arXiv:2601.00417v2 [cs.LG] 29 Jan 2026

深度 Delta 学习

Yifan Zhang[1] **YifengLiu**[2] **MengdiWang**[1] **Quanquan Gu**[2]普林斯顿大学 [2]加州大学洛杉矶分校yifzhang@princeton.edu

1月1日， 2026*

摘要

深度残差网络的有效性取决于恒等捷径连接。虽然这种机制缓解了梯度消失问题，但它也对特征变换施加了严格的加性归纳偏置，限制了网络对复杂隐藏状态转移的建模能力。本文提出**深度 Delta 学习（DDL）**，将快捷连接从固定的恒等映射推广为可学习、依赖于状态的线性算子。由此得到的 Delta 算子是恒等变换的秩一扰动：$\mathbf{A}(\mathbf{X}) = \mathbf{I} - \beta(\mathbf{X})\mathbf{k}(\mathbf{X})\mathbf{k}(\mathbf{X})^\top$，由单位方向$\mathbf{k}(\mathbf{X})$和标量门控 $\beta(\mathbf{X})$参数化。我们通过谱分析证明，$\beta(\mathbf{X})$会在恒等变换（$\beta = 0$）、正交投影（$\beta = 1$）和 Householder 反射（$\beta = 2$）之间连续插值。此外，我们将残差更新改写为同步的秩一 Delta 写入：β 同时缩放当前$\mathbf{k}$分量的移除与新$\mathbf{k}$分量的注入。这种统一形式能够沿依赖于数据的方向显式控制捷径谱，同时保持稳定的训练行为。在实证方面，以 DDL 替换 Transformer 的残差加法可改善验证损失和困惑度，并提升语言建模任务的下游评估准确率；在扩展状态设置下，提升幅度更大。

Project Page: https://github.com/yifanzhang-pro/deep-delta-learning

1 引言

深度残差网络（He 等人，2016年）通过实现前所未有深度模型的稳定训练，代表了神经网络设计的一次范式转变。其核心机制——恒等捷径连接——将每一层重新定义为学习相对于其输入 $\mathbf{F}(\mathbf{X})$ 的残差函数 $\mathbf{X}$。在其标准形式中，残差更新是逐元素相加：

$$\mathbf{X}_{l+1} = \mathbf{X}_l + \mathbf{F}(\mathbf{X}_l). \tag{1.1}$$

我们可以将其视为常微分方程的一个前向欧拉步：$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X})$。这一视角将深度网络与动力系统联系起来（Chen 等人，2018年；Tong 等人，2025年；Li 等人，2022年；Kan 等人，2025年；Jemley，2026年）。严格的加性更新也会对学习到的动力学施加强烈的平移偏置。捷径路径的雅可比矩阵固定为恒等映射。

*修订日期：2026年1月29日

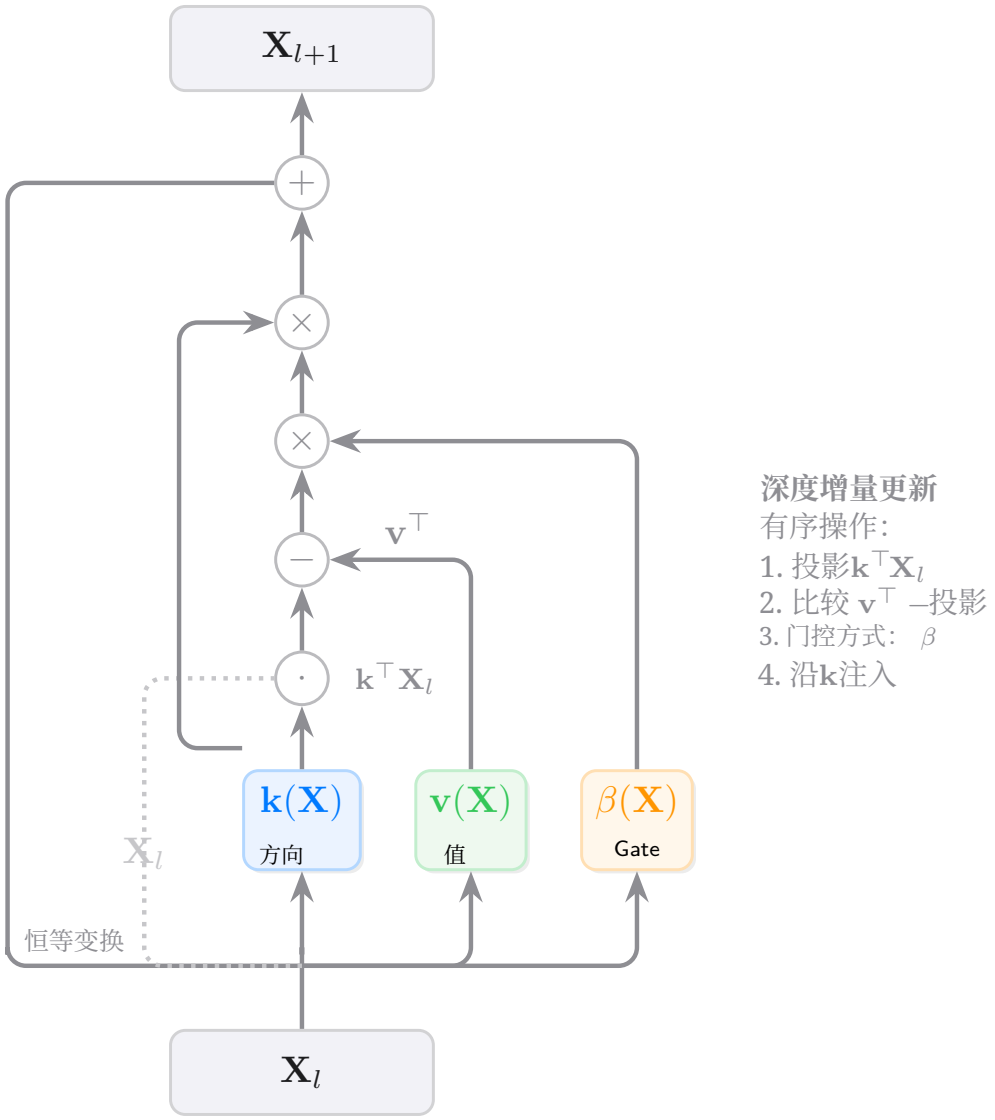


图1 深度增量残差块。该架构对标准残差连接进行了泛化。可学习标量门控 β 控制一种秩一几何变换。

这一视角还揭示了一种结构偏置：跳跃连接会在每层的雅可比矩阵中引入恒等项，使逐层线性化的特征值偏向 $+1$附近。这会使某些转换（尤其是需要某些方向上出现负特征值的转换）更难实现，除非采用较大的残差修正（Grazzi 等人，2025年）。相关研究探索了固定加性跳跃连接的替代方案，以及扩大残差流规模的方法，包括交替更新（AltUp）（Baykal 等人，2023年）和超连接（Zhu 等人，2025年）。相比之下，DDL 保留了单流残差形式，同时通过低秩算子赋予捷径明确可分析的数据依赖谱。

为克服这一局限，我们提出一种以几何线性代数为基础、具有严谨理论依据的残差连接推广形式。我们引入**深度 Delta 学习（DDL）**，其残差块对捷径连接应用状态依赖的恒等变换秩一扰动，并在隐藏状态矩阵中进行

同步秩一写入$\mathbf{X} \in \mathbb{R}^{d\times d_v}$。这一表述将深度与记忆增强架构联系起来，把隐藏状态视为动态值矩阵。关键在于，可学习标量门控 $\beta(\mathbf{X})$ 在恒等映射、正交投影和（当 $\beta = 2$ 时）几何反射之间插值，以确定捷径算子，同时同步缩放写入项。我们的贡献如下：

1. 我们提出了**Delta残差块**，这是一种多分支架构，能够学习对矩阵值捷径连接应用门控（广义）Householder型算子，该算子由学习到的方向$\mathbf{k}(\mathbf{X})$、残差值向量$\mathbf{v}(\mathbf{X})$和学习得到的门控 $\beta(\mathbf{X})$参数化，如图1所示。

2. 我们对 Delta 算子进行了谱分析，推导出其完整的特征系统，并展示了 $\beta(\mathbf{X})$ 如何通过塑造其谱来控制变换。

3. 我们在一个连续可微模块中统一了恒等映射、投影和反射。我们还展示了 DDL 能够还原 Delta 规则更新，其中门控 β 的作用类似于逐层步长。

4. 我们在 124M 和 353M 规模的语言建模任务上，评估了 DDL 作为 Transformer 残差相加的即插即用替代方案，结果表明其在损失、困惑度和下游基准评测方面均持续取得提升；将残差状态扩展至 $d_v = 4$ 可带来最显著的增益。

2 Deep DeltaLearning

我们的方法以 Householder 反射的数学基础为基础，并将其推广为一种可学习的状态依赖算子。

2.1 预备知识：Householder变换

定义 2.1 (Householder矩阵 (Householder，1958年))。对于非零向量$\mathbf{k} \in \mathbb{R}^d$，Householder矩阵 $\mathbf{H}_\mathbf{k}$ 定义如下：

$$\mathbf{H}_\mathbf{k} = \mathbf{I} - 2\frac{\mathbf{k}\mathbf{k}^\top}{\|\mathbf{k}\|_2^2}. \tag{2.1}$$

Geom从几何上看，$\mathbf{H}_\mathbf{k}$ 将任意向量关于法向量为 $\mathbf{k}$.

Householder矩阵是数值线性代数的基石，具有若干重要性质：它是对称的（$\mathbf{H}_\mathbf{k} = \mathbf{H}_\mathbf{k}^\top$）、正交的（$\mathbf{H}_\mathbf{k}^\top\mathbf{H}_\mathbf{k} = \mathbf{I}$），并且是对合的（$\mathbf{H}_\mathbf{k}^2 = \mathbf{I}$）。其谱由一个特征值 -1 和 $d-1$ 个特征值 1 组成。

2.2 Delta 算子的构式

标准的 Householder 反射可写成归一化形式：$\mathbf{H}_\mathbf{u} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$，其中$\mathbf{u}$为任意单位向量。我们从两个方面推广这一构造：(i) 将常数因子 2 替换为可学习的数据依赖标量门控 $\beta(\mathbf{X})$；(ii) 通过学习映射 $\mathbf{k}(\mathbf{X})$，使法向方向依赖于当前状态。由此得到**Delta 残差（Delta-Res）**模块。设隐藏状态为矩阵 $\mathbf{X} \in \mathbb{R}^{d\times d_v}$，其中 d 表示特征维度，d_v 表示值通道的数量。我们将加性残差改为沿$\mathbf{k}$方向对齐的秩一更新。

归一化方向。令 $\widetilde{\mathbf{k}}(\mathbf{X}) \in \mathbb{R}^d$ 为由轻量分支生成的未归一化方向。为便于分析，我们采用单位方向

$$\mathbf{k}(\mathbf{X}) = \frac{\widetilde{\mathbf{k}}(\mathbf{X})}{\|\widetilde{\mathbf{k}}(\mathbf{X})\|_2}, \qquad \widetilde{\mathbf{k}}(\mathbf{X}) \neq \mathbf{0}, \tag{2.2}$$

使得 $\mathbf{k}(\mathbf{X})^\top \mathbf{k}(\mathbf{X}) = 1$。

Delta 算子。 项 $\mathbf{A}(\mathbf{X})$ 是作用于特征维度上的 Delta 算子 d :

$$\mathbf{A}(\mathbf{X}) = \mathbf{I} - \beta(\mathbf{X})\, \mathbf{k}(\mathbf{X})\mathbf{k}(\mathbf{X})^\top. \tag{2.3}$$

在给定当前状态（因此也给定 $\beta(\mathbf{X})$ 和 $\mathbf{k}(\mathbf{X})$) 的条件下，$\mathbf{A}(\mathbf{X})$ 是空间特征空间 $\mathbb{R}^d$ 上的线性映射。整体层映射 $\mathbf{X} \mapsto \mathbf{A}(\mathbf{X})\mathbf{X}$ 是非线性的，因为 β 和 $\mathbf{k}$（以及下文的 $\mathbf{v}$ ）都是$\mathbf{X}$的函数。

Delta-Res 块输出。块输出的计算方式为

$$\mathbf{X}_{l+1} = \mathbf{A}(\mathbf{X}_l)\mathbf{X}_l + \beta(\mathbf{X}_l)\mathbf{k}(\mathbf{X}_l)\mathbf{v}(\mathbf{X}_l)^\top, \tag{2.4}$$

其中，$\mathbf{v}(\mathbf{X}_l) \in \mathbb{R}^{d_v}$ 是值分支 ϕ_v 产生的残差值向量 $\mathbb{R}^{d\times d_v} \to \mathbb{R}^{d_v}$ （即$\mathbf{v}(\mathbf{X}) := \phi_v(\mathbf{X})$）。这里，外积 $\mathbf{k}\mathbf{v}^\top$ 构成秩一写入。关键的是，我们对这一构造项也应用相同的门控 $\beta(\mathbf{X})$，使擦除和写入保持同步。

由于$\mathbf{X}$ 是一个矩阵，算子$\mathbf{A}(\mathbf{X})$会沿值维度 d_v进行广播，将几何变换同时应用于隐藏状态的每一列。

将式 (2.3) 代入式 (2.4)，可得到精确的加法秩一 Delta 形式：

$$\mathbf{X}_{l+1} = \mathbf{X}_l + \beta(\mathbf{X}_l)\, \mathbf{k}(\mathbf{X}_l)\Big(\mathbf{v}(\mathbf{X}_l)^\top - \mathbf{k}(\mathbf{X}_l)^\top \mathbf{X}_l\Big), \tag{2.5}$$

这明确表明，同一个标量 β 协调了 (i) 通过秩一项 $\mathbf{k}(\mathbf{k}^\top\mathbf{X}_l) = (\mathbf{k}\mathbf{k}^\top)\mathbf{X}_l$移除当前的 $\mathbf{k}$ 分量，以及 (ii) 通过 $\mathbf{k}\mathbf{v}^\top$注入新的 $\mathbf{k}$ 分量。在单位范数约定 $\mathbf{k}(\mathbf{X})^\top\mathbf{k}(\mathbf{X}) = 1$ （式 (2.2)）下，$\mathbf{k}(\mathbf{X}_l)^\top\mathbf{X}_l \in \mathbb{R}^{1\times d_v}$ 是一个行向量，其j个元素中的第j个等于 $\mathbf{X}_l$ 的第 j个值列在 $\mathbf{k}(\mathbf{X}_l)$上的正交投影系数。

沿选定方向的投影动力学。 将式 (2.5) 投影到层 l 所用的方向上，可得到一个封闭形式：

$$\mathbf{k}_l^\top \mathbf{X}_{l+1} = (1-\beta_l)\, \mathbf{k}_l^\top \mathbf{X}_l + \beta_l\, \mathbf{v}_l^\top, \tag{2.6}$$

其中 $\beta_l = \beta(\mathbf{X}_l)$、$\mathbf{k}_l = \mathbf{k}(\mathbf{X}_l)$ 和 $\mathbf{v}_l = \mathbf{v}(\mathbf{X}_l)$。特别地，当 $\beta_l = 1$ 该模块会覆盖每个值列的$\mathbf{k}_l$分量：$\mathbf{k}_l^\top \mathbf{X}_{l+1} = \mathbf{v}_l^\top$。

门控函数 $\beta(\mathbf{X})$ 通过对标量logit应用 Sigmoid函数进行参数化，使其取值位于 (0,2) 内：

$$\beta(\mathbf{X}) = 2 \cdot \sigma(\mathrm{Linear}(\mathcal{G}(\mathbf{X}))), \tag{2.7}$$

其中 $\mathcal{G}(\cdot)$ 是池化、卷积或展平操作。随着 logit 饱和，端点行为 $\beta \to 0$ 和$\beta \to 2$ 会逐渐显现；在全文中，我们将情形 $\beta = 0$ 和 $\beta = 2$ 视为这些极限。选择这一特定范围，是因为它具有丰富的几何解释，接下来我们将对此进行分析。

3 DDL 分析

Delta-Res 块的表达能力源于算子$\mathbf{A(X)}$的谱性质，而这些性质由学习得到的门控 $\beta\mathbf{(X)}$控制。为方便记号，我们固定一层 l，并写作 $\beta_l := \beta(\mathbf{X}_l)$、$\mathbf{k}_l := \mathbf{k}(\mathbf{X}_l)$ 和 $\mathbf{v}_l := \mathbf{v}(\mathbf{X}_l)$；分析谱时，我们省略层索引，并将 β 和 $\mathbf{k}$ 视为固定值。

3.1 Delta 算子的谱分解

定理 3.1（Delta 算子的谱）。设 $\mathbf{A} = \mathbf{I} - \beta\mathbf{k}\mathbf{k}^\top$，其中 $\mathbf{k} \in \mathbb{R}^d$ 是一个单位向量（$\mathbf{k}^\top\mathbf{k} = 1$），且 $\beta \in \mathbb{R}$ 是一个标量。$\mathbf{A}$ 的特征值为 1，重数为 $d-1$；以及 $1-\beta$，重数为 1。对应特征值 $1-\beta$ 的特征向量为 $\mathbf{k}$。特征值 1 对应的特征子空间是$\mathbf{k}$的正交补，记作 $\mathbf{k}^\perp = \{\mathbf{u} \in \mathbb{R}^d \mid \mathbf{k}^\top\mathbf{u} = 0\}$。

证明。设$\mathbf{k}$ 正交超平面中的任意向量为$\mathbf{u}$（即，$\mathbf{u} \in \mathbf{k}^\perp$，满足$\mathbf{k}^\top\mathbf{u} = 0$）。将$\mathbf{A}$ 作用于$\mathbf{u}$，得到：

$$\begin{aligned}\mathbf{A}\mathbf{u} &= (\mathbf{I} - \beta\mathbf{k}\mathbf{k}^\top)\mathbf{u} = \mathbf{I}\mathbf{u} - \beta\mathbf{k}(\mathbf{k}^\top\mathbf{u}) \\ &= \mathbf{u} - \beta\mathbf{k}(0) = \mathbf{u} = 1 \cdot \mathbf{u}.\end{aligned}$$

因此，$(d-1)$维子空间$\mathbf{k}^\perp$ 中的任意向量都是特征值为 $\lambda = 1$的特征向量。现在，考虑将$\mathbf{A}$ 作用于向量$\mathbf{k}$ 本身：

$$\begin{aligned}\mathbf{A}\mathbf{k} &= (\mathbf{I} - \beta\mathbf{k}\mathbf{k}^\top)\mathbf{k} = \mathbf{I}\mathbf{k} - \beta\mathbf{k}(\mathbf{k}^\top\mathbf{k}) \\ &= \mathbf{k} - \beta\mathbf{k}(1) = (1-\beta)\mathbf{k}.\end{aligned}$$

因此，$\mathbf{k}$ 是特征值为 $\lambda = 1-\beta$ 的特征向量。由于我们已找到张成 $\mathbb{R}^d$ 的 d 个线性无关的特征向量，因此已刻画了$\mathbf{A}$的完整谱。 □

等价地，对于任意 $\mathbf{u} \in \mathbb{R}^d$，将 $\mathbf{u} = \mathbf{u}_\perp + (\mathbf{k}^\top\mathbf{u})\mathbf{k}$ 分解，其中 $\mathbf{u}_\perp \in \mathbf{k}^\perp$。于是

$$\mathbf{A}\mathbf{u} = \mathbf{u}_\perp + (1-\beta)\,(\mathbf{k}^\top\mathbf{u})\mathbf{k},$$

这明确表明，$\mathbf{A}$ 保持$\mathbf{k}^\perp$ 不变，并将$\mathbf{k}$分量缩放 $(1-\beta)$。

该定理清晰而有力地阐释了门控 $\beta\mathbf{(X)}$。通过学习一个标量，网络可以动态控制残差变换的几何形状，同时作用于状态矩阵的所有d_v 列。

提升到矩阵值状态。上述谱性质是空间层面的：它们描述了$\mathbf{u} \mapsto \mathbf{A}\mathbf{u}$ 在 $\mathbb{R}^d$ 上的线性映射。由于我们的隐藏状态是矩阵$\mathbf{X} \in \mathbb{R}^{d \times d_v}$，且快捷连接通过左乘发挥作用，因此每一列 d_v 都由同一个$\mathbf{A}$独立变换。等价地，在向量化后，诱导线性算子为$\mathbf{I}_{d_v} \otimes \mathbf{A}$。因此，提升后映射的谱由$\mathbf{A}$ 的特征值重复 d_v 次构成，其行列式等于 $\det(\mathbf{A})^{d_v}$。

由于$\mathbf{A}$ 是对称的，其奇异值是其特征值的绝对值，即 1（重数为 $d-1$）和 $|1-\beta|$。特别地，在单位范数假设下，$\mathbf{A}$ 是正交的，当且仅当 $|1-\beta| = 1$，等价地 $\beta \in \{0, 2\}$。采用公式（2.7）中的 Sigmoid 参数化时，这些端点情形会在饱和逻辑值极限下趋近。对于 $\beta \in (0, 2)$，$\mathbf{A}$ 会按幅度压缩$\mathbf{k}$-分量（当 $\beta > 1$ 时还会翻转其符号），同时保持 $\mathbf{k}^\perp$ 不变。

推论 3.2（空间行列式）。 假设满足单位范数条件 $\mathbf{k}(\mathbf{X})^\top\mathbf{k}(\mathbf{X}) = 1$（公式（2.2））。Delta 算子 $\mathbf{A}(\mathbf{X})$ 作用于空间特征 $\mathbb{R}^d$ 时，其行列式为：

$$\det(\mathbf{A}(\mathbf{X})) = 1 - \beta(\mathbf{X}).$$

由于快捷连接会跨值列广播，因此在整个矩阵状态空间上诱导出的行列式为 $\det(\mathbf{A}(\mathbf{X}))^{d_v} = (1 - \beta(\mathbf{X}))^{d_v}$。因此，$\beta(\mathbf{X})$ 控制沿特定空间方向 $\mathbf{k}(\mathbf{X})$ 的有符号体积缩放。具体而言，当 $\beta(\mathbf{X}) \in (1, 2]$ 空间特征值 $1-\beta(\mathbf{X})$ 为负时，沿 $\mathbf{k}(\mathbf{X})$ 会发生符号反转；当 $\beta(\mathbf{X}) = 2$ 时，则会发生完整的 Householder 反射。当行列式非零时，提升状态空间的全局定向当且仅当 d_v 为奇数且 $\beta(\mathbf{X}) > 1$ 时翻转。

3.2 几何运算的统一

定理 3.1 揭示，随着 $\beta(\mathbf{X})$ 从0 变为2，快捷连接算子会在三种基本线性变换之间进行插值。

- **恒等映射 ($\beta(\mathbf{X}) \to 0$)：** 当 $\beta \to 0$ 时，特征值为 $1 - \beta \to 1$。$\mathbf{A}(\mathbf{X})$ 的所有特征值都变为 1，因此 $\mathbf{A}(\mathbf{X}) \to \mathbf{I}$。由于 β 还会调节注入项 $\beta\mathbf{k}\mathbf{v}^\top$，整个更新量会消失，这意味着 $\mathbf{X}_{l+1} \approx \mathbf{X}_l$。这种恒等行为对于在极深的深度网络中保持信号传播至关重要。

- 正交投影（$\beta(\mathbf{X}) \to 1$）：当 $\beta \to 1$ 时，特征值为 $1 - \beta \to 0$。算子 $\mathbf{A}(\mathbf{X})$ 变为 $\mathbf{I} - \mathbf{k}\mathbf{k}^\top$，即投影到超平面 $\mathbf{k}^\perp$ 上的正交投影算子（秩为 $d-1$）。输入状态 $\mathbf{X}$ 的每一列中与 $\mathbf{k}$ 平行的分量会在加入残差之前被显式移除。该算子变为奇异算子，且 $\det(\mathbf{A}) \to 0$。就完整模块而言，这一状态可解释为沿 $\mathbf{k}$ 替换：快捷连接移除 $\mathbf{k}$ 分量，而秩一写入沿 $\mathbf{k}$ 注入由 $\mathbf{v}^\top$ 指定的新分量。公式（2.6）对此作出了明确说明：在 $\beta_l = 1$ 时，$\mathbf{k}_l^\top\mathbf{X}_{l+1} = \mathbf{v}_l^\top$。

- **完全反射（$\beta(\mathbf{X}) \to 2$）：** 当 $\beta \to 2$ 时，特征值为 $1 - \beta \to -1$，且 $\mathbf{A}(\mathbf{X}) \to \mathbf{I} - 2\mathbf{k}\mathbf{k}^\top$，即标准的 Householder 反射算子。它将 $\mathbf{X}$ 的每一列关于 $\mathbf{k}^\perp$ 进行反射。与恒等变换情况（$\beta \to 0$）类似，这是一个正交且保持体积不变的快捷连接（所有奇异值均等于 1）；但在此情况下，$\det(\mathbf{A}) \to -1$，表示定向反转。随后，完整模块会加入同步秩一写入项 $\beta\mathbf{k}\mathbf{v}^\top$。

3.3 特殊情形：门控残差学习

深度 Delta 学习的一个关键性质是：当门控标量趋于极限时，其行为会发生变化。当门控消失（$\beta(\mathbf{X}) \to 0$）时，Delta 算子收敛到单位矩阵（$\mathbf{A}(\mathbf{X}) \to \mathbf{I}$），构造项也随之消失。因此，式（2.4）中的更新规则可简化为：

$$\mathbf{X}_{l+1} = \mathbf{X}_l.$$

这会恢复恒等映射，实际上使该层可以被完全跳过。这种行为与训练极深网络时通常所需的零初始化策略一致。相反，当 $\beta \approx 1$ 时，该层的功能相当于门控秩一矩阵残差网络，其中 β 充当学习得到的步长，用于控制更新幅度。这表明，DDL 通过引入与值注入相耦合的乘性几何调制，推广了残差学习

，同步进行。更一般地，式（2.5）是步长为 β 的秩-1增量更新；当 $\beta \approx 1$ 时，对应于投影坐标中的沿$\mathbf{k}$ 覆盖行为（式（2.6））。

3.4 对角特征矩阵情形

为了单独考察捷径左乘所引入的混合，我们对分支输出进行条件化，并将$\mathbf{A} = \mathbf{I} - \beta\mathbf{k}\mathbf{k}^\top$ 视为固定值。考虑输入状态为对角形式$\mathbf{X} \in \mathbb{R}^{d\times d}$的特殊情形，其中$\mathbf{X} = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$，不含跨通道耦合。应用$\mathbf{A}$后得到：

$$(\mathbf{A}\mathbf{X})_{ij} = (\mathbf{X} - \beta\mathbf{k}\mathbf{k}^\top\mathbf{X})_{ij} = s_i\delta_{ij} - \beta s_j k_i k_j,$$

其中$\mathbf{X} = \mathrm{diag}(s_1, \ldots, s_d)$，$\delta_{ij}$为克罗内克δ。具体而言，非对角元素($i \neq j$)变为 $-\beta s_j k_i k_j$，而对角元素($i = j$)变为 $s_i(1 - \beta k_i^2)$。因此，即使输入状态为对角形式，非零的 β 也会引入受控的特征耦合，其大小与$k_i k_j$成正比。此计算单独展示了捷径映射$\mathbf{X} \mapsto \mathbf{A}\mathbf{X}$所引入的耦合。完整的块还会注入一个秩一写入 $\beta\mathbf{k}\mathbf{v}^\top$，即向每个值列添加一个与$\mathbf{k}$对齐的分量。

如果 $\beta \to 1$，快捷连接会移除每一列沿$\mathbf{k}$方向的分量，将状态映射到$\mathbf{k}^\perp$，然后写入项会重新引入由$\mathbf{v}^\top$指定的新$\mathbf{k}$分量。当 $\beta \to 0$ 时，混合项和写入项都消失，因此对角结构在极限情况下得以保留。

3.5 向量隐藏状态动态

虽然 DDL 作用于矩阵值状态$\mathbf{X} \in \mathbb{R}^{d\times d_v}$，但它自然地将标准的基于向量的深度学习纳入其中，作为一种特定的极限情形。我们将其分为两种不同的机制：
标量值极限（$d_v = 1$）。当值维度降为 1 时，隐藏状态退化为标准特征向量$\mathbf{x} \in \mathbb{R}^d$。在此极限下，值更新$\mathbf{v}$ 变为标量$v \in \mathbb{R}$。增量更新规则式（2.4）可简化为：

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \beta_l \underbrace{(v_l - \mathbf{k}_l^\top\mathbf{x}_l)}_{\gamma_l}\mathbf{k}_l. \tag{3.1}$$

此处，几何变换简化为动态标量门控机制。项 γ_l充当数据依赖系数，将更新幅度与提议写入值 v_l 和当前投影$\mathbf{k}_l^\top\mathbf{x}_l$之间的差异关联起来。

独立特征极限。 或者，也可以将第 3.4 节中的对角情形视为嵌入矩阵对角线中的向量状态的一种表示。如对角分析所示，Delta 算子通过以下项引入特征耦合 $\beta k_i k_j$。为消除快捷连接引起的空间混合，一个充分条件是 $\mathbf{k}$ 与某个标准基向量 $\mathbf{e}_i$对齐。此时$\mathbf{k}\mathbf{k}^\top = \mathbf{e}_i\mathbf{e}_i^\top$，且 $\mathbf{A}$ 仅将 i-th 坐标缩放 $(1 - \beta)$，同时保持其他坐标不变。不过请注意，单次秩-1 Delta 更新只能沿着张成空间$\{\mathbf{k}\}$写入；若要在所有坐标上实现完全通用的逐元素更新，则需要多个方向。

3.6 残差学习的 Delta 规则

深度 Delta 学习这一术语体现了其与 Delta 规则之间的结构同源性。Delta 规则是一种基础的更新机制，近期因高效序列建模而广受关注，例如 DeltaNet（Schlag 等人，

2021；Yang 等人，2024年）。

标准残差连接 $\mathbf{X}_{l+1} = \mathbf{X}_l + \mathbf{F}(\mathbf{X}_l)$具有严格的加性归纳偏置。由 $\mathbf{F}$生成的信息只会被简单累积。这可能导致“残差累积”：由于网络缺乏显式机制来有选择地筛除隐藏状态中的内容，含噪或相互干扰的特征会在各层中持续存在。深度 Delta 学习通过将 Delta 规则结构引入深度维度来解决这一问题。根据秩一残差的定义，展开式（2.4）中的 Delta 残差更新：

$$\mathbf{X}_{l+1} = \mathbf{X}_l + \beta_l \mathbf{k}_l \Big(\underbrace{\mathbf{v}_l^\top}_{\text{Write target}} - \underbrace{\mathbf{k}_l^\top \mathbf{X}_l}_{\text{Current readout}} \Big),$$

这正是式（2.5）。在我们的左乘约定下，这是矩阵形式的 Delta（Widrow-Hoff）更新：行向量 $\mathbf{k}_l^\top \mathbf{X}_l$ 包含当前的投影系数（读出），再乘以 $\mathbf{k}_l$ 便得到秩一移除项 $\beta_l \mathbf{k}_l (\mathbf{k}_l^\top \mathbf{X}_l) = (\beta_l \mathbf{k}_l \mathbf{k}_l^\top) \mathbf{X}_l$。差值 $(\mathbf{v}_l^\top - \mathbf{k}_l^\top \mathbf{X}_l)$就是校正信号。

由于 $\mathbf{X}_l \in \mathbb{R}^{d \times d_v}$ 是一个矩阵，项 $\mathbf{k}_l^\top \mathbf{X}_l$ 在 $\mathbb{R}^{1 \times d_v}$ 中产生一个行向量，表示将每个值列投影到 k_l 上。该更新使擦除和注入操作都严格沿着由投影器 $\mathbf{k}_l$ 定义的几何方向进行，并由标量步长 β_l 调节。

当 $\beta(\mathbf{X}_l) \approx 1$时，该减法项充当正交投影，有效擦除输入状态 $\mathbf{X}_l$ 中与 $\mathbf{k}(\mathbf{X}_l)$平行的分量。当 $\beta(\mathbf{X}_l) \approx 2$时，该项会减去两倍的投影，从而导致符号反转。这为网络提供了一种灵活机制，可以逐层选择性地清理或重新定向特定的特征子空间，防止干扰不断累积。关于深度 Delta 学习、Delta 网络和 Householder乘积之间的关系，请参阅附录 A。

4 DDL Transformer

我们评估了深度 Delta 学习作为 Transformer 语言模型中标准加性残差连接的即插即用替代方案。在所有比较中，我们固定骨干架构（预归一化 RMSNorm、旋转位置嵌入多头注意力和 SwiGLU MLP），仅改变残差更新规则。

4.1 $d_v = 1$ Transformer 设置

在标量值机制（$d_v = 1$）中，矩阵状态退化为向量状态 $\mathbf{x} \in \mathbb{R}^d$，并且 Delta 更新简化为公式（3.1）：

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \beta_l \left(v_l - \mathbf{k}_l^\top \mathbf{x}_l \right) \mathbf{k}_l.$$

我们在每个 Transformer 模块中实例化这一更新。所有 Transformer 实验均使用预归一化骨干网络。令 $\mathbf{x}_l^{\text{ctx}} := \text{RMSNorm}(\mathbf{x}_l)$ 表示归一化后的子层输入。对于给定的子层（注意力或 MLP），其输出为 $\mathbf{F}(\mathbf{x}_l^{\text{ctx}})$，我们比较两种参数化策略：

- $\mathbf{k}$-映射：我们将骨干网络的输出视为几何反射方向$\mathbf{k}$的来源。我们$\tilde{\mathbf{k}}_l = \mathbf{F}(\mathbf{x}^{\text{ctx}}_l)$，并对$\mathbf{k}_l = \tilde{\mathbf{k}}_l / \|\tilde{\mathbf{k}}_l\|_2$ 进行归一化（在实现中使用标准的 ϵ_k 保护措施，以应对

$\sim\|\mathbf{k}_l\|_2$ 极小的情况）。标量更新值 v_l 通过对残差流进行单独的线性投影得到，$v_l = \sigma(\mathbf{w}_v^\top \mathbf{x}_l)$。这一表述将 Transformer 的输出解释为定义了在状态空间中写入的位置，而内容则由对输入进行简单的线性投影给出。

- v-映射：我们将骨干网络的输出视为内容值的来源 v_*。我们设定 $v_l = \sigma(\mathbf{w}_p^\top \mathbf{F}(\mathbf{x}_{\mathrm{ctx}l}))$。反射方向由作用于$\tilde{}$上下文的独立辅助分支生成，$\mathbf{k}_l = \phi_k(\mathbf{x}_{\mathrm{ctx}l})$，随后进行归一化。这种表述将 Transformer 的输出解释为定义了要写入什么，从而将高复杂度的内容生成与几何解耦；几何由复杂度较低的辅助映射决定。

在我们的主要评估中，我们采用**k**-映射变体。在所有实验中，$\mathbf{k}_l$ 的 L_2 归一化$\tilde{}$均会执行，并在 $\|\mathbf{k}_l\|_2$ 极小时采用标准 ϵ_k 保护措施。在这两种配置中，门控 β_l 由作用于$\mathbf{x}_l^{\mathrm{ctx}}$的轻量线性层或多层感知机计算得出。

适用于高精度的归一化。为提高低精度训练时的数值精度，我们通过 RMS 归一化和固定缩放因子 $k_{\mathrm{scale}} = 1/\sqrt{d}\tilde{}$ 实现 $\|\mathbf{k}_l\|_2 \approx 1$ 约束，而不显式构造 $\|\mathbf{k}_l\|_2$。我们的 PyTorch 实现通过融合 RMSNorm 和常数缩放因子来计算单位L_2 方向：

$$\widehat{\mathbf{k}} = \mathrm{RMSNorm}(\widetilde{\mathbf{k}}; \epsilon_k^2/d), \quad k_{\mathrm{scale}} = \frac{1}{\sqrt{d}}, \quad \mathbf{k} = \widehat{\mathbf{k}}\, k_{\mathrm{scale}}.$$

对于 $\epsilon_k = 0$，这是精确的 L_2 归一化。对于 $\epsilon_k > 0$，它在代数上等价于$\mathbf{k} = \tilde{}\mathbf{k}/\sqrt{\|\tilde{}\mathbf{k}\|_2^2 + \epsilon_k^2}$，因此当 $\|\tilde{}\mathbf{k}\|_2 \gg \epsilon_k$时，$\|\mathbf{k}\|_2 \approx 1$；这正是谱分析所假设的机制。该公式使中间量级保持在$\mathcal{O}(1)$，并允许在点积之外应用 k_{scale}（例如在 $\mathbf{k}^\top\mathbf{X}$ 中），从而提高 bf16/fp16 下的数值精度。

4.2 $d_v > 1$ 与状态扩展

在这种设定下，我们放宽了残差状态大小必须与骨干网络计算宽度一致的约束。相反，我们研究一种内存增强机制，其中残差状态被显式扩展为矩阵 $\mathbf{X}_l \in \mathbb{R}^{d\times d_v}$，其中 $d = D_{\mathrm{model}}$ 设为标准模型宽度，而 $d_v \geq 2$则作为内存扩展因子。

该公式将网络的内存容量与 Transformer 骨干网络的计算容量解耦：前者随 d_v线性增长，而后者保持固定宽度 d。DDL 更新遵循标准的矩阵值定律：

$$\mathbf{X}_{l+1} = \mathbf{X}_l + \beta_l \mathbf{k}_l(\mathbf{v}_l^\top - \mathbf{k}_l^\top \mathbf{X}_l).$$

为了将此扩展状态接入标准的 Transformer 子层（注意力和多层感知机），这些子层要求输入为 $\mathbb{R}^d$ 中的向量，因此我们采用压缩-处理-扩展协议：

在嵌入层，我们通过沿值通道重复词元嵌入来初始化扩展状态。具体而言，对于嵌入 $\mathbf{x}_{\mathrm{emb}} \in \mathbb{R}^d$，我们构造 $\mathbf{X}_0 = \mathbf{x}_{\mathrm{emb}}\mathbf{1}_{d_v}^\top \in \mathbb{R}^{d\times d_v}$（等价地，以展平形式表示时，我们将长度为 d 的嵌入重复为长度为 $d\cdot d_v$ 的向量）。

1. 压缩： 我们通过聚合记忆列来推导子层输入 $\mathbf{x}_l^{\mathrm{in}} \in \mathbb{R}^d$。在我们的实现中，该读出操作首先对扩展流应用短的因果深度卷积（在每个通道上独立进行，并沿词元维度计算），然后执行学习得到的加权池化：

$$\widetilde{\mathbf{X}}_l = \mathrm{ShortConv}(\mathbf{X}_l), \qquad \mathbf{x}_l^{\mathrm{in}} = \widetilde{\mathbf{X}}_l \mathbf{w}_p,$$

其中，$\mathbf{w}_p \in \mathbb{R}^{d_v}$ 是一个可学习读取向量。

2. **处理：** 标准前归一化子层会处理这一压缩表示：$\mathbf{h}_l = \mathbf{F}(\text{RMSNorm}(\mathbf{x}_l^{\text{in}})) \in \mathbb{R}^d$。

3. **扩展：** 骨干网络的输出$\mathbf{h}_l$用于生成秩一更新分量。与标量情形（$d_v = 1$）一致，我们采用两种映射策略：

- **k-映射：** 骨干网络的输出决定反射方向$\mathbf{k}_l = \mathbf{h}_l$。值向量$\mathbf{v}_l \in \mathbb{R}^{d_v}$ 通过对压缩输入进行单独的线性投影得到：$\mathbf{v}_l = \mathbf{W}_v \mathbf{x}_{\text{in}l}$，其中$\mathbf{W}_v \in \mathbb{R}^{d_v \times d}$。

- **v-映射**：骨干网络的输出决定值内容$\mathbf{v}_l = \mathbf{W}_v \mathbf{h}_l$，其中$\mathbf{W}_v \in \mathbb{R}^{d_v \times d}$。反射几何$\mathbf{k}_l$ 由辅助分支根据子层输入上下文得出，$\widetilde{\mathbf{k}}_l = \phi_k(\mathbf{x}_{\text{in}l})$。

该架构将隐藏状态视为短期记忆库，每个特征包含 d_v 个槽位。Delta 算子会同步这些槽位上的更新，在允许值内容沿记忆维度变化的同时，确保几何变换共享。我们使用 $d_v = 4$评估这一设置，它表示相较于基线 Transformer，残差记忆占用扩展了 $4\times$，而注意力机制的浮点运算次数并未增加。

5 实验

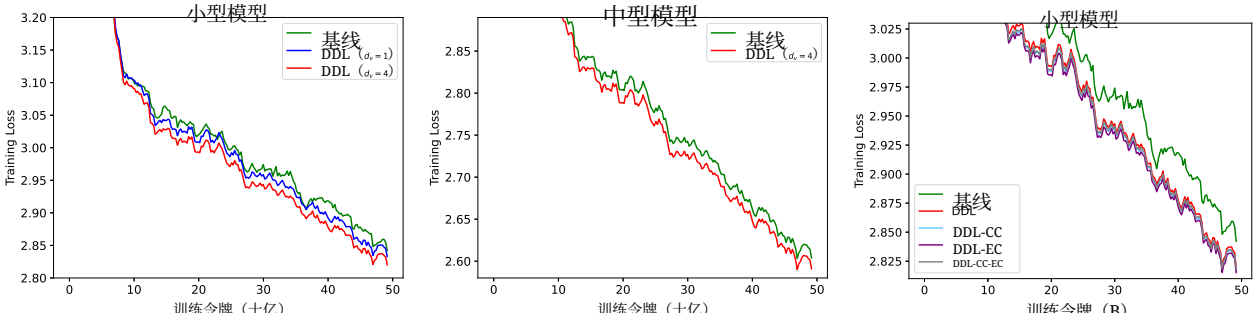
我们将提出的深度 Delta 学习（DDL）与 $d_v = 1$ 以及 $d_v = 4$ 基于 nanoGPT 代码库的基线进行比较（Karpathy，2022年）。对于采用 $d_v = 4$的 DDL 模型，我们沿序列长度维度使用逐通道卷积（Chollet，2017年），并使用 $d \cdot d_v$ 个通道。

表 1 小型模型在 FineWeb-Edu 100B 数据集上的评估结果（使用 lm-evaluation-harness 进行 1 样本评估）。每列中的最佳分数以粗体显示。缩写：WG = WinoGrande。

模型	ARC-C	ARC-E	Hellaswag	OpenBookQA	PIQA	SciQ	Social IQA	WG	Avg.
基线	29.01	55.85	37.59	30.20	65.94	80.60	37.87	51.38	48.56
DDL ($d_v = 1$)	**29.35**	57.49	38.08	31.80	64.85	78.50	37.77	52.01	48.73
DDL ($d_v = 4$)	27.90	58.16	38.26	30.80	**66.49**	80.30	**38.54**	50.83	48.91
DDL-CC	27.82	**58.92**	**38.44**	33.20	65.83	79.50	38.28	51.07	49.13
DDL-EC	28.75	57.37	38.41	**34.40**	64.47	82.00	38.38	52.01	**49.47**
DDL-CC-EC	28.33	57.87	38.24	32.20	64.09	**82.60**	38.43	**52.57**	49.29

表 2 中型模型在 FineWeb-Edu 100B 数据集上的评估结果（使用 lm-evaluation-harness 进行 1 样本评估）。每列中的最佳分数以粗体显示。缩写：WG = WinoGrande。

模型	ARC-C	ARC-E	Hellaswag	OpenBookQA	PIQA	SciQ	Social IQA	WG	Avg.
基线	33.62	**67.05**	47.42	33.20	70.24	87.30	40.28	52.57	53.96
DDL ($d_v = 1$)	**35.49**	65.70	46.94	34.20	**70.35**	88.90	**40.99**	54.93	54.69
DDL ($d_v = 4$)	33.02	66.16	**47.83**	**35.60**	69.86	**89.50**	**40.99**	**55.64**	**54.83**



(a) 小型 (124M) 训练集 **(b)** 中型 (353M) 训练集 **(c)** DDL 变体 (124M) 训练集

(d) 小型 (124M) 验证集 **(e)** 中型 (353M) 验证集 **(f)** DDL 变体 (124M) 验证集

图2 FineWeb-Edu 100B 上的损失曲线。上排显示训练损失；下排显示验证损失。各列分别代表小型（124M）、中型（353M）和小型 DDL 变体架构。

表 3 FineWeb-Edu 100B 数据集上，小型和中型模型在最后一步的验证损失和困惑度。每列中表现最好的损失和困惑度以粗体标出。

模型	小型		中型	
	验证损失	验证困惑度	验证损失	验证困惑度
基线	2.85426	17.3616	2.60532	13.5356
DDL ($d_v = 1$)	2.84817	17.2562	2.60388	13.5161
DDL ($d_v = 4$)	**2.83545**	**17.0381**	**2.59267**	**13.3654**

5.1 实验设置

我们在 FineWeb-Edu 100B 上进行训练（Lozhkov 等人，2024年），该数据集包含 1000 亿个训练令牌和 1 亿个验证令牌。我们在两个规模上进行实验：小型（124M 个参数）和中型（353M 个参数），采用带有旋转位置嵌入（Su 等人，2024年）嵌入和 SwiGLU 激活函数的 Llama 风格模型（Shazeer，2020）；我们还加入查询和键归一化，以稳定训练。

我们训练模型 100,000 步，全局批次大小为 480，序列长度为 1,024。此外，我们采用 μP 初始化和参数化（Yang 等人，2022年）。学习率设为 $1\mathrm{e}{-3}$，采用余弦学习率调度，并设置 2,000 个预热步数。我们使用 AdamW（Loshchilov 和 Hutter，2019），权重衰减为 0.1，$(\beta_1, \beta_2) = (0.9,\ 0.95)$，并将梯度裁剪设为 1.0。我们禁用偏置项，并将丢弃率设为 0.0。每项实验均使用 4 块 NVIDIA H200 GPU。其他超参数列于附录 C。

5.2 实验结果

训练和验证损失曲线见图 2a、2b，以及图 2d、2e；最终步的损失和困惑度列于表 3。结果表明，DDL $d_v = 4$ 优于基线模型；此外，DDL $d_v = 1$ 的训练损失和验证损失也低于基线。

此外，我们使用 `lm-evaluation-harness` 代码库（Gao 等人，2021）评估了 ARC（Yadav 等人，2019）、HellaSwag（Clark 等人，2019）、OpenBookQA（Mihaylov 等人，2018）、PIQA（Bisk 等人，2020）、SciQ（Welbl 等人，2017）、Social IQA（Sap 等人，2019）和 WinoGrande（Sakaguchi 等人，2021）上的 1 样本和 0 样本性能。1 样本结果见表 1 和表 2，0 样本结果则放在附录 D 中。总体而言，DDL 的表现优于基线，并且随着 d_v 增加，性能进一步提升。

5.3 消融研究

此外，我们还实现了 DDL 的一些变体 [1]，其中包括沿 $d_v = 4$维度进行通道卷积的模型（记为“DDL-CC”），以及在输入嵌入之后沿序列长度维度进行卷积的变体（分别记为“DDL-EC”和“DDL-CC-EC”）。训练和验证损失曲线见图2c 和2f。基准测试上的 1 样本和零样本评估结果见表1和5 （附录D）。结果表明，添加嵌入卷积以及沿 d_v 进行卷积，可能会进一步提升模型性能。

6 相关工作

我们的工作建立在深度学习的几个关键研究主题之上，包括门控和可逆架构、正交网络与酉网络，以及神经常微分方程。

门控与可逆架构。高速公路网络（Srivastava 等人，2015）将依赖数据的门控引入残差网络。随后，Chai 等人（2020）通过在门控中引入多层前馈网络，对高速公路网络进行了扩展。Menghani 等人（2025）、Pagliardini 等人（2024）和 Fang 等人（2023）进一步引入了从先前层延伸而来的密集连接，以促进信息流；Xiao 等人（2025）则采用额外的残差路径和动态密集连接来实现跨层信息传递。然而，它们的门控是在恒等路径与函数路径之间进行插值，而非修改变换本身。Mak 和 Flanigan（2025）将外积记忆存储引入 Transformer，用于信息的存储与检索。近期，Zhu 等人（2025）提出了超连接，通过引入深度与宽度连接，整合不同深度上的特征。随后，Xie 等人（2025）通过引入流形投影对此进行了改进，以保持恒等映射属性。可逆残差网络（i-ResNets）（Behrmann 等人，2019）通过约束 F 的 Lipschitz 常数来确保可逆性，这对于归一化流等应用很有用。当 $1-\beta \neq 0$ （等价地， $\beta \neq 1$）时，我们的 Delta 捷径算子可逆；在 $\beta = 2$ 时，它则成为正交对合（Householder 反射）。DDL 并不强制全局可逆，而是允许网络学习何时采用近似可逆的转换更有益，何时采用有意设置为奇异（投影式）的转换有助于实现受控遗忘。

[1] 这些变体的详细介绍见附录 B.3。

正交与酉网络。许多先前研究致力于将网络权重约束为正交（Zhang 等人，2021；Wang 等人，2025）或酉（Arjovsky 等人，2016；Jing 等人，2017；He 等人，2025），以提升梯度稳定性并保留几何结构。Fei 等人（2022）对视觉 Transformer 中的自注意力参数进行正交化，以保留特征空间的几何性质。Xie 等人（2025）提出了 mHC，利用 Sinkhorn–Knopp 操作重新缩放残差矩阵，使其具有正交性质。Householder 反射是参数化正交矩阵的一种经典方法。Yang 等人（2025）在位置编码中利用了这一性质，而 Dong 等人（2024）则将其应用于视觉 Transformer，以高效模拟奇异值分解。Arcas 等人（2025）将其融入低秩适配，以降低时间和空间复杂度。这些方法都将正交性作为严格约束。相比之下，我们的 Delta 残差网络通过门控 $\beta(\mathbf{X})$学习偏离恒等变换与正交性的程度，从而提供一种柔性、可自适应调整的约束，并可放宽为纯投影或反射。

神经常微分方程。神经常微分方程（ODE）（Chen 等，2018）用于对特征的连续演化进行建模。先前的工作已将常微分方程模型融入 Transformer（Tong 等，2025；Li 等，2022；Kan 等，2025；Jemley，2026），以促进时间序列建模（Chen 等，2023；Zhang 等，2025），并实现深度自适应 Transformer（Baier-Reinio 和 De Sterck，2020）。ODE-RNN（Rubanova 等，2019）将神经常微分方程与门控循环单元（Cho 等，2014）相结合，用于不规则时间序列建模。对于 Deep Delta 网络，标准残差网络公式 (1.1) 是常微分方程 $\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X})$ 的离散化，而我们的更新 $\mathbf{X}_{l+1} - \mathbf{X}_l = \beta(\mathbf{X}_l)\,\mathbf{k}(\mathbf{X}_l)\big(\mathbf{v}(\mathbf{X}_l)^\top - \mathbf{k}(\mathbf{X}_l)^\top \mathbf{X}_l\big)$ 可视为针对 $\dot{\mathbf{X}} = \mathbf{k}(\mathbf{X})(\mathbf{v}(\mathbf{X})^\top - \mathbf{k}(\mathbf{X})^\top \mathbf{X})$ 的前向欧拉步，其步长 $\beta(\mathbf{X})$ 是自适应的（依赖于状态），由此得到一族逐层动力学，这些动力学沿学习到的方向可以呈现收缩或符号翻转。

7 结论

我们提出了深度 Delta 学习，这是一种建立在自适应几何残差连接之上的模型架构。通过分析，我们证明了其核心组件 Delta 算子将恒等映射、投影和反射统一为一个连续可微模块。这种统一由一个简单的可学习标量门控控制，能够动态塑造层间转移算子的谱。通过使网络能够以数据依赖方式学习具有负特征值的变换，DDL 提供了一种在保留残差学习范式基础优势的同时提升表达能力的原则性方法。

参考文献

Alejandro Moreno Arcas、Albert Sanchis、Jorge Civera 和 Alfons Juan。Hoft：Householder 正交微调。*arXiv* 预印本 *arXiv:2505.16531*，2025年。

Martin Arjovsky、Amar Shah 和 Yoshua Bengio。酉演化循环神经网络。在国际机器学习大会，第1120–1128页。PMLR，2016年。

Aaron Baier-Reinio 和 Hans De Sterck。N-ode Transformer：一种利用神经常微分方程的深度自适应 Transformer 变体。*arXiv* 预印本 *arXiv:2010.11358*，2020年。

Cenk Baykal、Dylan Cutler、Nishanth Dikkala、Nikhil Ghosh、Rina Panigrahy 和 Xin Wang。用于高效 Transformer 的交替更新。神经信息处理系统进展，36：76718–76736，2023年。Jens Behrmann、Will Grathwohl、Ricky TQ Chen、David Duvenaud 和 Jörn-Henrik Jacobsen。可逆残差网络。载于国际机器学习大会，第573–582页。PMLR，2019年。Yonatan Bisk、Rowan Zellers、Jianfeng Gao、Yejin Choi 等。PIQA：自然语言中的物理常识推理。载于*AAAI*人工智能大会论文集，第34卷，第7432–7439页，2020年。Yekun Chai、Shuo Jin 和 Xinwen Hou。高速公路Transformer：通过自门控增强的自注意力网络。载于计算语言学协会第*58*届年会论文集，第6887–6900页，2020年。Ricky TQ Chen、Yulia Rubanova、Jesse Bettencourt 和 David K Duvenaud。神经常微分方程。神经信息处理系统进展，31，2018年。Yuqi Chen、Kan Ren、Yansen Wang、Yuchen Fang、Weiwei Sun 和 Dongsheng Li。Contiformer：用于不规则时间序列建模的连续时间Transformer。神经信息处理系统进展，36：47143–47175，2023年。Kyunghyun Cho、Bart van Merriënboer、Dzmitry Bahdanau 和 Yoshua Bengio。论神经机器翻译的特性：编码器–解码器方法。载于统计机器翻译语法、语义与结构第八届研讨会（*SSST-8*）论文集，第103–111页，2014年。François Chollet。Xception：采用深度可分离卷积的深度学习。载于*IEEE*计算机视觉与模式识别会议论文集，第1251–1258页，2017年。Christopher Clark、Kenton Lee、Ming-Wei Chang、Tom Kwiatkowski、Michael Collins 和 Kristina Toutanova。BoolQ：探索自然语言是非问题令人意外的难度。*arXiv* 预印本 *arXiv:1905.10044*，2019年。Wei Dong、Yuan Sun、Yiting Yang、Xing Zhang、Zhijun Lin、Qingsen Yan、Haokui Zhang、Peng Wang、Yang Yang 和 Hengtao Shen。通过 Householder变换高效适配预训练视觉Transformer。神经信息处理系统进展，37：102056–102077，2024年。Yanwen Fang、CAI Yuxi、Jintai Chen、Jingyu Zhao、Guangjian Tian 和 Guodong Li。通过层注意力进行跨层回溯检索。载于第十一届国际学习表征会议，2023年。Yanhong Fei、Yingjie Liu、Xian Wei 和 Mingsong Chen。O-ViT：正交视觉Transformer。*arXiv* 预印本 *arXiv:2201.12133*，2022年。Leo Gao、Jonathan Tow、Stella Biderman、Sid Black、Anthony DiPofi、Charles Foster、Laurence Golding、Jeffrey Hsu、Kyle McDonell、Niklas Muennighoff 等。少样本语言模型评测框架。*Zenodo*，2021。

Riccardo Grazzi、Julien Siems、Arber Zela、Jörg KH Franke、Frank Hutter 和 Massimiliano Pontil。通过负特征值解锁线性循环神经网络中的状态跟踪。载于第十三届国际学习表征会议，2025年。

何恺明、张祥雨、任少卿和孙剑。用于图像识别的深度残差学习。载于*IEEE*计算机视觉与模式识别会议论文集，第770–778页，2016年。

Yi He、Yiming Yang、Xiaoyuan Cheng、Hai Wang、Xiao Xue、Boli Chen 和 Yukun Hu。混沌遇上注意力：用于大规模动力学预测的 Transformer。*arXiv* 预印本 *arXiv:2504.20858*，2025年。

Alston S Householder。非对称矩阵的酉三角化。*ACM*学报（*JACM*），5(4):339–342，1958年。

Peter Jemley。《采用学习型控制动力学的连续深度变换器》。*arXiv* 预印本 *arXiv:2601.10007*，2026年。

Li Jing、Yichen Shen、Tena Dubcek、John Peurifoy、Scott Skirlo、Yann LeCun、Max Tegmark 和 Marin Soljačić。可调高效酉神经网络（eunn）及其在循环神经网络中的应用。载于国际机器学习大会，第1733–1741页。PMLR，2017年。

Kelvin Kan、Xingjian Li 和 Stanley Osher。最优传输变换器：一种采用最优传输正则化的连续时间变换器架构。*arXiv* 预印本 *arXiv:2501.18793*，2025年。

Andrej Karpathy。NanoGPT。 https://github.com/karpathy/nanoGPT，2022年。

Bei Li、Quan Du、Tao Zhou、Yi Jing、Shuhan Zhou、Xin Zeng、Tong Xiao、JingBo Zhu、Xuebo Liu 和 Min Zhang。常微分方程变换器：一种受常微分方程启发的序列生成模型。载于 Smaranda Muresan、Preslav Nakov 和 Aline Villavicencio 编，《style id='1'>计算语言学协会第60届年会论文集（第1卷：长篇论文），第8335–8351页，都柏林，爱尔兰，2022年5月。计算语言学协会。doi：10.18653/v1/2022.acl-long.571。URL https://aclanthology.org/2022.acl-long.571/。

Ilya Loshchilov 和 Frank Hutter。解耦权重衰减正则化。载于第七届国际学习表征会议（*2019*年国际学习表征会议），新奥尔良，路易斯安那州，美国，*2019*年*5*月*6*日至*9*日，2019年。

Anton Lozhkov、Loubna Ben Allal、Leandro von Werra 和 Thomas Wolf。FineWeb-Edu：最优质的教育内容合集，2024年。

Brian Mak 和 Jeffrey Flanigan。残差矩阵 Transformer：扩展残差流的规模。载于第四十二届国际机器学习大会，2025年。

Gaurav Menghani、Ravi Kumar 和 Sanjiv Kumar。Laurel：学习增强残差层。载于第四十二届国际机器学习大会，2025年。

Todor Mihaylov、Peter Clark、Tushar Khot 和 Ashish Sabharwal。盔甲能导电吗？一个用于开放书籍问答的新数据集。*arXiv*预印本 *arXiv:1809.02789*，2018年。

Matteo Pagliardini、Amirkeivan Mohtashami、Francois Fleuret 和 Martin Jaggi。Denseformer：通过深度加权平均增强 Transformer 中的信息流。神经信息处理系统进展，37：136479–136508，2024年。Yulia Rubanova、Ricky TQ Chen 和 David K Duvenaud。用于不规则采样时间序列的潜在常微分方程。神经信息处理系统进展，32，2019年。Keisuke Sakaguchi、Ronan Le Bras、Chandra Bhagavatula 和 Yejin Choi。Winogrande：大规模对抗性 Winograd模式挑战。《*ACM*通讯》，64(9)：99–106，2021年。Maarten Sap、Hannah Rashkin、Derek Chen、Ronan LeBras 和 Yejin Choi。Socialiqa：关于社会互动的常识推理。*arXiv*预印本 *arXiv:1904.09728*，2019年。Imanol Schlag、Kazuki Irie 和 Jürgen Schmidhuber。线性 Transformer 实际上是快速权重编程器。载于国际机器学习大会，第9355–9366页。PMLR，2021年。Noam Shazeer。GLU变体改进了 Transformer。*arXiv*预印本 *arXiv:2002.05202*，2020年。Rupesh Kumar Srivastava、Klaus Greff 和 Jürgen Schmidhuber。高速公路网络。*arXiv*预印本 *arXiv:1505.00387*，2015年。

Jianlin Su、Murtadha Ahmed、Yu Lu、Shengfeng Pan、Wen Bo 和 Yunfeng Liu。Roformer：采用旋转位置嵌入的增强型 Transformer。《神经计算》，568：127063，2024年。

Anh Tong、Thanh Nguyen-Tang、Dongeun Lee、Duc Nguyen、Toan Tran、David Leo Wright Hall、Cheongwoong Kang 和 Jaesik Choi。神经ODE Transformer：分析内部动力学与自适应微调。发表于第十三届国际学习表征会议，2025年。

Xiangming Wang、Haijin Zeng、Jiaoyang Chen、Sheng Liu、Yongyong Chen 和 Guoqing Chao。OTLRM：基于正交学习的多维逆问题低秩度量。发表于*AAAI*人工智能大会论文集，第39卷，第21278–21286页，2025年。

Johannes Welbl、Nelson F Liu 和 Matt Gardner。《众包科学选择题》。*arXiv* 预印本 *arXiv:1707.06209*，2017年。Da Xiao、Qingye Meng、Shengping Li 和 Xingyuan Yuan。《Muddformer：通过多路动态密集连接打破 Transformer 中的残差瓶颈》。载于第四十二届国际机器学习大会，2025年。Zhenda Xie、Yixuan Wei、Huanqi Cao、Chenggang Zhao、Chengqi Deng、Jiashi Li、Damai Dai、Huazuo Gao、Jiang Chang、Liang Zhao 等。《mHC：流形约束超连接》。*arXiv* 预印本 *arXiv:2512.24880*，2025年。Vikas Yadav、Steven Bethard 和 Mihai Surdeanu。《快速且（不那么）粗糙：为多跳问答无监督选择论证句》。*arXiv* 预印本 *arXiv:1911.07176*，2019年。

Greg Yang、Edward J Hu、Igor Babuschkin、Szymon Sidor、Xiaodong Liu、David Farhi、Nick Ryder、Jakub Pachocki、Weizhu Chen 和 Jianfeng Gao。《张量程序 V：通过零样本超参数迁移调整大型神经网络》。*arXiv* 预印本 *arXiv:2203.03466*，2022年。

Songlin Yang、Bailin Wang、Yu Zhang、Yikang Shen 和 Yoon Kim。利用 Delta 规则沿序列长度并行化线性 Transformer。载于第三十八届神经信息处理系统年会，2024年。

Songlin Yang、Yikang Shen、Kaiyue Wen、Shawn Tan、Mayank Mishra、Liliang Ren、Rameswar Panda 和 Yoon Kim。路径注意力：通过累积 Householder 变换实现位置编码。*arXiv* 预印本 *arXiv:2505.16381*，2025年。

Aston Zhang、Alvin Chan、Yi Tay、Jie Fu、Shuohang Wang、Shuai Zhang、Huajie S hao、Shuochao Yao 和 Roy Ka-Wei Lee。关于 Transformer 的正交约束。载于 Chengqing Zong、Fei Xia、Wenjie Li 和 Roberto Navigli 编，第*59*届计算语言学协会年会论文集和第*11*届自然语言处理国际联合会议论文集（第*2*卷：短文），第 375–382 页，线上，2021年8月。计算语言学协会。doi: 10.18653/v1/2021.acl-short.48。URL https://aclanthology.org/2021.acl-short.48/。

Yudong Zhang、Xu Wang、Xuan Yu、Zhengyang Zhou、Xing Xu、Lei Bai 和 Yang Wang。Diffode：用于不规则时间序列分析的可微隐藏状态神经常微分方程。载于*2025*年*IEEE*第*41*届国际数据工程会议*(ICDE)*，第 1–14 页。IEEE，2025年。

Defa Zhu、Hongzhi Huang、Zihao Huang、Yutao Zeng、Yunyao Mao、Banggu Wu、Qiyang Min 和 Xun Zhou。超连接。载于第十三届国际学习表征会议，2025年。

附录

与 DeltaNets 和 Householder乘积的关系19B 实现与参数化细节19B.1 反射方向的参数化 $\mathbf{k}(\mathbf{X})$ .19B.2 门控的参数化 $\beta(\mathbf{X})$和值$\mathbf{v}(\mathbf{X})$ .20B.3 扩展状态 Transformer 实现细节.20**C 超参数设置21D 补充结果22**

与 DeltaNets 和 Householder 乘积的关系

我们的工作与 DeltaNet 架构在理论上存在联系（Schlag 等人，2021），该架构用 Delta 规则形式的记忆更新取代了线性 Transformer 的加法累积。

我们证明，深度 Delta 学习是 DeltaNet 递推在深度方向上的同构形式。在 DeltaNet 中，隐藏状态（记忆）$\mathbf{S}_t$ 随时间 t演化。为使记号与我们的深度方向表述统一，我们使用左乘语义表示 DeltaNet 更新，其中记忆状态为$\mathbf{S}_t \in \mathbb{R}^{d_k \times d_v}$:

$$\mathbf{S}_t = (\mathbf{I} - \beta_t \mathbf{k}_t \mathbf{k}_t^\top)\mathbf{S}_{t-1} + \beta_t \mathbf{k}_t \mathbf{v}_t^\top. \tag{A.1}$$

这里，算子作用于键维度 d_k，这与 DDL 中的特征维度 d 相对应。将其与我们的 Deep Delta 层更新式（2.4）进行比较，该更新沿深度方向 l：式（A.1）在代数上等价于更常见的右乘 Delta 规则更新 $\mathbf{M}_t = \mathbf{M}_{t-1} + \beta_t$（$\mathbf{v}_t - \mathbf{M}_{t-1}\mathbf{k}_t$）$\mathbf{k}_t^\top$，只需令 $\mathbf{S}_t = \mathbf{M}_t^\top$（即对记忆矩阵采用转置约定）。

$$\mathbf{X}_{l+1} = (\mathbf{I} - \beta_l \mathbf{k}_l \mathbf{k}_l^\top)\mathbf{X}_l + \beta_l \mathbf{k}_l \mathbf{v}_l^\top, \tag{A.2}$$

其中 $\mathbf{v}_l$ 是值分支的向量输出。

这揭示了直接的结构对应关系：

- DeltaNet 中的记忆状态 $\mathbf{S}_t$（维度为 $d_k \times d_v$）对应于 DDL 中的特征激活 $\mathbf{X}_l$（维度为 $d \times d_v$）。

- 两种架构都采用相同的秩一捷径算子 $\mathbf{I} - \beta\,\mathbf{k}\mathbf{k}^\top$：当 $\|\mathbf{k}\|_2 = 1$ 时，它是正交投影算子；当 $\beta = 1$ 时，则是 Householder 反射。DeltaNet 在时间步 t上应用该算子，而 DDL 则沿网络深度 l应用。

- 我们修改后的残差更新 $\beta_l \mathbf{k}_l \mathbf{v}_l^\top$与 DeltaNet 的写入操作完全一致。通过将 β_l 纳入构造项，我们将 β_l 解释为深度方向常微分方程的逐层步长。这确保擦除分量和注入分量同步调节，从而使更新能够对状态$\mathbf{X}$进行连贯的几何变换。

因此，可以将 DDL 视为将 Delta 规则应用于逐层特征演化，使网络能够在特征从浅层向深层传播时将其遗忘或重写。

B 实现与参数化细节

Deep Delta Learning（DDL）框架依赖于对反射方向$\mathbf{k}(\mathbf{X})$、标量门控 $\beta(\mathbf{X})$和残差值$\mathbf{v}(\mathbf{X})$的高效估计。虽然理论结果不受用于近似这些函数的具体拓扑结构影响，但我们概述了生成器函数的两种主要架构实例：基于多层感知机的参数化和基于注意力的参数化。

设隐藏状态为$\mathbf{X} \in \mathbb{R}^{d \times d_v}$。我们将反射向量的生成器分支表示为一个函数 $\phi_k : \mathbb{R}^{d \times d_v} \to \mathbb{R}^d$。

B.1 反射方向$(\mathbf{X})$的参数化

Delta 算子的几何方向由 $\mathbf{k}$ 决定。我们为 ϕ_k提出了两种截然不同的机制，从而针对特征交互引入不同的归纳偏置。

选项 1：多层感知机参数化。

对于优先考虑以较低计算开销实现全局特征混合的架构，我们使用作用于状态矩阵聚合统计量的多层感知机（MLP）对 $\mathbf{k}$ 进行参数化。

$$\widetilde{\mathbf{k}}_{\mathrm{MLP}} = \mathrm{MLP}(\mathrm{Pool}(\mathbf{X}))\,, \quad \mathbf{k}_{\mathrm{MLP}} = \frac{\widetilde{\mathbf{k}}_{\mathrm{MLP}}}{\|\widetilde{\mathbf{k}}_{\mathrm{MLP}}\|_2}$$

此处，池化$(\cdot)$ 是指任何能够生成$\mathbf{X}$的定长向量表示的聚合操作，例如按列求平均（$\mathbb{R}^{d\times d_v} \to \mathbb{R}^{d}$）或展平（$\mathbb{R}^{d\times d_v} \to \mathbb{R}^{d\cdot d_v}$），然后通过一个输出 $\mathbb{R}^{d}$的多层感知机。我们施加 L_2 归一化，以满足定理 3.1中的谱假设。在实现中，˜仅当 ϵ_k 极小时才会对除法设置小型 $\|\mathbf{k}_{\mathrm{MLP}}\|_2$ 保护；分析假设$\|\mathbf{k}\|_2 = 1$。

选项 2：基于注意力的参数化。 为了捕捉值维度内部更细粒度的依赖关系，我们可以采用注意力机制。

B.2 门控 $\beta(\mathbf{X})$ 和值 $\mathbf{v}(\mathbf{X})$的参数化

门控分支。 标量门控 β 的输出必须有界于 $[0, 2]$。在 `model/DDL-gpt-mha-rope*.py`中，β 根据骨干网络所用的相同预归一化上下文，按词元计算：$\mathbf{c} = \mathrm{RMSNorm}(\mathbf{x}^{\mathrm{in}}) \in \mathbb{R}^{d}$ （其中 $\mathbf{x}^{\mathrm{in}}$ 是上述扩展状态的压缩读出）。我们让这一估计器保持轻量且可配置（`ddl_beta_single_linear`， `ddl_beta_hidden_size`）：

$$\beta(\mathbf{c}) = 2 \cdot \sigma(\mathrm{Linear}(\mathbf{c})) \quad \text{or} \quad \beta(\mathbf{c}) = 2 \cdot \sigma\big(\mathrm{Linear}\big(\tanh(\mathrm{Linear}(\mathbf{c}))\big)\big)\,.$$

为确保稳定性，我们使用单精度浮点数计算逻辑值，并将输出偏置初始化为所需的起始值 $\beta_0 \in [0, 2]$，即通过 logit($\beta_0/2$) 得到（代码中会进行截断； `ddl_beta_init`），从而确保恒等变换、投影和反射之间平滑插值。

值分支。残差值向量 $\mathbf{v} \in \mathbb{R}^{d_v}$ 表示内容更新。该分支ϕ_v ： $\mathbb{R}^{d\times d_v} \to \mathbb{R}^{d_v}$，可提供灵活的设计选择。在我们的实验中，我们采用与主干网络相同的架构（例如，如果在 Transformer 中应用 DDL，ϕ_v 则采用前馈网络或多头注意力模块的结构），以确保容量对齐。

B.3 扩展状态 Transformer 实现细节

我们针对扩展状态 Transformer 变体的 PyTorch 实现（$d_v > 1$）位于`model/DDL-gpt-mha-rope*.py`。为便于说明，我们先概述通用的张量布局，然后重点介绍代码库变体之间的关键差异。

代码库变体。

- `model/DDL-gpt-mha-rope.py`：基于 GPT（多头注意力 + 旋转位置嵌入）的基线扩展状态 DDL：通过沿值通道轴重复词元嵌入来初始化状态，并使用词元轴短因果卷积和学习得到的读出向量压缩扩展残差。

- `model/DDL-gpt-mha-rope-EC.py`：与基线相同，但使用可学习的因果深度卷积短卷积（`input_embed_shortconv_kernel_size`，以恒等变换初始化）扩展词元嵌入。

- `model/DDL-gpt-mha-rope-CC.py`：与基线相同，但将残差压缩短卷积切换为沿值通道轴进行运算 d_v 并消耗该轴（卷积返回长度 1； `ddl_state_shortconv_kernel_size` $= d_v$）。

- model/DDL-gpt-mha-rope-CC-EC.py：与 model/DDL-gpt-mha-rope-CC.py相同，但使用可学习的因果深度卷积短卷积扩展词元嵌入（input_embed_shortconv_-kernel_size，采用恒等初始化）。

状态布局与初始化。 对于一批序列，我们将扩展残差表示为形状为 $(B,\ T,\ d,\ d_v)$ 的四阶张量，其中 d 是模型宽度， d_v 是值通道数。

输入扩展。 在基线和 CC 模块中，我们通过沿值轴重复嵌入来初始化 $\mathbf{X}_0 = \mathbf{x}_{\text{emb}}\mathbf{1}_{d_v}^{\top}$ （实现方式为 x_emb.unsqueeze(-1).repeat(..., d_v)）。而在 EC 模块中，我们改为沿词元维度使用因果深度卷积短卷积，将 $(B,\ T,\ d) \to (B,\ T,\ d\cdot d_v)$，然后重塑为 $(B,\ T,\ d,\ d_v)$；该卷积采用恒等初始化，因此初始行为与简单重复相同。

沿 T (基线，DDL-EC) 进行 ShortConv 压缩。在 model/DDL-gpt-mha-rope.py和 model/DDL-gpt-mha-rope-EC.py中， ResidualShortConvCompressor 会先展平最后两个维度，将扩展残差视为 $(B,\ T,\ d\cdot d_v)$ 个通道，沿词元维度应用短因果深度卷积 Conv1d（卷积核大小为 ddl_state_shortconv_kernel_size），再重塑回 $(B,\ T,\ d,\ d_v)$，然后使用学习得到的读出向量 $\mathbf{w}_p \in \mathbb{R}^{d_v}$对值通道进行池化：

$$\widetilde{\mathbf{X}}_{l,t,i,j} = \sum_{s=0}^{k-1} c_{i,j,s}\,\mathbf{X}_{l,t-s,i,j}, \qquad \mathbf{x}^{\text{in}}_{l,t,i} = \sum_{j=1}^{d_v} w_{p,j}\,\widetilde{\mathbf{X}}_{l,t,i,j}.$$

默认情况下，我们将读取向量初始化为均匀平均值（ddl_state_read_init $= 1/d_v$ 未设置时）。

沿 d_v 进行 ShortConv 压缩（DDL-CC、DDL-CC-EC）。在 model/DDL-gpt-mha-rope-CC.py 和 model/DDL-gpt-mha-rope-CC-EC.py中，我们改为沿值通道轴应用深度可分离因果 Conv1d，并一次性消耗 d_v （该卷积返回长度为 1）。具体而言，我们将状态重塑为（$B\cdot T,\ d,\ d_v$），并将 d_v 视为卷积的“序列”长度，使用逐特征卷积核$c_{i,\ j}$，其卷积核大小为 $k = d_v$：

$$\mathbf{x}^{\text{in}}_{l,t,i} = \sum_{j=1}^{d_v} c_{i,j}\,\mathbf{X}_{l,t,i,j}.$$

这会将局部性先验从时间局部混合改为值通道局部混合。请注意，在 CC 中， ddl_state_shortconv_kernel_size 指的是沿 d_v 的卷积核大小（而不是沿 T的大小），并且必须等于 d_v，这样卷积才能返回长度为 1。

C 超参数设置

这里我们只是 此处列出了小型和中型模型的超参数，见表4 .

表 4 小型和中型模型的架构超参数。

模型	#参数	#层	#头	头维度	隐藏层大小
小型模型	124M	12	6	128	768
中型模型	353M	24	8	128	1024

D 补充结果

我们仅在表5和6中分别展示了小型和中型模型的零样本评估结果。这些结果也表明，DDL 在下游任务上的表现更佳。

表5 小型模型在 FineWeb-Edu 100B 数据集上的评估结果（使用 lm-evaluation-harness 进行 0 样本评估）。每列中的最佳分数以粗体标出。缩写：WG = WinoGrande。

模型	ARC-C	ARC-E	Hellaswag	OpenBookQA	PIQA	SciQ	Social IQA	WG	Avg.
基线	28.33	52.44	37.60	33.00	65.94	71.20	37.46	52.41	47.30
DDL ($d_v = 1$)	26.96	52.40	37.91	32.20	64.91	72.50	38.08	**53.59**	47.32
DDL ($d_v = 4$)	27.30	**52.95**	38.40	**33.80**	65.40	**73.70**	38.64	50.12	47.54
DDL-CC	27.05	51.47	38.72	32.20	**66.16**	71.80	38.08	51.07	47.07
DDL-EC	**28.50**	51.89	**38.82**	33.20	65.29	73.00	**39.05**	52.88	**47.83**
DDL-CC-EC	27.90	51.26	38.42	31.40	64.85	73.60	37.77	52.25	47.18

表 6 中型模型在 FineWeb-Edu 100B 数据集上的评估结果（使用 lm-evaluation-harness 进行 0 样本评估）。每列中的最佳分数以粗体标出。缩写：WG = WinoGrande。

模型	ARC-C	ARC-E	Hellaswag	OpenBookQA	PIQA	SciQ	Social IQA	WG	Avg.
基线	31.74	**59.85**	47.91	34.00	69.21	78.10	**40.69**	53.83	51.92
DDL ($d_v = 1$)	32.07	59.39	47.62	34.40	**70.08**	77.30	39.61	**55.01**	51.94
DDL ($d_v = 4$)	**32.08**	58.38	**48.08**	**35.80**	69.42	**79.90**	39.92	54.14	**52.22**