arXiv:2611.30277v1 [cs.NE] 23 Nov 2026

关于人工神经网络到脉冲神经网络转换的新视角：面向神经形态推理的硬件对齐的多阈值神经元动力学

Shaohong Gu[1,2]南湖电子与信息技术学院 电子工程与信息科学学院 浙江大学 中国浙江嘉兴 / 中国浙江杭州

Jiyuan Zhang[3]计算机科学与工程学院 电子科技大学 中国四川成都

Yimeng Shan[3]计算机科学与工程学院 电子科技大学 中国四川成都

Chenxiao Dou[1], *南湖电子与信息技术学院 中国浙江嘉兴 douchenxiao@cnaeit.com

Wei Hua[1]南湖电子与信息技术学院 中国浙江嘉兴

Malu Zhang[3]电子科技大学计算机科学与工程学院，中国四川成都

Honglin Cao[3]电子科技大学计算机科学与工程学院，中国四川成都

Choonghyun Lee[2]浙江大学电子工程与信息科学学院，中国浙江杭州

Jingwei Chen[1]南湖电子与信息技术学院 中国浙江嘉兴

摘要

通过在多个时间步内累积脉冲，频率编码近似表示人工神经网络的实值激活值，这一原理支撑了众多人工神经网络到脉冲神经网络的转换方法。然而，与原始人工神经网络相比，转换后的脉冲神经网络通常会产生准确率损失。现有方法要么增加时间步数量，要么引入更复杂的神经元动力学，以缩小这一准确率差距。然而，这两种方法最终都会导致相同的结果：信息或计算方面的冗余，从而造成效率下降。为解决这一问题，我们提出了**PRIME**（多阈值编码的电位残差积分），这是一种从硬件数据表示角度重新设计的多阈值并发脉冲神经元框架：a) 我们引入残余膜电位放大（RMPA）机制，通过主动放大残余膜电位，以积分编码取代频率编码。RMPA使脉冲的二进制表示能够承载多得多的信息，仅需两个时间步即可实现近无损转换。b) 作为补充，我们在多阈值神经元中实现并发脉冲发放，直接从浮点数据的二进制表示中解码脉冲，从而消除膜电位与阈值比较，并以符合硬件计算原语的轻量级位移和累加运算取代乘法。在 $T = 2$ 和 $D = 3$条件下，仅使用FP32比特传输成本的40.6%和FP32乘法能耗的13.4%，我们便在视觉和语言基准测试中均取得了具有竞争力的性能。PRIME框架为高效的人工神经网络到脉冲神经网络的转换确立了新的设计原则，弥合了算法效率与硬件约束之间的差距。

1 引言

人脑执行高级认知任务时所消耗的能量极低 [1, 2], ，为能效硬件设计提供了极具启发性的生物学蓝图。脉冲神经网络（脉冲神经网络）模拟基于脉冲的生物通信，具有高稀疏性和事件驱动处理特性 [3, 4]。尽管通过替代梯度进行直接训练会遭遇近似误差，且时空反向传播的开销高昂 [5, 6],，人工神经网络到脉冲神经网络的转换已成为获得高性能脉冲神经网络的主流高性价比范式 [7]。

现有转换方法主要依赖频率编码 [8] ，可分为两类：（1）**延长时间步**，随着时间步增多，保真度渐近提升 [9], ；（2）**引入辅助神经元动力学**，例如差分编码或多阈值机制，但这需要存储多个预设阈值 [10, 11]。前者会造成信息冗余，因为单个激活值需要通过多个时间步上的大量脉冲来表示；后者则因辅助神经元动力学带来的复杂计算而引入计算冗余 。这两种路径有一个共同根源：都未能利用脉冲神经网络脉冲表示与现代硬件原生二进制数据格式之间的内在契合性。

我们提出了两项互补的见解，以弥补这一空白。转换误差源于残余膜电位——即经过所有时间步后仍未编码的输入。传统方法被动等待这一残余逐渐消散，因此每步只能获得微小的增益。相反，主动放大残余可实现转换误差的指数级抑制。从硬件角度看，二进制脉冲序列与存储的二进制数据具有相同的底层表示，这表明可以直接从浮点数位字段读取脉冲阈值和幅值，从而使神经元动力学与原生硬件数据格式紧密结合。

受这些见解启发，我们提出了 **PRIME** （多阈值编码的电位残差积分）。**在编码层面**，残余膜电位放大（RMPA）机制会主动放大每一步的残余膜电位，从而在$T \leq 2$实现近无损转换，且无需辅助补偿。**在推理层面**，脉冲阈值和幅值

预印本。

直接从浮点数值的符号位、指数和尾数位中读取，从而消除了膜电位累积和多阈值比较。读取 D 个连续尾数位，即可在每一步并发发放多位脉冲；当 $T \leq 2$时，步间独立性使完全并行推理成为可能，并以移位加法运算取代乘法。PRIME 同时解决了时间步扩展带来的信息冗余，以及阈值存储带来的计算冗余。大量针对计算机视觉和自然语言处理任务的实验验证了其有效性。我们的主要贡献总结如下：

- **在编码层面**，我们提出了 RMPA 机制和积分编码，通过主动放大残余膜电位，最大化每步的信息密度，从而在 $T \leq 2$ 无需辅助补偿的情况下实现近无损转换。
- **在推理层面**，我们提出一种硬件原生的脉冲驱动推理方案，将神经元动力学与浮点位表示相结合，并以移位加法运算取代乘法。当 $T \leq 2$时，步间独立性支持完全并行推理；当 $D \leq 3$时，每次步内计算最多只需一个加法器。
- 基于这两项设计，我们提出**PRIME**，这是一种统一的人工神经网络到脉冲神经网络转换框架，可同时解决编码效率低下和硬件部署开销问题。
- 大量实验表明，在 $T = 2$、 $D = 3$条件下，PRIME 在具有代表性的计算机视觉和自然语言处理基准测试上实现了近无损转换精度；其能量消耗仅为 FP32 乘法的 13.4%，层间比特传输成本仅为 FP32 的 40.6%，为能效型神经形态硬件设计奠定了坚实基础。

2 相关工作

2.1 依赖训练的转换

依赖训练的方法会重新训练源ANN，使其激活在转换前与脉冲神经元的动态特性相匹配。在频率编码下，代表性工作会优化膜电位初始化 [12], 将 ReLU 替换为量化感知激活 [13], 或采用双阶段流程进行权重与阈值协同校准 [14, 15]。对于 Transformer，SpikeZIP-TF [16]使用脉冲兼容代理函数微调预训练权重，而 SpikeLLM [17] 则通过显著性感知编码，将脉冲机制扩展到十亿参数大语言模型。尽管这些方法效果显著，但它们需要访问训练流程，因此在源ANN固定或专有时适用范围有限。

2.2 免训练转换

相比之下，免训练方法可直接转换预训练ANN，无需微调。主流范式是频率编码 [8, 18], ，它至少需要 $T = N$ 个时间步才能区分 N个激活级别。为缓解这一问题，已有工作对残余膜电位和初始膜电位进行校准 [19,20], 将神经元输出扩展为三值 [21],，或改进阈值校准策略 [22, 23]。除了频率编码，相位编码 [24, 25]可表示 2^T 个值，但必须等待所有时间步结束后才能解码；差分编码（DCGS） [26]则消除了时间衰减，但代价是引入硬件不友好的除法运算。对于 Transformer 转换，ECMT [27] 采用多阈值神经元，但仍保留浮点乘法；STA [28] 可近似任意非线性函数，但需要数百个时间步；SpikedAttention [29] 则通过迹驱动矩阵乘法实现了完全基于脉冲的转换。尽管取得了这些进展，现有方法仍全部依赖对残余膜电位的迭代累积和逐渐耗散，并在不同时间步中编码冗余信息——这是一个根本性瓶颈，迄今尚未通过算法或硬件优化加以解决。

3 方法

在本节中，我们提出了**P**otential-**R**esidual**I**ntegration for **M**ulti-threshold **E**ncoding（**PRIME**）框架，该框架从两个互补维度优化多阈值范式：更具表达力的编码策略，以及为其量身打造的面向推理的硬件设计 。

在编码层面，PRIME 引入了积分编码及其配套机制残余膜电位放大（RMPA），该机制会逐步加倍残余

膜电位，使 MTN 能够确定性地恢复先前未编码的信息，从而快速收敛至相应的人工神经网络激活值。

在推理层面，PRIME 围绕 FP16 操作数的比特级结构协同设计脉冲生成与乘法：直接 FP16 二进制解析通过前导 1 检测器，将符号位、指数和尾数字段直接映射为阈值与脉冲；而脉冲驱动乘法则将尾数乘积简化为轨迹对齐的移位加法运算，从而消除对乘法器的使用。

3.1 预备知识

遵循许多先前工作 [16, 21, 26, 27], 的成功实践，本文采用输出位于 $\{-1, 0, +1\}$ 的基于三值脉冲的多阈值神经元作为脉冲神经网络的核心计算单元。第 ℓ 层中的每个 MTN 由基准阈值 θ^ℓ 和 $2n$ 个阈值刻画，其中包括 n 个正阈值和 n 个负阈值。第 ℓ层中的第 i个阈值记为 λ_i^ℓ，在正分支（$i = 1, \ldots, n$）上定义为 $\lambda_i^\ell = \theta^\ell / 2^{i-1}$，在负分支（$i = 1, \ldots, n$）上定义为 $\lambda_{n+i}^\ell = -\theta^\ell / 2^{i-1}$，从而在零附近形成一组对称的几何衰减值。

大多数人工神经网络到脉冲神经网络的转换方法都建立在频率编码之上 [16, 21, 30]。基于频率编码的多阈值神经元的神经元动力学可描述为以下方程：

$$\begin{aligned} \mathbf{m}^\ell[t] &= \mathbf{V}^\ell[t-1] + \mathbf{I}^\ell[t], \qquad & \mathbf{S}_i^\ell[t] &= \mathrm{MTH}_{\theta,n}(\mathbf{m}^\ell[t], i), \\ \mathbf{X}^\ell[t] &= \sum_i \mathbf{S}_i^\ell[t]\, \mathbf{W}^\ell \lambda_i^\ell, & \mathbf{V}^\ell[t] &= \mathbf{m}^\ell[t] - \mathbf{X}^\ell[t], \end{aligned} \tag{1}$$

$$\mathrm{MTH}_{\theta,n}(\mathbf{m}^\ell[t], i) = \begin{cases} 1, & \text{if } i = i^*(t) \text{ and } \mathbf{m}^\ell[t] \notin (\lambda_{2n}, \lambda_n), \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

其中，$\mathbf{m}^\ell[t]$、$\mathbf{V}^\ell[t]$、$\mathbf{I}^\ell[t]$、$\mathbf{S}_i^\ell[t]$、$\mathbf{W}^\ell$ 和$\mathbf{X}^\ell[t]$ 分别表示时间步 t 时的脉冲前膜电位、脉冲后膜电位、输入电流、第 i 个阈值对应的输出脉冲、权重，以及层 ℓ 的总输出信号。这里，$i^*(t) \triangleq \arg\min_{p \in \{1,\ldots,2n\}} |\mathbf{m}^\ell[t] - \lambda_p|$ 表示与当前膜电位最接近的阈值的索引，因此，只有当 i 与该最近阈值匹配且$\mathbf{m}^\ell[t]$ 位于死区（λ_{2n}, λ_n）之外时，MTN 才会在第 i 个通道上发放脉冲。在频率编码下，人工神经网络激活值与脉冲神经网络输出之间的量化误差源于低于发放阈值的残余膜电位。现有方法通过增加时间步数或阈值数量来减小该误差，但这两种方式都会带来额外的内存、计算和能量开销，从而阻碍边缘部署。为此，我们提出了基于积分编码的多阈值神经元，直接利用残余膜电位，而无需进行大量时间步迭代或设置大量阈值。

3.2 基于积分编码的多阈值神经元

基于积分编码的 MTN 的动态过程如图 1 所示。基于传统速率编码的神经元在每个时间步累积输入电流，而基于积分编码的 MTN 仅接收一次输入，因此输入电流 $\mathbf{I}^\ell$ 直接初始化膜电位 $m^\ell = \mathbf{I}^\ell$。随后，神经元识别能够触发脉冲发放的最大阈值，该过程与浮点数的二进制结构相一致（详见第 3.3.1 小节）。首先设定基准阈值 θ^ℓ，然后根据公式（3）进行软重置：

$$\theta^\ell = 2^p \text{ where } p = \max\{p \mid 2^p \le \mathbf{m}^\ell,\ p \in \mathbb{Z}\}, \quad \mathbf{V}^\ell = \mathbf{m}^\ell - \frac{\mathbf{S}_i^\ell \theta^\ell}{2^{i-1}}, \tag{3}$$

其中，i表示 RMPA 的循环索引。在后续每次迭代中，将 $\mathbf{m}^\ell$ 设为重置后的膜电位 $\mathbf{V}^\ell$。

基于 RMPA 的 MTN 从残余膜电位中迭代生成亚阈值脉冲，逐步减小量化误差。每次脉冲生成仅需进行一次位移操作和一次与基准阈值 θ^ℓ 的比较，从而无需存储大量阈值。形式上，发放规则定义如下，即公式（4）：

$$\mathbf{S}_i^\ell = \mathrm{MTH}_{\theta,n}(\mathbf{m}^\ell, i) = \begin{cases} 1, & \text{if } \mathbf{m}^\ell \cdot 2^{i-1} - \theta^\ell \ge 0 \\ 0, & \text{otherwise} \end{cases}. \tag{4}$$

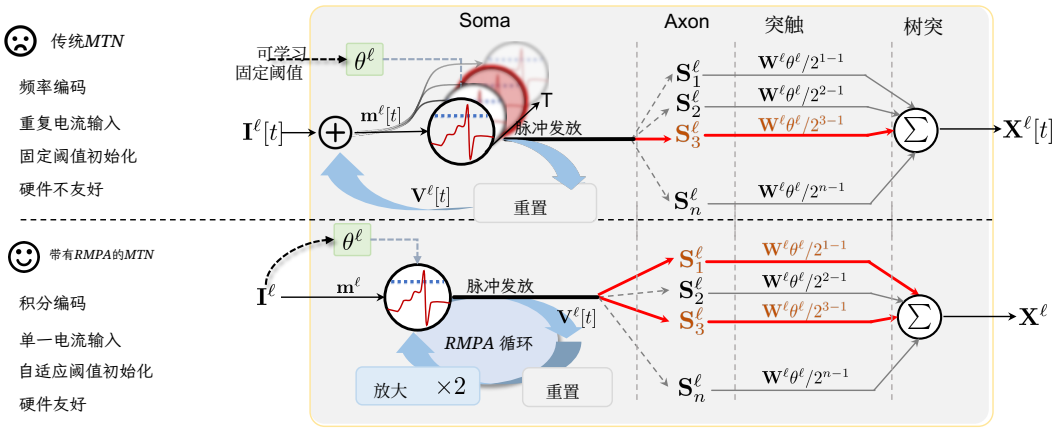

图 1：传统MTN与所提出的带有RMPA的MTN之间的比较。**上方**：传统MTN。重复电流输入会引入冗余的加法运算，而固定阈值初始化需要将膜电位与多个阈值进行比较，从而带来额外的硬件计算开销。**下方**：带有RMPA的MTN，采用单一电流输入和自适应阈值初始化，以实现更加硬件友好的设计。

随后按照公式（3）进行软重置。当膜电位降至零或达到规定的迭代次数后，神经元对发出的脉冲序列进行解码，以生成第 ℓ 层的总输出信号：

$$\mathbf{X}^{\ell} = \sum_{i} \mathbf{S}_i^{\ell} \mathbf{W}^{\ell} \frac{\theta^{\ell}}{2^{i-1}}. \tag{5}$$

我们通过一个具体示例说明RMPA的内部动态。考虑输入 $x = 1.4$。根据公式 (3)，阈值 θ^{ℓ} 被设为 1（2^p， $p = 0$）。发出一个 $\theta^{\ell} = 1$ 脉冲后，$\mathbf{m}^{\ell} = 1.4 - 1$ 变为 0.4，RMPA 循环的索引 i 为 1。在第二次迭代中，$\mathbf{m}^{\ell}$ 加倍为 0.8， i 前进至 2；由于 $\mathbf{m}^{\ell} = 0.8 < \theta^{\ell} = 1$，不会发出脉冲。在第三次迭代中，$\mathbf{m}^{\ell}$ 加倍为 1.6， i 前进至 3；由于 $\mathbf{m}^{\ell} = 1.6 > \theta^{\ell} = 1$，因此触发一个 $\theta^{\ell}/2^{i-1} = 0.25$ 脉冲，并将 $\mathbf{m}^{\ell}$ 重置为 $1.6 - 1 = 0.6$。此时，总编码值为 $1 + 0.25 = 1.25$，残余膜电位为 $\mathbf{m}^{\ell} = 0.6$。剩余编码误差为 $1.4 - 1.25$， 与 $\mathbf{m}^{l}/2^{i-1} = 0.6/4$ 的结果相同，并且该误差会随着迭代次数增加而呈指数衰减。

我们的神经元具有两大优势：(i) 阈值存储从 $2n$ 降至 $\mathcal{O}(1)$，相应地减少了输入输出操作；(ii) 与频率编码不同，基于积分编码的多阈值神经元会主动利用残余膜电位来抑制量化误差，这是其准确率提升的根本来源。

命题 1(基于RMPA的MTN误差界)。 传统速率编码的编码误差上界为 传统速率编码的编码误差上界为 $\varepsilon_{rate} < \theta/2T$，其仅随 T 线性下降。相比之下，提出的 *PRIME* 编码可达到指数级更紧的误差界 $\varepsilon_{PRIME} < \theta/2^{T \cdot D+1}$，其中 θ 表示脉冲阈值， T 表示时间步间隔数， D 表示时间步内并发脉冲数。

证明见附录 A。因此，与传统 MTN 相比，我们提出的方法可将误差降低数个数量级：例如，在 $T = 2$ 和 $D = 3$ 时，PRIME 的误差界为 $\theta/2^7 = \theta/128$，比相同 T 下频率编码的误差界 $\theta/2T = \theta/4$ 紧 $32\times$ 。

3.3 硬件高效推理

3.3.1 通过 DirectFP16 二进制解析生成脉冲

这里我们介绍如何将尖峰生成过程直接绑定到 FP16 数据的二进制表示，从而消除传统的膜电位计算。关键洞见在于，FP16 格式的每个字段：符号位、指数和尾数，都可以映射到与公式 (3) 和 (4) 对应的阈值和脉冲，从而实现纯粹基于读取的脉冲发射机制。

一个规格化的 FP16 浮点数表示为:

$$x_{\mathbf{10}} = (-1)^S \times 2^{E-15} \times \left(1 + \sum_{j=1}^{10} M_j \times 2^{-j}\right), \tag{6}$$

其中， S是 1 位符号位， E是 5 位指数（$1 \le E \le 30$）， $M_1 M_2 \cdots M_{10}$是带有隐含前导 1 的 10 位尾数。在第一个周期中，FP16 表示中的隐含前导 1 会直接作为第一个脉冲发出，符号位 S 与指数 E 共同决定相应的阈值 $\theta_1 = (-1)^S \times 2^{E-15}$。从第二个周期开始，使用前导 1 检测器（LOD）定位剩余尾数中第一个 1 的位置。LOD 是数字电路设计中一种成熟的组合逻辑模块。如果第一个 1 位于位置 k（从最高有效位开始计数， $k = 1,\ 2,\ \ldots, 10$），RMPA 循环将运行 k次，对应发出脉冲的阈值为 $\theta_k = (-1)^S \times 2^{E-15-k}$.。也就是说，位置 k直接对初始阈值对应的缩放因子 2^{-k} 进行编码，无需显式计算。发出脉冲后，位置 k 上的位会被清除，并在下一个周期对更新后的尾数应用 LOD，以查找下一个 1。此过程会持续进行，直到所有周期耗尽或尾数变为零。与需要执行完整减法、比较和条件更新的传统 ANN 到 SNN 方法不同，该方法将每个周期简化为 LOD 查找，随后进行位清除操作，从而大幅降低计算成本。

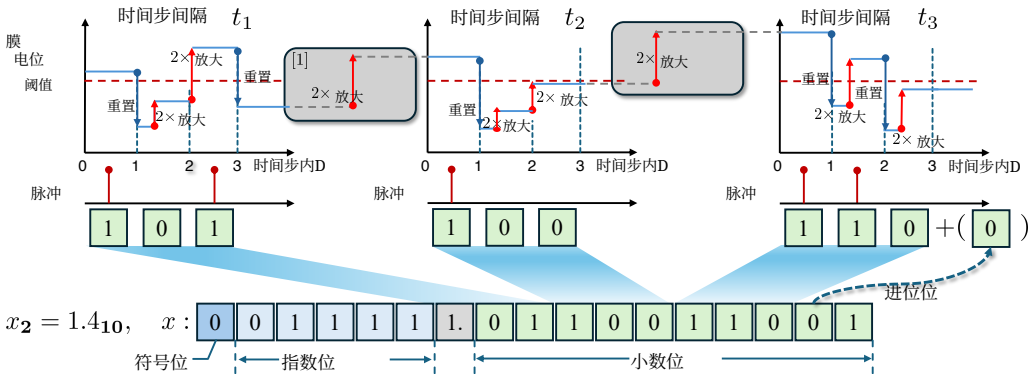

图 2：将 1.4 的 FP16 表示解析为 D=T=3 脉冲。第一个时间步间隔发放由隐式1与前 2 个尾数位拼接而成的内容，阈值为 $\theta = (-1)^S \times 2^{E-15-(D-1)}$；第二个时间步间隔使用前导零检测器定位剩余 8 位中的第一个 1，从该位置读取 D 位作为脉冲值，并相应调整阈值。[1] 放大次数并非固定为 1，而是等于下一个前导 1 相对于当前 T 位组末尾的位偏移量。

图 2 展示了将 1.4 的 FP16 数据解析为脉冲的示例。我们将 D 称为时间步内脉冲维度，它决定单个时间步内发放多少位；将 T 称为时间步间隔数，它决定使用多少个连续时间步来编码给定值。在基本情形下（$D = 1$），每个时间步发放一个 1 位脉冲，用于指示是否达到阈值。这种范式还支持多位脉冲发放，即每次读取 D 个连续位，而非单个位。当 $D > 1$时，每个脉冲都是一个 D 位无符号整数，取值范围为 0 到 $2^D - 1$，表示对应阈值的 0 到 $2^D - 1$ 倍。例如，当 $D = 2$时，每个时间步读取 2 位，编码的脉冲值为 $\{0, 1, 2, 3\}$；当 $D = 3$时，每个时间步读取 3 位，编码的值为 $\{0, 1, \ldots, 7\}$。在最后一个时间步中，还会额外读取一位（即第 (D+1) 位），超出 D 位脉冲。随后，将这额外的一位与作为无符号整数处理的 D 位脉冲值相加，以执行进位操作；这等效于四舍五入操作，且无需单独的舍入阶段即可进一步提高编码精度。

在硬件层面，这种时间步内扩展只需将读取窗口从 1 比特加宽至 D 比特；基于LOD的位置查找和阈值确定则保持不变。这种多比特发放显著提高了每个时间步的信息密度，而在传统的人工神经网络到脉冲神经网络转换算法中，每个时间步本质上都受限于二元发放或不发放决策，因此难以实现这种效果。

这种二进制解析方法的一个关键优势是，当 $T \le 2$ 时，两个时间步之间的数据相互独立：第一个时间步读取隐式1和前导尾数位，第二个则处理

剩余位并通过前导零检测器进行处理。由于第二个时间步仅取决于预先确定的位范围，两者可以并行执行，从而有效地将延迟减半。

3.3.2 脉冲轨迹下的高效乘法

上述脉冲生成机制将一个 FP16 值分解为跨越 T 个时间步间隔的一系列加权脉冲。下面我们将展示，这种分解如何自然地导出一种硬件高效乘法方案：无需执行完整的 11×11 位尾数乘法，而是通过各时间步中的脉冲驱动条件移位加法操作，逐步计算乘积 $x \times w$。

符号和指数。 给定两个 FP16 操作数 x 和 w，乘积的符号位和指数按标准方式计算。输出符号位通过一个异或门得到：$S_o = S_x \oplus S_w$；输出指数为 $E_o = E_x + E_w -$ 偏置，其中偏置 $= 01111_2 = 15_{10}$。这仅需一个 5 位加法器和一个 6 位减法器，与传统 FP16 乘法器相同。创新的关键完全在于尾数乘积的计算方式，详见下文。

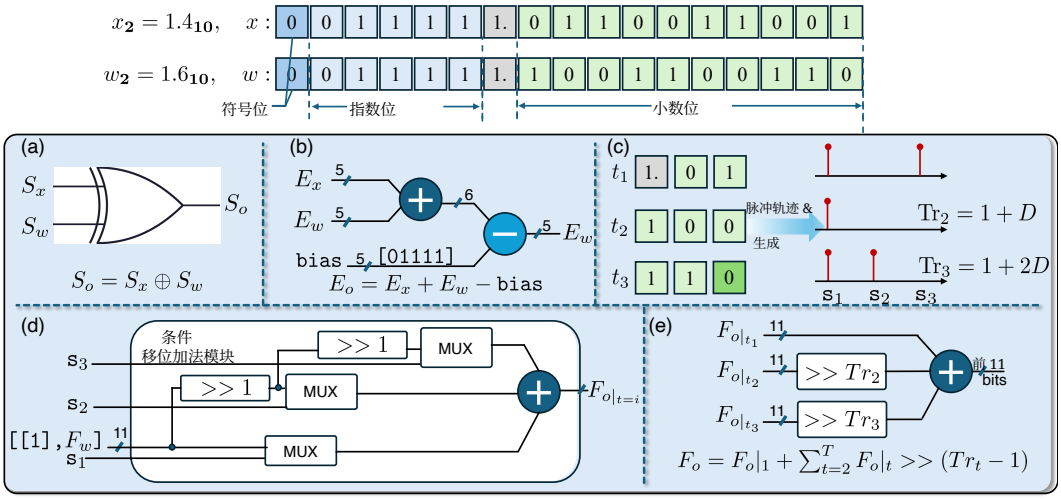

图3：脉冲驱动的FP16乘法。x 和 w 的符号位、指数和尾数并行处理：（**a**）$S_o = S_x \oplus S_w$ 通过异或；（**b**）$E_o = E_x + E_w$ –偏置通过5位加法器和6位减法器处理；（**c**）尾数 F_x 分解为 T 个脉冲 $\{s_t\}$，其轨迹为$\text{Tr}_t = 1 + (t-1)D$；（**d**）每个部分积 $F_o|_t = s_t \times F_w$ 由一个包含多路复用器和加法器树的条件移位加法模块计算，取代 11×11 乘法器阵列；（**e**）部分积通过右移进行轨迹对齐并求和，得到 F_o，再截断为11位。

通过脉冲轨迹进行尾数分解。设$F_x = 1.M_{x,1}M_{x,2}\cdots M_{x,10}$ 表示 x 的完整尾数（包括隐含前导 1），F_w 表示 w的完整尾数。上一小节介绍的脉冲生成过程将 F_x 分解为 T 个脉冲值 $\{s_t\}_{t=1}^{T}$，每个脉冲值都是一个 D位无符号整数，并关联一个轨迹值Tr_t，用于记录第 t个脉冲相对于二进制小数点的位位置偏移量。具体而言，第一个脉冲 s_1 由隐式1和前$D-1$ 个尾数位组成，其对应的轨迹为$\text{Tr}_1 = 1$。对于后续时间步 $t \geq 2$，前导零检测器会确定下一组的起始位置，并读取连续的 D 位作为 s_t。当尾数位密集排列（即各组之间没有间隔零）时，轨迹值满足：$\text{Tr}_t = 1 + (t-1) \times D$，。在这种分解下，尾数可重构为$F_x = \sum_{t=1}^{T} s_t \times 2^{-(\text{Tr}_t - 1)}$。将其代入尾数乘积可得:

$$F_o = F_x \times F_w = \sum_{t=1}^{T} \underbrace{(s_t \times F_w)}_{\triangleq F_o|_t} \times 2^{-(\text{Tr}_t - 1)}. \tag{7}$$

这种因式分解表明，11×11 位尾数乘法被分解为 T 次较小的乘法：D 位脉冲 s_t 与 11 位尾数 F_w 相乘，随后进行轨迹对齐累加。

每个时间步的部分积 $F_o|_t = s_t \times F_w$ 都是一次 D 位与 $\times 11$ 位的乘法，如图 2(d) 所示，该运算可通过条件移位加法模块高效实现。设 $s_t = \sum_{j=1}^{D} b_j \times 2^{D-j}$，其中 $b_j \in \{0, 1\}$ 是 D 位脉冲的各个比特。那么：

$$F_o|_t = s_t \times F_w = \sum_{j=1}^{D} b_j \times (F_w \gg (j-1)), \tag{8}$$

其中，$\gg$ 表示逻辑右移。每一项 $b_j \times (F_w \gg (j-1))$ 均通过由 b_j 控制的多路复用器（MUX）实现：当 $b_j = 1$ 时，选择移位后的值 $F_w \gg (j-1)$；当 $b_j = 0$ 时，选择零值。随后，D 选出的值由加法器树求和。这种条件移位加法结构以 D 个多路复用器（MUX）和一个（$D-1$）级加法器树取代组合乘法器阵列，从而显著减少硬件面积。需要注意的是，当脉冲为 $= 111$ 时，可将乘法重写为 $1000 - 0001$，从而更高效地完成计算，仅需执行一次减法。详细推导见附录 C。

与迹对齐的时间累加。 如图 3(e) 所示，最终尾数是与迹对齐的 T 个部分积 $F_o|_t$ 之和：

$$F_o = F_o|_1 + \sum_{t=2}^{T} F_o|_t \gg (\mathrm{Tr}_t - 1). \tag{9}$$

右移 $\gg (\mathrm{Tr}_t - 1)$ 将第 t 个部分积相对于第一个部分积对齐到正确的比特位置，随后将最终和截断为前 11 个比特数，以生成输出尾数。这种时间累加过程直接反映了脉冲神经网络中的突触整合动态：每个输入脉冲 s_t 都会触发一个加权响应 $F_o|_t$，而迹 Tr_t 控制时间衰减（此处通过右移实现），之后各响应在输出神经元处累加。
如附录 B 所示，在 T = 2、D = 3 的场景下，通过 DirectFP16 二进制解析生成脉冲和脉冲驱动乘法，使该过程的能耗仅为 FP32 乘法的 13.4%，层间比特传输成本仅为 FP32 的 40.6%。

4 实验

本节在计算机视觉（CV）和自然语言处理（NLP）任务上评估所提出的 PRIME 框架，并将转换精度与现有方法进行比较。我们还通过消融研究考察不同参数设置对性能的影响。所有实验均使用 PyTorch 2.10.0 和 CUDA 12.8，在单块 NVIDIA A100 GPU 上完成。

4.1 计算机视觉任务评估

对于计算机视觉任务，我们在具有代表性的卷积神经网络和 Transformer 架构上验证 PRIME 的有效性，包括 VGG、ResNet、ViT 和 EVA，并在 ImageNet-1K [31] 和 CIFAR [32]（见附录 G）数据集上进行评估。我们遵循 [33] 的实验设置。
ImageNet-1K 上的实验结果如表 1 所示。PRIME 在卷积神经网络和 Transformer 架构中均始终实现更低的转换误差。与最具代表性的基线 DCGS 相比， [10], PRIME 仅需两个时间步，便能在 VGG 和 ResNet 上实现更低的准确率下降（0.04% 对比 0.08%，0.00% 对比 0.38%）；在 ViT 和 EVA 架构上，优势则更为显著（0.00% 对比 0.39%，0.00% 对比 0.33%）。总而言之，在现有框架中，PRIME 最接近无损的人工神经网络到脉冲神经网络的转换，同时还能在硬件部署时进一步降低层间位宽（13 比特数对比 16 比特数）。

4.2 自然语言任务评估

对于自然语言处理任务，我们将该方法应用于通义千问和 Llama 系列模型 [42, 43], ，并在主要基准测试上进行评估，包括大规模多任务语言理解、PIQA、ARC-C、ARC-E、Winogrande 和 HellaSwag [44, 45, 46, 47]。评估过程基于文献 [48] 实现。我们使用 LLAMA2-7b 和 LLAMA2-13b 模型，在超低延迟场景下评估该方法在常识推理任务上的表现（$T = 2$）。按照脉冲神经网络转换中的常见做法，我们采用

表 1：ImageNet-1K 数据集上不同方法的性能比较。“比特数”表示在所有时间步上传输的比特总数（即脉冲在 T 个时间步上的累积位宽），反映了推理过程中的层间数据传输成本。

方法	架构	参数量（M）	人工神经网络准确率（%）	T	Bits	脉冲神经网络准确率（%）
TS[34]	VGG-16	138.0	72.40	64	64	70.97 (-1.43)
SNM[35]	VGG-16	138.0	73.18	64	128	71.50 (-1.68)
QCFS[36]	ResNet-34	21.8	74.32	64	64	72.35（-1.97）
	VGG-16	138.0	74.29	64	64	72.85（-1.44）
SRP[37]	ResNet-34	21.8	74.32	64	64	68.61 (-5.71)
	VGG-16	138.0	74.29	64	64	69.43 (-4.86)
MST[38]	Swin-T	28.5	80.51	512	512	78.51 (-2.00)
STA[39]	ViT-B/32	86.0	83.60	256	256	82.79 (-0.81)
SpikeZIP-TF[40]	SViT-S-32	22.1	81.59	64	128	81.45（-0.14）
	SViT-B-32	86.6	82.83	64	128	82.71（-0.12）
	SViT-L-32	304.3	83.86	64	128	83.82（-0.04）
ECMT[33]	ViT-S/16	22.0	78.04	10	40	77.07（-0.97）
	EVA-G	1074.0	89.62	8	32	89.40（-0.22）
自适应裂变[41] + QCFS[36]	ResNet-34	21.8	74.32	32	64	69.37 (-4.95)
DCGS[10]	VGG-4/1	138.0	73.25	8	24	73.17 (-0.08)
	ResNet34-4/1	21.8	76.42	8	24	76.04 (-0.38)
	ViT-L-8/4	304.3	85.84	4	16	85.45 (-0.39)
	EVA02-L-8/4	305.1	90.05	4	16	89.72 (-0.33)
PRIME（我们的方法）	VGG-16	138.0	71.12	**2** $(D{=}3)$	**13**	**71.08 (-0.04)**
	ResNet-34	21.8	76.09	**2** $(D{=}3)$	**13**	**76.13 (-0.00)**
	ViT-S	22.1	78.04	**2** $(D{=}3)$	**13**	**78.06 (-0.00)**
	ViT-B	86.6	82.91	**2** $(D{=}3)$	**13**	**82.87 (-0.04)**
	ViT-L	304.3	84.88	**2** $(D{=}3)$	**13**	**84.89（-0.00）**
	EVA02-T	5.8	80.62	**2** $(D{=}3)$	**13**	**80.63（-0.00）**
	EVA02-S	22.1	85.73	**2** $(D{=}3)$	**13**	**85.70 (-0.03)**
	EVA02-B	87.1	88.26	**2** $(D{=}3)$	**13**	**88.27 (-0.00)**
	EVA02-L	305.1	89.63	**2** $(D{=}3)$	**13**	**89.63 (-0.00)**
	EVA02-G	1013.0	89.62	**2** $(D{=}3)$	**13**	**89.58 (-0.04)**

表2：常识推理任务上的性能比较（%）。“比特数”表示所有时间步传输的总比特数，反映推理过程中的数据传输成本。

方法	模型	脉冲	T	Bits	PIQA	ARC-c n_	ARC-e n_	Winog.	HellS. n_
LLAMA	LLAMA2-7b	×	N/A	16	78.45	45.73	74.87	69.38	76.28
	LLAMA2-13b	×	N/A	16	80.52	48.89	76.56	72.22	79.71
SpikeLLM[11]	LLAMA2-7b	✓	2	32	65.67	28.41	41.88	52.80	49.87
	LLAMA2-13b	✓	2	32	67.68	30.38	44.53	57.22	58.02
FAS[49]	OPT-7B	✓	16	32	73.23	—	64.73	60.38	—
	OPT-7B	✓	16	64	76.22	—	65.95	65.75	—
LAS[50]	OPT-13B	✓	16	64	76.28	—	67.38	65.51	—
	OPT-66B	✓	16	64	78.67	—	71.25	68.27	—
PRIME（我们的方法）	LLAMA2-7b	✓	**2** $(D{=}3)$	**13**	**78.67**	**45.65**	**74.75**	**69.06**	**76.18**
	LLAMA2-13b	✓	**2** $(D{=}3)$	**13**	**80.58**	**49.15**	**76.47**	**72.14**	**79.69**

相同的架构配置，不作修改。结果见表2。据我们所知，这是首个仅用 2 个时间步，便能在常识推理基准上始终将准确率下降控制在 -0.3% 的转换框架。不同场景和数据集下的更多结果见附录H和I。

4.3 消融研究

默认情况下，我们采用完整的 FP16 指数位宽（5 比特）来设定阈值范围，从而保持转换精度。为评估效率与准确率之间的权衡，我们将指数位宽降至 4，并在表 3 中报告结果。在所有（T, D）配置中，准确率下降幅度很小，而激活阈值的数量和层间比特传输成本则显著降低。这证实，所提出的阈值范围优化为资源受限部署提供了一种实用的调节手段，且几乎不会牺牲性能。

表 3：在 ImageNet-1K 上对通过降低指数位宽（宽度=4）实现的轻量级阈值范围优化进行消融研究，并在不同（T, D）场景下进行评估。默认设置指采用完整的 FP16 指数位宽，仅对 Transformer 块中的 k 和 v 应用脉冲转换。

模型	ANN	D	默认			指数位宽 $=4$		
			$T=1$	$T=2$	$T=3$	$T=1$	$T=2$	$T=3$
ViT-S（patch16_224）	78.04	1	48.20	76.27	77.80	55.26	76.29	77.47
		2	73.78	77.94	78.03	73.94	77.66	77.65
		3	77.01	78.06	78.03	76.92	77.66	77.65
ViT-B（patch16_224）	82.91	1	74.00	82.21	82.74	75.80	81.30	81.77
		2	80.89	82.77	82.89	80.30	81.74	81.88
		3	82.41	82.87	82.91	81.38	81.87	81.87
ViT-L（patch16_224）	84.88	1	80.89	84.46	84.79	81.14	83.81	84.31
		2	84.03	84.80	84.89	83.25	84.31	84.32
		3	84.45	84.89	84.88	84.01	84.30	84.30

5 结论与局限性

在本文中，我们提出了 PRIME，一种统一的人工神经网络到脉冲神经网络的转换框架，协同解决时间步扩展带来的信息冗余和辅助神经元动力学带来的计算冗余。在编码层面，基于积分编码的 MTN 利用 RMPA 机制，在每次迭代中主动放大残余膜电位，实现了相比频率编码线性误差界 $\theta/2^T$ 指数级更紧的误差界 $\theta/2^{T\cdot D+1}$，从而在 $T \leq 2$ 下实现近无损转换。在推理层面，脉冲阈值和值直接从浮点二进制表示中读取，消除了膜电位累积和多阈值比较；当 $D \leq 3$ 时，所得的脉冲驱动乘法每个周期最多只需一个加法器。在 $T = 2$、$D = 3$ 条件下，能耗达到 FP32 乘法的 13.4%，比特传输成本达到 FP32 的 40.6%。这些数值基于解析估算；然而，所有底层原语（位移、整数加法和前导 1 检测）均由标准数字逻辑原生支持，因此物理实现是下一步的自然方向。

参考文献

[1] Wulfram Gerstner、Werner M Kistler、Richard Naud 和 Liam Paninski。《神经元动力学：从单个神经元到网络与认知模型》。剑桥大学出版社，2014 年。[2] Eugene M Izhikevich。脉冲神经元的简单模型。《IEEE 神经网络汇刊》，14(6):1569–1572，2003 年。[3] Wolfgang Maass。脉冲神经元网络：第三代神经网络模型。《神经网络》，10(9):1659–1671，1997 年。[4] Mike Davies、Narayan Srinivasa、Tsung-Han Lin、Gautham Chinya、Yongqiang Cao、Sri Harsha Choday、Georgios Dimou、Prasad Joshi、Nabil Imam、Shweta Jain、Yuyun Liao、Chit-Kwan Lin、Andrew Lines、Ruokun Liu、Deepak Mathaikutty、Steven McCoy、Arnab Paul、Jonathan Tse、Guruguhanathan Venkataramanan、Yi-Hsin Weng、Andreas Wild、Yoonseok Yang 和 Hong Wang。Loihi：一种支持片上学习的神经形态多核处理器。《IEEE Micro》，38(1):82–99，2018 年。

[5] Julia Gygax 和 Friedemann Zenke。阐明脉冲神经网络中替代梯度学习的理论基础，2024 年。[6] Yaoyu Zhu、Jianhao Ding、Tiejun Huang、Xiaodong Xie 和 Zhaofei Yu。脉冲神经网络的在线稳定化。载于《第十二届国际学习表征会议》，2024 年。[7] Yongqiang Cao、Yang Chen 和 Deepak Khosla。用于节能目标识别的脉冲深度卷积神经网络。《国际计算机视觉杂志》，113(1):54–66，2015年。[8] Bodo Rueckauer、Iulia-Alexandra Lungu、Yuhuang Hu、Michael Pfeiffer 和 Shih-Chii Liu。将连续值深度网络转换为用于图像分类的高效事件驱动网络。《神经科学前沿》，11，2017年。[9] Wei Fang、Yanqi Chen、Jianhao Ding、Zhaofei Yu、Timothée Masquelier、Ding Chen、Liwei Huang、Huihui Zhou、Guoqi Li 和 Yonghong Tian。Spikingjelly：面向基于脉冲的智能的开源机器学习基础设施平台。《科学进展》，9(40):eadi1480，2023年。[10] Zihan Huang、Wei Fang、Tong Bu、Peng Xue、Zecheng Hao、Wenxuan Liu、Yuanhong Tang、Zhaofei Yu 和 Tiejun Huang。用于免训练ANN到SNN转换的差分编码，2025年。[11] Xingrun Xing、Boyan Gao、Zheng Zhang、David A. Clifton、Shitao Xiao、Li Du、Guoqi Li 和 Jiajun Zhang。Spikellm：通过基于显著性的脉冲化，将脉冲神经网络扩展至大语言模型，2025年。[12] Tong Bu、Jianhao Ding、Zhaofei Yu 和 Tiejun Huang。面向低延迟脉冲神经网络的优化电位初始化。《AAAI人工智能会议论文集》，36(1):11–20，2022年6月。[13] Tong Bu、Wei Fang、Jianhao Ding、PengLin Dai、Zhaofei Yu 和 Tiejun Huang。面向高准确率和超低延迟脉冲神经网络的最优ANN-SNN转换。《arXiv预印本 arXiv:2303.04347》，2023年。[14] Jianhao Ding、Zhaofei Yu、Yonghong Tian 和 Tiejun Huang。深度脉冲神经网络中快速且准确推理的最优ANN-SNN转换。《arXiv预印本 arXiv:2105.11654》，2021年。[15] Ziming Wang、Shuang Lian、Yuhao Zhang、Xiaoxin Cui、Rui Yan 和 Huajin Tang。通过双阶段优化实现超低延迟下的无损ANN-SNN转换。《arXiv预印本 arXiv:2205.07473》，2022年。[16] Kang You、Zekai Xu、Chen Nie、Zhijie Deng、Qinghai Guo、Xiang Wang 和 Zhezhi He。Spikezip-tf：转换就是基于 Transformer 的脉冲神经网络所需的一切。《arXiv预印本 arXiv:2406.03470》，2024年。[17] Xingrun Xing、Boyan Gao、Zheng Zhang、David A Clifton、Shitao Xiao、Li Du、Guoqi Li 和 Jiajun Zhang。Spikellm：通过基于显著性的脉冲化，将脉冲神经网络扩展至大语言模型。《arXiv预印本 arXiv:2407.04752 》，2024 年。[18] Peter U Diehl 、 Daniel Neil 、 Jonathan Binas 、 Matthew Cook、Shih-Chii Liu 和 Michael Pfeiffer。通过权重与阈值平衡，实现快速分类、高准确率的脉冲深度网络。载于《2015年国际神经网络联合会议（IJCNN）》，第 1–8 页。IEEE，2015年。[19] Zecheng Hao、Tong Bu、Jianhao Ding、Tiejun Huang 和 Zhaofei Yu。通过残余膜电位减少ANN-SNN转换误差，2023年。[20] Zecheng Hao、Jianhao Ding、Tong Bu、Tiejun Huang 和 Zhaofei Yu。通过校准偏移脉冲弥合ANN与SNN之间的差距。载于《第十一届国际学习表征会议》，2023年。

[21] Yuchen Wang、Malu Zhang、Yi Chen 和 Hong Qu。带记忆的符号神经元：迈向简单、准确且高效的人工神经网络-脉冲神经网络转换。在*IJCAI*，第2501–2508页，2022年。[22] Ziqing Wang、Yuetong Fang、Jiahang Cao、Hongwei Ren 和 Renjing Xu。自适应校准：脉冲神经网络的统一转换框架。*AAAI*人工智能会议论文集，39(2):1583–1591，2025年4月。[23] Tong Bu、Maohua Li 和 Zhaofei Yu。面向高性能和低功耗应用的人工神经网络-脉冲神经网络转换中的推理规模复杂度。在计算机视觉与模式识别会议，第24387–24397页，2025年。[24] Jaehyun Kim、Heesu Kim、Subin Huh、Jinho Lee 和 Kiyoung Choi。具有加权脉冲的深度神经网络。*Neurocomputing*，311:373–386，2018。[25] Zhehui Wang、Xiaozhe Gu、Rick Siow Mong Goh、Joey Tianyi Zhou 和 Tao Luo。采用基数编码的高效脉冲神经网络。*IEEE*神经网络与学习系统汇刊，35(3):3689–3701，2022年。[26] Zihan Huang、Xinyu Shi、Zecheng Hao、Tong Bu、Jianhao Ding、Zhaofei Yu 和 Tiejun Huang。用于免训练人工神经网络到脉冲神经网络转换的差分编码。*arXiv*预印本 *arXiv:2503.00301*，2025年。[27] Zihan Huang、Xinyu Shi、Zecheng Hao、Tong Bu、Jianhao Ding、Zhaofei Yu 和 Tiejun Huang。通过人工神经网络到脉冲神经网络的转换实现高性能脉冲Transformer。在第*32*届*ACM*国际多媒体会议，第10688–10697页，2024年。[28] Yizhou Jiang、Kunlin Hu、Tianren Zhang、Haichuan Gao、Yuqian Liu、Ying Fang 和 Feng Chen。时空近似：一种用于Transformer的免训练脉冲神经网络转换。在第十二届国际学习表征会议，2024年。[29] Sangwoo Hwang、Seunghyun Lee、Dahoon Park、Donghun Lee 和 Jaeha Kung。Spike-dattention：通过面向胜者的脉冲移位实现免训练、完全脉冲驱动的Transformer到SNN转换，用于Softmax运算。神经信息处理系统进展，37:67422–67445，2024年。[30] Jingya Wang、Xin Deng、Wenjie Wei、Dehao Zhang、Shuai Wang、Qian Sun、Jieyuan Zhang、Hanwen Liu、Ning Xie 和 Malu Zhang。面向高性能脉冲Transformer的免训练人工神经网络到脉冲神经网络转换，2025年。[31] Jia Deng、Wei Dong、Richard Socher、Li-Jia Li、Kai Li 和 Li Fei-Fei。ImageNet：一个大规模分层图像数据库。在*2009*年 *IEEE* 计算机视觉与模式识别会议，第248–255页，2009年。[32] Alex Krizhevsky。从微型图像中学习多层特征。技术报告，2009年。[33] Zihan Huang、Xinyu Shi、Zecheng Hao、Tong Bu、Jianhao Ding、Zhaofei Yu 和 Tiejun Huang。通过人工神经网络到脉冲神经网络的转换实现高性能脉冲Transformer。载于 Jianfei Cai、Mohan S. Kankanhalli、Balakrishnan Prabhakaran、Susanne Boll、Ramanathan Subramanian、Liang Zheng、Vivek K. Singh、Pablo César、Lexing Xie 和 Dong Xu（编），第*32*届*ACM*国际多媒体会议，*MM 2024*，澳大利亚维多利亚州墨尔本，*2024*年*10*月*28*日–*2024*年*11*月*1*日，第10688–10697页。ACM，2024年。[34] Shikuang Deng 和 Shi Gu。传统人工神经网络到脉冲神经网络的最优转换。在国际学习表征会议，2021年。[35] Yuchen Wang、Malu Zhang、Yi Chen 和 Hong Qu。带记忆的符号神经元：迈向简单、准确且高效的人工神经网络-脉冲神经网络转换。在国际人工智能联合会议论文集，第2501–2508页，2022年。[36] Tong Bu、Wei Fang、Jianhao Ding、PengLin Dai、Zhaofei Yu 和 Tiejun Huang。面向高精度和超低延迟脉冲神经网络的最优人工神经网络-脉冲神经网络转换。*arXiv*预印本 *arXiv:2303.04347*，2023年。

[37] Zecheng Hao、Tong Bu、Jianhao Ding、Tiejun Huang 和 Zhaofei Yu。通过残余膜电位降低人工神经网络-脉冲神经网络转换误差。*AAAI*人工智能会议论文集，37(1):11–21，2023年6月。[38] Ziqing Wang、Yuetong Fang、Jiahang Cao、Qiang Zhang、Zhongrui Wang 和 Renjing Xu。掩码脉冲变换器。在*IEEE/CVF* 国际计算机视觉会议，第1761–1771页，2023年。[39] Yizhou Jiang、Kunlin Hu、Tianren Zhang、Haichuan Gao、Yuqian Liu、Ying Fang 和 Feng Chen。时空近似：一种用于Transformer的免训练 SNN 转换。在第十二届国际学习表征会议，2024年。[40] kang you、Zekai Xu、Chen Nie、Zhijie Deng、Qinghai Guo、Xiang Wang 和 Zhezhi He。SpikeZIP-TF：对于基于 Transformer 的 SNN，转换就是你所需要的一切。在第四十一届国际机器学习会议，2024年。[41] Yizhou Jiang、Feng Chen、Yihan Li、Yuqian Liu、Haichuan Gao、Tianren Zhang 和 Ying Fang。自适应裂变：用于低延迟脉冲神经网络的训练后编码。在第三十九届神经信息处理系统年会，2025年。

[42] An Yang、Anfeng Li、Baosong Yang、Beichen Zhang、Binyuan Hui、Bo Zheng、Bowen Yu、Chang Gao、Chengen Huang、Chenxu Lv、Chujie Zheng、Dayiheng Liu、Fan Zhou、Fei Huang、Feng Hu、Hao Ge、Haoran Wei、Huan Lin、Jialong Tang、Jian Yang、Jianhong Tu、Jianwei Zhang、Jianxin Yang、Jiaxi Yang、Jing Zhou、Jingren Zhou、Junyang Lin、Kai Dang、Keqin Bao、Kexin Yang、Le Yu、Lianghao Deng、Mei Li、Mingfeng Xue、Mingze Li、Pei Zhang、Peng Wang、Qin Zhu、Rui Men、Ruize Gao、Shixuan Liu、Shuang Luo、Tianhao Li、Tianyi Tang、Wenbiao Yin、Xingzhang Ren、Xinyu Wang、Xinyu Zhang、Xuancheng Ren、Yang Fan、Yang Su、Yichang Zhang、Yinger Zhang、Yu Wan、Yuqiong Liu、Zekun Wang、Zeyu Cui、Zhenru Zhang、Zhipeng Zhou 和 Zihan Qiu。通义千问 Qwen3 技术报告。*arXiv* 预印本 *arXiv:2505.09388*，2025年。

[43] Hugo Touvron、Louis Martin、Kevin Stone、Peter Albert、Amjad Almahairi、Yasmine Babaei、Nikolay Bashlykov、Soumya Batra、Prajjwal Bhargava、Shruti Bhosale、Dan Bikel、Lukas Blecher、Cristian Canton Ferrer、Moya Chen、Guillem Cucurull、David Esiobu、Jude Fernandes、Jeremy Fu、Wenyin Fu、Brian Fuller、Cynthia Gao、Vedanuj Goswami、Naman Goyal、Anthony Hartshorn、Saghar Hosseini、Rui Hou、Hakan Inan、Marcin Kardas、Viktor Kerkez、Madian Khabsa、Isabel Kloumann、Artem Korenev、Punit Singh Koura、Marie-Anne Lachaux、Thibaut Lavril、Jenya Lee、Diana Liskovich、Yinghai Lu、Yuning Mao、Xavier Martinet、Todor Mihaylov、Pushkar Mishra、Igor Molybog、Yixin Nie、Andrew Poulton、Jeremy Reizenstein、Rashi Rungta、Kalyan Saladi、Alan Schelten、Ruan Silva、Eric Michael Smith、Ranjan Subramanian、Xiao- qing Ellen Tan、Binh Tang、Ross Taylor、Adina Williams、Jian Xiang Kuan、Puxin Xu、Zheng Yan、Iliyan Zarov、Yuchen Zhang、Angela Fan、Melanie Kambadur、Sharan Narang、Aurelien Rodriguez、Robert Stojnic、Sergey Edunov 和 Thomas Scialom。Llama 2：开放基础模型和微调聊天模型，2023年。

[44] Dan Hendrycks、Collin Burns、Steven Basart、Andy Zou、Mantas Mazeika、Dawn Song 和 JacobSteinhardt。《衡量大规模多任务语言理解》。*CoRR*，abs/2009.03300，2020年。

[45] Yonatan Bisk、Rowan Zellers、Ronan Le Bras、Jianfeng Gao 和 Yejin Choi。《PIQA：用自然语言进行物理常识推理》。载于 第三十四届人工智能促进协会人工智能会议，2020 年。[46] Niket Tandon、Bhavana Dalvi Mishra、Keisuke Sakaguchi、Antoine Bosselut 和 Peter Clark。《Wiqa：针对程序性文本进行“如果……会怎样”推理的数据集》。*arXiv:1909.04739v1*，2019 年。[47] Peter Clark、Isaac Cowhey、Oren Etzioni、Tushar Khot、Ashish Sabharwal、Carissa Schoenick 和 Oyvind Tafjord。《你以为自己解决了问答问题？试试 ARC，即 AI2推理挑战》。*arXiv:1803.05457v1*，2018 年。[48] Leo Gao、Jonathan Tow、Baber Abbasi、Stella Biderman、Sid Black、Anthony DiPofi、Charles Foster、Laurence Golding、Jeffrey Hsu、Alain Le Noac’h、Haonan Li、Kyle McDonell、Niklas Muennighoff、Chris Ociepa、Jason Phang、Laria Reynolds、Hailey Schoelkopf、Aviya Skowron、Lintang Sutawika、Eric Tang、Anish Thite、Ben Wang、Kevin Wang 和 Andy Zou。《语言模型评测工具框架》，2024 年 7 月。

[49] Long Chen、Xiaotian Song、Andy Song、BaDong Chen、Jiancheng Lv 和 Yanan Sun。《FAS：面向脉冲大语言模型的快速 ANN-SNN 转换》，2025年。[50] Long Chen、Xiaotian Song 和 Yanan Sun。《LAS：面向完全脉冲驱动的大语言模型的无损 ANN-SNN 转换》，2025年。[51] Mark Horowitz。《1.1 计算能耗问题（以及我们能对此做些什么）》。载于 *2014* 年 *IEEE* 国际固态电路会议，第 10–14 页，2014 年。[52] P. Marikkannu B. Annapoorani。《高能效、高可扩展混合加法器的性能测量》。计算机系统科学与工程，45(3)：2659–2672，2023年。

命题 1 的证明

证明。 ANN 到 SNN 转换算法中的转换误差主要源于低于阈值、无法产生相应脉冲的残余膜电位。

传统速率编码

在传统的人工神经网络到脉冲神经网络的转换中，神经元在每个时间步累积输入信号 $\mathbf{I}$：

$$\mathbf{m} \leftarrow \mathbf{m} + \mathbf{I}.$$

The 随后将膜电位与阈值 θ 进行比较。如果 $\mathbf{m} > \theta$，则发出一个脉冲，并且 d the 膜电位更新为：

$$\mathbf{m} \leftarrow \mathbf{m} - \theta.$$

经过 T 个累积步数后：

$$T \cdot \mathbf{I} = \theta \cdot N_{\text{spike}} + \mathbf{m}_{\text{rest}},$$

其中， N_{spike} 表示发射的脉冲总数。理想编码值为 $\mathbf{I}$，而实际解码值为：

$$\frac{\theta \cdot N_{\text{spike}}}{T} = \mathbf{I} - \frac{\mathbf{m}_{\text{rest}}}{T}.$$

因此，经过 T 个时间步后的编码误差为：

$$\varepsilon = \frac{\mathbf{m}_{\text{rest}}}{T}, \quad \mathbf{m}_{\text{rest}} < \theta \implies \varepsilon < \frac{\theta}{T}.$$

通过优化初始膜电位初始化，可保证 $\mathbf{m}_{\text{rest}} < \theta/2$ [20], ，从而得到更紧的界：

$$\varepsilon < \frac{\theta}{2T}.$$

提出的积分编码

本文提出的方法采用积分编码，其中输入信号 $\mathbf{I}$ 仅在第一个时间步施加：

$$\mathbf{m} = \mathbf{I}.$$

在后续时间步中，残余膜电位在不再注入额外输入的情况下被放大。令 n_i 表示第 i 个脉冲发放时对应的放大索引；相应的脉冲幅度为 $\theta/2^{n_i}$。经过 n 次放大操作后，输入可分解为：

$$\mathbf{I} = \sum_i \frac{\theta}{2^{n_i}} + \frac{\mathbf{m}_{\text{rest}}}{2^n},$$

其中，第二项表示残余编码误差。因此，编码误差为：

$$\varepsilon = \frac{\mathbf{m}_{\text{rest}}}{2^n}, \quad \mathbf{m}_{\text{rest}} < \theta \implies \varepsilon < \frac{\theta}{2^n}.$$

应用进位机制后，误差进一步减半：

$$\varepsilon < \frac{\theta}{2^{n+1}}.$$

这里， n 表示放大操作的总次数。当 $T = 1$ 时，有 $n = D$。当 $T > 1$ 时，为满足在脉冲发放前$\mathbf{m}_{\text{rest}} > \theta$ 的条件，需要在时间步之间执行额外的放大操作，从而得到：

$$n \geq T \cdot D,$$

且误差有如下界限：

$$\varepsilon < \frac{\theta}{2^{T \cdot D + 1}}.$$

□

B 能耗分析

人工神经网络到脉冲神经网络的转换所带来的能量节省主要源于两个方面：（1）以轻量级加法运算取代浮点乘法；（2）通过所提出的脉冲编码方案，降低层与层之间的比特传输成本。本节将对这两个方面进行详细的定量分析。

B.1 计算能耗

基于所提出的脉冲采集方法，所有乘法运算 $w \times x$ 均被完全分解为整数级加法和位移运算。作为参考，已有研究表明，FP32 乘法的能耗为 3.7 皮焦耳，FP16乘法的能耗为 1.1皮焦耳 [51],，具体汇总于表4中。

表4：常见硬件运算的能耗

整数运算		**浮点运算**		**内存访问（64 位）**	
操作	能量	操作	能量	内存	能量
8位加法	0.03 皮焦耳	16位浮点加法	0.4 皮焦耳	8 KB缓存	10 皮焦耳
32位加法	0.1 皮焦耳	32位浮点加法	0.9 皮焦耳	32 KB缓存	20 皮焦耳
8 位乘法	0.2 皮焦耳	16 位浮点乘法	1.1 皮焦耳	1 MB 缓存	100 皮焦耳
32 位乘法	3.1 皮焦耳	32 位浮点乘法	3.7 皮焦耳	DRAM	1.3–2.6 纳焦耳

Taking $T = 2$, $D = 3$ 作为代表性配置（如图 3 所示，其中包含 $x = 1.4$、$w = 1.6$), the $x \times w$ 的计算过程如下。

步骤 1：从 x中进行脉冲提取。

- 周期 *1*：脉冲由尾数的隐含前导 1 与前 $D - 1 = 2$ 个小数位拼接而成。
- 周期 *2*：确定剩余 8 个尾数位中首个置位的位置 Tr_2（此处为 $\mathrm{Tr}_2 = 1$），得到原始脉冲 100。随后执行进位操作：100 后紧接的位为 1，因此将脉冲修正为 $100 + 1 = 101$。
- 备注：由于按照构造，每个脉冲的首位始终为 1，因此无需显式传输；只需传输末尾的 $D - 1$ 位，从而进一步降低层间比特传输成本。

步骤 2：指数和符号位处理。结果的符号位通过对 w 和 x的符号位执行异或运算获得。结果的指数通过将 w 和 x的指数域相加，再减去偏置 15 计算得到。随后，根据每个周期的脉冲值，对 w 的小数部分进行并行移位和累加：

- 脉冲 = 101：将 w 的尾数与其右移 2 比特后的结果相加。
- 脉冲 = 100：保持 w 的尾数不变。
- 脉冲 = 110：将 w 的尾数与其右移 1 比特后的结果相加。
- 脉冲 = 111：计算左移后的尾数与其右移 2 比特后的结果之差（可通过二进制补码加法实现）。

步骤 3：跨周期累加。将周期 1 的小数结果与周期 2 右移（$\mathrm{Tr}_2 - 1$）比特后的结果相加，同时考虑进位操作引入的任何阈值缩放。最终结果 $w \times x$ 通过拼接符号位、指数和累加尾数的前 11 个比特构成。

能耗估算依据

按照既定的硬件缩放规则，能耗随操作数位宽的增长呈次线性变化：位宽翻倍通常使能耗增加不到 $2\times$ [52]。据此，INT16 加法估计为 0.05–0.06 皮焦耳（保守取 0.06 皮焦耳），而符号位异或的能耗上限按 INT8 加法计算，为 0.03 皮焦耳。

首周期能耗

(1) **符号位异或**：0.03 皮焦耳。(2) **指数求和**：对指数位求和并减去偏置，能耗为 0.06 皮焦耳。(3) **脉冲–尾数乘法**：*D=1*：直接保留尾数；0 皮焦耳。*D=2*：以 $1/2$ 的概率（脉冲=10），直接保留（0 皮焦耳）；以 $1/2$ 的概率（脉冲=11），进行一次 INT16 加法。平均能耗：$(0 + 0.06)/2 = 0.03$皮焦耳。加上符号位和指数部分后总计：0.045 皮焦耳。*D=3*：以 $1/4$ 的概率（脉冲=100），能耗为 0 皮焦耳；以 $1/2$ 的概率（脉冲=101,110），进行一次 INT14 加法（≈ 0.06皮焦耳）；以 $1/4$ 的概率（脉冲=111），进行基于补码的减法（估算为 $1.5\times$ 次 INT 加法）。平均能耗：0.075 皮焦耳。(4) **舍入**：每次乘法结束时对尾数进行一次舍入操作，能耗为 0.03 皮焦耳，与周期 1 重叠。

后续周期能耗（$T \geq 2$）

(1) **前导 1 检测**：定位首个置位并记录偏移量 p 的能耗可忽略；其上限为 INT8 加法的能耗，即 0.03 皮焦耳。(2) **脉冲–尾数乘法**：规则与周期 1 相同；$D = 1, 2, 3$ 对应的平均能耗分别为 0 皮焦耳、0.045 皮焦耳和 0.075 皮焦耳。(3) **跨周期累加**：右移 ($D + Tr_2 - 1$) 比特后进行加法。操作数位宽最大为 INT24（$D = 2$）和 INT25（$D = 3$），二者均截断为 INT16 加法：$0.03 + 0.06 = 0.09$皮焦耳。

所有后续周期（$T > 2$）的能耗要求均与周期 2 相同。代表性配置（T, D）的总能耗见表 5。

表 5：不同（T, D）配置相对于 FP32 的总能耗和能耗比

能量（皮焦耳）	T=1	T=2	T=3
D=1	0.090	0.165	0.195
D=2	0.210	0.330	0.390
D=3	0.330	0.495	0.585
FP16（人工神经网络）		1.100	
FP32（人工神经网络）		3.700	
能耗比（脉冲神经网络/FP32）	T=1	T=2	T=3
D=1	0.024	0.045	0.053
D=2	0.057	0.089	0.105
D=3	0.089	0.134	0.158

B.2 层间比特传输成本评估

在传统方案中，层间比特传输成本取决于数据类型：采用 FP32 格式时，每次乘法计算都需要传输 32 比特数据；采用 FP16 或 BF16 格式时，每次乘法计算都需要传输 16 比特数据。本文提出的层间比特传输成本由三部分组成：1 比特符号信息、首个周期传输的 5 比特阈值信息和（$D - 1$）比特脉冲信息（默认首比特为 1，因此可以省略），以及后续

周期传输的（）比特脉冲信息（额外的 3 比特用于表示阈值偏移量 Tr_t，上限为 8 比特：000 表示 1，111 表示 8，超过 8 比特的偏移量将直接忽略）。不同配置下的详细传输量汇总于表6。如表6所示，

表6：不同指数位宽和参数配置下的层间比特传输成本

指数_维	D	$T=1$	$T=2$	$T=3$
5	1	6	9	15
	2	7	11	18
	3	8	13	21
4	1	5	8	14
	2	6	10	17
	3	7	12	20
FP32（人工神经网络）			32	
FP16 或 BF16（人工神经网络）			16	

所提出方案的传输位宽显著低于传统人工神经网络方案，从而进一步降低数据传输过程中的能耗。

C 特殊情况：脉冲 = 111

当 $D = 3$ 且脉冲为 $= 111$时，朴素的移位加法形式需要三项：

$$[1, F_w] \cdot 111_2 = [1, F_w] + ([1, F_w] \gg 1) + ([1, F_w] \gg 2), \tag{10}$$

这将朴素地需要一个两级加法器树。然而，这种情况可以采用更高效的分解方式。注意到 $111_2 = 1000_2 - 0001_2$，并且乘以 1000_2 和 0001_2 分别对应于左移 1 比特数和右移 2 比特数（相对于隐含的最高位），我们得到：

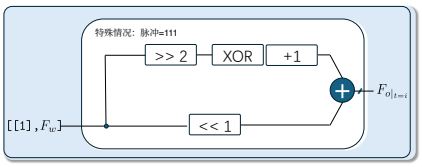

图4：输入脉冲等于 "111" 的特殊情况

$$[1, F_w] \cdot 111_2 = ([1, F_w] \ll 1) - ([1, F_w] \gg 2),$$

这只需要进行一次减法。在硬件中，减法通过二进制补码加法实现：对 $[1, F_w] \gg 2$ 取负可通过按位取反后加 1 来完成（将其吸收到加法器的进位输入中），从而得到：

$$[1, F_w] \cdot 111_2 = ([1, F_w] \ll 1) + \overline{([1, F_w] \gg 2)},$$

其中，$\ll$ 和 $\gg$ 分别表示左逻辑移位和右逻辑移位，而 $\overline{(\cdot)}$ 表示二进制补码取负。如图4所示，数据通路由下方路径上的左移 1 位单元、上方路径上依次连接的右移 2 位单元和二进制补码单元，以及一个合并两路结果的加法器组成。

这种分解确保，对于具有 $D \leq 3$ 的所有脉冲模式，每次时间步内计算**最多只需一个加法器**，且不存在需要执行超过一次加法或减法运算的情况。因此，无论脉冲值如何，硬件开销都受到严格限制，这证实了所提设计的面积效率。

D 单步与多步执行模式

在人工神经网络到脉冲神经网络的转换中，可以根据时间调度策略区分两种执行模式：单步模式 和 多步模式。

单步模式。 在单步模式下，每个时间步的输入数据都会依次逐层传播 through整个网络，然后才开始下一个时间步。也就是说，在时间步 t，脉冲序列会先从输入到输出经过所有层的完整处理，然后才开始计算时间步 $t + 1$。

多步模式。 在多步模式下，执行顺序则相反：每一层都会先完成所有 T 个时间步的计算，再进入下一层。具体而言，第 ℓ 层会处理所有时间步 $t = 1$，2、..... T的输入，然后才将累积结果传递给第 $\ell + 1$ 层。

RMPA 的建议。 即使在未采用硬件级优化的情况下部署 RMPA，我们仍建议采用多步模式，理由如下：

• **硬件兼容性**。 多步模式与当代硬件加速器的数据流模式更加自然地契合。在这类加速器中，最大限度地减少内存访问开销对于维持稳定的吞吐量至关重要。

• **减少权重重新加载**。 多步模式会在进入下一层之前完成当前层的所有时间步，因此无需在每个时间步反复重新加载相同的突触权重，从而显著降低内存带宽压力。

• **提升并行性**。 多步模式能够自然地展现跨时间步的层内并行性，从而更高效地利用向量化或批量计算资源。

使 RMPA 适配单步模式。 如果部署场景要求采用单步模式，则需要对 RMPA 进行小幅修改。具体而言，在第二个及之后的所有时间步（$t \geq 2$），当新生成的脉冲驱动数据被传递至下游层时，必须通过纳入按 2^{i-1} 缩放的传入贡献来更新第 ℓ 层的膜电位 m_ℓ：

$$\mathbf{m}_\ell(t) \leftarrow V_\ell(t) + 2^{i-1} \cdot \mathbf{I}_\ell(t), \qquad t \geq 2, \tag{11}$$

其中，$\mathbf{I}_\ell(t)$表示在时间步 t到达层 ℓ 的输入增量， i 表示RMPA循环的次数。这种缩放确保累积的膜电位能够正确反映经过适当放大的值。

E Qeury转换后的其他实验结果

对于Transformer模型的脉冲转换，本文研究了查询（q）表示在计算效率与模型准确率之间的权衡。默认情况下，q保留为浮点形式，不进行脉冲转换，而仅对键（k）和值（v）应用脉冲转换。这种设计将q视为类权重参数，在保持模型准确率的同时，消除了注意力机制中的浮点乘加运算。为评估这一设计选择的影响，我们针对是否将q转换为脉冲开展了消融实验，结果见表7。

F ImageNet-1K上的其他实验结果

本节展示了我们的SNN转换框架在ImageNet-1K数据集上针对多种模型的全面实验结果，包括EVA02、ViT、ResNet和VGG。我们在表8中报告了不同时间步 T 和脉冲维度 D 下的准确率。人工神经网络基线准确率作为参考。

G CIFAR数据集上的实验结果

我们在不同参数组合下评估转换后的 ViT-S 模型在 CIFAR任务中的表现，并在图5中报告转换后的准确率。对于 CIFAR-10，在 $D = 1$ 条件下，我们的方法在 $T = 3$ 时达到**98.40%** 准确率，与人工神经网络基线（98.42%）非常接近，同时仅需 3 个时间步。值得注意的是，即使在 $T = 2$时，准确率仍达到 97.97%，仅下降 0.45%。对于 CIFAR-100，我们的方法在 $T = 3$ 时达到**89.46%**，并在 $T = 2$ 时达到 87.95%，同样接近人工神经网络的 89.81% 基线。

我们进一步观察到，更深的网络能够实现更快的收敛。在 $D = 3$ 条件下，CIFAR-10 上的脉冲神经网络在 $T = 1$ 时就已达到 98.05%，并在 $T = 3$ 时饱和至 98.37%，基本与人工神经网络相当。同样，在 CIFAR-100 上，$D = 3$ 在 $T = 1$ 时达到 88.59%，并在 $T = 2$ 时达到 89.82%，将与人工神经网络的差距缩小至 0.06% 以内。这些结果表明，我们的方法几乎实现

表 7：不同（T， D）场景下 ViT 模型的实验。q_include 表示一种变体，除 k 和 v 外，还会将查询（q）转换为脉冲；而 Default 仅转换 k和 v。

Task	模型	ANN	D	默认			q_include		
				T=1	T=2	T=3	T=1	T=2	T=3
ImageNet-1K	vit_small_patch16_224	78.04	1	48.20	76.27	77.80	42.34	76.25	77.87
			2	73.78	77.94	78.03	73.26	77.93	78.04
			3	77.01	78.06	78.03	76.94	78.06	78.03
	ViT_基础版_图块16_224	82.91	1	74.00	82.21	82.74	72.37	82.17	82.73
			2	80.89	82.77	82.89	80.71	82.76	82.87
			3	82.41	82.87	82.91	82.43	82.87	82.92
	ViT_大型版_图块16_224	84.88	1	80.89	84.46	84.79	80.26	84.43	84.79
			2	84.03	84.80	84.89	83.96	84.81	84.89
			3	84.45	84.89	84.88	84.41	84.89	84.88
MMLU	通义千问3_0.6b_基础版	52.41	1	29.06	50.29	52.11	25.58	49.47	52.31
			2	45.85	52.55	52.56	43.40	52.24	52.73
			3	51.08	52.60	52.49	50.11	52.60	52.60
	通义千问3_1.7b_基础版	62.61	1	41.59	60.80	62.37	31.94	60.12	62.47
			2	57.83	62.42	62.53	55.62	62.34	62.78
			3	60.87	62.26	62.50	60.93	62.63	62.56
	通义千问3_4b_基础版	72.90	1	57.58	72.34	72.70	43.85	71.57	72.72
			2	69.29	72.75	73.10	67.40	72.62	73.07
			3	72.21	72.92	73.04	71.85	73.00	73.04
	通义千问3_8b_基础版	76.85	1	65.26	76.13	76.86	54.42	75.97	76.51
			2	74.88	76.71	76.78	73.06	76.59	76.81
			3	76.45	76.90	76.81	75.99	76.81	76.73
	通义千问3_14b_基础版	80.47	1	70.70	79.73	80.39	62.08	79.58	80.37
			2	78.63	80.47	80.45	77.43	80.42	80.65
			3	80.37	80.49	80.49	79.83	80.40	80.44

表8：不同 T和 D 场景下 ImageNet-1K 数据集上的准确率（%）

模型	$T=1$	$T=2$	$T=3$	人工神经网络
EVA02_tiny_patch14_336				
$d=1$	73.26	80.11	80.53	80.62
$d=2$	78.82	80.60	80.65	
$d=3$	80.19	80.63	80.61	
EVA02_small_patch14_336				
$d=1$	80.18	85.22	85.68	85.73
$d=2$	84.54	85.74	85.75	
$d=3$	85.45	85.70	85.73	
EVA02_base_patch14_448				
$d=1$	81.53	87.85	88.25	88.26
$d=2$	86.88	88.28	88.26	
$d=3$	88.00	88.27	88.26	
EVA02_大型_补丁14_448				
$d=1$	85.66	89.32	89.57	89.63
$d=2$	88.88	89.60	89.63	
$d=3$	89.46	89.63	89.63	
EVA_巨型_补丁14_336				
$d=1$	88.65	89.41	89.57	89.62
$d=2$	89.27	89.56	89.60	
$d=3$	89.47	89.58	89.62	

模型	$T=1$	$T=2$	$T=3$	人工神经网络
ViT_小型_补丁16_224				
$d=1$	48.20	76.27	77.80	78.04
$d=2$	73.78	77.94	78.03	
$d=3$	77.01	78.06	78.03	
ViT_基础_补丁16_224				
$d=1$	74.00	82.21	82.74	82.91
$d=2$	80.89	82.77	82.89	
$d=3$	82.41	82.87	82.91	
ViT_大型_补丁16_224				
$d=1$	80.89	84.46	84.79	84.88
$d=2$	84.03	84.80	84.89	
$d=3$	84.45	84.89	84.88	
resnet34				
$d=1$	20.23	72.45	75.83	76.09
$d=2$	66.59	76.08	76.12	
$d=3$	74.81	76.13	76.09	
vgg16				
$d=1$	49.82	68.86	70.97	71.12
$d=2$	65.25	70.94	71.10	
$d=3$	69.72	71.08	71.12	

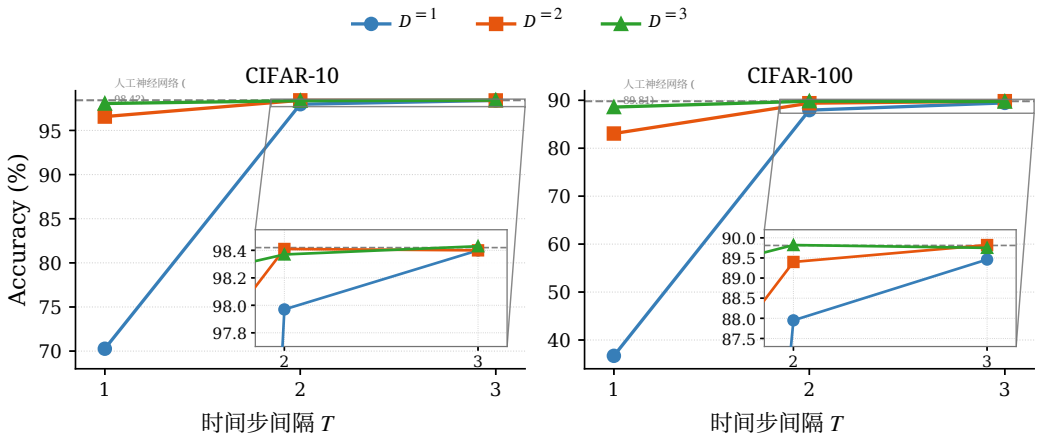

图5：在 T 和 D的不同场景下，ViT-S在CIFAR-10（左）和CIFAR-100（右）上的测试准确率。

仅需少至 $T = 2$ 个时间步即可实现无损转换，更深的架构进一步降低所需延迟。

H 大规模多任务语言理解基准上的实验结果

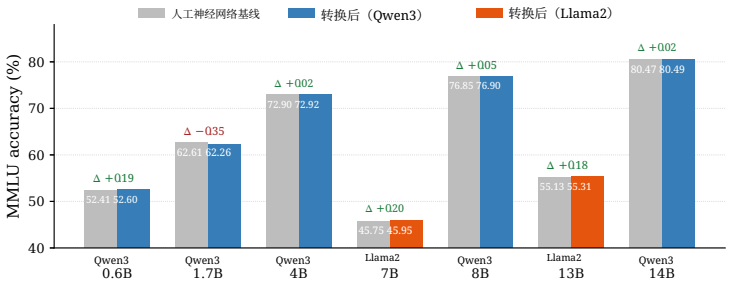

图 6：不同模型在 MMLU 上的转换结果

本文在 MMLU 基准测试上评估了通义千问3[42] 和 Llama2[43] 系列模型。我们对这一重要的自然语言处理基准进行了全面评估，结果如图 6 所示。

在 29 个评估项目中，只有 Winogrande 任务上的 Llama2-7B 模型和 MMLU 任务上的 Qwen3-1.7B-Base 模型出现了准确率下降，分别为 0.32% 和 0.35%。此外，Llama2-7B 模型在 ARC-E 和 HellaSwag_norm 任务上的准确率分别下降了 0.12% 和 0.1%。其余 25 项任务的准确率损失均低于 0.1%，这验证了所提出范式在计算机视觉和自然语言处理任务中的普适性和有效性。

I LLaMA2-7B 和 LLaMA2-13b 的其他实验结果

本节在表9中展示了我们的脉冲神经网络转换框架应用于 LLaMA2-7B 和 LLaMA2-13B 模型在各类自然语言处理任务上的全面实验结果。我们报告了不同时间步 T和脉冲维度 D下的准确率。人工神经网络基线准确率供参考。

表9：LLaMA2-7B 和 LLaMA2-13B 在不同 T 和 D 场景下自然语言处理任务的准确率（%）

	LLaMA2-7B				LLaMA2-13B			
Task	T=1	T=2	T=3	ANN	T=1	T=2	T=3	ANN
MMLU								
d=1	26.90	43.88	45.93	45.75	33.76	53.62	54.99	55.13
d=2	39.97	45.67	45.73		51.40	55.05	55.23	
d=3	44.92	45.95	45.93		54.15	55.31	55.14	
PIQA-规范化								
d=1	72.14	79.11	78.56	78.45	74.32	80.09	80.52	80.52
d=2	77.37	78.73	78.84		79.60	80.69	80.25	
d=3	78.78	78.67	78.51		80.25	80.58	80.47	
ARC-Challenge 规范化								
d=1	33.19	44.88	45.48	45.73	39.76	48.55	48.55	48.89
d=2	43.60	45.48	45.65		47.78	48.72	48.38	
d=3	46.16	45.65	45.99		49.40	49.15	48.55	
ARC-Easy 规范化								
d=1	58.33	73.61	74.75	74.87	66.79	75.93	76.60	76.56
d=2	72.26	73.99	74.54		75.51	76.26	76.47	
d=3	73.82	74.75	74.66		75.80	76.47	76.43	
Winogrande 归一化								
d=1	60.38	69.38	69.46	69.38	62.67	71.90	71.90	72.22
d=2	67.01	69.53	69.30		71.43	72.53	72.22	
d=3	67.56	69.06	69.46		70.56	72.14	72.53	
HellaSwag 归一化								
d=1	65.61	75.72	76.30	76.28	66.05	79.04	79.55	79.71
d=2	74.65	76.25	76.18		78.57	79.63	79.71	
d=3	75.77	76.18	76.25		79.23	79.69	79.72	

J Qwen3 系列模型的大规模多任务语言理解结果

本节在表10中补充了通义千问 Qwen3 系列在不同时间步T和脉冲维度 D 下的 MMLU 任务准确率，并以原始人工神经网络准确率作为基线参考。

表10：Qwen3 系列模型在不同 T 和 D 场景下的 MMLU准确率（%）

模型	T=1	T=2	T=3	**ANN**
qwen3-0.6b-base				
d=1	29.06	50.29	52.11	52.41
d=2	45.85	52.55	52.56	
d=3	51.08	52.60	52.49	
Qwen3-1.7B-Base				
d=1	41.59	60.80	62.37	62.61
d=2	57.83	62.42	62.53	
d=3	60.87	62.26	62.50	
qwen3-4b-base				
d=1	57.58	72.34	72.70	72.90
d=2	69.29	72.75	73.10	
d=3	72.21	72.92	73.04	
qwen3-8b-base				
d=1	65.26	76.13	76.86	76.85
d=2	74.88	76.71	76.78	
d=3	76.45	76.90	76.81	
qwen3-14b-base				
d=1	70.70	79.73	80.39	80.47
d=2	78.63	80.47	80.45	
d=3	80.37	80.49	80.49	

NeurIPS 论文清单

该清单旨在鼓励负责任的机器学习研究最佳实践，关注可复现性、透明度、研究伦理和社会影响等问题。请勿删除清单：**未附清单的论文将被直接拒稿**。清单应置于参考文献之后，并放在（可选的）补充材料之前。清单不计入页数限制。

请仔细阅读清单指南，了解如何回答这些问题。对于清单中的每个问题：

- 您应回答 [是], [否],或 [不适用]。
- [不适用] 表示该问题不适用于这篇论文，或相关信息不可获得。

- 请在答案后立即附上简短说明（1–2 句话），即使回答为 [不适用]也应如此。

清单的回答是论文投稿不可或缺的一部分。 审稿人、领域主席、高级领域主席和伦理审稿人都可以看到这些回答。您还需要在论文最终版本中（经过可能的修改后）附上这份清单，其最终版本将与论文一同发表。

论文审稿人会将清单作为评估因素之一。虽然 [是] 通常优于 [否], 但只要给出充分的理由，回答 [否] 也是完全可以接受的（例如，“由于计算开销过大，我们没有报告误差条”，或“我们无法找到所用数据集的许可证”）。一般而言，回答[否] 或 [不适用] 并不会成为拒稿的理由。虽然问题采用了二选一的表述方式，但我们承认，实际情况往往更加微妙，因此请自行判断，并撰写理由加以说明。所有支持性证据都可以放在论文正文或补充材料中，也可以放在附录中。如果您对某个问题回答 [是]，请在理由中指出可以找到相关材料的章节。

1. **主张**
   问题：摘要和引言中的主要主张是否准确反映了论文的贡献和范围?

   回答： [是]
   理由：摘要和引言明确陈述了主要贡献，包括 PRIME 框架、RMPA 机制、硬件友好型推理方案，以及在计算机视觉和自然语言处理任务上的实验结果。

   指南：
   - 该回答 [不适用] 意味着摘要和引言未包含论文中的相关主张。

   - 摘要和／或引言应清楚陈述所提出的主张，包括论文的贡献以及重要的假设和局限性。 [否]或[不适用] 对这个问题的回答会给审稿人留下不好的印象。

   - 所提出的主张应与理论和实验结果相符，并反映这些结果在多大程度上可望推广到其他场景。
   - 将愿景目标作为动机是可以的，但应明确说明论文并未实现这些目标。

2. **局限性**
   问题：这篇 论文是否讨论了所开展工作的局限性 由作者提出?
   答案： [是]
   理由：在第 5 节中，我们讨论了这项工作的局限性：与大多数算法层面的研究一样，此处报告的效率指标基于分析估算，而非直接进行芯片上测量。不过，鉴于所有底层计算基元（位移、整数加法、前导 1 检测）均受标准数字逻辑原生支持，物理实现仍是自然而直接的下一步。

指南：

- 回答 [不适用] 表示论文没有局限性，而回答 [否] 表示论文存在局限性，但论文中未讨论这些局限性。
- 建议作者在论文中单独设置“局限性”一节。
- 论文应指出任何强假设，并说明结果对这些假设遭到违反的情况有多稳健（例如，独立性假设、无噪声设置、模型设定正确性，以及仅在局部成立的渐近近似）。作者应思考这些假设在实践中可能如何被违反，以及这会带来什么影响。

- 作者应思考所提出主张的范围，例如，该方法是否仅在少数数据集上或经过少数几次运行测试。一般而言，经验结果往往依赖于隐含假设，应明确阐述这些假设。

- 作者应思考影响该方法性能的因素。例如，当图像分辨率较低或图像是在低照明环境下拍摄时，人脸识别算法可能表现不佳。又如，语音转文本系统可能无法可靠地用于为在线讲座生成隐藏式字幕，因为它无法处理技术术语。

- 作者应讨论所提出算法的计算效率，以及其如何随数据集规模增长而扩展。

- 如果适用，作者应讨论其方法可能存在的局限性，以应对隐私和公平性问题。

- 作者或许担心，完全坦诚地说明局限性可能会被审稿人作为拒稿的理由，但更糟糕的情况可能是，审稿人发现论文中未提及的局限性。作者应审慎判断，并认识到，个人采取行动支持透明度，对于形成维护社区诚信的规范至关重要。审稿人将被特别告知，不得因作者坦诚说明局限性而予以惩罚。

3. **理论假设与证明**

   问题：对于每个理论结果，论文是否给出了完整的假设集以及完整且正确的证明?

   回答： [是]

   理由：误差分析部分给出了常规频率编码和提出的积分编码的完整理论误差界。请参见附录A。

   指南：

   - 回答 [不适用]表示论文不包含理论结果。
   - 论文中的所有定理、公式和证明都应编号并相互引用。

   - 所有假设都应明确陈述，或在任何定理的陈述中注明其出处。
   - 证明可以出现在论文正文或补充材料中；但如果出现在补充材料中，建议作者提供简短的证明概要，以帮助读者直观理解。

   - 反过来，论文正文中提供的任何非形式化证明，都应辅以附录或补充材料中的形式化证明。
   - 应适当引用证明所依据的定理和引理。

4. **实验结果可复现性**

   问题：论文是否充分披露了复现论文主要实验结果所需的全部信息，以达到能够影响论文主要论点和/或结论的程度（无论是否提供代码和数据）？

   回答： [是]

   理由：实验部分和附录提供了充分的实验细节，包括数据集、模型架构和评估配置，以便复现主要结果。论文被接收后，源代码将公开发布。

指南：

- 答案 [不适用]表示论文不包含实验。
- 如果论文包含实验，[否] 将不会得到审稿人的认可：无论是否提供代码和数据，使论文具备可复现性都很重要。
- 如果贡献是数据集和/或模型，作者应描述为使其结果具备可复现性或可验证性而采取的步骤。
- 根据贡献的不同，可以通过多种方式实现可复现性。例如，如果贡献是新型架构，完整描述该架构可能就足够了；或者，如果贡献是特定模型及其实证评估，则可能有必要让他人能够使用相同的数据集复现该模型，或提供模型访问权限。一般而言，发布代码和数据通常是实现这一目标的一种好方法，但也可以通过详细说明如何复现实验结果、提供托管模型的访问权限（例如，对于大语言模型而言）、发布模型检查点，或采用其他适合所开展研究的方式来实现可复现性。
- 虽然 NeurIPS 不要求发布代码，但会议要求所有投稿都提供某种合理的可复现途径，具体方式可根据贡献的性质而定。例如：(a) 如果贡献主要是一种新算法，论文应明确说明如何复现该算法。(b) 如果贡献主要是一种新的模型架构，论文应清晰、完整地描述该架构。(c) 如果贡献是一种新模型（例如大语言模型），则应提供访问该模型的途径，以便复现结果；或者提供复现该模型的方法（例如使用开源数据集，或说明如何构建该数据集）。(d) 我们认识到，在某些情况下，可复现性可能难以实现；在这种情况下，欢迎作者说明其提供可复现性的具体方式。对于闭源模型，模型访问权限可能受到某些限制（例如仅限注册用户），但其他研究人员应当能够通过某种途径复现或验证结果。

5. **数据和代码开放获取**

问题：论文是否开放获取数据和代码，并提供充分的说明，以便忠实复现补充材料中所述的主要实验结果?

回答： [是]

理由：代码将在论文被接收后公开。所有使用的数据集均为标准公开基准。

指南：

- 回答 [不适用] 表示论文不包含需要代码的实验。
- 详情请参阅 NeurIPS 代码和数据投稿指南（https://neurips.cc/public/guides/CodeSubmissionPolicy）。
- 虽然我们鼓励发布代码和数据，但我们理解这可能无法实现，因此 [否] 是可以接受的回答。不能仅仅因为未包含代码而拒绝论文，除非代码是论文贡献的核心部分（例如，提出新的开源基准）。
- 说明中应包含用于运行实验以复现结果所需的确切命令和环境。详情请参阅 NeurIPS 代码和数据投稿指南（https: //neurips.cc/public/guides/CodeSubmissionPolicy）。
- 作者应提供有关数据访问和准备的说明，包括如何访问原始数据、预处理数据、中间数据和生成数据等。
- 作者应提供脚本，以复现新提出的方法和基线方法的所有实验结果。如果只有部分实验可复现，则应说明脚本中省略了哪些实验以及原因。

- 投稿时，为保留匿名性，作者应发布匿名版本（如适用）。

- 建议在补充材料（附于论文之后）中提供尽可能多的信息，但允许提供数据和代码的网址。

6. **实验设置/细节**

   问题：论文是否详细说明了理解实验结果所需的全部训练和测试细节（例如数据划分、超参数、超参数的选择方式以及优化器类型）？

   回答： [是]

   理由：这项工作提出了一种免训练ANN到SNN转换框架。实验部分对推理配置和评估设置进行了完整说明。

   指南：

   - 答案 [不适用]表示论文未包含实验。
   - 实验设置应在论文正文中说明，详细程度应足以帮助读者理解结果及其意义。

   - 完整细节可通过代码、附录或补充材料提供。

7. **实验统计显著性**

   问题：论文是否恰当地报告了误差条，并对其进行了正确定义，或提供了其他关于实验统计显著性的适当信息?

   回答： [否]

   理由：未报告误差条。在给定固定的预训练人工神经网络时，提出的方法是确定性的，且不同运行之间的结果方差可以忽略不计。

   指南：

   - 该答案 [不适用]表示论文未包含实验。
   - 作者应回答 [是] 如果结果附带误差条、置信区间或统计显著性检验，至少对于支持论文主要论断的实验应如此。

   - 应明确说明误差条所反映的变异因素（例如训练/测试集划分、初始化、某些参数的随机抽取，或在给定实验条件下的整体运行）。

   - 应解释误差条的计算方法（闭式公式、调用库函数、自助法等）。

   - 应说明所作的假设（例如，误差呈正态分布）。
   - 应明确误差条表示的是标准差还是均值标准误。

   - 报告1σ误差条是可以的，但应明确说明。若误差正态性的假设未经验证，作者最好报告2σ误差条，而不是声称置信区间为96%。

   - 对于非对称分布，作者应注意不要在表格或图中显示对称误差条，以免得出超出合理范围的结果（例如，错误率为负）。

   - 如果在表格或图中报告误差条，作者应在正文中说明其计算方法，并在正文中引用相应的图或表。

8. **实验计算资源**

   问题：对于每项实验，论文是否提供了足够的信息，说明重现实验所需的计算机资源（计算工作节点的类型、内存、执行时间）？

   回答： [是]

   理由：尽管本研究聚焦于人工神经网络到脉冲神经网络的转换方法，我们承认，硬件设备的差异可能会导致推理准确率出现微小偏差。因此，我们报告了实验中使用的推理设备。

指南：

- 该答案 [不适用]表示论文不包含实验。
- 论文应说明所使用的计算工作节点类型——中央处理器或图形处理器、内部集群或云服务提供商——并包括相关的内存和存储信息。
- 论文应说明每次单独实验运行所需的计算量，并估算总计算量。
- 论文应披露完整研究项目所需的计算量是否超过论文中报告的实验所需的计算量（例如，未纳入论文的初步实验或失败实验）。

9. **道德准则**

问题：论文中开展的研究是否在各个方面都符合 NeurIPS 道德准则 https://neurips.cc/public/EthicsGuidelines?

回答： [是]

理由：本工作关注节能神经网络推理，不涉及伦理问题。

指南：

- 回答 [不适用] 表示作者尚未审阅 NeurIPS 道德准则。
- 如果作者回答 [否], 则应说明需要偏离道德准则的特殊情况。
- 作者应确保保持匿名性（例如，若其所在司法管辖区的法律或法规要求特殊考虑）。

10. **更广泛影响**

问题：论文是否讨论了所开展工作潜在的正面社会影响和负面社会影响?

回答： [不适用]

理由：这是基础硬件效率研究，预计不会产生直接的负面社会影响。

指南：

- 该回答 [不适用] 意味着所开展的工作不会产生社会影响。
- 如果作者回答 [不适用] 或 [否], ，则应解释为什么其工作没有社会影响，或者为什么论文未讨论社会影响。
- 负面社会影响的例子包括潜在的恶意或意外用途（例如，虚假信息、生成虚假资料、监控）、公平性考量（例如，部署可能对特定群体作出不公平决策的技术）、隐私考量和安全性考量。
- 会议预计，许多论文属于基础研究，与特定应用无关，更不用说实际部署了。不过，如果存在通向任何负面应用的直接路径，作者应指出这一点。例如，指出生成模型质量的提升可能被用于生成深度伪造内容以传播虚假信息，是合理的。另一方面，则不必指出，用于优化神经网络的通用算法可能使人们能够更快地训练出生成深度伪造内容的模型。
- 作者应考虑以下可能造成的危害：技术按预期使用且正常运行时可能产生的危害；技术按预期使用但给出错误结果时可能产生的危害；以及因（有意或无意）滥用技术而造成的危害。
- 如果存在负面的社会影响，作者也可以讨论可能的缓解策略（例如，受控发布模型、在提供攻击方法的同时提供防御措施、监测滥用的机制、监测系统如何随着时间推移从反馈中学习的机制，以及提高机器学习的效率和可及性）。

11. **保障措施**

问题：论文是否描述了为负责任地发布滥用风险较高的数据或模型而采取的保障措施（例如预训练语言模型、图像生成器或抓取的数据集）？

回答： [不适用]

理由：论文未发布高风险模型或数据集。

指南：

- 回答 [不适用] 表示论文不存在此类风险。
- 发布存在较高误用或双重用途风险的模型时，应采取必要的保障措施以确保模型得到受控使用，例如要求用户遵守使用指南，或限制模型的访问权限，或实施安全过滤器。
- 从互联网抓取的数据集可能带来安全风险。作者应说明他们如何避免发布不安全的图像。
- 我们认识到，提供有效的保障措施颇具挑战，许多论文也没有这项要求，但我们鼓励作者对此予以考虑，并尽最大努力做到这一点。

12. **现有资源的许可证**

问题：论文中使用的资产（例如代码、数据、模型）的创作者或原始所有者是否都得到了适当署名？许可证和使用条款是否得到了明确说明并得到妥善遵守？

回答： [是]

理由：本文使用的所有数据集和代码库均已适当引用，并注明了其许可证。

指南：

- 答案 [不适用]表示论文未使用现有资源。
- 作者应引用生成该代码包或数据集的原始论文。
- 作者应说明所使用的资产版本，并尽可能提供 URL。
- 应注明每项资产的许可证名称（例如，知识共享署名 4.0）。
- 对于从特定来源（例如，网站）抓取的数据，应提供该来源的版权信息和服务条款。
- 如果资产已发布，应提供软件包中的许可证、版权信息和使用条款。对于热门数据集，paperswithcode.com/datasets 为部分数据集整理了许可证信息。其许可指南有助于确定数据集的许可证。
- 对于重新打包的现有数据集，应提供原始许可证以及衍生资源的许可证（如果已更改）。
- 如果无法在线获取此信息，建议作者联系资产的创建者。

13. **新资源**

问题：论文中引入的新资源是否有完善的文档说明，且是否随资源一并提供文档？

回答： [不适用]

理由：本文未发布新的数据集、模型权重或基准测试。

指南：

- 答案 [不适用] 表示该论文未发布新资产。
- 研究人员应通过结构化模板，在投稿中说明数据集/代码/模型的详细信息，包括训练、许可证、局限性等。
- 论文应讨论是否已获得资产使用者的同意，以及如何获得同意。

• 投稿时，请记得将您的资产匿名化（如适用）。您可以创建匿名化网址，或附上匿名化压缩文件。

14. **众包和涉及人类受试者的研究**

问题：对于众包实验和涉及人类受试者的研究，论文是否包含向参与者提供的说明全文（如适用，还包括截图），以及补偿详情（如有）？

答案： [不适用]

理由：本文不涉及众包或人体受试者研究。

指南：

• 答案 [不适用] 表示该论文不涉及众包，也不涉及人类受试者的研究。

• 将这些信息纳入补充材料是可以的，但如果论文的主要贡献涉及人体受试者研究，则应在论文正文中尽可能详细地说明。

• 根据 NeurIPS 道德准则，参与数据收集、整理或其他劳动的工作人员，获得的报酬至少应达到数据收集者所在国家的最低工资标准。

15. **涉及人类受试者的研究所需的机构审查委员会（IRB）批准或同等批准**

问题：论文是否描述了研究参与者可能承担的潜在风险，是否向受试者披露了此类风险，以及是否获得了机构审查委员会（IRB）的批准（或根据贵国或贵机构要求所需的同等批准/审查）？

回答： [不适用]

理由：本文不涉及人体受试者研究。

指南：

• 答案 [不适用] 表示论文不涉及众包或人体受试者研究。

• 根据开展研究所在的国家，任何人体受试者研究都可能需要获得机构审查委员会批准（或同等批准）。如果您已获得机构审查委员会批准，应在论文中明确说明。

• 我们认识到，相关程序可能因机构和地区而有很大差异，并期望作者遵守 NeurIPS 道德准则及其所在机构的指南。

• 对于初次投稿，请勿包含任何可能破坏匿名性的信息（如适用），例如开展审查的机构。

16. **大语言模型使用声明**

问题：论文是否描述了大语言模型的使用，且大语言模型是本研究核心方法中的重要、原创或非标准组成部分？请注意，如果大语言模型仅用于撰写、编辑或格式化，且不会影响研究的核心方法论、科学严谨性或原创性，则无需声明。

回答： [不适用]

理由：大语言模型（例如 LLaMA）在本论文中仅作为所提出转换框架的评估目标出现，而不是核心方法论的组成部分。

指南：

• 答案 [不适用] 意味着本研究的核心方法开发不涉及作为任何重要、原创或非标准组成部分的大语言模型。

• 关于应描述或不应描述的内容，请参阅 NeurIPS 手册中的大语言模型政策。