重构残差连接：DeepSeek mHC 架构中的几何与数学原理

大家好，我是居丽叶。

今天我们来深度解读 DeepSeek 在元旦发布的一篇论文 **mHC：Manifold-Constrained Hyper-Connections**。

这项工作解决了一个一直存在的问题：为了提升大模型性能，字节曾尝试使用比标准残差连接（Residual Connection）更复杂的 Hyper-Connections (HC)，但这破坏了**恒等映射**属性，导致梯度爆炸和训练崩溃。DeepSeek 的解法非常数学化——通过引入 **Sinkhorn-Knopp 算法**，将连接矩阵投影到 **Birkhoff 多面体**（双随机矩阵流形）上，从底层原理上保证了训练的稳定性。

本文将深度拆解背后的数学原理与工程落地，目录如下：

- 为什么大模型需要重构残差连接?
- 无约束连接带来的稳定性难题
- 基于流形约束的数学解法
- 算法实现：Sinkhorn-Knopp 迭代与投影
- 实验结果

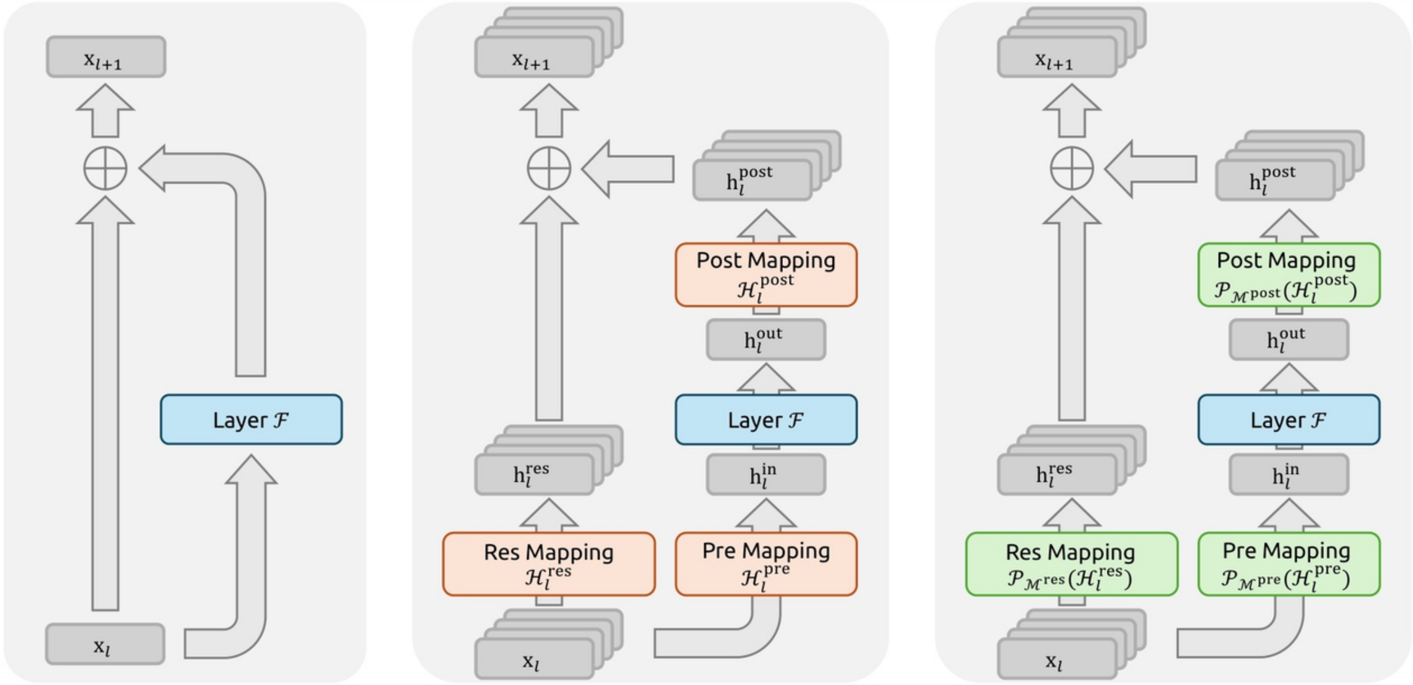
为什么大模型需要重构残差连接?

在深入理解 DeepSeek 的 mHC 之前，我们需要先回到一切的起点：**深度神经网络是如何传递信息的?**

传统的DNN网络包含很多的层，每一层都可以理解为是一个函数：

$$x_{l+1} = \mathcal{F}(x_l, W_l)$$

x_l、$x_{l+1} \in R^{1\times C}$ 分别表示第 l 层的输入输出，W_l 表示这一层的参数，$\mathcal{F}()$ 表示映射关系（比如卷积、注意力等）。**随着层数的堆叠，信息在传递过程中会逐渐衰减，导致深层网络难以训练。**



(a) Residual Connection (b) Hyper-Connections (HC) (c) Manifold-Constrained HC (*m*HC)

Residual Connection

为了解决这个问题，ResNet 引入了**残差连接（Residual Connection）**。它在每层的输出上直接加上输入项，如上图（a）：

$$x_{l+1} = x_l + \mathcal{F}(x_l, W_l)$$

$$\mathbf{x}_L = \mathbf{x}_i + \sum_{l=i}^{L-1} \mathcal{F}(\mathbf{x}_l, W_l)$$

这种设计具有**恒等映射（Identity Mapping）**特性，使得浅层的信息能直接传递到深层，保证了模型训练过程中的稳定性。

然而，在大模型时代，这种设计面临着一个瓶颈：**只有这一条快捷通道，只能传递有限的信息**。

Hyper-Connections（HC）

为了突破这一限制，Hyper-Connections (HC) 应运而生。**通过扩展残差流的宽度并增强连接的复杂性，HC 在不改变计算开销的前提下，显著提升了拓扑复杂性**，如上图（b）：

$$\mathbf{x}_{l+1} = \mathcal{H}_l^{\mathrm{res}} \mathbf{x}_l + \mathcal{H}_l^{\mathrm{post}^T} \mathcal{F}(\mathcal{H}_l^{\mathrm{pre}} \mathbf{x}_l, W_l)$$

其中 x_l、 x_{l+1} 分别表示第 l 层的输入输出，特征维度拓展到 $n \times C$， n 表示拓展率，可以视为n路的残差，通过调整 n 的大小控制残差流的宽度，从而实现了模型扩宽。

- $\mathcal{H}_l^{\mathrm{res}} \in R^{n \times n}$ 学习不同深度之间的特征
- $\mathcal{H}_l^{\mathrm{pre}} \in R^{1 \times n}$ 将来自 nC 维的特征聚合为 C 维给到 $\mathcal{F}$
- $\mathcal{H}_l^{\mathrm{post}} \in R^{1 \times n}$ 将 $\mathcal{F}$ 的输出反向操作。

虽然增加了更多可学习的参数 $\mathcal{H}_l^{\mathrm{res}}$、$\mathcal{H}_l^{\mathrm{post}}$、$\mathcal{H}_l^{\mathrm{pre}}$，但并没有增加 $\mathcal{F}$ 的计算量，且由于 n 的值远小于 C，这种设计在不显著增加计算量的前提下，大幅提升了模型的表达能力。

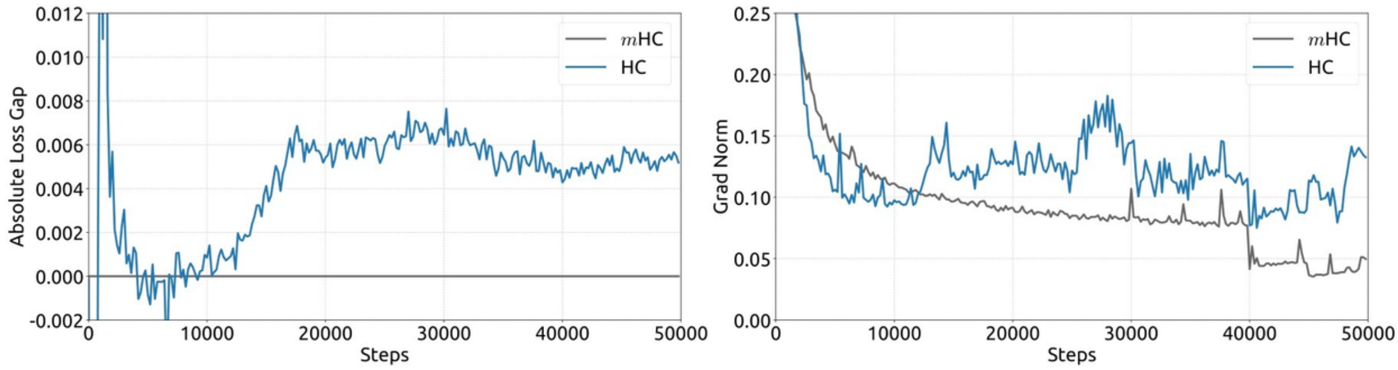
无约束连接带来的稳定性难题

虽然 HC 拓宽了信息通路，但其引入的复杂性也带来了**两个副作用**：

1. **网络架构多层扩展之后，会破坏残差连接的恒等映射特性**：

$$\mathbf{x}_L = \left(\prod_{i=1}^{L-l} \mathcal{H}_{L-i}^{\text{res}}\right) \mathbf{x}_l + \sum_{i=l}^{L-1} \left(\prod_{j=1}^{L-1-i} \mathcal{H}_{L-j}^{\text{res}}\right) \mathcal{H}_i^{\text{post}\top} \mathcal{F}(\mathcal{H}_i^{\text{pre}} \mathbf{x}_l, W_l)$$

从浅层 l 到深层 L 的信号被复合映射 $\left(\prod_{i=1}^{L-l} \mathcal{H}_{L-i}^{\text{res}}\right)$ 决定，但后者由于不受约束，不能保证恒等映射，这种差异会导致前向和反向传播中信号被无限放大或者衰减，导致梯度爆炸或者消失。如下图所示，HC在训练到12k步时损失激增，这与梯度范数的不稳定性高度相关。



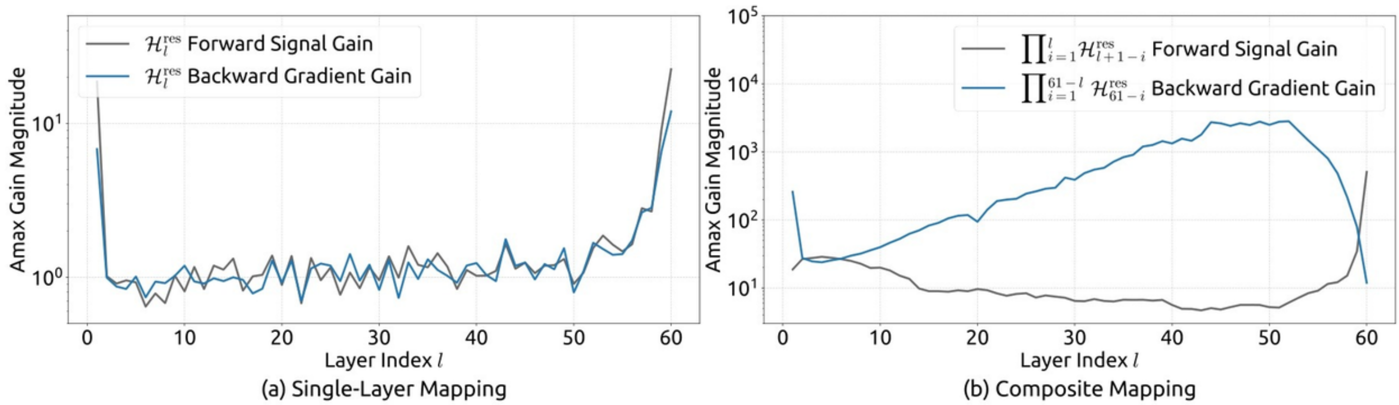
(a) Absolute Training Loss Gap $vs.$ Training Steps (b) Gradient Norm $vs.$ Training Steps

> HC相比于mHC的训练不稳定性：（a）HC相对于mHC的绝对损失差距，（b）梯度范数

进一步对复合残差映射 $\left(\prod_{i=1}^{L-l} \mathcal{H}_{L-i}^{\text{res}}\right)$ 进行验证，mHC定义了 `Amax` 指标，用于衡量偏离恒等映射的程度，原文中提到：

> we utilize two metrics. The first, based on the maximum absolute value of the row sums of the composite mapping, captures the worst-case expansion in the forward pass. The second, based on the maximum absolute column sum, corresponds to the backward pass. We refer to these metrics as the Amax Gain Magnitude of the composite mapping.
>
> 我们采用两项指标：第一项基于复合映射**行和的最大绝对值**，用于刻画前向传播过程中的**最坏情况扩张程度**；第二项基于**列和的最大绝对值**，对应反向传播过程。我们将这两项指标统称为复合映射的**最大绝对增益幅度**（Amax Gain Magnitude）。

该指标在HC中可以飙升到3000，证实HC容易导致残差流信息爆炸现象，导致前向和反向传播不稳定。



> HC的残差映射在单层和跨层复合后，都会显著偏离恒等映射，在前向和反向传播中可能会导致梯度消失/爆炸。

2. **内存访问增加**：加宽残差流会增大内存访问开销（与 n 近似成正比），过高的I/O需求会显著降低训练吞吐量。此外由于引入了可学习的参数 $\mathcal{H}_l^{res}$、$\mathcal{H}_l^{post}$、$\mathcal{H}_l^{pre}$，反向传播时需要用到中间激活值，也会增大内存占用量。

Method	Operation	Read (Elements)	Write (Elements)
Residual Connection	Residual Merge	$2C$	C
	Total I/O	$\mathbf{2C}$	$\mathbf{C}$
Hyper-Connections	Calculate $\mathcal{H}_l^{pre}, \mathcal{H}_l^{post}, \mathcal{H}_l^{res}$	nC	n^2+2n
	$\mathcal{H}_l^{pre}$	$nC+n$	C
	$\mathcal{H}_l^{post}$	$C+n$	nC
	$\mathcal{H}_l^{res}$	$nC+n^2$	nC
	Residual Merge	$2nC$	nC
	Total I/O	$\mathbf{(5n+1)C+n^2+2n}$	$\mathbf{(3n+1)C+n^2+2n}$

> 每个token的内存访问成本

基于流形约束的数学解法（mHC）

DeepSeek 提出的 mHC (Manifold-Constrained HC) 旨在解决上述稳定性难题。其主要思想非常直观：既然无约束的 $\mathcal{H}_l^{res}$ 像混乱的水流一样不可控，那我们就把它限制在一条规则的“河道”里。

这条“河道”在数学上被称为**流形（Manifold）：**

简要解释一下流形

核心是 **局部平坦，整体复杂** —— 它是一种几何对象，你凑近了看（局部），它和我们熟悉的平坦空间（比如直线、平面）没区别；但拉远了看（整体），它可能是弯曲、闭合或不规则的。本质是用局部的简单平坦，描述整体的复杂形状。

举个例子，比如地球是一个2维流形：

- 从局部看，我们站在地面上，感觉脚下是平的，地球是一个二维平面。

- 从整体看，宇航员眼中的地球是球形，是三维的。

在HC中，$\mathcal{H}_l^{\mathrm{res}}$ 这个整体复杂的残差映射，虽然能促进残差流之间的相互作用，但不同残差流之间可能相互干扰，缺乏稳定性。mHC将 $\mathcal{H}_l^{\mathrm{res}}$ 降低到更低维度的、有规则约束的流形上，相当于把混乱的水流 $\mathcal{H}_l^{\mathrm{res}}$ 限制在局部平坦的河道里，**既保留了HC宽残差流和多通路的优势，保留了残差链接的恒等映射属性。**

mHC将复合残差映射 $\left(\prod_{i=1}^{L-l} \mathcal{H}_{L-i}^{\mathrm{res}}\right)$ 约束在**Birkhoff 多面体（双随机矩阵流形）** $\mathcal{M}^{res}$ 上，必须同时满足以下三个条件：

1. 行加和为1
2. 列加和为1
3. 元素非负

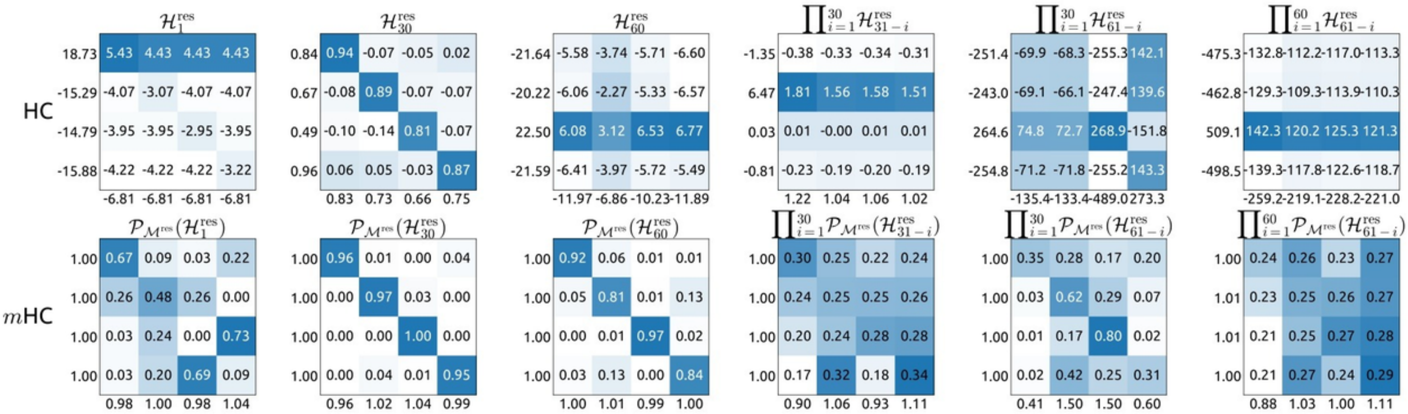
$$\mathcal{P}_{\mathcal{M}^{\mathrm{res}}}\left(\mathcal{H}_l^{\mathrm{res}}\right) := \left\{\mathcal{H}_l^{\mathrm{res}} \in \mathbb{R}^{n \times n} \mid \mathcal{H}_l^{\mathrm{res}} \mathbf{1}_n = \mathbf{1}_n, \mathbf{1}_n^{\top} \mathcal{H}_l^{\mathrm{res}} = \mathbf{1}_n^{\top}, \mathcal{H}_l^{\mathrm{res}} \geqslant 0\right\},$$

$\mathbf{1}_n$ 表示 n 维的全1向量。当 $n=1$ 时，就退化为resnet中的恒等映射。

- 双随机矩阵流形不会增大或缩小 x_l 的信号，这就**保证了能保持恒等映射属性。**
- 双随机矩阵流形没有限制 n 条残差流之间的相互作用，**保持了模型的表达能力**

此外双随机矩阵还有一下优秀的性质**：**

1. **谱范数保持**：谱范数被限制在1以内（即 $\|\mathcal{H}_l^{\mathrm{res}}\| \leq 1$），这意味着可学习的 $\mathcal{H}_l^{\mathrm{res}}$ 映射是非扩张的，也就是这个信号不会被放大，有效缓解了梯度爆炸问题，不会像HC那样放大3000倍。
2. **组合封闭性**：双重随机矩阵相乘得到的还是双随机矩阵。这确保了多层复合残差映射 $\left(\prod_{i=1}^{L-l} \mathcal{H}_{L-i}^{\mathrm{res}}\right)$ 保持双重随机性，增加模型层数不会变的不稳定。
3. **几何解释**：集合 $\mathcal{M}^{res}$ 构成了Birkhoff多面体，即排列矩阵集合的凸包。这个凸包的顶点就是0-1矩阵，也就是说，双随机矩阵相当于对 x_l 进行加权平均，相当于对不同流之间的信息进行混合，这种混合是稳定且信息恒定的。



此外对输入映射 $\mathcal{H}_l^{\mathrm{pre}}$ 和输出映射 $\mathcal{H}_l^{\mathrm{post}}$ 也添加非负约束（通过sigmoid），该约束可防止由正负系数组合引起的信号抵消，也可视为一种特殊的流形投影。

算法实现：Sinkhorn-Knopp 迭代与投影

那么，怎么保证 $\mathcal{H}_l^{\text{res}}$ 能落在双随机流形呢？

给定第 l 层的输入 $x \in R^{n\times C}$ ，首先展开为 $\vec{x} \in R^{1\times nC}$ ，参考HC的公式，得到动态映射和静态映射：

$$\begin{cases} \vec{x}_l' = \text{RMSNorm}(\vec{x}_l) \\ \tilde{\mathcal{H}}_l^{\text{pre}} = \alpha_l^{\text{pre}} \cdot (\vec{x}_l'\varphi_l^{\text{pre}}) + \mathbf{b}_l^{\text{pre}} \\ \tilde{\mathcal{H}}_l^{\text{post}} = \alpha_l^{\text{post}} \cdot (\vec{x}_l'\varphi_l^{\text{post}}) + \mathbf{b}_l^{\text{post}} \\ \tilde{\mathcal{H}}_l^{\text{res}} = \alpha_l^{\text{res}} \cdot \text{mat}(\vec{x}_l'\varphi_l^{\text{res}}) + \mathbf{b}_l^{\text{res}}, \end{cases}$$

$\varphi_l^{\text{res}} \in R^{nC\times n^2}$, φ_l^{pre}、 $\varphi_l^{\text{post}} \in R^{1\times nC}$ 是动态线性映射， $\mathbf{b}_l^{\text{res}} \in R^{n\times n}$, $\mathbf{b}_l^{\text{pre}}$、 $\mathbf{b}_l^{\text{post}} \in R^{1\times n}$ 是静态可学习的偏置映射。 $mat(\cdot)$ 将 $R^{1\times n^2}$ 转换为 $R^{n\times n}$ 。

$$\begin{cases} \mathcal{H}_l^{\text{pre}} = \sigma(\tilde{\mathcal{H}}_l^{\text{pre}}) \\ \mathcal{H}_l^{\text{post}} = 2\sigma(\tilde{\mathcal{H}}_l^{\text{post}}) \\ \mathcal{H}_l^{\text{res}} = \text{Sinkhorn-Knopp}(\tilde{\mathcal{H}}_l^{\text{res}}), \end{cases}$$

$\sigma(\cdot)$ 表示sigmoid函数。 $Sinkhorn$ **算子先将所有元素变为正数（从而保证所有元素非负），再交替对行和列进行缩放，使其和为1**。具体来说，给定一个初始正矩阵 $\mathbf{M}^0 = exp(\tilde{\mathcal{H}}_l^{\text{res}})$ ，迭代过程如下：

$$\mathbf{M}^{(t)} = \mathcal{T}_r\left(\mathcal{T}_c(\mathbf{M}^{(t-1)})\right),$$

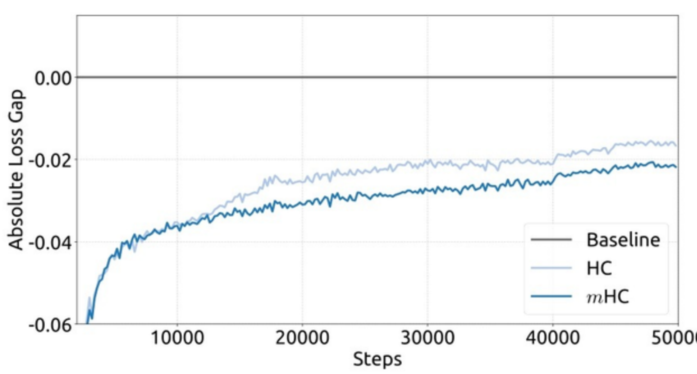
$\mathcal{T}_r$ 和 $\mathcal{T}_c$ 分别表示行归一化和列归一化，通过多次迭代（为了保证效率只迭代20次），虽然不能保证HC完美落到Birkhoff流形上，但也足够接近（上图中很多列的求和不等于1）。

在工程上，这篇论文使用 Kernel Fusion 实现Sinkhorn迭代流程，将分散的操作合并为一个kernel，从而只需要一次HBM读写；采用梯度检查点，反向传播时重新计算激活值，降低内存占用峰值。还扩展了DualPipe通信调度机制，解耦计算与通信依赖，缓解多残差流带来的额外通信延迟。以上所有优化使mHC只增加了6.7%的时间开销。

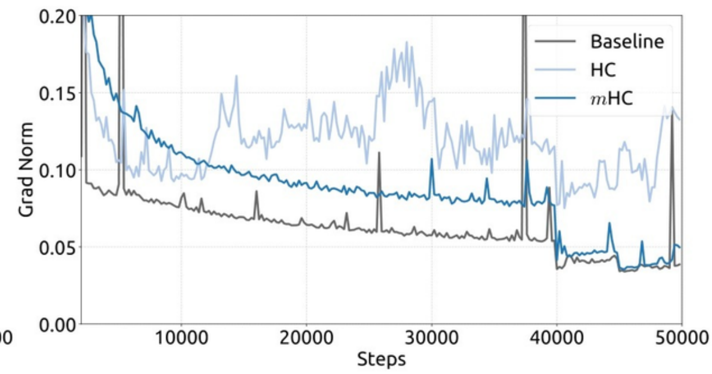
实验结果

稳定性：1.6倍保真

如下图所示，mHC的最终损失降低了0.021。之前介绍过，在HC中，信息从第一层传到最后一层，最高能被放大3000倍。而mHC下，信息放大最高只有1.6倍——几乎是“原样传递”。



(a) Absolute Training Loss Gap *vs.* Training Steps



(b) Gradient Norm *vs.* Training Steps

> mHC训练的稳定性：（a）mHC和HC相比于baseline的绝对损失差距，（b）梯度范数

性能：超越HC

Benchmark (Metric)	BBH (EM)	DROP (F1)	GSM8K (EM)	HellaSwag (Acc.)	MATH (EM)	MMLU (Acc.)	PIQA (Acc.)	TriviaQA (EM)
# Shots	3-shot	3-shot	8-shot	10-shot	4-shot	5-shot	0-shot	5-shot
27B Baseline	43.8	47.0	46.7	73.7	22.0	59.0	78.5	54.3
27B w/ HC	48.9	51.6	53.2	74.3	**26.4**	63.0	79.9	56.3
27B w/ *m*HC	**51.0**	**53.9**	**53.8**	**74.7**	26.0	**63.4**	**80.5**	**57.6**

在多个benchmark上，mHC的性能都比HC更优，原因很简单：之前HC的残差流经常梯度消失或者爆炸；mHC将残差流双随机矩阵流形，保证了恒等映射属性。

成本：只增加 6.7% 的时间

当开启4路残差流时，mHC只比HC多6.7%的训练时间，相当于买了个保险，就换来了极强的训练稳定性。

论文：https://arxiv.org/pdf/2512.24880