arXiv:2301.06064v1 [cs.LG] 15 Jan 2023

# 基于变分自编码器的符号表达式生成

**Sergei Popov**
计算机科学系
HSE大学LAMBDA

**Mikhail Lazarev**
计算机科学系
HSE大学LAMBDA

俄罗斯国家科技大学 MISIS

通讯作者：sapopov@edu.hse.ru

**Vladislav Belavin**
计算机科学系
HSE大学LAMBDA

**Denis Derkach**
计算机科学系
HSE大学LAMBDA

**Andrey Ustyuzhanin**计算机
科学系 HSE大学LAMBDA

俄罗斯国家科技大学 MISIS

功能智能材料研究所，新加坡国立大学

Constructor大学，瑞士

**摘要**

在物理学、生物学和其他自然科学领域，许多问题都可以通过符号回归获得有价值的见解，并发现新的自然定律。广泛应用的深度神经网络无法提供可解释的解答。与此同时，符号表达式能够清晰地呈现观测结果与目标变量之间的关系。然而，目前符号回归任务尚无占主导地位的解决方案，我们希望通过自己的算法缩小这一差距。在这项工作中，我们提出了一种新颖的深度学习框架，通过变分自编码器（VAE）生成符号表达式。简而言之，我们建议使用 VAE 生成数学表达式，并通过训练策略确保生成的公式能够拟合给定数据集。我们的框架可以将公式的先验知识编码为快速检查谓词，从而加快优化过程。我们将该方法与现代符号回归基准测试进行比较，结果表明，在噪声条件下，我们的方法优于其他方法。在噪声水平为 10% 的 Ngyuen 数据集上，SEGVAE 的恢复率为 65%，比此前报告的最先进水平高出 20%。我们还证明，这一数值取决于数据集，甚至可能更高。

## 引言

乍看之下，从实验观测中发现新的定律似乎是一个古老而早已被遗忘的话题。的确，在数据驱动科学时代，神经网络可以轻松拟合相当复杂的依赖关系。然而，这些网络的泛化能力和可解释性通常都不尽如人意。对于生物技术反应、分子动力学势或流行病传播等现象而言，学习算法若能以简单且易于人理解的方式（例如常用公式）表示目标属性与依赖特征之间的关系，将极具价值。此外，还有许多任务中，黑箱算法不能让人毫无保留地信任，例如自动驾驶汽车、医学、市场分析、飞机设计等领域。换言之，在这些领域中，误差代价极高，因此有必要弄清算法为何做出某种决策。这样的例子随处可见。此外，符号回归技术适用于解决最优控制任务 [1]。最优控制任务旨在为给定系统寻找一条控制律，使其达到某项特定准则；该问题的解是时间的函数 [2]。

近期，人们提出了若干改进所谓符号回归方法的方法 [3], ，其目标是从《费曼物理学讲义》中重建 100 个公式。近期的一篇论文也利用符号回归，根据实验数据找出离子输运的热力学描述 [4] ，所得公式反映了离子输运所依据的物理原理。符号回归类似于简单回归，因为它会拟合实验数据。不过，符号回归试图寻找合适且有效的特征变换，并通过计算图将其表示在原始特征向量之上。使用基于梯度下降的方法优化这类图时，会遇到额外的困难。

这类方法经常面临的另一项难题，是实验测量本身固有的噪声。因此，理想情况下，每种算法都应针对噪声水平和鲁棒性提供理论或经验保证。各种实验数据中都可能存在噪声数据，有时噪声水平会因实验类型而相当高 [5] [6]。由于科学理论或模型必须通过实验来证明或验证，因此，重建观测效应的符号解至关重要。本文提出的算法正是为了解决这一问题。

我们的主要贡献之一，是设计了一种新颖的符号回归训练流程，用于训练 SEGVAE。该流程不同于当前最先进方法，具有更高的噪声稳定性和数据效率，并且可以根据用户的物理直觉调整符号表达式的先验分布。科学家往往知道待搜索公式的框架（例如，界限和近似条件），以及数据必须遵循的规律。我们引入了用于公式搜索的谓词机制，并使已知守恒定律的实现成为可能。这些特性对于处理实验数据至关重要。

本文结构如下：第 2 节概述已有的先验符号方法，第 3 节介绍我们的方法。第 4 节展示了所有比较性能的实验。第 5 节总结全文。

## 文献综述

利用人工智能帮助发现实验数据背后科学规律这一目标，已在多项研究中得到探索。其中一些研究假设人们对所关注的未知环境已有先验知识。然而，与我们的研究最相关的是那些尽可能减少假设的研究。

由于符号回归问题是一项离散优化任务，其搜索空间会随表达式长度呈指数级增长，因此大多数传统方法通常会采用遗传算法 [7] 和 [8]。在这些算法中，优化过程受到自然选择的启发，并依赖变异、交叉和选择等运算符 [9]。新种群通过对当前种群中的个体迭代应用上述遗传运算符而产生。这些方法中最成功的是商业软件 Eureqa ([10])，它于十多年前开发，至今仍在该领域占据领先地位。

近期有多项研究致力于以符号形式恢复物理定律。 [3] 提出了 AI Feynman算法，并在 [11]AI Feynman 2.0 中进一步改进。该算法采用：a）受物理学启发的深度学习策略；b）通过量纲分析等方法搜索对称性、可分离性等特征；c）对简化后的方程进行穷举搜索，以便从实验数据中恢复物理方程。虽然该算法擅长简化表达式，但在恢复那些无法充分简化的表达式时表现不佳。

PySR [12] 基本上是 Eureqa 的重新诞生，它采用了易于使用的界面，支持引入谓词。PySR 基于 Julia 构建，使用正则化进化、模拟退火和无梯度优化，并通过 Python 提供接口。

在 [13] 中展示了一种有趣且直观易懂的方法；此方法后来在 [14] 中得到了更新，并在其最新版本 [15] 中进一步改进。作者提出了一种类似多层感知机的架构，其中不再为所有输出使用单一激活函数，而是采用一组自定义激活函数。作者声称，与多层感知机神经网络相比，该方法在训练集区域之外效率更高，因此展现出了良好的外推能力。但遗憾的是，作者没有在通用数据集上展示与其他现有方法的比较结果。

[16]提出了另一种用于符号回归的深度学习方法。作者提出了一种基于强化学习的符号回归方法，即基于梯度的方法，并将其称为深度符号回归（DSR）。DSR由一个循环神经网络组成，该网络输出数学公式的概率分布。该网络用于采样方程，并根据给定的数据集对这些方程进行评估。随后，评估结果将用于进一步改进数学公式的概率分布，使更好的表达式具有更高的概率。最近，DSR方法通过引入遗传编程组件进行了更新，显著提升了其在多项基准测试中的表现。据我们所知，最新版本的算法表现优于其他方法。因此，我们将其（[17]）用作基准，称为DSO。DSO使用一些预先构建的谓词来避免嵌套重复函数，例如$sin(sin(sin(..)))$，但要引入更复杂或受物理学启发的谓词颇具挑战性。
近期一篇手稿专门探讨了将计算机视觉方法与Transformer结合用于符号回归问题的方法 [18]。其工作流程如下：将输入数据表示为图像，再通过卷积层处理该图像以生成图像嵌入，并将其输入Transformer。然而，所提出的方法可能仅适用于包含大量数据点的数据集。此外，所提出算法的结果并未与已发表的最先进水平方法进行比较。
SciNet（[19]）方法的灵感源自人类进行物理建模时的思维过程。就像人类物理学家并不依赖实际观测，而是依靠对观测的压缩表示来得出理论结论一样，SciNet 将实验数据编码为潜在表示，其中存储着不同的物理参数，并利用这种表示回答有关底层物理系统的特定问题。毫无疑问，SciNet 成功学习了相关的物理概念。然而，它的目标与我们的截然不同：它并不以符号形式恢复定律，而是使用神经网络对其进行建模。

在《可扩展的神经符号回归》（ [20] ）一文中，作者提出使用预训练 Transformer [21] 来预测符号表达式。该网络由一个 Transformer 编码器和一个 Transformer 解码器组成，使用生成的公式进行训练，并采用 BFGS 算法进行常数优化。他们将 Transformer 的结果与 DSR（DSO 的先前版本）进行比较，作为基线。尽管评估速度较快，但在 Nguyen 数据集上的结果质量低于 DSR。
我们的方法与文献 [22]中的工作类似。它通过在编码器和解码器中使用长短期记忆循环神经网络，对变分自编码器进行了改造。因此，形成了一种序列自编码器，并以高斯先验作为隐编码的正则化项。所提出的生成模型融入了对整个句子的分布式潜在表示。通过考察穿过这一潜在空间的路径，可以生成连贯的新句子，并在已知句子之间进行插值。

## 方法

本节介绍基于变分自编码器（SEGVAE）的符号表达式生成算法 [23]。简而言之，我们的架构是一种变分自编码器 [24]，其编码器和解码器均基于循环神经网络。使用变分自编码器进行符号序列表示的主要动机是，变分自编码器提供了一种正则化学习方法，可缩小低能量（噪声）表示空间的范围，从而保留信号（相关序列）表示的丰富性。我们还将在此介绍这一架构的实现方式和训练流程。

### 架构

SEGVAE 按顺序生成公式。这得益于词元序列与公式之间的一一对应关系。每个词元可以是以下三种类型之一：输入变量、常数和运算符。本文中，我们典型的运算符库为 ['$add$', '$sub$', '$mul$', '$div$', '$sin$', '$cos$', '$log$', '$exp$' ]。变量 $X$ 和常数的数量取决于具体任务。我们将在下面的小节中讨论处理常数的方法。我们使用标准波兰表示法，将公式表示为运算符先于其操作数的序列。与传统表示法相比，波兰表示法的主要优势在于其表示无歧义，且无需使用括号。
**变分自编码器**。 VAE 是一种基于潜变量模型的生成式编码器-解码器。给定一个具有分布 $p(x)$ 的观测空间 $\mathcal{X}$，模型的编码器会将其映射到具有分布 $p(z)$ 的潜在空间 $\mathcal{Z}$，而模型的解码器则会将 $\mathcal{Z}$ 映射回观测空间 $\mathcal{X}$。令 $m$ 为潜在空间的维度， $\mu_i$、$\sigma_i^2$为 $i^{th}$

向量 $\mu(x)$ $\sigma^2(x)$ 的分量。若 $p(z) = \mathcal{N}(0,1)$，且 $q_{\theta_E}(z|x) = \mathcal{N}(\mu(x)\ \sigma^2(x))$，则目标函数具有以下形式：

$$-KL(q_{\theta_E}(z|x)||p(z)) + \mathbb{E}_{q_{\theta_E}(z|x)} \log p_{\theta_D}(x|z) \tag{1}$$

这种方法使模型能够从潜在空间中先验分布赋予合理概率的每个点解码出合理的方程。只要我们能将表达式表示为词元序列，就可以采用传统的自然语言处理方法。我们的编码器和解码器都是单层长短期记忆网络，包含 64 个隐藏单元。

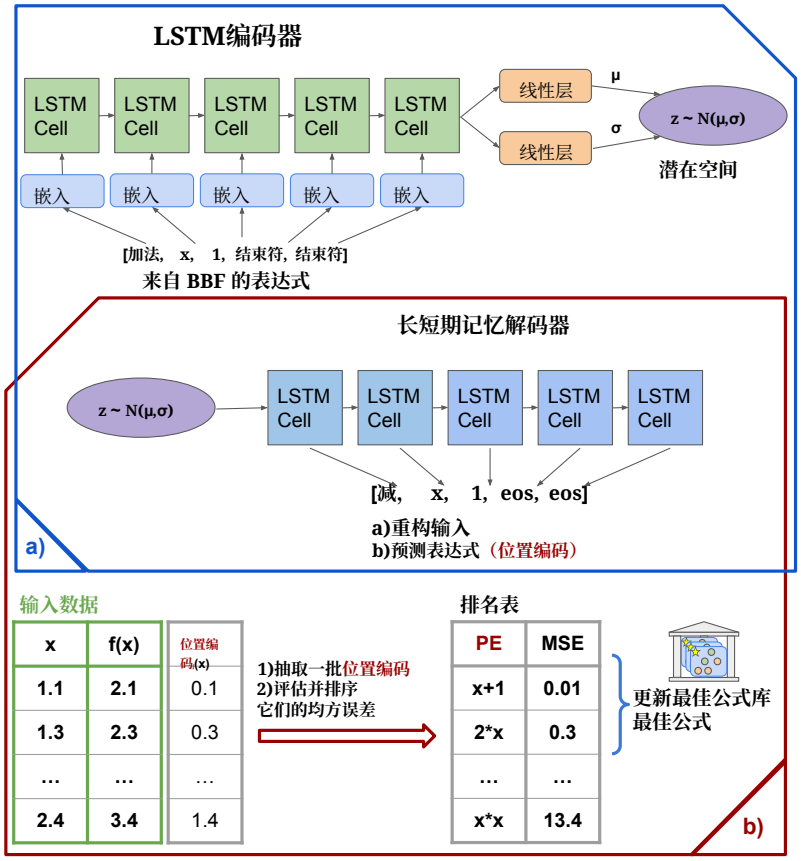



图1：SEGVAE 的架构与训练方案。（a）预训练和训练方案。（b）采样方案，输出公式经过计算后加入公式库

### 训练流程

这里我们概述训练方案的具体细节，该方案包含两个步骤：预训练和主要训练周期。

**预训练**。此步骤使模型能够记忆通用公式结构，并在之后生成有效公式。首先，我们从给定的库 $\mathcal{L}$ 中随机抽取词元序列。然而，均匀采样包含 $n$ 个内部节点的表达式并非易事。朴素算法往往偏向特定类型的表达式。因此，我们采用 [25] 中介绍的数据生成器技术。其次，我们只选择满足谓词条件的语句，并据此创建预训练数据集。此外，我们还向预训练数据集中添加一组自然科学中常见的广义公式。然后，使用这些公式训练变分自编码器。训练完成后，算法后续步骤中生成的绝大多数公式都是有效公式，因此为简便起见，我们可以在后续阶段安全地忽略无效公式。请注意，此步骤完全不依赖目标任务，因此我们针对每个库 $\mathcal{L}$ 只进行一次预训练。预训练过程和算法架构的示意图见图1 (a)。图1 (b) 展示了最佳公式库的构建过程。

**Algorithm 1** SEGVAE: Overall algorithm

**Input:** data $(X, y)$, library of operators $\mathscr{L}$, $N_{epochs} > 0$
$F_{pre-train}$ = **GeneratePretrainFormulas**($\mathscr{L}$)
$F_{pre-train}$ = [ $f$ **if** $predicate(f)$ **for** $f$ **in** $F_{pre-train}$ ]
$vae$.**train**($F_{pre-train}$)

**for** $i = 1$ **to** $N_{epochs}$ **do**
  $F_{train}$ = $vae$.**sample**($batch_size$);
  $F_{train}$ = [ $f$ **if** $predicate(f)$ **for** $f$ **in** $F_{train}$ ]
  $\overline{mse} = \frac{1}{|X|}\sum_j (F_{train}(X_j) - y)^2$
  **if** $F_{train}(x)$ = $NaN$ or out of $Y$ **discard** $F_{train}(x)$
  **if simplify** $F_{train}$ = $simplify(F_{train})$;
  $bbf$ = $bbf$.**update**($F_{train}$, $\overline{mse}$)
  $vae$.**train**($bbf$)
**end for**
**Result:** $bbf.pareto_front$

**主要训练周期**。 当模型能够生成有效公式后，就可以教它生成能够描述待研究环境或系统的有效公式。首先，每个训练轮次都使用变分自编码器采样一批公式。然后，移除所有重复和无效的公式。其次，根据均方误差评估每个公式 $f$ 在数据集 $\mathcal{D} = (X_d,\ y_d)$ 上的表现，计算 $f(X_d)$ 与 $Y_d$ 之间的均方误差：

$$error(f) = \frac{1}{|\mathcal{D}|}\sum_{x,y\in\mathcal{D}} (f(x) - y)^2 \tag{2}$$

然后，将均方误差最小的 $P$ percent 公式作为候选，保存到最佳公式库（BBF）中。首先，需要检查这些候选公式的定义和值是否正确，默认由数据集定义为$x \in (X_{min}, X_{max})$ 和 $y \in (Y_{min}, Y_{max})$。我们在从（$X_{min}, X_{max}$）上的均匀分布中采样得到的点上评估函数 $f(x)$，并计算这些点对应的 $\hat{}y = f(x)$。如果 $\hat{}y$超出（$Y_{min}, Y_{max}$）区域，或出现 $NaN$，我们就丢弃该公式。此方法无法保证函数 $f$在给定领域上有定义，例如，它无法发现不连续点。不过，正如我们的实验所示，此方法提升了模型性能，且与解析评估相比计算效率更高。其次，如有需要，可在保存到 BBF 前使用 $sympy$ 库简化公式。该库保留最近 $N$个训练轮次中的公式。这两个超参数 $P$和 $N$ 的典型取值分别为 20 和 5。最后，使用 BBF 中的公式对变分自编码器进行微调。网络训练的详细信息见补充材料。训练概览如图2所示。

**常数**。 处理所得公式中的常数有两种方法。第一种方法假定所有常数都已纳入库 $\mathcal{L}$。在这种情况下，常数是普通词元，无需对我们的算法进行任何修改。这种方法的主要缺点是表达能力有限。不过，它能显著帮助算法避免对噪声数据过拟合。

第二种方法是将词元 $'const'$ 加入库 $\mathcal{L}$，以便为后续的常数生成占位符。然后，在采样阶段之后，将每个占位符替换为一个参数，并通过 BFGS优化算法 ([26]) 最小化 $f(X_d,\ consts)$ 与 $Y_d$ 之间的均方误差。

**谓词**。 在所提出的算法框架中纳入关于目标公式的先验知识相对简单。在训练阶段，我们通过变分自编码器选择采样得到的最佳公式时，会忽略不满足初始条件的公式，而不考虑其指标。这项技术缩小了搜索空间，有助于 SEGVAE 找到正确的方程，正如下文的实验部分所示。

假设我们知道，多项式之比能够很好地描述物理系统的动力学，而任务是确定这种精确的依赖关系。一种帮助模型的方法是缩小运算符库，从而减少搜索空间，例如排除三角运算。不过，还可以更进一步，为剩余词元可能出现的位置设定条件。在我们的例子中，['$div$'] 运算符只能出现在首位。最终，模型将学会不生成违反先验条件的公式。本文下文将展示算法在选定的公式和谓词列表上的运行结果。公式列表及其相关谓词见表3。

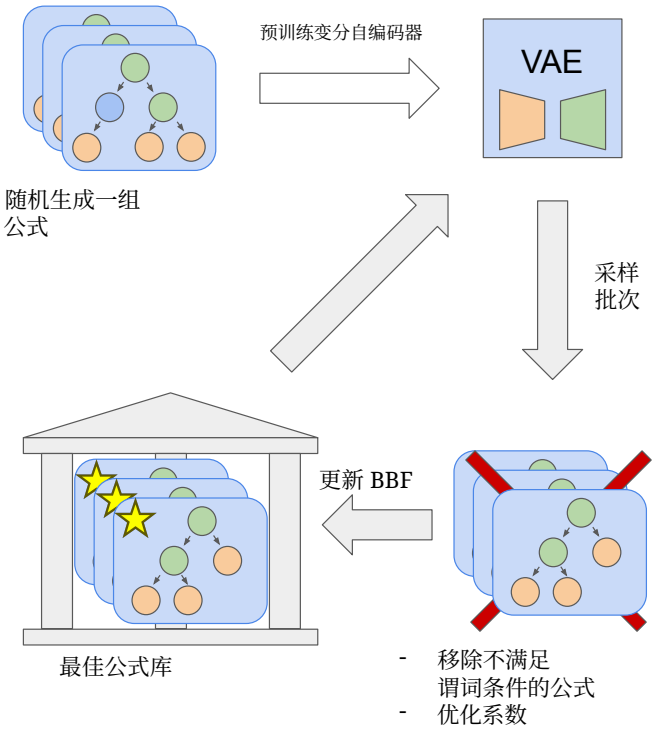



图2：SEGVAE 算法的训练方案。预训练阶段和主要训练周期。

**推理**

训练好变分自编码器模型后，可以采样一批候选表达式，这些表达式应当能够拟合给定的数据集。那些在均方误差方面最能描述数据集的公式，可能会过拟合数据中的噪声。因此，误差与复杂度之间需要权衡。为了选出最终的表达式，我们将每个方程的复杂度评估为：

$$C(t) = \sum_{i}^{T} c(\gamma_i) \tag{3}$$

其中，$c$ 是给定词元的复杂度，所有输入变量、常数和运算符 ['*add*', '*mul*', '*sub*']的复杂度均为 1；$c$ 对于 ['*div*'], 为 2，对于 ['*sin*', '*cos*'] 为 3，最后，对于 ['*log*', '*exp*']为 4。然后，我们使用这些 $C$ 和误差值来确定帕累托前沿，并允许用户挑选符合自身需求的公式。

## 结果

本节报告我们的方法与最先进的符号回归软件包之间的比较结果。我们选取了据我们所知表现最佳的算法，即深度符号优化（DSO） [17],，因为图 3 展示了 DSO 的优异性能。我们使用 Nguyen 1-12 公式（表 1）以及表 3 中列出的公式生成数据集。我们将 SEGVAE 的结果与从 GitHub 仓库获取的 DSO 算法结果进行比较。

**数据集**。 每个符号回归任务都对应一个数字表格，其行的形式为 $x_1$, .., $x_n$, $y$，其中 $y = f(x_1, .., x_n)$。该任务的目标是发现正确的符号表达式 $f$。

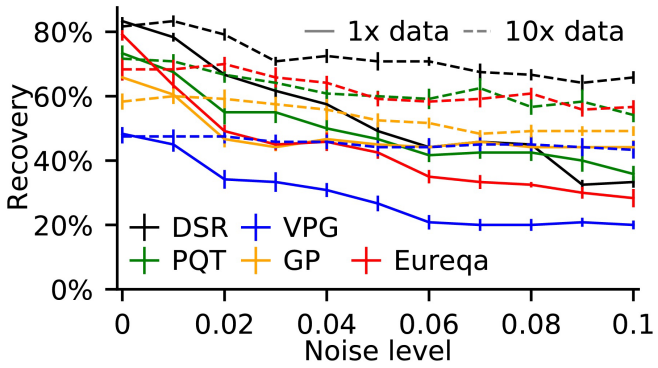



图 3：所有 Nguyen 基准测试中的恢复率与数据集噪声和数据集大小的关系。误差棒表示标准误差。（改编自 [17]）

表 1：Nguyen 数据集。变量表示为 $x_1$ 和 $x_2$。变量均匀采样，U(a, b, c) 表示对于每个输入变量在 $a$ 和 $b$ 之间进行 $c$ 次采样，$N$ 自然数，$L_0 = [add, sub, mul, div, exp, ln, sin, cos]$。

| Name | 表达式 | 数据集 | 库 |
|---|---|---|---|
| Nguyen-1 | $x_1^3 + x_1^2 + x_1$ | $U(-1,\ 1,\ 20)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Nguyen-2 | $x_1^4 + x_1^3 + x_1^2 + x_1$ | $U(-1,\ 1,\ 20)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Nguyen-3 | $x_1^5 + x_1^4 + x_1^3 + x_1^2 + x_1$ | $U(-1,\ 1,\ 20)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Nguyen-4 | $x_1^6 + x_1^5 + x_1^4 + x_1^3 + x_1^2 + x_1$ | $U(-1,\ 1,\ 20)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Nguyen-5 | $sin(x_1^2)cos(x_1)\ - 1$ | $U(-1,\ 1,\ 20)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Nguyen-6 | $sin(x_1) + sin(x_1 + x_1^2)$ | $U(-1, 1, 20)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Nguyen-7 | $log(x_1 + 1) + log(x_1^2 + 1)$ | $U(0,\ 2,\ 20)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Nguyen-8 | $sqrt(x_1)$ | $U(0,\ 4,\ 20)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Nguyen-9 | $sin(x_1) + sin(x_2^2)$ | $U(0,\ 1,\ 20)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Nguyen-10 | $2sin(x_1)cos(x_2)$ | $U(0,\ 1,\ 20)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Nguyen-11 | $x_1^{x_2}$ | $U(0,\ 1,\ 20)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Nguyen-12 | $x_1^4 - x_1^3 + 0.5x_2^2 - x_2$ | $U(0、\ 1、\ 20)$ | $L_0 + [x^N, 0.5, N, e, pi]$ |

• **Nguyen**。 Nguyen 基准常用作符号回归基准。Nguyen 数据集包含 12 个公式。我们使用了与 [16] 及其更新版本中相同的数据集（见表 1）。为检验该方法的鲁棒性，我们添加了与 $y$成比例的高斯噪声。

• **Livermore.** 由 [17] 和 [16]的作者提出的新基准数据集。该数据集包含 22 个公式。不过，由于这些公式与 Nguyen 数据集明显相似，我们并未全部采用。

**消融研究**。 如上所述，SEGVAE 有许多参数，例如层数、预训练阶段采样的公式数量、训练轮次数以及生成表达式的最大长度。为了找到最优参数，我们进行了消融研究。为了调整 SEGVAE 参数，我们选取 Ngyuen 数据集的一个子集，即 Ngyuen-4,5,9,10，作为基准。首先，我们得出结论，最大表达式长度为 30 时，在模型表达能力和模型稳定性之间达到了理想的平衡。其次，我们发现，隐藏维度在 64 到 256 之间时，不会影响 50% 的恢复率。这里的恢复是指建议表达式与原始公式完全符号等价。例如，对于由某个公式生成的给定数据集，我们使用不同的随机种子初始化算法 100 次。在 100 次运行中，有 80 次正确表示了初始公式。在这种情况下，恢复率为 80%。不过，恢复率取决于潜在空间，如表2所示。我们发现，在本研究中，潜在配置空间大小为 128、隐藏单元数为 128 时效果最优。另一个关键观察是，SEGVAE 的恢复率取决于库选择。我们考察了 DSO 和 SEGVAE 算法的恢复率与库中词元数量之间的关系。库规模较小可能是算法无法找到正确公式的原因，因为可用于描述公式的词元数量不足。另一方面，库过大则会在搜索迭代次数有限的情况下，使算法搜索空间呈指数级增长。因此，选择过多的库内容可能导致公式重构效果很差。在实际研究过程中，科学家对未知的依赖关系通常掌握一些先验知识。因此，我们可以同时使用谓词和任务相关库，模拟实际的公式搜索情境。在最后一系列消融

表2：在隐藏维度固定的情况下，子 Nguyen 数据集上的平均恢复率与潜在空间维度的关系。

| 潜在空间维度 | 8 | 16 | 32 | 64 | 128 | 256 |
|---|---|---|---|---|---|---|
| 平均恢复率 | 50 | 58 | 60 | 65 | 75 | 75 |

实验中，我们展示了所提出的方法：丢弃不满足给定领域要求的公式。对于 Nguyen 数据集，该方法使算法收敛速度提高了 50%。此外，实验表明，该方法使 Livermore-5 和 Livermore-7 方程的恢复率提高了 100%

**数据中的噪声**。 通过添加均值为零、标准差与因变量 $y$ 的均方根成正比的高斯噪声，生成含噪数据。为检验模型的鲁棒性，我们展示了噪声从 0%（无噪声）到最高 10% 时的平均恢复率。含噪回归的主要难点在于模型容易对数据过拟合。有时，增加数据集中的点数会有所帮助。

**实验**。 我们在最常用的 Nguyen 基准上评估了 SEGVAE 算法，该基准包含 12 个公式。我们在 Nguyen 数据集的两个变体上比较并评估模型。对于第一个（$Dataset\ I$）变体，我们仅采样 20 个均匀分布的点，如表 1 所示。所有运行中，SEGVAE 使用相同的超参数（详见补充材料）。每次运行考察的表达式数量设为 200 万，与 DSO 论文（[17]）中的设置相同。

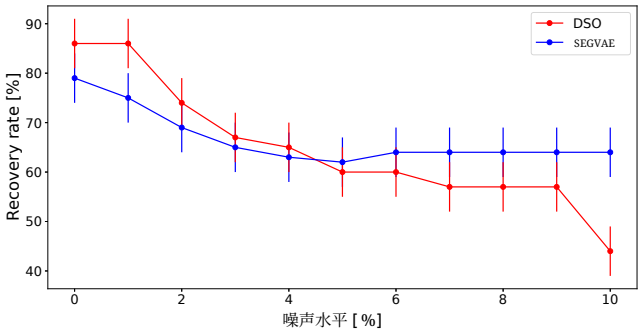



图4：平均恢　复率：SEGVAE（蓝色）和 DSO（红色）算法在 Ngyuen 数据集上的表现，误差条　rs.

图4展示了 DSO 和 SEGVAE 在该数据集上的比较，呈现平均恢复率随噪声水平变化的情况。红色和蓝色分别表示 DSO 和 SEGVAE 算法的输出。需要重申的是，这里所说的恢复，是指帕累托前沿中至少有一个结果与目标公式完全匹配。

在低于中等的噪声水平下，SEGVAE 和 DSO 的平均恢复率（ARR）相近。我们对给定噪声水平下所有 Nguyen 公式的恢复率取平均，以计算 ARR。随着噪声水平升高，DSO 的恢复率缓慢下降。另一方面，SEGVAE 在噪声水平高达 10% 时仍表现出良好的恢复率稳定性，恢复率为 70%。在高噪声水平下，二者的差异开始显现：SEGVAE 略优于 DSO，在最高噪声水平下分别为 70% 和 45%。即使没有先验知识，SEGVAE 算法在该数据集上仍表现出更高的噪声稳定性。
我们比较了原始 DSO论文中的算法和选出的公式，其中 DSO 算法表现出令人瞩目的结果。这些公式列于表 3。本次我们使用特定的谓词和符号库来揭示这些公式，同时展示我们方法的强大之处。我们以相同的方式，通过图5中的平均恢复率（ARR）总结了这项比较的结果。无噪声数据上的详细结果见表4。

科学家通常似乎已经知道研究对象的一般形式。因此，添加谓词以便让 SEGVAE 在特定领域内搜索公式是很自然的。我们已经看到，在没有任何先验知识的情况下，我们的基于VAE的算法在 Nguyen 数据集上取得了与近期 DSO 结果相近的结果。然而，有许多方程无法通过 DSO 足够准确地求解。为了展示谓词在 SEGVAE 中的作用，我们采用了表3所列的谓词，并将结果与 DSO 进行了比较。无噪声数据上的比较结果见表4。使用谓词的 SEGVAE 方法具有比 DSO 更高的恢复率，尤其是在无噪声数据上。无噪声数据上的更详细结果见表4。在这种情况下，恰当的谓词和最优的库起着至关重要的作用。实际上，科学家并不知道研究效应的确切函数形式。在这种情况下，我们不能

表3：公式和所用谓词列表。变量表示为 $x$ 和 $y$。变量均匀采样，U(a, b, c) 表示对每个输入变量在 a 和 b 之间采样 $c$次， $N$ 自然数，$L_0 = [add, sub, mul, div, exp, ln, sin, cos]$。

| 公式名称 | 公式 | 谓词 | 数据集 | 库 |
|---|---|---|---|---|
| Nguyen-12 | $x_1^4 - x_1^3 + \frac{1}{2}x_2^2 - x_2$ | $f(x_1) + g(x_2)$ | $U(0, 10, 200)$ | $L_0 + [x^N, 0.5, N, e, pi]$ |
| Neat-8 | $\frac{exp(-(x_1-1)^2)}{1.2+(x_2-2.5)^2}$ | $exp(f(x_1))/(g(x_2))$ | $U(0.3, 4, 100)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Neat-9 | $\frac{1}{1+x_1^4} + \frac{1}{1+x_2^4}$ | $1/f(x_1) + 1/g(x_2)$ | $U(-5, 5, 21)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Livermore-5 | $x_1^4 - x_1^3 + x_1^2 - x_2$ | $f(x_1) + g(x_2)$ | $U(0, 1, 20)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Livermore-7 | $\frac{1}{2}e^{x_1} - \frac{1}{2}e^{-x_1}$ | $f(x_1) - 1/g(x_1)$ | $U(-1, 1, 20)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Livermore-8 | $\frac{1}{2}e^{x_1} + \frac{1}{2}e^{-x_1}$ | $f(x_1) + 1/g(x_1)$ | $U(-1, 1, 20)$ | $L_0 + [0.5, 1, -1, 2]$ |
| Livermore-10 | $6sin(x_1)cos(x_2)$ | $const * f(x_1) * g(x_2)$ | $U(0, 1, 20)$ | $L_0 + [x^N, 0.5, N, e, pi]$ |
| Livermore-17 | $4sin(x_1)cos(x_2)$ | $const * f(x_1) * g(x_2)$ | $U(0, 1, 20)$ | $L_0 + [x^N, 0.5, N, e, pi]$ |
| Livermore-22 | $exp(-\frac{1}{2}x_1^2)$ | $exp(f(x_1))$ | $U(0, 1, 20)$ | $L_0 + [0.5, 1, -1, 2]$ |
| R-2* | $\frac{(x_1+1)^3}{x_1^2-x_1+1}$ | $f(x_1)/g(x_1)$ | $U(-10, 10, 20)$ | $L_0 + [0.5, 1, -1, 2]$ |

将恢复率用作基准。我们唯一可以信赖的基准是均方误差，以及公式形状或承载科学常识的谓词。

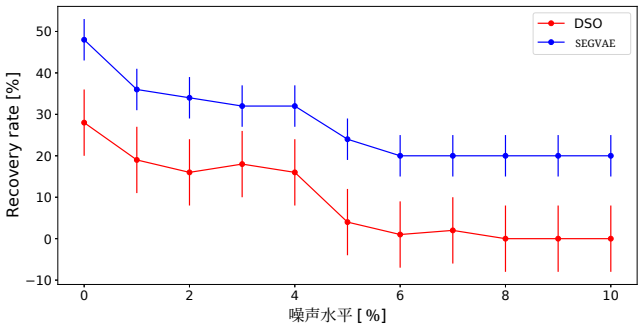



图5：基于表3所列公式的 SEGVAE（蓝色）和 DSO（红色）算法的平均恢复率及误差棒。

## 结论

我们提出了一种新算法 SEGVAE，用于从因变量中搜索函数依赖关系的符号表示。该方法基于能够生成数学表达式的 VAE生成模型，并

表4：在 Nguyen 数据集的子集上，固定隐藏维度时平均恢复率与潜在空间维度的关系。

| | $N-12$ | $Neat-8$ | $Neat-9$ | $L-5$ | $L-7$ |
|---|---|---|---|---|---|
| SEGVAE 100% | | 0% | 0% | 60% | 20% |
| DSO | 0% | 0% | 0% | 80% | 0% |
| | $L-8$ | $L-10$ | $L-17$ | $L-22$ | $R-2*$ |
| SEGVAE 0% | | 100% | 100% | 100% | 0% |
| DSO | 0% | 63% | 57% | 84% | 4% |

受到以快速检查谓词形式编码的先验知识约束。这些谓词可以表示允许的公式模式、领域和可能的输出区间。

作为基准，我们采用了 DSO论文中引入的一组公式，即 Nguyen 数据集和 Livermore 数据集。我们的方法能够灵活地将先验物理知识构建为：a）函数库，b）预训练函数，以及 c）用于剪枝错误生成表达式的选择谓词。除了方法本身的准确性之外，我们还重点关注算法在现实噪声环境中的性能。与深度学习模型相比，符号回归方法表现出显著优势，尤其是在可解释性至关重要的情况下。因此，在材料科学、生物技术和天体物理学等科学领域（仅举几例）中，对这类方法的需求很高。

我们系统地将 SEGVAE 与 DSO 进行了比较，并证明了我们的方法具有更优的性能，从而胜过 Eureqa、Wolfram 及类似方法，而 DSO 此前在这些方法中占据主导地位。在稀缺数据情形和高噪声情形下，SEGVAE 的性能显著优于竞争方法。然而，这种方法也存在局限性：a）输出公式的质量取决于数据集及其规模；b）获得合适的公式可能需要用户协助，即选择谓词，并从帕累托前沿中选出最终公式；c）运行时间可能较长，在现代图形处理器上通常约需几十分钟。

我们指出了库规模的重要性，并展示了 SEGVAE 如何凭借本方法所支持的谓词灵活性，发现 DSO 无法找到的公式。SEGVAE 的恢复率比此前报告的最先进水平提高了 20%。由于实验数据通常含有噪声，且通常可以获得关于函数依赖关系的一些先验知识，因此从应用角度来看，SEGVAE 的优势显而易见。因此，在需要通过可解释的符号解来理解实验观测背后过程的实际场景中，我们的模型将十分有用。代码可通过以下链接获取： $https://github.com/anonnipsuser/segvae$

## 参考文献

[1] Subham S. Sahoo、Christoph H. Lampert 和 Georg Martius。用于外推和控制的方程学习。载于 第*35*届国际机器学习会议论文集，*ICML 2018*，斯德哥尔摩，瑞典，*2018*，第80卷，第4442–4450页。PMLR，2018。[2] A.I. Diveev、S.V. Konstantinov 和 A.M. Danilova。通过符号回归方法求解最优控制问题。*ProcediaComputerScience*，186：646–653，2021。第14届“智能系统”国际研讨会。[3] Silviu-Marian Udrescu 和 Max Tegmark。AI Feynman：一种物理启发的符号回归方法。*ScienceAdvances*，6(16)，2020。[4] Eibar Flores、Christian Wölke、Peng Yan、Martin Winter、Tejs Vegge、Isidora Cekic-Laskovic 和 Arghya Bhowmik。使用符号回归学习电解质中的锂离子传输定律。*Digital Discovery*，1：440–447，2022。[5] Nils Eling、Michael D. Morgan 和 John C. Marioni。测量和理解生物噪声的挑战。*Nature Reviews Genetics*，20(9)：536–548，2019年9月。[6] Patrick A. K. Reinbold、Logan M. Kageorge、Michael F. Schatz 和 Roman O. Grigoriev。通过物理约束符号回归，从含噪、不完整的高维实验数据中进行稳健学习。*NatureCommunications*，12(1)：3219，2021年5月。[7] Dominic P. Searson、David E. Leahy 和 Mark J. Willis。GPTIPS：用于多基因符号回归的开源遗传编程工具箱。2010。[8] John R. Koza。将遗传编程作为通过自然选择编程计算机的手段。*Statistics andcomputing*，4(2)：87–112，1994。[9] Zbigniew Michalewicz 和 Marc Schoenauer。约束参数优化问题的进化算法。*EvolutionaryComputation*，4(1)：1–32，1996年3月。[10] Michael Schmidt 和 Hod Lipson。从实验数据中提炼自由形式自然定律。*Science*，324(5923)：81–85，2009。[11] Silviu-Marian Udrescu、Andrew Tan、Jianhai Feng、Orisvaldo Neto、Tailin Wu 和 Max Tegmark。AI Feynman 2.0：利用图模块性的帕累托最优符号回归。载于 神经信息处理系统进展*33*预出版论文集（*NeurIPS 2020*），2020年12月，2020。[12] Miles D. Cranmer、Alvaro Sanchez-Gonzalez、Peter W. Battaglia、Rui Xu、Kyle Cranmer、David N. Spergel 和 Shirley Ho。利用归纳偏置，通过深度学习发现符号模型。*CoRR*，abs/2006.11287，2020。

[13] Georg Martius 和 Christoph H. Lampert。外推与方程学习。*CoRR*，abs/1610.02995，2016。[14] Subham Sahoo、Christoph Lampert 和 Georg Martius。用于外推和控制的方程学习。载于 Jennifer Dy 和 Andreas Krause（编），第*35*届国际机器学习会议论文集，机器学习研究论文集第80卷，*Proceedings of Machine Learning Research*，第4442–4450页。PMLR，2018年7月10–15日。[15] Matthias Werner、Andrej Junginger、Philipp Hennig 和 Georg Martius。知情方程学习，2021。[16] Brenden K Petersen、Mikel Landajuela Larma、Terrell N. Mundhenk、Claudio Prata Santiago、Soo Kyung Kim 和 Joanne Taery Kim。深度符号回归：通过风险寻求策略梯度从数据中恢复数学表达式。载于国际学习表征会议，2021。[17] T. Nathan Mundhenk、Mikel Landajuela、Ruben Glatt、Claudio P. Santiago、Daniel M. Faissol 和 Brenden K. Petersen。通过神经网络引导的遗传编程种群播种进行符号回归。载于 第*35*届神经信息处理系统会议（*NeurIPS 2021*），2021。[18] Jiachen Li、Ye Yuan 和 Hong-Bin Shen。符号表达式转换器：一种用于符号回归的计算机视觉方法，2022。[19] Raban Iten、Tony Metger、Henrik Wilming、Lídia del Rio 和 Renato Renner。利用神经网络发现物理概念。*Phys. Rev. Lett.*，124：010508，2020年1月。[20] Luca Biggio、Tommaso Bendinelli、Alexander Neitz、Aurelien Lucchi 和 Giambattista Parascandolo。可扩展的神经符号回归。载于 Marina Meila 和 Tong Zhang（编），第*38*届国际机器学习会议论文集，机器学习研究论文集第139卷，*Proceedings of Machine Learning Research*，第936–945页。PMLR，2021年7月18–24日。[21] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Ł ukasz Kaiser 和 Illia Polosukhin。注意力机制即你所需的一切。载于 I. Guyon、U. V. Luxburg、S. Bengio、H. Wallach、R. Fergus、S. Vishwanathan 和 R. Garnett（编），神经信息处理系统进展，第30卷。Curran Associates, Inc.，2017。[22] Samuel R. Bowman、Luke Vilnis、Oriol Vinyals、Andrew Dai、Rafal Jozefowicz 和 Samy Bengio。从连续空间生成句子。载于 第*20*届*SIGNLL*计算自然语言学习会议论文集，第10–21页，德国柏林，2016年8月。计算语言学协会。[23] Popov Sergei、Lazarev Mikhail、Belavin Vladislav、Derkach Denis 和 Ustyuzhanin Andrey。基于变分自编码器的符号表达式生成，2022年11月。[24] Diederik P Kingma 和 Max Welling。自动编码变分贝叶斯。*arXiv* 预印本 *arXiv:1312.6114*，2013。[25] Guillaume Lample 和 François Charton。符号数学的深度学习。载于 国际学习表征会议，2020。[26] R. Fletcher。优化的实用方法。John Wiley & Sons, Ltd，1987。