通过时序逻辑研究状态空间模型的表达能力

Eric Alsmann、Lowejatan Noori 和 Martin Lange理论计算
机科学 / 形式化方法 卡塞尔大学，德国{
eric.alsmann,l.noori,martin.lange}@uni-kassel.de

arXiv:2601.19467v1 [cs.LO] 27 Jan 2026

摘要

我们研究状态空间模型（SSM）的表达能力。近年来，状态空间模型逐渐成为大型语言模型中 Transformer 架构的一种潜在替代方案。基于近期研究，我们通过有限轨迹上的线性时序逻辑的片段与扩展来分析 SSM 的表达能力。研究结果表明，SSM 的表达能力会因其底层门控机制不同而存在显著差异。我们还区分了基于定宽算术运行的 SSM（量化模型）与具有无限精度的 SSM：前者的表达能力局限于正则语言，后者则能够捕捉计数性质和非正则语言。此外，我们系统地比较了不同的 SSM 变体与关于 Transformer 的已有研究结果，从而阐明这两种架构在表达能力方面的关系。

1 引言

状态空间模型（SSM）已成为传统序列建模架构（如广受欢迎的 Transformer 架构）的一种潜在替代方案。本文通过建立其计算能力的下界，考察 SSM 的表达能力。分析聚焦于影响表达能力的两个关键维度：所用 SSM 层的类型，以及计算中采用的算术精度。我们具体研究对角门控状态空间模型，例如 S6（参见 Gu & Dao，2024年），其中门控矩阵可以依赖于输入，但必须保持对角结构；以及时不变状态空间模型，例如 S4（参见 Gu 等人，2022年），其中门控矩阵无论输入如何都保持不变。对于每种模型变体，我们都考察定宽算术（所有计算均采用固定的比特数）和对数精度算术（精度随输入长度呈对数级增长）两种情形。

尽管实证研究推动了序列建模领域的大部分进展，基础理论分析则通过揭示各种架构原则上能够或不能表示什么，为我们提供了互补的见解，且不受训练动态的影响。受近期 Transformer 相关分析所取得成功的启发（Strobl 等人（2024年）），我们采用相同的逻辑与复杂性理论框架研究状态空间模型。这使我们能够进行精确的相对比较：我们将状态空间模型变体映射到近期用于刻画 Transformer 变体（如 UHAT、AHAT）的完全相同的逻辑片段。我们的分析围绕表达能力的三个概念层次展开。最基础的层次是模式匹配（线性时序逻辑／一阶逻辑），它体现了根据事件的相对顺序检测事件序列、但不进行计数的能力。在此基础上，我们考察模谓词，它能够追踪周期性位置，类似于 Transformer 架构中的位置嵌入。最后，我们分析全局计数，它支持对整个序列中的数量进行比较，并且必然需要对数精度算术。

通过将状态空间模型映射到这些层级，我们为特定架构在理论上能够学习或可证明无法学习的模式提供了结构性保证。我们证明，对角门控状态空间模型能够识别由有限轨迹上的纯过去线性时序逻辑（PLTL$_f$）定义且具有计数能力的语言，而时不变状态空间模型能够表示由包含一元时序算子和模谓词的片段所表达的语言。此外，我们证明，全局计数严格要求使用对数精度算术。本文确立的下界为

序列建模神经网络架构理论基础的相关研究日益增多，本文的工作为其添砖加瓦，并补充了关于这些架构实际表现的经验观察。尤为重要的是，由此得到的不可能性证明揭示了根本性的架构瓶颈，例如，对角固定精度状态空间模型无法表示像（aa）*这样的非单调模式；这一结论不受训练数据或优化过程的影响。从这个意义上说，我们的基础性刻画补充了经验基准测试，明确指出了哪些任务对于特定的状态空间模型变体在结构上遥不可及。

相关工作。近期关于状态空间模型理论基础的研究涉及电路复杂度（参见 Merrill 等人（2024））、自动机理论（参见 Sarrof 等人（2024））和通信复杂度（参见 Zubic 等人（2025））。Alsmann 和 Lange（2025）研究了验证状态空间模型的计算复杂性。据我们所知，本文首次将状态空间模型的表达能力与形式逻辑联系起来。

关于状态空间模型形式表达能力的研究始于 Merrill 等人（2024），他们通过电路复杂度给出了上界。他们证明，在对数精度下运行的对角门控状态空间模型和时不变状态空间模型，以及在固定精度（例如浮点）下运行的模型，都属于 TC^0。这是一个函数类，其中的函数可由多项式规模、常数深度且使用阈值门的布尔电路定义。他们还证明，具有任意门的状态空间模型（即门可以是任意输入相关矩阵）能够识别所有正则语言，包括 NC^1-完全语言。这意味着表达能力严格提升，除非 $\mathsf{TC}^0 = \mathsf{NC}^1$。Sarrof 等人（2024）的后续研究为对角门控状态空间模型给出了下界，证明它们能够识别所有无星语言。他们还证明，在某些限制条件下，该下界是紧的，即这些状态空间模型能够识别一个正则语言，当且仅当该语言是无星语言。我们的工作大幅扩展了 Sarrof 等人（2024）给出的下界。他们证明了对角状态空间模型（采用特定输出函数）能够识别无星语言，而我们的定理 1 将其作为更广泛层级中的一个特例重新得到。关键的是，我们沿着三个新的方向扩展了分析：（1）我们刻画了时不变 和混合 状态空间模型，并将它们与模谓词（$\mathrm{PLTL}_f[\mathrm{MOD}]$）联系起来；（2）我们分析了算术精度的影响，表明对于非正则计数语言（例如 $a^n b^n c^n$），对数精度 是严格必需的；（3）我们将这些发现纳入现有的 Transformer 架构体系（UHAT/AHAT），从而能够直接比较不同架构。Zubic 等人（2025）证明，在固定精度下运行的状态空间模型只能识别正则语言。

与状态空间模型表达能力的研究并行，许多研究通过与逻辑建立联系来探讨 Transformer 架构的表达能力（有关这些结果的综述，参见 Strobl 等人（2024））。与本文最相关的工作包括 Yang 等人（2024a），他们证明了唯一硬注意力变换器与一阶逻辑之间的联系；以及 Barcelo 等人（2024）和 Yang 等人（2025），他们通过扩展线性时序逻辑，研究了多种 Transformer 架构的计数能力。

2 基础知识

状态空间模型。我们在 Merrill 等人（2024）提出的广义设定下研究状态空间模型。为对这一模型进行结构化分析，我们将参照 Sarrof 等人（2024）和 Merrill 等人（2024）采用的类似方法，对其架构进行形式化。状态空间模型层 l 定义为一个元组（$\boldsymbol{h}_0$，门，inc ϕ），其中 $\boldsymbol{h}_0 \in \mathbb{R}^d$。函数 门 是一个映射 $\mathbb{R}^d \to \mathbb{R}^{d \times d}$，$inc$ 是一个映射 $\mathbb{R}^d \to \mathbb{R}^d$，而 ϕ 是一个映射 $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$。状态空间模型层将向量输入序列 $\boldsymbol{x}_1 \cdots \boldsymbol{x}_k \in (\mathbb{R}^d)^*$ 通过以下过程转换为输出序列 $\boldsymbol{z}_1 \cdots \boldsymbol{z}_k \in (\mathbb{R}^d)^*$：它通过线性递推计算出中间序列 $\boldsymbol{h}_1 \cdots \boldsymbol{h}_k \in (\mathbb{R}^d)^+$。

$$\boldsymbol{h}_t = gate(\boldsymbol{x}_t) \cdot \boldsymbol{h}_{t-1} + inc(\boldsymbol{x_t}) \quad \text{for } 1 \leq t \leq k$$

随后通过 $\boldsymbol{z}_t = \phi(\boldsymbol{h}_t, \ \boldsymbol{x}_t)$ 生成输出。

一个由 L 层组成的 SSM S，用于处理有限字母表 Σ 上的序列。它表示为一个元组 $S =$ (嵌入 l_1, $\cdots$, l_L, 输出)。每个 l_i 均指上文定义的一层，而 嵌入 是一个函数 $\Sigma \to \mathbb{R}^d$，输出 是一个函数 $\mathbb{R}^d \to \mathbb{R}^d$。SSM 按如下方式计算函数 $\Sigma^* \to \mathbb{R}$：设 $\sigma = \sigma_1 \cdots \sigma_k \in \Sigma^*$ 为一个词。首先，SSM 通过令 $\boldsymbol{x}_i^0 =$ 嵌入(σ_i)，计算该词的嵌入 $\boldsymbol{x}_1^0 \cdots \boldsymbol{x}_k^0$。随后，对每一层 $1 \leq j \leq L$：计算

$\boldsymbol{z}_1^j \cdots \boldsymbol{z}_k^j = l_j(\boldsymbol{x}_1^{j-1} \cdots \boldsymbol{x}_k^{j-1})$。每一层的输出都作为后续层的输入，即 $\boldsymbol{x}_1^{j+1} \cdots \boldsymbol{x}_k^{j+1} = \boldsymbol{z}_1^j \cdots \boldsymbol{z}_k^j$。SSM 的最终输出记为 $\boldsymbol{y}_1 \cdots \boldsymbol{y}_k$，通过逐元素应用 输出 得到：$\boldsymbol{y}_i = \text{输出}(\boldsymbol{z}_i^L)$。最后，$S(\sigma)$ 的输出由 $\boldsymbol{y}_k$ 计算得出。若 S 接受词 σ，则称 $S(\sigma) = 1$。否则，该词被拒绝。我们将 S 接受的语言记为 $L(S)$。

近期关于状态空间模型表达能力的研究（参见 Merrill 等人（2024）和 Sarrof 等人（2024））区分了两类主要的状态空间模型，它们在允许的门函数方面有所不同。时不变状态空间模型这一类（参见 Mehta 等人（2023）和 Orvieto 等人（2023））只允许使用门函数满足以下条件的层：存在一个矩阵 $A \in \mathbb{R}^{d \times d}$，使得对所有 $\boldsymbol{x} \in \mathbb{R}^d$，$\text{门}(\boldsymbol{x}) = A$。对角门控状态空间模型这一类（参见 Gu & Dao（2024年）、De 等人（2024年）和 Yang 等人（2024b））只允许使用门函数满足以下条件的层：$\text{门}(\boldsymbol{x})$ 是一个对角矩阵（其元素非负），但可以依赖于 $\boldsymbol{x}$。一些状态空间模型架构（例如 Sun 等人（2023年）提出的 RetNet）使用对角时不变门。我们将结合对角层和时不变层的状态空间模型称为混合状态空间模型。据我们所知，混合状态空间模型尚未在实践中使用，但从理论角度来看，它们很有意义，因为它们为表达能力提供了上界，而这些上界直接适用于时不变和对角状态空间模型。因此，我们将其纳入研究。最后，我们将门控机制不受任何限制的状态空间模型称为任意状态空间模型（参见 Hasani 等人（2022年））。全局输出输出以及每一层的输出 ϕ（该层还会接收模拟残差连接的初始输入）均由非线性函数计算。实践中会使用不同类型的非线性函数。本文假设这些函数由带 ReLU 激活函数的前馈神经网络（FNN）表示。

算术。在我们的表达能力分析中，我们考察 SSM 计算的算术精度的两种不同设定。首先，我们研究定精度算术，即所有数值和计算都使用固定数量的位 b（例如浮点或定点表示），且不受输入长度影响。如果 SSM 的所有数值和计算都仅使用这些 b 位进行，我们就称该 SSM 采用定精度算术。我们还假设定精度算术具有饱和特性：发生溢出时，结果被限制为可表示的最大值（下溢时则限制为可表示的最小值），并且算术运算相对于舍入是单调的。这是例如浮点算术中的标准行为，参见 876（2019年）。其次，我们研究对数精度算术，其精度随输入长度呈对数增长——具体而言，对于长度为 n的输入，使用 $O(\log n)$位。Merrill 等人（2024）研究了 SSM 的表达能力，Merrill 和 Sabharwal（2023）以及 Hao 等人（2022）则研究了 Transformer 架构在这一设定下的表达能力，因为这种设定既符合实际，又具有理论意义：它提供了足够的精度，可以准确计数出现次数，或计算随输入规模线性增长的总和，同时避免了无限精度这一不切实际的假设。这种平衡方案尤其值得关注，因为它为限制严格的定精度模型与不切实际的无限精度模型提供了折中方案。

有限轨迹上的线性时序逻辑。我们采用 De Giacomo & Vardi（2013）提出的有限轨迹上的线性时序逻辑（LTL$_f$）来分析 SSM 的表达能力。它通过时序算子扩展了命题逻辑，从而能够表达涉及事件顺序和时机的性质。我们考虑 De Giacomo 等人（2021）研究的 LTL$_f$ 纯过去片段 PLTL$_f$，它使用时序算子 昨天 Y（“在前一个位置”）、先前 P（“在过去某个时刻”）和 自从 S（“自某个事件发生以来，另一个事件持续发生”）。令 $\mathcal{P}$ 为一个有限的原子命题集。PLTL$_f$ 的语法定义如下：

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathrm{Y}\,\varphi \mid \mathrm{P}\,\varphi \mid \varphi\ \mathrm{S}\ \varphi$$

PLTL$_f$ 的公式是在字母表 $\Sigma = 2^{\mathcal{P}}$ 上的有限词上进行解释的。给定一个词$\sigma = \sigma_1 \cdots \sigma_n \in \Sigma^*$ 和 $i \in \{1, \ldots, n\}$，PLTL$_f$ 的语义归纳定义如下：

$$
\begin{aligned}
\sigma, i \models p \quad &\iff p \in \sigma_i \\
\sigma, i \models \neg\varphi \quad &\iff \sigma, i \not\models \varphi \\
\sigma, i \models \varphi \wedge \psi \quad &\iff \sigma, i \models \varphi \text{ and } \sigma, i \models \psi \\
\sigma, i \models \mathtt{Y}\,\varphi \quad &\iff i > 1 \text{ and } \sigma, i-1 \models \varphi \\
\sigma, i \models \mathtt{P}\,\varphi \quad &\iff \exists\, 1 \le k \le i : \sigma, k \models \varphi \\
\sigma, i \models \varphi\ \mathtt{S}\ \psi \quad &\iff \exists\, 1 \le k \le i : \sigma, k \models \psi \text{ and f.a } \ k < j \le i : \sigma, j \models \varphi
\end{aligned}
$$

我们称 σ 是 φ 的模型，当且仅当 $\sigma, n \models \varphi$（或简记为 $\sigma \models \varphi$）。此外，我们将公式 $L(\varphi)$ 的语言记为 $= \{\sigma \in \Sigma^* \mid \sigma \models \varphi\}$。我们使用常见缩写 tt（恒为真）、ff（恒为假）以及 $\mathrm{H}\,\varphi = \neg \mathrm{P}\,\neg\varphi$（$\varphi$ 在所有过去位置上成立）。此外，我们将 PLTL$_f$的一元片段称为 UN-PLTL$_f$，该片段仅使用 昨天 和 先前。

计数扩展。Yang 和 Chiang（2024年）以及 Barcelo 等人（2024年）通过在 LTL$_f$ 中加入计数算子，分析了若干 Transformer 架构在计数能力方面的表达能力。PLTL$_f[\overleftarrow{\#}\ \overrightarrow{\#}]$ 在语法上通过增加一个情形扩展了 PLTL$_f$：若 $\varphi_1, \cdots, \varphi_i, \psi_1, \cdots, \psi_j$ 是 PLTL$_f[\overleftarrow{\#}\ \overrightarrow{\#}]$ 公式，且 $a_1 \cdots a_i, b_1, \cdots b_j \in \mathbb{Z}$、 $c \in \mathbb{N}$和$\sim \in \{<, \le, =, \ge, >\}$，则

$$\sum_i a_i \overleftarrow{\#}\, \varphi_i + \sum_j b_j \overrightarrow{\#}\, \psi_j \sim c$$

也是一个 PLTL$_f[\overleftarrow{\#}\ \overrightarrow{\#}]$ 公式。在词 σ 上的位置 i处求值公式时，$\overleftarrow{\#}\ \varphi$ 会统计满足 $\sigma, j \models \varphi$ 的位置 $j \le i$ 的数量，而 $\overrightarrow{\#}\ \varphi$ 则以类似方式统计位置 $j \ge i$。

PLTL$_f[\overleftarrow{\#}]$ 表示这样一种逻辑：其中每个计数子公式都对所有 j具有 $b_j = 0$。使用 $\overleftarrow{\#}$（而非 $\overrightarrow{\#}$）与状态空间模型的特性有关：不同于具有完整序列注意力的 Transformer，状态空间模型在每个步骤只能访问输入的前缀。

模谓词。我们引入 PLTL$_f$ 的第三种变体，加入了模谓词。PLTL$_f$[MOD] 的语法扩展了额外的原子公式 $\mathtt{MOD}_r^m$，适用于 $m > 1$ 和 $0 \le r < m$。其语义与 PLTL$_f$ 中的相同。对于新算子，语义扩展为

$$\sigma, i \models \mathtt{MOD}_r^m \iff i \equiv r \pmod{m}\,.$$

关键的是，允许使用这些模谓词，会使表达能力严格超越无星语言（对应于标准 PLTL$_f$）。这使得我们能够定义周期性质，例如识别语言 $(aa)^*$，而这需要区分偶数位置和奇数位置。形式上，这一扩展将表达能力提升到 AC^0 中所有可定义正则语言的类（Straubing, 1994年）。这一复杂度类是我们后续分析的重要参照点，因为我们将证明，它与混合状态空间模型以及配备位置编码的 Transformer 的能力相符。

3 结果

我们的分析为不同架构变体和精度设置下的 SSM 表达能力确立了下界。结果揭示了一个有趣的层级：表达能力会因门控机制和算术精度的不同而显著变化。图 1 概述了这些发现，也展示了我们的 PLTL$_f$ 片段与电路复杂度和一阶逻辑之间的关系。附录 A.1 对这些联系作了详细说明。

对角状态空间模型。定理 1 表明，采用定宽算术的对角状态空间模型能够识别由 PLTL$_f$ 可定义的语言，这对应于一阶可定义语言类，也就是无星正则语言。该结果具有构造性：我们展示了如何根据公式的嵌套深度对其进行分解，并通过适当的门控机制实现各个时态算子，从而将任意 PLTL$_f$ 公式系统地转换为等价的对角状态空间模型。

这一构造的核心见解是，尽管 Y 和 P 等时态算子可以通过简单的门控模式实现，但由于 自从-算子的递归定义 φ S $\psi \equiv \psi \vee (\varphi \wedge$ Y(φS ψ))，它需要输入相关的对角门控。这种对当前输入求值的依赖使门具有对角结构。

我们用一个匹配的不可表达性结果补充这一下界。定理 3 证明，对角状态空间模型在固定精度下无法识别简单的非星自由语言 $(aa)^*$。这一局限源于一项基本的单调性（引理 2）：当对角状态空间模型反复处理同一个输入符号时，其输出最终必然趋于稳定。

采用对数精度算术的对角状态空间模型，其表达能力显著增强。定理 4 表明，这些模型能够识别所有可由纯过去时线性时序逻辑扩展语言 PLTL$_f[\overleftarrow{\#}]$，定义的语言；该逻辑扩展了纯过去时线性时序逻辑，并加入向后计数算子。Barcelo 等人（2024）；Yang 与 Chiang（2024）也运用这一逻辑研究了 Transformer 架构的表达能力。计数公式的扩展使模型能够识别非正则语言（甚至非上下文无关语言），例如 $\{a^n b^n c^n \mid n \geq 0\}$，它可以表示为

$$\varphi = \mathtt{H}\left((a \to \neg\, \mathtt{P}(b \vee c)) \wedge (b \to \neg\, \mathtt{P}\, c)\right) \wedge (\overleftarrow{\#}\, a - \overleftarrow{\#}\, b = 0) \wedge (\overleftarrow{\#}\, c - \overleftarrow{\#}\, b = 0)$$

时不变状态空间模型。尽管我们猜想时不变状态空间模型无法表达 since 算子所刻画的时间依赖关系（猜想 1），但它们具备另一种能力：计算关于序列位置的模谓词。引理 6 展示了时不变状态空间模型如何利用循环置换矩阵维护模 m 的计数器，从而能够识别诸如 $(aa)^*$ 这样的语言，而固定精度的对角状态空间模型无法识别这些语言。
由此得到定理 7 中的刻画：固定精度的时不变状态空间模型能够识别由 UN-PLTL$_f$[MOD]，（即扩展了模谓词的纯过去时线性时序逻辑一元片段）定义的所有语言。采用对数精度时，它们还能处理计数算子，识别 UN-PLTL$_f$[MOD $\overleftarrow{\#}$] 中的语言。

混合门与任意门。通过考察结合多种门控机制的状态空间模型，我们补全了这一层级。推论 8 证明，固定精度的混合状态空间模型（结合对角层与时不变层）能够识别 AC^0，中的所有正则语言，从而有效地涵盖了两种单独架构各自能力的并集。具有任意门的状态空间模型表达能力更强，能够识别所有正则语言；这一点已由 Merrill 等人（2024）的先前工作证明。

4 对角状态空间模型的表达能力

我们证明，对角状态空间模型的表达能力至少与 PLTL$_f$ 相当，而 PLTL 在表达能力上等价于无星语言集。我们的证明思路沿袭 Sarrof 等人（2024），但将其作为基础，进一步构造 PLTL$_f$ 的扩展与限制。

为了在词 σ 的位置 i 处计算 PLTL$_f$ 公式 φ，可以自底向上方式计算φ的子公式。我们对这些子公式进行排序，使得每个子公式都在其自身的子公式之后进行计算。相互独立的子公式可以并行计算（在单个状态空间模型层中）。这意味着，我们所有构造所需的层数对应于公式 φ 的语法树深度，本文简称为嵌套深度。
定义 1。PLTL$_f$ 公式 φ 的嵌套深度nd(φ)归纳定义如下：nd(p) =nd(MOD$_r^m$) $= 0$，nd($\varphi \circ_{\text{bin}} \psi$) $=$max(nd(φ)，nd(ψ)) $+ 1$，nd($\circ_{\text{un}} \varphi$) $=$nd(φ) $+ 1$，以及 nd $\left(\sum_{j=1}^{k} a_j \cdot \overleftarrow{\#}\, \varphi_j \sim c\right)$ $=$max$\{$nd(φ_j) $\mid 1 \leq j \leq k\} + 1$，其中 $\circ_{\text{bin}} \in \{\wedge, \text{S}\}$ 和 $\circ_{\text{un}} = \{\neg, \text{P}, \text{Y}\}$。

这使我们能够按照上述方式对 φ 的子公式进行排序。对于 φ的每个子公式 ψ，我们构造一个对角状态空间模型层 l_ψ，用于计算每个位置 i 上 $\sigma, i \models \psi$ 是否成立。用于整个公式 φ 的状态空间模型 S_φ 随后计算 $\sigma, n \models \varphi$ 是否成立。Alsmann 和 Lange（2025）已经采用了类似的思路，证明固定精度下对角状态空间模型的可满足性问题是 PSPACE 完全。

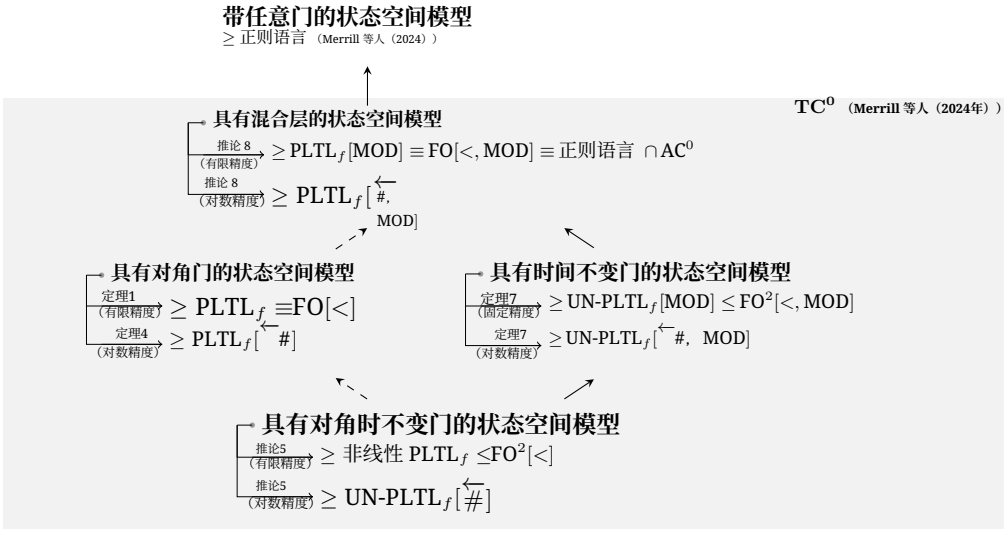

图 1：映射到逻辑片段和复杂度类的状态空间模型架构表达能力层级。我们建立了对角、时不变和混合状态空间模型的下界，并区分固定精度与对数精度（支持计数算子 $\overleftarrow{\#}$）。虚线箭头表示可证明的严格包含关系（定理 3）。灰色框界定了先前工作（Merrill 等人，2024）给出的 TC^0 上界。

定理 1。 固定精度的对角状态空间模型能够识别 PLTL 中可定义的所有语言 f.

证明概要。 详细证明见附录 A.2。每个子公式对应状态空间模型隐藏状态的一个维度。每层的增量函数与逐位置前馈神经网络共同补充前面各层中已求值子公式的信息，并处理合取、否定和比较等非时态子公式。每层处理完毕后，每个隐藏状态都是一个布尔向量，表示各个位置上哪些子公式成立。时态算子 Y 和 P 可以通过对角且时不变的门实现，因为它们不需要与当前输入进行任何合取。Y 始终指向紧邻的前一个位置，而 P 则累积截至当前位置的出现情况。相比之下，自从算子本质上具有递归性，其定义为 $\varphi \,\mathrm{S}\, \psi \equiv \psi \vee (\varphi \wedge \mathrm{Y}(\varphi \mathrm{S}\, \psi))$，因此会通过合取结合先前状态和当前状态。对该算子求值本质上依赖于当前输入对 φ 的求值，因此需要一个输入相关门。 □

接下来，我们将证明，这一刻画是紧的，即在固定精度下运行的对角状态空间模型无法识别 (aa)*，而该语言无法由 PLTL^f 定义。这一表达能力不足的结果基于固定精度下运行的对角状态空间模型的单调性。任何在固定精度下运行的对角状态空间模型，在反复接收相同输入后，最终都必须变为常量。

引理 2。 设 $\mathcal{S}$ 是字母表 Σ 上的对角状态空间模型。对于 $\sigma \in \Sigma$，令 $f_\sigma\colon\ \mathbb{N} \to \mathbb{F}^d$ 为由 $f_\sigma(n) = \boldsymbol{y}_n$ 定义的函数，其中 $\boldsymbol{y}_n$ 是 $\mathcal{S}$ 在词 σ^n 上经过最后一层后的输出序列的最后一个向量。那么存在一个数 $N \in \mathbb{N}$，使得对于所有 $n \geq N\colon\ f_\sigma(n) = f_\sigma(N)$。

证明。 关键在于，由于门控矩阵是对角的，隐藏状态的每个维度都会独立演化。对于第一层，这意味着对于每个维度 i，$(\boldsymbol{h}_t)_i = \text{门}(\text{嵌入}(\sigma))_i \cdot (\boldsymbol{h}_{t-1})_i + \text{增量}(\text{嵌入}(\sigma))_i$。由于 门 是非负的，该序列必然是单调的。在定精度算术下，每个维度只能取有限个不同的值。因此，随着 t 增加，每个维度最终必然稳定为某个常数值。这意味着存在某个 N，使得对于所有 $n \geq N$，第一层的输出都保持不变。将同样的论证归纳应用于后续各层，可以得出结论：整个状态空间模型（SSM）的输出在有限次重复输入符号后会稳定下来。 □

以下结论直接源自引理 2，以及这样一个事实：$(aa)^*$ 并非单调的，即语言中存在这样的词，可以扩展为一个不属于该语言的词，反之亦然。

定理 3. 对角状 态空间模型在固定精度下可以识别 $(aa)^*$。

证明。 假设存在一个对角状态空间模型 $\mathcal{S}$，可以识别 $(aa)^*$。根据引理 2，存在一个数 N，使得对于所有 $n \geq N$：$f_a(n) = f_a(N)$。这意味着 $\mathcal{S}$ 要么接受、要么拒绝所有形如 a^n 且满足 $n \geq N$ 的词，与假设矛盾。 □

引理 2 表明，对角状态空间模型对每个输入符号的行为都是单调的。对于对角状态空间模型，尽管它们对每个输入符号都有单调的行为，但在看到不同符号时，仍然可以“重置”隐藏状态，从而计算 自从算子。

在刻画了固定精度下工作的对角状态空间模型的表达能力后，我们现在转而讨论对数精度下工作的对角状态空间模型。在这种设定下，对角状态空间模型也能够识别非正则、甚至非上下文无关语言。Sarrof 等人（2024）已经观察到这一点。此外，Alsmann 和 Lange（2025）证明了这类状态空间模型的可满足性问题是不可判定的。我们证明，对数精度下工作的对角状态空间模型能够识别 PLTL 中可定义的所有语言$_f$，并扩展了该逻辑以包含向后看的 计数算子 $\overleftarrow{\#}$。

定理 4。 具有 log精度的对角状态空间模型能够识别所有可由 $\text{PLTL}_f[\overleftarrow{\#}]$ 定义的语言.

证明概要。 详细证明见附录 A.2。该证明是定理 1 证明的扩展。给定一个计数子公式 $\sum_{j=1}^{k} a_j \cdot \overleftarrow{\#}\ \varphi_j \sim c$，我们在状态空间模型中增加一层，使用增量函数对隐藏状态中每个 φ_j 的出现次数进行计数。随后，逐位置前馈神经网络检查它们的线性组合是否满足与 c 的比较条件，并将结果写入隐藏状态中相应的维度。其余构造保持不变。由于该构造不需要任何非对角门，因此它同样适用于时不变对角 SSM。 □

5 时不变 SSM 的表达能力

如上一节所述，对角 SSM 能够计算 PLTL$_f$ 中的所有公式。然而，*since* 运算符本质上需要输入相关门。这引出了一个问题：具有常数门控的时不变 SSM 是否能够计算 *since* 运算符。我们推测它们不能，因此也无法识别所有可由 PLTL$_f$ 定义的语言。

Co猜想 1. 不存在能够在 $\Sigma = \{a, b, c\}$ 上识别 $L(a\mathsf{S}b)$ 的定精度时不变状态空间模型 .

这一猜想背后的直觉是，自从算子要求以一种依赖于当前输入的方式，将当前输入的信息与先前位置的信息结合起来。时不变状态空间模型在遇到不同符号时可以“重置”隐藏状态，但时不变模型无法根据当前输入符号调整其门 。由于这一限制，在词中看到 b 还是 c 的唯一区别，在于状态空间模型的加法增量部分。我们怀疑，试图识别 $a\mathsf{S}\ b$ 的时不变状态空间模型最终会遇到这样的情况：无法判断最后一个非 a 符号是 b 还是 c。当 n, m 较大时，$\sigma = a^n(ca^nba^n)^m$ 和 $\sigma' = a^n(ba^nca^n)^m$ 是这种情况可能发生的词的理想候选。不过，要形式化地证明这一猜想有两个难点。首先，时不变状态空间模型的行为比对角状态空间模型更复杂，因为隐藏状态的各个维度会彼此相互作用。其次，由于定精度算术中的饱和效应，以及定精度算术不满足结合律，时不变状态空间模型的行为很难分析。

由于自从算子是唯一需要输入相关对角门的时态算子，因此对于 SSM 层同时为对角且时不变的情况，我们可以立即得到以下结论。不含自从算子时，对角且时不变的状态空间模型能够识别 PLTL$_f$ 的一元片段中的语言。

推论 5。 同时具有对角和时不变特性的状态空间模型，能够在UN-PLTL$_f[\overleftarrow{\#}]$ 可定义的语言上，在log精度算术下识别所有此类语言；并且能够在定宽算术下识别UN-PLTL$_f$ 可定义的语言。

尽管其表达能力可能不如对角状态空间模型，但时不变且非对角的状态空间模型仍能识别无法用 PLTL$_f$定义的语言。我们通过展示时不变状态空间模型能够计算关于序列位置的模谓词来证明这一点。对于序列中的任意位置 t，模谓词决定了 t 除以某个模数 m后是否具有特定的余数。这种能力源于时不变状态空间模型能够使用循环置换矩阵作为门，在其隐藏状态中维护关于 m 的模计数器。

引理 6。 对于任意整数 $m \geq 2$，存在一个时不变状态空间模型层 l_{MOD^m}，当且仅当 $t \equiv r$（模 m）时，该层会在位置 t输出$\boldsymbol{e}_r \in \mathbb{R}^d$。

证明。 我们如下构造状态空间模型层 l_{MOD^m}。初始状态为 $\boldsymbol{h}_0 = \boldsymbol{e}_1$，即第一个标准基向量。门矩阵 P 是循环置换矩阵，它将 $\boldsymbol{e}_i$ 映射为 $\boldsymbol{e}_{i+1}$，其中 i =1、.....、$m-1$，并将 $\boldsymbol{e}_m$ 映射为 $\boldsymbol{e}_1$。具体而言， P的所有元素均为零，除了次对角线以及位置 (1，m) 上的元素为 1：

$$P = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}.$$

增量矩阵 B 被设为零，因为该谓词仅取决于位置，而不取决于输入。关键洞见在于，隐藏状态按照 $\boldsymbol{h}_t = P^t \boldsymbol{h}_0 = P^t \boldsymbol{e}_1$演化。自从 P是一个阶为 m的循环置换以来，我们有 $P^m = I$，因此有 $P^t \boldsymbol{e}_1 = \boldsymbol{e}_{(t \bmod m)+1}$。这意味着，在每个位置 t，隐藏状态恰好是其索引编码了 t除以 m所得余数的标准基向量。在位置 t，状态 $\boldsymbol{h}_t$在位置 (t 模 m) + 1 上为 1，在其他位置均为 0。 □

示例 1. 为了说明模谓词的概念，考虑 MOD_1^2，，它用于判断某个位置是否为奇数。此处 $m = 2$，因此置换矩阵为 $P = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$，初始状态为 $\boldsymbol{h}_0 = (1, 0)^T$。状态序列在奇数位置交替为 $\boldsymbol{e}_1 = (0, 1)^T$，在偶数位置交替为 $\boldsymbol{e}_0 = (1, 0)^T$。

循环置换实际上在隐藏状态中维护一个模 m 的计数器，从而能够通过单个时不变状态空间模型层计算任意模谓词。这尤其使得定义非星自由语言成为可能，例如 $L(\text{H}\, a \wedge \text{MOD}_0^2) = (aa)^*$，这类语言无法在 PLTL$_f$ 中定义，参见 Straubing（1994年）。从这个意义上说，对角且时不变的状态空间模型在表达能力上互不包含，因为对角状态空间模型无法识别 $(aa)^*$，而时不变状态空间模型似乎无法识别 $L(a\mathsf{S}\, b)$。

定理 7. 固定精度的时不变状态空间模型能够识别UN-PLTL$_f$[MOD] 中定义的所有语言；而对数精度的时不变状态空间模型能够识别UN-PLTL$_f$[MOD $\overleftarrow{\ }$#]中定义的所有语言。

证明概要。 详细证明见附录 A.3。它与定理 1 的证明类似。主要区别在于，我们需要为公式中出现的每个模谓词 MOD_r^m添加一层。该层按照引理 6 中所述的方法构造。其余构造保持不变。由于这一构造不需要任何非对角门，因此也适用于时不变对角 SSM。

□

在为对角状态空间模型和时不变状态空间模型建立下界后，自然会想到考察每层要么为对角、要么为时不变的状态空间模型。混合状态空间模型既能计算*since*算子，也能计算模谓词。这使我们能够证明，固定精度的混合状态空间模型可以识别 PLTL$_f$[MOD] 中的所有语言；这些语言恰好是 AC^0，中的正则语言（参见 Straubing（1994年）），也就是由具有无界扇入的常数深度多项式规模电路识别的语言类。该类既包含所有无星语言，也包含诸如 $(aa)^*$ 这样的语言。对数精度的混合状态空间模型与 PLTL$_f$[MOD $\overleftarrow{\ }$#] 也同样成立。

推论 8。 采用固定精度的混合状态空间模型可识别 AC^0 中的所有正则语言。采用 对数精度时，它们可识别 PLTL$_f$[$\overleftarrow{\ }$#、MOD] 中的所有正则语言。

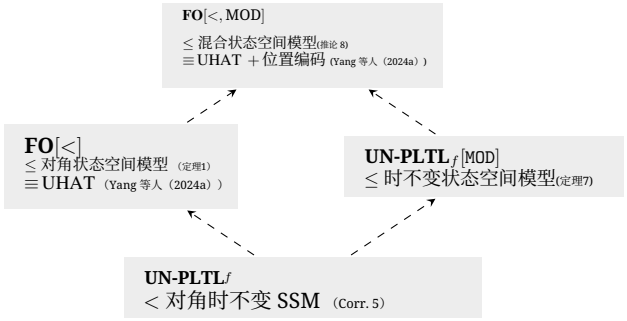

图 2：固定精度状态空间模型变体与独特硬注意力 Transformer（UHAT）Yang 等人（2024a）的详细比较。该图展示了结构上的对应关系：对角状态空间模型能够刻画无星语言（FO[<]），其能力与不使用位置编码的 UHAT 相当。时不变层引入模谓词，带来表达能力上的提升，类似于为 Transformer 添加位置编码。因此，混合状态空间模型能够刻画完整的 FO[<,MOD] 类（AC^0 中的正则语言）。

6 与 Transformer 架构的比较

本节将讨论有关状态空间模型表达能力的研究结果如何融入近期关于 SSM 表达能力以及各种 Transformer 架构比较表达能力的更广泛研究领域。

形式表达能力领域研究的 Transformer 架构可分为两大类：采用硬注意力机制的架构和采用软注意力机制的架构。硬注意力变换器（例如，唯一硬注意力（UHAT）和平均硬注意力（AHAT））使用离散注意力机制，使其能够聚焦于输入序列的特定部分。相比之下，软注意力 Transformer（SAT），如 Vaswani 等人（2017）提出的原始 Transformer 模型及其变体，采用连续注意力机制，对整个输入序列计算加权平均。唯一硬注意力变换器只能关注序列中的一个位置，而平均硬注意力和软注意力 Transformer 可以关注多个位置，从而能够统计某些属性出现的次数。这种差异对应于 SSM 中固定精度算术与对数精度算术之间的差异。

此外，由于注意力机制的特性，若不添加额外的位置编码，Transformer 无法区分序列中的不同位置。这些编码为模型提供序列中元素的顺序和位置信息。相比之下，SSM 通过其循环结构天然地编码位置信息。然而，正如第 5 节所观察到的，允许使用时不变状态空间模型层能够提升表达能力，因为模型还可以在其隐藏状态中维护一个模 m计数器，从而能够计算关于序列位置的模谓词。与允许使用时不变状态空间模型层的情况类似，为 Transformer 添加位置编码也使其能够计算模谓词，参见 Yang 等人（2024a）和 Barcelo 等人（2024）。

Yang 等人（2024a）证明，唯一硬注意力变换器（UHAT）能够识别所有可由带序关系谓词的一阶逻辑（FO[<]）定义的语言；加入位置编码（UHAT+PE）后，它们还能够识别所有可由 FO[<,MOD] 定义的语言。这与我们关于固定精度下对角且时不变的状态空间模型的结果一致，如图 2 所示。类似地，Barcelo 等人（2024）表明，平均硬注意力变换器（AHAT）能够识别所有可由 $\mathrm{PLTL}_f[\overleftarrow{\#}\ \overrightarrow{\#}, \mathrm{MOD}]$ 定义的语言。这与我们关于对数精度下混合状态空间模型的结果一致。AHAT 能力的提升，源于 Transformer 能够关注输入序列中的所有位置。Yang 和 Chiang（2024）通过分析一种名为 C-RASP 的逻辑及其扩展 C-RASP[MOD]，研究了软注意力变换器（SAT）的计数能力。C-RASP 对应于 $\text{UN-PLTL}_f[\overleftarrow{\#}]$，的一个真子集，这一点我们在附录 A.4 中证明了。图 3 展示了我们的结果如何纳入现有研究脉络。

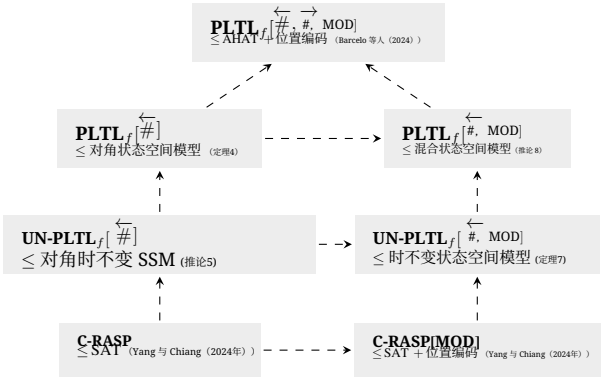

图 3：对数精度下的比较层级，展现了计数能力。该图将状态空间模型变体与软注意力（SAT）（Yang 和 Chiang，2024年）以及平均硬注意力 Transformer（AHAT）（Barcelo 等人，2024年）进行比较。尽管状态空间模型凭借更强的局部时序处理能力（例如 昨天算子）严格包含 SAT（C-RASP）的逻辑，但其表达能力仍严格弱于 AHAT+位置编码。这一区别源于因果性：状态空间模型只能进行向后计数（$\overleftarrow{\#}$），而 AHAT 的全局注意力则支持向前计数（$\overrightarrow{\#}$）。

7 展望

我们通过展示各种状态空间模型架构识别由时序逻辑不同片段定义的语言的能力，得出了其表达能力的下界。图 1 可视化了这些结果，全面呈现了状态空间模型变体之间的表达能力层级。尽管我们的分析依赖于显式的权重构造，但由此得到的推论对训练方法具有稳健性。具体而言，我们的分离结果确立了架构层面的严格限制：例如，单调性引理意味着，无论优化器或数据集大小如何，固定精度的对角状态空间模型都无法通过训练识别 $(aa)^*$ 这类非星自由语言。此外，这些结果还揭示了本文所确立的下界与 Merrill 等人（2024）已知的 TC^0 上界之间一个有趣的差距。本文证明可由采用定宽算术的各种受限状态空间模型架构识别的语言类，属于 AC^0，，而 AC^0，是 TC^0 的真子集。这意味着两种可能：要么可以将这些状态空间模型架构的 TC^0 上界收紧为 AC^0，要么这些状态空间模型实际上能够识别 AC^0，之外的语言，例如奇偶性。第一种可能似乎更合理，因为我们尚未发现这些状态空间模型架构中存在任何机制，能够实现对某个常数取模的计数——而识别奇偶性及其他 AC^0 之外的语言都需要这种能力。Sarrof 等人（2024）甚至证明，对角状态空间模型无法表达奇偶性，但这一结论仅针对特定的非线性层输出函数。如果确实如此，那么状态空间模型就与唯一硬注意力变换器相符；后者也已被证明受限于 AC^0，（参见 Hao 等人，2022年）。

此外，研究基于公式嵌套深度的潜在表达能力层级，可能有助于进行更细粒度的复杂性分类。对于唯一硬注意力（参见 Yang 等人（2024a））和软注意力 Transformer（参见 Yang 等人（2025）），类似的表达能力层级已经得到证明。

参考文献

IEEE 浮点算术标准。*IEEE Std 754-2019* 标准（*IEEE 754-2008* 标准的修订版），第 1–84 页，2019年。数字对象标识符：10.1109/IEEESTD.2019.8766229。

Eric Alsmann 和 Martin Lange。状态空间模型中可满足性的计算复杂性。载于欧洲人工智能会议，2025。

Raman Arora、Amitabh Basu、Poorya Mianjy 和 Anirbit Mukherjee。使用修正线性单元理解深度神经网络。载于国际学习表征会议，2018年2月。Pablo Barcelo、Alexander Kozachinskiy、Anthony Widjaja Lin 和 Vladimir Podolskii。硬注意力 Transformer 编码器所接受的逻辑语言。载于第十二届国际学习表征会议，2024年。Soham De、Samuel L. Smith、Anushan Fernando、Aleksandar Botev、George Cristian-Muraru、Albert Gu、Ruba Haroun、Leonard Berrada、Yutian Chen、Srivatsan Srinivasan、Guillaume Desjardins、Arnaud Doucet、David Budden、Yee Whye Teh、Razvan Pascanu、Nando De Freitas 和 Caglar Gulcehre。Griffin：将门控线性递归与局部注意力相结合，打造高效语言模型，2024年2月。

Giuseppe De Giacomo 和 Moshe Y. Vardi。有限轨迹上的线性时序逻辑与线性动态逻辑。载于第二十三届国际人工智能联合会议论文集，IJCAI '13，第 854–860 页，北京，中国，2013年8月。AAAI出版社。ISBN 978-1- 57735-633-2。

Giuseppe De Giacomo、Antonio Di Stasio、Francesco Fuggitti 和 Sasha Rubin。有限轨迹上的纯过去线性时序与动态逻辑。载于第二十九届国际人工智能联合会议论文集，IJCAI'20，第 4959–4965 页，横滨，横滨，日本，2021 年 1 月。ISBN 978-0-9992411-6-5。

Kousha Etessami、Moshe Y. Vardi 和 Thomas Wilke。带有两个变量和一元时序逻辑的一阶逻辑。《信息与计算》，179(2):279–295，2002年。ISSN 0890-5401。doi: 10.1006/inco.2001.2953。

Michael Fisher、Dov M. Gabbay 和 L. Vila（编）。《人工智能时序推理手册》。人工智能基础丛书第 v. 1 卷。爱思唯尔，阿姆斯特丹、波士顿，第 1 版，2005年。ISBN 978-0-08-053336-0 978-0-444-51493-6。

Merrick Furst、James B. Saxe 和 Michael Sipser。奇偶性、电路与多项式时间层级。《数学系统理论》，17(1):13–27，1984年12月。ISSN 1433-0490。doi: 10.1007/BF01744431。

Albert Gu 和 Tri Dao。Mamba：基于选择性状态空间的线性时间序列建模。发表于第一届语言建模会议，2024年8月。

Albert Gu、Karan Goel 和 Christopher Re。利用结构化状态空间高效建模长序列。发表于国际学习表征会议，2022年。

Yiding Hao、Dana Angluin 和 Robert Frank。基于硬注意力 Transformer 的形式语言识别：来自电路复杂度的视角。计算语言学协会汇刊，10：800–810，2022年7月。国际标准连续出版物号 2307-387X。数字对象标识符：10.1162/tacl a 00490。

Ramin Hasani、Mathias Lechner、Tsun-Hsuan Wang、Makram Chahine、Alexander Amini 和 Daniela Rus。液态结构化状态空间模型。载于第十一届国际学习表征会议，2022年9月。

Johan Anthony Willem Kamp。时态逻辑与线性序理论。加利福尼亚大学，1968年。

Harsh Mehta、Ankit Gupta、Ashok Cutkosky 和 Behnam Neyshabur。基于门控状态空间的长程语言模型。在第十一届国际学习表征会议，2023年。

William Merrill 和 Ashish Sabharwal。《并行性权衡：对数精度 Transformer 的局限性》。计算语言学协会会刊，11：531–545，2023年6月。国际标准连续出版物号 2307-387X。数字对象标识符：10.1162/tacl a 00562。

William Merrill、Jackson Petty 和 Ashish Sabharwal。《状态空间模型中的状态幻觉》。载于第*41*届国际机器学习会议论文集，第235卷，*ICML'24*，第35492–35506页，奥地利维也纳，2024年7月。JMLR.org。

Antonio Orvieto、Samuel L Smith、Albert Gu、Anushan Fernando、Caglar Gulcehre、Razvan Pas- canu 和 Soham De。《为长序列复兴循环神经网络》。载于第*40*届国际机器学习大会论文集，ICML'23，美国夏威夷檀香山，2023年。JMLR.org。

Yash Sarrof、Yana Veitsman 和 Michael Hahn。状态空间模型的表达能力：形式语言视角。载于第三十八届神经信息处理系统年会，2024年11月。

Howard Straubing。有限自动机、形式逻辑与电路复杂度。Birkh¨auser，波士顿，马萨诸塞州，1994年。ISBN 978-1-4612-6695-2 978-1-4612-0289-9。数字对象标识符：10.1007/978-1-4612-0289-9。

Lena Strobl、William Merrill、Gail Weiss、David Chiang 和 Dana Angluin。Transformer 能表达哪些形式语言？一项综述。计算语言学协会会刊，12：543–561，2024年5月。国际标准连续出版物号 2307-387X。数字对象标识符：10.1162/tacl a 00663。 – –

Yutao Sun、Li Dong、Shaohan Huang、Shuming Ma、Yuqing Xia、Jilong Xue、Jianyong Wang 和 Furu Wei。《 Retentive Network：面向大语言模型的 Transformer 后继者》，2023 年 8 月。

Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N. Gomez、Łukasz Kaiser 和 Illia Polosukhin。《注意力就是你所需要的一切》。载于第三十一届神经信息处理系统国际会议论文集，NIPS'17，第 6000–6010 页，雷德胡克，纽约州，美国，2017年12月。Curran Associates Inc. ISBN 978-1-5108-6096-4。

Andy Yang 和 David Chiang。《像Transformer一样计数：将时序计数逻辑编译为 Softmax Transformer》。载于第一届语言建模会议，2024年。

Andy Yang、David Chiang 和 Dana Angluin。《掩码硬注意力变换器恰好识别无星语言》。载于第三十八届神经信息处理系统年会，2024a。

AndyYang、Micha¨elCadilhac 和 David Chiang。《深入探究 C-RASP：一种 Transformer 层次结构，
2025。

Songlin Yang、Bailin Wang、Yikang Shen、Rameswar Panda 和 Yoon Kim。采用硬件高效训练的门控线性注意力Transformer。载于第*41*届国际机器学习会议论文集，ICML'24，奥地利维也纳，2024b。JMLR.org。

Nikola Zubic、Federico Sold`a、Aurelio Sulser 和 Davide Scaramuzza。深度学习的局限：从复杂性理论的视角进行序列建模。载于第十三届国际学习表征会议，2025。

A 附录

A.1 PLTL$_f$ 及其扩展/片段的表达能力

Kamp定理 Kamp（1968年）结合 De Giacomo & Vardi（2013）；De Giacomo 等人（2021年）的结果，证明 PLTL$_f$ 与词上的一阶逻辑 FO[<]具有相同的表达能力，而后者恰好对应无星表达式类。同样，逻辑 PLTL$_f$[MOD] 与带模谓词的一阶逻辑 FO[$<$, MOD], 具有相同的表达能力，恰好刻画了 AC^0 中包含的正则语言（Straubing，1994年）。一元片段 UN-PLTL$_f$ 是二变量片段一阶逻辑 $FO^2[<]$ 可定义语言的子集（Etessami 等人，2002年）。UN-PLTL$_f$ 仅限于一元时序算子昨天 和先前，而 $FO^2[<]$则完整刻画了可由 LTL$_f$ 使用一元算子昨天、先前、下一步和将来某时定义的语言。可以证明，UN-PLTL$_f$必定是 $FO^2[<]$ 的真子集，因为将时态公式分解为过去与未来部分的标准构造（Fisher 等人，2005年）依赖于自从 算子。这种关系也适用于模谓词扩展，其中 UN-PLTL$_f$[MOD] 是 $FO^2[<, \text{MOD}]$的真子集。关于 PLTL$_f$的计数扩展，我们观察到，将其限制为计数子公式

且仅包含正权重，并不会增强表达能力。然而，引入负权重后，便能够表示可能不属于 AC^0的非正则语言。例如，考虑语言 $\{w \in \{a,b\}^* \mid |w|_a = |w|_b\}$，它由 a 和 b出现次数相同的字符串组成。该语言不属于 AC^0（Furst 等人（1984年）），但甚至可以用 UN-PLTL $_f[\overleftarrow{}\#]$ 中的简单公式 $\overleftarrow{}\#\ a - \overleftarrow{}\#\ b = 0$简洁地表达。

A.2 第 4 节的证明

前馈神经网络。一个（前馈神经网络）节点是一个函数 $v\colon \mathbb{R}^k \to \mathbb{R}$，满足 $v(\boldsymbol{x}) = \mathrm{relu}(\sum_{i=1}^{k} c_i x_i + b)$，其中 k 是输入维度，$c_i \in \mathbb{R}$ 称为权重，$b \in \mathbb{R}$ 是偏置，而 $\mathrm{relu}: \mathbb{R} \to \mathbb{R}$ 满足 $\mathrm{relu}(x) = \max(0, x)$，它是 v的激活函数。一个（前馈神经网络）层l 是由若干 n 节点组成的元组 $(v_1, \ldots, v_n)$，其中每个节点具有相同的输入维度 m。它通过 $l(\boldsymbol{x}) = (v_1(\boldsymbol{x}) \ldots, v_n(\boldsymbol{x}))$计算函数 $l\colon \mathbb{R}^m \to \mathbb{R}^n$。我们称 m 为 l的输入维度，称 n为其输出维度。一个前馈神经网络（FNN）N由 k层 $l_1, \ldots, l_k$ 组成，其中 l_1 的输入维度为 m，对于 $i < k$，l_i 的输出维度等于 l_{i+1}的输入维度，而 l_k 的输出维度为 n。FNN N 通过$N(\boldsymbol{x}) = l_k(l_{k-1}(\ldots l_1(\boldsymbol{x}) \ldots))$计算一个从 $\mathbb{R}^m$ 到 $\mathbb{R}^n$ 的函数。

对于下界构造，有时我们需要使用前馈神经网络来检查特定性质，例如：某个向量分量是否等于特定值？为了便于记号，我们将定义用于检查后续所需特定性质的前馈神经网络组件。

引理 9。设 $n, n_1, \cdots, n_k, m, b \in \mathbb{Z}$。存在前馈神经网络

- $N_{\sim b}$ *s.t.* $N_{\sim b}(n) = 1$ *if* $n \sim b$ *and* $N_{\sim b}(n) = 0$ *otherwise, for all* $\sim \in \{<, \leq, =, \geq, >\}$.
- $N_{\min(0,x)}$ *s.t.* $N_{\min(0,x)}(n) = \min(0, x)$.

证明。设 $N_{=b}$ 为计算 $N_{=b}(x) = \mathrm{relu}\,(\mathrm{relu}(x - (b-1)) - 2 \cdot \mathrm{relu}(x - b))$ 的前馈神经网络，$N_{\leq b}$为计算 $N_{\leq b}(x) = \mathrm{relu}\,(\mathrm{relu}(b + 1 - x) - \mathrm{relu}(b - x))$ 的前馈神经网络，$N_{\geq b}$ 为计算 $N_{\geq b}(x) = \mathrm{relu}\,(\mathrm{relu}(x - (b-1)) - \mathrm{relu}(x - b))$ 的前馈神经网络，$N_{\min(1,x)}$ 为计算$N_{\min(1,x)}(x) = \mathrm{relu}(x) - \mathrm{relu}(-x) - \mathrm{relu}(x - 1)$ 的前馈神经网络。由于我们只处理整数，可以将 $x < b$ 改写为 $x \leq b - 1$，并将 $x > b$ 改写为 $x \geq b + 1$。显然，给定的前馈神经网络计算了所需的函数。□

有时我们需要将前馈神经网络组合起来。设 N_1 和 N_2 为计算函数 $\mathbb{R}^{m_i} \to \mathbb{R}^{n_i}$（$i \in \{1, 2\}$）的前馈神经网络。$N_1$ 和 N_2 的复合定义如下。当 $n_2 = m_1$时，前馈神经网络 $N_1 \circ N_2$ 计算函数 $\mathbb{R}^{m_2} \to \mathbb{R}^{n_1}$，其中 $N_1 \circ N_2(\boldsymbol{x}) = N_1(N_2(\boldsymbol{x}))$。从语法上看，复合只需将 N_2 的输出层与 N_1 的输入层连接起来，并将所有权重设为 1。

为简化记号，我们使用线性投影来定义对角状态空间模型层的门函数和增量函数。对于固定矩阵 A, B，令 $门(\boldsymbol{x}_t) = \mathrm{diag}(A\boldsymbol{x}_t)$，并令 $增量函数(\boldsymbol{x}_t) = B\boldsymbol{x}_t$。由此，对角状态空间模型层递推可简化为

$$\boldsymbol{h}_t = (A\boldsymbol{x}_t) \circ \boldsymbol{h}_{t-1} + B\boldsymbol{x}_t, \tag{1}$$

其中 $\circ$ 表示逐元素乘法。我们利用 PLTL$_f$ 公式的序贯分解，构造一个状态空间模型，按照嵌套深度递增的顺序逐层计算子公式。

定义 2（序列分解）**.** 设 φ 是一个满足 $\mathrm{nd}(\varphi) = n$ 的 LTL$_f$ 公式。φ 的**序列分解** 是唯一的集合序列 $M_0, \ldots, M_n$，满足 $M_i = \{\psi \in \mathrm{Sub}(\varphi) \mid \mathrm{nd}(\psi) = i\}$。

首先，我们分别构造每个时态算子的翻译，然后通过对嵌套深度进行归纳，给出完整的构造。给定一个关于命题$\mathcal{P} = \{a_1, \ldots, a_k\}$的 PLTL$_f[\overleftarrow{}\#]$ 公式 φ，我们固定一个单射函数 $\iota\colon \mathcal{P} \cup \mathrm{Sub}(\varphi) \cup \{1\} \to \{1, \ldots, |\mathcal{P}| + |\varphi| + 1\}$，将每个命题 $p = a_i$ 映射到索引 i。其余子公式的赋值任意。这样的排序为每个子公式指定一个固定索引，该索引对应于所得状态空间模型计算出的向量中的位置，其维度为 $d = |\mathcal{P}| + |\varphi| + 1$。每个向量的最后一个维度设为常数

1. 为了便于定义线性投影，我们引入复制矩阵 $C^{(i\leftarrow j)} \in \mathbb{R}^{d\times d}$，其定义为$C^{(i\leftarrow j)} = \boldsymbol{e}_i \boldsymbol{e}_j^T$，其中索引为 $1 \le i, j \le d$。将向量 $\boldsymbol{x} \in \mathbb{R}^d$ 乘以 $C^{(i\leftarrow j)}$，得到一个向量，其第 j 个分量为 x_i，其余分量均为零。

定理 1 和定理 4 是以下更一般结果的直接推论。我们将在本节末证明该结果。

定理 10。 设 φ 是关于命题 $\mathcal{P}$ 的 $\mathrm{PLTL}_f[\overleftarrow{\#}]$ 公式。则存在一个字母表为 $\Sigma = 2^{\mathcal{P}}$、维度为 $d = |\mathcal{P}| + |\varphi| + 1$ 的 *SSM* $\mathcal{S}_\varphi$，其恰好具有 $nd(\varphi)$ 层，并满足以下条件：对于任意满足 $|\sigma| = n$ 的词 $\sigma \in \Sigma^*$，如果 $\boldsymbol{y}_1 \cdots \boldsymbol{y}_n$ 是 $\mathcal{S}_\varphi$ 在输入 σ 经过最终层后产生的输出，则对于所有位置 $1 \le t \le n$ 和子公式 $\psi \in Sub(\varphi)$，当且仅当 σ、$t \models \psi$ 时有 $(\boldsymbol{y}_t)_{\iota(\psi)} = 1$，否则为 $(\boldsymbol{y}_t)_{\iota(\psi)} = 0$。

为证明该定理，我们首先需要一些中间结果。具体而言，求值涉及 昨天-算子的公式，需要通过门控机制将前一步的信息适当地编码到当前隐藏状态中。该编码最初由 Sarrof 等人（2024）提出；我们针对当前语境给出了一个自洽的证明。

引理 11 (Sarrof 等人（2024年）)。 设 σ 是一个二进制序列 $\sigma = a_1 a_2 \cdots a_n \in \{0,1\}^*$。对于每个 $1 \le t \le n$，我们递归地定义数值 $h_t \in \mathbb{R}$，使其满足 $h_0 = 0$ 和 $h_t = \frac{1}{4} h_{t-1} + a_t$。存在一个前馈神经网络 N_{prev}，对于所有 $t \ge 2$，它在位置 t 输出数值 a_{t-1}，并在位置 1输出0 。

证明。 对于 $t \ge 1$，隐藏状态按照 $h_t = \frac{1}{4} h_{t-1} + a_t$ 演化。展开该递推关系，可得 $h_t = a_t + \frac{1}{4} a_{t-1} + \frac{1}{16} a_{t-2} + \frac{1}{64} a_{t-3} + \cdots$。关键在于，当 $a_i \in \{0,1\}$时，数值 h_t 的二进制表示为 $a_t.0 a_{t-1} 0 a_{t-2} 0 \cdots a_{1_2}$。这种编码将当前输入 a_t 置于整数部分，而先前的输入 a_{t-1}、a_{t-2}、.....则占据小数部分中交替的二进制位，彼此之间以零隔开。

要理解这为何会导致不相交区间，请注意，位置 $t-2$ 及更早位置的所有位所贡献的值都有界：无穷级数 $\frac{1}{16} a_{t-2} + \frac{1}{64} a_{t-3} + \cdots$ 在所有 $a_i = 1$时至多为$\frac{1}{16} + \frac{1}{64} + \frac{1}{256} + \cdots = \frac{1/16}{1-1/4} = \frac{1}{12}$，在所有 $a_i = 0$时至少为 0。因此，“尾部”的贡献有界于区间 $[0, \frac{1}{12}]$内。

根据这个界，我们可以基于 $(a_t,\ a_{t-1})$ 分析四种情况：

- 对于 $(a_t,\ a_{t-1}) = (0,\ 0)$：我们有 $h_t = 0 + 0 +$尾部，因此 $h_t \in [0, \frac{1}{12}]$。
- 对于 $(a_t,\ a_{t-1}) = (0,\ 1)$：我们有 $h_t = 0 + \frac{1}{4} +$尾部，因此 $h_t \in [\frac{1}{4}, \frac{1}{4} + \frac{1}{12}] = [\frac{1}{4}, \frac{1}{3}]$。
- 对于 $(a_t,\ a_{t-1}) = (1,\ 0)$：我们有 $h_t = 1 + 0 +$尾部，因此 $h_t \in [1, 1 + \frac{1}{12}] = [1, \frac{13}{12}]$。
- 对于 $(a_t,\ a_{t-1}) = (1,\ 1)$：我们有 $h_t = 1 + \frac{1}{4} +$尾部，因此 $h_t \in [\frac{5}{4}, \frac{5}{4} + \frac{1}{12}] = [\frac{5}{4}, \frac{4}{3}]$。

这些区间显然互不相交。在有限精度运算的实际实现中，我们可以使用稍窄一些且具有精确二进制表示的区间。由于所有边界都可以用小数点后最多 3 位精确表示，因此可以安全地使用：$(0,\ 0) \mapsto [0, \frac{1}{8}]$、$(0,\ 1) \mapsto [\frac{1}{4}, \frac{1}{2}]$、$(1,\ 0) \mapsto [1, \frac{9}{8}]$ 和 $(1,\ 1) \mapsto [\frac{5}{4}, \frac{3}{2}]$。这些细化后的区间既保持了互不重叠的性质，也为舍入误差提供了额外的安全裕度。

N_{prev} 通过将包含 $(a_t, a_{t-1}) \in \{(0,0), (1,0)\}$ 的区间映射为 0，并将包含 $(a_t, a_{t-1}) \in \{(0,1), (1,1)\}$ 的区间映射为 1，从而提取出 a_{t-1}。利用实现阶跃函数的标准技术，这个分段线性函数可以由带 ReLU 激活函数的前馈神经网络精确表示（Arora 等人（2018））。对于 $t = 1$，我们有 $h_1 = a_1$，并且根据构造，$N_{\text{prev}}(a_1) = 0$。对于 $t \ge 2$，包含 h_t 的区间唯一确定了 a_{t-1}，而 N_{prev}可以正确地提取它。因子 1_4（而非 1_2）至关重要，因为它在编码比特之间提供了足够的间隔，从而避免有限精度运算下的数值误差。 □

在说明了如何从之前的位置恢复信息以计算*yester- day* 公式之后，我们现在将展示如何构造能够计算一般 $\mathrm{PLTL}_f[\overleftarrow{\#}]$ 公式的层。

引理 *12.* 设 $\sigma = \sigma_1 \cdots \sigma_n \in \Sigma^*$ 是一个词，$\varphi_1, \ldots, \varphi_k$ 是 $PLTL_f[\overleftarrow{\#}]$ 公式，$\boldsymbol{x}_1 \cdots \boldsymbol{x}_n$ 是一个序列，且满足性质 $(\boldsymbol{x}_t)_{\iota(\varphi_i)} = 1 \iff \sigma, t \models \varphi_i$ 对所有 $1 \le t \le n$ 和 $1 \le i \le k$ 均成立。那么，对所有 $\psi \in \{\neg\varphi_1, \varphi_1 \wedge \varphi_2, \varphi_1 \quad \varphi_2 \quad \varphi_1 \quad \varphi_1 \sum_j a_j \cdot \boldsymbol{S}, \boldsymbol{Y}, \boldsymbol{P}, \overleftarrow{\#}\ \varphi_j \sim c\}$，都存在对角门控状态空间模型层，使得对于层输出 $\boldsymbol{z}_1 \cdots \boldsymbol{z}_n$，我们有 $(\boldsymbol{z}_t)_{\iota(\psi)} = 1 \iff \sigma, t \models \psi$，并且 $(\boldsymbol{z}_t)^j = (\boldsymbol{x}_t)^j$ 对所有 $j \ne \iota(\psi)$ 和 $1 \le t \le n$ 均成立。

证明。 我们将说明如何为每个公式单独构造各层。所有层均设为 $\boldsymbol{h}_0 = \boldsymbol{0}$。定义仅影响输入中单个维度的前馈神经网络时，我们总是假设其他所有输入保持不变。

情况 1： $\neg\varphi_1$。 注意，为了计算否定，我们无需通过门从之前的位置传递任何信息。利用增量函数，可以在每个位置静态地计算该公式。我们定义

$$A = 0 \text{ and } B = I + C^{(\iota(\neg\varphi) \leftarrow \iota(1))} - C^{(\iota(\neg\varphi_1) \leftarrow \iota(\varphi_i))}.$$

按照公式 1 中定义的线性递推进行计算后，我们得到一个序列 $\boldsymbol{h}_1 \cdots \boldsymbol{h}_n$，满足 $(\boldsymbol{h}_t)_{\iota(\neg\varphi_1)} = 1 - (\boldsymbol{x}_t)_{\iota(\varphi_1)}$。由于这已经满足所需结论，我们可以将这一层的非线性激活函数 ϕ 设为恒等函数 $\phi(\boldsymbol{x}, \boldsymbol{h}) = \boldsymbol{h}$。

情况 2： $\varphi_1 \wedge \varphi_2$。 令

$$A = 0 \text{ and } B = I + C^{(\iota(\varphi_1 \wedge \varphi_2) \leftarrow \iota(\varphi_1))} + C^{(\iota(\varphi_1 \wedge \varphi_2) \leftarrow \iota(\varphi_2))} - C^{(\iota(\varphi_1 \wedge \varphi_2) \leftarrow \iota(1))}$$

按照公式 1 中定义的线性递推进行计算后，我们得到一个序列 $\boldsymbol{h}_1 \cdots \boldsymbol{h}_n$，满足 $(\boldsymbol{h}_t)_{\iota(\varphi_1 \wedge \varphi_2)} = (\boldsymbol{x}_t)_{\iota(\varphi_1)} + (\boldsymbol{x}_t)_{\iota(\varphi_2)} - 1 \in \{-1, 0, 1\}$。非线性层输出 ϕ 现在将该序列映射为 $\boldsymbol{z}_1 \cdots \boldsymbol{z}_n$，使得 $(\boldsymbol{z}_t)_{\iota(\varphi_1 \wedge \varphi_2)} = N_{\ge 1}((\boldsymbol{h}_t)_{\iota(\varphi_1 \wedge \varphi_2)})$。由此得到所需性质。

Cas情形 3： $\mathrm{P}\,\varphi_1$。 在这种情况下，我们需要使用一个门，将有关先前 s posi满足 φ_1 的位置的信息传递到当前位置。

$$A = C^{\iota(\mathrm{P}\,\varphi_1) \leftarrow \iota(1)} \text{ and } B = I + C^{(\iota(\mathrm{P}\,\varphi_1) \leftarrow \iota(\varphi_1))}$$

请注意，该门实际上是时不变且对角的，因为信息传递不依赖于当前输入。该门只会传播先前的值，因此

$$(\boldsymbol{h}_t)_{\iota(\mathrm{P}\,\varphi_1)} = (\boldsymbol{h}_{t-1})_{\iota(\mathrm{P}\,\varphi_1)} + (\boldsymbol{x}_t)_{\iota(\varphi_1)} = \sum_{i=1}^{t} (\boldsymbol{x}_i)_{\iota(\varphi_1)}.$$

因此，$(\boldsymbol{h}_{t-1})_{\iota(\mathrm{P}\,\varphi_1)}$ 保存了 $\mathrm{P}\ \varphi_1$ 在先前位置为真的次数。非线性层输出 ϕ 现在将此序列映射到 $\boldsymbol{z}_1 \cdots \boldsymbol{z}_n$，使得 $(\boldsymbol{z}_t)_{\iota(\mathrm{P}\,\varphi_1)} = N_{\ge 1}((\boldsymbol{h}_t)_{\iota(\mathrm{P}\,\varphi_1)})$。

情形 4： $\sum_j a_j \cdot \overleftarrow{\#}\ \varphi_j \sim c$。 为了采用更紧凑的记法，我们用 ψ 表示该公式。这种情形基本上是对先前情形的扩展。我们不再仅仅检验公式成立的位置数是否至少为一个，而是计算出现次数的加权和，并将其与特定值进行比较。

$$A = C^{(\iota(\psi) \leftarrow \iota(1))} \text{ and } B = I + \sum_{j=1}^{k} a_j \cdot C^{(\iota(\psi) \leftarrow \iota(\varphi_j))}$$

与前一种情况一样，这个门也是时不变且对角的，因为它只独立传播特定维度的值，而不受 $\boldsymbol{x}_t$ 影响。我们得到

$$(\boldsymbol{h}_t)_{\iota(\psi)} = (\boldsymbol{h}_{t-1})_{\iota(\psi)} + \sum_{j=1}^{k} a_j \cdot (\boldsymbol{x}_t)_{\iota(\varphi_j)} = \sum_{i=1}^{t} \sum_{j=1}^{k} a_j \cdot (\boldsymbol{x}_i)_{\iota(\varphi_j)} = \sum_{j=1}^{k} a_j \cdot \underbrace{\sum_{i=1}^{t} (\boldsymbol{x}_i)_{\iota(\varphi_j)}}_{\overleftarrow{\#}\ \varphi_j}.$$

非线性层输出 ϕ 现在将该序列映射到 $\boldsymbol{z}_1 \cdots \boldsymbol{z}_n$，使得 $(\boldsymbol{z}_t)_{\iota(\psi)} = N_{\sim c}((\boldsymbol{h}_t)_{\iota(\psi)})$。

情况 5: $\varphi_1 \mathsf{S} \varphi_2$**.** 我们使用 自从-算子的时序展开： $\varphi_1 \mathsf{S} \varphi_2 \equiv (\varphi_1 \wedge \mathsf{Y}(\varphi_1 \mathsf{S} \varphi_2)) \vee \varphi_2$。仔细观察可以发现，该展开与 SSM 层中的线性递推具有相似的结构。 φ_1 的当前求值会与 $\varphi_1 \mathsf{S} \varphi_2$ 的上一次求值相乘，此外还会加上 φ_2 的求值，以便在再次出现 φ_2时重新启动递推。我们定义

$$A = C^{(\iota(\varphi_1 \mathsf{S} \varphi_2) \leftarrow \iota(\varphi_1))} \text{ and } B = I + C^{(\iota(\varphi_1 \mathsf{S} \varphi_2) \leftarrow \iota(\varphi_2))}$$

由此得到 $(\boldsymbol{h}_t)_{\iota(\varphi_1 \mathsf{S} \varphi_2)} = (\boldsymbol{x}_t)_{\iota(\varphi_1)} \cdot (\boldsymbol{h}_{t-1})_{\iota(\varphi_1 \mathsf{S} \varphi_2)} + (\boldsymbol{x}_t)_{\iota(\varphi_2)}$。我们看到，在这种情况下，门不是时不变的，因为它实际上取决于 $(\boldsymbol{x}_t)_{\iota(\varphi_1)}$ 的值。显然， $\varphi_1 \mathsf{S} \varphi_2$ 在某个位置 t 满足，当且仅当 $(\boldsymbol{h}_t)_{\iota(\varphi_1 \mathsf{S} \varphi_2)} > 0$。非线性层输出 ϕ 现在将该序列映射为 $\boldsymbol{z}_1 \cdots \boldsymbol{z}_n$，使得 $(\boldsymbol{z}_t)_{\iota(\varphi_1 \mathsf{S} \varphi_2)} = N_{\min(1,x)}((\boldsymbol{h}_t)_{\iota(\varphi_1 \mathsf{S} \varphi_2)})$。

情况 6: $\mathsf{Y} \varphi_1$。我们使用引理 11 中的编码技巧。令

$$A = \frac{1}{4} C^{(\iota(\mathsf{Y} \varphi_1) \leftarrow \iota(1))} \text{ and } B = I + C^{(\iota(\mathsf{Y} \varphi_1) \leftarrow \iota(\varphi_1))}$$

我们得到 $(\boldsymbol{h}_t)_{\iota(\mathsf{Y} \varphi_1)} = -\frac{1}{4}(\boldsymbol{h}_{t-1})_{\iota(\mathsf{Y} \varphi_1)} + (\boldsymbol{x}_t)_{\iota(\varphi_1)}$。令 N_{prev} 为引理 11 中定义的前馈神经网络。非线性层输出 ϕ 现在将该序列映射为 $\boldsymbol{z}_1 \cdots \boldsymbol{z}_n$，使得 $(\boldsymbol{z}_t)_{\iota(\mathsf{Y} \varphi_1)} = N_{\text{prev}}((\boldsymbol{h}_t)_{\iota(\mathsf{Y} \varphi_1)})$。这从位置 $t-1$ 恢复 $\mathsf{Y}\ \varphi$ 的先前值，并得出所需性质。 □

现在我们可以证明定理 10。证明表明，我们不需要为 φ 的每个子公式设置单独的一层，而是可以同时计算具有相同嵌套深度的所有子公式。因此，我们需要根据嵌套深度对每个 PLTL$_f$ 公式进行分解。

定理 *10* 的证明。设 φ 是一个满足 $\mathsf{nd}(\varphi) = k$ 的 $\mathrm{PLTL}_f[\overleftarrow{\ }\#]$ 公式，并且 $M_0, \cdots, M_k$ 是其序列分解。我们通过对序列分解和 k of φ 的嵌套深度进行归纳来证明该命题。

情形 $k = 0$**:** M_0 仅包含原子公式。令 $emb:\ \Sigma \to \mathbb{R}^d$，其中 $emb(\sigma) = \boldsymbol{x}_i$，且对于 $(\boldsymbol{x}_i)_{\iota(1)} = 1$ 和 $(\boldsymbol{x}_i)_{\iota(p)} = 1$，若 $p \in \sigma$ 则为 $p \in \mathcal{P}$，否则对所有 $p \in \mathcal{P}$ 均为 0。所有其他维度均设为 0。设 $\sigma \in \Sigma^*$ 为某个词，且 $\boldsymbol{x}_1 \cdots \boldsymbol{x}_n = emb(\sigma)$，则显然 $(\boldsymbol{x}_t)_{\iota(p)} = 1 \iff \sigma,\ t \models p$ 对所有 $p \in \mathcal{P}$ 和 $1 \le t \le |\sigma|$ 均成立。

情况 $k > 0$**:** 利用归纳假设，我们假设该命题对M_{k-1}中的所有公式均成立。因此，我们可以对每个 $\psi \in M_k$应用引理 12。令（A_ψ, B_ψ）表示根据引理 12 针对每个 $\psi \in M_k$得到的门和*inc* 元组。现在，我们定义层 l_k，其中包含$A = \sum_{\psi \in M_k} A_\psi$ 和 $B = \sum_{\psi \in M_k} B_\psi$。回顾引理 12 可知，每个 A_ψ 和 B_ψ 只会影响ψ的维度，而保持所有其他维度不变。这使我们能够简单地累积所有门和*inc*的影响，从而得到一个能够同时计算所有 $\psi \in M_k$的层。对于非线性输出，情况也相同。我们论证了，对于每个 ψ，层输出 ϕ_ψ 只会影响 ψ的维度。因此，我们可以复合所有 FNN，而不会影响所需的输出。于是，令 $\phi = \bigcirc_{\psi \in M_k} \phi_\psi$, ，其中 $\bigcirc$表示 FNN 的重复复合。结合这些论证和引理 12，我们得到：对于每个词 $\sigma \in \Sigma^*$， S_φ经过层 l_n后的输出 $\boldsymbol{z}_1^l \cdots \boldsymbol{z}_{|\sigma|}^l$ 具有性质 $(\boldsymbol{z}_t^l)_{\iota(\psi)} = 1 \iff w,\ t \models \psi$ 对所有 $\psi \in M_n$ 和 $1 \le t \le |\sigma|$均成立。 □

定理*1*的证明。设 $\sigma = \sigma_1 \cdots \sigma_n \in \Sigma^*$ 是一个词， $\boldsymbol{z}_1 \cdots \boldsymbol{z}_n$ 是根据定理 10 构造的、经过 S_φ 最后一层后得到的向量序列。对于所有 $\psi \in \mathsf{Sub}(\varphi)$ 和 $1 \le t \le n$，我们有 $(\boldsymbol{z}_t)_{\iota(\psi)} = 1 \iff w,\ t \models \psi$。因此， $\sigma \in L(\varphi) \iff \sigma,\ n \models \varphi \iff (\boldsymbol{z}_n)_{\iota(\varphi)} = 1$。我们定义 S_φ 的最终输出前馈神经网络，使得 $y_t = N_{=1}((\boldsymbol{z}_t)_{\iota(\varphi)})$。由此得到 $\sigma \in L(\varphi) \iff S_\varphi(\sigma) = 1$。 □

值得注意的是，SSM 对含计数子公式的 $\mathrm{PLTL}_f[\overleftarrow{\ }\#]$ 公式的求值能力，在很大程度上取决于计算所使用的算术模型。在使用位数固定的定宽算术时，SSM 只能对系数 a_j 均为正数的计数子公式$\sum_{j=1}^{k} a_j \cdot \overleftarrow{\ }\#\ \varphi_j \sim c$ 求值（即 $\mathrm{PLTL}_f[\overleftarrow{\ }\#]^+$中的公式）。之所以存在这一限制，是因为只有当系数均为正数时，累加值才会不断增大，而

即使采用有界精度近似，与阈值 c 的比较结果仍保持稳定。判定过程只需确定累加和是大于、等于还是小于常数 c，而这可以通过固定精度实现。相反，当算术精度随输入长度呈对数增长时，SSM 能够对 $\mathrm{PLTL}_f[\overleftarrow{\#}]$, 中的一般计数公式求值，包括含负系数 a_j的公式。这是因为对数精度提供了足够的位数，能够精确统计长度为 n的词中子公式的出现次数，并保持正项与负项之间的精确差值。例如，

识别由公式 $\mathrm{H}(a \to \neg \mathrm{Y}\ b) \wedge ($ 表达的语言 $\{a^n b^n \mid n \geq 0\}$ $\overleftarrow{\#}\ a-\ \overleftarrow{\#}\ b = 0)$需要追踪 a 与 b 的出现次数之差的精确值。随着 n 增大，这就需要无限精度，因为在处理足够长的输入时，任何定精度算术最终都会导致溢出或舍入误差。

A.3 第 5 节的证明

对于时不变状态空间模型，每一层的门都是一个与输入无关的常数矩阵。对于此处的构造，我们与上一节一样，假设增量函数仅通过线性投影计算。这意味着我们可以将时不变状态空间模型层的递推关系表示为

$$\boldsymbol{h}_t = A\boldsymbol{h}_{t-1} + B\boldsymbol{x}_t\ .$$

为了便于构造，我们将证明，即使一个公式中出现多个模谓词，实际上也只需维护一个计数器。

引理 13. 对于每一个使用除数为 d_1、……、d_k 的模谓词的 $\varphi \in \text{UN-PLTL}_f[\overleftarrow{\#}、\texttt{MOD}]$，存在一个公式 $\varphi' \in \text{UN-PLTL}_f[\overleftarrow{\#}、\texttt{MOD}]$,，它仅使用具有相同除数 d' 的模谓词。

证明。 取最小公倍数 $d' = \mathrm{lcm}(d_1, \cdots, d_k)$。然后通过将每个谓词 $\mathrm{MOD}_r^{d_i}$ 替换为 $\mathrm{MOD}_r^{d'} \vee \mathrm{MOD}_{r+d_i}^{d'} \vee \mathrm{MOD}_{r+2d_i}^{d'} \vee \cdots$ 来构造 φ'。 □

现在我们将证明，每一个可由 $\text{UN-PLTL}_f[\overleftarrow{\#}, \mathrm{MOD}]$ 公式识别的语言，也可由时不变状态空间模型识别。我们将通过扩展定理 10 的证明来说明这一点。设 $\varphi \in \text{UN-PLTL}_f[\overleftarrow{\#}, \mathrm{MOD}]$ 是一个关于模谓词的公式，且这些模谓词均具有同一个除数 m。所得状态空间模型将具有 $d = |\mathcal{P}| + |\varphi| + m + 1$ 个维度。我们将函数 ι 细化为函数ι： $P \cup \{\mathrm{MOD}_r^m \mid 0 \leq r < m\} \cup \mathrm{Sub}(\varphi) \cup \{1\} \to \{1, \cdots, d\}$，该函数具有与第 3 节中相同的性质。唯一的区别在于，对于每一个可能的模谓词 MOD_r^m，我们都增加了额外的维度。为简单起见，我们确保模谓词的索引在内存中彼此相邻，并且对所有 $0 \leq r < m$ 都有 $\iota(\mathrm{MOD}_r^m) = \iota(\mathrm{MOD}_0^m) + r$，从而构成一个索引的连续区间。至此，我们便可以证明，时不变状态空间模型能够识别在 $\text{UN-PLTL}_f[\overleftarrow{\#}, \mathrm{MOD}]$ 中可定义的所有语言。

定理 14. 设 φ 是一个 $\text{UN-PLTL}_f[\overleftarrow{\#}、\texttt{MOD}]$ 公式，其定义在 $\mathcal{P}$ 上，且仅使用除数为 m的模谓词。存在一个定义在字母表 $\Sigma = 2^{\mathcal{P}}$ 上的状态空间模型 $\mathcal{S}_\varphi$，其维度为 $d = |\mathcal{P}| + m + |\varphi| + 1$，并具有 $\boldsymbol{nd}(\varphi) + 1$ 层，使得对于所有满足 $|\sigma| = n$的词 $\sigma \in \Sigma^*$：若 $\boldsymbol{y}_1 \cdots \boldsymbol{y}_n$ 是 $\mathcal{S}_\varphi$在输入 σ经过最后一层后的输出，则对于所有$1 \leq t \leq n$和 $\psi \in \boldsymbol{Sub}(\varphi)$，当且仅当 w、$t \models \psi$ 时，$(\boldsymbol{y}_t)_{\iota(\psi)} = 1$，否则为0。

证明。 证明过程与定理 10 完全相同。唯一的区别是，为了计算模谓词，我们需要在嵌入之后立即添加一层。令$\sigma \in \Sigma^*$ 和 $\boldsymbol{x}_1 \cdots \boldsymbol{x}_{\boldsymbol{n}}$ 为嵌入之后的向量序列。我们按照引理 6 中的描述添加一层$l_{\texttt{MOD}^m}$。对于这一层，我们将引理 6 中的置换矩阵 P 扩展到完整维度 d，方法是将它嵌入一个更大的矩阵中。具体而言，我们构造一个 $d \times d$矩阵，其中循环置换 P 仅作用于维度 $\iota(\mathrm{MOD}_r^m)$，其中 $0 \leq r < m$；如前所述，这些维度构成一个连续的索引块。所有其他元素均设为零。令 A 为此矩阵。我们将增量 B 设为一个看起来像单位矩阵的矩阵，但对于所有维度 $\iota(\mathrm{MOD}_r^m)$，其中 $0 \leq r < m$，其对角线元素均为零。因此我们得到 $(\boldsymbol{h}_t)_{\iota(\texttt{MOD}_r^m)} = 1 \iff t \equiv r \pmod{d}$，而对于所有其他维度 i，则有 $(\boldsymbol{h}_t)_i = (\boldsymbol{x}_t)_i$。因此，所有

原子公式在第一层之后计算。其余各层的定义与定理 10 中相同。

□

定理 7 现在可直接由前一个定理推出。

需要注意的是，关于时不变状态空间模型的定宽算术的考量，与附录 A.2 中讨论的对角状态空间模型的考量相似。在使用位数固定的定宽算术时，时不变状态空间模型只能可靠地计算具有正系数的计数子公式（即 UN-PLTL$_f[\overleftarrow{\ }\#$、MOD$]^+$ 中的公式），因为累积值只会增大，且与阈值的比较在有界精度近似下保持稳定。要计算具有负系数的公式，则需要随输入长度对数增长的算术精度。加入模谓词并不会影响这一根本局限，因为模计数器组件独立于计数机制运行。因此，在定宽算术的限制下，时不变状态空间模型的表达能力似乎仍局限于 UN-PLTL$_f$[MOD],，它对应于 FO$^2[<, \text{MOD}]$ 的一个片段。

A.4 第 6 节的证明

C-RASP 最初由 Yang 和 Chiang（2024）作为一种称为 $K_t[\#]$ 的逻辑提出。C-RASP 的语法与 UN-PLTL$_f[\overleftarrow{\ }\#]$，非常相似，唯一的区别是 C-RASP 没有昨天，也没有先前算子。唯一的时态算子是计数公式中的 $\overleftarrow{\ }\#$ 算子。

我们证明，C-RASP 的表达能力严格弱于 UN-PLTL$_f[\overleftarrow{\ }\#]$。证明基于这样一个事实：C-RASP 虽然能够模拟先前算子，却无法模拟昨天。

引理 15。 *C-RASP* $\equiv$ UN-PLTL$_f[\overleftarrow{\ }\#]$ 不含昨日算子。

证明。 C-RASP 包含于 UN-PLTL$_f[\overleftarrow{\ }\#]$ 是显然的，因为 C-RASP 的所有算子也都属于 UN-PLTL$_f[\overleftarrow{\ }\#]$。对于另一个方向，设 φ 是一个不含昨日算子的 UN-PLTL$_f[\overleftarrow{\ }\#]$ 公式。我们只需将所有先前子公式 P ψ 替换为计数公式 $\overleftarrow{\ }\#\ \psi \geq 1$。由此得到的 C-RASP 公式显然与 φ 等价。

□

为了说明 C-RASP 的表达能力严格弱于 UN-PLTL$_f[\overleftarrow{\ }\#$], $L =$，我们考虑包含 的语言。该语言可以由 UN-PLTL # 公式 P(Y) 定义，但正如 Yang 与 Chiang（2024）所指出的，它无法用 C-RASP 定义。 $\{w \in \{a,b\}^* \mid w \quad aa\}$ $_f[\overleftarrow{\ }]$ $a \wedge \ a$