[20]验证集(取500张生成轮廓图),生成分辨率包括512×512,768×1024,1024×1024,随机种子固定为42.

表1 实验环境配置

Table 1 Experimental environment configuration

配置	配置环境
CPU	Intel (R) Xeon (R) Platinum 8468V
GPU	NVIDIA Tesla H100 PCIe
操作系统	Ubuntu 22.04.5 (Linux 5.15.0)
CUDA	12.8 (Driver 570.133)
PyTorch	2.5.0
Triton	3.1.0

4.2 主要结果 全面评估在应用多分支特征的全局算子融合与切片、时空冗余自适应裁剪以及面向图编译的低精度量化部署等全栈策略后,模型在端到端推理中的表现. 为了验证该系统级优化的泛化能力与鲁棒性,在512×512,768×1024与1024×1024三种典型的生成分辨率下进行详尽的基准测试,*FID*,*PSNR*和*SSIM*均以FlashAttention基线生成结果为参考,衡量本文方法与基线输出之间的一致性. 实验结果如表2所示,表中箭头的方向表示性能更优的方向. 由表可见,本文构建的全栈优化框架在跨维度的空间分辨率下均实现了显著加速效果,并展现出高度稳定的系统扩展性. 在常规的512×512分辨率下,单图生成延迟从原始的4.5 s减少至1.7 s,加

表2 不同分辨率下的端到端推理性能与生成质量评估

Table 2 Evaluation of end-to-end inference performance and generation quality across different resolutions

分辨率	*FID*↓	*PSNR* (dB)↑	*SSIM*↑	加速比↑
512×512	28.7	31.5 (±4.5)	0.98 (±0.03)	2.6×
768×1024	8.6	38.3 (±5.8)	0.99 (±0.01)	2.4×
1024×1024	29.9	29.1 (±5.7)	0.96 (±0.05)	2.5×

速比为2.6×，取得28.7的*FID*，31.5 dB的*PSNR*和0.98的*SSIM*，展现了良好的一致性．对于中间尺寸768×1024，优化架构将推理延迟从8.8 s降低至3.7 s，加速比为2.4×，与基线相比，该分辨率下*FID*为8.6，*PSNR*为38.3 dB，*SSIM*为0.99．当面临空间维度翻倍、计算负载呈平方级膨胀的1024×1024百万像素生成任务时，优化架构依然将推理延迟从12.1 s减少至4.8 s，加速比为2.5×．优化后的模型输出在所有分辨率下均取得了基本的结构和内容一致性，如图2所示．



图2　同提示词不同分辨率下基线与加速后生成效果对比

Fig. 2　Comparison of baseline and accelerated generation results under the same prompt at different resolutions

4.3　消融实验　为了深入量化全栈加速框架中各独立模块的性能贡献，在768×1024核心分辨率下进行了逐级递进的消融实验，结果如图3所示．以集成FlashAttention-4的原生EasyControl模型（单图推理耗时8.8 s）为基准，首先，引入全局算子融合后，系统有效缓解了多分支架构的显存带宽墙，将延迟降低至5.7 s（加速为1.5×）；随后，叠加时空冗余自适应裁剪机制，通过在控制流层面直接旁路条件分支的无效运算，实现算法层面的计算削减，使延迟进一步下探至4.5 s（累计加速2.0×）；最后，在接入面向图编译的静态FP8低精度量化后，利用张量计算核心的计算能力，将最终延迟降低至3.7 s（对应约1.22×的额外加速），实现了端到端2.4×的总加速比．上述结果表明本文提出的“底层访存重构-算法冗余剥离-硬件低精度”全栈协同设计的必要性与系统级有效性．

表3展示了不同优化模块对生成一致性的影响，表中的箭头表示性能更优的方向．由表可见，算子融合阶段对生成结果的影响较小．FA4+算子融合相对于FA4基线取得了36.2 dB的*PSNR*和0.99的*SSIM*，说明该阶段主要改变算子执行路径，没有显著改变模型输出．进一步引入自适应裁剪后，*PSNR*仍保持在35.1 dB，*SSIM*保持在0.99，表明裁剪策略仅引入了较小的数值扰动，生成结果在结构层面仍与基线高度一致．最终，在叠加FP8低精度量化后，本文提出的完整方法在500张样本上的*PSNR*为29.1 dB，*SSIM*为0.96，说明低精度量化虽然带来一定误差，但整体生成结构和视觉内容仍保持较好一致性．

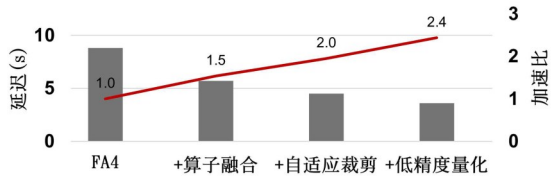


图3　全栈加速策略的逐步消融实验与性能增益分析

Fig. 3　Progressive ablation study and performance gain analysis of the full-stack acceleration strategies

表3　不同优化模块对生成一致性的影响

Table 3　Impact of different optimization modules on generation consistency

方法	样本数	*PSNR* (dB)↑	*SSIM*↑
FA4+算子融合	10	36.2	0.99
FA4+算子融合＋自适应裁剪	10	35.1	0.99
FA4+算子融合＋自适应裁剪＋FP8	500	29.1	0.96

4.4　FP8加速评估　尽管底层GEMM算子实现了显著提速，但系统端到端的最终加速比仍受限于GEMM在整体计算链路中的耗时占比．表4展示了应用“融合＋裁剪”优化后DiT块内各项操作的耗时分解．汇总模型内19个双流块和38个单流块的数据可知，GEMM操作在去噪的总延迟中占比约为65%．当核心GEMM算子提速达$S_{\mathrm{GEMM}}=2.0$时，系统的理论最大加速比S_{total}高于图3中实测的综合加速表现：

表 4 融合与裁剪优化后的操作耗时分解(BF16)

Table 4 Breakdown of operation latency post-fusion and pruning (BF16)

操作	双流/单流块耗时(μs)	FP8可用性
AdaLN+Scale+Shift	188/571	N
QKV+norm_mlp投影	526	Y
ctx_proj+proj_out	165/1080	Y
FFN	700	Y
FA4注意力机制	467/462	N
门控残差	280/100	N
GEMM耗时综合占比	65%	/

$$S_{\text{total}}=\frac{1}{(1-\alpha)+\alpha/S_{\text{GEMM}}}\approx 1.5$$

实际FP8阶段带来约1.22×的额外增益,低于理论上限,主要受非GEMM算子、量化转换、图编译融合边界以及内核启动开销等因素的影响,但整体趋势与理论分析一致.剩余约35%的未被GEMM加速覆盖的时间主要由FA4注意力机制、Triton融合内核、门控残差以及底层内核启动开销等部分构成.

5 结论

本文针对可控DiT模型在推理中面临的显存带宽墙与图编译调度瓶颈,提出一种端到端的全栈加速框架.通过底层多分支特征切片与定制化Triton算子融合,减少了碎片化内核带来的高频访存开销;基于时空自适应裁剪机制,剥离了控制流中的冗余拷贝和计算;同时,结合面向图编译的FP8首轮校准与静态缩放冻结量化策略,缓解了动态极值统计与编译器追踪之间的系统级冲突,提高了低精度张量计算资源利用率,并缓解了访存受限与图调度开销问题.

未来将把该软硬协同优化的底层范式扩展至更高维度的长序列视频生成以及动态稀疏注意力架构中,推动大参数量多模态可控生成模型向低延迟、可部署的实际应用场景发展.

参考文献

[1] Ho J, Jain A N, Abbeel P. Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems, 2020, 33: 6840-6851.

[2] Peebles W, Xie S N. Scalable diffusion models with transformers// 2023 IEEE/CVF International Conference on Computer Vision. Paris, France: IEEE, 2023: 4195-4205.

[3] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation// International Conference on Medical Image Computing and Computer-Assisted Intervention. Munich, Germany: Springer, 2015: 234-241.

[4] Zhang Y X, Yuan Y R, Song Y R, et al. EasyControl: Adding efficient and flexible control for diffusion transformer// 2025 IEEE/CVF International Conference on Computer Vision. Honolulu, HI, USA: IEEE, 2025: 19513-19524.

[5] Hu E J, Shen Y L, Wallis P, et al. LoRA: Low-rank adaptation of large language models// The 10th International Conference on Learning Representations. Online: ICLR, 2021: 1-13.

[6] Tillet P, Kung H T, Cox D. Triton: An intermediate language and compiler for tiled neural network computations// Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages. New York, NY, USA: Association for Computing Machinery, 2019: 10-19.

[7] Micikevicius P, Stosic D, Burgess N, et al. FP8 formats for deep learning. http://arxiv.org/abs/2209.05433, 2022-09-29.

[8] Ansel J, Yang E, He H, et al. PyTorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation// Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. Volume 2. New York, NY, USA: Association for Computing Machinery, 2024: 929-947.

[9] Bolya D, Fu C Y, Dai X L, et al. Token merging: Your vit but faster. https://arxiv.org/abs/2210.09461v3, 2022-10-17.

[10] Zhu H W, Tang D H, Liu J, et al. DiP-go: A diffusion pruner via few-step gradient optimization// The 38th Conference on Neural Information Processing Systems. Vancouver, Canada: NeurIPS, 2024: 1-24.

[11] Zou C，Liu X Y，Liu T，et al. Accelerating diffusion transformers with token-wise feature caching//The 13th International Conference on Learning Representations. Singapore：ICLR，2024：1－22.

[12] Zhang J T，Xiang C D，Huang H F，et al. SpargeAttention：Accurate and training-free sparse attention accelerating any model inference// Proceedings of the 42nd International Conference on Machine Learning. New York，NY，USA：PMLR，2025：76397－76413.

[13] Tan Z X，Xue Q C，Yang X Y，et al. OminiControl2：Efficient conditioning for diffusion transformers. http://arxiv.org/abs/2503.08280，2025－03－11.

[14] Zhang L M，Rao A Y，Agrawala M. Adding conditional control to text-to-image diffusion models//2023 IEEE/CVF International Conference on Computer Vision. Paris，France：IEEE，2023：3836－3847.

[15] Zhao S H，Chen D D，Chen Y C，et al. Uni-controlnet：All-in-one control to text-to-image diffusion models. Advances in Neural Information Processing Systems，2023，36：11127－11150.

[16] Zadouri T，Hoehnerbach M，Shah J，et al. FlashAttention-4：Algorithm and kernel pipelining co-design for asymmetric hardware scaling. http://arxiv.org/abs/2603.05451，2026－03－05.

[17] Amdahl G M. Validity of the single processor approach to achieving large scale computing capabilities//Proceedings of the Spring Joint Computer Conference. New York，NY，USA：Association for Computing Machinery，1967：483－485.

[18] Labs B F，Batifol S，Blattmann A，et al. FLUX.1 kontext：Flow matching for in-context image generation and editing in latent space. http://arxiv.org/abs/2506.15742，2025－06－24.

[19] Su J L，Lu Y，Pan S F，et al. RoFormer：Enhanced transformer with rotary position embedding. http://arxiv.org/abs/2104.09864，2023－11－08.

[20] Lin T Y，Maire M，Belongie S，et al. Microsoft COCO：Common objects in context//The 13th European Conference on Computer Vision. Zurich，Switzerland：Springer，2014：740－755.

（责任编辑　杨可盛）