arXiv:2305.02460v1 [cs.LG] 3 May 2023

张量化流：一种变分推断工具

Yuehaw Khoo*、Michael Lindsey†、和 Hongli Zhao‡

摘要。 得益于深度神经网络强大的表达能力，归一化流在生成建模领域取得了卓越成就，即基于有限的训练样本数据集，学习从分布中生成新样本。归一化流也成功应用于变分推断；在变分推断中，人们试图根据分布的对数似然或能量函数的表达式，而非数据，学习一个采样器。在变分推断中，归一化流所用的参考高斯分布具有单峰性，这可能会给学习多峰分布带来困难。我们提出一种归一化流的扩展方法，用通过张量网络构造的参考分布取代高斯参考分布，具体而言，该张量网络为矩阵乘积态或张量列车。我们表明，在具有挑战性的变分推断任务中，将流与张量网络结合使用，能够取得优于单独使用任一工具的结果。

关键词。 变分推断、高维近似、归一化流、张量训练、矩阵乘积态

美国数学学会主题分类。 68T07, 15A69,62Dxx

1. 引言。 归一化流 [63, 62] 定义了一类概率分布，可以直接从中抽取独立样本。神经网络参数化 [54] 的应用极大提升了归一化流在实践中的表达能力，使其在生成建模方面取得了显著成功，即在给定有限样本的情况下，学习从某个分布中抽取样本。归一化流的基础思想是通过一系列可逆映射变换高斯基分布，这一基本概念已经通过日益多样化的架构得以实现 [15, 16, 36, 7]。关于归一化流及其众多应用的最新综述，请参见 [53]。

在现代生成建模的其他关键范式中，尤其是生成对抗网络（GANs） [23] 和变分自编码器 [38], ，归一化流的独特之处在于，该模型生成的样本都带有精确的样本密度。这一特性使得归一化流可以直接应用于变分推断，例如在 [49, 1], 中，人们希望学习从某个概率分布中抽取样本，或计算关于该概率分布的期望；其密度函数 $p(\mathbf{x})$ 是已知的先验地 （除归一化常数外）。

变分推断（VI） [33, 4]近几十年来逐渐成为马尔可夫链蒙特卡罗（MCMC）方法的替代方案 [59], 。尽管马尔可夫链蒙特卡罗原则上是精确的，但较长的自相关时间可能会影响其效果。尽管两者有所不同，变分推断仍可与马尔可夫链蒙特卡罗有效结合，例如 [20, 2]所示。变分推断已广泛应用于图像处理与生成 [37, 68], 、强化学习[18],、凝聚态物理 [43, 64], 等领域。

变分推断从一个参数化的概率密度族 $\{p_{\boldsymbol{\theta}}\}$ 开始，并通过最小化.....来寻找该族内对目标密度 p 的最佳近似。

*芝加哥大学统计系，伊利诺伊州芝加哥，IL 60637，美国。邮箱：ykhoo@uchicago.edu

†加州大学伯克利分校数学系，美国加利福尼亚州伯克利，CA 94720。邮箱：lindsey@math.berkeley.edu

‡芝加哥大学计算与应用数学委员会，美国伊利诺伊州芝加哥，IL 60637。邮箱：honglizhaobob@uchicago.edu

Kullback–Leibler（KL）散度

$$D_{\mathrm{KL}}(p_{\boldsymbol{\theta}} \| p) = \int \log \left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x})}{p(\mathbf{x})} \right) p_{\boldsymbol{\theta}}(\mathbf{x}) \, \mathbf{dx}$$

通过低维变分参数向量 $\boldsymbol{\theta}$。变分推断的一项主要难题，是设计具有足够表达能力、同时仍支持精确采样和高效密度评估的概率分布参数化类别。除归一化流之外，传统的变分推断方法还包括通过混合模型 [22], 、推断网络 [47], 和隐式变分推断 [65]进行参数化。在基于流的变分推断方法中， $p_{\boldsymbol{\theta}}$ 被参数化为基准高斯分布 p_0 经映射 $T_{\boldsymbol{\theta}}$ 推前后的分布，该映射支持高效求逆和雅可比计算。

变分推断的另一项关键难题是优化模型参数本身，尤其是在目标分布 p 具有多峰特征时。在最极端的情况下，如果目标分布具有多个彼此相距甚远的众数，而当前的变分估计 $p_{\boldsymbol{\theta}}$ 仅准确捕捉到一个模态分量，那么对变分参数进行局部优化便无法发现缺失的众数，因此永远无法成功学到它们。在将归一化流应用于变分推断时，高斯参考分布 p_0 的单峰性会导致参数优化过程中出现“模式坍塌”的倾向，即只能准确学到一个模态，而其他模态仍未被发现。即使可以选择高斯混合分布作为参考分布，如果对模态结构没有任何先验知识，也无法保证所有模态都会被发现。关于这些难题的概述，请参阅 [19] 。

原则上，如果可以交换 $p_{\boldsymbol{\theta}}$ 和 p 的角色来优化 KL散度，即 $D_{\mathrm{KL}}(p \| p_{\boldsymbol{\theta}})$，这个问题就能得到解决；事实上，这正是生成建模中使用的目标函数 [54]。然而，要训练这个目标，必须能够从 p 中采样，而这恰恰是我们最初试图完成的任务。在 [19, 2], 中，提出了使用马尔可夫链蒙特卡罗从越来越准确的近似中自适应采样的方法，以逼近 p ，同时训练 $p_{\boldsymbol{\theta}}$。然而，尽管这类方法可以避免模式坍塌，却无法从根本上提升训练发现新众数的能力（除非这些众数可以通过足够长的马尔可夫链蒙特卡罗探索发现）。

更根本地说，归一化流的所有应用都高度依赖神经网络的表达能力，以修正基分布 p_0。如果 p 与高斯分布相差甚远，即使能够有效地学习到近似 $p_{\boldsymbol{\theta}} \approx p$ ，要获得良好的近似，可能也需要使用训练成本高昂的神经网络参数化。

因此，构造一个能够近似 p全局结构的参考分布 p_0 ，有充分的动机。如果能够做到这一点，或许就能避开多峰性带来的困难，并且可以采用更紧凑、更易训练的流映射神经网络参数化。已有多项研究探讨了对基分布的改进。为捕捉重尾分布，[31] 考虑了使用密度分位数函数，而 [44] 则提出使用 Copula 作为基分布。与此同时， [60]自适应地利用接受-拒绝重采样。然而，这些方法都无法纳入关于目标分布全局模态结构的先验知识。

张量网络最初源于对量子多体问题的数值研究。它们提供了一种低维参数化方法，能够克服维数灾难，而维数灾难使得多粒子波函数无法被精确存储。迄今为止，应用最广泛且最成功的张量网络仍然是矩阵乘积态（MPS）。矩阵乘积态与用于计算多体基态的密度矩阵重整化群（DMRG）算法同时出现 [67]。

这种结构在应用数学文献中也称为张量列车（TT） [52] ，通常将其置于将奇异值分解推广到高阶张量 [42, 14]这一更广泛的背景下进行研究。近年来，低秩张量分解受到越来越多的关注，例如应用于代理建模和深度学习 [50, 8, 61]。在众多格式中，张量列车格式的一大特点是能够轻松执行各种关键运算；只要核心张量秩保持有界，其复杂度就仅随问题维度线性增长。此外，各种张量列车运算也便于利用并行性 [56, 13]。许多高维经典问题，例如生成建模 [29], 、变分贝叶斯推断 [69, 25], 、监督学习 [12], 、数值偏微分方程 [34, 55], 和动力系统[40] ，在一些重要情形下都可以改写为张量列车格式并高效求解。

在 VI 设置中，给定密度 $p(\mathbf{x})$（忽略归一化因子），且该密度是一个黑盒函数。因此，可以应用交叉近似来获取 TT 的各个分量。在这项工作中，我们提出结合 NF 和 TT 的优势来进行 VI。我们通过对函数张量列车 [58] （由对目标密度 [48] 的平方根进行交叉近似 [51]得到）求平方，构造一个基分布为 TT 的分布。相较于 [17], , 这确保了基分布的非负性和精确采样能力。然后，我们通过由神经网络参数化的流映射，将基分布向前推送。我们将模型称为张量化流（TF），因为它结合了张量分解与基于流的模型的优势。与 NF 一样，TF 能够进行精确采样并提供样本密度。然而，在计算成本相当的情况下，以变分目标衡量，TF 的表现优于直接使用 NF。同时，流还能改善通过数值线性代数程序直接得到的基 TT 分布。我们通过一个 Ginzburg-Landau 模型 [26, 45] 以及其他值得关注的高维测试分布，展示了本方法的优势；我们发现，单独使用 NF 或 TT 都难以处理该模型。

我们的贡献：本文解决了归一化流中基分布的初始化问题。我们通过张量网络方法实现这一点，最终可归结为快速的线性代数程序。具体而言，我们利用张量列车正交化，改进了近期提出的平方逆罗森布拉特传输 [10] 方法在插值概率密度时的线性缩放特性，同时保证非负性。从另一个角度看，我们的方法也可以视为对基于张量列车的生成模型 [11], 的改进：通过为其添加神经网络流模型，弥补张量列车表示能力有限的问题。

1.1. 结构安排。第 2 节回顾变分推断和归一化流的背景知识。第 3 节详细介绍如何利用平方 TT 表示构造基分布，以及如何生成样本（并计算其密度）；当秩固定时，计算成本随维度线性增长。第 4 节将对最终的 TF 模型在多种高维概率分布上进行测试。

1.2. 符号约定。对于正整数 $n \in \mathbb{N}$，全文中我们记 $[n] := \{1, 2, \ldots, n\}$。全文中，$d$ 表示问题维度，即采样变量 $\mathbf{x}$ 的维度。给定 $n_1, \ldots, n_k \in \mathbb{N}$ 和指标 $i_j \in [n_j]$、$1 \le j \le d$，我们通常考虑如下形式的多重指标：$\mathcal{I} := (i_1, \cdots, i_d)$。我们使用方括号表示离散索引的张量值，例如 $\mathcal{A}[\mathcal{I}]$ ，其中 d 是一个适当的d阶张量 $\mathcal{A}$。同时，我们使用圆括号表示连续函数求值，例如 $p(\mathbf{x})$。为描述张量的子张量，我们采用广泛使用的“MATLAB

表示法”。例如，若 $\mathbf{A} \in \mathbb{R}^{m\times n}$ 表示一个矩阵或二阶张量，则 $\mathbf{A}[i,j]$ 表示单个元素，而 $\mathbf{A}[i,:]$ 和 $\mathbf{A}[:,j]$ 分别表示第 i 行和第 j 列。在对象命名时，我们用非粗体小写字母表示标量（例如 $a,u,v\in\mathbb{R}$），用粗体小写字母表示向量（例如 $\mathbf{v},\mathbf{w}\in\mathbb{R}^n$），用粗体大写字母表示矩阵（例如 $\mathbf{A},\mathbf{B}\in\mathbb{R}^{m\times n}$），用花体大写字母表示一般的高维张量（例如 $\mathcal{A},\mathcal{C}\in\mathbb{R}^{n_1\times\cdots\times n_d}$）。必要时可能会有例外，但届时会明确说明。

最后，我们展示了几个张量网络图 [5]，以更好地说明张量列车运算。在本文中，张量图由节点和边组成，其中节点表示“核心张量”，由此构建出与该图相关联的大型张量。一般而言，我们允许张量同时具有离散和连续指标，将其简单地视为某些离散变量与某些连续变量的联合函数。一个节点伸出的边数表示张量的阶数，即指标的数量。伸出边之间的连接表示相应的指标在整体张量中共享，并被求和（或积分）消去；参见第2.3节，我们将在需要时回顾其他特殊记法。

2. 背景与预备知识。 我们在第2.1节中介绍一般的变分推断问题，并在第2.2节中介绍使用归一化流实现该问题的方法。第2.3节回顾矩阵乘积态（MPS）/张量列车（TT）分解，总结该表示形式下所需的运算，并讨论其在高效压缩高维概率分布方面的应用。

2.1. 变分推断。给定函数 $U:\Omega\to\mathbb{R}$，我们希望学习如下定义的目标密度：

$$p(\mathbf{x})=\frac{1}{Z}e^{-U(\mathbf{x})}, \tag{2.1}$$

其中 $Z=\int_\Omega e^{-U(\mathbf{x})}\,d\mathbf{x}$ 是归一化常数或配分函数。

在变分推断（VI）中，人们将配分函数表示为一个优化问题的最优值。更准确地说，给定一个分布族 $\mathcal{P}$，我们最小化

$$\begin{aligned} L(\tilde{p}) &:= \mathbb{E}_{\mathbf{x}\sim\tilde{p}}\Big[\log\tilde{p}(\mathbf{x})+U(\mathbf{x})\Big] \\ &= D_{\mathrm{KL}}(\tilde{p}||p)-\log Z. \end{aligned} \tag{2.2}$$

关于 $\tilde{p}\in\mathcal{P}$。此处 D_{KL} 表示 Kullback-Leibler（KL）散度。

假设 $p\in\mathcal{P}$，该最小化问题的最优值为 $-\log Z$，当且仅当 $D_{\mathrm{KL}}(\tilde{p}||p)=0$ 时取得，即当且仅当 $\tilde{p}=p$。如果 $p\notin\mathcal{P}$，那么优化（2.2）将恢复 p 在分布族 $\mathcal{P}$ 中的最佳近似，其衡量标准为 KL 散度。

如果我们能够计算样本的密度 $\tilde{p}(\mathbf{x}^{(s)})$，则可以使用从 $\tilde{p}$ 中抽取的有限样本集 $\{\mathbf{x}^{(s)}\}_{s=1}^{S}$ 来估计（2.2）中的期望。

2.2. 使用归一化流进行变分推断。 在实际应用中，我们考虑由 $\boldsymbol{\theta}$ 参数化的分布族。现在将参数化分布记为 $p_{\boldsymbol{\theta}}$，我们希望求解：

$$\boldsymbol{\theta}^\star=\operatorname*{argmin}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}),$$

这里我们稍微滥用一下记号，将 $L(\boldsymbol{\theta})$ 记作 $= L(p_{\boldsymbol{\theta}})$。

变分推断的准确性最终受限于参数化族 $\{p_{\boldsymbol{\theta}}\}$的表达能力。在这项工作中，我们专注于基于归一化流的族。更准确地说，我们通过一系列可微且可逆的映射 $F_{\boldsymbol{\theta}}^{(k)}$，$k = 1, \ldots, K$的复合，将 $p_{\boldsymbol{\theta}}$ 构造为固定基分布 p_0 的 **推前**，如同在

$$T_{\boldsymbol{\theta}} := F_{\boldsymbol{\theta}}^{(K)} \circ \cdots \circ F_{\boldsymbol{\theta}}^{(1)},$$

它们本身由一组神经网络参数 $\boldsymbol{\theta}$参数化。映射的数量 K 就是流的长度。将其记为很有用。

$$T_{\boldsymbol{\theta}}^{(k)} := F_{\boldsymbol{\theta}}^{(k)} \circ \cdots \circ F_{\boldsymbol{\theta}}^{(1)},$$

其中 $T_{\boldsymbol{\theta}}^{(0)}$:=Id，因此 $T_{(k+1)\boldsymbol{\theta}} = F_{(k+1)\boldsymbol{\theta}} \circ T_{(k)\boldsymbol{\theta}}$对于 $k = 0, \ldots, K-1$，以及 $T_{(K)\boldsymbol{\theta}} = T_{\boldsymbol{\theta}}$。

具体而言，为了对 $\mathbf{x} \sim p_{\boldsymbol{\theta}}$进行采样，先对 $\mathbf{z} \sim p_0$ 进行采样，然后计算 $\mathbf{x} = T_{\boldsymbol{\theta}}(\mathbf{z})$。根据变量变换，样本 $\mathbf{x}$ 的模型密度可计算为

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = p_0(T_{\boldsymbol{\theta}}^{-1}(\mathbf{x}))\, |DT_{\boldsymbol{\theta}}^{-1}(\mathbf{x})|.$$

接着可以看到，损失可写成关于基分布的期望:

$$L(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z}\sim p_0}\Big[\log p_0(\mathbf{z}) - \log |DT_{\boldsymbol{\theta}}(\mathbf{z})| + U(T_{\boldsymbol{\theta}}(\mathbf{z}))\Big],$$

进一步展开可得

$$L(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z}\sim p_0}\left[\log p_0(\mathbf{z}) - \sum_{k=1}^{K} \log \left|DF_{\boldsymbol{\theta}}^{(k)}\left(T_{\boldsymbol{\theta}}^{(k-1)}(\mathbf{z})\right)\right| + U(T_{\boldsymbol{\theta}}(\mathbf{z}))\right]. \tag{2.3}$$

通过抽取 S 个相互独立的样本 $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(S)} \sim p_0$，损失 (2.3) 可以通过经验估计为

$$\widehat{L}(\boldsymbol{\theta}) = \frac{1}{S}\sum_{s=1}^{S}\left[\log p_0(\mathbf{z}^{(s)}) - \sum_{k=1}^{K} \log \left|DF_{\boldsymbol{\theta}}^{(k)}\left(T_{\boldsymbol{\theta}}^{(k-1)}(\mathbf{z}^{(s)})\right)\right| + U(T_{\boldsymbol{\theta}}(\mathbf{z}^{(s)}))\right], \tag{2.4}$$

梯度 $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$ 可以通过“重参数化技巧”进行估计，即固定样本 $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(S)}$，并对求值公式 (2.4) 自动求导。

映射序列 $F_{(1)\boldsymbol{\theta}}, \cdots, F_{\boldsymbol{\theta}}^{(K)}$ 旨在平衡表达能力与高效计算雅可比矩阵行列式。我们在[41] 中介绍了多种选择。常见架构包括 NICE [15], FFJORD[24], 和 RealNVP [16]。本文选择使用可逆残差网络（即 iResNets） [3, 6]。

2.3. 张量运算。本节总结了若干基本的张量运算。

我们通过示例来说明张量缩并及其相关记法。更为正式的定义可参见 [39]。张量缩并通常定义为对重复指标/参数进行求和/积分。例如，考虑一个 3-张量 $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ 和一个矩阵$\mathbf{B} \in \mathbb{R}^{n_2 \times m_2}$。通过缩并可得到一个新张量

$\mathcal{C}$，其大小为 $n_1 \times m_2 \times n_3$，具体方法是将 $\mathcal{A}$ 的中间指标与$\mathbf{B}$的第一个指标进行缩并：

$$\mathcal{C}[i_1, j_2, i_3] = \sum_{k_2} \mathcal{A}[i_1, k_2, i_3]\mathbf{B}[k_2, j_2] =: \langle \mathcal{A}[i_1, k_2, i_3], \mathbf{B}[k_2, j_2] \rangle_{k_2 \in [n_2]} \tag{2.5}$$

可以将一个函数的索引与其函数参数对应起来，从而将其视为一个张量，并以类似的方式定义适当的张量缩并。例如，设 $k\colon X_1 \times X_2 \to \mathbb{R}$ 和 $g\colon X_2 \to \mathbb{R}$，则可以定义一个新的函数 $f\colon \mathbb{R} \to \mathbb{R}$，如下所示：

$$f(x_1) = \int_{X_2} k(x_1, x_2) g(x_2) dx_2 = \langle k(x_1, x_2), g(x_2) \rangle_{x_2 \in X_2}. \tag{2.6}$$

还可以定义如下的混合类型缩并[1]：

$$\mathcal{C}_{i_1 i_3}(x_2) := \mathcal{C}[i_1, x_2, i_3] = \sum_{i_2} \mathcal{C}[i_1, i_2, i_3] f_{i_2}(x_2) =: \langle \mathcal{C}[i_1, i_2, i_3], f_{i_2}(x_2) \rangle_{i_2 \in [n_2]} \tag{2.7}$$

在上述方程中，我们引入了括号记法，作为一种简写形式，用于表示对某些指标或连续空间进行缩并。我们有时会再次使用这种记法，以使涉及多个指标的冗长求和表达式更加易读。

在这项工作中，经常会遇到一些特殊的缩并，例如张量-张量、张量-矩阵和张量-向量缩并。缩并可用于将低阶张量组合成高阶张量。例如，张量列车可由多个 3-张量构造出一个 d 维张量。

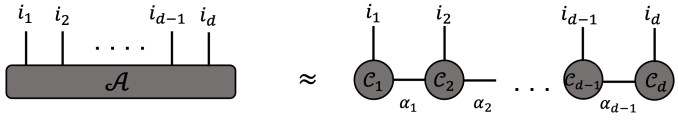


图 2.1： d 维张量的张量列车分解

DEFINITION 2.1. （张量列车分解）张量 $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ 是秩为 $(r_0, \cdots, r_d)$ 的张量列车（*TT*），其中 $r_0 = r_d = 1$，如果它可以写成

$$\begin{aligned}\mathcal{A}[\mathcal{I}] &= \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_{d-1}=1}^{r_{d-1}} \mathcal{C}_1[1, i_1, \alpha_1]\mathcal{C}_2[\alpha_1, i_2, \alpha_2] \cdots \mathcal{C}_d[\alpha_{d-1}, i_d, 1] \\ &= \mathcal{C}_1[1, i_1, :]\mathcal{C}_2[:, i_2, :] \cdots \mathcal{C}_d[:, i_d, 1],\end{aligned}$$

其中 $\mathcal{I} = (i_1, \ldots\ldots, i_d)$，最后一个等式只是矩阵乘法的序列，而 *3* 阶张量 $\mathcal{C}_i \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$ 称为核心张量。

图 2.1 展示了张量图示意图，采用第 1.2 节所述的约定。

这种分解的优点在于，我们可以存储并计算核列表 $\{\mathcal{C}_i\}_{i=1}^d$，而不必处理完整张量 $\mathcal{A}$；后者包含 $O(n^d)$ 个元素。如果秩较低，各种运算的计算成本便可显著降低

[1]当指标混合（即同时包含离散指标和连续指标）时，我们通常倾向于将离散指标写成下标。

。实际上，存储量按 $O(dnr^2)$ 增长，而计算张量单个元素的值所需的计算量按 $O(dr^2)$ 增长。加法等常见的张量列车运算，只需操作核即可在张量列车格式中实现，具体总结见 [52]。通过对展开矩阵 [52] 进行 SVD，可以最优地截断张量列车秩。

最后，我们介绍 TT 上的正交化操作，在 MPS 文献中也称为规范化。该过程依次对每个 TT 核进行正交化，直到除一个核之外的所有核都完成正交化。使用张量图表示法，我们在图2.2中展示了正交化后的 TT 分解；感兴趣的读者可参阅 [52, 21] 了解更多详情。按照 [27]中的介绍，我们将正交化后的核标记为半实心，以区别于原始核。节点的浅色一侧表示，对于正交化后的核 $\mathcal{Q}_k$ ，经过适当矩阵化后，其行是标准正交的，即

$$\langle \mathcal{Q}_k[:, i_k, \alpha_k], \mathcal{Q}_k[:, i_k, \alpha_k] \rangle_{i_k, \alpha_k} = I_{r_{k-1}}, \text{ for all } k = 2, \ldots, d \tag{2.8}$$

图2.2中的 TT 称为右-左正交。正交化过程的计算开销为 $O(dnr^3)$ 次运算。

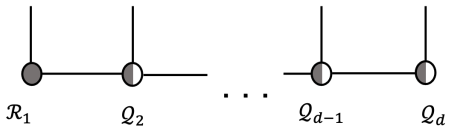


图2.2：TT 右-左正交形式示意图。

3. 所提方法。本节介绍张量化流的构造。我们考虑在区域 $\Omega = [a_1, b_1] \times \cdots \times [a_d, b_d] \subset \mathbb{R}^d$上近似形式为 (2.1) 的目标分布 p 。通过平移和缩放区域，我们总可以假设 $\Omega = [-1, 1]^d$。在 (2.3)-(2.4) 的流框架下，我们不再选择高斯基分布 p_0，而是使用 TT 构造一个基分布 p_0 ，使其已经能够较好地近似目标分布 p 。为了优化经验损失 (2.4)，我们需要能够使用 p_0高效地完成以下操作：

- 绘制样本 $\mathbf{z} \sim p_0$。
- 计算样本密度 p_0（$\mathbf{z}$）。

在第 3.1 节中，我们将采用张量列车构造 $p_0 \approx p$ ，使其满足这些条件。在第 3.2 节中，我们将介绍基于可逆残差网络的流映射 $F_{\boldsymbol{\theta}}^{(k)}$ ，作为我们的具体选择。

3.1. 基分布。我们将构造 $p_0(\mathbf{x})$，使其成为函数张量列车 [10] （记作 $q_0(\mathbf{x})$）的逐点平方。在第 3.1.1 节中，我们将说明如何从这样的 p_0 中抽取带有密度的样本；在第 3.1.4 节中，我们将说明如何仅利用计算定义目标分布 $p(\mathbf{x})$ 的能量函数 $U(\mathbf{x})$ $\propto \exp(-U(\mathbf{x}))$ 的能力来构建 q_0 。

3.1.1. 自回归采样框架。从多元分布中采样通常并非易事。与其他使用 MPS/张量列车进行生成建模的工作类似 [9, 29, 48], ，我们将以“自回归”的方式抽取样本 $\mathbf{x}^* \sim p_0$ ，

即写作:

$$p_0(x_1,\dots,x_d) = p_1(x_1)p_2(x_2|x_1)\cdots p_d(x_d|x_1,\dots,x_{d-1}), \tag{3.1}$$

然后从适当的条件分布中依次采样每个后续坐标 x_k^*，并以先前采样的坐标 x_j、 $j=1$、.....、$k-1$ 为条件。

换句话说，对于每个 $k=1$，2，$\cdots$，d，我们都希望进行采样:

$$x_k^* \sim p_k(x_k|\mathbf{x}_{<k}^*), \tag{3.2}$$

其中 $\mathbf{x}_{<k}^* = (x_1^*,\ x_2^*,\ \dots\dots,\ x_{k-1}^*)$。这种方法将获取多变量样本 $\mathbf{x}^* = (x_1^*,\ \dots\dots,\ x_d^*)$ 的问题转化为一系列单变量采样问题，而这些问题本质上可视为微不足道。难点在于构造这些单变量分布。根据条件概率的定义:

$$p_k(x_k|\mathbf{x}_{<k}^*) = \frac{p_0(\mathbf{x}_{<k}^*, x_k)}{p_{<k}(\mathbf{x}_{<k}^*)} \propto p_{<k+1}(\mathbf{x}_{<k}^*, x_k), \tag{3.3}$$

将先前采样的坐标视为固定值。这里我们定义了边缘分布:

$$p_{<k}(\mathbf{x}_{<k}) := \int p_0(x_1,\dots,x_k,x_{k+1},\dots,x_d)\,dx_{k+1}\dots dx_d \tag{3.4}$$

因此，为了构造所需的单变量分布，我们实际上只需对 p_0 求边缘分布，必要时可固定一些前置变量。

3.1.2. 表示 p_0。即使只有一个分量，要处理连续统，我们也会通过考虑截断的正交归一基 ϕ_1，.....，ϕ_n 来进行离散化，该基可由正交多项式等构造得到。实际上，我们可以为每个维度 $j=1$，.....，d选择不同的基大小 n_j，但为简化讨论，我们只考虑 $n_j = n$统一的情况。

暂时假设我们还给定了一个 d 维张量列车 $\mathcal{C} \in \mathbb{R}^{n\times\cdots\times n}$，它规定了某个函数的系数

$$q_0(\mathbf{x}) \approx q(\mathbf{x}) := \sqrt{p(\mathbf{x})}$$

在由 $\{\phi_i\}_{i=1}^n$诱导的 d 重张量积基中。然后，我们的近似密度 $p_0 \approx p$ 可以通过逐点平方 $p_0(\mathbf{x}) = q_0(\mathbf{x})^2$ 得到。

具体而言，假设

$$q_0(\mathbf{x}) = \sum_{\mathcal{I}} \mathcal{C}[\mathcal{I}]\,\phi_{\mathcal{I}}(\mathbf{x}) = \sum_{i_1=1}^{n}\cdots\sum_{i_d=1}^{n} \mathcal{C}[i_1,\cdots,i_d]\phi_{i_1}(x_1)\cdots\phi_{i_d}(x_d), \tag{3.5}$$

其中，纯张量积函数 $\phi_{\mathcal{I}}$ 和系数张量 $\mathcal{C}[\mathcal{I}]$ 满足

$$\phi_{\mathcal{I}}(\mathbf{x}) := \phi_{i_1}(x_1)\cdots\phi_{i_d}(x_d), \quad \mathcal{C}[\mathcal{I}] = \langle q_0(\mathbf{x}), \phi_{\mathcal{I}}(\mathbf{x})\rangle_{\mathbf{x}\in\Omega}.$$

将 $\mathcal{C}$ 的核记为 $\mathcal{C}_1,\dots,\mathcal{C}_d$。目前暂不介绍如何将 $\mathcal{C}$ 转换为 TT，相关细节将在第 3.1.4 节中说明。

我们在此指出，尽管我们打算从 $p_0(\mathbf{x}) = q_0(\mathbf{x})^2$,中获取样本，但我们从未显式地将其系数张量构造成张量列车，因为这样做需要处理大小为 $r^2\times n\times r^2$ [11]的核心张量。相反，我们只直接处理

$\mathcal{C}$ 的核心张量。具体而言，我们证明，将 $\mathcal{C}$ 的核置于右-左正交形式后，采样过程 (3.2) 中所需的关于变量 $x_{k+1}, \ldots, x_d$ 的边缘化可以免费完成，且整个抽取样本的成本仅随维度 d 线性增长。

因此，在进一步讨论细节之前，我们定义 $\mathcal{C}$的右-左正交化，

$$\mathcal{C}[i_1, \ldots, i_d] = \mathcal{R}_1[1, i_1, :]\mathcal{Q}_2[:, i_2, :] \cdots \mathcal{Q}_d[:, i_d, 1],$$

如第2.3节所述，该过程可在 $O(dnr^3)$时间内完成。示意图见图2.2。

3.1.3. 采样算法。 现在我们讨论如何在已知$\mathbf{x}^* \sim p_0$为右-左正交形式的张量列车 $\mathcal{C}$的情况下生成精确样本。在任一步骤 $k = 1, \ldots, d$，我们都需要有边缘分布的表示

$$p_{<k+1}(\mathbf{x}^*_{<k}, x_k) = \int q_0(x^*_1, \ldots, x^*_{k-1}, x_k, \ldots, x_d)^2 \, dx_{k+1} \cdots dx_d \tag{3.6}$$

以生成坐标样本 x^*_k。函数 $q_0(\mathbf{x})$可以通过将混合张量 $\phi_i(x)$连接到 $\mathcal{C}$的张量列车图的外腿上，用带有连续外腿的张量网络图表示，如图3.1所示。

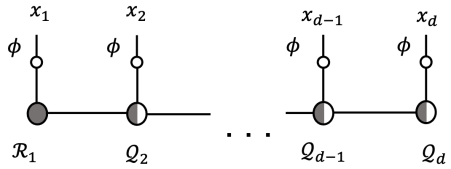


图3.1：将 $q_0(\mathbf{x})$表示为张量网络的示意图，该网络由张量列车系数$\mathcal{C}$ 组成，并且每个维度中都包含连续单变量函数 $\phi_i(x_i)$。它可以被解释为同时具有离散和连续指标的混合型张量。

计算（3.6）所需的张量缩并如图3.2所示。单变量基函数 $\phi_1, \ldots, \phi_n$ 的正交归一性保证了（3.6）中的积分会得到单位矩阵，它们构成了图3.2(a)中第 k 个核右侧的“梯子”横档。与此同时，固定前面的坐标 $x^*_1, \ldots, x^*_{k-1}$ 相当于计算核 $\mathcal{R}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_{k-1}$ 与基函数求值向量 $\boldsymbol{\phi}^*_1, \boldsymbol{\phi}^*_2, \ldots, \boldsymbol{\phi}^*_{k-1}$ 之间的张量 - 向量缩并，这些向量定义如下：

$$\boldsymbol{\phi}^*_j = \begin{bmatrix} \phi_1(x^*_j) \\ \vdots \\ \phi_n(x^*_j) \end{bmatrix} \in \mathbb{R}^n, \tag{3.7}$$

如图3.2(a)所示。这里，我们分别将这些张量-向量缩并得到的矩阵记为 $\mathbf{R}^*_1, \mathbf{Q}^*_2, \ldots, \mathbf{Q}^*_{k-1}$。

从图3.2的框架 (a) 到框架 (b)，右左正交性使我们能够在张量网络图中从右向左简单地“收缩梯子”，无需任何额外计算。从框架 (b) 到框架 (c)，火车的左侧尾部可以通过所示方式收缩为单个向量 $\mathbf{v}_{k-1}$

矩阵-向量乘法序列。为保持线性维度缩放，$\mathbf{v}_{k-2}$ 的计算可以直接从前一步复用，而 $\mathbf{v}_{k-1}$ 只需再进行一次矩阵乘法即可得出。

在框架 (c) 中，核心 $\mathcal{Q}_k \in \mathbb{R}^{r_{k-1}\times n\times r_k}$ 与 $\mathbf{v}_{k-1} \in \mathbb{R}^{r_{k-1}}$ 收缩，得到矩阵 $\mathbf{B}_k \in \mathbb{R}^{n\times r_k}$。从框架 (c) 过渡到框架 (d) 时，我们执行定义 $\mathbf{A}_k$ 的矩阵-矩阵乘法: $= \mathbf{B}_k\mathbf{B}_k^\top$，从框架 (d) 可以清楚地看出

$$f_k(x_k) := p_{<k+1}(\mathbf{x}^*_{<k}, x_k) = \sum_{i,j=1}^{n} \mathbf{A}_k[i,j]\phi_i(x_k)\phi_j(x_k). \tag{3.8}$$

注意，$\mathbf{A}_k$ 根据构造是半正定的，因此可以保证该表达式作为 x_k 的函数处处非负。

现在根据 (3.3)，我们知道单变量函数 f_k 与单变量条件概率密度 $p_k(x_k|\mathbf{x}^*_{<k})$ 成正比，而我们希望从该密度中抽取一个样本。如果 $\phi_1, \dots, \phi_n$ 是勒让德多项式或切比雪夫多项式等多项式，则可以精确计算累积分布函数，随后通过高效的求根，使用逆累积分布函数法抽取样本。在实践中，在间隔足够细的网格上计算 f_k，并通过梯形法则实现逆累积分布函数法，也十分高效。

给定由我们的采样程序生成的多元样本 $\mathbf{x}^* = (x_1^*, \dots, x_d^*)$，恢复其密度 $p_0(\mathbf{x}^*)$ 是轻而易举的。事实上，根据定义 (3.8) 可知，$f_d(x_d^*) = p_0(\mathbf{x}^*)$，因此我们只需在最终坐标样本 x_d^* 处计算所构造的最终单变量函数 f_d。

总而言之，一旦 $\mathcal{C}$ 已作为张量列车（TT）提供，从 p_0 生成 N个样本的全部成本将按.....的规模增长

$$O(dnr^3 + Ndn^2r^2),$$

这里的第一项与右左正交化预处理步骤相关，我们将 $r := \max_{i=1,\dots,d-1}\{r_i\}$ 定义为 $\mathcal{C}$ 的最大张量列车秩。此外，请记住，n 是单变量基 $\phi_1, \dots, \phi_n$ 的大小，而 d 是问题的维度。

3.1.4. 获取 $\mathcal{C}$ 作为张量列车。接下来讨论如何以张量列车格式生成 $q_0(\mathbf{x})$ 的系数张量 $\mathcal{C}$。本节的关键难点在于弄清楚如何利用 TT-cross 算法 [51]。

回顾（2.1），$p \propto \tilde{p} := e^{-U}$，其中我们将 $\tilde{p}$ 定义为一种未归一化密度，可逐点求值。还要记住，我们希望得到 $q = \sqrt{p}$的基于张量列车的近似 q_0。首先，我们将得到 $\tilde{q} := \sqrt{\tilde{p}}$的基于张量列车的近似 $\tilde{q}_0$，然后通过归一化步骤恢复适当的 q_0。

我们还将相对于 t，用系数张量表示 $\tilde{q}_0$，即 $\mathcal{B}$ o
我们的函数张量积基:

$$\tilde{q}_0(\mathbf{x}) = \sum_{\mathcal{I}} \mathcal{B}[\mathcal{I}]\, \phi_{\mathcal{I}}(\mathbf{x}). \tag{3.9}$$

系数张量 $\mathcal{B}$ 与我们所求的系数张量 $\mathcal{C}$ 仅相差一个标量常数因子，稍后我们将计算该因子。

TT-cross 最适合恢复定义在多变量网格上的函数，该函数可以视为带有离散指标的普通张量。因此，我们

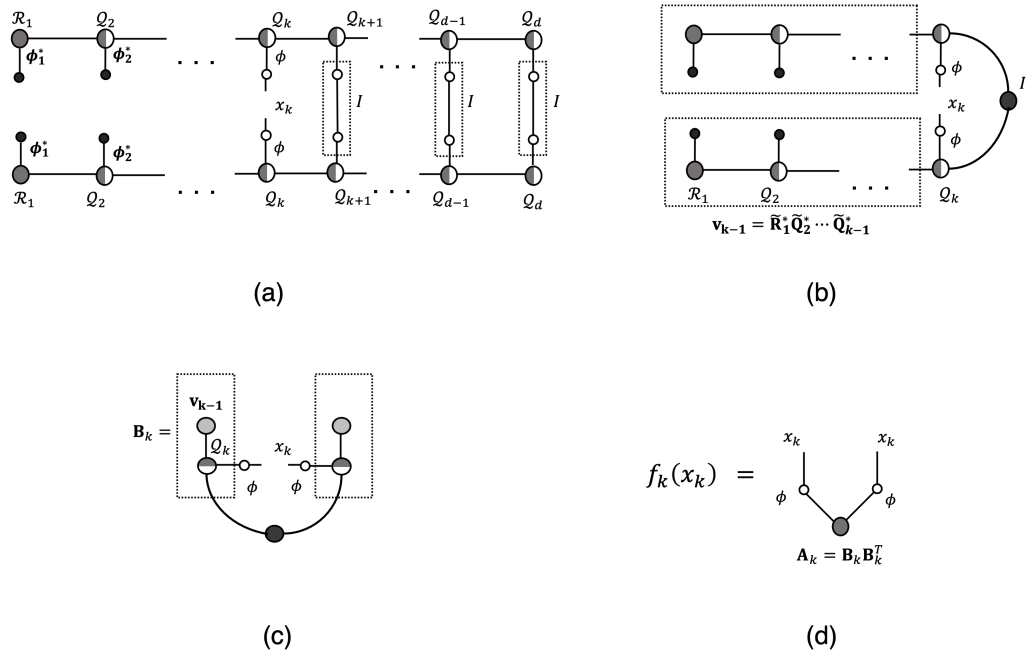


图 3.2： $f_k(x_k)$ 的构造：$= p_{<k+1}(\mathbf{x}^*_{<k}, x_k)$。有关该过程的详细分步说明，请参阅第 3.1.3 节的正文。

将使用 TT-cross 在这样的网格上恢复 $\tilde{q}$ 的基于张量列车的近似，然后通过数值求积推导出适当的系数张量 $\mathcal{B}$。

为此，令 $x^{(1)}, \dots, x^{(m)} \in [-1, 1]$ 表示一元求积点，其对应的求积权重为 $w^{(1)}, \dots, w^{(m)} \in \mathbb{R}$。正如我们在讨论一元基函数 $\phi_1, \dots, \phi_n \in L^2([-1, 1])$ 时所述，可以为每个维度 $x_k^{(1)}, \dots, x^{(m_k)_k}$ 选择不同的一组求积点（以及权重），但为简单起见，我们对每个维度采用相同的选择，特别是对所有 $k = 1, \dots, d$。

为避免与我们用于表示基系数的多重指标记法 $\mathcal{I}$ 混淆，我们将使用 $\mathcal{J} = (j_1, \dots, j_d)$ 对多元网格进行索引，并定义 $\tilde{q}$的网格值张量 $\mathcal{T} \in \mathbb{R}^{m \times \cdots \times m}$ 如下：

$$\mathcal{T}[\mathcal{J}] := \tilde{q}(x^{(j_1)}, \dots x^{(j_d)}). \tag{3.10}$$

现在可以应用 TT-cross，仅通过有限次查询张量 $\mathcal{T}$的元素，精确地确定一个 TT 近似 $\mathcal{S} \approx \mathcal{T}$。设 $\mathcal{S}$ 的核记为$\mathcal{A}_1, \dots, \mathcal{A}_d$。

为了保证 $\tilde{q}_0 \approx \tilde{q}$，系数张量只需满足

$$\mathcal{B}[\mathcal{I}] \approx \langle \tilde{q}(\mathbf{x}), \phi_{\mathcal{I}}(\mathbf{x}) \rangle_{L^2(\Omega)}. \tag{3.11}$$

因此，通过用数值求积替换 (3.11) 右侧的积分，并用 TT 近似 $\mathcal{S}$ 给出的值替换网格上的 $\tilde{q}$ 值，我们可以如图 3.3 所示构造 $\mathcal{B}$，其中权重矩阵

$\mathbf{W} \in \mathbb{R}^{n\times m}$ 定义为

$$\mathbf{W}[i,j] = \phi_i(x^{(j)})w^{(j)}.$$

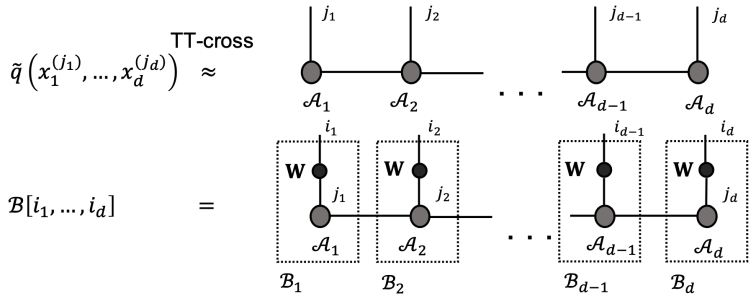


图3.3：基系数的张量列车表示 $\mathcal{B}$。(a) 展示了我们的张量列车 $\mathcal{S}$，其核为 $\mathcal{A}_1, \ldots, \mathcal{A}_d$，用于近似表示多元网格上 $\tilde{q}$ 的取值所构成的张量 $\mathcal{T}$。(b) 展示了如何将 $\mathcal{B}$ 构造为张量列车。

最后，我们讨论如何从 $\mathcal{B}$ 中恢复经过正确归一化的 $\mathcal{C}$。回顾一下，$p_0 = q_0^2$，其中

$$q_0(\mathbf{x}) = \sum_{\mathcal{I}} \mathcal{C}[\mathcal{I}],$$

利用张量积基的正交归一性可以很容易看出，

$$\int_\Omega p_0 \, d\mathbf{x} = \|q_0\|_{L^2(\Omega)} = \|\mathcal{C}\|_{\mathrm{F}} = \langle \mathcal{C}[\mathcal{I}], \mathcal{C}[\mathcal{I}] \rangle_{\mathcal{I}}.$$

为了确保我们的基分布 p_0 确实是归一化的概率分布，因此我们希望确保 $\|\mathcal{C}\|_{\mathrm{F}} = 1$。

事实上，$\|\mathcal{B}\|_{\mathrm{F}}$ 可以通过初等张量列车收缩计算。或者，如果将$\mathcal{B}$ 置于右-左正交形式，则可以将 $\|\mathcal{B}\|_{\mathrm{F}}$ 恢复为仅第一个张量核的 Frobenius范数。因此，我们有动机定义

$$\mathcal{C} = \frac{\mathcal{B}}{\|\mathcal{B}\|_F}, \tag{3.12}$$

确保 $\|\mathcal{C}\|_{\mathrm{F}} = 1$ 符合预期。$\mathcal{C}$ 的张量列车格式可以通过仅对$\mathcal{B}$ 的一个核心进行重新缩放来恢复，例如第一个核心（如果 $\mathcal{B}$ 已经转换为右-左正交形式，则此操作将保持右左正交性）。

3.2. 流模型。继第 2.2 节之后，我们已经构建了基分布 p_0，现在只需讨论流模型 $F_\theta^{(k)}$ 的选择，以及如何优化损失 (2.3)-(2.4)。

在本研究中，我们使用基于可逆残差网络（即 iResNets）的流 [3, 6]。我们选择残差流模型是因为它具有较强的表达能力，同时可以方便地将流映射初始化为接近单位矩阵，因为基分布可能已经能够对目标分布提供有用的近似。

在残差流模型中，流映射构造如下：

$$F_{\boldsymbol{\theta}}^{(k)}(\mathbf{x}) = \mathbf{x} + G_{\boldsymbol{\theta}}^{(k)}(\mathbf{x}), \tag{3.13}$$

其中， $G_{\boldsymbol{\theta}}^{(k)}$ 本身由前馈神经网络参数化。在变分推断中，我们无需通过计算求出 $F_{\boldsymbol{\theta}}^{(k)}$ 的逆，不过如果 $G_{\boldsymbol{\theta}}^{(k)}$ 是压缩映射，则可保证其可逆，而巴拿赫不动点定理则支持通过高效迭代计算求得其逆 [32]。

然而，如第 2.2 节所述，能够计算 $F_{\boldsymbol{\theta}}^{(k)}$ 的雅可比行列式十分重要。这可以通过泰勒级数展开实现：

$$\log|DF_{\boldsymbol{\theta}}^{(k)}| = \log|I_d + DG_{\boldsymbol{\theta}}^{(k)}| = \mathrm{Tr}(\log(I_d + DG_{\boldsymbol{\theta}}^{(k)})) = \sum_{m=1}^{\infty}(-1)^{m+1}\frac{\mathrm{Tr}[(DG_{\boldsymbol{\theta}}^{(k)})^m]}{m}. \tag{3.14}$$

我们可以截断级数 (3.14)，并使用 Hutchinson 估计量 [57]进行随机迹近似，以加速计算。

为加速训练，我们选择在每个残差连接后添加批量归一化 [30] 。此外，由于我们的基分布已经相当接近目标分布，因此我们希望将权重初始化为使映射接近单位矩阵。为此，残差连接 $g_{\boldsymbol{\theta}}$ 中的所有权重均从区间 $[-0.25, 0.25]$上的均匀分布中生成。

3.3. 总结。这里我们总结所提出的变分推断方法，该方法针对密度 $p(\mathbf{x}) \propto \exp(-U(\mathbf{x}))$，并假设能够计算函数$U(\mathbf{x})$。

1. 使用 TT-cross 构造一个张量列车 $\mathcal{S}$ ，以近似我们多元网格上函数 $\tilde{q}(\mathbf{x}) := \exp(-U(\mathbf{x})/2)$ 的取值张量 $\mathcal{T}$(3.10)。2. 通过图 3.3 和公式 (3.12) 的构造，建立一个基系数张量列车 $\mathcal{C}$ ，并根据公式 (3.5) 指定 $q_0(\mathbf{x})$。根据构造， $p_0(\mathbf{x}) := q_0(\mathbf{x})^2$ 是归一化的，并近似目标分布 $p(\mathbf{x})$。3. 按照第 3.1.3 节所述的算法，抽取一组样本 $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(S)} \sim p_0$，并为其配备密度 $p_0(\mathbf{z}^{(s)}), s = 1, \ldots, S$。4. 使用第 3.2 节指定的流模型，优化经验损失 (2.4)。

训练的更多细节将在下文第 4 节讨论，其中将介绍数值实验。

4. 数值实验。本节展示张量化流学习高维目标分布的能力。我们将在第 4.1 节介绍构建张量列车基分布的实现细节，并在第 4.3 节介绍流训练的实现细节。第 4.3 节将介绍关于高斯混合分布（第 4.3.1 节）和金兹堡-朗道模型（第 4.3.2 节和第 4.3.3 节）的实验。

4.1. 基分布构造的详细信息。 回顾一下，在构造基分布时，我们对式 (3.10) 中定义的张量 $\mathcal{T}$ 应用 TT-cross，以获得张量列车 $\mathcal{S}$。对于各种实验，我们在表 4.1 中记录了 $\mathcal{S}$ 的最大张量列车秩，该秩也是系数张量$\mathcal{C}$的最大张量列车秩。为了进行基准测试，我们还通过在 TT-cross 中指定局部相对精度 10^{-10} ，并允许秩增大到确保达到该精度所需的值，来获得真实系数张量 $\mathcal{C}_{\text{true}}$。这些秩通常远高于我们在实验中使用的低秩近似的秩，相关结果也一并报告。

见表 4.1，以及单变量基函数的数量 n ϕ_1，……， ϕ_n，我们选择勒让德多项式作为这些基函数。

实验	d	n	最大秩为 $\mathcal{C}_{\text{true}}$	最大秩为 $\mathcal{C}$
高斯混合	30	512	116	2
一维金兹堡-朗道	25	50	30	2
二维金兹堡-朗道	64	30	694	3

表 4.1：各实验的基分布构造细节。

在计算方面，我们将所有问题平移并缩放至立方体 $[-1,1]^d$，并将所有归一化流的基分布初始化为 $\mathcal{N}(\mathbf{0}, 0.2I)$，以确保所有样本均落在计算域内。在报告结果时，我们再将其平移并缩放回 $p(\mathbf{x})$ 支撑集对应的原始边界框。

4.2. 流模型的细节。 我们使用带有 ReLU 激活的多层感知机（MLP）神经网络架构来定义式（3.13）中的残差连接。实验中使用的隐藏层数量（网络深度）和层宽度见表 4.2。训练时，我们使用 Adam [35],进行优化，并采用衰减因子为 0.9999 的指数学习率调度器。训练超参数汇总于表 4.3。正如 [66], 所观察到的，较小的学习率和较大的批量大小通常能使模型更稳定地收敛。为进一步稳定训练，我们实施了梯度裁剪，使梯度各项始终处于范围 $[-10^4, 10^4]$内。

实验	层宽度	网络深度
高斯混合	32	5
一维金兹堡-朗道	32	5
二维金兹堡-朗道	64	5

表 4.2：神经网络宽度和深度。

实验	批量大小	学习率	训练轮数	流长度
高斯混合	128	5×10^{-4}	200	10
一维金兹堡-朗道	256	5×10^{-4}	200	12
二维金兹堡-朗道	64	3×10^{-5}	200	12

表 4.3：训练超参数。

为保持一致性，我们在完全相同的训练条件下比较 TF 和 NF 的性能。所有流模型均使用大小为 $S_{\text{train}} = 10^4$ 的训练集进行训练，并在大小为 $S_{\text{holdout}} = 10^4$ 的留出集上进行验证。对于 TF 和 NF，我们将绘制使用留出样本评估的损失（2.4）。验证损失将取 10 次训练运行的平均值，其中网络初始化采用随机方式。

4.3. 示例。 在这里，我们将张量化流应用于一系列高维密度，并将其性能与传统归一化流进行比较。

为了量化 TF 相对于 NF 的性能，我们将 $-\log Z_{\text{true}}$ 定义为根据第 4.1 节所述方法获得的高保真度张量列车所估算的损失 (2.4)。同样，我们将 $-\log Z_{\text{TF}}$ 和 $-\log Z_{\text{NF}}$ 分别定义为张量化流和归一化流训练后的最终损失。然后我们定义

误差比：

$$\text{Error Ratio} := \frac{\log Z_{\text{true}} - \log Z_{\text{TF}}}{\log Z_{\text{true}} - \log Z_{\text{NF}}}. \tag{4.1}$$

注意，分子和分母都必须为正。

4.3.1. 高斯混合模型。首先，我们考虑一个维度为 $d = 30$ 的合成示例，它定义为五个高斯分布的混合。均值为： $\boldsymbol{\mu}^{(1)} = [0, \cdots, 0, 2, 2]^\top$, $\boldsymbol{\mu}^{(2)} = [0, \cdots, 0, 2, -2]^\top$, $\boldsymbol{\mu}^{(3)} = [0, \cdots, 0, -2, 2]^\top$, $\boldsymbol{\mu}^{(4)} = [0, \cdots, 0, -2, -2]^\top$， 以及 $\boldsymbol{\mu}^{(5)} = [0, \cdots, 0, 0, 0]^\top$. 然后我们将矩阵 $\boldsymbol{\Sigma}^{(i)}$, $i = 1, \ldots, 5$ 定义为单位矩阵，但最后两个维度对应的 2×2 块除外，在该处我们取：

$$\boldsymbol{\Sigma}^{(1)}_{d-1:d,d-1:d} = \boldsymbol{\Sigma}^{(4)}_{d-1:d,d-1:d} = \begin{bmatrix} 1 & 19/20 \\ 19/20 & 1 \end{bmatrix}$$

and

$$\boldsymbol{\Sigma}^{(2)}_{d-1:d,d-1:d} = \boldsymbol{\Sigma}^{(3)}_{d-1:d,d-1:d} = \begin{bmatrix} 1 & -19/20 \\ -19/20 & 1 \end{bmatrix}.$$

然后，我们将 $p(\mathbf{x})$ 定义为高斯分布$\mathcal{N}(\boldsymbol{\mu}^{(i)}, 0.4 \times \boldsymbol{\Sigma}^{(i)})$、 $i = 1, \ldots, 5$ 的等权混合。

图 4.1 (a) 和 (d) 展示了 p 的各种边缘分布，清晰地体现了多峰性。图 4.1 (b)、(c)、(e) 和 (f) 展示了由训练后的 NF 和 TF 得到的边缘密度估计。

作为比较，图 4.2 展示了由张量列车基分布 p_0 生成的相应边缘密度估计。

我们可以看到，即使基分布非常粗糙，仅涉及秩为 2 的 TT（这是非平凡的最小秩），TF 也更能捕捉目标分布的多峰性。与此同时，使用单峰基分布的 NF 无法捕捉这些众数。我们通过图 4.3 中的验证损失图和表4.4，定量地确认了 TF 的有效性。

基分布	开始	End	−对数 Z_{true}	误差比
张量列车（秩 2）	21.5266	20.1952	20.1794	0.0271
高斯	33.4478	20.7634		

表4.4：应用于高斯混合模型的TF和归一化流训练概要

4.3.2. 一维金兹堡-朗道模型。金兹堡-朗道理论提供了各种非平凡的采样问题，这些问题与相变和稀有事件的研究有关 [28]。即使对于简单模型，传统采样方法通常也无能为力 [46]。本节中，我们考虑一维晶格上的反铁磁体的金兹堡-朗道型模型，其中相邻格点的状态倾向于朝向相反方向。具体而言，我们将 $(d + 2)$ 个格点上的状态记为u_0， u_1，……， $u_{d+1} \in \mathbb{R}$，并满足边界条件 $u_0 = u_{d+1} = 0$。能量函数定义如下：

$$E(u_1, \ldots, u_d) := -\sum_{i=1}^{d+1} \frac{\delta}{2} \left(\frac{u_i - u_{i-1}}{h} \right)^2 + \frac{1}{4\delta} \sum_{i=1}^{d} (1 - u_i^2)^2, \tag{4.2}$$

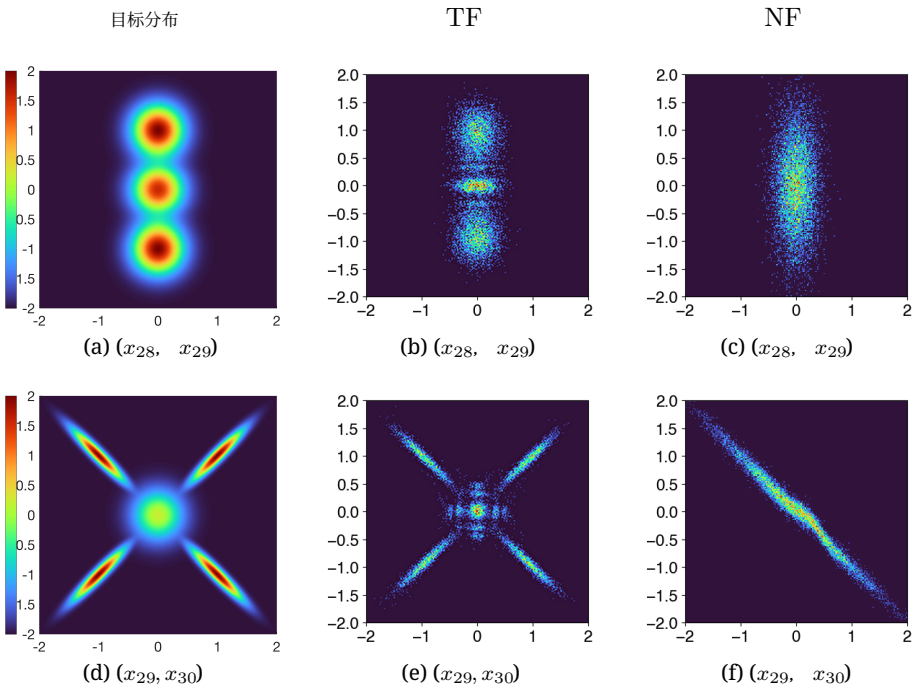


图4.1：对应变量（x_{28}, x_{29}）和（x_{29}, x_{30}）的高斯混合模型边缘分布。(a)(d)：真实边缘密度。(b)(e)：通过对训练后的TF进行采样获得的边缘密度估计。(c)(f)：通过对训练后的NF进行采样获得的边缘密度估计。

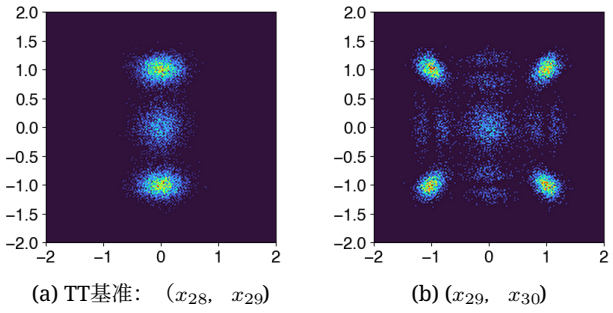


图4.2：直接从秩-2张量列车基分布 p_0获得的高斯混合模型估计边缘分布，未使用流。

其中 $\delta > 0$ 是一个标量参数， $h > 0$ 是晶格间距。(4.2) 中的最后一项是一个双阱势，它确保（当 δ 较小时） u_i 取值接近 ± 1。于是，目标概率密度定义为

$$p(\boldsymbol{u}) \propto \exp(-\beta E(\boldsymbol{u})), \tag{4.3}$$

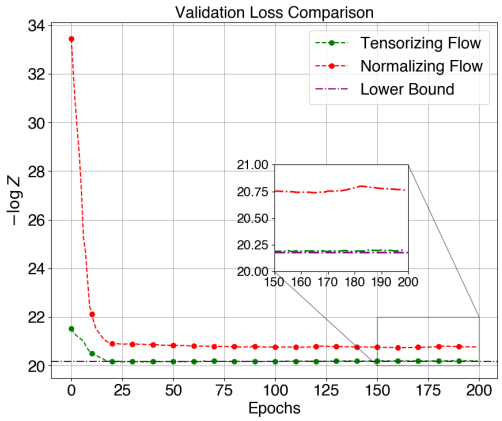


图 4.3：高斯混合分布在训练过程中 TF（张量化流）和 NF（归一化流）的验证损失图。

其中， β 是逆温度。能量函数有两个极小化点，从而产生双峰分布。

在我们的实验中，我们设置 $d = 35$, $h = 1/(d+1) = 1/36$, $\beta = 6.25 \times 10^{-2}$, 以及$\delta = 0.04$。我们在图 4.4 中展示了从 p_{true} 中抽取的每个模态的一个典型代表。

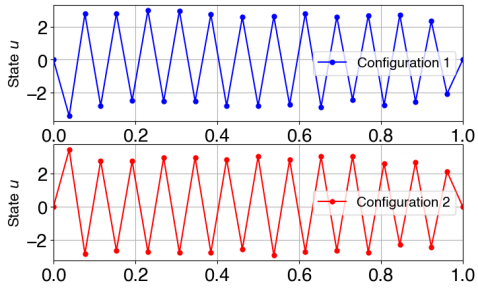
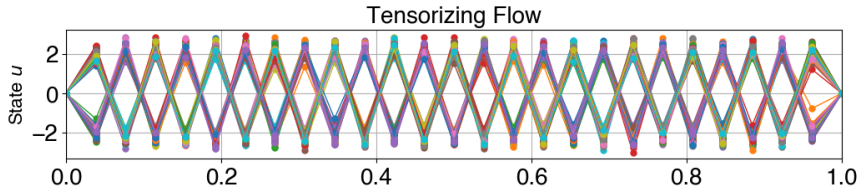


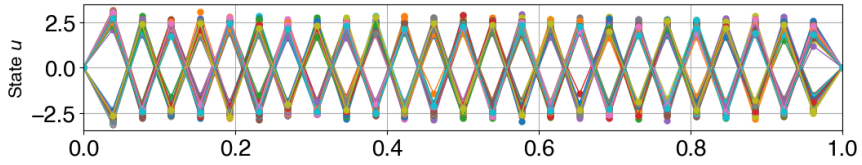
图 4.4：式 (4.3) 中每种模态各自的一个典型样本构型。

图 4.5 和图 4.6 分别绘出了训练前后由 TF 和归一化流生成的典型样本构型。我们发现，训练后的NF生成的样本有时在定性上似乎并不正确；表 4.5 总结并在图 4.7 中绘出的验证损失定量证实了 TF 优于归一化流。

基分布	开始	End	−对数 Z_{true}	误差比
张量列车（秩 2）	222.736	−202.756	−208.014	0.146
高斯	551.411	−172.069		

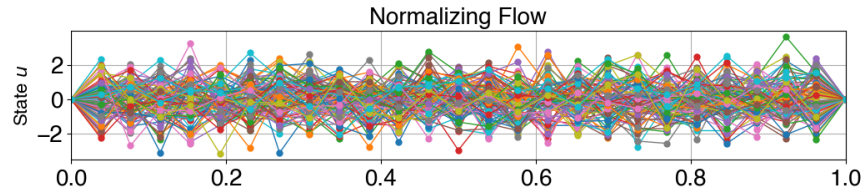


(a) 基准

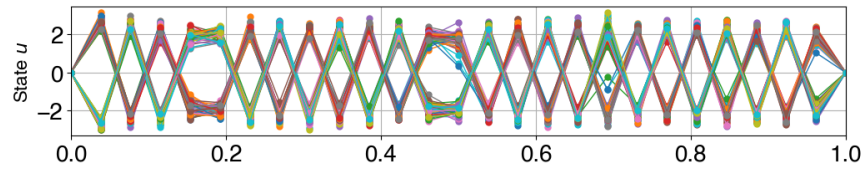


(b) 推前

图 4.5：从张量化流中抽取的一维金兹堡-朗道模型的若干典型样本构型，(a) 为训练前，(b) 为训练后。



(a) 基准



(b) 推前

图 4.6：从归一化流中抽取的一维金兹堡-朗道模型的若干典型样本构型，(a) 为训练前，(b) 为训练后。

表 4.5：将 TF 和 NF 应用于一维 Ginzburg-Landau 模型的训练概要。

4.3.3. 二维金兹堡-朗道模型。本节考虑定义在二维$(d+2) \times (d+2)$ 方形晶格上的类似金兹堡-朗道型模型。我们将格点处标量场的取值记为 $u_{i,\ j}$，

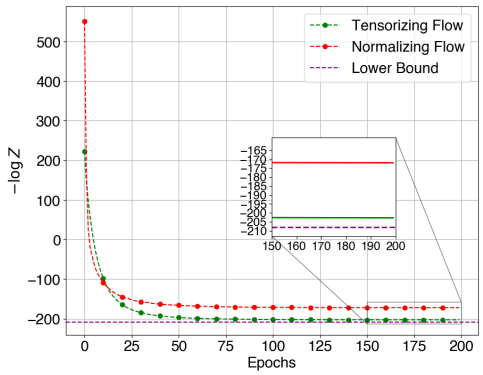


图4.7：1维金兹堡-朗道模型在训练过程中TF和NF的验证损失图。

i，$j = 0$、……、$d+1$，并定义能量：

$$E(\boldsymbol{u}) := \frac{\delta}{2}\left[\sum_{i=1}^{d+1}\sum_{j=1}^{d+1}\left(\frac{u_{i,j}-u_{i-1,j}}{h}\right)^2 + \left(\frac{u_{i,j}-u_{i,j-1}}{h}\right)^2\right] + \frac{1}{4\delta}\sum_{i=1}^{d}\sum_{j=1}^{d}(1-u_{i,j}^2)^2, \tag{4.4}$$

边界条件为

$$u_{0,:} = u_{d+1,:} = \mathbf{1}, \quad u_{:,0} = u_{:,d} = -\mathbf{1}. \tag{4.5}$$

同样，$\delta > 0$ 是一个标量参数，$h > 0$ 是晶格间距。目标分布再次定义为

$$p(\boldsymbol{u}) \propto \exp(-\beta E(\boldsymbol{u})). \tag{4.6}$$

我们设定 $d = 64$、$h = 1/(d+1) = 1/9$、$\beta = 0.2$以及 $\delta = 0.04$。

训练后的TF和NF生成的典型样本构型绘于图4.8中。定性差异在视觉上显而易见。

在图4.9中，我们绘制了以下量的经验估计：

$$\overline{u}_{i,j} := \mathbb{E}[u_{i,j}] \tag{4.7}$$

这些估计由 TF 和 NF 的 $S = 2\times10^4$ 样本以及参考分布p_{true}得到，分别记为 u_{TF}、u_{NF} 和 u_{true}。在图4.10中，我们绘制了 $\overline{u}_{\text{TF}}$ 和 $\overline{u}_{\text{NF}}$的绝对误差，更清楚地展示了 TF 的优势。

通过验证损失再次定量证实了 TF 相较于 NF 的改进；验证损失绘制于图4.11，并汇总于表 4.6。

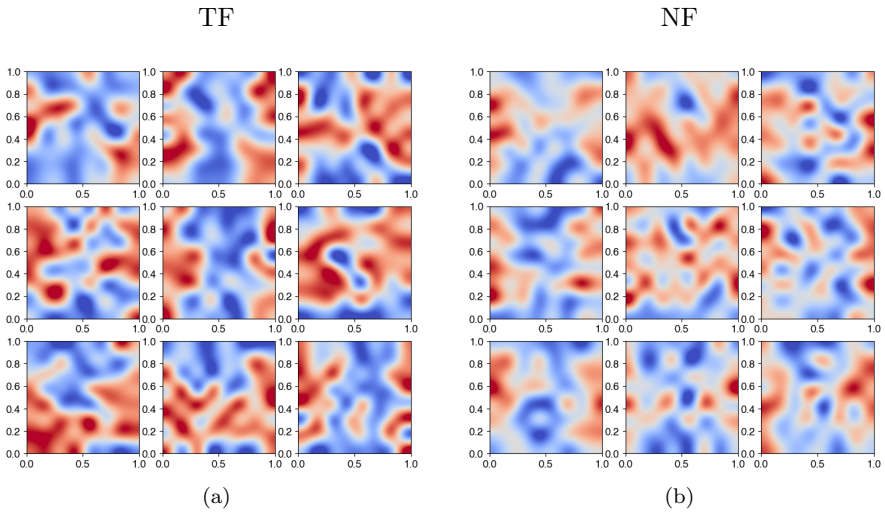


图4.8：从（a）训练后的TF和（b）训练后的NF中绘制的二维金兹堡-朗道模型的几个典型样本构型。

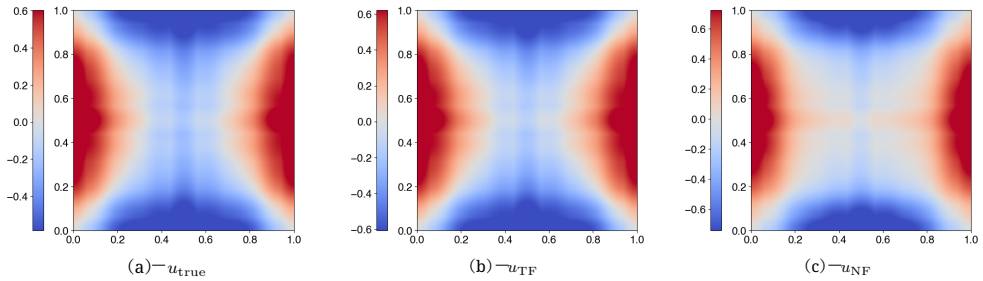


图4.9：GL二维：u 在（4.7）中定义的采样状态的一阶矩估计。$^-$（a）真实值（b）TF（c）NF。所有 u_{true}、 u_{TF}和 $\bar{u}_{\mathrm{NF}}$ 均使用 $S=\bar{2}0000$个样本计算。

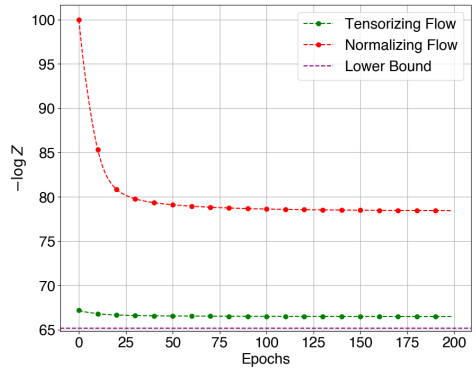
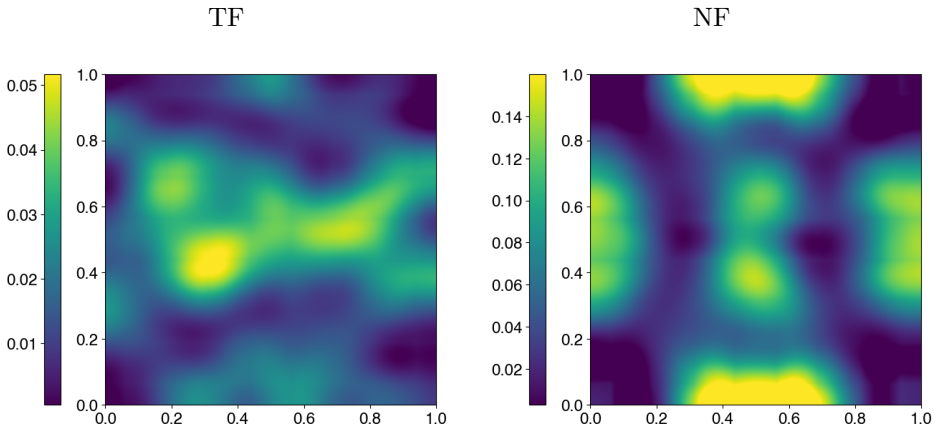


图4.11：20二维金兹堡-朗道模型训练过程中TF和NF的验证损失图。



(a) $|u_{\mathrm{TF}} - u_{\mathrm{true}}|$ (b) $|u_{\mathrm{NF}} - u_{\mathrm{true}}|$

图 4.10：GL二维：(a) u_{TF} 与 u_{true}之间的逐点绝对误差，定义见(4.7)。(b) u_{NF} 与 u_{true} 之间的逐点绝对误差，定义见(4.7)。误差是针对$[0,1]^2$中的每个格点（i, j）绘制的。

基分布	开始	End	−对数 Z_{true}	误差比
秩-3 张量列车	67.202	66.488	65.198	0.0974
$\mathcal{N}(\mathbf{0}, 0.2\mathbf{I}_{64})$	99.979	78.438		

表 4.6：将 TF 和 NF 应用于二维 Ginzburg-Landau 模型的训练概要。

5. 结论。本文提出了一种用于变分推断的灵活试探形式——张量化流，它将神经网络流模型与低秩 TT 基分布相结合。这种基分布可以通过数值线性代数算法获得，并能捕捉与目标分布之间的非平凡重叠。神经网络流模型的表示能力则用于弥合近似差距。最后，在若干高维示例中，更紧的变分下界收敛具体体现了 TT 基测度的有效性。

参考文献

[1] M. S. Albergo, G. Kanwar, and P. E. Shanahan, *Flow-based generative models for markov chain monte carlo in lattice field theory* 100 2019 , Physical Review D, ( ).

[2] Michael S. Albergo and Eric Vanden-Eijnden, *Building normalizing flows with stochastic interpolants* 2022 , . [3] Jens Behrmann, Will Grathwohl David Duvenaud and Jörn-Henrik , Ricky T. Q. Chen, , , , Jacobsen *Invertible residual networks* 2018 ( ). [4] David M. Blei, Alp Kucukelbir *Variational inference: A review* , *for statisticians* 112 2017 859 877 and Jon D. McAuliffe, , 美国统计学会会刊, ( ), 第 – 页。[5] Jacob C Bridgeman and Christopher T Chubb, *Hand-waving and interpretive dance: anintroductory course on tensor networks*, JournalofPhysicsA:Mathematicaland Theoret- ical, 50 (2017年), 第 223001 页。[6] Ricky T. Q. Chen, Jens Behrmann David Duvenaud and Jörn-Henrik Jacobsen *Resid- , , , , , ual flows for invertible generative modeling* 2019

[7] Ricky T. Q. Chen, Yulia Rubanova Jesse Bettencourt and David Duvenaud *Neural , ordinary differential equations* 2018 , , , .[8] Andrzej Cichocki, Namgil Lee Ivan Oseledets Anh-Huy Phan Qibin Zhao and , Danilo P. Mandic , , , , *Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions*, 《机器学习基础与趋势》, 9 (2016年), 第 249–429 页。

[9] Brian Coyle, Daniel Mills, Vincent Danos, and Elham Kashefi, *The born supremacy: quantum advantage and training of an ising born machine*, 《npj量子信息》, 6 (2020年), 第 60 页。

[10] Tiangang Cui and Sergey Dolgov, *Deep composition of tensor-trains using squared inverse rosenblatt transports* 2021 , 《计算数学基础》, ( ).[11] , *Deep composition of tensor-trains using squared inverse rosenblatt transports*, 《计算数学基础》, (2021).

[12] M. Nazareth da Costa, R. Attux, A. Cichocki, and J. M. T. Romano, *Tensor-train networks for learning predictive modeling of multidimensional data* 2021 , .[13] Hussam Al Daas, Grey Ballard and Peter Benner *Parallel algorithms for tensor train , , , . arithmetic* 2020

[14] Lieven De Lathauwer, Bart De Moor and Joos Vandewalle *A multilinear singular value , , , 《SIAM矩阵分析与应用期 decomposition* 21 2000 1253 刊》, ( ), 第 – 1278 页。[15] Laurent Dinh, David Krueger and Yoshua Bengio *Nice: Non-linear independent compo- , , , .*[16] Laurent Dinh, *nents estimation* 2014 Jascha Sohl-Dickstein and Samy Bengio *Density estimation using real , , , .*[17] Sergey Dolgov, *nvp* 2016 Karim Anaya-Izquierdo, Colin Fox, and Robert Scheichl, *Approximation and sampling of multivariate probability distributions in the tensor train decomposition* 2018 , .[18] Matthew Fellows, Anuj Mahajan and Shimon Whiteson *Virel: A* , Tim G. J. Rudner, , , . *variational inference framework for reinforcement learning* 2018

[19] Marylou Gabrié, Grant M. Rotskoff, and Eric Vanden-Eijnden, *Efficient bayesian sampling using normalizing flows to assist markov chain monte carlo methods*, arXiv:2107.08001。

[20] Marylou Gabrié , Grant M. Rotskoff, and Eric Vanden-Eijnden, *Adaptive monte carlo augmented with normalizing flows*, 《美国国家科学院院刊》, 119 (2022年)。[21] Patrick Gelss, *The Tensor-Train Format and Its Applications*, 博士论文, 2017年。[22] Samuel Gershman, Matt Hoffman, and David Blei, *Nonparametric variational inference*, 2012年。[23] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair Aaron Courville and Yoshua Bengio *Generative adversarial net- , , , , . works* 2014

[24] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Du-venaud, *Ffjord: Free-form continuous dynamics for scalable reversible generative models*, 2018年。

[25] Jesper L. Hinrich and Morten Mørup, *Probabilistic tensor train decomposition* 2019 , 收录于第 th 届欧洲信号处 27 1 5 理会议（EUSIPCO）, 2019年, 第 – 页。[26] P.C. Hohenberg and A.P. Krekhov, *An introduction to the ginzburg–landau theory of phase* , 《物理学报告》, ( ), 第 – 页。 *transitions and nonequilibrium patterns* 572 2015 1 42

[27] Sebastian Holtz, Thorsten Rohwedder, and Reinhold Schneider, *The alternating linear scheme for tensor optimization in the tensor train format*, SIAM科学计算期刊, 34 (2012年), 第 A683–A713 页。

[28] Rudolf Peter Huebener, *Ginzburg-Landau Theory*, Springer Berlin Heidelberg, 柏林, 海德堡, 1979年, 第 33–57 页。[29] YH. Hur, J. G. Hoskins, M. Lindsey, E. M. Stoudenmire, and Y. Khoo, *Generative modeling via tensor train sketching* 2022 , .

[30] Sergey Ioffe and Christian Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift* 2015 , .[31] Priyank Jaini, Ivan Kobyzev Yaoliang Yu and Marcus Brubaker *Tails of lipschitz , , , , .*[32] C.A. Jensen, *triangular flows* 2019 R.D. Reed R.J. Marks M.A. El-Sharkawi Jae-Byung Jung R.T. , , , , , , , Miyamoto G.M. Anderson and C.J. Eggen *Inversion of feedforward neural networks: algorithms and applications* 87 1999 1536 1549 , 《IEEE会刊》, ( ), 第 – 页。

[33] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul, *An introduction to variational methods for graphical models*, Machine Learning, 37（1999年），第183–233页。
[34] Boris N. Khoromskij and Christoph Schwab, *Tensor-structured galerkin approximation of parametric and stochastic elliptic pdes*, SIAM科学计算期刊，33（2011年），第364–385页。
[35] Diederik P. Kingma and Jimmy Ba, *Adam: A method for stochastic optimization*, 2014年。
[36] Diederik P. Kingma and Prafulla Dhariwal, *Glow: Generative flow with invertible 1x1 convolutions* 2018 , .
[37] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling, *Semi-supervised learning with deep generative models*, 载于《神经信息处理系统进展》，Z. Ghahramani、M. Welling、C. Cortes、N. Lawrence和K.Q. Weinberger编，第27卷，Curran Associates, Inc.，2014年。
[38] Diederik P Kingma and Max Welling, *Auto-encoding variational bayes*, 2013。[39] Ilya Kisil, Giuseppe G. Calvi, Kriton Konstantinidis, Yao Lei Xu, and Danilo P. Mandic *Reducing computational complexity of tensor contractions via tensor-train net-* , , 。 *works* 2021
[40] Stefan Klus, Patrick Gelss Sebastian Peitz and Christof Schütte *Tensor-based dy-* , *namic mode decomposition* 31 2018 3359 3380 , , , Nonlinearity, （），第 – 页。[41] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker, *Normalizing flows: Anintroduction and review of current methods*, IEEE模式分析与机器智能汇刊，43（2021），第3964–3979页。[42] Tamara G. Kolda and Brett W. Bader, *Tensor decompositions and applications*, SIAM评论，51（2009），第455–500页。
[43] Alejandro Lage-Castellanos, Roberto Mulet, Federico Ricci-Tersenghi, and Tomaso Rizzo *Inference algorithm for finite-dimensional spin glasses: Belief propagation* , , Phys. Rev. E, *on the dual lattice* 84 2011 046706 （），第页。[44] Mike Laszkiewicz, Johannes Lederer, and Asja Fischer, *Copula-based normalizing flows*, 2021。[45] Qianxiao Li, Bo Lin and Weiqing Ren *Computing committor functions for the study of* , , , 化学物理学杂 *rare events using deep learning* 151 2019 054112 志，（），第页。[46] Qianxiao Li, Bo Lin and Weiqing Ren *Computing committor functions for the study of* , , , 化学物理学杂 *rare events using deep learning* 151 2019 054112 志，（），第页。[47] Andriy Mnih and Karol Gregor, *Neural variational inference and learning in belief networks* 2014 , （）。
[48] Georgii S. Novikov, Maxim E. Panov, and Ivan V. Oseledets, *Tensor-train density es-timation*, 载于《第三十七届人工智能不确定性会议论文集》，Cassio de Campos和Marloes H. Maathuis编，《机器学习研究论文集》第161卷，PMLR，2021年7月27日至30日，第1321–1331页。[49] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu, *Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning*, Science，365（2019年），第eaaw1147页。[50] Román Orús, *A practical introduction to tensor networks: Matrix product states and projected entangled pair states* 349 2014 117 158 , 物理学年鉴，（），第–页。[51] Ivan Oseledets and Eugene Tyrtyshnikov, *Tt-cross approximation for multidimensional arrays* 432 2010 70 88 , 线性代数及其应用，（），第–页。[52] I. V. Oseledets, *Tensor-train decomposition*, SIAM科学计算期刊，33（2011），第2295–2317页。[53] G. Papamakarios, E. Nalisnick, D. Rezende, S. Mohamed, and B. Lakshminarayanan, *Normalizing flows for probabilistic modeling and inference* 22 2021 , J. Mach. Learn. Res.，（），第1–64页。[54] D. Rezende and S. Mohamed, *Variational inference with normalizing flows*, 载于国际机器学习大会，2015年。[55] Abram Rodgers and Daniele Venturi, *Implicit step-truncation integration of nonlinear pdes on low-rank tensor manifolds* 2022 , 。[56] Tianyi Shi, Maximilian Ruth and Alex Townsend *Parallel algorithms for computing the* , , , 。 *tensor-train decomposition* 2021
[57] Maciej Skorski, *Modern analysis of hutchinson's trace estimator*, 载于2021年第55届年度信息科学与系统会议（CISS），2021，第1–5页。[58] Micheline B. Soley, Paul Bergold, Alex A. Gorodetsky, and Victor S. Batista, *Functional tensor-train chebyshev method for multidimensional quantum dynamics simulations*, 化学理论与计算杂志，18（2021），第25–36页。[59] Joshua S. Speagle, *A conceptual introduction to markov chain monte carlo methods*, 2019年。

[60] Vincent Stimper, Bernhard Schölkopf and José Miguel Hernández-Lobato *Resam-* , , , *pling base distributions of normalizing flows* 2021 ( ).[61] E. Miles Stoudenmire and David J. Schwab *Supervised learning with quantum-inspired tensor networks* 2016 , ( ).

[62] E. G. Tabak and C. V. Turner, *A family of nonparametric density estimation algorithms*,《纯粹与应用数学通讯》, 66 (2013), 第 145 页。[63] E. G. Tabak and E. Vanden-Eijnden, *Density estimation by dual ascent of the log-likelihood*,《数学科学通讯》, 8 (2010), 第 217 页。

[64] Robert Tempke and Terence Musho *Autonomous design of new chemical reactions using a variational autoencoder* 5 2022 40 ,《通讯化学》, ( ), 第 页。

[65] Michalis K. Titsias, *Learning model reparametrizations: Implicit variational inference by fitting mcmc distributions* 2017 , .[66] Yu Wang, Fang Liu *Variational inference with nofas: Normal-* , *izing flow with adaptive surrogate for computationally expensive models* 2021 and Daniele E. Schiavazzi, , .[67] Steven R. White, *Density-matrix algorithms for quantum renormalization groups*,《物理评论 B》, 48 (1993), 第 10345–10356 页。

[68] Mike Wu, Kristy Choi Noah Goodman and Stefano Ermon *Meta-amortized variational* , , , , . *inference and learning* 2019

[69] Le Xu, Lei Cheng Ngai Wong and Yik-Chung Wu *Tensor train factorization and com-* , , , , . *pletion under noisy data with prior analysis and rank estimation* 2020