arXiv:2305.03143v1 [cs.AI] 3 May 2023

迈向逻辑公式的可逆语义保持嵌入

Gaia Saveri[1,2] 和 Luca Bortolussi[2]

[1] 意大利比萨大学计算机科学系[2] 意大利的里雅斯特大学数学与地球科学系

摘要。 逻辑是进行自动推理的主要形式语言，同时也是一种人类可理解的语言，至少对于较小的公式而言如此。在人工智能领域，学习和优化逻辑需求与规则始终是一项重要课题。最先进的机器学习（ML）方法大多基于连续空间中的梯度下降优化，而逻辑学习则是在公式的离散句法空间中进行的。利用连续优化来学习逻辑属性是一项颇具挑战的任务，需要以有意义的方式将公式嵌入连续空间，也就是保留其语义。诸如 [BGKN22] 这样的方法能够通过核方法为线性时序逻辑构建有效的语义保持嵌入，但其定义的映射不可逆。在本研究中，我们针对这一问题展开探讨，借助基于图变分自编码器框架的深度架构，学习如何对这类嵌入进行逆映射。我们提出了一种专为这一场景设计的新模型，并通过全面的实验评估论证了设计选择的合理性。在命题逻辑背景下获得的结果令人鼓舞，同时我们也指出并应对了公式可逆嵌入学习面临的若干挑战。

1 引言

L 逻辑及其诸多变体，可以说是用于表示知识、为复杂系统规定需求并提供解释，以及以人类可理解的方式进行推理的最重要语言之一 [RN20]。另一方面，图神经网络（GNN， [SGT+09]）在关系学习方面已达到最先进水平，提供了置换不变性和稀疏性感知等有意义的归纳偏置 [WPC+21，BHB+18]。近年来，人们投入了大量注意力，将符号知识表示与图神经网络所执行的神经计算相结合，以解决组合推理和逻辑推理任务 [LdGG+20，BLP21]。在此背景下，利用图神经网络学习逻辑公式的实值表示，将为在公式空间中使用基于梯度的优化技术打开大门，从而将需求挖掘从离散搜索问题转变为连续搜索问题。此类模型的一个关键目标应是保持语义一致性，即语义相似的公式应映射到潜在空间中彼此接近的向量。当前该领域针对线性时序逻辑的最先进方法，基于

核方法，因此它学习的是从公式到其嵌入的不可逆函数 [BGKN22]。我们针对这一缺陷提出了一种基于图变分自编码器框架（GVAE， [KW16,GZ20a]）的模型，其目标是构建从逻辑公式的离散句法空间到连续空间的可逆映射，并在该空间中保留语义相似性。

我们的贡献 从 GVAE 模型用于有向无环图（DAG， [ZJC+19]）的理论依据出发，我们构建了一种模型，能够将逻辑公式的句法树编码为连续的判别性表示。为使这些嵌入保留语义，我们提出了一种有原则的方法，将语义信息融入学习到的潜在空间。我们展示了该方法在命题逻辑中的结果。尽管命题逻辑的语法十分简单，但其命题固有的对称性以及句法-语义交互仍会带来挑战；我们通过一系列有充分依据的设计选择来应对这些挑战。实验结果令人鼓舞，可视为这一方法迈向更丰富逻辑形式体系的概念验证。

相关工作 针对有向无环图的自编码器模型学习，最初通过序列模型进行研究 [TSM15, SZWW16, MLZ+16] ，随后在 [ZJC+19, TC21, LVD+18a] 中转化为图相关任务。该领域的大量研究致力于分子生成 [JBJ18,LABG18,KDGS23]。而在 [KPH17a, MCX18]，中，则分别探讨了带有句法约束或语义约束的生成模型学习。逻辑公式语义相似性度量的构建则见于 [BGKN22]。

2 背景

如第 1 节所述，本研究将探讨范围限定在命题逻辑。设 $\{x_i\}_{i=1}^n$ 为一组命题变量，其中每个 x_i 都是二元变量，其取值可以为真（1）或假（0）。命题公式集合（下文也称为命题或需求）通过语法 $\varphi := 1 \mid x_i \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2$ 递归定义。语义则按照布尔运算符的通常含义递归定义（详情见附录 A.1）。

在此背景下，表示命题的一种便捷方式是使用其抽象语法树（AST），即以树的形式表示公式的句法结构，其中内部节点为布尔运算符，叶节点为命题变量。由于布尔运算符的元数最多为二，命题公式对应的 AST 都是二叉树。这种表示方式为应用图神经网络（GNN， [SGT+09]）模型打开了大门；这类深度学习模型能够原生处理图结构输入。

在这项工作中，我们希望同时学习一个编码器和一个解码器，以实现输入$\mathbf{x}$与连续向量$\mathbf{z}$之间的相互映射。变分自编码器（VAE， [KW14]）框架旨在同时学习两个参数化的模型。

函数：概率生成网络（解码器） $p_\theta(\mathbf{x}|\mathbf{z})$ 和近似后验分布（编码器） $q_\phi(\mathbf{z}|\mathbf{x})$，通过最大化证据下界：

$$\mathcal{L}(\phi,\theta;\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - KL[q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})] \tag{1}$$

其中， $KL[q(\cdot)||p(\cdot)]$ 表示 $q(\cdot)$ 与 $p(\cdot)$ 之间的 Kullback-Leibler 散度，而 $p(\mathbf{z})$ 表示潜变量 $\mathbf{z}$ 的先验分布。

通过采用 GNN 作为编码器（也可能作为解码器）模型 [KW16]，VAE 模型已推广到图结构数据。图的深度生成模型被划分为两类（借用 [ZDW+22、GZ20b] 中的术语），划分依据是图的节点和边的生成方式：顺序生成，即按照预定义的顺序逐一处理节点和边；一次性生成，即一次性生成图的邻接矩阵。

如果需要将关于输入 $\mathbf{x}$ 的辅助信息 $\mathbf{y}$ （例如类别标签或语义上下文）纳入学习过程，则条件变分自编码器（CVAE， [SLY15,IFV19]）等框架可以通过对编码器（在此处定义为 $q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x})$）、先验分布（$p_\psi(\mathbf{z}|\mathbf{y})$，其为参数化分布）以及解码器（$p_\theta(\mathbf{x}|\mathbf{y},\mathbf{z})$）施加条件，来控制生成过程。网络通过优化以下条件边际对数似然进行训练（符号表示法与公式 1 相同）：

$$\mathcal{L}(\phi,\theta,\psi;\mathbf{x},\mathbf{y}) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{y},\mathbf{z})] - KL[q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x})||p_\psi(\mathbf{z}|\mathbf{y})] \tag{2}$$

对于欧几里得输入格式，通常会在计算编码（或解码）函数之前，将向量 $\mathbf{y}$ 与输入 $\mathbf{x}$ （或潜向量$\mathbf{z}$）拼接起来；而在图生成模型中，则有多种可能：如果图中所有节点都有附加信息，就将其与初始节点状态 [YZS+19] 拼接；当$\mathbf{y}$ 是整个图的属性时，可以在解码过程 [LVD+18b] 之前将其与潜向量 $\mathbf{z}$ 拼接， [LZL18]也可以将其融入图神经网络的解码架构本身。

2.1 基于核的逻辑嵌入

在 [BGKN22] 中，通过核技巧解决了学习保持语义相似性的逻辑公式嵌入问题。更具体地说，通过定义一个合适的核函数（以及与之对应的学习算法），将公式映射到连续希尔伯特空间 L^2 的子空间中，从而处理了寻找信号时序逻辑（STL）公式之间相似度度量的问题；对于语义相似的公式，该核函数产生较高的值（否则产生较低的值；更多细节请参阅附录 B）。由于布尔运算符是 STL 运算符的子集，我们可以将 [BGKN22] 中核的定义限制在命题逻辑的语境下。考虑至多包含 n 个变量的命题，令 $\mathcal{T}$ 表示 2^n 个可能的变量配置构成的集合，并记作：如果公式 φ 在 $\tau \in \mathcal{T}$ 上求值为真（分别为假），则记 $\varphi(\tau) = 1$（分别为 $\varphi(\tau) = -1$），那么

我们构造公式 φ 与 ψ 之间的 布尔 核（通过将 [BGKN22] 中布尔 STL核的定义加以限制），形式如下：

$$k(\varphi, \psi) = \int_{\tau \in \mathcal{T}} \varphi(\tau) \cdot \psi(\tau)\, d\tau \tag{3}$$

并通过蒙特卡洛近似计算，在配置空间 $\mathcal{T}$ 上采用均匀概率测度。实验确认，公式 3 的布尔核会将命题公式映射到一个保留语义的连续空间中，详见附录 B；我们还在附录 B 中展示了该核与命题公式之间的杰卡德相似系数的相关性。此外，使用主成分分析（PCA）降低嵌入空间的维度后发现，嵌入空间的维度远小于用于构造公式 3 的核的公式数量，也是合理的。有关降维的实验结果，请参阅附录 B。

3 通过图神经网络实现逻辑嵌入

本工作的主要目标是研究用于计算命题逻辑公式可逆嵌入的算法，从其语法出发，探索包含语义信息的方法，以学习表征该逻辑语义的潜在空间。由于每个公式都可以通过其抽象语法树表示，且不会损失任何信息（见第 2 节和附录 A），基于图神经网络框架的架构为我们的设置提供了合理的归纳偏置。此外，由于我们要求同时学习公式的编码器和解码器，因此我们认为采用图变分自编码器（GVAE）模型是有意义的。形式上，给定按照附录 C.1 中详述的算法采样的命题公式 φ，我们将模型的输入设计为由其抽象语法树产生的有根树，即输入是图$G = (V, E)$，其顶点集合 V对应于出现在 φ 中的运算符和变量（按其重数计），有向边集合 $E \subseteq V \times V$满足：若 v_j 是 φ中的 v_i 的主子公式，则 $(v_i, v_j) \in E$。尽管命题逻辑存在范式，我们决定主要出于两个原因而不采用其中任何一种：首先，将公式重写为范式通常会增加项的数量（即，用合取范式或析取范式表示的含有 n 个变量的每个公式最多可以有 2^n 项），当公式被用作图神经网络的输入时，这可能导致可扩展性问题；其次，在本工作中，我们使用命题逻辑作为概念验证，用于验证我们希望部署到时序逻辑中的模型，而时序逻辑通常不允许简单的范式。

通常，GNN 模型（尤其是基于消息传递范式 [GSR+17] 的模型）采用同步更新方案来组织计算，即所有节点同时更新其状态并交换消息。然而，在此情境下我们所考虑的图中的节点具有明确的依赖关系，因此，遵循 [ZJC+19] 所采用的方法，我们按顺序处理节点，即为节点建立拓扑序

输入图（由树的深度优先遍历给出），并采用异步方案执行计算：仅当某个节点的所有前驱节点的状态都已更新后，才更新该节点的状态。也可以将此顺序反转（即考虑边方向相反的输入有向无环图，并为反向图添加一个虚拟根节点，将其连接到原树的所有叶节点），并采用自底向上的方案交换消息，由此形成我们所称的双向架构。从某种意义上说，在考虑双向消息传递方案时，计算的执行方式类似于对输入公式本身求值的方式。

3.1 编码

编码器是负责将离散输入图 G映射为连续潜在表示 $\mathbf{z}$ 的网络。在本例中，它由一个采用异步更新方案（可能是双向的）的图神经网络（GNN）构成，具体如本节前文所述。在我们的模型中，编码器通过将图注意力网络（GAT， [VCC+18]）适配到 [ZJC+19] 的异步消息传递方案来构建，数学细节参见附录 A.2。

初始节点的隐藏状态是其类型的独热编码（请注意，增加允许的变量数量会增加节点类型的数量）。随后，消息阶段结束时，所有节点的状态都会计算完成，我们取没有后继的节点状态作为编码器 out_e 的输出（称为 终止节点；如果不存在这样的节点，则会向输入图中添加一个虚拟节点）。对于双向编码，我们会拼接两个计算方向上终止节点的状态。对于公式 1 和 2 的近似后验 $q(\cdot)$，我们采用高斯分布，其均值和方差通过将 out_e 输入两个多层感知机得到。

3.2 解码

给定一个实值潜向量 $\mathbf{z}$，解码器的目标是将其转换为一个离散图 G，用于表示公式 φ的抽象语法树。解码器的主要要求是保证有效性，即应生成语法正确的公式。为此，我们决定（消融实验见 D.1 节）设计一种在架构层面实现语法规则的解码器，也就是说，该解码器不允许生成无效公式。这一过程通过逐节点生成流程实现（源自 [ZJC+19]），每次构建一个节点 v_i，并按如下方式更新其状态 $\mathbf{h}_{v_i}$（记 $\mathcal{N}(v_i)$ 为 v_i 的前驱节点集合，并记 x_{v_i} 为其类型的独热编码）：

$$\begin{aligned} \mathbf{h}_{v_i} &= GRU(x_{v_i}, \mathbf{h}^{\text{agg}}_{v_i}) \\ \mathbf{h}^{\text{agg}}_{v_i} &= \sum_{v_j \in \mathcal{N}(v_i)} g(\mathbf{h}_{v_j}) \odot m(\mathbf{h}_{v_j}) \end{aligned} \tag{4}$$

其中， GRU 是门控循环单元 [CvMG+14] ， $\odot$ 表示映射网络 m 与门控网络 g 之间的门控求和，它们均由多层感知机实现。

因此，解码会得到以下自顶向下迭代算法：

1. 给定初始潜向量$\mathbf{z}$，将其输入具有 Softmax 激活的 MLP，即可生成初始图状态$\mathbf{h}_G = \mathbf{h}_0$。根据构造，第一个创建的节点 v_0 的类型为$start$，即合成起始节点；2. 每个节点 v_i（其中 $1 \leq i \leq \max_v$ （$\max_v$ 是一个输入参数，用于指定生成过程中允许的最大节点数））按如下方式生成：(a) 首先将当前图状态$\mathbf{h}_G$输入具有 Softmax 激活的 MLP，计算所有可能类型的分布，然后从该分布中采样，以推断 v_i 的类型；(b) 使用公式 4 更新 v_i 的隐藏状态；(c) 按如下方式更新图状态：$\mathbf{h}_G = \mathbf{h}_{v_i}$；3. 按照树的深度优先遍历顺序扩展节点，并考虑节点类型所表示运算符的元数，从而避免违反句法约束；4. 当树无法继续生长（即所有叶节点都是变量），或达到 $i = \max_v$ 时，图生成过程停止（后一种情况是输出无效公式的唯一可能情形）。

在第 2.1 节（附录 B 中有更详细的描述）中，我们介绍了一种有效的方法，用于计算命题公式的有限维语义嵌入。值得注意的是，为了构造公式φ的此类表示，无需显式知道 φ 本身，只需知道它在一组赋值 $\{\tau_i\}_{i=1}^n \subseteq \mathcal{T}$下的赋值即可。因此，采用我们模型的条件变体是合理的，即使用条件变分自编码器架构，将输入公式 φ 的核嵌入$\mathbf{y}_\varphi$ （经主成分分析降维后得到）与潜在表示$\mathbf{z}$ 拼接作为语义上下文向量，再开始解码计算。

3.3 学习

训练采用教师强制法，即在损失计算期间，将解码器的输入设为需要重构的真实图。在第3.2节解码算法的每次迭代中，我们通过向网络输入一个具有真实节点类型的顶点来累积负对数似然，从而迫使网络生成接近真实节点序列的结果。

4 实验

对我们模型的一系列变体进行的实验主要有两个目标：测量变分自编码器重构和生成能力，并定性评估潜在空间的平滑性。因此，我们沿用 [ZJC+19,KPH17b], , 进行以下测试：

– VAE模型的能力：我们进行标准实验，以检验重构准确率、先验有效性、唯一性和新颖性；– CVAEmodels的能力：除了与上一点相同的重构准确率实验外，我们还测试模型在潜在空间中保持语义相似性的能力；– 潜在空间可视化：通过对潜在空间进行可视化和插值，我们定性评估其捕捉特征性结构特征的能力。

4.1 实验设置

我们提出了以下模型版本（以下称为 LogicG-VAE）：

– 仅语法：给定输入公式 φ，对其抽象语法树进行编码，不添加任何额外信息，即只能获取φ的语法信息；– 语义条件：给定输入公式 φ，使用第 2.1 节的核计算其语义上下文向量，并按照第3.2节末尾所述执行条件解码；

对于这两种变体，我们使用一组不同的异步编码图神经网络进行实验，分别为 GRU、图卷积网络（GCN， [KW17]）和 GAT。所有模型的架构和训练细节见附录 D。

4.2 实验结果

变分自编码器和条件变分自编码器的能力 本系列实验旨在评估：(a) 准确率，即完美重建输入抽象语法树的能力；(b) 有效性，即生成语法有效公式的能力；(c) 唯一性，即有效生成结果中不同图所占的比例；(d) 新颖性，即生成图中不属于训练集的比例。为评估准确率，我们对每个测试图 G 进行编码，以获得后验近似 $q_\phi(\mathbf{z}|G)$的均值和方差，随后从该分布中采样 10 个潜在向量，并对每个向量解码 10 次。准确率定义为这 100 个图中与输入 G完全相同的图所占的百分比。有效性则通过从先验分布 $p(\mathbf{z})$中采样 1000 个潜在向量，并对每个向量解码 10 次，计算所得语法有效抽象语法树的比例。对于我们架构的条件变体， $q_\phi(\cdot)$ 和 $p_\psi(\cdot)$ 分别取公式 2 中的值。

在条件设置下，我们首先按照上述方法测试准确率，然后通过计算以下指标，测试学习到的潜在空间保留语义相似性的效果：(a) 与输入上下文向量的平均距离。也就是说，给定用于条件解码的语义向量$\mathbf{y}$ ，我们从先验分布$\mathbf{z} \sim p_\psi(\mathbf{z}|\mathbf{y})$中采样100个向量，并对每个向量解码10次；对每个$\mathbf{z}$ ，保留解码频率最高的公式，计算其语义嵌入，并与真实值进行比较

真实值$\mathbf{y}$，使用 L_2 范数；(b) 具有相同语义向量$\mathbf{y}$的公式之间的平均核相似度。也就是说，按照上一项所述的步骤，我们计算每个$\mathbf{y}$对应的解码频率最高的公式之间的核相似度。

表1：VAE 和条件变分自编码器能力测试的结果，百分比是对 300 个包含 5 个变量的测试公式取平均所得。GAT 和 GCN 模型准确率列的第二行表示仅考虑每个数据点解码频率最高的公式时计算出的准确率。

(a) VAE 能力测试结果。

编码器	准确率。	Val.	独特	新颖性。
GAT	**93.92**	89.38	**56.53**	**55.34**
	95.42			
GCN	91.27	**98.78**	29.91	28.32
	93.56			
GRU	82.24	75.52	10.63	19.16

(b) 条件变分自编码器能力测试结果。

编码器	Acc.	Val.	Sem. 距离	Ker. 值
GAT	**87.43**	**93.72**	**6.317**	**0.7985**
	92.45			
GCN	85.35	89.14	6.808	0.6924
	91.79			

表1a报告了检验VAE对不同编码器能力的结果（附录D中对这些编码器进行了完整描述）。首先需要注意的是，卷积式图神经网络（即GCN或GAT）在所有性能指标上都优于递归式网络（GRU）。这可能是因为，与GRU架构不同，卷积能够利用更多的局部子结构模式，并利用正在更新的节点的隐藏状态，而不仅仅是其邻居节点的隐藏状态。GCN和GAT的重构准确率都很高且相当；然而，使用GAT进行编码时，我们观察到潜在空间捕捉数据结构特征的能力更强，这一点可由其更高比例的独特且新颖的解码公式得到证明。在需求优化场景中，可以利用这些结果，将给定公式映射为其潜在表示，并在连续嵌入空间中优化其结构。

表1b报告了条件变分自编码器能力测试的结果（根据表1a，仅使用最有前景的编码网络）。如果我们取一组包含5000个随机布尔公式的样本，每个公式有5个变量，并用它们为每个测试实例生成语义上下文向量，那么这些公式的语义嵌入之间的平均距离为18.7325，平均核相似度为0.1695。由于表1b报告的语义距离和核值要低得多（分别要高得多），我们观察到，以语义向量为条件后，输入数据实际上被映射到了一个在一定程度上保留语义相似性的空间中。

值得注意的是，尽管条件架构（表1b）的重构准确率低于无条件架构（表1a），但前者可应用于需求挖掘场景：在该场景中，只能观测到一组配置中某个公式的赋值（从而可以构造语义上下文向量），而且还必须找到（即解码出）一个能够优化适应度函数的显式公式，从而允许

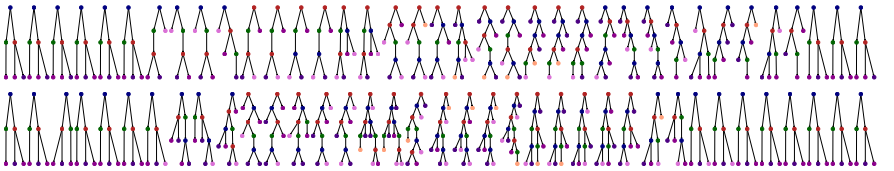


图1：在采用 GAT（上）和 GCN（下）编码器的 LogicVAE 模型中，围绕一个随机公式进行 Slerp 插值。节点的颜色表示其类型。

构造语义上下文向量，并找到（即解码出）一个能够优化适应度函数的显式公式。从本质上说，在这种情况下，我们需要将条件变分自编码器用作生成模型，对公式进行采样，使其（近似地）具有由给定语义上下文向量所指定的语义；该向量则通过在语义嵌入空间中求解优化问题获得。在仅利用句法信息的普通变分自编码器场景中，甚至无法定义这一任务。此外，如表1b的准确率列第二行所示，如果只考虑每个测试点最常被解码出的公式，重构准确率会显著提高。

更多实验结果和消融研究见附录D。我们特别强调解码器架构中句法约束的重要性（第3.2节）。

潜在空间可视化 我们使用球面插值（$slerp$，[Whi16]）围绕给定的潜在点进行插值，以定性评估潜在空间的平滑性。简而言之，这一方法是在与潜在空间维度相同的空间中，沿超球面上的大圆对点进行插值。从一点 $\mathbf{z}_\varphi$ 出发对公式 φ 进行编码后，我们沿着从 $\mathbf{z}_\varphi$出发的大圆，选取35个间隔均匀的点进行解码。从图1可以看出，使用 GAT 编码器的模型生成的潜在空间更加平滑，因为在插值的每一步中，它改变的节点更少。

5 结论与未来工作

在本研究中，我们提出了 LogicVAE，这是一种可配置的 GVAE 模型，旨在学习命题逻辑公式的数据驱动表示。通过在编码器中采用异步消息传递计算方案，并在解码器中引入特定的句法约束，该模型能够学习具有表达力的潜在空间。我们提出了一种将语义信息融入学习过程的新方法，旨在构建一个保留语义的嵌入空间。该方法的主要瓶颈在于难以扩展到包含超过 5 个变量的数据集；一种可能的解决方案是采用分层方法，附录D.2对此进行了概述。

如前所述，本文报告的结果应视为概念验证，旨在将模型扩展到更复杂的形式体系，

例如时序逻辑。这类形式体系面临额外挑战：需要学习与运算符（或变量）节点相关的时间参数和/或阈值参数，但通常只处理少量信号（一般不超过 5 [EAB+20]）。此外，受 AlphaCode [LCC+22] 的启发——它为程序定义了一种生成模型（这是句法—语义交互具有重要作用的另一种场景）——我们计划在编码器中加入自监督预训练阶段（调整 AlphaCode 中使用的掩码语言建模损失），并检验使用大型语言 Transformer 模型 [VSP+17] 对逻辑公式进行编码和解码的可行性。

参考文献

ASY+19. Takuya Akiba、Shotaro Sano、Toshihiko Yanase、Takeru Ohta 和 Masanori Koyama。Optuna：新一代超参数优化框架。载于第*25*届 *ACM SIGKDD*知识发现与数据挖掘国际会议论文集，第2623–2631页。ACM，2019年。BGKN22. Luca Bortolussi、Giuseppe Maria Gallo、Jan Kretínský 和 Laura Nenzi。随机过程信号时序逻辑的模型检验学习与核技巧。载于系统构造与分析工具和算法——第*28*届国际会议（*TACAS 2022*），作为欧洲软件理论与实践联合会议（*ETAPS 2022*）的一部分举办，德国慕尼黑，*2022*年*4*月*2*日至*7*日，论文集，第一部分，《计算机科学讲义》第13243卷，第281–300页。施普林格，2022年。BHB+18. Peter W. Battaglia、Jessica B. Hamrick、Victor Bapst、Alvaro Sanchez-Gonzalez、Vinícius Flores Zambaldi、Mateusz Malinowski、Andrea Tacchetti、David Raposo、Adam Santoro、Ryan Faulkner、Çaglar Gülçehre、H. Francis Song、Andrew J. Ballard、Justin Gil mer、George E. Dahl、Ashish Vaswani、Kelsey R. Allen、Charles Nash、Victoria Langston、Chris Dyer、Nicolas Heess、Daan Wierstra、Pushmeet Kohli、Matthew M. Botvinick、Oriol Vinyals、Yujia Li 和 Razvan Pascanu。关系归纳偏置、深度学习与图网络。*CoRR*，abs/1806.01261，2018年。BLP21. Yoshua Bengio、Andrea Lodi 和 Antoine Prouvost。组合优化的机器学习：方法论概览。《欧洲运筹学杂志》，290(2):405–421，2021年。CvMG+14. Kyunghyun Cho、Bart van Merrienboer、Çaglar Gülçehre、Dzmitry Bahdanau、Fethi Bougares、Holger Schwenk 和 Yoshua Bengio。使用 RNN 编码器—解码器学习短语表示，以用于统计机器翻译。载于*2014*年自然语言处理实证方法会议论文集（*EMNLP 2014*），*2014*年*10*月*25*日至*29*日，卡塔尔多哈，*ACL* 特别兴趣小组 *SIGDAT* 会议，第1724–1734页。ACL，2014年。EAB+20. Gidon Ernst、Paolo Arcaini、Ismail Bennani、Alexandre Donze、Georgios Fainekos、Goran Frehse、Logan Mathesen、Claudio Menghi、Giulia Pedrielli、Marc Pouzet、Shakiba Yaghoubi、Yoriyuki Yamagata 和 Zhenya Zhang。ARCH-comp 2020类别报告：反例生成。载于*ARCH20*：第*7*届连续与混合系统应用验证国际研讨会（*ARCH20*），《EPiC计算系列》第74卷，第140–152页，2020年。

GSR+17. Justin Gilmer、Samuel S. Schoenholz、Patrick F. Riley、Oriol Vinyals 和 George E. Dahl。用于量子化学的神经消息传递。载于第*34*届国际机器学习会议（*ICML 2017*），澳大利亚新南威尔士州悉尼，*2017*年*8*月*6*日至*11*日，《机器学习研究论文集》第70卷，第1263–1272页。PMLR，2017年。GZ20a. Xiaojie Guo 和 Liang Zhao。用于图生成的深度生成模型系统综述。*CoRR*，abs/2007.06686，2020年。GZ20b. Xiaojie Guo 和 Liang Zhao。用于图生成的深度生成模型系统综述。*CoRR*，abs/2007.06686，2020年。HMP+17. Irina Higgins、Loïc Matthey、Arka Pal、Christopher P. Burgess、Xavier Glorot、Matthew M. Botvinick、Shakir Mohamed 和 Alexander Lerchner。β-变分自编码器：通过受约束变分框架学习基本视觉概念。载于第*5*届国际学习表征会议（*ICLR 2017*），法国土伦，*2017*年*4*月*24*日至*26*日，会议分会论文集，2017年。IFV19. Oleg Ivanov、Michael Figurnov 和 Dmitry P. Vetrov。支持任意条件化的变分自编码器。载于第*7*届国际学习表征会议（*ICLR 2019*），*2019*年*9*日，2019年*6* 美国路易斯安那州新奥尔良，*5*月—。JBJ18. Wengong Jin、Regina Barzilay 和 Tommi S. Jaakkola。用于分子图生成的连接树变分自编码器。载于第*35*届国际机器学习会议（*ICML 2018*），瑞典斯德哥尔摩斯德哥尔摩国际展览中心，*2018*年*7*月*10*日至*15*日，《机器学习研究论文集》第80卷，第2328–2337页。PMLR，2018年。KB15. Diederik P. Kingma 和 Jimmy Ba。Adam：一种随机优化方法。载于第*3*届国际学习表征会议（*ICLR 2015*），美国加利福尼亚州圣迭戈，*2015*年*5*月*7*日至*9*日，会议分会论文集，2015年。KDGS23. Vladimir Kondratyev、Marian Dryzhakov、Timur Gimadiev 和 Dmitriy Slutskiy。基于连接树变分自编码器的生成模型：HOMO值预测与分子优化。《化学信息学杂志》，15(1):11，2023年。KPH17a. Matt J. Kusner、Brooks Paige 和 José Miguel Hernández-Lobato。语法变分自编码器。载于第*34*届国际机器学习会议（*ICML 2017*），澳大利亚新南威尔士州悉尼，*2017*年*8*月*6*日至*11*日，《机器学习研究论文集》第70卷，第1945–1954页。PMLR，2017年。KPH17b. Matt J. Kusner、Brooks Paige 和 José Miguel Hernández-Lobato。语法变分自编码器。载于第*34*届国际机器学习会议（*ICML 2017*），澳大利亚新南威尔士州悉尼，*2017*年*8*月*6*日至*11*日，《机器学习研究论文集》第70卷，第1945–1954页，2017年。KW14. Diederik P. Kingma 和 Max Welling。自编码变分贝叶斯。载于第*2*届国际学习表征会议（*ICLR 2014*），加拿大艾伯塔省班夫，*2014*年*4*月*14*日至*16*日，会议分会论文集，2014年。KW16. Thomas N. Kipf 和 Max Welling。变分图自编码器。*CoRR*，abs/1611.07308，2016年。KW17. Thomas N. Kipf 和 Max Welling。基于图卷积网络的半监督分类。载于第*5*届国际学习表征会议（*ICLR 2017*），法国土伦，*2017*年*4*月*24*日至*26*日，会议分会论文集，2017年。

LABG18. Qi Liu、Miltiadis Allamanis、Marc Brockschmidt 和 Alexander L. Gaunt。用于分子设计的受约束图变分自编码器。载于神经信息处理系统进展*31*：*2018* 年神经信息处理系统年会，*NeurIPS 2018*，*2018*年*12*月*3*日至*8*日，蒙特利尔，加拿大，第7806–7815页，2018年。LCC+22。Yujia Li、David Choi、Junyoung Chung、Nate Kushman、Julian Schrittwieser、Rémi Leblond、Tom Eccles、James Keeling、Felix Gimeno、Agustin Dal Lago、Thomas Hubert、Peter Choy、Cyprien de Masson d'Autume、Igor Babuschkin、Xinyun Chen、Po-Sen Huang、Johannes Welbl、Sven Gowal、Alexey Cherepanov、James Molloy、Daniel J. Mankowitz、Esme Sutherland Robson、Pushmeet Kohli、Nando de Freitas、Koray Kavukcuoglu 和 Oriol Vinyals。利用 AlphaCode 实现竞赛级代码生成。《科学》，378(6624):1092–1097，2022年。LdGG+20。Luís C. Lamb、Artur S. d'Avila Garcez、Marco Gori、Marcelo O. R. Prates、Pedro H. C. Avelar 和 Moshe Y. Vardi。图神经网络与神经符号计算的结合：综述与展望。载于第二十九届人工智能国际联合会议论文集，*IJCAI 2020*，第4877–4884页，2020年。LVD+18a。Yujia Li、Oriol Vinyals、Chris Dyer、Razvan Pascanu 和 Peter W. Battaglia。学习图的深度生成模型。*CoRR*，abs/1803.03324，2018年。LVD+18b。Yujia Li、Oriol Vinyals、Chris Dyer、Razvan Pascanu 和 Peter W. Battaglia。学习图的深度生成模型。*CoRR*，abs/1803.03324，2018年。LZL18。Yibo Li、Liangren Zhang 和 Zhenming Liu。利用条件图生成模型进行多目标从头药物设计。《化学信息学杂志》，10(1):33，2018年7月。MCX18。Tengfei Ma、Jie Chen 和 Cao Xiao。通过正则化变分自编码器生成语义有效图。载于神经信息处理系统进展*31*：*2018*年神经信息处理系统年会，*NeurIPS 2018*，*2018*年*12*月*3*日至*8*日，蒙特利尔，加拿大，第7113–7124页，2018年。MLZ+16。Lili Mou、Ge Li、Lu Zhang、Tao Wang 和 Zhi Jin。面向编程语言处理的树结构卷积神经网络。载于第三十届 *AAAI* 人工智能会议论文集，*2016*年*2*月*12*日至*17*日，菲尼克斯，亚利桑那州，美国，第1287–1293页。AAAI出版社，2016年。MN04。Oded Maler 和 Dejan Nickovic。连续信号时序性质的监测。载于形式技术、定时系统建模与分析及容错系统：定时系统形式建模与分析国际联合会议 *FORMATS 2004*，以及实时与容错系统形式技术会议 *FTRTFT 2004*，法国格勒诺布尔，*2004*年*9*月*22*日至*24*日，论文集，《计算机科学讲义》第3253卷，第152–166页。施普林格，2004年。PGC+17。Adam Paszke、Sam Gross、Soumith Chintala、Gregory Chanan、Edward Yang、Zachary DeVito、Zeming Lin、Alban Desmaison、Luca Antiga 和 Adam Lerer。PyTorch 中的自动微分。载于*NIPS 2017* 自动微分研讨会，2017年。RN20。Stuart Russell 和 Peter Norvig。《人工智能：一种现代方法》（第*4*版）。培生，2020年。

SGT+09。Franco Scarselli、Marco Gori、Ah Chung Tsoi、Markus Hagenbuchner 和 Gabriele Monfardini。图神经网络模型。*IEEE* 神经网络汇刊，20(1):61–80，2009年。SLY15。Kihyuk Sohn、Honglak Lee 和 Xinchen Yan。利用深度条件生成模型学习结构化输出表示。载于神经信息处理系统进展*28*：*2015*年神经信息处理系统年会，*2015*年*12*月*7*日至*12*日，加拿大魁北克省蒙特利尔，第3483–3491页，2015年。SZWW16。Bing Shuai、Zhen Zuo、Bing Wang 和 Gang Wang。用于场景标注的有向无环图循环神经网络。载于*2016*年 *IEEE* 计算机视觉与模式识别会议，*CVPR 2016*，*2016*年*6*月*27*日至*30*日，*27*3620 3629 2016美国内华达州拉斯维加斯第 – 页。IEEE计算机学会，。TC21。Veronika Thost 和 Jie Chen。有向无环图神经网络。载于第九届国际学习表征会议，*ICLR 2021*，线上活动，奥地利，*2021*年*5*月*3*日至*7*日，2021年。TSM15。Kai Sheng Tai、Richard Socher 和 Christopher D. Manning。通过树结构长短期记忆网络改进语义表示。载于计算语言学协会第五十三届年会暨亚洲自然语言处理联合会第七届自然语言处理国际联合会议论文集，*ACL 2015*，*2015*年*7*月*26*日至*31*日，中国北京，第*1*卷：长篇论文，第1556–1566页。计算语言学协会，2015年。VCC+18。Petar Velickovic、Guillem Cucurull、Arantxa Casanova、Adriana Romero、Pietro Liò 和 Yoshua Bengio。图注意力网络。载于第六届国际学习表征会议，*ICLR 2018*，加拿大不列颠哥伦比亚省温哥华，*2018*年*4*月*30*日至*5*月*3*日，会议分会论文集，2018年。VSP+17。Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N. Gomez、Lukasz Kaiser 和 Illia Polosukhin。注意力就是你所需要的一切。载于神经信息处理系统进展*30*：*2017*年神经信息处理系统年会，*2017*年*12*月*4*日至*9*日5998 6008，2017年美国加利福尼亚州长滩，第 – 页。Whi16。Tom White。生成网络采样：几种有效技术札记。*CoRR*，abs/1609.04468，2016年。WPC+21。Zonghan Wu、Shirui Pan、Fengwen Chen、Guodong Long、Chengqi Zhang 和 Philip S. Yu。图神经网络综述。*IEEE* 神经网络与学习系统汇刊，32(1):4–24，2021年。YZS+19。Carl Yang、Peiye Zhuang、Wenhan Shi、Alan Luu 和 Pan Li。通过图变分生成对抗网络进行条件结构生成。载于神经信息处理系统进展*32*：*2019*年神经信息处理系统年会，*NeurIPS 2019*，*2019*年*12*月*8*日至*14*日，加拿大不列颠哥伦比亚省温哥华，第1338–1349页，2019年。ZDW+22。Yanqiao Zhu、Yuanqi Du、Yinkai Wang、Yichen Xu、Jieyu Zhang、Qiang Liu 和 Shu Wu。深度图生成综述：方法与应用。载于图学习会议，*LoG 2022*，*2022*年*12*月*9*日至*12*日，线上活动，《机器学习研究论文集》第198卷，第47页。PMLR，2022年。ZJC+19。Muhan Zhang、Shali Jiang、Zhicheng Cui、Roman Garnett 和 Yixin Chen。D-VAE：一种用于有向无环图的变分自编码器。载于神经信息处理系统进展*32*：*2019*年神经信息处理系统年会，*NeurIPS 2019*，*2019*年*12*月*8*日至*14*日，加拿大不列颠哥伦比亚省温哥华，第1586–1598页，2019年。

附录

A 扩展背景

A.1 命题逻辑

命题逻辑是一种形式语言，其语法通过组合变量 $\{x_i\}_{i=1}^n$ 和逻辑运算符 $\neg$（否定）、 $\wedge$ （合取）和 $\vee$ （析取），递归地定义良构公式。更准确地说，变量是有效公式（也称为原子句）；给定有效句子 α 和 β，则 $\neg\alpha$、 $\alpha \wedge \beta$ 和 $\alpha \vee \beta$ 也都是有效句子。

命题逻辑的语义从为每个命题变量 x_i 赋予真值的赋值 τ 开始，然后递归地将以下规则应用于每个子公式 α （ α 在 τ 上的赋值记作$\alpha(\tau)$）： $\neg\alpha(\tau) = 1$ 当且仅当 $\alpha(\tau) = -1$, $(\alpha \wedge \beta)(\tau) = 1$ 当且仅当 $\alpha(\tau) = 1$ 且 $\beta(\tau) = 1$, $(\alpha \vee \beta)(\tau) = 1$ 当且仅当 $\alpha(\tau) = 1$ 或 $\beta(\tau) = 1$。

如第 2 节所述，每个命题公式 φ 都可以用其抽象语法树（AST）表示，如图 2 所示。

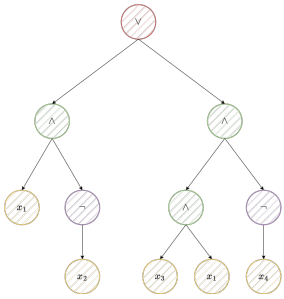


图 2：公式 $(x1 \wedge \neg x_2) \vee ((x_3 \wedge x_1) \wedge \neg x_4)$ 的抽象语法树。节点的颜色编码了其类型。

A.2 图神经网络

图神经网络（GNN， [SGT+09]）是运行于图领域的深度学习架构，即它们旨在以端到端的方式解决图相关任务。

可以说，GNN 最常用的框架是所谓的消息传递神经网络框架（MPNN， [GSR+17]），它抽象了现有 GNN 模型中的许多共性。顾名思义，MPNN 的定义性特征是采用一种神经消息传递形式，在固定次数的迭代中，节点之间（尤其是一跳邻居顶点之间）交换实值向量消息。

具体而言，在 MPNN 的每次消息传递迭代 t 中，每个节点 v 的隐藏表示 $\mathbf{h}_v^{(t)}$都会根据从其邻域 $\mathcal{N}(v)$ 聚合的信息 $\mathbf{m}_v^{(t)}$进行更新，如下所示：

$$\mathbf{h}_v^{(t+1)} = U_t(\mathbf{h}_v^{(t)}, \sum_{w \in \mathcal{N}(v)} M_t(\mathbf{h}_v^{(t)}, \mathbf{h}_w^{(t)})) \tag{5}$$

其中， M_t 称为消息函数， U_t 称为消息更新函数：二者都是任意可微函数（通常通过神经网络实现）。

运行此消息传递过程 T 次迭代后，执行读出阶段以计算每个节点的最终嵌入 $\mathbf{y}_v$ v ，计算方式如下：

$$\mathbf{y}_v = R(\{\mathbf{h}_v^{(T)} | v \in G\}) \tag{6}$$

其中， R （读出函数）同样需要是一个可微函数，且对节点状态的置换保持不变。

因此，MPNN 的核心直觉是：在每次迭代中，每个节点都会从其直接的图邻居那里聚合信息。随着迭代次数的增加，每个节点的嵌入所包含的信息将逐渐扩展到图中更远的区域。

公式 5 的消息函数和更新函数的一种实例化方式是图卷积网络（GCN， [KW17]）模型。该模型将卷积操作推广到图结构数据，通过更新每个节点 v 的表示$\mathbf{h}_v$ 来实现：

$$\mathbf{h}_v^{(t+1)} = \sigma\left(\sum_{w \in \mathcal{N}(v)} \frac{1}{c_{vw}} h_w^{(t)} W^{(t+1)} \right) \tag{7}$$

其中 σ非线性激活函数， $c_{vw} = \{D^{\frac{1}{2}}(A+I)D^{\frac{1}{2}}\}_{vw}$分别为 A和D 输入图的邻接矩阵与节点度矩阵，而 I为单位矩阵。

公式 5 的另一种可能计算方案是图注意力网络（GAT， [VCC+18]），它使用自注意力来学习相邻节点之间的相对权重。它包括：

$$\begin{aligned} \mathbf{h}_v^{(t+1)} &= \sigma(\sum_{w \in \mathcal{N}(v) \cup v} \alpha_{vw}^{(t+1)} W^{(t+1)} \mathbf{h}_w^{(t)}) \\ \alpha_{vw}^{(t+1)} &= \mathrm{softmax}(\mathrm{LeakyReLU}(\mathbf{a}^T [W^{(t+1)} \mathbf{h}_v^{(t)} || W^{(t+1)} \mathbf{h}_w^{(t)}])) \end{aligned} \tag{8}$$

包括 σ 非线性激活函数、一个 可学习参数向量，以及 $||$ 逐节点操作（如拼接或求和）。GAT 还可以使用多头注意力机制来增强模型能力，方法是独立地多次复制公式 8 的操作并聚合结果。

在其原始版本中，消息传递神经网络模型预期会在节点之间同步交换消息，但是当输入图为有向图时，它会

按照 [ZJC+19]中的观察，遵循节点之间的偏序是合理的。在这种异步消息传递方案中，公式 5 的每次迭代 t 都按照为输入图中的节点建立的拓扑序执行：在此方案中，节点 v 必须等待其所有前驱节点完成更新后，才能计算新的隐藏状态$\mathbf{h}_v^{(t)}$，也就是说，它会在更新自身状态之前使用其直接邻居的当前层表示$\mathbf{h}_w^{(t)}\}_{w\in\mathcal{N}(w)}$，而不是上一层的表示。

B 逻辑公式的核技巧

信号时序逻辑的核函数 我们在为逻辑公式定义语义相似性度量时所采用的核函数，定义于 [BGKN22]。它最初是为信号时序逻辑（STL [MN04]）开发的；STL 是一种线性时间时序逻辑，适用于表达实值轨迹上的性质。在此语境中，轨迹是一个函数 $\xi: I \to D$，其中 $I \subseteq \mathbb{R}_{\geq 0}$ 是时间域，$D \subseteq \mathbb{R}^n, n \in \mathbb{N}$ 是状态空间（我们也用 $\mathcal{T}$表示轨迹空间）。STL 的一个关键特征是，它既可以采用定性（或布尔）满足，也可以采用定量满足（后者称为鲁棒性）。具体而言，给定一个 STL 公式 φ、一条轨迹 ξ 和一个时间点 t，对于定性满足，如果 ξ 在时间 t满足（分别为不满足） φ，我们记为 $s(\varphi, \xi, t) = 1$（分别记为 $s(\varphi, \xi, t) = -1$）；对于定量满足，我们用$\rho(\varphi, \xi, t) \in \mathbb{R}$ 表示 φ 对于轨迹 ξ的鲁棒性，即 φ 在信号受到扰动时仍能满足的程度。成立的是：$\rho(\varphi, \xi, t) > 0 \to s(\varphi, \xi, t) = 1$，且 $\rho(\varphi, \xi, t) < 0 \to s(\varphi, \xi, t) = -1$（完备性）。

定量满足的定义允许将 STL 谓词φ 视为将轨迹映射到其鲁棒性的泛函，即 $\rho(\varphi, \cdot): \mathcal{T} \to \mathbb{R}$，因此公式可以嵌入到一个（可能是无限维的）希尔伯特空间中。在这个空间里，我们可以将以下内容视为成对公式φ，ψ 之间的“内积”：

$$k(\varphi, \psi) = \int_{\xi \in \mathcal{T}} \rho(\varphi, \xi) \cdot \rho(\psi, \xi)\, d\mu_0 \xi \tag{9}$$

其中 μ_0 是定义在轨迹上的概率测度。可以证明，方程 9 中的 $k(\cdot, \cdot)$ 是一个适当的核函数，并且可以给出概率近似正确（PAC）界，为使用该核在公式空间中进行学习时产生的误差提供概率界。在 [BGKN22] 中开展的实验证实，方程 9 中的核能够有效捕捉 STL 需求之间的语义相似性。

命题逻辑的核函数 基于上述 STL 核的思路，我们按照第 2.1 节中的详细说明，定义命题公式之间的相似性度量。

杰卡德指数（或相似系数）用于衡量两个集合 A 和 B 之间的相似度，其定义为 $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$；给定两个命题 φ, ϕ 和 the

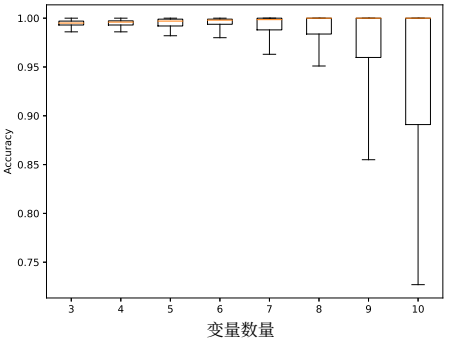


图3：核分类准确率随训练和测试公式中变量数量的变化情况，分位数为 100 次实验的平均值。

其赋值估值集合 X_φ, X_ψ 分别基于变量配置 $\mathcal{T}$，杰卡德指数 $J(\varphi,\psi) = \frac{|X_\varphi \cap X_\psi|}{|X_\varphi \cup X_\psi|}$ 表示两个公式一致的配置所占的比例，因此衡量了其输入之间的语义相似性。另一方面，$k(\varphi,\psi)$ 的值表示配置集合 $\tau \in \mathcal{T}$ 中 φ 与 ψ 一致和不一致部分之间的归一化差值；当两个公式在语义上等价时，该值为 1。该量与杰卡德指数呈正相关，因为 $k(\varphi,\psi) = 2J(\varphi,\psi) - 1$，这进一步证实了公式 3 可以用作衡量公式之间语义相似性的指标。综合以上各点可知，核是衡量公式之间相似性的可解释指标。例如，核值为 0.8 可以理解为两个公式在 90% 的配置上保持一致。

此外，我们还可以通过展示图 3 中报告的、基于公式的赋值满足性进行核分类的结果，更深入地了解布尔核在衡量命题公式语义相似性方面的性能。

由于核计算得到的嵌入的维度等于用于评估核本身的训练公式数量（即，根据公式 3，对输入公式与训练集中的所有公式逐一计算核评估值后得到的向量），我们通过在嵌入空间中使用核主成分分析（核 PCA）进行降维，检验能否在不造成显著信息损失的情况下减少维度。我们通过展示以下结果，对这个问题作出肯定回答：对于至多包含 5 个变量的公式，保留 100 个主成分；对于包含 6 至 10 个变量的公式，保留 500 个主成分，便能保留由 5000 个公式组成的原始数据集中的大部分方差，如图 4 所示。

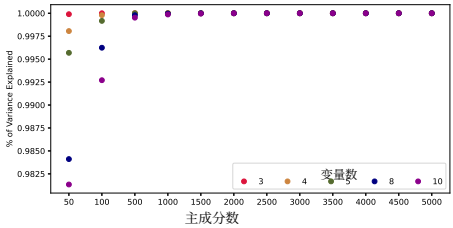


图4：对于公式中出现的不同变量数，保留的主成分数与解释方差比例之间的关系（使用5000个公式计算核）。

C 更多实验细节

C.1 数据集生成算法

为构建命题公式的合成数据集，我们将 [BGKN22] 中定义的算法调整到我们的框架中。由此得到算法 1 中描述的、用于递归生成每个公式语法树的方案，其中 p_{leaf} 表示节点成为叶节点的概率， n 表示允许的变量索引最大数量，而 类别$p_{\text{leaf}}(\{\wedge, \vee, \neg, \text{VAR}\})$ 表示类别概率分布：它为结果 VAR（即语法树的叶节点）分配 p_{leaf} 概率，并为每个运算符节点（即语法树的内部节点）分配 $\frac{1-p_{\text{leaf}}}{3}$。

算法1 语法树随机递归增长方案

要求： $p_{\text{leaf}} \in (0,1)$， $n \in \mathbb{N}$
  $q \leftarrow \emptyset$ ▷ 未完成节点队列
  根节点 $\leftarrow \mathcal{U}(\{\wedge, \vee, \neg\})$ ▷ 根节点是一个运算符节点
  q.推入(根节点)
  当 $q \neq \emptyset$ **执行**
    $N \leftarrow q$.弹出() ▷ 下一个要展开的节点
    如果 $N \in \{\wedge, \vee\}$ **那么**
      M， $R \leftarrow$ 猫$_{p_{\text{leaf}}}(\{\wedge, \vee, \neg,$变量$\})$ $\times$ 猫$_{p_{\text{leaf}}}(\{\wedge, \vee, \neg,$变量$\})$
      N.左 _子节点 $\leftarrow M$
      N.右 _子节点 $\leftarrow R$
      q.推入(M)
      q.推入(R)
    否则如果 N 是 $\neg$**那么**
      $M \leftarrow$猫$_{p_{\text{leaf}}}(\{\wedge, \vee, \neg,$ 变量$\})$
      N.子节点 $\leftarrow M$
    else ▷ N 是一个变量节点
      N.索引 $\leftarrow \mathcal{U}(n)$ ▷ 变量索引是随机采样的

C.2 更多训练细节

第 4 节中描述的所有模型均使用 Python 实现，并借助 PyTorch [PGC+17] 库来利用 GPU 加速。

然后，我们通过小批量梯度下降优化的损失函数，是架构条件版本中的公式 1，或其公式 2 变体。我们允许使用可调超参数 $\beta > 0$ 对此类损失的 KL 项加权，即采用所谓的 β-变分自编码器 [HMP+17] 模型的损失。

每种架构都使用随机梯度下降（SGD）在一组 4000 个不同的命题公式上进行训练（如 C.1 节所述生成），这些公式被划分为大小为 32 的小批量，初始学习率为10^{-3}，使用Adam优化器 [KB15]，并采用早停来调节训练轮数（我们每 30 个训练轮数进行一次验证，如果验证损失在 3 个检查点的耐心值内没有显著改善，则停止训练；最终，非条件模型训练了 ~ 300 个训练轮数，条件模型训练了 ~ 600 个训练轮数）。

训练和测试模型所用的数据集根据算法1生成，固定 $p_{\text{leaf}} = 0.4$ 并改变 $n \in \{3, 4, 5\}$。由此得到的数据集平均包含 $[10.2955, 13.6632, 17.7475]$ 个节点，平均深度为$[5.4687, 8.5132, 9.4419]$，且随着 n增加而变化。值得注意的是，允许的变量索引数量增加，会使网络需要学习的节点类型数量增加（从而进一步提高数据集的复杂度）。

我们测试了三种不同的编码器架构，每种架构都由一个图神经网络组成，并具有以下显著特征：

– 门控循环单元，与 [ZJC+19]相同：编码器是一个消息传递神经网络，使用公式 5的记法时，M_t 是一个门控循环单元，U_t 是门控求和（即采用与公式 4中的解码器相同的更新方案）；– 图卷积网络：编码器是一个具有 2 层的图卷积网络（公式 7），除非另有说明；– 图注意力网络：编码器是一个具有 3 层的图注意力网络，分别有 (3,3,4) 个注意力头，使用求和作为公式 8中的节点级运算；在内部层中，使用拼接作为单个注意力头结果的聚合器，在最终层中则使用平均值，除非另有说明。

它们全部实现了异步消息传递方案。对于所有编码器，我们报告了双向设置下的结果，这一点由附录 D.1 中的消融研究所支持。为了计算近似潜在分布 $q_\phi(\cdot)$ 的均值和协方差，我们使用了两个 1 层多层感知机。

模型的隐藏层维度设为 250，潜在维度设为 56；这些超参数以及上文列出的超参数均使用超参数优化框架 Optuna [ASY+19]进行了调优。如 [ZJC+19] 中所述，我们还发现，将 $\beta > 0.001$ 设为损失函数中 KL 散度的权重会导致重构准确率显著下降。

关于我们模型 CVAE 版本中使用的语义上下文向量，我们按照附录 B 中的描述，使用核主成分分析计算了该向量。由于我们的

训练集和测试集包含的公式最多有 5 个变量，因此我们保留 100 个主成分，理由见图 4。

D 更多实验结果

在表2和表3中，我们报告了在复杂度不断增加的不同数据集上，更多关于变分自编码器和条件变分自编码器能力的结果。我们注意到，图注意力网络是表现最佳的编码器，无论是在句法任务还是语义条件任务中。特别是在条件情况下，它能够更好地捕捉每个$\mathbf{y}$相对于公式 φ 的多模态性，并更好地刻画潜在空间中的语义。

关于后者，图5揭示了公式 (φ, ψ) 之间的核相似度值与其核主成分分析嵌入之间距离的关系。由于两个公式之间的核值是衡量其语义相似性的可解释指标（详见第 B 节），这一报告的负相关关系证明了用于评估条件变分自编码器能力的指标是合理的。

关于公式的有效性，我们回顾一下，我们的解码算法唯一可能生成无效公式的情况，是生成的图达到允许的最大节点数 $\max_v$ 。如第 3.2 节所述，在我们的实验中，该上限设为 30。

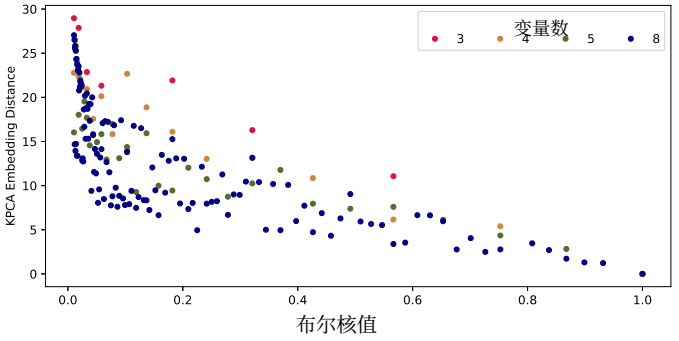


图5：对于包含500个公式的池，随着变量数量变化，核 PCA 嵌入与公式之间核相似度的相关性。

关于重构公式的准确率，我们还注意到，如果对每个测试公式仅考虑最常见的解码结果（这是一种更现实的情形），准确率会有所提高。对于条件和无条件情形，图注意力网络和图卷积网络编码器在这种情形下的准确率分别列于表2和表3中对应模型的第二行。

在图6中，我们展示了如前文所述，使用图注意力网络和图卷积网络编码器的 LogicVAE 所计算的核值和语义距离的分布，

表 2：VAE 能力测试结果，对 300 个测试公式取平均值。GAT 和 GCN 模型准确率列的第二行表示仅考虑每个数据点解码频率最高的公式所计算出的准确率。

	3 变量				4 变量				5 变量			
编码器	准确率	Val.	唯一性。	11月	Acc.	Val.	唯一性。	新颖性	Acc.	Val.	唯一性。	新颖性
GAT	**93.45**	93.42	**58.09**	**54.24**	**93.96**	93.70	**54.13**	**53.59**	**93.92**	89.38	**56.53**	**55.34**
	96.43				**97.32**				**95.42**			
GCN	91.76	**99.64**	28.06	24.83	92.03	**98.82**	31.72	29.35	91.27	**98.78**	29.91	28.32
	94.23				96.34				93.56			
GRU	88.77	98.88	15.56	15.44	83.23	86.96	14.96	13.58	82.24	75.52	10.63	19.16

表 3：条件变分自编码器能力测试结果，百分比为 300 个测试公式的平均值。GAT 和 GCN 模型准确率列的第二行表示仅考虑每个数据点解码频率最高的公式时计算出的准确率。

	3 变量				4 变量				5 变量			
编码器	Acc.	Val.	Sem. 距离	Ker. 值	Acc.	Val.	Sem. 距离	Ker. 值	Acc.	Val.	语义 距离	核 值
GAT	87.75	**98.59**	**8.819**	**0.5025**	88.11	97.55	**7.512**	**0.7692**	**87.43**	**93.72**	**6.317**	**0.7985**
	93.23				**92.76**				**92.45**			
GCN	**89.43**	98.34	9.052	0.4671	**88.87**	**99.55**	8.447	0.6115	85.35	89.14	6.808	0.6924
	94.97				92.45				91.79			

该模型是在包含5个变量的数据集上训练的。此外，我们在表4中报告了它们的第一、第二和第三分位数。

D.1 消融实验

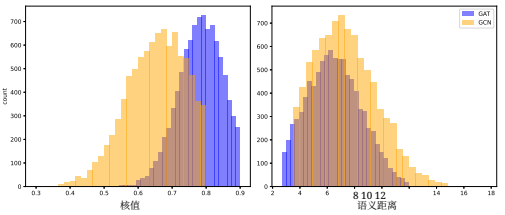
表5报告了我们对 LogicVAE 进行的几项消融研究的结果，具体如下：

1. 无约束解码：这是最重要的设计选择。我们保留了前文所述的编码和目标函数，但移除了对解码器的约束，不再要求其遵循逻辑语法规则（即第3.2节中所述的规则）。这导致重构准确率大幅下降，凸显了这种归纳偏置的根本重要性。2. 单向消息传递：消息仅沿输入有向无环图中边的原始方向传递（即不反转边的方向）。对于 GCN 和有向无环图编码器，这会降低重构准确率，但降幅并不显著。

表4：图注意力网络和

图6：核值（左）和语义的分布

GAT 和 GCN 模型在包含 5 个变量的数据集上的距离（右）。



图卷积网络模型在包含5个变量的数据集上的核值和语义距离分布分位数。

核值			
	Q1	中位数	Q3
图注意力网络	0.7395	0.7843	0.8276
图卷积网络	0.5932	0.6578	0.7169
语义距离			
	Q1	中位数	Q3
图注意力网络	5.097	6.455	7.887
图卷积网络	5.658	7.166	8.846

表5：模型若干消融实验的变分自编码器准确率，在包含5个变量的300个测试公式上取平均。

	非约束单向		层数
编码器	解码		1/2/3/4/5
GAT	32.53	89.66	81.38/89.21/**93.92** /91.37/87.57
GCN	25.64	90.58	89.18/**91.27**/90.53/87.31/85.29

3. 卷积层/注意力层的数量：使用 2 层图卷积网络编码器和 3 层图注意力网络编码器时，我们获得了最高准确率。

D.2 逻辑公式的层次化学习

在进行实验时，我们注意到 LogicVAE 在学习变量索引方面表现不佳；也就是说，如果我们将每个输入公式的简化 AST 作为输入 φ，其重构准确率会提高（架构也会更快收敛）。该图是通过从图 2 和第 3 节所述的输入中移除变量索引构建的。我们的想法是构建一个分层学习模型：先使用 LogicVAE 的句法版本学习重构简化树，然后通过图神经网络架构恢复变量索引。我们使用消息传递神经网络架构实例化了后一种模型，该架构类似于附录 C.2 中所述的架构，并在叶节点上采用节点级读出函数。我们训练该模型，使其最小化一个损失，该损失同时包含一个分类项（即在 n 个类别上计算的交叉熵损失，其中 n 是允许的变量索引最大数量）和一个语义项，用于衡量真实公式 φ （如附录 C.2 所述）的语义向量 $\mathbf{y}_\varphi$ 与.....的语义向量之间的平方损失。

重构公式 $\hat{\varphi}$，在训练算法的每一步中记为 $\hat{\mathbf{y}}_{\hat{\varphi}}$。形式上，我们最小化以下目标（对于具有 M 个叶节点、从 n 个可能索引中选取索引的公式 φ）：

$$\mathcal{L}(\varphi, \hat{\varphi}) = -\frac{1}{M}\sum_{v=1}^{M}(\sum_{i=1}^{n} y_{vi}\log\hat{y}_{vi}) + \lambda||\mathbf{y}_{\varphi} - \hat{\mathbf{y}}_{\hat{\varphi}}||_2 \tag{10}$$

其中，y_{vi}，$\hat{y}_{vi}$ 分别表示叶节点 v具有索引 i的真实概率（及重构概率），而 $\lambda \in \mathbb{R}_{\geq 0}$ 是一个可调参数，用于衡量损失中语义项相对于分类项的权重。有趣的是，我们发现，公式 10 的损失在语义项取值较低时会达到平台期。事实上，我们验证了，在学习进入平台期时，真实公式 φ 与重构公式 $\hat{\varphi}$ 在[10.34% 12.29% 12.48% 16.85%] 的可能配置上存在差异，其中 $n \in [3, 4, 5, 10]$（使用超参数优化得到的结果；在损失中去掉分类项或语义项均未能改善结果）。因此，在某种程度上，节点级消息传递神经网络学会了关注输入公式中语义上相关的部分。我们认为，这一结果有助于更好地理解图神经网络架构在学习逻辑公式时的学习动态，并计划在未来对此进行更深入的研究。