arXiv:2303.04156v1 [cs.LG] 7 Mar 2023

机器学习中的范畴计算*

Eli Sennesh[1],Tom Xu[2], 和 Yoshihiro Maruyama[2]

[1] 美国波士顿东北大学 sennesh.e@northeastern.edu[2] 澳大利亚堪培拉澳大利亚国立大学{ tom.xu,yoshihiro.maruyama}@anu.edu.au

摘要。 范畴论已成功应用于科学的各个领域，揭示了统一多种不同现象的普遍原理，从而促进了这些领域之间的知识迁移。近年来，人们开始探索其在机器学习中的应用，但抽象的数学基础与机器学习任务的具体应用之间仍存在差距。本文介绍 DisCoPyro，将其作为一种范畴结构学习框架，结合范畴结构（如对称幺半范畴和运算子）与摊销变分推断，并可应用于变分自编码器的程序学习等领域。我们既提供数学基础，也介绍具体应用，并将其实验性能与其他模型（如神经符号模型）进行比较。我们推测，DisCoPyro 最终有望为通用人工智能的发展作出贡献。

关键词： 结构学习 · 程序学习 · 对称幺半范畴 · 拟运算子 · 摊销变分贝叶斯推断。

1 引言

范畴论已应用于数学科学的多个领域，使我们能够发现统一不同数学现象的普遍原理，并由此促进它们之间的知识迁移 [7]。近年来，人们开始探索其在机器学习中的应用 [21]；然而，基础数学与具体机器学习任务中的实际应用之间仍存在巨大鸿沟。本研究旨在弥合这一鸿沟。我们提出范畴结构学习框架 DisCoPyro，这是一种采用摊销变分推断的概率生成模型。我们既提供其数学基础，也通过一个示例应用将其与其他神经符号模型进行比较。

在此，我们阐述为何相信 DisCoPyro 长远而言能够为开发人类水平的通用人工智能作出贡献。人类智能支持程度化的统计推理 [15], ，并且在我们将其用于符号（代数）定义域之前，便已演化出表征空间（几何）定义域的能力。对称幺半范畴为以下工作提供了数学框架：

* 本研究得到 JST人工智能机器人登月计划（JPMJMS2033-02）的支持。

构造符号计算（如本文所示）和几何空间（例如 [17]）。我们采纳 Lake [15]通过概率论表征程度化统计推理的建议，并将神经网络整合到变分推断中，以提高计算的可处理性。在应用方面，通过变分贝叶斯，我们取得了有竞争力的性能（见下文第 3.2 小节），无需像模块化神经网络 [13,20]那样通过强化学习来学习结构。

本文其余部分安排如下。第 2 节首先介绍 DisCoPyro 的数学基础（第 2 节和第 2.1 节）。接着在第 3 节中，我们说明如何针对某项任务训练 DisCoPyro（第 2.1 节），并提供实验结果与性能比较（第 3.2 节）。图 1 展示了示例任务训练过程中的执行流程。我们在第 4 节总结全文，并讨论进一步的应用。示例实现见 https://github.com/neu-pml/discopyro，实验见 https://github.com/esennesh/categorical bpl。DisCoPyro 构建于 Pyro [2]（一种深度通用概率编程语言）、DisCoCat [4]（一种用于自然语言处理 [4]的分布式组合模型）以及用于范畴计算的 DisCoPy [6] 库之上。

1.1 记号

本文以对称幺半范畴（SMCs）$\mathcal{C}$ 及其对应的拟运算子 $\mathcal{O}$ 作为数学框架。欢迎读者参阅 Fong [7]，了解相关介绍。SMCs 由对象 $Ob(\mathcal{C})$ 和以下集合构成：

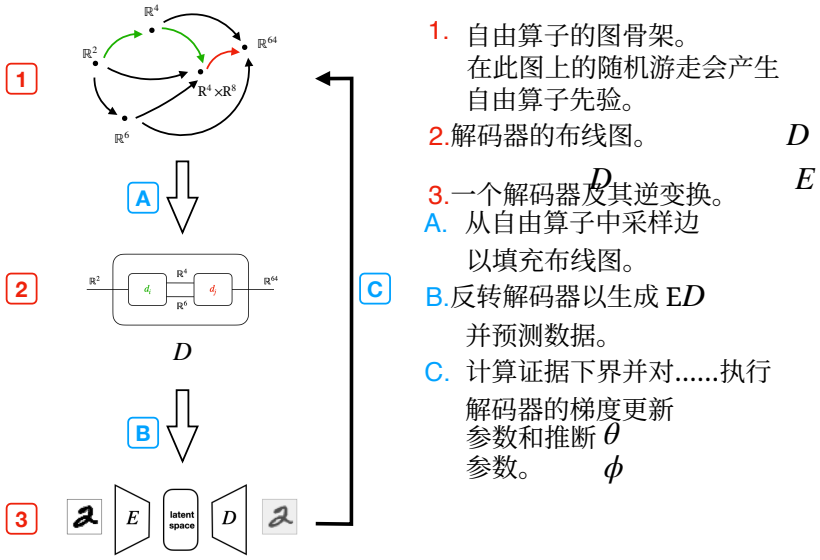


图 1：实验示例。在每个训练轮次中，DisCoPyro 根据布线图从其骨架中采样变分自编码器结构，并学习这些结构的忠实逆，将其作为近似后验。

对象之间的态射 $\mathcal{C}$（τ_1，τ_2）集合 τ_1，$\tau_2 \in Ob$（$\mathcal{C}$）。拟运算子由类型 Ty（$\mathcal{O}$）以及类型之间的态射 $\mathcal{O}$（τ_1，τ_2）集合 τ_1，$\tau_2 \in Ty$（$\mathcal{O}$）构成。SMC 通常写作（$\mathcal{C}$，$\otimes$，I），其中包含作用于对象和态射的乘积运算 $\otimes$，以及一个单位元 I，属于 $\otimes$。在这两种框架中，每个对象/类型 τ 都有唯一的恒等态射 id_τ。范畴支持对态射进行复合 $g \circ f$，而拟运算子支持对态射进行带索引的复合 $g \circ_i f$（适用于 $i \in \mathbb{N}$）。

2 DisCoPyro 的基础

本质上，下面的定义 1 从一个对称幺半范畴中揭示出有限个构件（生成元），以及通过将这些生成元与 $\circ$ 和 $\otimes$复合而构造的态射。例如，在可执行程序范畴中，有限元签名指定了一种领域特定编程语言。

定义 1（对称幺半范畴中的有限元签名）。给定一个对称幺半范畴（对称幺半范畴） $\mathcal{C}$，其对象记为 $Ob(\mathcal{C})$，该对称幺半范畴中的有限元签名 $\mathcal{S}$ $=(O, M)$ 由以下部分组成：

– 一个有限集 $O \subseteq Ob(\mathcal{C})$；以及– 一个有限集 M，由形如 $m : \mathcal{C}(\tau_1, \tau_2)$ 的元素组成，其中某些 $\tau_1, \tau_2 \in O$ 满足 $\forall \tau \in O, m \neq id_\tau$。

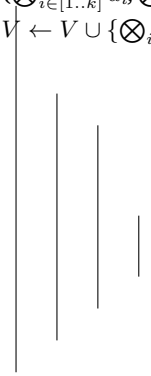
以下这个基于有限元签名的自由算子，表示由上述构件（由有限元签名指定的生成元）合成的所有可能程序所构成的空间。采用拟运算子而非单纯的范畴，使我们能够将复合视为嵌套而非仅仅是传递组合；采用拟运算子而非单纯的文法，则使我们能够推理运算的输入和输出，而不仅仅是其输出。

定义 2（由签名生成的自由拟运算子）。自由拟运算子 $\mathcal{O}_\mathcal{S}$ 由签名 $\mathcal{S} = (O, M)$ 由以下部分组成

– 一个类型（表示）集合 $Ty(\mathcal{O}_\mathcal{S}) = \{I\} \cup O^\otimes$；– 对每个 $n \in \mathbb{N}^+$，一个运算（映射）集合 $\mathcal{O}_\mathcal{S}(\tau_0, \ldots, \tau_{n-1}; \tau_n)$，由有限个分支和叶节点构成的所有树组成，其中每个具有 $n-1$ 个子节点的顶点 v 都标记为一个生成器 $m(v) \in M$，使得 $dom(m(v))$ 的乘积长度为 $n-1$；– 对每个 $\tau \in Ty(\mathcal{O}_\mathcal{S})$，一个恒等运算 id_τ: $\mathcal{O}_\mathcal{S}(\tau; \tau)$；以及 – 一个代入算子 $\circ_i$，其定义为：当 $i \in [1...n-1]$ 时，将语法树 $\mathcal{O}_\mathcal{S}(\sigma_1, \ldots, \sigma_m; \tau_i)$ 嵌套到另一个 $\mathcal{O}_\mathcal{S}(\tau_1, \ldots, \tau_{n-1}; \tau_n)$ 中，从而生成语法树 $\mathcal{O}_\mathcal{S}(\tau_1, \ldots, \tau_{i-1}, \sigma_1, \ldots, \sigma_m, \tau_{i+1}, \ldots \tau_{n-1}; \tau_n)$。

直观地说，自由歌剧与上下文无关文法有许多共同之处；事实上，Hermida [10] 证明了它们都可以表示为有向无环超图。对称幺半范畴中的签名定义已暗示了相应超图的结构，但算法 1 会将其明确表示出来，并在超图中添加边，以表示将各个运算嵌套并行组合（等价地说，即原始对称幺半范畴中的幺半积）。在算法 1 生成的超图中，每个顶点都对应一个非乘积

输入：签名 $\mathcal{S} = (O, M)$ 输出：超图 $H = (V, E)$，递归位置 $RV \leftarrow O$;$E \leftarrow$ map($\lambda m.$(dom(m), cod(m)) M);$R \leftarrow \emptyset$; 栈 $\leftarrow \{v \in V \mid |v| > 1\}$; 当栈 $\neq \emptyset$ 非空时 ty $\leftarrow$pop(stack); inhabitants $\leftarrow$ map($\lambda c.$\{(dom(e) c) $\mid e \in E$, cod(e) $= c$}, chunks(ty V)); 对每个 $((d_1, c_1) \ldots, (d_k, c_k)) \in \bigotimes$ inhabitants, 若 not sublist($\bigotimes_{i\in[1..k]} d_i$, ty) 则 $R \leftarrow R \cup \{\otimes[(d_1, c_1) \ldots, (d_k, c_k)]\} E \leftarrow E \cup \{(\bigotimes_{i\in[1..k]} d_i, \bigotimes_{i\in[1\ldots k]} c_i)\}$; 若 $d \notin V$ 则 push(stack d); $V \leftarrow V \cup \{\bigotimes_{i\in[1..k]} d_i\}$; 结束 结束 结束 结束 返回 (V, E) R



A算法 1： 将自由拟运算子表示为超图的算法。
chunks 函数将 ty 划分为多个子列表，每个子列表都是集合 V 的一个元素。

类型，并且每条超边都有一个顶点列表作为其定义域和陪域。每个这样的超图也都可以表示为图，其中超边充当节点，其定义域和陪域中的列表充当边。我们将使用这种图表示 $G \simeq H$，把态射视为其定义域与陪域之间的路径来进行推理。

我们将根据这一图表示的有向邻接矩阵 A_G，推导自由拟运算子上态射的概率生成模型。

定义 3（有向图中的转移距离）。两个带索引的顶点 v_i、v_j 之间的“转移距离”，是指数化邻接矩阵／转移矩阵中 i、j 项的负对数

$$d(v_i, v_j) = -\log\left(\left[e^{A_G}\right]_{i,j}\right), \tag{1}$$

其中矩阵指数由以下级数定义

$$e^{A_G} = \sum_{n=1}^{\infty} \frac{(A_G)^n}{n!}.$$

对这一转移距离的软最小化分布，在期望意义下并保持索引目标顶点不变，将定义一个在超图上生成路径的概率生成模型。

数据： 超图（V， E）
函数 $Path(\boldsymbol{\tau}_-, \boldsymbol{\tau}_+, \beta, \mathbf{w})$
  $i \leftarrow 1$;
  $\boldsymbol{\tau}_i \leftarrow \boldsymbol{\tau}_-$;
  $f \leftarrow id_{ty-}$;
  当 $\boldsymbol{\tau}_i \neq \boldsymbol{\tau}_+$ **时执行**
    $e_i \sim \pi(e \in E \mid \boldsymbol{\tau}_i,\ \boldsymbol{\tau}_+\ \beta)$;
    $\boldsymbol{\tau}_i \leftarrow \mathrm{cod}(e_i)$;
    $f \leftarrow f \fatsemi$ 生成器$(e_i,\ \beta,\ \mathbf{w})$;
    $i \leftarrow i + 1$;
  end
  返回 f
end

算法 2： 构造类型之间路径的马尔可夫链

定义 4（自由算子先验）。考虑一个签名 $\mathcal{S} = (O, M)$ 及其生成的图表示 $G = (V, E)$ 和递归位置 R，然后以由图中顶点表示的定义域和陪域 $\boldsymbol{\tau}_-, \boldsymbol{\tau}_+ \in Ty(\mathcal{O}_{\mathcal{S}})$ 为条件。自由算子先验通过吸收马尔可夫链，为所有有限路径$\mathbf{e} = (e_1, e_2, \ldots, e_n)$ 赋予概率密度，其中 $\mathrm{dom}(\mathbf{e})$ 为 $= \boldsymbol{\tau}_-$，且 $\mathrm{cod}(\mathbf{e})$ 为 $= \boldsymbol{\tau}_+$。首先，模型会采样一个“精度” β 和一组“权重”$\mathbf{w}$

$$\beta \sim \gamma(1,1) \qquad\qquad \mathbf{w} \sim \mathrm{Dirichlet}\left(\vec{\mathbf{1}}^{(|M|+|R|)}\right).$$

然后，它通过对转移距离进行软最小化（偏向于较短的路径，偏向程度为 β），从吸收马尔可夫链（在算法 *2* 中定义）中采样一条路径

$$\pi(e \in E \mid \boldsymbol{\tau}_1, \boldsymbol{\tau}_2; \beta) := \frac{\exp\left(-\frac{1}{\beta} d(\mathrm{cod}(e), \boldsymbol{\tau}_2)\right)}{\sum_{e' \in E : \mathrm{dom}(e') = \boldsymbol{\tau}_1} \exp\left(-\frac{1}{\beta} d(\mathrm{cod}(e'), \boldsymbol{\tau}_2)\right)}. \tag{2}$$

方程 *2* 将诱导出一个转移算子 T，根据 *La- touche* 和 *Ramaswami* *[16]*，中的定理 *2.5.3*，该算子几乎必然会到达与 $\boldsymbol{\tau}_2$ 对应的吸收态。随后可按照算法 *3* 填充这条路径。精度 β 在每次递归时都会提高，从而以较短的路径终止。我们将诱导出的联合分布记为

$$p(f, \mathbf{w}, \beta; \boldsymbol{\tau}_-, \boldsymbol{\tau}_+) = p(f \mid \beta, \mathbf{w}; \boldsymbol{\tau}_-, \boldsymbol{\tau}_+) p(\mathbf{w}) p(\beta). \tag{3}$$

有了定义在某个签名上的自由算子中运算的概率生成模型后，我们现在需要一种方式来指定结构学习问题。定义 5 通过指定要采样哪些路径（每个方框都指定一次对算法 2 的调用）以及如何组合这些路径，给出了这种方式。

定义 5（布线图）。无环的、带 O 类型的布线图 *[22,7]*是一个映射，其定义域为一系列内部方框，每个方框都由其定义域和

数据： 超图 (V, E)、生成元 M、递归位置 R**函数** $Generator(e, \beta, \mathbf{w})$

$gs \leftarrow \{m \in M \mid (\text{dom}(m), \text{cod}(m)) = (\text{dom}(e), \text{cod}(e))\}$; $gs \leftarrow gs \cup \{\otimes[(d_1, c_1) \ldots, (d_k, c_k)] \in R \mid \bigotimes_{i\in[1\ldots k]} d_i = \text{dom}(e) \wedge \bigotimes_{i\in[1\ldots k]} c_i = \text{cod}(e)\}$; foreach $j \in \{1, \ldots, |gs|\}$ do if $gs_j = \otimes(\ldots)$ then $\mathrm{w}_j \leftarrow \mathrm{w}_j/\beta$; end end $\mathrm{w}_e \leftarrow [\mathrm{w}_n \mid g \in gs, g \in M, n = \text{index}(g, M)]$; $\mathrm{w}_e \leftarrow \mathrm{w}_e + [\mathrm{w}_{|M|+n} \mid g \in gs, g \in R, n = \text{index}(g, R)]$; $j \sim$ 离散(w_e); if $gs_j = \otimes[(d_1, c_1) \ldots, (d_k, c_k)]$ then return $\bigotimes_{l=1}^{k}$ 路径$(d_l, c_l, \beta + 1, \mathrm{w})$ else return gs_jend end

算法 3：用态射填充路径中的一条边

陪域对 $(\boldsymbol{\tau}_i^-, \boldsymbol{\tau}_i^+)$ 对应一个外框，该外框由定义域和陪域$(\boldsymbol{\tau}_n^-, \boldsymbol{\tau}_n^+)$定义

$$\Phi : \mathcal{O}_{\mathcal{S}}(\boldsymbol{\tau}_1^-, \boldsymbol{\tau}_1^+) \times \ldots \times \mathcal{O}_{\mathcal{S}}(\boldsymbol{\tau}_{n-1}^-, \boldsymbol{\tau}_{n-1}^+) \to \mathcal{O}_{\mathcal{S}}(\boldsymbol{\tau}_n^-, \boldsymbol{\tau}_n^+).$$

无环性要求连接（“线”）只能从外框的定义域延伸至内部方框的定义域，从内部方框的陪域延伸至外框的陪域，以及在内部方框之间延伸，并且内部方框之间的连接在有向图中不能形成环。

给定用户指定的**布线图** Φ，我们便可以写出生成模型中所有潜变量的完整先验分布。

$$p(f, \mathbf{w}, \beta; \Phi, \mathcal{S}) = p(\beta)p(\mathbf{w}) \prod_{(\boldsymbol{\tau}_i^-, \boldsymbol{\tau}_i^+) \in \Phi} p(f_i \mid \beta, \mathbf{w}; \boldsymbol{\tau}_i^-, \boldsymbol{\tau}_i^+). \tag{4}$$

如果用户提供似然 $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z} \mid f)$，将学习到的结构 f 与数据$\mathbf{x}$（通过潜变量$\mathbf{z}$）联系起来，我们就能得到一个联合密度

$$p(\mathbf{x}, \mathbf{z}, f, \mathbf{w}, \beta; \Phi, \mathcal{S}) = p(\mathbf{x} \mid f)p(f, \mathbf{w}, \beta; \Phi, \mathcal{S}), \tag{5}$$

而方程 5 随后便可通过贝叶斯反演，从数据$\mathbf{x}$ 进行推断

$$p(\mathbf{z}, f, \mathbf{w}, \beta \mid \mathbf{x}; \Phi, \mathcal{S}) = \frac{p(\mathbf{x}, \mathbf{z}, f, \mathbf{w}, \beta; \Phi, \mathcal{S})}{p_{\boldsymbol{\theta}}(\mathbf{x}; \Phi, \mathcal{S})}. \tag{6}$$

第 2.1 节将说明如何通过随机梯度优化近似方程 6，从而得到 $\boldsymbol{\theta}$ 的最大似然估计，以及参数族 $\boldsymbol{\phi}$ 对真实贝叶斯逆的最优近似。

2.1 模型学习与变分贝叶斯推断

贝叶斯反演依赖于对模型证据 $p_{\boldsymbol{\theta}}(\mathbf{x};\, \Phi,\, \mathcal{S})$ 的评估，而它通常没有闭式解。不过，我们可以将针对联合密度的高维积分转换为一个期望

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{x}; \Phi, \mathcal{S}) &= \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}, f_{\boldsymbol{\theta}}, \mathbf{w}, \beta; \Phi, \mathcal{S}) d\mathbf{z}\, df_{\boldsymbol{\theta}}\, d\mathbf{w}\, d\beta \\ &= \mathbb{E}_{p(f_{\boldsymbol{\theta}}, \mathbf{w}, \beta; \Phi, \mathcal{S})}\left[p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}, f_{\boldsymbol{\theta}}, \mathbf{w}, \beta; \Phi, \mathcal{S})\right], \end{aligned}$$

然后将该期望改写为关于提议分布的期望

$$\begin{aligned} &\mathbb{E}_{p(f_{\boldsymbol{\theta}}, \mathbf{w}, \beta; \Phi, \mathcal{S})}\left[p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}, f_{\boldsymbol{\theta}}, \mathbf{w}, \beta; \Phi, \mathcal{S})\right] = \\ &\qquad\qquad \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}, f_{\boldsymbol{\theta}}, \mathbf{w}, \beta \mid \mathbf{x}; \Phi, \mathcal{S})}\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}, f_{\boldsymbol{\theta}}, \mathbf{w}, \beta; \Phi, \mathcal{S})}{q_{\boldsymbol{\phi}}(\mathbf{z}, f_{\boldsymbol{\theta}}, \mathbf{w}, \beta \mid \mathbf{x}; \Phi, \mathcal{S})}\right]. \end{aligned}$$

为了构造这一期望，DisCoPyro 提供了第 3.1 节所述的函子式反演，以及一种适用于任何通用概率程序的自动结构化变分推断 [1] 摊销形式。

詹森不等式指出，对数密度比的期望是对数期望密度比的下界

$$\begin{aligned} &\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}, f_{\boldsymbol{\theta}}, \mathbf{w}, \beta \mid \mathbf{x}; \Phi, \mathcal{S})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}, f_{\boldsymbol{\theta}}, \mathbf{w}, \beta; \Phi, \mathcal{S})}{q_{\boldsymbol{\phi}}(\mathbf{z}, f_{\boldsymbol{\theta}}, \mathbf{w}, \beta \mid \mathbf{x}; \Phi, \mathcal{S})}\right] \leq \\ &\qquad\qquad \log \mathbb{E}_{p(f_{\boldsymbol{\theta}}, \mathbf{w}, \beta; \Phi, \mathcal{S})}\left[p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}, f_{\boldsymbol{\theta}}, \mathbf{w}, \beta; \Phi, \mathcal{S})\right], \end{aligned}$$

因此，左侧为真实模型证据提供了一个下界

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}, f_{\boldsymbol{\theta}}, \mathbf{w}, \beta \mid \mathbf{x}; \Phi, \mathcal{S})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}, f_{\boldsymbol{\theta}}, \mathbf{w}, \beta; \Phi, \mathcal{S})}{q_{\boldsymbol{\phi}}(\mathbf{z}, f_{\boldsymbol{\theta}}, \mathbf{w}, \beta \mid \mathbf{x}; \Phi, \mathcal{S})}\right] \leq \log p_{\boldsymbol{\theta}}(\mathbf{x}; \Phi, \mathcal{S}).$$

通过蒙特卡洛估计其值和梯度（使用 Pyro 内置的梯度估计器）来最大化此证据下界（ELBO），将通过最大似然估计模型参数 $\boldsymbol{\theta}$，并训练提议参数 $\boldsymbol{\phi}$，使其近似贝叶斯逆（方程 6） [12]。

3 示例应用与训练

通过似然将态射与数据连接起来，并引入中间潜在随机变量，这一框架支持广泛的应用。本节将展示 DisCoPyro 框架由此具备的能力。第 3.1 节将介绍该框架在用于生成式建模的深度概率程序学习中的一个应用示例。第 3.2 节将介绍该应用作为生成模型时的性能。

3.1 使用 DisCoPyro 进行深度概率程序学习

作为一项演示性实验，我们构造了一个拟运算子 $\mathcal{O}$，其生成元实现了用于深度生成模型的 Pyro 构件 $f_{\boldsymbol{\theta}}$（取自

模型	图像尺寸学习结构 log-		$\hat{z}$/dim
序贯注意力 [19]	28x28	✗	-0.1218
变分同胚编码器 [11] (PixelCNN)	28x28	✗	-0.0780
图变分自编码器 [9]	28x28	✓	-0.1334
生成式神经符号 [5]	105x105	✓	-0.0348
自由算子 DGM（我们的方法）	28×28	✓	**-0.0148**

表1：各模型在 Omniglot评估集上的平均对数证据。我们的自由算子模型在每个数据维度上获得了最高的更高的对数证据。

关于结构化变分自编码器的研究 [12,19,24]），其参数为 $\boldsymbol{\theta}$。随后，我们指定了单盒布线图 $\Phi : (I,\ \mathbb{R}^{28\times 28}) \to (I,\ \mathbb{R}^{28\times 28})$，以参数化 DisCoPyro 生成模型。我们在 MNIST 上训练由此得到的自由算子模型，只为检验它是否可行；随后又将其用于下采样后的（28×28）Omniglot数据集上的少样本学习 [14] 这一挑战。由于数据 $\mathbf{x} \in \mathbb{R}^{28\times 28}$，我们的实验设置诱导出联合似然

$$p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}, f_{\boldsymbol{\theta}}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}, f_{\boldsymbol{\theta}}), \boldsymbol{I\tau})$$
$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z} \mid f_{\boldsymbol{\theta}}) = p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}, f_{\boldsymbol{\theta}}) p_{\boldsymbol{\theta}}(\mathbf{z} \mid f_{\boldsymbol{\theta}}).$$

DisCoPyro 通过神经提议分布 $q_{\boldsymbol{\phi}}$，对其自身的随机变量进行摊销变分推断：针对“置信度” $\beta \sim q_{\boldsymbol{\phi}}(\beta \mid \mathbf{x})$，以及针对生成元上的“偏好” $\mathbf{w} \sim q_{\boldsymbol{\phi}}(\mathbf{w} \mid \mathbf{x})$。运行核心 DisCoPyro 生成模型以处理结构 $f_{\boldsymbol{\theta}}$，即可得到自由算子中态射的提议分布，为 DisCoPyro 的潜变量提供通用提议分布

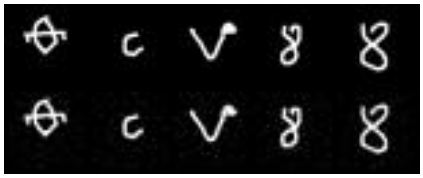
$$q_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}, \mathbf{w}, \beta \mid \mathbf{x}; \Phi, \mathcal{S}) = p(f_{\boldsymbol{\theta}} \mid \mathbf{w}, \beta; \Phi, \mathcal{S}) q_{\boldsymbol{\phi}}(\beta \mid \mathbf{x}) q_{\boldsymbol{\phi}}(\mathbf{w} \mid \mathbf{x}).$$

由于我们示例应用中的态射是深度生成模型的组成部分，因此只需将每个生成态射“翻转过来”，即可得到相应的提议分布神经网络设计。我们将该提议分布指定为 $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}\ f_{\boldsymbol{\theta}})$；它通过匕首函子以复合方式构造忠实逆 [23]（关于将贝叶斯反演描述为匕首函子的进一步说明，请参阅 Fritz [8]）。这样，我们的应用就拥有了完整的提议密度

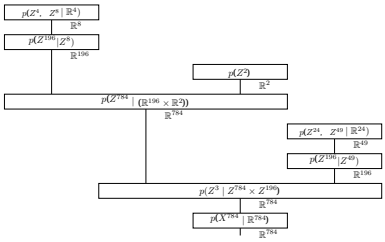
$$q_{\boldsymbol{\phi}}(\mathbf{z}, f_{\boldsymbol{\theta}}, \mathbf{w}, \beta \mid \mathbf{x}; \Phi, \mathcal{S}) = q_{\boldsymbol{\phi}}(\mathbf{z} \mid f_{\boldsymbol{\theta}}, \mathbf{x}) q_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}, \mathbf{w}, \beta \mid \mathbf{x}; \Phi, \mathcal{S}). \tag{7}$$

3.2 实验结果与性能比较

表 1 将我们的自由算子模型与其他结构化深度生成模型的性能进行了比较。我们报告了估计的模型对数证据。我们的深度生成模型自由算子先验在每个数据维度上的对数证据最佳，不过似乎没有可供比较的基线标准差。一些较早的基线模型，例如序列注意力模型和变分同胚编码器，会预先固定复合结构，而不像我们这样从数据中学习复合结构。图 2 展示了训练后模型的后验分布样本，包括评估数据的重构（图 2a）以及该数据对应的一个示例结构（图 2b）。



(a) Omniglot 字符（上）及其重构（下）



(b) 从自由算子模型的贝叶斯逆中采样得到的弦图。

图 2：在 Omniglot评估集的手写字符上，图示生成模型（右）通过推断生成的重构结果（左）。弦图展示了一个模型：它生成一个局部视图，通过变分阶梯解码器将其解码到图像画布中，然后执行一个更简单的过程，生成另一个局部视图并将其插入画布。

早在 2015 年，Lake [14]就提出了 Omniglot 数据集，旨在挑战机器学习领域，让研究者通过从极少量样本中学习单个生成模型，实现类人的概念学习；Omniglot 挑战要求模型能够执行分类、潜在特征识别、从类型生成概念，以及生成某个概念的范例。深度生成模型研究领域一直致力于开发能够对未见过的字符进行少样本重构的模型。 [19] 和 [11] 将模型架构固定为常量，试图用静态维度来描述数据中的组合结构。相比之下， [9] 和 [5] 与我们一样，进行了联合结构学习、潜变量推断和数据重建。

4 讨论

本文介绍了用于生成式贝叶斯结构学习的 DisCoPyro 系统，以及其变分推断训练流程和一个示例应用。第 2 节阐述了 DisCoPyro 在范畴论、算子理论和变分贝叶斯推断方面的数学基础。第 3 节表明，在一个挑战性数据集上，DisCoPyro 的表现可与其他模型媲美。

正如 Lake [15] 所指出的，（深度）概率程序可以在比手写字符更多的领域中建模人类智能。除了程序、神经网络架构或可三角剖分流形之外，研究者还将算子和序贯蒙特卡洛方法应用于化学反应网络、自然语言处理，以及人类智能的系统性 [18,3]。如此广泛的应用促使我们关注如何用算子结构来表示一般智能体必须解决的问题，并从数据中与其内容一同学习这些结构。

参考文献

1. Ambrogioni, L. 等：自动结构化变分推断。载于：AIS-TATS论文集，第676–684页（2021） 2. Bingham, E. 等：Pyro：深度通用概率编程。机器学习研究杂志 20，第973–978页（2019） 3. Bradley, T.D.：什么是应用范畴论？arXiv预印本 arXiv:1809.05923（2018） 4. Coecke, B., Sadrzadeh, M., Clark, S.：组合分布式语义模型的数学基础。arXiv预印本 arXiv:1003.4394（2010） 5. Feinman, R., Lake, B. M.：通过生成式神经符号建模学习任务通用表征。载于：国际学习表征会议论文集（2021） 6. de Felice, G., Toumi, A., Coecke, B.：Discopy：Python中的幺半范畴。arXiv预印本 arXiv:2005.02975（2020） 7. Fong, B., Spivak, D.I.：《组合性七讲：应用范畴论入门》。剑桥大学出版社（2019） 8. Fritz, T.：马尔可夫核、条件独立和充分统计量定理的合成方法。数学进展 370，107239（2020） 9. He, J. 等：具有联合优化潜在依赖结构的变分自编码器。载于：2019年国际学习表征会议论文集，第1–16页（2019） 10. Hermida, C., Makkaiy, M., Power, J.：高维多重图。计算机科学逻辑研讨会论文集，第199–206页（1998） 11. Hewitt, L.B. 等：变分同胚编码器：学习如何从少量示例中学习高容量生成模型。2018年不确定性人工智能会议论文集，第988–997页（2018） 12. Kingma, D.P., Welling, M.：自编码变分贝叶斯。arXiv预印本 arXiv:1312.6114（2013） 13. Kirsch, L., Kunze, J., Barber, D.：模块化网络：学习分解神经计算。载于：2018年神经信息处理系统大会论文集，第2408–2418页（2018） 14. Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B.：Omniglot挑战：三年进展报告。行为科学当前观点 29，第97–104页（2019） 15. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.：构建像人一样学习和思考的机器。行为与脑科学 40，e253（2017） 16. Latouche, G., Ramaswami, V.：《随机建模中的矩阵解析方法导论》。工业与应用数学学会（1999） 17. Milnor, J.：关于具有CW复形同伦类型的空间。美国数学会汇刊 90，第272–280页（1959） 18. Phillips, S., Wilson, W.H.：范畴组合性：关于人类认知系统性的范畴论解释。公共科学图书馆计算生物学 6，e1000858（2010） 19. Rezende, D.J. 等：深度生成模型中的单样本泛化。载于：国际机器学习会议，第48卷（2016） 20. Rosenbaum, C., Klinger, T., Riemer, M.：路由网络：为多任务学习自适应选择非线性函数。载于：2018年国际学习表征会议论文集，第1–10页（2018） 21. Shiebler, D., Gavranovi´c, B., Wilson, P.：机器学习中的范畴论。载于：应用范畴论会议论文集（2021） 22. Spivak, D.I.：接线图的算子：为数据库、递归和即插即用电路形式化图形语言，arxiv: 1305.0297（2013） 23. Webb, S. 等：生成模型的忠实反演，实现有效的摊销推断。载于：2018年神经信息处理系统大会论文集，第3074–3084页（2018） 24. Zhao, S., Song, J., Ermon, S.：从生成模型中学习层次特征。载于：国际机器学习会议（2017）