

状态空间模型作为基础模型： 控制理论概述

Carmen Amo Alonso*, Jerome Sieber*, and Melanie N. Zeilinger

Abstract—近年来，人们越来越关注将线性状态空间模型 (SSM) 集成到基础模型的深度神经网络架构中。Mamba 最近的成功就是一个例子，它在语言任务中显示出比最先进的 Transformer 架构更好的性能。基础模型，如 GPT-4，旨在将顺序数据编码到潜在空间中，以便学习数据的压缩表示。控制理论家也追求同样的目标，他们使用 SSM 来有效地对动态系统进行建模。因此，SSM 可以自然地与深度序列建模联系起来，从而为在相应研究领域之间创造协同效应提供了机会。本文旨在为控制理论家介绍基于 SSM 的架构，并总结最新的研究进展。它对最成功的 SSM 提案进行了系统回顾，并从控制理论的角度强调了它们的主要特征。此外，我们还对这些模型进行了比较分析，评估了它们在标准化基准上的性能，该基准旨在评估模型在学习长序列方面的效率。

Index Terms—Machine learning, Linear systems, Time-varying systems.

I. 介绍

最近，基础模型已成为人工智能领域的核心。这些模型是大规模学习模型，最初在广泛的数据集上进行预训练，然后针对特定任务进行微调。术语基础模型强调了这些模型在各种模态中学习和有效泛化的能力，包括语言、音频、图像、视频、基因组学等。从本质上讲，基础模型的主要架构是 Transformer [?]。该架构基于注意力机制，可以有效地处理信息并对复杂数据中的全局依赖关系进行建模；但它有两个主要局限性。一个是计算复杂性：每次生成输出时，它都需要将完整的序列输入到模型中，这导致时间范围窗口的可扩展性较差¹，因此在长上下文任务 [?] 中的性能较差。另一个限制是可解释性：尽管它的数学表示很简单，但目前无法解释或理解 Transformer [?] 所做的输出选择。为了解决 Transformer 的可扩展性挑战，各种架构变体仍然利用了注意力机制的优点。此类变体的例子有 Longformer [?]，BigBird [?]，the Reformer [?]，the Performer [?]，以及利用 Axial Attention [?] 的方法。然而，尽管在这些方面进行了广泛的研究，但所提出的解决方案往往会降低架构的固有优点，或者在实践中表现不佳 [?]。

最近一个有前途的研究途径建议用基于状态空间模型 (SSM) 的不同表示完全取代注意力机制。SSM 表示的优点在于其循环性质，其中只有最新的输入必须传递给模型，因为状态能够捕获有关过去输入的信息。此外，由于它们的数学结构，它们适合于计算高效的训练和推理——与它们的前辈递归神经网络 (RNN) [?]。这个基于 SSM 的新架构系列已被证明在长上下文任务中击败了 Transformers，例如 Long Range Arena (LRA) 基准测试 [?]，而最近的提案 (如 Mamba) [?] 在长上下

文任务中表现出优于最先进的 Transformer 的性能和计算效率。这些结果凸显了 SSM 在克服变压器的许多电流限制方面的潜力。尽管 SSM 在作为基础模型方面显示出巨大的前景，但大多数关于 SSM 的现有文献都集中在提供高性能架构和高效实现上。尽管与控制理论 (特别是线性系统理论) 有明确的联系，但迄今为止还缺乏对这些模型的原则性理解，并且大多数设计选择都是从经验性能而不是系统理论角度出发的。利用现有的系统理论结果和分析来补充当前的实现并增强可解释性、设计和性能方面具有巨大的潜力。

为了实现这一目标，本文的目的是从控制理论的角度对最先进的 SSM 进行概述。在第 ?? 节中，我们概述了 SSM 中的基本组件和注意事项。在第 ?? 节中，我们回顾了迄今为止最相关的 SSM 提案。由于这些模型主要是出于其处理长上下文的能力，因此我们在 Section ?? 中首次对 LRA 基准进行了性能比较。最后，我们以结论性评论和开放性研究问题结束 ??，这些问题可以帮助推进 SSMs，并在基础模型和系统与控制理论领域进行交叉授粉。

II. 状态空间模型

我们首先提出了一个通用语言建模任务来定义基础模型的学习目标。然后，我们概述了指导文献中介绍的 SSM 的状态空间模型架构、数学结构和计算考虑因素。

A. 学习设置

基础模型，例如语言建模中使用的模型，可以看作是输入和输出信号之间的映射，即

$$y(k) = f(u(k), \dots, u(k-T); \theta), \quad (1)$$

，在每次 k 时，输出 $y(k)$ 是在评估长度为 $k-T$ 的输入信号 (即 $u(k), \dots, u(k-T)$) 和一组参数 θ 后产生的。 θ 参数与任务相关，可以相应地进行微调。由于一般 $f(\cdot; \theta)$ 的搜索空间太宽，因此可以使用不同的 $f(\cdot; \theta)$ 参数化来使问题易于处理。例如，模型 $f(\cdot; \theta)$ 可以由多个堆叠模型组成，例如 Transformer 或最近的 SSM。 $f(\cdot; \theta)$ 的架构选择是决定模型能否从数据中有效学习结构的基本因素。

用作大型语言模型的基础模型的目标是学习语言中存在的结构的压缩表示，以便执行机器翻译或人类级对话 (例如 ChatGPT) 等任务。为了学习这种表示，参数化模型 $f(\cdot; \theta)$ 用输入-输出对 $(u(k), y(k)) \forall k$ 表示，其中 θ 表示参数。然后迭代更新 θ 参数以最小化损失函数 $\mathcal{L}(\cdot)$ ，即迭代求解以下优化问题

$$\min_{\theta} \mathcal{L}(y - f(u; \theta)). \quad (2)$$

* These authors contributed equally; ordered alphabetically.

All authors are with the Institute for Dynamic Systems and Control, ETH Zurich, 8092 Zurich, Switzerland {camoalonso, jsieber, mzeilinger}@ethz.ch. This work was partially supported by the ETH AI Center.

¹在 Transformer 的文献中称为输入长度。

对于语言模型，输入 u ² 句子标记化，输出 y 是相同输入的移位版本，即自动回归设置。

B. 参数

让我们考虑以下具有动力学的连续时间线性系统

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (3a)$$

$$y(t) = Cx(t) + Du(t), \quad (3b)$$

其中 $x \in \mathbb{C}^p$ 表示复值状态， $u, y \in \mathbb{R}^q$ 分别表示输入和输出， t 表示连续时间索引。我们注意到，输入系统的输入表示为 u 不是控制输入；它被视为激发系统 (??) 的外生输入。这种符号的选择是为了保持与相应文献的一致性。 A, B, C, D 是具有适当维度的复值矩阵，并且在表示 (??) 中，这些矩阵被假定为时不变的。在考虑其时变版本时，将附加一个时间子指数，即 A_t, B_t, C_t, D_t 。

在 SSM 文献中，系统 (??) 用作基础模型中的黑盒表示。在这里，外生输入 $u(t)$ 表示在给定时间馈入模型的信号或输入标记 t 。状态 $x(t)$ 表示隐藏状态，该状态存储了截至时间 t 的当前和先前输入的相关信息， $y(t)$ 是模型在时间 t 的输出。在学习设置中，矩阵 A, B, C, D 是参数，通常通过随机梯度下降来学习。由于计算效率和初始化是该框架中的基本方面，因此通常假设动态矩阵 A 具有特定的结构。因此，SSM 通常被称为结构化 SSM。

Assumption 2.1: 动力学 (??) 中的动态矩阵具有对角线结构，即 $A = \text{diag}(\lambda_1, \dots, \lambda_p)$ 与 $\lambda_i \in \mathbb{C} \forall i$ 。

虽然最初的建议 [?], [?] 略微偏离假设 ??，但大多数结构化 SSM 的文献都假设了一个对角线 A 矩阵。具体选择将在第 ?? 节中讨论。

C. 离散化

为了实现 SSM，使用了系统 (??) 的离散时间版本。因此，在离散时间内实现系统 (??) 是

$$x(k+1) = \bar{A}x(k) + \bar{B}u(k), \quad (4a)$$

$$y(k) = \bar{C}x(k) + \bar{D}u(k), \quad (4b)$$

其中 $\bar{A}, \bar{B}, \bar{C}, \bar{D}$ 是用时间步长 $\Delta \in \mathbb{R}$ 离散化的离散时间动态矩阵，可能使用复值分量， k 表示离散时间指数。SSM 文献中所提出的模型所选择的离散化方案的选择差异很大，概述见第 ?? 节。

我们注意到，也可以直接从离散时间模型开始，如方程 (??) 中，而忽略了它的连续时间表示 (??)。然而，在大多数 SSM 文献中，动力学的连续时间视图是首选，以便更好地激励动力学矩阵 [?] 初始化的选择。

D. 结构和初始化

由于动力学 (??) 是通过梯度下降来学习的，因此发现参数的初始化至关重要。特别是，矩阵 A 的初始值对训练后的性能有显著影响：在简单的分类任务中，性能从随机初始化 A 时的 67 % 增加到使用原则策略初始化 [?, Section 4.4] 时 A 的 80 %。为了实现成功的初始化，已经提出了不同的策略和参数化，即初始化导致状态 $x(k)$ 能够捕获输入的最近历史记录 $u(k), \dots, u(k-T)$ 一段时间 T 。此属性在标准 SSM 文献中称为内存。正如控制

²输入标记是表示输入数据的最小有意义组件的单位，无论是文本、图像还是模型处理的其他任何形式的信息。

理论中众所周知的那样，系统 (??) 的记忆与矩阵 A 的特征值直接相关。

Lemma 2.2: (非正式) 具有动态 (??) 的动力系统具有长程记忆，即从过去的输入中捕获信息，如果 A 的特征值在单位圆内并且非常接近单位圆周，即 $|eig(A)| \leq 1$ 和 $|eig(A)| \approx 1 \forall eig(A)$ 。

因此，SSM 文献中介绍的各种初始化方案旨在确保学习 A 矩阵的特征值的模近似等于（但不大于）1。对于其他矩阵的初始化，即 B, C 和 D ，使用标准初始化方法，例如 Glorot [?] 或 LeCun [?]，它们基本上从变换后的均匀分布或正态分布中提取初始值。因此，我们省略了 B, C 的初始化细节，并在下文中 D ，并请读者参考原始论文 [?], [?]。

E. 实现

SSM 文献中解决的主要挑战之一是如何有效地学习（训练时间）和部署（推理时间）递归 (??)。

在推理时，需要因果表示，因为模型无法访问当前时间步长之外的激励输入。因此，从初始激励 $u(1)$ 和零初始状态 $x(1) = 0$ 开始，直接使用循环表示 (??)。为了加快这一过程，使用了并行扫描算法 [?]，通过并行计算每个输出分量并缓存中间结果来有效地计算重复。

在训练期间，使用非因果表示是可能的（也是可取的），因为输入-输出对 $(u(k), y(k))$ 可用于所有 k 。文献中提出了不同的技术。某些架构可以利用并行扫描算法，并使用公式 (??) 中的循环表示。其他一些架构依赖于系统 (??) 的卷积表示，即

$$y(k) = \sum_{\tau=0}^k \bar{C} \bar{A}^{k-\tau} \bar{B} u(\tau). \quad (5)$$

这种卷积表示可以加快学习速度，因为完整的输入序列 $u(k) \forall k$ 可以在一个步骤中通过模型。

在学习算法方面，SSM 模型通常使用标准的随机梯度下降变化进行训练，即 Adam [?] 和反向传播 [?]。此外，它们可以利用与其他深度学习模型相同的启发式方法来改进训练，例如，辍学或归一化 [?]。

F. 脚手架和图层

尽管学习方程 (??) 中的动力学是 SSM 的主要关注点，但这些动力学并不是简单地孤立地实现的。事实上，输入 u 的预处理和输出 y 的后处理对于确保良好的性能是必要的。在本文中，我们将前处理和后处理的代数运算称为动力学 (??) 中围绕 SSM 计算的脚手架。图 ?? 中提供了 SSM 中使用的架构的一般概述。

文献中提出了一系列不同的脚手架选择，从标准多层感知器 (MLP) 选择到门控操作，如定义 ?? 中定义。通常，在输入 \bar{u} 馈送到系统 (??) 之前，先对输入执行线性或非线性映射。计算出输出 y 后，通常会执行门控操作来控制从输入 \bar{u} 到输出 \bar{y} 的信息流。直观地说，gate $g(\bar{y}, \bar{u})$ 根据通过 softmax 操作 \bar{u} 的输入来控制哪些输出 \bar{y} 设置为零。

Definition 2.3: 给定两个向量 $x_1, x_2 \in \mathbb{R}^p$ ，门控运算定义为 $g(x_1, x_2) := x_1 \odot \sigma(Wx_2)$ ，其中 $W \in \mathbb{R}^{p \times p}$ ， \odot 是逐元素乘法， σ 是 softmax 运算。³

³在 SSM 文献中，有时会使用其他非线性，例如 ReLU，SiLU 等。

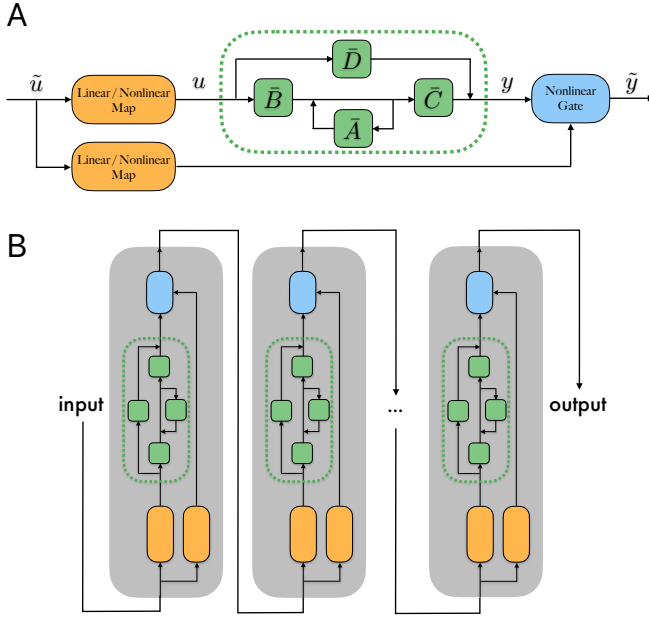


Fig. 1: A. SSM 的一般脚手架。动态模型 (??) 以绿色表示。SSM 的输入经过预处理，并在跳跃连接 (较低信号) 中分叉。预处理图的性质 (线性或非线性) 取决于特定的脚手架。然后使用非线性门对递归输出进行后处理。B. SSM 的整体架构。每个 SSM, 包括其脚手架 (图 1.A.) 都是以分层方式构建的，其中一层的输出是下一层的输入。

与深度学习中的常见做法一样，几层 SSM (动态 (??) 和随附的脚手架) 堆叠在一起，其中每层都处理前一层的输出作为其输入，然后将其输入到下一层。这是可能的，因为输入 y 和输出 $u \in \mathbb{R}^q$ 具有相同的维度。例如，在较小的任务中，例如 LRA 基准 [?]，SSM 由 6 个结构相同的层 (具有不同的动态矩阵) 组成，系统的大小以 $p \in [64, 512]$, $q \in [32, 1024]$ 为单位。对于语言建模，层数和系统大小可以明显增加。

需要注意的是，脚手架的选择和设计不是很清楚，通常选择在实践中性能最好的脚手架。

III. 审查现有方法

在本节中，我们概述了文献中最突出的 SSM 建议。由于现有的 SSM 是相互依赖的，因此本节中的呈现顺序是按时间顺序排列的。我们详细介绍了每种架构如何解决第 ?? 节中描述的注意事项。我们还在表 ?? 中总结了它们的主要特征。

A. 结构化状态空间序列模型 (S4)

S4 模型 [?] 是第一个基于状态空间表示的模型。

a) 参数：S4 模型从连续时间模型 (??) 开始，其中强加在矩阵 A 上的结构为

$$A = \text{diag}(\lambda_1, \dots, \lambda_p) + rs^* \quad (6)$$

与 $\lambda_i \in \mathbb{C} \forall i$ 和 $r, s \in \mathbb{C}^p$ 。这是一个对角矩阵加上一个低秩更新。我们注意到，这种结构类似于闭环动力学矩阵 $A_{CL} = A + BK$ 。

b) 离散化：离散时间版本 (??) 是通过将双线性变换应用于具有离散化步长 $\Delta \in \mathbb{R}$ 的动力学 (??) 来计算的，即

$$\bar{A} = (I - \frac{\Delta}{2}A)^{-1}(I + \frac{\Delta}{2}A), \quad \bar{B} = (I - \frac{\Delta}{2}A)^{-1}\Delta B, \quad (7)$$

$\bar{C} = C$ 和 $\bar{D} = D$ 。请注意，这种离散化方法的选择通过离散化步骤 Δ 耦合了 \bar{A} 和 \bar{B} 的参数化，这是大多数 SSM 的共同特征。

c) 结构和初始化：该模型以单输入单输出 (SISO) 方式构建，即输入的每个分量 (称为输入通道) u_i $i = 1, \dots, q$ 被馈送到一个单独的系统 (??)，每个分量产生一个标量输出 y_j 与 $j = 1, \dots, q$ 。每个 q SISO 子系统的每个动力学矩阵 A 都使用 HiPPO 理论 [?] 进行初始化，得到图 ?? 所示的特征值。从本质上讲，HiPPO 理论提供了一种基于数学的方法，可以放置连续时间动力学矩阵的特征值，以便它可以长输入序列上的信息压缩到其状态中。尽管原始 S4 不会将初始化偏向边缘稳定性以确保远程内存 (根据 Lemma ??)，但后续工作 SaShiMi [?] 强制执行 $\text{Re}(\lambda_i) \in \mathbb{R}^- \forall i$ 以确保稳定性。

d) 实现：在训练时，使用卷积表示 (??)。为了提高计算效率，利用了 \bar{A} (??) 的结构，因为 Sherman-Morrison 公式 [?] 可用于计算其逆 in (??)，从而仅导致标量的反转。在推理时，直接使用模型 (??) 的循环表示。

e) 脚手架：最初，用于 S4 块的预处理和后处理的脚手架与用于门控 MLP 的脚手架相同。后来，引入了更复杂的脚手架 H3 [?] 来模仿变压器的操作。H3 脚手架使用原始信号的总和，输入信号的时移版本用于上信号的线性图和下部信号的标准线性图，如图 ?? .A 所示。后处理仍然是一个门控功能。

B. 对角线结构态空间序列模型 (S4D)

最初提出的对角线状态空间 (DSS) [?] 模型及其增强 S4D [?] 建立在 S4 模型之上。他们通过首次引入假设 ?? 来简化动力学矩阵的结构，从而改进了计算。

a) 参数：S4D 论文的主要贡献是引入了一种新的、更有效的矩阵结构 A 与假设 ?? 一致：

$$A = \text{diag}(\lambda_1, \dots, \lambda_p). \quad (8)$$

b) 离散化：离散时间版本 (??) 是通过将动态 (??) 应用精确离散化来计算的，离散化步骤为 $\Delta \in \mathbb{R}$ ，即

$$\bar{A} = e^{\Delta A}, \quad \bar{B} = (\Delta A)^{-1}(\bar{A} - I)\Delta B, \quad (9)$$

$\bar{C} = C$ 和 $\bar{D} = D$ 。

c) 结构和初始化：S4D 中使用的 SISO 结构与 S4 中的结构相同。S4D 的初始化也是使用 HiPPO 理论完成的，并增加了对得矩阵可以对角化以提高计算效率的见解。与 SaShiMi [?] 类似，用于初始化的 A 的特征值被约束在负半平面内。此初始化产生图 ?? 所示的特征值。

d) 实现：与 S4 类似，在训练时使用卷积表示 (??)，在推理时 (??) 循环表示。给定矩阵 \bar{A} 的对角线结构，可以有效地计算离散化 (??)。

e) 脚手架：S4D 的脚手架与 S4 中使用的脚手架相同。

Model	Features				
	Parametrization	Discretization	Structure	Implementation	Scaffolding
S4 [?]	LTI	Bilinear	SISO	Convolution and Recurrence	MLP / H3
S4D [?]	LTI	Exact	SISO	Convolution and Recurrence	MLP / H3
S5 [?]	LTI	Exact / Bilinear	MIMO	Parallel Scan	MLP / H3
LRU [?]	LTI	None	MIMO	Parallel Scan	MLP / H3
S6 [?]	LTV	Exact	MIMO	Custom Parallel Scan	Mamba
RG-LRU [?]	LTV	None	MIMO	Custom Parallel Scan	Mamba / Hawk / Griffin

TABLE I: Overview of the model features for the different SSM models considered. Acronyms used are as follows: Linear Time-Invariant (LTI), Linear Time-Varying (LTV), Single Input Single Output (SISO), Multiple Input Multiple Output (MIMO). Details on the scaffolding can be found in MLP [?], H3 [?], Mamba [?], Hawk and Griffin [?].

C. 简化结构化状态空间序列模型 (S5)

S5 参数化 [?] 对之前提出的 S4D 进行了简化, 并利用多输入多输出 (MIMO) 系统 (与 SISO 相对) 的概念来简化架构组件并增强计算。

a) 参数: 使用的参数化与 S4D 相同。

b) 离散化: S5 适用于 S4 和 S4D 中提出的两种离散化: 双线性 (??) 和精确 (??)。

c) 结构和初始化: S5 模型的主要贡献是对先前提出的模型引入了 MIMO 解释, 从而实现了显著的计算增强。特别是, 完整的输入向量 $u \in \mathbb{R}^q$ 被馈送到单个 MIMO 系统 (??) (更大尺寸), 而不是 q SISO 标量系统 (较小尺寸)。这是通过堆叠 S4 和 S4D 中使用 $\bar{A}, \bar{B}, \bar{C}$ 子系统矩阵来实现的。使用 HiPPO 理论再次初始化矩阵 A , 并产生与 S4D 相同的初始特征值 (图 ??)。

d) 实现: MIMO 结构与 \bar{A} 的对角线参数化一起, 允许通过并行扫描算法从输入组件并行计算单个输出组件 [?]。因此, 训练时的计算和推理时的计算都可以在它们的循环表示 (??) 中有效地计算。

e) 脚手架: MIMO 表示允许简化先前为 S4 和 S4D 提出的脚手架。其原因是, 尽管堆叠动力学矩阵 \bar{A} 是对角线的, 但堆叠矩阵 \bar{B}, \bar{C} 是密集的, 因此输入和输出分量耦合。这允许删除 S4 和 S4D 输出后处理中存在的混叠层。

D. 线性循环单元 (LRU)

LRU 模型试图通过揭示其基本组件来简化以前的 SSM 提案。LRU 的主要贡献之一是通过特征值显式编码远程内存。这允许摆脱 HiPPO 理论, 直接使用离散时间模型以及控制理论的边际稳定性概念。

a) 参数化: LRU 模型直接参数化离散时间动力学 (??), 即

$$\bar{A} = e^{-e^{diag(\lambda_1, \dots, \lambda_p) + i diag(\theta_1, \dots, \theta_p)}}, \quad \bar{B} = e^{\gamma \Gamma} \quad (10)$$

i 复数单位, $\lambda_j, \theta_j \in \mathbb{R} \forall j = 1, \dots, p$, $\Gamma \in \mathbb{C}^{p \times q}$ 密集复值矩阵, $\gamma \in \mathbb{R}$ 。请注意, 此参数化直接表示 \bar{A} 的对角线条目, 因此极坐标中的特征值 (即 $r_j = e^{-e^{\lambda_j}}$ $a_j = r_j + i \theta_j$) 被构造限制在区间 $[0, 1]$ 。这也是第一个在 \bar{A} 和 \bar{B} 之间没有共享参数的参数化。

b) 离散化: LRU 模型是第一个不被视为连续时间模型离散化的 SSM。取而代之的是, 直接使用 $\bar{A}, \bar{B}, \bar{C}, \bar{D}$ 的离散参数化。

c) 结构和初始化: 该模型的结构与 S5 相同, 其中考虑的是 MIMO 系统, 而不是 q SISO 子系统。给定参数化 (??), 引理 ?? 通过约束 \bar{A} 的特征值位于单位圆盘中来自动强制执行。因此, 通过定义 r 和 θ 的范围, 直接在极坐标中执行初始化, 其中 r 和 θ 被均匀采样, 从而产生图 ?? 所示的特征值。

d) 实现: 与 LRU 类似, 该模型使用并行扫描算法实现 [?] 用于训练和推理。

e) 脚手架: LRU 中使用的脚手架与 S5 中使用的脚手架相同。

E. 扫描选择性结构化状态空间序列模型 (S6)

S6 参数化 [?] 首次引入了动态 (??) 的线性时变表示。该系统的时变性质源于矩阵 \bar{A}_k, \bar{B}_k 和 \bar{C}_k 是每个时间步长 k 输入 $u(k)$ 的函数, 作者称之为选择性。虽然更具表现力, 但时变表示带来了计算挑战。本文的主要贡献是解决这些问题, 以便在实践中利用该系统更具表现力的时变性质。

a) 参数: 与 S4D 类似, S6 参数化依赖于假设 ?? 中时不变的对角线 A 矩阵 (??)。S6 参数化的新颖之处在于, 鉴于 B 和 C 的输入依赖性, 它们被参数化为时变:

$$B_k = W_B u(k) \quad C_k = W_C u(k) \quad (11)$$

其中 W_B 和 W_C 是适当尺寸的线性投影矩阵。

b) 离散化: 与 S4D 类似, S6 模型也使用精确离散化来计算离散时间动力学 (??)。然而, 在这种情况下, 时间步长 Δ_k 本身是时变的, 因为它是输入的函数

$$\Delta_k = \sigma(W_\Delta u(k)), \quad \bar{A}_k = e^{\Delta_k A}, \quad \bar{B}_k = (\Delta_k A)^{-1}(\bar{A}_k - I)\Delta_k B_k, \quad (12)$$

$\bar{C}_k = C_k$ 和 $\bar{D}_k = D_k$, $W_\Delta \in \mathbb{R}^{1 \times q}$ 并 $\sigma(\cdot)$ softplus 功能。

c) 结构和初始化: 与 S5 类似, 该模型以 MIMO 方式构建。为了初始化动态矩阵 A , 利用其对角线参数化: $\lambda_i = -i \forall i = 1, \dots, p$, 确保特征值位于负半平面内。由于离散化步骤 Δ_k 的时变特性, 离散时间矩阵的特征值 \bar{A}_k 具有与输入相关的初始化, 如图 ?? 所示。然而, 为了强制执行引理 ??, 得到的特征值保证位于单位圆盘中, 因为 Δ_k 和 A in (??) 分别是正值和负值。

d) 实现: [?] 工作的主要贡献之一是在推理和训练时提供时变动态 (??) 的有效实现, 矩阵 (??) 和 (??)。一般来说, S6 模型的时变特性使得卷积表示在实际使用中计算成本太高。为了克服这些局限性, S6 论文提出了一种高度定制的并行扫描算法 [?] 用于训练和推理的变体。

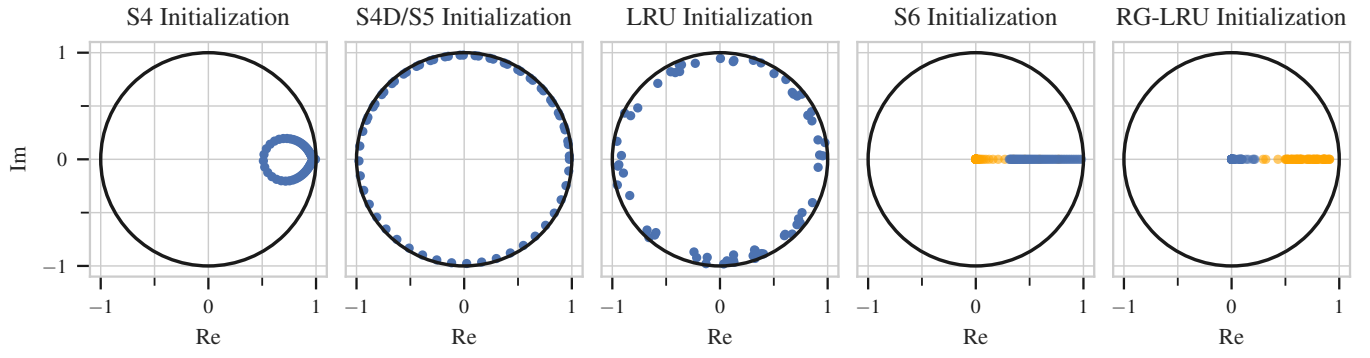


Fig. 2: 单位盘的复平面表示和离散时间动力学矩阵的特征值 \bar{A} (??) 由每个模型 S4、S4D、S5、LRU、S6 和 RG-LRU 中的初始化方法产生。由于 S6 和 RG-LRU 的初始化与输入相关，因此我们绘制了两个样本输入（蓝色和橙色）的初始化。

Model	LRA Task [%]						
	ListOps	Text	Retrieval	Image	Pathfinder	Path-X	avg.
Random	10.00	50.00	50.00	10.00	50.00	50.00	36.67
Transformer [?] (paper results)	36.37	64.27	57.46	42.44	71.40	FAIL	53.66
S4 [?] (paper results)	59.60	86.82	90.90	88.65	94.20	96.35	86.09
S4D [?] (paper results)	60.52	87.34	91.09	88.19	93.96	92.80	85.65
S5 [?] (paper results)	62.15	89.31	91.40	88.00	95.33	98.58	87.46
LRU [?] (paper results)	60.20	89.40	89.90	89.00	95.10	94.20	86.30
S6 [?]	38.02	82.98	72.14	69.82	69.26	67.32	66.59
RG-LRU [?]	32.34	71.75	66.58	61.15	73.38	69.53	62.45

TABLE II: Model performance in terms of test accuracy on the LRA benchmark. The first entry (Random) represents the performance of random guessing on the task, i.e., indicating the baseline above which a model is considered to have learned a meaningful representation. Models failing to exceed this baseline on a task are marked as FAIL. The best model on each task is highlighted in bold .

对于 S4、S4D、S5 和 LRU，我们报告了原始论文中最佳变体的性能，以呈现最具竞争力的结果。这些模型的其他变体可能在 LRA 基准中未包含的任务上表现更好；有关这些变体的更多详细信息，请参阅原始论文。由于文献中没有报道 S6 和 RG-LRU 的 LRA 基准测试的性能，因此我们提供了我们自己实现这些架构的结果，我们提供 这里⁶模型的超参数和我们实现的训练细节在公共代码存储库中说明。

在 LRA 基准测试中，基于 LTI 的型号 S4、S4D、S5、LRU 优于基于 LTV 的型号 S6、RG-LRU 和 Transformer。从控制理论的角度来看，这是令人惊讶的，因为一般的 LTV 定义将 LTI 系统作为一种特例，即 LTV 系统的性能至少应与 LTI 系统一样好。然而，对于 S6 或 RG-LRU 的特定变参数化，情况并非如此，因为无法实现 $\bar{A} = \bar{A}_k \forall k$ 。我们试图通过改变 S6 和 RG-LRU 的初始化来提高基于 LTV 的模型的性能，并根据 Lemma ?? 强制 $\bar{A}_k \forall k$ 的输入相关特征值更接近边际稳定性。然而，这导致两个模型的表现都差得多，或者根本无法学到任何有意义的东西。虽然边际稳定的特征值对于基于 LTI 的模型似乎很重要，但对于基于 LTV 的模型来说并非如此。迄今为止，这种行为还不是很清楚。最后，尽管基于 LTV 的模型与 Transformer [?] 密切相关，但它们在 LRA 基准测试中通常表现更好。

V. 结论和未来机遇

本文概述了最先进的状态空间模型 (SSM)，并从控制理论的角度探讨了它们的特点。在此过程中，我们强调了与标准控制理论概念的许多联系，例如记忆和边际稳定性之间的联系。此外，我们在远程竞技场 (LRA) 基准测试中比较了经过审查的 SSM，发现最近基于 LTV 的 SSM 的表现比基于 LTI 的 SSM 更差。从控制理论的角度来看，这提出了许多有趣的研究问题，涉及获得与 LTI 模型相同性能的 LTV 参数化，以及对特征值在基于 LTV 的模型中的作用的更深入理解。

SSM，尤其是 LTV 版本，依赖于动态，其中动态矩阵取决于系统的输入（激励）。然而，在 SSM 文献中，这些动力学产生的理论性质仍然知之甚少。SSM 和线性系统理论之间的明显联系为大型基础模型提供了充足的机会。此外，从 LRU 模型可以看出，控制理论的见解有可能为 SSM 的更好设计提供信息。