通过潜在空间虚时间演化实现的物理启发的变换器量子态

Kimihiro Yamazaki,[1]、2、 * Itsushi Sakata、[3] Takuya Konishi、[1,3] 和 Yoshinobu Kawahara[1,3]
[1] *Graduate School of Information Science and Technology、The University of Osaka、 1-5 Yamadaoka、 Suita、 Osaka、 Japan.*[2] *Fujitsu Laboratories、 Fujitsu Limited、 4-1-1 Kamikodanaka、 Nakahara-ku、 Kawasaki、 Kanagawa、 Japan*[3] *Center for Advanced Intelligence Project、 RIKEN、 1-4-1 Nihonbashi、 Chuo-ku、 Tokyo、 Japan.*（日期：2026年2月4日）

神经量子态（NQS）是变分蒙特卡罗框架中的强大试探波函数，但其架构通常被视为黑箱。我们提出了一种具有物理透明性的框架，将神经量子态视为潜在空间虚时间演化的神经近似。这一观点表明，标准的基于变换器的神经量子态（TQS）架构对应于依赖潜在空间中虚时间的、缺乏物理动机的有效哈密顿量。在此基础上，我们引入物理启发的变换器量子态（PITQS）：通过在各层之间共享权重来强制使用静态有效哈密顿量，并借助特罗特–铃木分解提高传播精度，同时不增加变分参数的数量。对于受挫的 J_1-J_2 Heisenberg模型，我们的试探波函数在使用显著更少变分参数的情况下，达到了与当前最先进的TQS相当甚至更高的精度。本研究表明，将深度网络结构重新诠释为潜在冷却过程，可以实现更加具有物理依据、系统且紧凑的设计，从而弥合黑箱式表达能力与物理透明构造之间的差距。

arXiv:2602.03031v1 [cond-mat.dis-nn] 3 Feb 2026

引言。—由于多体关联的复杂性，准确求得量子多体系统，尤其是强关联系统的基态波函数，是现代物理学面临的一项重大挑战。近年来，神经量子态（NQS）已成为变分蒙特卡洛（VMC）中的一类强大变分拟设。尤其是，基于变换器的神经量子态（TQS）在一些最具挑战性的基准问题上达到了最先进的精度，其中尤以受挫海森堡模型为代表。

尽管取得了这些成功，现有的神经量子态架构，尤其是基于变换器的神经量子态（TQS），在很大程度上仍是对通用机器学习设计的改造，因此依然是黑箱拟设。因此，精度的提升往往依赖大幅过参数化，却难以深入揭示拟设如何以及为何能够达到这样的性能。黑箱特性也使系统性的架构改进变得困难，因为缺乏将架构变化与精度的可控提升联系起来的统一原则。

与变分拟设的发展并行，虚时演化 *(ITE)*为获得基态提供了一条由来已久且物理图像清晰的途径。虚时演化的基本原理是将任意初态 $|\Psi_0\rangle$冷却至基态 $|\Psi\rangle$，使其

$$|\Psi\rangle \propto e^{-\beta\hat{\mathcal{H}}}|\Psi_0\rangle, \tag{1}$$

ˆ其中 $\mathcal{H}$ 是哈密顿量， β 是足够大的虚时间 [9]。辅助场量子蒙特卡罗（AF-QMC）方法 [10, 11] 通过采用哈伯德–斯特拉托诺维奇变换来实现虚时演化。尽管这种方法能够对特定哈密顿量给出精确估计，但显式随机采样会受到严重的符号问题困扰，尤其是在受挫自旋和费米子系统中，这严格限制了其适用性。即使是近期基于深度玻尔兹曼机的精确虚时演化构造 [12] ，在受挫系统中本质上也继承了符号问题。因此，如何在一般受挫体系中稳健地实现神经网络量子态与虚时演化的无符号问题结合，仍是一项重大挑战。

在本快报中，我们通过提出一种基于潜在虚时演化 *(LITE)*的框架，弥合黑箱神经网络量子态与物理图像清晰的构造方法之间的差距。受变分蒙特卡洛方案中式 (1) 虚时间图像的启发，我们将神经网络量子态拟设表示为潜在空间中的冷却过程。在这一方案中，我们不直接使用裸物理哈密顿量 $\mathcal{H}$，而是用潜在空间中由神经网络参数化的可学习有效哈密顿量来描述 LITE。至关重要的是，这种变分形式使我们的方法本质上不受符号问题困扰。

从 LITE 的视角来看，现有的 TQS 架构 [4–7]实际上意味着冷却过程由一个具有虚时间依赖性的有效哈密顿量驱动。这种在每个演化步骤中的变化，对于基态冷却而言是一种不符合物理实际的冗余，并会导致严重的过参数化。为解决这一问题，我们提出物理启发的变换器量子态 *(PITQS)*。由于冷却过程自然由一个与虚时间无关的单一哈密顿量支配，我们通过在所有层之间共享权重来强制采用静态有效哈密顿量。

该框架为网络架构提供了直接的物理直觉，并凸显了现有 TQS 的一个根本局限。具体而言，一旦将 TQS 的每一层视为由有效哈密顿量中的非对易项生成的潜在空间虚时间传播子的近似——

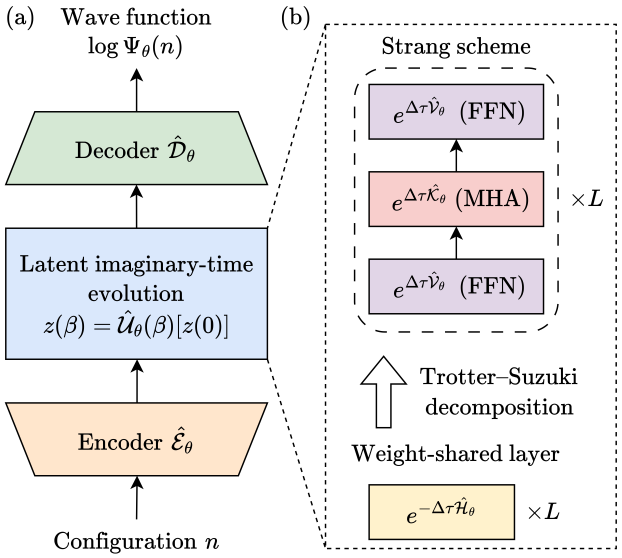

图 1. PITQS 概览。(a) 构型 n 经编码器 $\hat{\mathcal{E}}_\theta$ 映射为初始潜在态 $z(0)$，再由 LITE算符 $\hat{\mathcal{U}}_\theta(\beta)$演化为 $z(\beta)$，最后经 $\hat{\mathcal{D}}_\theta$ 解码得到波函数对数振幅 $\log \Psi_\theta(n)$。(b) 在 PITQS 中，$\hat{\mathcal{U}}_\theta(\beta)$由 L 权重共享层实现，以单个有效哈密顿量 $\hat{\mathcal{H}}_\theta$近似 $e^{-\Delta\tau\hat{\mathcal{H}}_\theta}$；每一步都采用特罗特–铃木分解（图示为斯特朗分解方案），交替应用非局部算子 $\mathcal{K}_\theta$ 和局部算子 $\hat{\mathcal{V}}_\theta$。

那么，标准 TQS更新便可自然地解释为一阶 Lie–Trotter分解 [13, 14]。与有限的虚时间步长会引入系统性的 Trotter误差。受此观察启发，我们系统地对静态有效哈密顿量实施高阶特罗特–铃木分解 [15–17]，从而在不增加变分参数数量的情况下提高传播子的精度。

我们展示了该框架在受挫的 J_1-J_2 海森堡模型上的能力；该模型是 NQS 领域公认的基准测试 [8, 18]。结果表明，静态有效哈密顿量具有足够的表达能力，能够捕捉复杂的量子关联，这说明传统 TQS 中大量的参数预算都是冗余的，因为传统 TQS 隐式采用了具有虚时间依赖性的有效哈密顿量。基于这一物理视角，我们明确设计了有效哈密顿量和铃木–Trotter 方案，在使用显著更少变分参数的同时，取得了与最先进基准测试相当、某些情况下甚至更优的精度。总体而言，我们的方法将 NQS 架构的设计从经验性启发式方法提升为一种系统的、以物理原理为依据的构造。

潜在空间虚时间演化。—本文提出一种框架，通过 LITE 构造神经网络量子态试探态，并在变分蒙特卡罗方案中对其进行优化。在离散构型基 $\{|n\rangle\}$ 中工作（对于 N 自旋-1/2 系统，n 表示希尔伯特空间的自旋构型），我们用含有变分参数 θ的多个深度神经网络对 $\Psi_\theta(n) = \langle n|\Psi_\theta\rangle$ 进行参数化，这些网络由三个逻辑模块组成：潜在空间编码器 $\hat{\mathcal{E}}_\theta$、LITE算符 $\hat{\mathcal{U}}_\theta(\beta)$和波函数解码器 $\hat{\mathcal{D}}_\theta$。总变分拟设定义为以下各部分的复合

$$\log \Psi_\theta(n) \;=\; \hat{\mathcal{D}}_\theta\Big[\hat{\mathcal{U}}_\theta(\beta)\big[\hat{\mathcal{E}}_\theta[n]\big]\Big] \in \mathbb{C}, \tag{2}$$

其中 β 表示潜在空间中的总虚时间（图 1(a)）。

首先，潜在空间编码器算符 $\hat{\mathcal{E}}_\theta$ 将离散构型 n 映射为潜在态 $z(0) \in \mathbb{R}^{N_{\text{tok}}\times d}$。这里，$z(0)$由 N_{tok} 个潜在标记 $z_i(0) \in \mathbb{R}^d$表示，其中 i 是潜在标记的索引，d 是标记维度。引入潜在标记的理念与在 AF-QMC 中引入辅助场类似 [10, 11]；它提供了额外的自由度，从而能够高效地表示非平凡关联。然而，与 AF-QMC 不同，我们的潜变量是可学习的，并在变分蒙特卡洛中进行端到端优化，而不是通过随机采样后再积分消去。因此，这种潜在空间视角提供了一种具有物理动机的方式，可在保留变分蒙特卡洛表述无符号问题特性的同时，扩展变分流形。

接着，$\hat{\mathcal{U}}_\theta(\beta)$在总虚时间 β 内执行 LITE 冷却，并在潜在空间中实现低能投影。在我们的方法中，LITE算符 $\hat{\mathcal{U}}_\theta(\beta)$由神经网络实现，我们将其写作

$$z(\beta) = \hat{\mathcal{U}}_\theta(\beta)\big[z(0)\big]. \tag{3}$$

将这一作用解释为潜在态 $z(\tau) \in \mathbb{R}^{N_{\text{tok}}\times d}$, $0 \le \tau \le \beta$ 会很方便，其中 $z(0) = \hat{\mathcal{E}}_\theta[n]$，且 $z(\beta) = \hat{\mathcal{U}}_\theta(\beta)[z(0)]$。在连续极限下，可以认为 $z(\tau)$ 遵循潜在虚时演化方程 $dz(\tau)/d\tau = -\mathcal{H}_\theta[z(\tau)],$，其解为

$$z(\beta) = e^{-\beta\hat{\mathcal{H}}_\theta}[z(0)]. \tag{4}$$

这里，$\mathcal{H}_\theta$ 表示一个由潜在空间中的神经网络参数化的有效哈密顿量。这与公式 (1) 中的虚时演化原则直接类似，不同之处在于，潜在态是在有效哈密顿量 $\hat{\mathcal{H}}_\theta$ 下演化，而不是在物理哈密顿量 $\mathcal{H}$ 下演化。

最后，波函数解码器 $\hat{\mathcal{D}}_\theta$ 将演化后的潜在态 $z(\beta)$ 映射为波函数对数振幅 $\log \Psi_\theta(n)$。

与现有 *TQS* 的关系。—我们从 LITE 的视角重新审视现有的 TQS 架构 [4–8]。在这一

视角下，现有的 TQS 可分为三个部分：将物理自由度映射为初始潜在表示的嵌入层；将最终潜在表示映射为变分多体波函数的波函数解码器；以及逐层潜在更新，该更新通过对潜在标记反复施加多头注意力（MHA）和前馈网络（FFN）变换来实现。

首先，我们确定潜在空间编码器和解码器。标准 TQS [4, 5, 8] 中使用的嵌入 $\mathcal{E}_\theta$对应于潜在空间编码器 n ：它将每个构型 z划分为局部块，并将其映射到 $\in \mathbb{R}^{N_{\rm tok}\times d}$ (0) $M\times M$。对于采用大小为 $b\times b$ 的方形块的二维 $N_{\rm tok}=$ 晶格，这会生成 M/b(2) [4, 5, 8] 个词元。在解码器一侧，标准 TQS 的形式如下：

$$\hat{\mathcal{D}}_\theta[z(\beta)]=\sum_{a=1}^{d}\log\cosh\big(w_a^\top\zeta+b_a\big), \tag{5}$$

其中，$\zeta = \sum_{i=1}^{N_{\rm tok}} z_i$(β) $\in \mathbb{R}^d$ 和 $\{w_a,b_a\}_{a=1}^{d}$ 是复数值变分参数。

接下来，聚焦于堆叠的 TQS 层，我们可以将由多头注意力和前馈网络驱动的更新解读为 LITE。我们将式（4）中的 $\tau\in[0,\beta]$ 划分为 L个切片，每个切片的大小为 $\Delta\tau=\beta/L$，并引入 $\tau_\ell=\ell\Delta\tau$，再定义离散潜在态 $z^{(\ell)}\equiv z(\tau_\ell)$（$\ell=0$、...、...、$L$），使得 $z^{(0)}=z$(0) 且 $z^{(L)}=z(\beta)$。按照这一记法，第 ℓ层将潜在态从 $z^{(\ell-1)}$ 更新为 $z^{(\ell)}$，可以视为对作用于潜在空间中的短虚时传播子的神经网络近似。

一般而言，量子多体哈密顿量 $\mathcal{H}$ 可以ˆ写成非局域项 $\mathcal{K}$ 与局域ˆ（位点）项 $\mathcal{V}$之和。受这一结构启发，我们将 TQS 层中的两个子映射解释为潜在空间中类似的分解：MHA 子层通过聚合所有其他标记来更新每个潜在标记，从而实现长程标记间耦合，并发挥有效非局域相互作用项的作用；而 FFN 子层对每个标记独立应用相同的非线性映射，因此发挥有效位点项的作用。具体而言，将第 ℓ层中的 MHA 和 FFN 残差更新分别记为映射 ˆ $\mathcal{K}_\theta^{(\ell)}$ 和 ˆ$\mathcal{V}_\theta^{(\ell)}$，我们将潜在空间中的第 ℓ 层有效哈密顿量定义为

$$\hat{\mathcal{H}}_\theta^{(\ell)}\equiv-\Big(\hat{\mathcal{V}}_\theta^{(\ell)}+\hat{\mathcal{K}}_\theta^{(\ell)}\Big), \tag{6}$$

其中，负号的选择使得虚时间演化冷却对应于应用 LITE算符。根据此定义，ˆ$\mathcal{H}_\theta^{(\ell)}$ 具有量子多体哈密顿量的典型结构，即由一个局域项和一个非局域项构成，因此可以清楚地看出，TQS 会在潜在空间中自然地诱导出一个有效哈密顿量，

这种形式非常适合表示量子多体系统。

现有 TQS 的每一层都依次应用 MHA 和 FFN，这可以自然地解释为短虚时传播子的一级 Lie–Trotter 分解 [13, 14] ；对于第 ℓ层中的单步演化，

$$e^{-\Delta\tau\hat{\mathcal{H}}_\theta^{(\ell)}}=e^{\Delta\tau(\hat{\mathcal{V}}_\theta^{(\ell)}+\hat{\mathcal{K}}_\theta^{(\ell)})}\;\approx\;e^{\Delta\tau\hat{\mathcal{V}}_\theta^{(\ell)}}\,e^{\Delta\tau\hat{\mathcal{K}}_\theta^{(\ell)}}. \tag{7}$$

由于 $[\hat{\mathcal{K}}_\theta^{(\ell)},\hat{\mathcal{V}}_\theta^{(\ell)}]\neq 0$，该分解会产生局部标度为 $\mathcal{O}($ $\Delta\tau^2$) 的特罗特误差。用欧拉型更新近似每个分量，便得到具体的层更新

$$\begin{aligned}\tilde{z}^{(\ell)}&=e^{\Delta\tau\hat{\mathcal{K}}_\theta^{(\ell)}}[z^{(\ell-1)}]\approx z^{(\ell-1)}+\Delta\tau\,\hat{\mathcal{K}}_\theta^{(\ell)}[z^{(\ell-1)}],\\ z^{(\ell)}&=e^{\Delta\tau\hat{\mathcal{V}}_\theta^{(\ell)}}[\tilde{z}^{(\ell)}]\approx\tilde{z}^{(\ell)}+\Delta\tau\,\hat{\mathcal{V}}_\theta^{(\ell)}[\tilde{z}^{(\ell)}],\end{aligned} \tag{8}$$

其中 $\tilde{z}^{(\ell)}$ 表示在第 ℓ层内施加非局域更新后的中间隐状态。这与每一层内标准 TQS 的层序（多头注意力 $\to$ 前馈网络）一致。最后，堆叠 L 层将 LITE算符实现为短虚时传播子的有序乘积，

$$\hat{\mathcal{U}}_\theta(\beta)\;\approx\;T_\tau\prod_{\ell=1}^{L}e^{\Delta\tau\hat{\mathcal{V}}_\theta^{(\ell)}}\,e^{\Delta\tau\hat{\mathcal{K}}_\theta^{(\ell)}}, \tag{9}$$

其中 T_τ 表示潜在空间中的虚时编序积。

重要的是，在传统张量量子态架构中， [4, 5,8], 各层之间不共享权重，因此所诱导的隐空间算符随 ℓ变化：对于 $\ell\neq\ell'$，一般有 ˆ$\mathcal{H}_\theta^{(\ell)}\neq$ ˆ$\mathcal{H}^{(\ell')}$ θ 。等价地，标准深度-L TQS 实现了依赖于隐虚时间的有效哈密顿量。然而，从物理角度看，变分蒙特卡洛的目标问题由一个ˆ单一的、与虚时间无关的哈密顿量 $\mathcal{H}$所规定，而虚时间演化冷却是一个自治过程。因此，一旦将堆叠的TQS层解释为实现 LITE，ˆ$\mathcal{H}_\theta^{(\ell)}$ 的逐层变化便构成一种物理上冗余的自由度，从而导致显著的过参数化：变分参数的数目随 L 线性增长，尽管底层机制仅由单一哈密顿量支配。

物理启发的变换器量子态。——基于这一观点，我们引入PITQS，这是一种对ˆ $\mathcal{U}_\theta(\beta)$的物理启发构造，它强制采用一个与虚时间无关的单一有效哈密顿量，并通过高阶特罗特–铃木分解系统地提高传播精度（图1(b)）。具体而言，我们不采用标准 TQS中隐含的层依赖有效哈密顿量 $\{\hat{\mathcal{H}}_\theta^{(\ell)}\}_{\ell=1}^{L}$ ，而是施加一个单一的权重共享的有效哈密顿量

$$\hat{\mathcal{H}}_\theta=-\big(\hat{\mathcal{V}}_\theta+\hat{\mathcal{K}}_\theta\big), \tag{10}$$

表 I. 使用 Lie–Trotter 方案，通过 PITQS 和标准 TQS 获得的每格点能量 E。括号内为统计（蒙特卡洛）误差。我们还报告了各设置对应的总虚时间 β 以及变分参数数量 N_p。

β	E (TQS)	N_p (TQS)	E (PITQS)	N_p (PITQS)
1.0	$-0.49593(11)$	81,800	$-0.49596(14)$	44,890
2.0	$-0.49652(9)$	155,620	$-0.49669(9)$	44,890
3.0	$-0.49656(11)$	229,440	$-0.49676(10)$	44,890
4.0	$-0.49651(10)$	303,260	$-0.49696(9)$	44,890

表 II。在 $\beta = 2.0$ 处，若干 PITQS 方案与标准 TQS 基线（$\beta = 0.5$ 和 2.0）的每格点能量。N_p 表示变分参数的数量。

方案	每格点能量	N_p
TQS（$\beta = 0.5$）	$-0.49318(15)$	44,890
TQS（$\beta = 2.0$）	$-0.49652(9)$	155,620
PITQS（Lie–Trotter）	$-0.49669(9)$	44,890
PITQS（Strang）	$-0.49671(9)$	44,890
PITQS（铃木）	$-0.49697(9)$	44,890
PITQS（Blanes–Moan）	$-0.49683(7)$	44,890

并近似LITE算符，

$$\hat{\mathcal{U}}_\theta(\beta) \;=\; \left(e^{-\Delta\tau\,\hat{\mathcal{H}}_\theta}\right)^L = e^{-\beta\hat{\mathcal{H}}_\theta}. \tag{11}$$

这恰好对应于在潜在空间中实现式 (1) 的虚时演化。至关重要的是，跨层的权重共享约束消除了标准 TQS [式 (9)]中存在的、物理上冗余的逐层变化，并将变分参数的数量减少了约 L倍。

此外，LITE 的视角为超越一阶 Lie–Trotter 分解提供了一条系统化路径。我们将单步传播算符替换为 m阶特罗特–铃木分解，

$$e^{-\Delta\tau\hat{\mathcal{H}}_\theta} \approx \prod_{i=1}^{k} \exp\!\left(a_i^{(m)}\Delta\tau\,\hat{\mathcal{V}}_\theta\right)\,\exp\!\left(b_i^{(m)}\Delta\tau\,\hat{\mathcal{K}}_\theta\right), \tag{12}$$

其中，k 是阶段数，$\{a_i^{(m)}, b_i^{(m)}\}$ 是定义 m阶分解的固定系数，其局部 Trotter误差为 $\mathcal{O}(\Delta\tau^{m+1})$，并满足 $\sum_i a_i^{(m)} = \sum_i b_i^{(m)} = 1$。例如，二阶 Strang（$m$ =2）分解 [15] 的形式为

$$e^{-\Delta\tau\hat{\mathcal{H}}_\theta} \;\approx\; e^{\frac{\Delta\tau}{2}\hat{\mathcal{V}}_\theta}\, e^{\Delta\tau\hat{\mathcal{K}}_\theta}\, e^{\frac{\Delta\tau}{2}\hat{\mathcal{V}}_\theta}, \tag{13}$$

局部误差为 $\mathcal{O}(\Delta\tau^3)$。此外，在我们的实验中，我们采用了两种四阶方案，分别归因于铃木 [16] 和 Blanes–Moan [17]。重要的是，提高分解阶数只会改变这些固定系数，不会引入额外的变分参数，从而在不扩大参数预算的情况下提高潜空间虚时间传播的精度。

数值结果——接下来，我们使用方格晶格 J_1-J_2 海森堡模型验证我们的框架，其哈密顿量为

$$\hat{\mathcal{H}} = J_1 \sum_{\langle i,j\rangle} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j + J_2 \sum_{\langle\langle i,j\rangle\rangle} \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_j, \tag{14}$$

其中 $\hat{\mathbf{S}}_i = (\hat{S}_i^x,\ \hat{S}_i^y,\ \hat{S}_i^z)$ 是自旋-1/2 算符，$\langle i,\ j\rangle$ 表示最近邻，$\langle\langle i,\ j\rangle\rangle$ 表示次近邻。所有数值实验均在 10×10 晶格上进行，采用 $J_2/J_1 = 0.5$ 和周期性边界条件。我们采用标准 TQS 编码器—解码器架构 [4, 5, 8]，并在变分蒙特卡洛框架内进行所有计算；除非另有说明，我们报告五个随机种子 [19]中能量最低的结果。

表 I 比较了使用 PITQS 拟设 [式 (11)] 和标准 TQS 拟设[式 (9)]得到的能量。两种情况下，我们都对每一层采用一阶 Lie–Trotter 方案。我们固定 $\Delta\tau = 0.5$，并在总虚时间 $\beta \in \{1.0, 2.0, 3.0, 4.0\}$匹配的条件下比较 PITQS 和 TQS，即 $L \in \{2, 4, 6, 8\}$ 层。对于固定数量的变分参数N_p，PITQS 拟设在每个 β下都能达到与 TQS 相当的能量。对于更大的 β，PITQS 还能进一步达到更低的能量。相比之下，标准 TQS 的 N_p 随 β近似线性增长；然而，通过增大 β 来增加 N_p 并不能系统性地提高精度。这些结果表明，采用依赖虚时间的有效哈密顿量的传统 TQS 存在严重的过参数化：其核心表征能力源自有效哈密顿量本身的结构。

此外，在固定总虚时间 $\beta = 2.0$ 的情况下，我们对多种更新策略进行了基准测试，以评估分解格式对精度的影响。表 II 显示，提高分解阶数通常会持续改善能量表现，相较于一阶 Lie–Trotter 方案更是如此，这表明序列分解族中的 Trotter 误差得到了系统性降低。当我们将参数预算匹配到 TQS 基线的 $\beta = 0.5,\ N_p = 44,890$ 时，我们的 PITQS ans¨atze 展现出显著的性能优势。此外，与标准 TQS 基线相比，四阶铃木方案和 Blanes–Moan 方案所达到的能量优于 TQS，尽管它们使用的变分参数远少于更深层的 TQS 模型（其参数量为 $\beta = 2.0,\ N_p = 155,620$）。不过，我们注意到，高阶分解在计算成本和数值稳定性方面有所取舍，具体情形不同，其实际优势也可能因此受到限制 [20]。

最后，我们以实现高精度为目标，展示了该框架的可扩展性。我们训练了规模更大的 PITQS 模型，即一个包含 d =的二阶 Strang 试探式

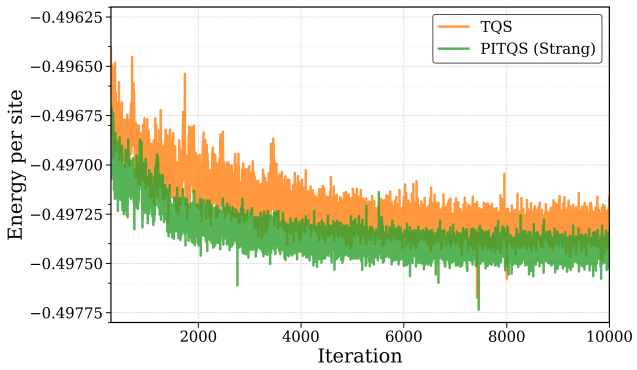

图 2. 每位点能量的优化过程（为便于阅读，省略了前 300 次迭代）。我们比较了参数量为 $N_p = 143,010$ 的 PITQS（Strang）拟设与参数量为 $N_p = 155,620$ 的标准 TQS。

108，$n_h = 18$，$L = 8$，$\Delta\tau = 0.5$，$N_p = 143,010$，并用 8,192 个蒙特卡洛样本进行每轮迭代，共优化 10,000 轮。为确保比较公平，我们将其与参数量相当的标准 TQS 进行基准比较（$d = 60$，$L = 4$，$\Delta\tau = 1.0$，$N_p = 155,620$）。我们采用这一基线，是因为在当前配置下复现大规模 TQS [8]($N_p \approx 268$k–995k) 的计算开销过高。

如图 2 所示，在整个优化过程中，我们的 PITQS 拟设显著优于可比的标准 TQS；尽管使用的变分参数更少（$N_p = 143,010$ 对比 155，620），仍达到了更高的精度。尤其值得注意的是，Strang 拟设达到了 $E \approx -0.49741$ (3) 的能量，这一结果可与参考文献 [8] of $E \approx -0.49725$ ($N_p = 267,720$) 和 $E \approx -0.49730$($N_p = 994,700$) 报告的数值相媲美，而仅使用了 143,010 个参数。这表明，拟设中的物理启发的归纳偏置，而非单纯增加参数规模，对精度起着关键作用。

机器学习视角。——从机器学习视角来看，采用单一有效哈密顿量的 PITQS 拟设与 Universal Transformers [21] 自然相关：单个 Transformer 层沿深度以权重共享进行迭代，类似于循环神经网络 [22]在广义上所用的循环。尽管 Transformer 中的权重共享通常是为提高参数效率而引入的（例如 ALBERT [23]），并可能引发对表达能力的担忧，但在我们的设定中，它对应于与虚时间无关的演化，因而构成一种建设性的归纳偏置，在不牺牲精度的情况下增强物理透明性，并在某些情形下甚至进一步提高精度。

总结.—在本快报中，我们提出了受物理启发的框架，将深层TQS重新解释为由单一有效哈密顿量驱动的LITE。通过在各层强制权重共享并采用高阶特罗特–铃木分解，我们表明PITQS 架构仅用远少于标准张量量子态架构的变分参数，即可在 J_1-J_2 海森堡模型上媲美最先进的TQS模型。本工作弥合了机器学习表达能力与物理透明性之间的差距，将神经网络量子态架构设计从启发式试错转变为系统且具有物理基础的构造。

本工作得到 JST CREST 资助编号 JPMJCR1913 以及 JSPS KAKENHI 资助编号 JP22H00516、JP22H05106（Y. K.）、JP25K15230（T.K.）和 JP24K20864（I.S.）的支持。

补充材料

ˆ ˆ $\mathcal{K}_\theta$ 与 $\mathcal{V}_\theta$ 的实现

我们介绍用于 PITQS 数值实验的非局部算子 ˆ ˆ ator $\mathcal{K}_\theta$ 和局部算子 $\mathcal{V}_\theta$ 的实现。作为一个说明性示例，我们以 Lie–Trotter 方案的形式给出层更新。离散构型 n 首先映射为潜在标记状态 $z(0) \in \mathbb{R}^{N_{\text{tok}} \times d}$，其中包含 N_{tok} 个标记。第 i 行 $z_i(0) \in \mathbb{R}^d$ 是索引为 $i \in \{1, \ldots, N_{\text{tok}}\}$ 的标记所对应的 d 维向量。随后，我们对 $z(0)$ 应用由 L 层组成的变换，得到 $z(\beta)$。按照正文中的记号，我们用 $z^{(\ell)} \in \mathbb{R}^{N_{\text{tok}} \times d}$ 表示第 ℓ 层的状态。我们将从 $z^{(\ell-1)}$ 到 $z^{(\ell)}$ 的更新写作

$$\tilde{z}^{(\ell)} = z^{(\ell-1)} + \Delta\tau\, \hat{\mathcal{K}}_\theta\big[z^{(\ell-1)}\big], \tag{15}$$

$$z^{(\ell)} = \tilde{z}^{(\ell)} + \Delta\tau\, \hat{\mathcal{V}}_\theta\big[\tilde{z}^{(\ell)}\big], \tag{16}$$

其中，两个算子 ˆ $\mathcal{K}_\theta[\cdot]$ 和 ˆ $\mathcal{V}_\theta[\cdot]$ 都会在内部对其输入进行层前归一化（pre-LN）。

ˆ 非局部算子 $\mathcal{K}_\theta$ 由因式分解多头注意力（FMHA）[8, 24]实现。FMHA首先使用层前归一化对输入$z^{(\ell-1)}$ 进行归一化：$\bar{z}^{(\ell-1)}$ =LN($z^{(\ell-1)}$)。然后，对于每个头 $h = 1, \ldots\ldots, n_h$，值特征计算为

$$V^{(h)} = \bar{z}^{(\ell-1)}\, W_v^{(h)} \in \mathbb{R}^{N_{\text{tok}} \times d_h}, \tag{17}$$

其中 $d_h = d/n_h$，且 $W_v^{(h)} \in \mathbb{R}^{d \times d_h}$。接下来，FMHA对头 h 的词元进行混合，如下

$$Y^{(h)} = A^{(h)} V^{(h)} \in \mathbb{R}^{N_{\text{tok}} \times d_h}. \tag{18}$$

此处，$A^{(h)}$ 是一个可训练的、与输入无关的词元间核矩阵，由下式给出

$$A^{(h)} = [\alpha_{ij}^{(h)}] \in \mathbb{R}^{N_{\text{tok}} \times N_{\text{tok}}}, \tag{19}$$

其中，(i, j)-项 $\alpha_{i,j}^{(h)}$ 是注意力权重，表示词元 i与 j之间的有效相互作用强度。将注意力头输出 $Y^{(h)}$ 沿

特征维度拼接并应用输出投影$W_o \in \mathbb{R}^{d\times d}$后，该映射可写为

$$\hat{\mathcal{K}}_\theta[z^{(\ell-1)}] = \mathrm{Concat}\big(Y^{(1)},\dots,Y^{(n_h)}\big)\, W_o \in \mathbb{R}^{N_{\mathrm{tok}}\times d}. \tag{20}$$

这种表示使非局部性变得显式：核 $A^{(h)}$ 耦合所有标记对（i，j）。对于平移不变的情形，可以进一步约束 $\alpha_{ij}^{(h)}$ 仅依赖于相对位移。

局部算子逐位置前馈网络实现。与 $\mathcal{K}_\theta$中一样，我们将层前归一化吸收到算子的定义中。具体而言，给定输入 $\tilde{z}^{(\ell)}$，我们首先得到归一化词元$\check{z}^{(\ell)} = \mathrm{LN}(\tilde{z}^{(\ell)})$，然后对每个标记索引 $i = 1$，.....，N_{tok}独立地应用同一个两层多层感知器：

$$\big(\hat{\mathcal{V}}_\theta[\tilde{z}^{(\ell)}]\big)_i = \mathrm{GELU}\Big(\check{z}_i^{(\ell)} W_1 + b_1^\top\Big)\, W_2 + b_2^\top. \tag{21}$$

这里，$\check{z}_i^{(\ell)} \in \mathbb{R}^d$ 表示$\check{z}^{(\ell)}$的第 i行。$W_1 \in\mathbb{R}^{d\times d_{\mathrm{FFN}}}$ 和 $W_2 \in \mathbb{R}^{d_{\mathrm{FFN}}\times d}$ 是权重矩阵，$b_1 \in\mathbb{R}^{d_{\mathrm{FFN}}}$ 和 $b_2 \in \mathbb{R}^d$ 是偏置向量，隐藏维度选取为 $d_{\mathrm{FFN}} = 4d$。该映射是显式局部的（格点上）：索引 i 处的输出仅依赖于特定的输入标记 $\check{z}_i^{(\ell)}$，而不与其他词元混合。

费米子系统的PITQS

我们介绍了专为费米子多体系统设计的 PITQS。与自旋系统PITQS相比，它有三个本质区别：(i) 将占据构型映射为词元的编码器；(ii) LITE模块的实现；以及 (iii) 在解码器中纳入反对称性。按照例如文献 [6, 7]，中费米子 TQS 的构造方式，我们构建费米子 PITQS 如下。

我们用$n \in \{0,1\}^{2N}$表示一个费米子占据构型。对于每个空间格点 $i = 1,\ \dots,\ N$，我们考虑一个局部类别标签 c_i，该标签取四个值之一，用于编码格点占据数：

$$c_i \equiv n_{i,\uparrow} + 2\, n_{i,\downarrow} \in \{0,1,2,3\}, \tag{22}$$

其中，$n_{i\sigma}$ 表示格点 i上自旋 σ 的占据数。编码器使用可学习的连续嵌入矩阵 $E \in \mathbb{R}^{4\times d}$对这些离散标签进行嵌入，生成词元特征

$$z_i(0) = E_{c_i} + p_i \in \mathbb{R}^d, \qquad i = 1,\dots,N, \tag{23}$$

其中 $\{p_i\}_{i=1}^N$ 是学习得到的位置嵌入。收集词元后得到 $z(0) \in \mathbb{R}^{N\times d}$，因此在费米子系统中我们设定$N_{\mathrm{tok}} = N$。

接下来，非局部算子 $\hat{\mathcal{K}}_\theta$ 通过标准多头注意力 [25]实现，使用查询和键，并遵循标准的费米子 TQS 架构 [6, 7]。与 FMHA 一样，我们首先将输入 $z^{(\ell-1)} \in \mathbb{R}^{N_{\mathrm{tok}}\times d}$ 归一化为 $\bar{z}^{(\ell-1)} = \mathrm{LN}(z^{(\ell-1)})$。然后，对于每个头 $h = 1$，.....，n_h且 $d_h = d/n_h$，我们计算查询、键和值矩阵，

$$Q^{(h)} = \bar{z}^{(\ell-1)} W_Q^{(h)}, \tag{24}$$

$$K^{(h)} = \bar{z}^{(\ell-1)} W_K^{(h)}, \tag{25}$$

$$V^{(h)} = \bar{z}^{(\ell-1)} W_V^{(h)}, \tag{26}$$

其中 $W_{(h)Q}$，$W_K^{(h)}$，$W_{(h)V} \in \mathbb{R}^{d\times d_h}$。输入相关的词元间核矩阵由逐行 softmax 函数定义

$$A^{(h)}(\bar{z}^{(\ell-1)}) = \mathrm{Softmax}\left(\frac{Q^{(h)} K^{(h)\top}}{\sqrt{d_h}}\right). \tag{27}$$

随后，注意力头输出可写成与 FMHA 相同的形式，

$$Y^{(h)} = A^{(h)}(\bar{z}^{(\ell-1)})\, V^{(h)} \in \mathbb{R}^{N_{\mathrm{tok}}\times d_h}, \tag{28}$$

拼接并进行输出投影 $W_o \in\mathbb{R}^{d\times d}$ 后，我们得到

$$\hat{\mathcal{K}}_\theta[z^{(\ell-1)}] = \mathrm{Concat}\big(Y^{(1)},\dots,Y^{(n_h)}\big)\, W_o \in \mathbb{R}^{N_{\mathrm{tok}}\times d}. \tag{29}$$

这种表示明确体现了非局部性：$A^{(h)}(z^{(\ell-1)})$ 将所有词元对（i，j）耦合起来，并通过 $Q^{(h)}$ 和 $K^{(h)}$依赖于状态。局部算子 $\hat{\mathcal{V}}_\theta$ 遵循与式（21）相同的结构，并吸收层前归一化步骤。这里，我们采用 SiLU 而非 GELU 作为非线性函数，并将隐藏维度设为$d_{\mathrm{FFN}} = 2d$:

$$\big(\hat{\mathcal{V}}_\theta[\tilde{z}^{(\ell)}]\big)_i = \mathrm{SiLU}\Big(\check{z}_i^{(\ell)} W_1 + b_1^\top\Big)\, W_2 + b_2^\top, \tag{30}$$

其中，$\check{z}_i^{(\ell)}$ 表示归一化词元 $\mathrm{LN}(\tilde{z}^{(\ell)})$ 的第 i 行。除隐藏维度外，参数形状与式（21）中的相同。最后，我们在各层之间采用权重共享，即 PITQS 具有一个静态的有效哈密顿量

$$\hat{\mathcal{H}}_\theta = -(\hat{\mathcal{V}}_\theta + \hat{\mathcal{K}}_\theta), \tag{31}$$

而 LITE 算符为 $\hat{\mathcal{U}}_\theta(\beta) = e^{-\beta\,\hat{\mathcal{H}}_\theta}$。在实践中，$\hat{\mathcal{U}}_\theta(\beta)$ 通过以 $\mathcal{V}_\theta$ 和 $\mathcal{K}_\theta$ L 次应用共享的 Transformer 层 $\hat{\ }$ $\hat{\ }$ 来实现。该层采用指数算符的特罗特–铃木分解实现（步长为 $\Delta\tau = \beta/L$）。

对于解码器，我们采用多斯莱特回流 [6, 7]来确保反对称性。对于每个解码后的标记 $z_i(\beta)$，我们通过共享的逐位置仿射映射生成依赖于构型的回流轨道，

$$x_i(\beta) = \mathrm{LN}(z_i(\beta))\, W_{\mathrm{out}} + \mathbf{1} b_{\mathrm{out}}^\top, \tag{32}$$

其中 $W_{\text{out}} \in \mathbb{R}^{d\times(2KN_e)}$ 以及 $b_{\text{out}} \in \mathbb{R}^{2KN_e}$。向量 x_i 被解释为 K 组自旋分辨轨道：对于每个行列式 $k \in \{1,...,K\}$，我们提取两个 N_e维向量 $x_i^{(k)}{}_{,\downarrow}$ 和 $x_i^{(k)}{}_{,\uparrow}$，并将它们视为回流轨道矩阵 $M^{(k)}(n) \in \mathbb{R}^{2N\times N_e}$的行。具体而言，按照选定的自旋轨道顺序，与自旋轨道 (i,σ) 对应的 $M^{(k)}(n)$中的行为 $x_{i\,\sigma}^{(k)}$。给定一个占据构型 $n \in \{0,1\}^{2N}$，令 $P(n) \in \{0,1\}^{N_e\times 2N}$为一个行选择矩阵，用于选取被占据的自旋轨道。于是，第 k个斯莱特矩阵 $\Phi_\theta^{(k)}(n)$可表示为以下矩阵乘积:

$$\Phi_\theta^{(k)}(n) = P(n)\, M_\theta^{(k)}(n) \in \mathbb{R}^{N_e\times N_e}. \tag{33}$$

最后，解码器输出与实数多行列式回流形式相对应的复数对数振幅,

$$\hat{\mathcal{D}}_\theta[z(\beta)] = \log\left|\sum_{k=1}^{K}\det\Phi_\theta^{(k)}(n)\right| \tag{34}$$

$$+\, i\,\pi\,\mathbb{I}\left[\sum_{k=1}^{K}\det\Phi_\theta^{(k)}(n) < 0\right], \tag{35}$$

其中 $\mathbb{I}[\cdot]$ 是示性函数。

实验细节

所有数值实验均在变分蒙特卡洛框架下进行，其中变分能量根据蒙特卡洛样本 $n \sim p_\theta\ n \qquad p_\theta\ n\ \propto |\Psi_\theta\ n\ |^2 \quad \mathbb{E}_{n\sim p_\theta}[\ \mathcal{H}\Psi_\theta\ \ n\ /\Psi_\theta\ n\ ]$ ( )

利用（ ）（ ）估计为（ ˆ ）（ ）。我们的 PITQS 架构在 NetKet 库 [26, 27],中实现，该库基于 JAX [28], 、Flax [29], 和 mpi4jax [30]构建。除非另有说明，我们报告五次采用随机初始化的独立运行中，经过 800 次迭代后获得的最佳最终能量。有关随机种子间稳定性的详细分析，请参见本补充材料中的“ PITQS 的计算稳定性”一节。对于可装入内存的模型，计算在两块各配备 40 GB 内存的 NVIDIA A100 GPU 上进行；较大的模型则在两块各配备 80 GB 内存的 NVIDIA H100 GPU 上运行。

对于 J_1-J_2 海森堡模型，我们使用最小步长随机重构（MinSR） [8, 18] 优化变分参数，学习率为 0.0075，对角平移为 10^{-4}。除非另有说明，我们进行了 800 次优化迭代，每次迭代使用 4,096 个蒙特卡洛样本，并将超参数设为模型维度 d =60、补丁大小 2×2、头数 $n_h = 10$和前馈网络隐藏维度 $4d$。优化预算及这些超参数遵循 NetKet教程中视觉变换器波函数的基准设置，参见 https://netket.readthedocs.io/en/latest/tutorials/ViT-wave-function.html。

表 III. PITQS 和 TQS ans¨atz 在 $\beta = 1.0$，2。0，3。0，4。0 时，五个随机种子对应的每格点能量平均值、随机种子间的均值标准误以及变分参数数量 N_p 。

	$\beta = 1.0$		
方案	Mean	均值标准误（$\times 10^{-4}$）	N_p
TQS	−0.49576	0.82	81，800
PITQS（Lie–Trotter）	−0.49579	0.56	44,890
PITQS（Strang）	−0.49563	2.3	44,890
PITQS（铃木）	−0.49642	0.88	44，890
PITQS（Blanes–Moan）	−0.49578	3.6	44, 890
	$\beta = 2.0$		
方案	Mean	均值标准误（$\times 10^{-4}$）	N_p
TQS	−0.49583	4.8	155，620
PITQS（Lie–Trotter）	−0.49626	1.2	44,890
PITQS（Strang）	−0.49633	1.3	44,890
PITQS（铃木）	−0.49653	2.7	44，890
PITQS（Blanes–Moan）	−0.49672	0.32	44,890
	$\beta = 3.0$		
方案	Mean	均值标准误（$\times 10^{-4}$）	N_p
TQS	−0.49621	2.0	229，440
PITQS（Lie–Trotter）	−0.49669	0.34	44,890
PITQS（Strang）	−0.49665	0.35	44,890
PITQS（铃木）	−0.49608	2.9	44,890
PITQS（Blanes–Moan）	−0.49658	1.1	44, 890
	$\beta = 4.0$		
方案	Mean	均值标准误（$\times 10^{-4}$）	N_p
TQS	−0.49645	0.27	303，260
PITQS（Lie–Trotter）	−0.49654	2.5	44，890
PITQS（Strang）	−0.49659	0.64	44，890
PITQS（铃木）	−0.49630	2.8	44，890
PITQS（Blanes–Moan）	−0.49613	3.6	44, 890

对于哈伯德模型，我们使用 MinSR 优化变分参数，学习率为 0.01，对角平移为 10^{-4}。我们进行了 10,000 次优化迭代，每次迭代使用 4,096 个蒙特卡洛样本。超参数设为模型维度$d = 16$、头数 $n_h = 4$、深度 $L = 4$ 和虚时间步长 $\Delta\tau = 0.5$。

PITQS 的计算稳定性

我们考察 PITQS 的计算稳定性如何取决于总虚时间 $\beta = L\Delta\tau$。虽然正文报告的是多次独立运行中获得的最佳能量，但此处我们关注运行间变异性，并评估汇总统计量，即均值和标准误。我们将步长设为 $\Delta\tau = 0.5$，并在 $\{1.0, 2.0, 3.0, 4.0\}$ 范围内扫描 β，同时保持所有其他设置不变。这些 β 值分别对应深度 $L \in \{2, 4, 6, 8\}$。为评估稳定性，我们使用五个独立随机种子重复每项实验，并汇总所得的运行间变异性。实验在两张 NVIDIA A100 GPU 上进行，但在 $\beta = L\Delta\tau$处的标准 TQS 基线实验除外，

由于内存限制，该实验使用两张 NVIDIA H100 GPU 运行。

表 III 报告了五个随机种子对应的每位点能量均值，以及跨随机种子的均值标准误（SEM）。在小β 区间（$\beta = 1.0, 2。0$）内，高阶分解（铃木和 Blanes–Moan）表现出更高的精度，得到更低的平均能量。然而，在大β 区间（$\beta = 3.0, 4。0$）内，这些高阶方法的性能显著下降；它们的平均能量更高，SEM 也更大，表明存在优化不稳定性。相反，低阶 Lie–Trotter 和 Strang 方案即使在更大的 β下仍保持很强的鲁棒性，其稳定性优于高阶方案，同时达到与 TQS 相当或更好的能量。这些观察结果凸显了高阶分解在实践中的权衡：虽然它们能够在适中的 β下减小 Trotter误差，但在较大的 β 下优化敏感性增加，因此在实际选择（L， $\Delta\tau$）时应谨慎。

PITQS的计算成本

我们根据两个实际指标评估PITQS和标准 TQS 的计算成本：GPU 内存的总峰值使用量，以及每次优化迭代的计算时间。所有基准测试均在固定步长 $\Delta\tau = 0.5$和总虚时间 $\beta \in \{1.0, 2.0, 3.0, 4.0\}$下进行，所有计算成本均通过使用相同超参数、在两块 NVIDIA H100 GPU 上分别运行的性能分析测试进行测量。

图 3(a) 比较了PITQS和标准 TQS 的GPU峰值内存使用量。对于PITQS，低阶 Lie–Trotter 和 Strang 方案表现出基本不依赖于 β 的内存使用量，这与 LITE模块中的权重共享相一致；权重共享使变分参数的数量与展开深度无关。即使是四阶铃木和 Blanes–Moan 分解，在较大的 β 下也只表现出适度增加，这是由于高阶阶段所需的中间缓冲区所致。相比之下，TQS 的内存使用量随着 β 显著增加，反映出增大 β 需要增加深度 $L = \beta/\Delta\tau$，从而增加不同层参数的数量。总体而言，在较大的 β 下，PITQS 的内存效率仍显著高于 TQS。

图 3(b) 展示了优化 TQS 和 PITQS 时每次迭代的计算时间。我们发现，PITQS 的 Lie–Trotter 和 Strang 方案的计算时间与 TQS 的相当，尽管 TQS 在较大的 β 下呈现出轻微的上升趋势，这与其相较于 PITQS 大得多的参数数量一致。相比之下，四阶铃木和 Blanes–Moan 分解会产生显著更高的成本，通常是低阶方案的 $\sim$ 5–7 倍。这一增长反映了四阶分解中更大的阶段数量

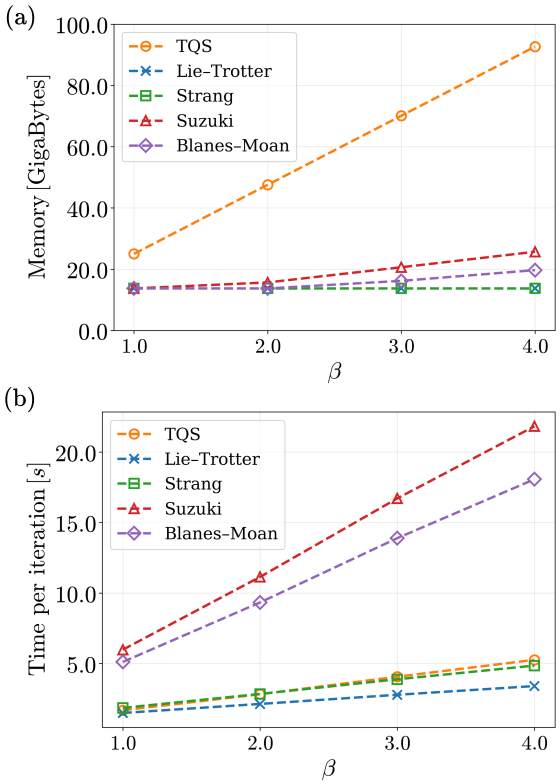

图 3. TQS 和 PITQS 的计算成本随总虚时间 $\beta \in \{1.0, 2.0, 3.0, 4.0\}$ 的变化。(a) GPU峰值内存使用量。(b) 每次优化迭代的计算时间（秒）。

k，并凸显了高阶方案在精度与计算成本之间的实际权衡。综上，这些结果表明，在许多实际应用场景中，Lie–Trotter 或 Strang 方案能够在计算成本、内存效率和精度之间实现良好平衡。

潜在空间中的冷却曲线

为直接检验 LITE 是否起到冷却过程的作用，我们评估预训练 PITQS 在不同虚时间长度下的能量。具体而言，我们采用正文图 2 中使用 Strang 方案的 PITQS 拟设，并通过将展开长度截断至 $\beta = 0.5, \ldots\ldots, 4。0$ 来评估能量，同时保持在 $\beta_{\text{train}} = 4.0$ 处预训练的变分参数 θ^* 不变。按照正文中的定义，虚时间长度为 $\beta = L\Delta\tau$。在当前基准测试中，我们固定 $\Delta\tau = 0.5$，由此得到八个评估点 $\beta \in \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0\}$。这一过程定义了一组波函数 $\Psi_{\theta^*}^{(\beta)}$，它们由施加 $L = \beta/\Delta\tau$ 个共享演化模块 $\hat{}$ 后，再通过同一个解码器 $\mathcal{D}_{\theta^*}$ 得到。

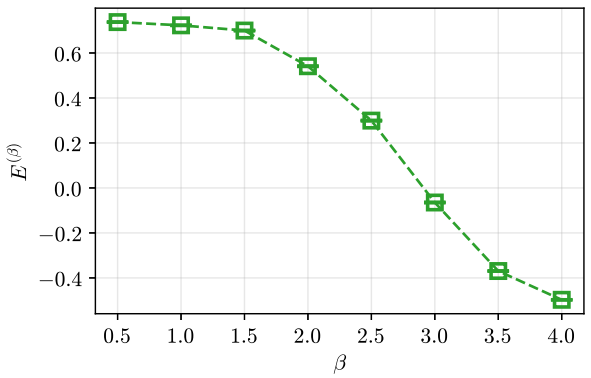

图 4. 使用在 $\beta = 4.0$ 处训练的 Strang 方案截断 PITQS 拟设，并通过 8，192 个蒙特卡洛样本评估得到的每格点能量 $E^{(\beta)}$ 。误差棒表示蒙特卡洛误差。

对于每个 β，我们使用固定的训练后变分参数 θ^* 估算每位点能量 $E^{(\beta)}$ 。具体而言，我们在每个β处的评估中使用 8，192 个蒙特卡洛样本。

图 4 展示了八个评估点的每位点能量 $E^{(\beta)}$ 估计结果。在训练范围内，增大 β 会系统性地降低变分能量。这一现象表明，在潜在空间中迭代 LITE步骤可发挥有效的冷却过程作用，这与 LITE解释一致。

评估仅限于 $\beta \le \beta_{\text{train}} = 8\Delta\tau$，也就是训练范围内，并不声称能够在其范围之外进行可靠的外推。实际上，在 $\beta > \beta_{\text{train}}$ 处评估相当于将潜在演化外推至训练范围之外，此时解码器可能会遭遇分布偏移。因此，要在更大的 β 处获得可靠预测，需要针对扩展后的展开长度进行微调。

对虚时间步长的依赖 $\Delta\tau$

我们考察每位点能量对虚时间步长 $\Delta\tau$ 的依赖关系，该步长用于对 LITE 算符进行离散化。我们将 $\Delta\tau$ 视为超参数，并在深度固定为 $L = 4$的条件下，评估三个取值 $\Delta\tau \in \{0.1, 0.5, 1.0\}$，即 $\beta \in \{0.4, 2.0, 4.0\}$。

表 IV 总结了所得的每格点能量，并列出 N_p 供参考。 $\Delta\tau = 0.5$列对应正文中使用的设置。在深度固定为 $L = 4$ 时，各方案对 $\Delta\tau$ 的依赖有所不同，反映了它们各自不同的理论特性。对于 PITQS，观察到的趋势与 Trotter–Suzuki 分解的理论特性一致。随着 $\Delta\tau$ 增加，Lie–Trotter 方案给出的能量略高，这符合一阶近似的预期。斯特朗分解在统计不确定性范围内没有表现出明显的精度下降，这与其二阶特性一致。值得注意的是，即使在 $\Delta\tau = 1.0$ 下，四阶铃木分解和 Blanes–Moan 分解仍保持高精度，表明高阶分解在较大 $\Delta\tau$时具有优势。

采用 Lie–Trotter 方案时，TQS 呈现出与 PITQS 相反的趋势。尽管 TQS 实际上实现了一阶李–特罗特分解，但我们确认，其精度会随着 $\Delta\tau$增大而提高。由于深度 L 固定，增大 $\Delta\tau$ 意味着总虚时间 β更长，这通常有助于基态冷却。TQS 的这种表现可能归因于其层依赖有效哈密顿量；这些哈密顿量带来了更大的变分自由度，使 TQS 能够吸收有限-$\Delta\tau$ 下的 Trotter误差。

连续极限与龙格–库塔积分

我们的以算子为中心的 LITE表述存在连续的潜在虚时间极限，这与神经常微分方程的观点相呼应 [31]：有效哈密顿量定义了潜在空间动力学，其解可通过深度离散化更新来近似。这种离散化会引入两种不同的近似误差来源。

首先，当有效哈密顿量被分解为$\hat{}\,\hat{}$非对易分量（即 $\mathcal{K}_\theta$ 和 $\mathcal{V}_\theta$）时，Trotter–Suzuki分解会产生Trotter误差，我们可以通过采用更高阶的分解来减小这种误差。其次，当我们用离散更新近似各分量的作用时，每个分量都会导致数值积分误差。为减轻此误差，我们可以采用更高阶的数值方法，而非“潜在虚时间演化”中介绍并在其他实验中使用的一阶欧拉方法。具体而言，我们实现了二阶龙格–库塔方法（RK2），即海恩方法 [32]。对于每个分量$\hat{}\,\hat{}\,\hat{}\mathcal{A}_\theta \in \{\mathcal{K}_\theta, \mathcal{V}_\theta\}$，该方法通过以下预测-校正步骤将潜在态从$z$ 更新为 z'：

$$
\begin{aligned}
k_1 &= \hat{\mathcal{A}}_\theta[z], \\
\tilde{z} &= z + \Delta\tau k_1, \\
z' &= z + \frac{\Delta\tau}{2}\left(k_1 + \hat{\mathcal{A}}_\theta[\tilde{z}]\right).
\end{aligned}
\tag{36}
$$

这种方法降低了局部积分误差，从而更准确地近似指数$\hat{}\,\hat{}$ 算符 $e^{\Delta\tau\mathcal{K}_\theta}$ 和 $e^{\Delta\tau\mathcal{V}_\theta}$。

表 V 报告了采用 Lie–Trotter 方案的欧拉方法和二阶龙格－库塔法的每格点能量。欧拉方法的结果与正文表 I 中 PITQS 的结果相对应。在 $\beta = 1.0$ 和 $\beta = 2.0$时，

表 IV. $\Delta\tau \in \{0.1, 0.5, 1.0\}$ 在固定深度 $L = 4$下的每位点能量。括号内为蒙特卡罗误差， N_p 表示变分参数的数量。

方案	$\Delta\tau = 0.1$	$\Delta\tau = 0.5$	$\Delta\tau = 1.0$	N_p
TQS	−0.49620(11)	−0.49652(9)	−0.49666(9)	155，620
PITQS（Lie–Trotter）	−0.49681(10)	−0.49669(9)	−0.49659(10)	44，890
PITQS（Strang）	−0.49667(9)	−0.49671(9)	−0.49667(9)	44，890
PITQS（铃木）	−0.49678(8)	−0.49697(9)	−0.49696(9)	44，890
PITQS（Blanes–Moan）	−0.49679(10)	−0.49683(7)	−0.49701(11)	44，890

表 V. 在相同总虚时间下，使用欧拉和二阶龙格–库塔方法子步得到的每格点能量 E 。

β	E (欧拉)	E (二阶龙格–库塔方法)
1.0	−0.49596(14)	−0.49628(11)
2.0	−0.49669(9)	−0.49682(10)
3.0	−0.49676(10)	−0.49679(10)
4.0	−0.49696(9)	−0.49695(8)

表 V。采用并行公式和串行公式、在总虚时间相同的条件下得到的每格点能量 E 。

β	E （序列化）	N_p（序列化）	E （并行）	N_p （并行）
1.0	−0.49596（14）	44，890	−0.49285（16）	44，770
2.0	−0.49669(9)	44,890	−0.49589(11)	44,770
3.0	−0.49676(10)	44，890	−0.49647(11)	44,770
4.0	−0.49696(9)	44,890	−0.49665(10)	44，770

二阶龙格－库塔法的结果在统计不确定性范围内略优于欧拉方法；而在$\beta = 3.0$和 $\beta = 4.0$时，两种方法在统计上已无法区分。这些结果表明，在较大的 β下，二阶龙格－库塔法可能带来的精度提升会被优化不稳定性掩盖，这与高阶铃木分解和 Blanes–Moan 分解中观察到的情况相似。

并行公式

在正文以及本补充材料前面的章节中，我们主要采用一种串行公式，其中非局域分量和局域分量在每一层内依次施加。对于一阶 Lie–Trotter 格式，$\Delta\tau = \beta/L$ 对应的一步可写为

$$\tilde{z}^{(\ell)} = z^{(\ell-1)} + \Delta\tau\, \hat{\mathcal{K}}_\theta\big[z^{(\ell-1)}\big], \tag{37}$$

$$z^{(\ell)} = \tilde{z}^{(\ell)} + \Delta\tau\, \hat{\mathcal{V}}_\theta\big[\tilde{z}^{(\ell)}\big]. \tag{38}$$

等价地，使用 $\hat{\mathcal{H}}_\theta = -(\hat{\mathcal{V}}_\theta + \hat{\mathcal{K}}_\theta)$，该串行公式对应于用各分量指数算符的乘积来近似完整演化算符 $\hat{\mathcal{U}}_\theta(\Delta\tau) = e^{-\Delta\tau\hat{\mathcal{H}}_\theta} = e^{\Delta\tau(\hat{\mathcal{K}}_\theta+\hat{\mathcal{V}}_\theta)}$。当 $\mathcal{K}_\theta$ 和 $\mathcal{V}_\theta$ 不对易时，这种分解会产生 Trotter误差，因此有必要采用本文其他部分讨论的高阶分解。

这里我们考虑另一种并行表述，其中有效哈密顿量保持为精确求和形式 $\hat{\mathcal{H}}_\theta = -(\hat{\mathcal{V}}_\theta + \hat{\mathcal{K}}_\theta)$，并且更新过程不进行算符分解。在这种表述中，使用欧拉方法进行更新的形式为

$$z^{(\ell)} = e^{-\Delta\tau\hat{\mathcal{H}}_\theta}[z^{(\ell-1)}] \approx z^{(\ell-1)} - \Delta\tau\hat{\mathcal{H}}_\theta\,[z^{(\ell-1)}]. \tag{39}$$

由于该表述直接使用完整的有效哈密顿量 $\mathcal{H}_\theta$ ，因此不会引入因拆分 $\mathcal{K}_\theta$ 和 $\mathcal{V}_\theta$ 而产生的 Trotter误差。近年来，机器学习领域的大规模模型采用了相关的并行表述，以减少每一层中的串行计算 [33, 34]。然而在实践中，并行结构在优化过程中可能更为敏感，因而可能需要进行稳定化处理。

表 VI 比较了在固定步长 $\Delta\tau = 0.5$ 和 $\beta \in \{1.0, 2.0, 3.0, 4.0\}$下，采用相同超参数时并行公式与串行公式的结果。串行结果对应于正文表 I 中 PITQS 的结果。在测试范围内，串行公式得到的变分能量略低于并行方案，而参数数量几乎相同。性能差距在 $\beta = 1.0$ 时最为明显，并在更大的 β时有所缩小。这一趋势表明，对于当前的模型规模和优化设置，串行更新的实际优势并非仅受 Trotter误差限制；相反，在每个层中依次应用 $\mathcal{K}_\theta$ 和 $\mathcal{V}_\theta$，相比式（39）中的并行耦合，能够带来更有利的归纳偏置和优化表现。

哈伯德模型的基准测试

我们在二维哈伯德模型上对PITQS进行基准测试，并与标准费米子 TQS进行比较。哈伯德哈密顿量由下式给出

$$\hat{\mathcal{H}} = -t \sum_{\langle i,j\rangle,\sigma} \left(\hat{c}^\dagger_{i\sigma}\hat{c}_{j\sigma} + \text{H.c.}\right) + U\sum_i \hat{n}_{i\uparrow}\hat{n}_{i\downarrow}, \tag{40}$$

表 VII. 半填充条件下，$U/t = 4$ 和 8 格点周期性边界条件的 4×4 哈伯德模型每位点能量。N_p 表示变分参数的数量。还列出了参考文献 [35] 中报告的精确对角化（ED）参考值。

方法	$E(U/t = 4)$	$E(U/t = 8)$	N_p
ED	-0.85137	-0.52931	N/A
TQS	$-0.85114(13)$	$-0.52909(25)$	11，424
PITQS	$-0.85113(22)$	$-0.52919(35)$	4,752

其中，$\hat{c}^\dagger_{i\sigma}$（$\hat{c}_{i\sigma}$）在格点 i处产生（湮灭）一个自旋为 $\sigma \in \{\uparrow, \downarrow\}$ 的电子，而$\hat{n}_{i\sigma} = \hat{c}^\dagger_{i\sigma}\hat{c}_{i\sigma}$ 是数目算符。第一项描述最近邻格点 $\langle i, j\rangle$ 之间振幅为 t的跃迁，第二项表示强度为 U的在位库仑排斥。实现与实验细节见本补充材料中“费米子系统的PITQS”一节和“实验细节”一节。

表 VII 总结了周期性边界条件下半填充时 4×4哈伯德模型的能量，并在相同超参数下比较了 PITQS 和 TQS 在相互作用强度为 $U/t = 4$ 和 $U/t = 8$ 时的表现。与正文表 I 中的 J_1-J_2 海森堡基准测试一样，PITQS 在使用显著更少变分参数的同时，达到了与 TQS 相当的精度。这些结果表明，通过权重共享强制采用静态有效哈密顿量，能够提供一种参数高效的归纳偏置，而且其效果并不限于自旋系统，也适用于相互作用费米子。

* 联系方式：k-yamazaki@ist.osaka-u.ac.jp
[1] M. Imada、A. Fujimori 和 Y. Tokura，《Rev. Mod. Phys.》**70**，1039（1998年）。[2] G. Carleo 和 M. Troyer，《Science》**355**，602（2017年）。[3] W. L. McMillan，《Phys. Rev.》**138**，A442（1965年）。[4] L. L. Viteritti、R. Rende 和 F. Becca，《Phys. Rev. Lett.》**130**，236401（2023年）。[5] L. L. Viteritti、R. Rende、A. Parola、S. Goldt 和 F. Becca，《Phys. Rev. B》**111**，134411（2025年）。[6] M. Geier、K. Nazaryan、T. Zaklama 和 L. Fu，《Phys. Rev. B》**112**，045119（2025年）。[7] Y. Gu、W. Li、H. Lin、B. Zhan、R. Li、Y. Huang、D. He、Y. Wu、T. Xiang、M. Qin、L. Wang 和 D. Lv，（2025年），arXiv:2507.02644。[8] R. Rende、L. L. Viteritti、L. Bardone、F. Becca 和 S. Goldt，《Co mmunications Physics》**7**，260（2024年）。[9] M. Gell-Mann 和 F. Low，《Phys. Rev.》**84**，350（1951年）。[10] R. Blankenbecler、D. J. Scalapino 和 R. L. Sugar，《Phys. Rev. D》**24**，2278（1981年）。[11] G. Sugiyama 和 S. Koonin，《Annals of Physics》**168**，1（1986年）。[12] G. Carleo、Y. Nomura 和 M. Imada，《Nature Communication》9, 5322（2018）。[13] S. Lie，*Theorie der Transformationsgruppen*，第1卷（BG Teubner，1888）。[14] H. F. Trotter，《美国数学会会刊》10, 545（1959）。[15] G. Strang，SIAM 数值分析杂志 5, 506（1968）。[16] M. Suzuki，《物理快报 A》146, 319（1990）。[17] S. Blanes 和 P. Moan，《计算与应用数学杂志》142, 313（2002）。[18] A. Chen 和 M. Heyl，《自然·物理》20, 1476（2024）。[19] 有关所有实现细节，包括确切的网络规格、优化设置和计算环境，以及种子平均结果和跨随机种子的相应均值标准误，请参见补充材料。[20] 有关计算成本和数值稳定性的详情，另请参见补充材料。[21] M. Dehghani、S. Gouws、O. Vinyals、J. Uszkoreit 和 L. Kaiser，载于*International Conference on Learning Representations* 2019（）。[22] J. L. Elman，《认知科学》14, 179（1990）。[23] Z. Lan、M. Chen、S. Goodman、K. Gimpel、P. Sharma 和 R. Soricut，载于*International Conference on Learning Representations* 2020（）。[24] R. Rende 和 L. L. Viteritti，《机器学习：科学与技术》6, 010501（2025）。[25] A. Vaswani、N. Shazeer、N. Parmar、J. Uszkoreit、L. Jones、A. N. Gomez、L. Kaiser 和 I. Polosukhin，载于*Proceedings of the 31st International Conference on Neural Information Processing Systems* 2017 6000 6010（）第–页。[26] G. Carleo、K. Choo、D. Hofmann、J. E. Smith、T. Westerhout、F. Alet、E. J. Davis、S. Efthymiou、I. Glasser、S.-H. Lin、M. Mauri、G. Mazzola、C. B. Mendl、E. van Nieuwenburg、O. O'Reilly、H. Th´eveniaut、G. Torlai、F. Vicentini 和 A. Wietek，《SoftwareX》10, 100311（2019）。[27] F. Vicentini、D. Hofmann、A. Szab´o、D. Wu、C. Roth、C. Giuliani、G. Pescia、J. Nys、V. Vargas-Calder´on、N. Astrakhantsev 和 G. Carleo，《SciPost Phys. Codebases》，7（2022）。[28] J. Bradbury、R. Frostig、P. Hawkins、M. J. Johnson、C. Leary、D. Maclaurin、G. Necula、A. Paszke、J. VanderPlas、S. Wanderman-Milne 和 Q. Zhang（2018）。[29] J. Heek、A. Levskaya、A. Oliver、M. Ritter、B. Rondepierre、A. Steiner 和 M. van Zee，《Flax：面向 JAX 的神经网络库与生态系统》（2024）。[30] D. H¨afner 和 F. Vicentini，《Journal of Open Source Software》6, 3419（2021）。[31] R. T. Q. Chen、Y. Rubanova、J. Bettencourt 和 D. Duvenaud，载于*Proceedings of the 32nd International Conference on Neural Information Processing Systems*（2018），第6572–6583页。[32] K. Heun，《数学物理杂志》45, 23（1900）。[33] A. Chowdhery、S. Narang、J. Devlin、M. Bosma、G. Mishra、A. Roberts、P. Barham、H. W. Chung、C. Sutton、S. Gehrmann、P. Schuh、K. Shi、S. Tsvyashchenko、J. Maynez、A. Rao、P. Barnes、Y. Tay、N. Shazeer、V. Prabhakaran、E. Reif、N. Du、B. Hutchinson、R. Pope、J. Bradbury、J. Austin、M. Isard、G. Gur-Ari、P. Yin、T. Duke、A. Levskaya、S. Ghemawat、S. Dev、H. Michalewski、X. Garcia、V. Misra、K. Robinson、L. Fe-dus、D. Zhou、D. Ippolito、D. Luan、H. Lim、B. Zoph、A. Spiridonov、R. Sepassi、D. Dohan、S. Agrawal,

M. Omernick、A. M. Dai、T. S. Pillai、M. Pellat、A. Lewkowycz、E. Moreira、R. Child、O. Polozov、K. Lee、Z. Zhou、X. Wang、B. Saeta、M. Diaz、O. Firat、M. Catasta、J. Wei、K. Meier-Hellstern、D. Eck、J. Dean、S. Petrov 和 N. Fiedel，《机器学习研究杂志》**24**, 1（2023）。[34] M. Dehghani、J. Djolonga、B. Mustafa、P. Padlewski、J. Heek、J. Gilmer、A. P. Steiner、M. Caron、R. Geirhos、I. Alabdulmohsin、R. Jenatton、L. Beyer、M. Tschan-nen、A. Arnab、X. Wang、C. Riquelme Ruiz、M. Min-derer、J. Puigcerver、U. Evci、M. Kumar、S. V. Steenkiste，G. F. Elsayed、A. Mahendran、F. Yu、A. Oliver、F. Huot、J. Bastings、M. Collier、A. A. Gritsenko、V. Birodkar、C. N. Vasconcelos、Y. Tay、T. Mensink、A. Kolesnikov、F. Pavetic、D. Tran、T. Kipf、M. Lucic、X. Zhai、D. Key- sers、J. J. Harmsen 和 N. Houlsby，载于 *Proceedings of the 40th International Conference on Machine Learning*，第 202 卷（2023年），第 7480–7512 页。[35] J. S. Anderson、M. Nakata、R. Igarashi、K. Fujisawa 和 M. Yamashita，《计算与理论化学》**1003**，22（2013年）。