GenSR：基于方程生成空间的符号回归

Qian Li[1,2], **Yuxiao Hu**[2,3], **Juncheng Liu**[4], **YuntianChen**[2*1]上海交通大学，中国上海[2]宁波东方理工大学，中国宁波[3]香港理工大学，中国香港[4]伦敦帝国理工学院，英格兰伦敦
qianl01205@sjtu.edu.cn, yuxiao.hu@connect.polyu.hk,
junchengliu23@imperial.ac.uk, ychen@eitech.edu.cn

arXiv:2602.20557v1 [cs.LG] 24 Feb 2026

摘要

符号回归（SR）旨在揭示观测数据背后隐藏的方程。然而，大多数方法都在离散的方程空间中搜索，其中方程的结构修改很少与其数值行为相匹配，导致拟合误差反馈过于嘈杂，无法有效引导探索。为应对这一挑战，我们提出 GenSR[1]，一种基于生成式潜在空间的符号回归框架，遵循“地图构建 $\rightarrow$ 粗定位 $\rightarrow$ 精细搜索”的范式。具体而言，GenSR 首先预训练一个双分支条件变分自编码器（CVAE），将符号方程重新参数化到一个具有符号连续性和局部数值平滑性的生成式潜在空间中。该空间可视为方程空间的一幅结构良好的“地图”，为搜索提供方向性信号。在推理阶段，CVAE 将输入数据粗略定位到潜在空间中有希望的区域。随后，改进的 CMA-ES 利用平滑的潜在梯度对候选区域进行细化。从贝叶斯视角看，GenSR 将符号回归任务重新表述为最大化条件分布 p（Equ.|Num。），而 CVAE 训练则通过证据下界（ELBO）实现这一目标。这一新视角为 GenSR 的有效性提供了理论保证。大量实验表明，GenSR 能够同时优化预测准确性、表达式简洁性和计算效率，并且在噪声环境下仍保持稳健。

1 引言

符号回归（SR）是一项核心机器学习任务，旨在发现能够解释观测数据、且具有可解释性的数学表达式。与假定固定函数形式（例如线性或多项式形式）的传统回归不同，SR 会探索方程空间，以找出最适合数据的函数。形式上，给定 N 观测样本 $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$，目标是找到一个数学表达式 f：$\boldsymbol{x} \mapsto y$，以捕捉底层数据分布。凭借其灵活性和可解释性，SR 被广泛应用于科学发现和工程学领域（Xu 等人，2023 年；Chen 等人，2022），在这些领域中，理解底层方程与预测准确性同样重要。SR 有效地弥合了黑箱模型与可解释的科学洞见之间的鸿沟。因此，越来越多的研究致力于开发能够兼顾拟合精度、计算效率和表达式简洁性的符号回归算法。

*通讯作者[1]https://github.com/tokaka22/ICLR26-GENSR

许多现有的符号回归方法直接在离散符号空间中搜索方程，采用遗传编程（GP）（Stephens 等人，2015；Virgolin 等人，2021）、蒙特卡洛树搜索（MCTS）（Sun 等人）或其他组合优化方法（Yu 等人，2025 年；Zhang 等人，2025 年）等启发式策略。这些方法通过交叉、变异和树扩展等操作，迭代调整方程的符号结构，以搜索更优解。遗憾的是，这些修改并不能可靠地降低拟合误差，甚至可能使搜索陷入次优区域。这是因为编辑距离与数值行为相似性是彼此无关的度量：结构相似的方程可能呈现截然不同的数值行为，反之亦然。因此，在以编辑距离衡量相似性的离散方程空间中，数值误差反馈既不对应编辑距离，也无法提供可靠的搜索方向。搜索过程严重依赖随机变异、组合和回溯，导致搜索轨迹不稳定且时间复杂度高。因此，采用编辑距离的离散方程空间效率低下，甚至不适用于符号回归。

在本文中，我们将离散的方程空间重新参数化为连续潜在空间，可将其视为方程世界的一幅“地图”（其中方程表示为向量，而各维度自然定义了搜索方向）。为了在这一空间中实现高效搜索，必须解决两个关键问题：（1）**空间应具备生成能力**。由生成模型构建的生成式潜在空间支持连续采样，使潜在向量能够解码为语法有效的方程。这一特性使得方程潜在空间中的平滑插值和细粒度搜索成为可能。相比之下，基于预训练的方法（Meidani 等人；Kamienny 等人，2022）通常学习用于相似性判断或预测的判别式嵌入，而非对生成分布建模。这类判别式空间支离破碎，且充斥着不可解码区域，妨碍连续搜索并产生无效样本。

（2）**空间应具有良好的组织结构**。尽管一些方法尝试为方程构建生成式潜在空间（Meˇznar 等人，2023 年；Popov 等人，2023 年），其中的欧几里得距离却只能反映编辑距离，与方程的数值行为无关。因此，数值误差无法映射为有意义的欧几里得距离以实现高效优化，这类潜在空间仍无法支持平滑、高效的搜索。理想情况下，数值行为相似的方程应在潜在空间中彼此接近，从而让平滑的数值误差信号引导探索。然而，像 $\cos^2 x$，$1 - \sin^2 x$ 这样的表达式及其泰勒展开具有相同的数值行为，却有着截然不同的符号形式。若将这类表达式置于彼此接近的位置，解码器就不得不将相邻向量映射为差异极大的表达式，从而导致解码不稳定。因此，如何平衡符号连续性与数值平滑性是一项关键的设计挑战。受人类定位方式的启发——先确定粗略区域，再精确定位——我们构建了一个兼具全局符号连续性和局部数值平滑性的生成式潜在空间。前者确保稳定解码，并形成清晰的结构聚类以实现粗定位；后者则使拟合误差能够在找到相关区域后提供方向信号，支持细粒度搜索。

基于这一洞见，我们提出了 GenSR，这是一种遵循“地图构建 $\rightarrow$粗定位 $\rightarrow$ 精细搜索”范式的新型框架。GenSR 首先对双分支条件变分自编码器（CVAE）进行预训练：后验分支学习给定符号和数值输入时符号结构的分布，而先验分支仅根据数值输入学习数值特征的分布。联合优化重构损失和 KL 散度，可为**符号回归**生成具有全局符号连续性和局部数值平滑性的生成式潜在空间，充当方程世界的“地图”。在推理阶段，先验分支将输入数据映射为初始定位分布，从而突出有希望的潜在区域。随后，改进的 CMA-ES 算法将该分布逐步收缩至最优解，利用潜在空间中的平滑方向信号实现高效收敛。从概率论视角看，GenSR 将符号回归重新表述为贝叶斯优化问题，即最大化p(Equ.|Num.)，其中 CVAE 训练对应于证据下界优化。这一视角为 GenSR 提供了理论保证。大量实验表明，GenSR 始终能在预测准确性、表达式简洁性和运行效率之间实现最佳平衡，同时在噪声条件下保持鲁棒性。

我们的贡献主要有四个方面：(1) GenSR 构建了一个连续生成潜空间，将符号连续性与局部数值平滑性统一起来，解决了离散方程空间效率低下的问题；(2) GenSR 遵循“地图构建 $\rightarrow$ 粗定位 $\rightarrow$ 精细搜索”这一全新的搜索范式，以实现有效的符号回归；(3) GenSR 为符号回归提供了全新的贝叶斯视角，并通过证据下界优化求解该问题；(4) 大量实验全面分析了潜在空间属性，并证明我们的方法能够在准确率、方程复杂度和效率之间实现更好的平衡。

2 相关工作

符号回归领域经历了多种方法的发展。最常见的符号回归方法依赖于在离散方程空间中使用启发式搜索进行组合优化。遗传编程（GP）技术（Stephens 等人，2015；Virgolin 等人，2021；Burlacu 等人，2020；Cranmer，2023）通过选择和变异来进化表达式种群，而蒙特卡洛树搜索（MCTS）方法（Sun 等人）则系统地探索表达式树空间。近年来，受神经网络引导的方法整合了强化学习（Petersen 等人）、基于 Transformer 的规划（Shojaee 等人，2023）、检索增强生成（Zhang 等人，2025 年）以及最小描述长度原理（Yu 等人，2025 年），以提升搜索效率。大语言模型也被用于通过提示技术生成方程（Shojaee 等人；Grayeli 等人，2024）；然而，由于固有的幻觉问题，基于大语言模型的方法仍类似于在离散空间中进行枚举，需要进行大量查询才能获得高质量解。归根结底，在离散方程空间中进行符号回归从根本上仍然效率低下，因为结构相似性无法很好地反映数值行为，导致搜索缺乏引导，误差反馈也难以提供有效信息。

另一些策略试图完全避免离散搜索。序列到序列模型（Kamienny 等人，2022；Li 等人，2022；Biggio 等人，2021）直接生成方程，但在训练过程中并未约束符号形式与数值行为之间的对齐，从而限制了模型的泛化能力。SNIP 框架（Meidani 等人）通过对比学习来学习共享的潜在空间，取得了进展，并由此创建了有助于相似性评估的表征。然而，由于 SNIP 学习的是判别式而非生成式空间，其潜在空间中的许多点并不对应有效方程，因此无法满足符号回归对生成式搜索的需求。

我们的工作通过采用双分支条件变分自编码器，学习连续的生成式潜在空间，从而解决了这些局限。该方法将离散的方程空间转换为结构化流形，其中相邻点对应符号形式和数值属性相似的表达式。由此得到的空间提供了有意义的梯度，使高效的三阶段搜索策略——地图构建、粗定位和细粒度搜索——成为可能，直接针对以往方法的核心弱点。

3 GENSR：基于**符**号**回**归的**生**成潜在空间

E

如图 1 所示，GenSR 的核心思想是利用双分支条件变分自编码器为方程构建一个连续生成潜空间（可视为方程世界的“地图”），并直接在该空间中进行高效搜索。双分支条件变分自编码器由基于 Transformer 的编码器 – 解码器网络、两个分支专属网络以及一个特征融合模块组成。该条件变分自编码器在约 500 万个合成方程–样本对上进行预训练，以确保覆盖多样的函数形式。在推理阶段，GenSR 利用先验分支在潜在空间中生成初始定位分布，并使用 CMA-ES 的退化版本（Hansen，2016）在潜在空间中细化求解结果，从而实现对候选方程分布的快速而精确的搜索。

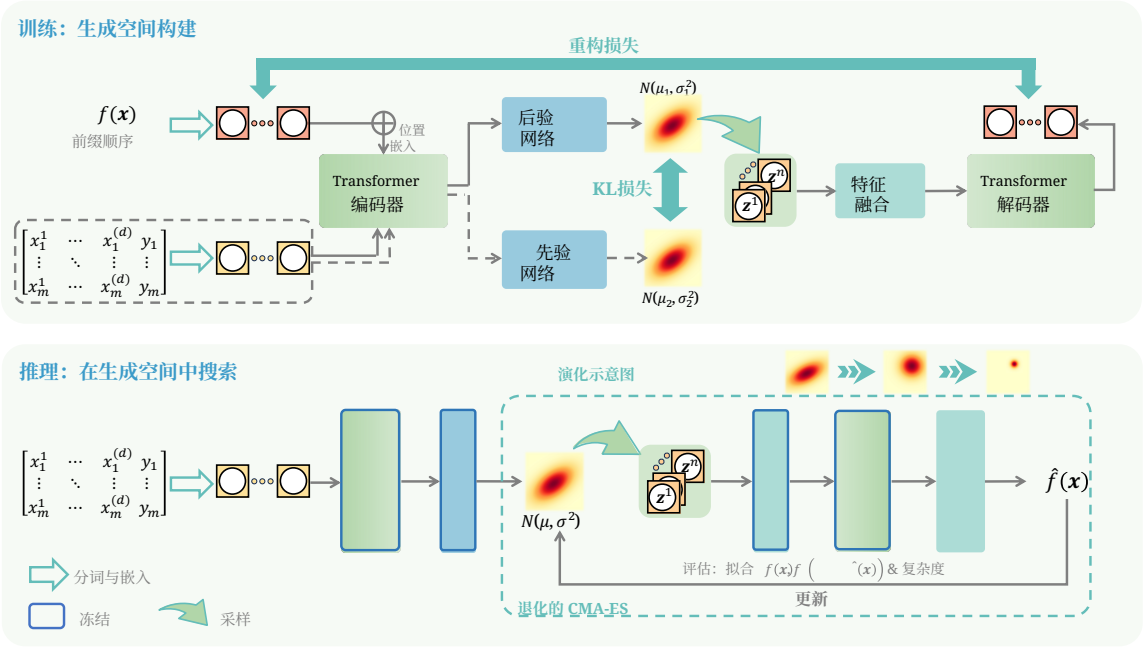


图 1：GenSR 概览。训练期间，虚线表示先验分支，实线表示后验分支。推理期间，仅使用先验分支。

3.1 准备工作

为构建生成式潜在空间，我们首先准备合成数据，并对输入的数值样本和方程进行预处理，从而为双分支 CVAE 预训练奠定基础。

合成数据的准备。 为了构建生成式潜在空间，我们在大型合成数据集上预训练条件变分自编码器（图 1）。遵循 Kamienny 等人（2022）；Meidani 等人的做法，我们构建了一个数据集，其中每个方程 f 都与 m 个数值示例 $(\boldsymbol{x},\ y)_{i=1}^{m} \in \mathbb{R}^{m\times(D+1)}$配对，其中 D 表示输入变量的维度。请注意，我们控制表达式长度，使复杂方程趋向潜在空间的外围，从而减轻搜索过程中对复杂解的过拟合。

数值样本的预处理。 遵循 Kamienny 等人（2022）的做法，我们使用十进制浮点表示对每个数值样本进行分词，将其四舍五入至四位有效数字，并分解为符号、尾数（0–9999）和指数（$E-100$–$E100$），例如，0.7895 被分词为 $[+, 7895, E-4]$。每个方程 f 的m 个样本表示为 $\{(\boldsymbol{x},\ y)\}_{i=1}^{m} \in \mathbb{R}^{m\times 3(D+1)}$。具有降维功能的可学习嵌入器会将这些词元嵌入映射到 $X \in \mathbb{R}^{m\times d}$，作为 Transformer 编码器的数值输入（Kamienny 等人，2022）。

方程的预处理。每个表达式 f 都表示为一棵二叉树，并按前缀顺序线性化（Lample 和 Charton）。运算符、变量和整数会映射为专用词元，而常数则与数值样本采用相同的方式进行分词。插入特殊词元 [〈BOS〉] 和 [〈EOS〉]，分别标记每个序列的开头和结尾。具体而言，序列会填充至固定长度 m，以匹配数值样本的序列长度。每个词元都会嵌入到一个带有位置编码的 d 维空间中，形成 $\boldsymbol{F} \in \mathbb{R}^{m\times d}$，作为输入传递给 Transformer 编码器。

3.2 预训练用于方程的生成式潜在空间

我们采用图 1 所示的双分支条件变分自编码器，为方程学习一个结构良好的生成式潜在空间，可将其视为方程空间的“地图”。这张地图能对高概率区域进行初始定位，并支持朝最优解进行平滑的定向搜索，从而实现高效的方程发现。双分支条件变分自编码器由后验分支和先验分支组成。遵循 Kingma 与 Welling（2013）和 Sohn等人（2015），我们假设潜变量的条件分布为多元独立高斯分布。接下来，我们将介绍各分支及其训练细节。

后验分支（图 1 中的实线）：由一个 8 层 Transformer 编码器、一个后验网络、一个特征融合多层感知机和一个 8 层 Transformer 解码器组成。它将 $\boldsymbol{X}$（数值样本）和 $\boldsymbol{F}$（符号方程）作为输入，通过后验网络学习后验分布 $q(\boldsymbol{z}|\boldsymbol{X},\ \boldsymbol{F}) = \mathcal{N}(\boldsymbol{\mu}_1,\ \boldsymbol{\sigma}_1^2\boldsymbol{I})$。利用重参数化技巧，我们采样得到 n 潜向量 $\boldsymbol{Z} = \{\boldsymbol{z}_i = \boldsymbol{\mu}_1 + \boldsymbol{\sigma}_1 \odot \boldsymbol{\epsilon}\}_{i=1}^n$，其中 $\boldsymbol{\epsilon} \sim \mathcal{N}(0,\ I)$，$\odot$ 表示逐元素乘法。特征融合多层感知机聚合采样潜向量中的关键信息，Transformer 解码器重构原始符号方程，从而确保潜在空间能够捕捉符号结构的连续性。

先验分支（图 1 中的虚线）：先验分支与后验分支共享所有模块（Transformer 编码器/解码器、特征融合多层感知机），但以后验网络替换为先验网络。它仅将数值样本 $\boldsymbol{X}$ 作为输入，并学习先验分布 $p(\boldsymbol{z}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{\mu}_2,\ \boldsymbol{\sigma}_2^2\boldsymbol{I})$，该分布捕捉方程的数值行为。

训练过程：训练期间，两个分支在配对数据（$\boldsymbol{F}$，$\boldsymbol{X}$）上通过组合目标函数进行联合优化：（1）重构损失 L_{rec} 通过从潜在空间准确解码方程，确保符号结构的连续性；（2）KL 散度 D_{KL} 使后验分布 $q(\boldsymbol{z}|\boldsymbol{X},\ \boldsymbol{F})$ 与先验分布 $p(\boldsymbol{z}|\boldsymbol{X})$ 对齐，从而确保局部数值平滑性。分布对齐对先验网络进行正则化，使其在真实方程未知时仍能支持可靠推理。训练期间，我们采用 KL 退火来缓解后验坍塌。

3.3 生成式潜在空间属性分析

我们进一步解释为什么第 3.2 节中的 GenSR 能够产生一种生成式潜在空间：其符号结构具有全局连续性，数值特征具有局部平滑性。

符号结构的连续性。令 $q^{(1)}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{\mu}^{(1)},\ \boldsymbol{\sigma}^{(1)2}\boldsymbol{I})$ 和 $q^{(2)}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{\mu}^{(2)},\ \boldsymbol{\sigma}^{(2)2}\boldsymbol{I})$ 分别表示输入样本对（$\boldsymbol{F}^{(1)},\ \boldsymbol{X}^{(1)}$）和（$\boldsymbol{F}^{(2)}\ \boldsymbol{X}^{(2)}$）对应的两个输出后验分布。我们将每个分布在置信度水平 α 下的高置信区域定义为：

$$\mathcal{R}_\alpha^{(i)} = \left\{ \boldsymbol{z} \in \mathbb{R}^d : \sum_{j=1}^{d} \frac{(z_j - \mu_j^{(i)})^2}{(\sigma_j^{(i)})^2} \le \chi^2_{d,\alpha} \right\}, \quad i = 1, 2,$$

其中$\chi^2_{d,\alpha}$ 是自由度为 d 的卡方分布的 α分位数。对于两个分布，它们的 n采样集合 $\boldsymbol{Z}^{(j)} = \{\boldsymbol{z}_i^{(j)} = \boldsymbol{\mu}^{(j)} + \boldsymbol{\sigma}^{(j)} \odot \boldsymbol{\epsilon}\}_{i=1}^n$（其中 $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$，$j = 1, 2$）通常落在各自的高概率区域 $\mathcal{R}_\alpha^{(1)}$ 和 $\mathcal{R}_\alpha^{(2)}$内。在此，与传统VAE框架仅采样单个 $\boldsymbol{z}$不同，我们采用重复概率采样和特征融合模块，以确保重构损失影响分布的高概率区域，从而增强潜在空间的符号连续性。若存在 $\boldsymbol{z}^* \in \mathcal{R}_\alpha^{(1)} \cap \mathcal{R}_\alpha^{(2)}$，则它由 $f_{(1)}$和 $f^{(2)}$两者的重构目标共同塑造，从而产生在二者之间插值的解码结构。这促使潜在空间中形成平滑的符号过渡。在多样化方程上训练后，这种局部插值可全局泛化，使结构相似的方程聚集在相邻区域，并诱导出具有连续符号结构的潜在空间。

数值特征的平滑性。 我们利用先验分支捕捉数值分布的特征，并通过KL 散度损失将其与符号潜在空间对齐，从而对分布施加数值约束。该设计类似于SNIP（Meidani 等人）中的对比学习，但有一个关键区别：我们对齐的是整个分布而非单个点，这确保了生成式潜在空间内的局部数值平滑性。具体而言，KL损失 $D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F}) \parallel p(\boldsymbol{z}|\boldsymbol{X}))$将后验分布正则化，使其保持接近先验分布。对于我们的条件变分自编码器（$\boldsymbol{\Sigma}_1 = \boldsymbol{\sigma}_1^2\boldsymbol{I}$, $\boldsymbol{\Sigma}_2 = \boldsymbol{\sigma}_2^2\boldsymbol{I}$）中所假设的独立高斯分布，该散度可简化为:

$$D_{\mathrm{KL}}(q \parallel p) = \frac{1}{2}\sum_{j=1}^{d}\left(\frac{(\mu_{1,j}-\mu_{2,j})^2}{\sigma_{2,j}^2} + \frac{\sigma_{1,j}^2}{\sigma_{2,j}^2} - \ln\frac{\sigma_{1,j}^2}{\sigma_{2,j}^2} - 1\right).$$

在我们的双分支条件变分自编码器中最小化该项，可约束后验均值与先验均值之间的距离，同时对齐它们的方差，从而确保匹配的是高概率区域——而非孤立点——以保持局部数值平滑性。

因此，GenSR 为符号回归建立了一个结构良好的生成式潜在空间。我们将在第 5.3 节中对该空间的性质提供更多验证。

3.4 生成式潜在空间中的搜索

在推理过程中，由于真实方程不可用，仅使用先验分支。推理过程如图 1 所示。

对于给定的一组数值样本 $\{(\boldsymbol{x}, y)_i\}_{i=1}^m$，我们首先按照第 3.1 节所述对其进行预处理。随后，先验分支将其编码为潜在空间中的初始高斯分布 $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0^2\boldsymbol{I})$。该分布提供粗定位，突出可能包含能良好拟合数据的方程的区域。例如，高概率区域与合理的符号族（例如三角、指数函数）以及输入维度相对应，从而为细粒度搜索提供结构化的起点。

由于先验和后验潜在空间仅为近似，以端到端方式直接解码向量 $\boldsymbol{\mu}_0$ 可能无法得到最准确的结果。为了将初始分布 $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0^2\boldsymbol{I})$ 向最优解细化，我们采用改进的协方差矩阵自适应进化策略（CMA-ES）（Hansen，2016），利用潜在空间的数值平滑性实现高效收敛。算法如下:

1. **初始化**：将初始搜索分布设为 $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0^2\boldsymbol{I})$（其中 $\boldsymbol{\mu}_0$ 和 $\boldsymbol{\sigma}_0$ 为先验网络的输出）。2. **迭代**（对于每次生成 i）：(a) 从 $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\boldsymbol{I})$ 中采样多个潜在向量 $\{\boldsymbol{z}_j\}$。(b) 使用训练良好的解码器将 $\{\boldsymbol{z}_j\}$ 解码为候选方程 $\{f_j(\boldsymbol{x})\}$。(c) 通过 BFGS 优化细化 $\{f_j(\boldsymbol{x})\}$ 的常数。(d) 评估候选方程的适应度：适应度 $= R^2 - \omega \cdot$复杂度。(e) 选择前 p 个候选，并使用 CMA-ES 更新规则更新分布参数 $\boldsymbol{\mu}_i \to \boldsymbol{\mu}_{i+1}$ 和 $\boldsymbol{\sigma}_i \to \boldsymbol{\sigma}_{i+1}$。由此得到更新后的分布

先验分支为 CMA-ES 提供了组织良好的生成式潜在空间，从而实现更快、更稳定的收敛。然而，随着潜在维度 d 增大，CMA-ES 的开销也会增加。为提高效率，我们对标准 CMA-ES 做出两项关键改进:

1. **对角协方差假设**：受 Ros 与 Hansen（2008）启发，我们将协方差矩阵 $\boldsymbol{\Sigma}$ 限制为对角形式 $\boldsymbol{\sigma}^2\boldsymbol{I}$，仅独立地更新每个潜在维度的方差。这与变分自编码器中的潜变量假设一致（Kingma 和 Welling，2013）（建模为独立高斯分量），并减少了需要优化的参数数量。

2. **Top-k 方差更新**：仅更新方差最大的前 k 个潜在维度（其余维度设为零）。这使搜索集中于最不确定且最相关的方向，从而加快收敛，同时不损害解的质量。

这些改进能够快速而精确地收缩初始分布，从而快速定位目标方程。图 1 展示了 CMA-ES 分布收缩过程的示意图。

4 GenSR：贝叶斯视角下的符号回归

GenSR 可以自然地解释为概率推理，并将符号回归任务重新表述为一个贝叶斯问题：给定数值样本 $\boldsymbol{X} \in \mathbb{R}^{m\times d}$，目标是在方程空间上推断能够最佳解释数据的后验分布 $p(\boldsymbol{F}|\boldsymbol{X})$，即最大化 $p(\boldsymbol{F}|\boldsymbol{X})$。与寻求单一最优解的传统符号回归任务不同，这种贝叶斯表述对候选方程的整个后验分布进行推理，从而能够以有原则的方式量化不确定性，并对候选方程进行概率比较。

为了近似难解分布 $p(\boldsymbol{F}|\boldsymbol{X})$，我们引入变分分布 $q(\boldsymbol{z}|\boldsymbol{X},\ \boldsymbol{F})$，其中 $\boldsymbol{z} \in \mathbb{R}^d$ 是一个潜向量。遵循 Sohn 等人（2015）的方法，我们推导边缘对数似然 $\log p(\boldsymbol{F}|\boldsymbol{X})$ 的证据下界（ELBO），如下所示。完整推导见附录 B。

$$\begin{aligned}\log p(\boldsymbol{F}|\boldsymbol{X}) &= \int_z q(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F}) \log p(\boldsymbol{F}|\boldsymbol{X}) d\boldsymbol{z} \\ &\geq \int_z q(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F}) \log \frac{p(\boldsymbol{F},\boldsymbol{z}|\boldsymbol{X})}{q(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F})}\, d\boldsymbol{z} \\ &= \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F})}\left[\log p(\boldsymbol{F}|\boldsymbol{X},\boldsymbol{z})\right] - D_{\mathrm{KL}}\left(q(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F}) \,||\, p(\boldsymbol{z}|\boldsymbol{X})\right). \end{aligned} \tag{1}$$

最大化 $p(\boldsymbol{F}|\boldsymbol{X})$ 等价于最大化证据下界 1。为了优化这个目标函数，GenSR 使用后验编码器 $q_{\phi,\varphi}(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F})$、先验编码器 $p_{\phi,\psi}(\boldsymbol{z}|\boldsymbol{X})$ 和解码器 $p_\theta(\boldsymbol{F}|\boldsymbol{X},\boldsymbol{z})$，分别对 $q(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F})$、$p(\boldsymbol{z}|\boldsymbol{X})$ 和 $p(\boldsymbol{F}|\boldsymbol{X},\boldsymbol{z})$ 进行参数化。随后，使用 L_{rec} 和 D_{KL} 训练我们的双分支条件变分自编码器，等价于最大化证据下界 1：最大化 $\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F})}[\log p(\boldsymbol{F}|\boldsymbol{X},\boldsymbol{z})]$，对应于最大化使用从 $q_{\phi,\varphi}(\boldsymbol{z}|\boldsymbol{X},F)$ 输出中采样得到的 $\boldsymbol{z}$ 来重构 $\boldsymbol{F}$ 的概率，即最小化重构损失 L_{rec}；不等式 1 中的第二项对应于第 3.2 节中的 KL 散度损失，它使变分分布 $q_{\phi,\varphi}(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F})$ 与先验分布 $p_{\phi,\psi}(\boldsymbol{z}|\boldsymbol{X})$ 对齐。在推理过程中，我们从先验分布 $p_{\phi,\psi}(\boldsymbol{z}|\boldsymbol{X})$ 中采样 $\boldsymbol{z}$ 来重构方程 $\boldsymbol{F}$，以此近似难以计算的期望 $\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F})}[\log p_\theta(\boldsymbol{F}|\boldsymbol{X},\boldsymbol{z})]$，即 $\mathbb{E}_{p_\psi(\boldsymbol{z}|\boldsymbol{X})}[\log p_\theta(\boldsymbol{F}|\boldsymbol{X},\boldsymbol{z})]$。为减轻固有的近似误差，我们按照第 3.4 节所述对先验分布进行优化。

值得注意的是，GenSR 与 Holt 等人提出的方法不同，后者在 token 层面近似 $p(\text{Token}|\text{Num.})$，却没有构建符号方程上的潜在分布；GenSR 也不同于 Meˇznar 等人（2023 年）和 Popov 等人（2023 年）的方法，后者只对 $p(\text{Equ.})$ 建模。GenSR 从贝叶斯推断视角出发，明确将符号回归任务表述为最大化条件分布 $p(\text{Equ.}|\text{Num.})$。据我们所知，GenSR 是首个通过估计并优化 $p(\text{Equ.}|\text{Num.})$ 的证据下界来实现符号回归的框架。在 GenSR 框架下，我们的双分支条件变分自编码器预训练模型直接对应一个证据下界，而 CMA-ES 精化过程则充当对条件变分分布的近似优化。这不仅为 GenSR 提供了坚实的理论保证，也为符号回归引入了新的贝叶斯视角和技术路径。

5 实验

5.1 实验设置

为验证我们的方法，我们在 SRBench基准（La Cava 等人，2021；Cavalab，2022）上进行评估。该基准包含 119 个费曼方程、14 个 ODE-Strogatz挑战，以及 57 个没有已知底层函数的黑盒回归任务。

我们将方法与 18 种基线算法进行比较，涵盖基于预训练的方法和多种基于启发式的方法。评估指标包括准确率R^2、时间复杂度和方程复杂度。为更好地权衡并直观呈现这三项指标之间的取舍，我们采用帕累托前沿进行表示。有关数据集、评估指标和基线方法的详细信息，请参见附录 G.1、G.2 和 G.3。为清晰起见，本节仅展示关键实验结果；完整结果见附录 G.5。其他分析，包括消融研究以及基于我们生成空间的其他任务，见附录 E 和 G.4。

5.2 GenSR 与其他方法的比较

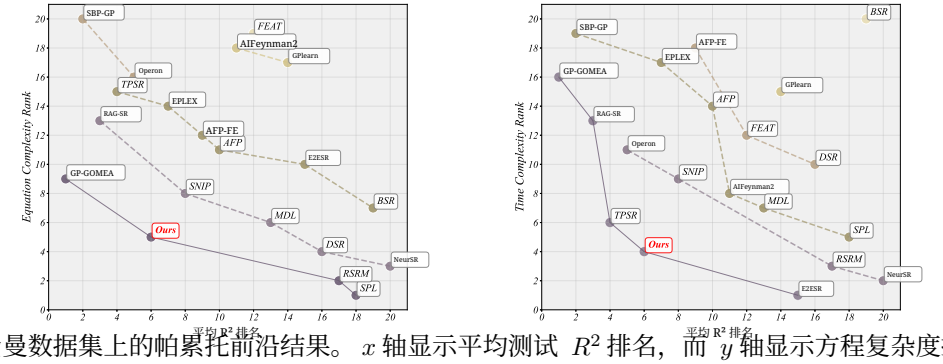


图 2：费曼数据集上的帕累托前沿结果。x 轴显示平均测试 R^2 排名，而 y 轴显示方程复杂度排名（左）和时间复杂度排名（右）。实线表示最优帕累托前沿，虚线从左下到右上表示排名较低的前沿。

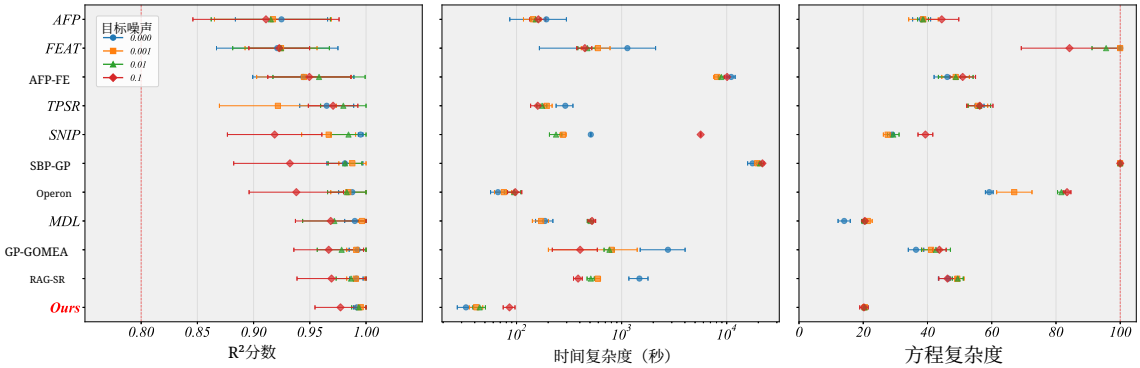


图 3：不同噪声水平下斯特罗加茨数据集上的比较。从左到右的子图分别报告 R^2得分、时间复杂度（秒）和方程复杂度。噪声水平以 蓝色圆圈 (0.000),橙色方块(0.001), 绿色三角形 (0.01)，以及 红色菱形 (0.1) 表示，误差条表示标准差。仅纳入在不同噪声设置下平均 超过 0.9 的方法。

帕累托性能分析。 如图2所示，我们的方法在准确率（R^2）、方程复杂度和时间复杂度之间实现了均衡优化，并始终位于秩1帕累托前沿，整体上优于相互竞争的基线方法。这一优势源于通过条件变分自编码器训练学习到的生成式连续潜在空间，该空间在保持全局结构连续性的同时确保了局部数值平滑性。基于这一表示，GenSR能够高效地实现分层优化

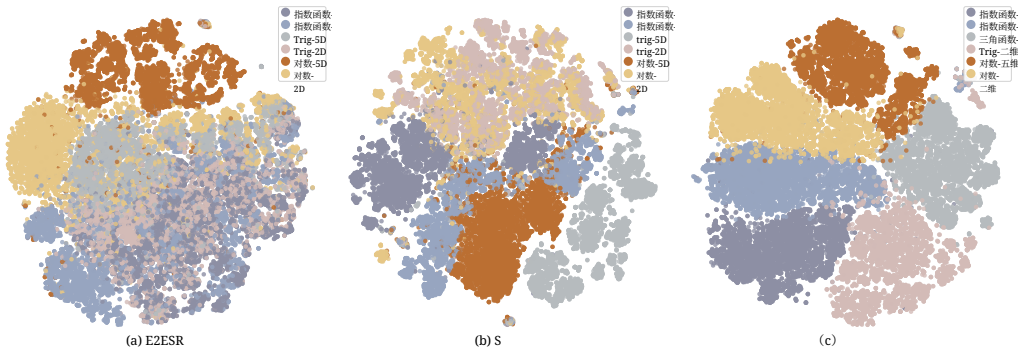
“地图构建 → 粗定位 → 精细搜索”范式。与依赖启发式变异和回溯的离散符号空间不同，我们组织良好的生成式潜在空间提供了稳定的方向信号，使搜索既更高效，又更不易过拟合。因此，GenSR在准确率、表达式简洁度和计算效率方面均取得了稳健提升。

噪声目标下的性能。 如图 3 所示，我们的方法在不同噪声水平下均实现了最高 R^2 ，且方差小于竞争基线方法，表明其在扰动下具有稳定且鲁棒的拟合性能。此外，该方法所需运行时间显著更少，并能生成复杂度更低的方程。这些结果突出表明，即使存在噪声，我们的生成式潜在空间仍能保持结构组织和局部平滑性，从而为搜索提供稳定的优化方向，避免离散符号空间中常见的随机振荡和低效探索。得益于这一设计，GenSR 在噪声条件下仍能在准确率、效率和紧凑性之间保持良好平衡，展现出强大的鲁棒性。

5.3 潜在空间的可视化比较

我们将 GenSR 与 E2ESR（Kamienny 等人，2022）和 SNIP（Meidani 等人）进行比较。E2ESR 使用 Transformer 预测目标方程，而 SNIP 利用对比预训练将符号形式与数值数据对齐。与 GenSR 不同，E2ESR 和 SNIP 的空间都是判别式的，而非生成式的。

潜在空间的结构可分性比较。如图 4 所示，不同方法会使方程潜向量呈现出不同的分布特征。E2ESR 的潜在空间无法有效区分函数类型和输入维度：尽管它能够部分区分对数算子的两种输入变体，但 Trig 和指数函数分量仍然高度纠缠，形成混合表示。SNIP 在算子层面实现了稍好的分离，但缺乏类内紧凑性；例如，exp-5D 簇分裂为两个不同区域，且不同类别之间存在明显重叠。相比之下，我们的 GenSR 呈现出清晰的结构化解耦潜在空间，不同函数类型和输入维度均得到良好分离。这一空间为 GenSR 提供了有利条件，使其能够
在确保解码稳定性的同时进行初始方程定位。



(a) E2ESR (b) S (c)

图4：E2ESR、SNIP 和 GenSR 的潜变量的二维 t-SNE 可视化。图例区分了六类，分别对应三个具有代表性的函数族中的方程，每个函数族均在二维和五维输入维度下进行评估，展示了所学习潜在空间的聚类行为。

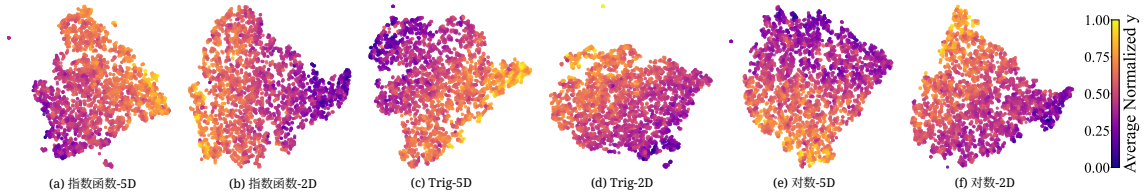


图 5：在二维和五维输入设置下，三个函数族（指数函数、三角函数、对数函数）中方程的 GenSR 潜变量二维 t-SNE 可视化，分别显示在子图 (a)–(f) 中。颜色表示归一化 y 值的平均值，如旁边的色条所示。

函数类别内的数值连续性。对于每个观测样本 $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$，我们使用 $\{y_i\}_{i=1}^N$ 的归一化均值来表示数值特征。图 5 可视化了 GenSR 的局部数值特征分布，其中每个子图展示了图 4 (c) 中一个簇的数值特征分布。对于每个结构簇，GenSR 呈现出平滑的数值特征分布，这使得搜索算法能够在定位高概率区域后，沿平滑的数值方向细化收敛。详细内容及其他潜在空间实验见附录 C。

6 结论与未来工作

我们提出了 GenSR，这是一种基于生成式潜在空间的符号回归框架，可将离散的符号方程空间映射到连续且结构良好的空间中。GenSR 首先通过双分支条件变分自编码器联合保证符号连续性和局部数值平滑性，随后通过粗定位和细粒度搜索实现高效的符号回归。该设计将组合式符号搜索与连续优化衔接起来，在准确率、复杂度和效率之间实现了均衡。 此外，GenSR 将不同的方程族组织到各自独立的潜在区域中，从而自然支持方程分类等任务；我们在附录中进一步展示了这一点。这些结果凸显了生成式潜在空间作为方程发现基础的广泛潜力，并且更广泛地说，也展现了其作为组合式科学搜索基础的潜力。未来的工作包括引入更丰富的生成先验和物理约束，并利用更先进的生成模型构建更强大、更具泛化能力的方程空间，从而推动面向科学发现、具备可解释性并能感知领域知识的机器学习发展。

7 致谢

本工作得到国家自然科学基金委员会（12572266）以及宁波东方理工大学和宁波数字孪生研究院高性能计算中心的支持。

可复现性声明

为确保结果的可复现性，我们在第 3.1 节、第 3.2 节和附录 F 中详细描述了我们的方法和实验设置。我们还通过实验（附录 G.5）确认了结果的鲁棒性。此外，为进一步便于复现，我们将发布代码以及已训练模型的检查点。

参考文献

Ignacio Arnaldo、Krzysztof Krawiec 和 Una-May O'Reilly。多元回归遗传编程。载于《2014 年遗传与进化计算年度会议论文集》，第 879–886 页，2014 年。

Luca Biggio、TommasoBendinelli、Alexander Neitz、Aurelien Lucchi 和 Giambattista Parascandolo。可扩展的神经符号回归。载于国际机器学习大会，第 936–945 页。PMLR，2021 年。

Bogdan Burlacu、Gabriel Kronberger 和 Michael Kommenda。Operon c++ ：一种高效的符号回归遗传编程框架。载于 2020 年遗传与进化计算大会伴随会议论文集，第 1562–1570 页，2020 年。

Cavalab。Srbench：符号回归的动态基准测试框架，2022 年。网址：https: // cavalab.org/srbench/results/#symbolically-verfied-solutions。访问日期：2024-09-26。

Tianqi Chen 和 Carlos Guestrin。Xgboost：一种可扩展的树提升系统。载于《第 22 届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集》，第 785–794 页，2016 年。

Yuntian Chen、Yingtao Luo、Qiang Liu、Hao Xu 和 Dongxiao Zhang。用于发现开放形式偏微分方程的符号遗传算法（SGA-PDE）。《Physical Review Research》，4(2):023174，2022 年。

Miles Cranmer。借助 PySR 和 symbolicregression.jl 实现科学领域的可解释机器学习。arXiv 预印本 arXiv:2305.01582，2023 年。

Fabricio Olivetti de Franca 和 Guilherme Seidyo Imai Aldeia。用于符号回归的交互—变换进化算法。《Evolutionary Computation》，29(3):367–390，2021 年。

Qihao Duan、Bingding Huang、Zhenqiao Song、Irina Lehmann、Lei Gu、Roland Eils 和 Benjamin Wild。JanusDNA：强大的双向混合 DNA 基础模型。arXiv 预印本 arXiv:2505.17257，2025 年。

陈峰、刘子权、支卓、Ilija Bogunovic、Carsten Gerner-Beuerle 和 Miguel Rodrigues。Prosac：针对对抗攻击下机器学习模型的可证明安全认证。发表于 AAAI人工智能会议论文集，第 39 卷，第 2933–2941 页，2025 年。陈峰、沈明和、Ananth Balashankar、Carsten Gerner-Beuerle 和 Miguel RD Rodrigues。虽有噪声，但仍有效：使用不完美评判者对大语言模型进行稳健统计评估。发表于第十四届国际学习表征会议，2026年。URL https://openreview.net/forum?id=hEhxreaLdU。

Yoav Freund 和 Robert E Schapire。《在线学习的决策理论泛化及其在提升法中的应用》。计算机与系统科学杂志，55(1):119–139，1997年。

Arya Grayeli、Atharva Sehgal、Omar Costilla Reyes、Miles Cranmer 和 Swarat Chaudhuri。《利用学习得到的概念库进行符号回归》。神经信息处理系统进展，37:44678–44709，2024年。Nikolaus Hansen。《CMA 进化策略：教程》。arXiv 预印本 arXiv:1604.00772，2016。Samuel Holt、Zhaozhi Qian 和 Mihaela van der Schaar。《深度生成式符号回归》。载于第十一届国际学习表征会议。Yuxiao Hu、Qian Li、Xiaodan Shi、Jinyue Yan 和 Yuntian Chen。《结合综合监测与再分析数据、融合领域知识的多空间多时间 PM2.5 预测》。《国际环境》，192:108997，2024年。Yuxiao Hu、Qian Li、Dongxiao Zhang、Jinyue Yan 和 Yuntian Chen。《上下文对齐：激活并增强大语言模型在时间序列中的能力》。arXiv 预印本 arXiv:2501.03747，2025 年。Tao Huang、Rui Wang、Xiaofei Liu、Yi Qin、Li Duan 和 Liping Jing。《通过证据不确定性量化检测大型视觉语言模型的错误行为》。arXiv 预印本 arXiv:2602.05535，2026年。Yanhao Jia、Ji Xie、S Jivaganesh、Hao Li、Xu Wu 和 Mengmi Zhang。《看见声音，听见景象：揭示人工智能模型在声音定位中的模态偏差与冲突》。arXiv 预印本 arXiv:2505.11217，2025 年。Ying Jin、Weilin Fu、Jian Kang、Jiadong Guo 和 Jian Guo。《贝叶斯符号回归》。AAAI，2020。Pierre-Alexandre Kamienny、St´ephane d'Ascoli、Guillaume Lample 和 Franc¸ois Charton。《基于变换器的端到端符号回归》。神经信息处理系统进展，35:10269–10281，2022年。Guolin Ke、Qi Meng、Thomas Finley、Taifeng Wang、Wei Chen、Weidong Ma、Qiwei Ye 和 Tie-Yan Liu。《LightGBM：一种高效的梯度提升决策树》。神经信息处理系统进展，30，2017年。Diederik P Kingma 和 Max Welling。《自动编码变分贝叶斯》。arXiv 预印本 arXiv:1312.6114，2013年。William La Cava、Lee Spector 和 Kourosh Danai。《用于回归的 Epsilon-lexicase 选择》。载于 2016 年遗传与进化计算会议论文集，第 741–748 页，2016 年。William La Cava、Tilak Raj Singh、James Taggart、Srinivas Suri 和 Jason H Moore。《通过演化树网络学习简洁的回归表征》。ICLR，2019。William La Cava、Bogdan Burlacu、Marco Virgolin、Michael Kommenda、Patryk Orzechowski、Fabr´ıcio Olivetti de Franc¸a、Ying Jin 和 Jason H Moore。《当代符号回归方法及其相对性能》。神经信息处理系统进展，2021(DB1):1，2021年。Guillaume Lample 和 Franc¸ois Charton。《用于符号数学的深度学习》。载于国际学习表征会议。Qian Li、Yuxiao Hu 和 Yang Cao。《基于动态阈值的自适应 Perona–Malik 模型：在保留细节的同时去除图像中的多种噪声》。《计算机与应用数学》，137:28–43，2023a。

Qian Li、Yuxiao Hu、Ye Liu、Dongxiao Zhang、Xin Jin 与 Yuntian Chen。离散逐点攻击是不够的：面向人脸识别的广义流形对抗攻击。载于 IEEE/CVF 计算机视觉与模式识别会议论文集，第 20575–20584 页，2023b。Qian Li、Yuxiao Hu、Yinpeng Dong、Dongxiao Zhang 与 Yuntian Chen。关注隐藏者：探索隐藏威胁以增强对抗训练。载于 IEEE/CVF 计算机视觉与模式识别会议论文集，第 24442–24451 页，2024。Wenqiang Li、Weijun Li、Linjun Sun、Min Wu、Lina Yu、Jingyi Liu、Yanjie Li 与 Songsong Tian。通过联合监督学习实现符号回归的基于Transformer的模型。载于第十一届国际学习表征会议，2022。Kazem Meidani、Parshin Shojaee、Chandan K Reddy 与 Amir Barati Farimani。SNIP：通过统一预训练连接数学符号领域与数值领域。载于第十二届国际学习表征会议。Sebastian Meˇznar、Saˇso Dˇzeroski 与 Ljupˇco Todorovski。面向符号回归的高效数学表达式生成器。《机器学习》，112(11)：4563–4596，2023 年。Randal S Olson、William La Cava、Patryk Orzechowski、Ryan J Urbanowicz 与 Jason H Moore。PMLB：用于机器学习评估与比较的大型基准测试套件。生物数据挖掘，10：1–13，2017。Brenden K Petersen、Mikel Landajuela Larma、Terrell N Mundhenk、Claudio Prata Santiago、Soo Kyung Kim 与 Joanne Taery Kim。深度符号回归：通过风险偏好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。Sergei Popov、Mikhail Lazarev、Vladislav Belavin、Denis Derkach 与 Andrey Ustyuzhanin。通过变分自编码器进行符号表达式生成。PeerJ计算机科学，9：e1241，2023 年。Steven J Rigatti。随机森林。《保险医学杂志》，47(1)：31–39，2017。Raymond Ros 与 Nikolaus Hansen。对 CMA-ES 的一种简单改进，以实现线性时间和空间复杂度。载于自然界并行问题求解国际会议，第 296–305 页。施普林格，2008。Michael D Schmidt 与 Hod Lipson。年龄 - 适应度帕累托优化。载于第 12 届遗传与进化计算年会论文集，第 543–544 页，2010。Parshin Shojaee、Kazem Meidani、Shashank Gupta、Amir Barati Farimani 与 Chandan K Reddy。LLM-SR：借助大语言模型编程进行科学方程发现。载于第十三届国际学习表征会议。Parshin Shojaee、Kazem Meidani、Amir Barati Farimani 与 Chandan Reddy。面向符号回归的基于 Transformer 的规划。神经信息处理系统进展，36：45907–45919，2023 年。Kihyuk Sohn、Honglak Lee 与 Xinchen Yan。利用深度条件生成模型学习结构化输出表征。神经信息处理系统进展，28，2015。Trevor Stephens 等人。gplearn：采用受 scikit-learn 启发的 API 的 Python 遗传编程。《燃气轮机与动力工程杂志》，30，2015。Steven H Strogatz。非线性动力学与混沌：及其在物理学、生物学、化学和工程学中的应用。CRC出版社，2018。

Fangzheng Sun、Yang Liu、Jian-Xun Wang 与 Hao Sun。符号物理学习器：通过蒙特卡洛树搜索发现控制方程。载于第十一届国际学习表征会议。Silviu-Marian Udrescu、Andrew Tan、Jiahai Feng、Orisvaldo Neto、Tailin Wu 与 Max Tegmark。AI Feynman 2.0：利用图模块化的帕累托最优符号回归。神经信息处理系统进展，33：4860–4871，2020。Marco Virgolin、Tanja Alderliesten 与 Peter AN Bosman。面向符号回归的、基于语义反向传播的遗传编程的外部与内部线性缩放。载于遗传与进化计算会议论文集，第 1084–1092 页，2019。Marco Virgolin、Tanja Alderliesten、Cees Witteveen 与 Peter AN Bosman。改进基于模型的遗传编程，用于小型表达式的符号回归。《Evolutionary Computation》，29(2)：211–237，2021。Hao Xu、Jinglong Lin、Dongxiao Zhang 与 Fanyang Mo。基于分位数几何增强图神经网络的色谱对映体分离保留时间预测。自然通讯，14(1)：3095，2023 年。Yilong Xu、Yang Liu 与 Hao Sun。RSRM：强化学习符号回归机。ICLR，2024。Zihan Yu、Jingtao Ding、Yong Li 与 Depeng Jin。通过MDLFormer引导搜索进行符号回归：从最小化预测误差到最小化描述长度。载于第十三届国际学习表征会议，2025 年。Hengzhe Zhang、Qi Chen、Wolfgang Banzhaf、Mengjie Zhang 等人。RAG-SR：面向神经符号回归的检索增强生成。载于第十三届国际学习表征会议，2025 年。

A 大语言模型（LLM）的使用

大语言模型（LLM）仅用作语言润色和轻微编辑的辅助工具。它们对研究构思、实验设计或写作的贡献程度均不足以将其视为贡献者。本文中提出的所有实质性研究思路、方法和结果均完全出自作者之手。

B $p(\boldsymbol{F}|\boldsymbol{X})$ 最大化的推导

我们参考 Sohn 等人（2015），使用变分分布 $q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F})$ 来近似难解分布 $p(\boldsymbol{F}|\boldsymbol{X})$；该变分分布可以是任意分布（在我们的工作中，假设其为多元独立高斯分布）。详细推导如下。

$$
\begin{aligned}
\log p(\boldsymbol{F}|\boldsymbol{X}) &= \int_{\boldsymbol{z}} q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F}) \log p(\boldsymbol{F}|\boldsymbol{X}) d\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F}) \log \frac{p(\boldsymbol{F}, \boldsymbol{z}|\boldsymbol{X})}{p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F})} d\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F}) \log \left(\frac{p(\boldsymbol{F}, \boldsymbol{z}|\boldsymbol{X})}{q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F})} \frac{q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F})}{p(\boldsymbol{z}|\boldsymbol{F}, \boldsymbol{X})} \right) d\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F}) \log \left(\frac{p(\boldsymbol{F}, \boldsymbol{z}|\boldsymbol{X})}{q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F})} \right) d\boldsymbol{z} + \int_{\boldsymbol{z}} q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F}) \log \left(\frac{q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F})}{p(\boldsymbol{z}|\boldsymbol{F}, \boldsymbol{X})} \right) d\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F}) \log \left(\frac{p(\boldsymbol{F}, \boldsymbol{z}|\boldsymbol{X})}{q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F})} \right) d\boldsymbol{z} + D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F}) || p(\boldsymbol{z}|\boldsymbol{F}, \boldsymbol{X})\right).
\end{aligned}
$$

由于 $D_{KL}(q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F})||p(\boldsymbol{z}|\boldsymbol{F}, \boldsymbol{X})) \geq 0$，因此，我们得到 $\log p(\boldsymbol{F}|\boldsymbol{X})$ 的下界：

$$
\begin{aligned}
\log p(\boldsymbol{F}|\boldsymbol{X}) &\geq \int_{\boldsymbol{z}} q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F}) \log \left(\frac{p(\boldsymbol{F}, \boldsymbol{z}|\boldsymbol{X})}{q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F})} \right) d\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F}) \log \left(\frac{p(\boldsymbol{F}|\boldsymbol{z}, \boldsymbol{X}) p(\boldsymbol{z}|\boldsymbol{X})}{q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F})} \right) d\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F}) \log p(\boldsymbol{F}|\boldsymbol{z}, \boldsymbol{X}) d\boldsymbol{z} + \int_{\boldsymbol{z}} q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F}) \log \left(\frac{p(\boldsymbol{z}|\boldsymbol{X})}{q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F})} \right) d\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F}) \log p(\boldsymbol{F}|\boldsymbol{z}, \boldsymbol{X}) d\boldsymbol{z} - D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F}) || p(\boldsymbol{z}|\boldsymbol{X})\right) \\
&= \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F})}\left[\log p(\boldsymbol{F}|\boldsymbol{X}, \boldsymbol{z})\right] - D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{F}) || p(\boldsymbol{z}|\boldsymbol{X})\right).
\end{aligned}
$$

为优化这一下界，GenSR 使用后验编码器 $q_{\phi,\ \varphi}(\boldsymbol{z}|\boldsymbol{X},\ \boldsymbol{F})$、先验编码器 $p_{\phi,\ \psi}(\boldsymbol{z}|\boldsymbol{X})$ 和解码器 $p_{\theta}(\boldsymbol{F}|\boldsymbol{X},\ \boldsymbol{z})$，分别对 $q(\boldsymbol{z}|\boldsymbol{X},\ \boldsymbol{F})$、$p(\boldsymbol{z}|\boldsymbol{X})$ 和 $p(\boldsymbol{F}|\boldsymbol{X},\ \boldsymbol{z})$ 进行参数化。其中，ϕ、φ 和 ψ 分别表示 Transformer 编码器、后验网络和先验网络的模型参数，而 θ 包含 Transformer 解码器和特征融合模块的参数。因此，使用重构损失和 KL 散度训练条件变分自编码器，等价于最大化该下界，即最大化条件分布 $p(\boldsymbol{F} \mid \boldsymbol{X})$。

C t-SNE 数据集构建与补充实验

C.1 t-SNE 数据集构建

在图 4 和图 5 中，我们构建了一个专用数据集，用于分析函数类别的可分性以及数值特征在潜在空间中的连续性。该数据集涵盖三类具有代表性的函数：指数函数、三角函数和对数函数。指数类别包括运算符 exp(·)，三角类别包括 sin(·)、cos(·) 和 tan(·)，而对数类别使用 log(·)。

为系统考察输入维度对潜在空间分布的影响，我们针对每类函数生成了两种情况：二维输入（$D = 2$）和五维输入（$D = 5$）。这种设置使我们能够验证，在低维和更高维输入下，潜在空间是否都能保持清晰的结构可分性和数值连续性。

表达式生成过程如下：首先指定变量 x 的维度，然后将基本运算符 $+$、$-$、$\times$、$\div$ 与各类别对应的函数运算符组合起来。为确保可比性，我们限制表达式长度，使词元数量（包括系数、运算符和变量）少于 25。

在规模方面，对于每种函数类别与维度组合，我们生成了 5000 个方程，并为每个方程采样 200 个数据点，以构建相应的数据集。这种设计提供了充分的统计覆盖度，并确保 t-SNE 可视化结果具有代表性。

总而言之，该数据集系统地覆盖了函数类别、输入维度、表达式结构和样本量，可作为验证潜在空间的全局结构可分性和局部数值特征连续性的坚实基础。

C.2 潜在空间中的插值结果

表 1：不同方法在潜在空间中两个表达式（1→2）之间的线性插值。对于每一对表达式，我们从 $\boldsymbol{z}(ratio) = (1 - ratio)\boldsymbol{z}_1 + ratio\,\boldsymbol{z}_2$ 进行解码，其中 $ratio \in \{0, 0.25, 0.5, 0.75, 1\}$。标记为红色的表达式在语法上无效。

ratio	Ours	E2ESR	SNIP
第 1 对： 表达式 1：c指数函数$(x) \cdot \sin(x - c) \rightarrow$ 表达式 2：$\log(x + c) \cdot \cos(x)$			
0	c指数函数(x) $\sin(x - c)$	c指数函数(x) $\sin(x - c)$	c指数函数(x) $\sin(x - c)$
0.25	c指数函数(x) 正弦$(x - c) + c$正弦(x)	无效	指数函数$(x + c)$ $\sin(x) + c\sin(x)$
0.5	c指数函数$(c \cdot c)$ $\sin(x) +$ 对数$(c \cdot x)$ $\cos(c)$	无效	指数函数(x) $\cos(x) + c\cos(x + c)$
0.75	指数函数$(c) +$ 对数$(x + c)$ $\cos(x)$	对数(x) $\sin(x) \cdot \cos(x)$	无效
1	对数$(x + c)$ $\cos(x)$	对数$(x + c)$ $\cos(x)$	对数$(x + c)$ $\cos(x)$
配对 2： 表达式 1：$x\cos(c\,x) + c \rightarrow$ 表达式 2：$x - \sin(x - c)$			
0	$x\cos(c\,x) + c$	x余弦$(c\,x) + c$	$x\cos(c\,x) + c$
0.25	x余弦$(c\,x) + c -$ 余弦(x)	无效	$x\cos(c) + c\sin(x)$
0.5	c余弦$(c \cdot x) + c -$ 余弦$(x - c)$	无效	无效
0.75	$cos(c \cdot x) - \sin(x - c)$	无效	无效
1	$x - \sin(x - c)$	$x - \sin(x - c)$	$x - \sin(x - c)$

我们通过潜在空间中的插值实验，比较生成式方法（我们的 GenSR）与判别式方法（如 SNIP 和 E2E）潜在空间的连续插值能力。对于每种方法，我们

首先将表达式 1 的数值样本输入其数值编码分支。在 GenSR 中，该分支对应于仅使用数值输入的先验分支；而在 E2ESR 和 SNIP 中，该分支分别由各自的数值编码器实现。将该分支的输出作为潜在表示 z_1。表达式 2 以相同方式处理，以获得 z_2。

随后，我们通过线性插值构造中间潜在点，

$$z(ratio) = (1 - ratio)z_1 + ratio\, z_2$$

并使用 $ratio \in \{0, 0.25, 0.5, 0.75, 1\}$。随后，将每个插值潜在向量 $z(ratio)$ 输入相应的解码器，以生成中间表达式。这一统一流程能够公平比较不同方法在潜空间插值下的结构平滑性和解码鲁棒性，并使我们能够评估其潜在表示是否支持方程之间的连续插值。

表 1 展示了潜在空间中两个表达式（1→2）之间的线性插值。得益于连续生成潜空间的设计，我们的方法在每个插值步骤中都能生成语法有效且有意义的表达式，同时保留方程在插值过程中的可解码性。相比之下，E2ESR 和 SNIP 经常产生语法无效的表达式（以红色标记），这表明它们的判别式潜在空间缺乏足够的句法正则化，并且常会退化为无法转换成有效前缀形式的结构。此外，我们的方法生成的变化十分平滑：表达式从 1 逐渐转变为 2，展现了潜在空间的结构连续性。相比之下，即使潜变量只有微小变化，E2ESR 和 SNIP 也会出现剧烈的结构变化，表明其潜在表示不够稳定。为清晰起见，所有常数均统一记为 c，以排除具体数值的影响。

D 后验分支与先验分支的重建性能

表 2：合成表达式数据集上的重构误差。我们报告表达式与其重构结果之间的**Levenshtein（编辑）距离**（均值 ± 标准差；越低越好）。考虑两种算子集合：**(i) LogExp** = {+, −, ×, ÷, 对数、指数函数}；**(ii)Trig** =LogExp ∪ {正弦、余弦、正切}。每个数据集包含 2k 个样本；名称中标出了最大表达式长度。

数据集（算子、长度、变量）	后验分支	先验分支
LogExp-15（变量 ≤ 3）	0.094 (±0.023)	0.106 (±0.031)
LogExp-20（变量 ≤ 5）	0.095 (±0.027)	0.112 (±0.029)
LogExp-25（变量 ≤ 5）	0.107 (±0.018)	0.120 (±0.022)
Trig-15（变量 ≤ 3）	0.115 (±0.032)	0.137 (±0.019)
Trig-20（变量 ≤ 5）	0.113 (±0.026)	0.142 (±0.043)
Trig-25 (变量 ≤ 5)	0.146 (±0.041)	0.171 (±0.038)

合成数据集构建。为系统评估我们的变分自动编码器的重构能力，我们在两种算子集合下构建了六个合成数据集：（i）**LogExp**，包含基本算术运算符 $\{+, -, \times, \div\}$ 以及对数和指数函数；（ii）**Trig**，在 LogExp 的基础上加入三角函数 {sin、cos、tan}。对于每个算子集合，我们分别生成三个数据集，其最大表达式长度为 15、20 和 25 个词元，并将变量数量分别限制为 3、5 和 5 个。该范围涵盖了略长于真实世界基准测试中常见长度的表达式（费曼数据集的平均长度为 13.56），从而能够在结构复杂度不断增加的情况下评估重建性能。每个数据集包含 2k 个随机合成样本。

我们使用 Levenshtein（编辑）距离量化重构准确性，该距离统计将重构表达式转换为真实表达式所需的最少插入、删除和替换次数。该指标直接衡量符号准确性，而非数值近似质量，因此非常适合用于评估精确的表达式重构。

结果分析。 表 2 报告了使用后验分支和先验分支的潜在编码进行重建的性能。我们总结出三个一致的观察结果：

• **后验分支的表现始终优于先验分支**。 这是预期之中的，因为后验分支通过直接最大化期望 $\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F})}[\text{对数 } p_\theta(\boldsymbol{F} \mid \boldsymbol{X}, \boldsymbol{z})]$ 进行训练，并以真实潜变量作为监督，因此编辑距离更小，重建也更准确。

• **先验分支的表现略逊一筹，但仍有竞争力**。 与后验分支不同，先验分支并未直接以最大化上述期望为目标进行训练，而是通过 KL 散度损失，使其输出分布与后验分布对齐，因此编辑距离略大。尽管如此，其重建质量仍相当不错。这一点至关重要，因为在推理过程中，我们只能从先验分布中采样。结果表明，KL损失能够有效地使先验分布与后验分布对齐，即使无法获取真实潜变量，也能实现准确重建。

• **表达式复杂度对重构的影响可控**。随着表达式长度增加并引入三角运算符，重构误差逐渐增大，但仍保持稳定。即使对于长度为 25 的表达式，后验分支仍能实现近乎精确的重构，这证明所学习的潜在空间保留了足够的结构信息。

总的来说，这些结果表明，所学习的潜在空间既具有良好的表达能力，又经过了良好的正则化：它能够根据后验样本忠实地重构方程，同时先验仍与其充分对齐，从而支持推理时可靠的生成。

E 消融研究

为了更好地理解 GenSR 的设计选择，我们针对潜在空间配置和 CMA-ES 搜索过程进行了消融实验。这些实验揭示了不同超参数对准确率、方程复杂度和运行时间的影响，并为我们选择稳健的默认设置提供了依据。

E.1 生成空间的潜在维度

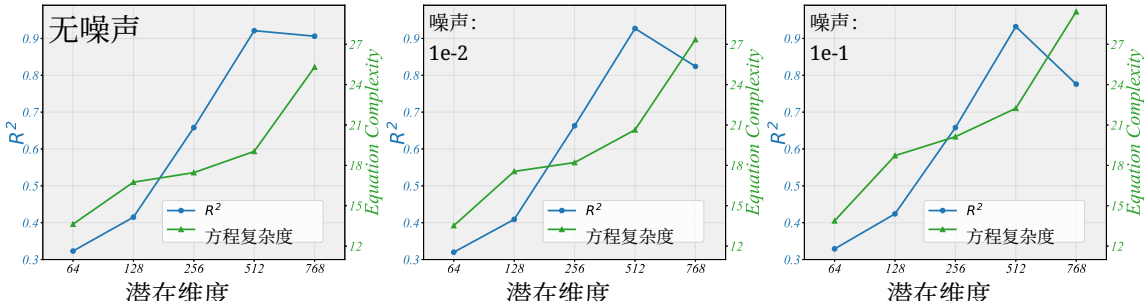


图 6：潜在维度对 GenSR 的影响。横轴表示预训练模型的潜在维度（64、128、256、512、768），左侧和右侧纵轴分别表示 R^2 和方程复杂度。结果分别在三种噪声水平下报告：无噪声、 1×10^{-2}和 1×10^{-1}。

图 6 展示了使用不同潜在维度对 GenSR 进行预训练的结果。为单独评估潜在空间本身，我们没有应用 CMA-ES 优化，而是直接比较预训练模型的输出。在无噪声设置（左侧子图）中，R^2 随潜在维度增加而稳步提升，在 512 维时达到最高点，随后在 768 维时略有下降。较小维度下，方程复杂度适度增长，但超过 512 维后则显著激增，这表明过大的潜在空间可能会导致表达式过于复杂。在噪声条件下，768 维模型的准确率下降更加明显，复杂度也进一步升高；随着噪声水平增加，性能退化也愈发严重。基于这些观察，我们在预训练时采用 512 维潜在空间，因为它实现了最佳权衡，并能确保在噪声下保持鲁棒性。

E.2 CMA-ES 超参数

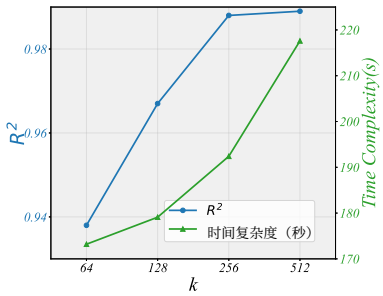


图 7： k 对 CMA-ES 性能的影响。

(1) 更新主维度的影响。在 CMA 算法中，我们采用对角近似，仅保留协方差矩阵的方差项，并进一步只更新方差最大的前 k 个主维度，以降低计算成本。图 7 展示了不同k 值的影响。我们观察到，随着 k 增大，R^2 不断提高，并在 $k = 256$ 时接近饱和，超过该维度后几乎没有进一步提升。相比之下，搜索时间近似线性增长

，随 k 增大而增长，导致较大的 k 会显著增加计算成本。为平衡准确率与效率，我们因此在后续所有实验中固定 $k = 256$，因为它能提供最佳的权衡。

(2) 其他超参数。我们进一步研究 CMA-ES 的三个内在超参数：种群规模 s、初始步长 t 和适应度权重 ω 在目标函数适应度 $= R^2 - \omega \cdot$复杂度中的作用。

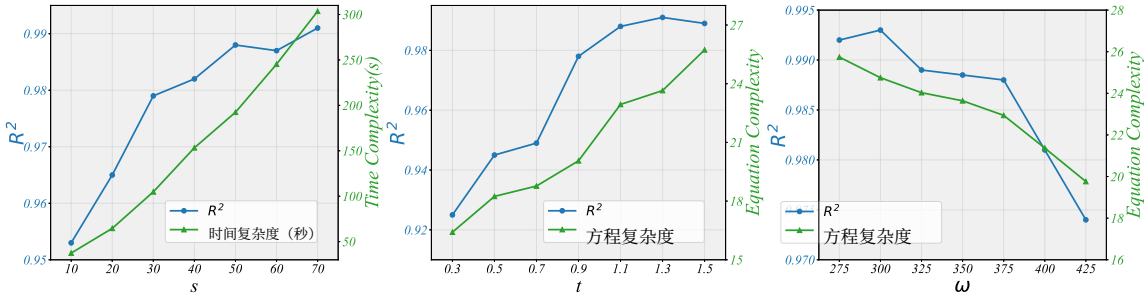


图 8：CMA-ES 超参数的消融结果。左：种群规模 s 对 R^2 和运行时间的影响。中：初始步长 t 对 R^2 和方程复杂度的影响。右：适应度函数中权重 ω 的影响。

种群规模 s。 </style}随着 s 增加， R^2 从 0.95 稳步提升至接近 0.99，但在 $s = 50$之后趋于饱和，这表明较大的种群规模能够增强全局探索，但收益会逐渐递减。同时，运行时间（即时间复杂度）几乎呈线性增长，从几十秒增至接近 300 秒，体现出较高的计算成本。因此，中等规模的种群在准确率与效率之间实现了更好的权衡。

初始步长 t。 </style}随着 t增加， R^2 先升后降，在 $t = 1.1$附近达到最佳表现，而方程复杂度则单调增长。较小的步长会将搜索限制在局部区域，而过大的步长则会导致收敛不稳定。因此，适中的 t 能更好地平衡准确率与复杂度。

适应度权重 ω。 </style}在适应度函数 Fitness $= R^2 - \omega \cdot$complexity 中， ω 决定了准确率与复杂度之间的权衡。较小的 ω 会侧重于最大化 R^2，但会产生过于复杂的方程；较大的 ω 则倾向于更简单的形式，但会牺牲准确率。适中的 ω 能够有效平衡这两个方面，避免过拟合和欠拟合。

总结。 总的来说，我们的消融实验表明，采用 512维潜在空间、 $k = 256$ 进行协方差更新，并将 s、 t 和 ω 设置为适中水平，可获得最佳性能。这些选择始终能在准确率、复杂度和运行时间之间实现稳健的权衡。

F 模型架构与训练配置

F.1 模型架构

如图 1 所示，GenSR 由两个输入嵌入器（分别用于数值和方程）、一个共享的 Transformer 编码器、后验与先验多层感知机、一个特征融合线性层以及一个 Transformer 解码器组成。

数值输入 $\{(\boldsymbol{x}, y)_i\}_{i=1}^m \in \mathbb{R}^{m\times(D+1)}$ 首先被分词为 $\{(\boldsymbol{x}, \boldsymbol{y})_i\}_{i=1}^m \in \mathbb{R}^{m\times 3(D+1)}$，随后由数值嵌入器映射到 $\boldsymbol{X}_e \in \mathbb{R}^{m\times 3(D+1)\times d_n}$，其中 $d_n = 64$表示数值嵌入维度。经过重塑与线性投影后得到最终的 $\boldsymbol{X} \in \mathbb{R}^{m\times d}$。类似地，方程表达式被填充至最大序列长度 m，并以 $d = 512$ 嵌入到 $\boldsymbol{F} \in \mathbb{R}^{m\times d}$。

对于后验分支，将 $\boldsymbol{X}$ 和 $\boldsymbol{F}$ 拼接后输入一个 8 层 Transformer 编码器（8 个注意力头，512 个隐藏单元），再通过一个多层感知机生成后验分布 $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\sigma}_1^2)$。对于先验分支，仅将 $\boldsymbol{X}$（应用掩码后）输入同一个编码器，以生成先验分布$\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\sigma}_2^2)$。值得注意的是，数值输入 $\boldsymbol{X}$ 不包含位置编码，以确保模型仅对函数关系而非顺序伪影进行建模。后验分布与先验分布之间的 KL 散度作为正则化项:

$$\mathcal{L}_{\mathrm{KL}} = D_{\mathrm{KL}}\big(\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\sigma}_1^2) \,\|\, \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\sigma}_2^2)\big)\,.$$

训练期间，从后验分布 $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\sigma}_1^2)$中抽取 m 个样本以构成 $\boldsymbol{Z}_1 \in \mathbb{R}^{m\times d}$，然后通过特征融合线性层处理，并由一个对称的 8 层 Transformer 解码器（8 个注意力头，512 个隐藏单元）进行解码。解码器生成符号表达式的 logits，并通过重构损失与真实方程进行比较:

$$\mathcal{L}_{\mathrm{CE}} = -\sum_{i=1}^{m} \log p_\theta(f_i \mid \boldsymbol{z}_1^i),$$

其中，f_i 表示目标方程的第 i 个词元，θ 表示解码器参数。
整体训练目标由这两项组成:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda\, \mathcal{L}_{\mathrm{KL}},$$

其中，λ是一个平衡超参数。我们采用 KL 退火，在训练过程中逐渐将 λ从 0 增至其完整值。这可以防止 KL 项在训练早期主导优化，并缓解后验坍塌，同时确保后期潜在空间具有良好的结构。

在推理阶段，我们使用先验分布 $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\sigma}_2^2)$ 形成潜变量 $\boldsymbol{Z}_2 \in \mathbb{R}^{m\times d}$，然后将其输入 Transformer 解码器以生成候选方程。我们会对每个方程进行评估并得到适应度分数，随后利用该分数通过 CMA-ES优化过程迭代改进搜索，从而获得最终预测方程。

F.2 训练配置

我们采用 Adam 优化器，批量大小为 256，并训练模型 200 个训练轮次，每轮包含 1000 步。学习率设为 1.0×10^{-3}，并遵循 Noam 调度，预热步数为 $w = 8000$。该调度使学习率在预热期间线性增加，随后按照步数的平方根倒数成比例衰减。对于 KL 散度权重 λ，我们采用退火策略：在前 50% 的训练步数中，将 λ 从 0 线性增加至 1.0，此后保持不变。这可以防止 KL 项在训练初期主导优化，并缓解后验坍塌，同时确保模型在后期形成结构化的潜在空间。模型包含 1.621 亿个可训练参数。所有实验均使用两块 NVIDIA H800 GPU（Hopper 架构，每块显存为 80GB）进行，每个训练轮次约需一小时。

G 详细实验结果

G.1 评估数据集详细说明

在实验评估中，我们采用 SRBench 基准（La Cava 等人，2021），这是一个广泛使用且具有挑战性的符号回归数据集集合。SRBench 包含三个具有代表性的数据集组：费曼数据集、斯特罗加茨数据集和黑盒数据集。这些数据集在输入维度、样本量、函数特性和任务难度方面各不相同，共同构成了一个全面的符号回归方法评估框架，并为比较提供了标准化平台。

费曼数据集。费曼数据集（Udrescu 等人，2020）源自《费曼物理学讲义》，包含 119 个方程，涵盖力学、电磁学和量子物理学等多个领域。这些方程在结构上丰富多样，并以真实的物理定律为基础，因此是评估符号回归在科学发现中表现的重要基准。为确保各项任务的一致性，该数据集将输入维度限制为 $D \leq 10$。其一项独特优势是底层真实函数完全已知，从而消除了黑盒任务中存在的歧义。该数据集累计约有 10^5 个数据点，可用于评估大规模回归准确率，以及模型对多种物理定律的泛化能力。

斯特罗加茨数据集。斯特罗加茨数据集（Strogatz，2018）源自非线性动力系统，灵感来自经典的 ODE-Strogatz 集合。每项任务都对应一个二维动力系统建模问题，输入维度固定为 $D = 2$。与侧重物理可解释性的费曼数据集不同，斯特罗加茨数据集关注动态行为，尤其适用于测试方法处理非线性和时间依赖关系的能力。每个子数据集约包含 $N = 400$ 个样本，并提供底层真实函数，以便在定义明确的动力学框架内进行评估。因此，该数据集对于在小样本、高非线性条件下评测性能至关重要。

黑盒数据集。 黑盒数据集（Olson 等人，2017）凸显了符号回归在复杂现实场景中的适用性。该数据集组主要来源于 PMLB 仓库，并已广泛应用于 SRBench。与费曼数据集和斯特罗加茨数据集不同，该数据集中的目标函数未知，从而模拟现实中的回归任务。这些数据集通常包含更高水平的噪声和不确定性，因此可有效检验鲁棒性。为确保可比性，输入维度限制为 $D \leq 10$。该数据集集合包含 57 个子数据集，涵盖现实案例和合成案例，数据集规模从几十个样本到数万个样本不等。这种广泛的差异不仅检验算法的准确率，也考察其在噪声条件下的稳定性和泛化能力。

总之，费曼数据集强调物理可解释性，斯特罗加茨数据集体现非线性动力系统的复杂度，而黑盒数据集则着重考察现实场景下不确定性和噪声条件下的鲁棒性。这三个数据集共同构成了一个互补且多元的符号回归评估环境。在这一基准测试中，我们能够全面考察准确率、效率和复杂度之间的权衡，同时展现所提出方法在多种任务中的鲁棒性和通用性。

G.2 评估指标

为评估符号回归方法的性能，我们考虑三个关键指标，分别衡量准确率、效率和可解释性：

• **决定系数（R^2）：** 该指标量化所发现的符号表达式对目标数据方差的解释程度。其形式化定义为:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{2}$$

其中，y_i 表示真实值，$\hat{y}_i$ 表示预测值，$\bar{y}$ 表示真实值的均值。较高的 R^2 表示预测准确性更好，其中 $R^2 = 1$ 代表完美拟合，而 $R^2 = 0$ 则表示未能优于均值预测器。

• **时间复杂度：** 该指标衡量算法完成符号回归任务并输出最终表达式所需的实际运行时间，以秒（s）为单位。它反映了该方法的计算效率，搜索时间越短，表明收敛越快、资源消耗越少。

• **方程复杂度：** 该指标评估所发现表达式的可解释性和紧凑性。我们将复杂度定义为最终方程中的词元总数:

$$\text{Complexity}(f) = \sum_{j=1}^{m} \mathbb{I}(t_j), \tag{3}$$

其中，f 表示所发现的方程，$\{t_j\}_{j=1}^{m}$ 表示其词元（包括系数、运算符和变量），而 $\mathbb{I}$（·）是一个指示函数，用于统计每个词元是否存在。复杂度越低，表达式越简单、越易于人类理解，也越不容易过拟合，并且越接近底层的物理或数学定律。

G.3 基线方法

基于进化算法的符号回归。 我们纳入了若干种遗传编程变体：GPlearn（Stephens 等人，2015），一种采用 scikit-learn 风格的遗传编程框架；AFP（Schmidt 和 Lipson，2010），通过年龄-适应度帕累托优化保持多样性；ITEA（de Franca 和 Aldeia，2021），利用交互-变换算子；EPLEX（La Cava 等人，2016），基于 ε-lexicase选择；MRGP（Arnaldo 等人，2014），结合多元回归与遗传编程；SBP-GP（Virgolin 等人，2019），采用语义反向传播和线性缩放；GP-GOMEA（Virgolin 等人，2021），一种采用高效连锁学习的基于模型的遗传编程方法；Operon（Burlacu 等人，2020），一个高性能的 C++SR 框架；以及 FEAT（La Cava 等人，2019），通过进化树网络来学习简洁的回归模型。

基于深度/神经网络及强化学习的符号回归。 我们进一步考虑基于神经网络和强化学习的方法：SPL（Sun 等人），采用蒙特卡洛树搜索进行物理发现；DSR（Petersen 等人），一种采用风险偏好策略梯度的强化学习方法；RSRM（Xu 等人，2024 年），一种强化学习符号回归机；TPSR（Shojaee 等人，2023 年），将符号回归表述为基于 Transformer 的规划问题；E2ESR（Kamienny 等人，2022），一种端到端Transformer框架；SNIP（Meidani 等人），统一符号预训练与数值预训练；RAG-SR（Zhang 等人，2025 年），利用检索增强生成；MDLformer（Yu 等人，2025 年），以最小描述长度原理为指导；以及 NeurSR（Biggio 等人，2021），一种可扩展的神经符号回归方法。

受物理学和概率论启发的方法。 我们还评估了 AI Feynman（Udrescu 等人，2020），该方法利用物理学先验和图模块化进行符号发现，以及 BSR（Jin 等人，2020），一种提供不确定性估计的贝叶斯符号回归方法。

传统机器学习基线方法。 最后，我们将与强大的非符号基线方法进行比较，包括基于树的集成方法：XGBoost（Chen 与 Guestrin，2016）、LightGBM（Ke 等人，2017）、AdaBoost（Freund 与 Schapire，1997年）和随机森林（Rigatti，2017）。

G.4 函数族分类任务

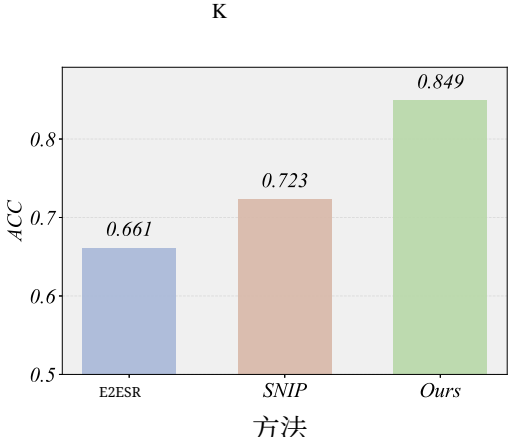


图 9：不同方法在分类任务上的准确率比较。GenSR（我们的方法）取得最高准确率（0.849），显著优于 E2ESR（0.661）和 SNIP（0.723）。

为了进一步考察所学习的生成式潜在空间在符号回归之外的迁移能力，我们设计了一个函数族分类任务。本实验旨在评估 GenSR 学到的潜在空间表征是否不仅捕捉结构连续性和数值平滑性，还能捕捉支持下游分类的判别特征。

数据集构建。 我们合成了一个涵盖五种代表性函数族的数据集：指数函数、对数函数、正弦函数、余弦函数和正切函数。为增加多样性和难度，每种函数族均采用四种不同的输入维度（2D、3D、4D 和 5D），总计形成 $5 \times 4 = 20$ 个类别。对于每个类别，我们随机生成 2000 个样本，每个样本由输入点及其对应的函数值构成。数据集按照 3:1:1 的比例划分为训练集、验证集和测试集，以确保评估公平且稳定。

模型与训练设置。 我们采用在符号回归任务上预训练的 GenSR 编码器，该编码器可生成 512 维潜在向量。在编码器之上，我们添加了一个轻量级分类头，由两个全连接层组成：第一层将潜向量映射为具有 ReLU 激活的隐藏表示，第二层输出一个 20 维 logits 向量，并通过 softmax 层获得类别概率。我们使用交叉熵损失和 Adam 优化器对整个网络进行微调，学习率为 1×10^{-3}，批量大小为 128。训练最多进行 100 个训练轮次；如果验证集上的表现连续五个训练轮次没有提升，则提前停止。测试时，选择概率最高的类别作为预测结果，并报告总体准确率。

比较与结果。 为进行公平比较，我们对 E2ESR（Kamienny 等人，2022）和 SNIP（Meidani 等人）的编码器采用相同的分类头设计、数据划分和训练流程。如图 9 所示，GenSR 的分类准确率显著高于两种基线方法。该结果表明，GenSR 学习到的潜在空间不仅具有生成性且结构连续，而且具有判别性与可泛化性，为符号回归之外的下游任务奠定了坚实基础。换言之，GenSR 产生的表征可同时用于生成、搜索和分类，凸显了其在更广泛应用中的潜力。

未来，由条件变分自编码器获得的潜在空间还可应用于多种领域，例如生物学情境（Duan 等人，2025）、心理学研究（Jia 等人，2025）、时空模式发挥关键作用的能源系统与气象预报（Hu 等人，2024；2025 年），以及可解释性与鲁棒性尤为重要的更广泛人工智能领域（Li 等人，2024；2023b；a；Huang 等人，2026；Feng 等人，2026年；2025 年）。

G.5 完整数值结果

表4、表5、表6和表7报告了不同方法在 Strogatz（Strogatz，2018）和费曼数据集（Udrescu 等人，2020）上，在不同噪声水平下的性能；表3则总结了在黑箱数据集（Olson 等人，2017）上的结果。在所有这些基准测试中，我们的方法在三项关键指标——准确率（R^2）、搜索效率（时间）和方程复杂度——之间始终表现出良好的平衡。值得注意的是，即使在噪声条件下，我们的方法仍能保持较高的 R^2 得分且方差较小，运行时间更短，并得出更简单的方程，这凸显了其鲁棒性以及相较于现有基线方法的整体优势。

表3：黑盒数据集的完整实验结果。

方法	测试 R^2 ↑	复杂度 ↓	时间（秒）↓
Operon	0.7945	65.69	2974
GP-GOMEA	0.7381	30.27	9636
MDL	0.6258	29.88	541.7
DSR	0.5625	9.465	36852
SBP-GP	0.7869	634	149344
FEAT	0.7621	82.49	6432
EPLEX	0.7372	53.14	15796
AFP-FE	0.6400	36.04	6184
AFP	0.6333	34.89	6033
GPlearn	0.5390	19.06	24254
线性	0.4437	17.4	0.2447
NeurSR	0.1228	13.33	11.7
XGB	0.7496	20186	236
AdaBoost	0.6939	9481	65.12
LGBM	0.6410	5734	29.9
ITEA	0.6295	116.7	12183
SNIP	0.3335	38.91	3.286
BSR	0.2725	22.52	59822
随机森林	0.6615	1517178	120.4
核岭回归	0.5952	1824	39.19
FFX	0.5575	1562	244.3
E2ESR	0.3612	61.09	7.101
MRGP	0.5300	10802	165007
MLP	0.5238	3882	30.49
AIFeynman2	0.2110	2240	86854
SPL	0.5472	12.96	2047
RSRM	0.3324	9.23	1752
Ours	0.8416	35.05	2725

表 4：噪声水平为 $\epsilon = 0.0$ 时的完整实验结果。结果包括费曼和斯特罗加茨数据集上的测试集 R^2、搜索时间和公式复杂度。

方法	斯特罗加茨数据集（$\epsilon = 0.0$）			费曼数据集（$\epsilon = 0.0$）		
	R^2↑	时间（秒）↓	复杂度↓	R^2↑	时间 (s)↓	复杂度↓
FEAT	0.9210(±0.054)	1135(±970)	119(±16)	0.9190(±0.014)	2417(±1737)	205.3(±13)
NeurSR	0.5206(±0.029)	14.82(±1.7)	11.3(±0.51)	0.3958(±0.013)	24.18(±2.2)	13.28(±0.15)
E2ESR	0.5341(±0.043)	3.729(±2)	32.49(±2.9)	0.8570(±0.013)	4.062(±1.3)	36.02(±0.88)
SNIP	0.9952(±0.002)	509(±21)	29(±1.2)	0.985(±0.004)	1203.5(±145)	31.71(±1.8)
GPlearn	0.7689(±0.067)	1195(±592)	28.96(±2.8)	0.8809(±0.027)	3900(±1052)	72.43(±29)
AFP	0.9248(±0.041)	192.2(±106)	38.17(±2.8)	0.9590(±0.005)	3655(±1999)	36.87(±2.2)
AFP-FE	0.9442(±0.045)	11041(±14277)	46.16(±4.1)	0.9806(±0.007)	17817(±530)	39.97(±1.9)
EPLEX	0.8125(±0.065)	548.2(±258)	50.09(±2.7)	0.9869(±0.006)	12771(±4998)	52.95(±1.3)
SBP-GP	0.9812(±0.016)	17591(±1765)	712.2(±39)	0.9945(±0.001)	28901(±37)	489.4(±16)
GP-GOMEA	0.9925(±0.009)	2760(±1258)	36.43(±2.4)	0.9956(±0.003)	5030(±967)	34.57(±1.5)
Operon	0.9878(±0.023)	66.58(±10)	59.23(±2.1)	0.9889(±0.006)	2174(±373)	69.88(±1.8)
SPL	0.7390(±0.047)	322.1(±180)	14.55(±2.5)	0.7073(±0.011)	209(±145)	12.88(±0.57)
DSR	0.7602(±0.086)	1858(±2617)	15.6(±1.7)	0.8441(±0.091)	1733(±3105)	14.86(±1)
RSRM	0.5501(±0.103)	121.9(±36)	13.09(±2.3)	0.8003(±0.013)	116.3(±31)	13.17(±0.47)
AIFeynman2	0.6459(±0.039)	762.1(±424)	22.26(±1.7)	0.9314(±0.016)	854.3(±24)	124.5(±16)
BSR	0.8455(±0.044)	31380(±23952)	38.98(±5.4)	0.6609(±0.018)	29065(±765)	25.5(±0.23)
MDL	0.9900(±0.009)	186.6(±35)	14.07(±1.9)	0.9171(±0.005)	467.3(±415)	23.4(±1.2)
TPSR	0.965(±0.024)	291(±53)	56.2(±1.4)	0.9921(±0.002)	236(±39)	57.27(±1.9)
RAG-SR	0.9914(±0.006)	1477(±305)	46.5(±3.1)	0.9926(±0.002)	3156(±305)	46.51(±1.4)
Ours	0.9918(±0.003)	33(±5.7)	20.53(±0.9)	0.9872(±0.003)	192.3(±21)	22.94(±0.5)

表 5：噪声水平为 $\epsilon = 0.001$ 时的完整实验结果。结果包括费曼和斯特罗加茨数据集中不同方法的测试集 R^2、搜索时间和公式复杂度。

方法	斯特罗加茨数据集（$\epsilon = 0.001$）			费曼数据集（$\epsilon = 0.001$）		
	R^2↑	时间（秒）↓	复杂度↓	R^2↑	时间 (s)↓	复杂度↓
FEAT	0.9244(±0.032)	594.4(±181)	106.7(±15)	0.9207(±0.006)	1726(±242)	196.5(±12)
NeurSR	0.5219(±0.031)	15.07(±1.7)	11.41(±0.31)	0.3979(±0.013)	24.51(±1.7)	13.24(±0.13)
E2ESR	0.5105(±0.060)	3.436(±0.75)	33.83(±4.4)	0.8585(±0.010)	3.894(±0.98)	35.85(±1.5)
SNIP	0.9667(±0.024)	277.9(±18.4)	27.57(±1.3)	0.9904(±0.002)	1365.07(±104)	31.43(±0.23)
GPlearn	0.7955(±0.067)	913.3(±121)	29.59(±2.5)	0.8902(±0.008)	3316(±540)	60.49(±12)
AFP	0.9172(±0.052)	143.4(±27)	38.75(±4.6)	0.9606(±0.006)	3711(±457)	39.33(±1.6)
AFP-FE	0.9447(±0.042)	8108(±584)	48.74(±3)	0.9805(±0.007)	26160(±157)	46.47(±1.2)
EPLEX	0.8488(±0.053)	416.1(±81)	49.26(±4.7)	0.9866(±0.007)	12341(±436)	56.03(±1.3)
SBP-GP	0.9879(±0.015)	19596(±1233)	820.5(±41)	0.9953(±0.001)	28940(±20)	574.4(±13)
GP-GOMEA	0.9914(±0.009)	804(±603)	41.14(±2.9)	0.9962(±0.001)	2904(±146)	45.23(±0.71)
Operon	0.9843(±0.036)	75.68(±14)	67.03(±5.5)	0.9916(±0.005)	2195(±404)	69.67(±1.6)
SPL	0.7526(±0.049)	358.8(±211)	14.4(±2.5)	0.7073(±0.016)	275.5(±206)	13.15(±0.44)
DSR	0.8067(±0.048)	500.8(±317)	18.66(±2.2)	0.8764(±0.003)	830.1(±282)	16.04(±0.31)
RSRM	0.5447(±0.105)	129.7(±8.1)	12.23(±0.65)	0.8104(±0.025)	128.2(±3.8)	13.04(±0.43)
AIFeynman2	0.6855(±0.091)	84.19(±74)	25.64(±3)	0.9177(±0.008)	638(±21)	130.6(±17)
BSR	0.8224(±0.121)	24299(±3478)	37.68(±2.2)	0.6538(±0.023)	30255(±4770)	25.85(±0.57)
MDL	0.9965(±0.004)	171.5(±30)	21.38(±1.4)	0.9079(±0.008)	428.9(±261)	32.35(±1.1)
TPSR	0.9216(±0.052)	193.22(±24.8)	55.64(±3.1)	0.992(±0.001)	190.2(±7.9)	59.75(±2.6)
RAG-SR	0.9908(±0.008)	591.4(±38.4)	49.29(±2.1)	0.9917(±0.003)	3590(±235)	49.42(±1.1)
Ours	0.9951(±0.007)	41.56(±6.1)	20.45(±0.8)	0.9883(±0.001)	279.31(±32.6)	23.5(±0.7)

表 6：噪声水平 $\epsilon = 0.01$ 下的完整实验结果。结果包括费曼数据集和斯特罗加茨数据集中不同方法的测试集 R^2、搜索时间和公式复杂度。

方法	斯特罗加茨数据集（$\epsilon = 0.01$）			费曼数据集（$\epsilon = 0.01$）		
	R^2↑	时间 (s)↓	复数↓	R^2↑	时间（秒）↓	复数↓
FEAT	0.9244(±0.043)	472.9(±90)	95.61(±16)	0.9212(±0.010)	1464(±365)	167.1(±6.5)
NeurSR	0.5179(±0.042)	15.48(±1.4)	11.63(±0.34)	0.3942(±0.011)	24.62(±1.7)	13.26(±0.12)
E2ESR	0.5031(±0.034)	3.392(±0.72)	35.94(±1.8)	0.8345(±0.007)	4.274(±0.58)	40.07(±0.90)
SNIP	0.9844(±0.025)	237.61(±32)	29.35(±1.8)	0.987(±0.007)	1210.1(±154)	32.62(±1.3)
GPlearn	0.7956(±0.059)	907.4(±110)	30.59(±4.3)	0.8890(±0.009)	3351(±437)	60.07(±19)
AFP	0.9153(±0.053)	152.2(±15)	38.62(±7.1)	0.9610(±0.005)	4090(±758)	40.86(±1.1)
AFP-FE	0.9582(±0.041)	8898(±579)	48.8(±5.4)	0.9819(±0.008)	27763(±297)	46.92(±2.3)
EPLEX	0.8562(±0.040)	437.6(±64)	53.07(±3.8)	0.9910(±0.002)	11043(±718)	54(±0.68)
SBP-GP	0.9813(±0.015)	20783(±776)	850.9(±34)	0.9950(±0.001)	28954(±15)	595.5(±12)
GP-GOMEA	0.9783(±0.029)	765.6(±1005)	42.64(±4.5)	0.9967(±0.001)	3020(±360)	44.67(±1.1)
Operon	0.9829(±0.031)	94.92(±15)	81.68(±1.2)	0.9878(±0.010)	3165(±549)	87.96(±1.3)
SPL	0.7388(±0.060)	413.3(±295)	14.71(±1.8)	0.7133(±0.007)	295.1(±175)	13.43(±0.58)
DSR	0.8199(±0.055)	492.9(±287)	18.51(±1.2)	0.8782(±0.004)	929.8(±422)	16.2(±0.41)
RSRM	0.5969(±0.077)	142.2(±21)	14.22(±1.5)	0.8092(±0.015)	131.6(±14)	12.97(±0.34)
AIFeynman2	0.7753(±0.047)	85.17(±75)	32.41(±4.1)	0.8732(±0.021)	629.4(±5.9)	155.2(±8)
BSR	0.8127(±0.070)	23622(±554)	38.74(±2.6)	0.6734(±0.018)	30411(±4711)	28.03(±0.49)
MDL	0.9718(±0.057)	505.1(±34)	20.31(±0.81)	0.9140(±0.009)	844.3(±561)	31.15(±1.5)
TPSR	0.9798(±0.028)	175.09(±19)	56.14(±3.5)	0.9911(±0.004)	217.06(±34)	64.27(±2.6)
RAG-SR	0.9867(±0.019)	509(±40)	49.43(±1.7)	0.9905(±0.007)	3241(±417)	72.41(±2.2)
Ours	0.9936(±0.007)	44.58(±24)	20.42(±0.6)	0.9872(±0.002)	486.83(±48)	23.36(±0.7)

表 7：噪声水平为 $\epsilon = 0.1$ 时的完整实验结果。结果包括费曼数据集和斯特罗加茨数据集上不同方法的测试集 R^2、搜索时间和公式复杂度。

方法	斯特罗加茨数据集（$\epsilon = 0.1$）			费曼数据集（$\epsilon = 0.1$）		
	R^2↑	时间（秒）↓	复杂度↓	R^2↑	时间（秒）↓	复杂度↓
FEAT	0.9228(±0.027)	446.9(±73)	84.2(±15)	0.9195(±0.006)	777.9(±102)	99.48(±5.6)
NeurSR	0.5054(±0.058)	17.47(±1.6)	12.74(±0.37)	0.3823(±0.020)	25.81(±1.7)	13.54(±0.16)
E2ESR	0.5152(±0.038)	5.621(±4.9)	38.49(±2.7)	0.7714(±0.014)	6.183(±0.83)	44.09(±0.61)
SNIP	0.9187(±0.042)	5650.7(±35)	39.35(±2.3)	0.9917(±0.020)	6514.8(±821)	37.71(±2.4)
GPlearn	0.8228(±0.052)	894.6(±108)	25.84(±2.9)	0.8911(±0.007)	2938(±543)	48.83(±9.5)
AFP	0.9110(±0.065)	161.4(±28)	44.44(±5.3)	0.9577(±0.007)	3886(±341)	40.79(±1.5)
AFP-FE	0.9496(±0.037)	10082(±565)	50.96(±4)	0.9826(±0.005)	28812(±0.51)	48.87(±1.4)
EPLEX	0.8822(±0.078)	405.8(±43)	53.46(±2.3)	0.9901(±0.004)	10283(±532)	45.62(±1.3)
SBP-GP	0.9323(±0.050)	21886(±782)	901.2(±34)	0.9905(±0.007)	28932(±83)	621.9(±6.6)
GP-GOMEA	0.9668(±0.031)	402.9(±802)	43.71(±2.1)	0.9957(±0.003)	3186(±454)	46.43(±0.91)
Operon	0.9380(±0.042)	97.13(±15)	83.44(±1.2)	0.9847(±0.008)	3090(±330)	89.23(±0.65)
SPL	0.7715(±0.044)	355.3(±59)	13.91(±1.6)	0.7109(±0.013)	270(±168)	13.54(±0.50)
DSR	0.8086(±0.047)	500(±300)	18.51(±2.2)	0.8779(±0.002)	814.9(±197)	16.03(±0.46)
RSRM	0.5553(±0.057)	138.3(±6.4)	13.54(±1.1)	0.8104(±0.016)	133.8(±6)	12.81(±0.41)
AIFeynman2	0.3170(±0.082)	65.97(±19)	23.53(±4.6)	0.2248(±nan)	710.7(±nan)	176.6(±nan)
BSR	0.7190(±0.076)	23292(±510)	49.54(±4.9)	0.6567(±0.024)	32497(±7914)	28.77(±1.1)
MDL	0.9686(±0.057)	522(±43)	20.5(±0.62)	0.9097(±0.009)	962.9(±918)	30.86(±1.4)
TPSR	0.9707(±0.022)	159.1(±23)	56.28(±4.1)	0.9836(±0.005)	184.01(±8.3)	66.96(±1.9)
RAG-SR	0.9693(±0.043)	385.3(±37)	46.23(±2.7)	0.9849(±0.006)	2957(±306)	75.12(±2.6)
Ours	0.9773(±0.029)	257.64(±11)	20.21(±1.3)	0.9886(±0.005)	618.9(±64)	23.62(±0.8)

H 在费曼数据集上与标准贝叶斯优化（贝叶斯优化）算法的比较

Table 8：在不同 k 条件下比较 CMA 与标准贝叶斯优化（512 维潜在空间） e

	方法	R^2↑	时间（秒）↓	复杂度↓
k=256	CMA	0.9872	192.3	22.94
	标准BO	0.9274	295.8	21.37
k=128	CMA	0.9675	179.1	23.26
	标准BO	0.9161	237.5	22.55
k=64	CMA	0.9388	173.2	23.93
	标准BO	0.8974	165.6	23.07

表 8 报告了在 512 维潜在空间下，CMA-ES 与标准贝叶斯优化（BO）在 GenSR 上的比较。在主要实验中，GenSR 采用 $k = 256$，对应方差最大的前k 个主潜在维度。因此，我们首先在相同设置下进行 BO 实验。如表所示，BO 的表达式复杂度略低于 CMA-ES，但其准确率（R^2）和优化时间方面的表现则明显较差。这可能是因为 BO 在处理高维优化问题时面临着挑战。

为进一步研究减少优化维度数量的影响，我们还评估了$k = 128$和 $k = 64$ （这一设置已在附录 7 报告的 k 消融研究中针对 CMA-ES 进行过考察）。虽然减小 k 提高了贝叶斯优化的运行时间，但准确率并未相应提升。同时，在较小的 k 取值下，CMA-ES 的准确率显著下降。这些结果表明， k 过小会导致重要潜在维度过度损失，从而限制搜索空间的表达能力，并降低两种优化方法的性能。

Tabl图 9：在 256 维潜在空间下，CMA 与标准贝叶斯优化在不同 k 条件下的比较 e

	方法	R^2↑	时间（秒）↓	复杂度↓
k=256	CMA	0.6984	181.4	18.23
	标准BO	0.6241	206.6	17.12
k=128	CMA	0.4935	174.5	19.08
	标准BO	0.4797	178.3	18.67

我们进一步评估了附录E.1中使用的另一种预训练模型，即潜在空间维度为 256 的 GenSR 变体，并使用 $k = 256$ 和 $k = 128$开展实验。如表 9 所示，在这种低维潜在表示下，CMA-ES 和标准BO的准确率都相对较低，这可能表明在此类设置中，潜在空间的表达能力显著降低。此外，贝叶斯优化与 CMA-ES 之间的性能差距也变得不那么明显。

总而言之，这些观察结果表明，贝叶斯优化在处理高维优化时可能存在一定局限，而 CMA-ES 对我们的任务似乎相对更稳健。通过降低潜在维度或减少优化维数来改进贝叶斯优化的尝试，仍可能受到准确率的限制，这可能是由于潜在表达能力不足或遗漏了重要维度。

I 后验符号编码器分支的消融分析

表10：GenSR与其受扰动变体的比较

方法	R^2↑	Complx↓
GenSR	0.921	19.04
无后验分支的GenSR	0.752	26.41

为了考察双分支框架对潜在空间的影响，我们构建了一个明确破坏后验分支的变体，记为**无后验分支的GenSR**。在保持整体双分支架构不变的同时，我们将数值样本 X 与随机采样方程 F 配对作为输入，并将后验分支的重构损失替换为针对与 X 对应的目标表达式计算的交叉熵损失。此外，为了直接评估模型在此设置下的固有性能，我们移除 CMA-ES 优化步骤，仅依赖预训练模型生成的初始定位。

如表 I 所示，禁用后验分支会导致性能大幅下降（R^2 从 0.921 降至 0.752，且复杂度显著增加），这表明潜在空间的底层结构遭到了严重破坏。因此，初始定位变得不可靠，凸显了双分支框架在维持结构良好的潜在表示和整体模型性能方面的关键作用。

J 与标准贝叶斯优化（BO）算法的比较

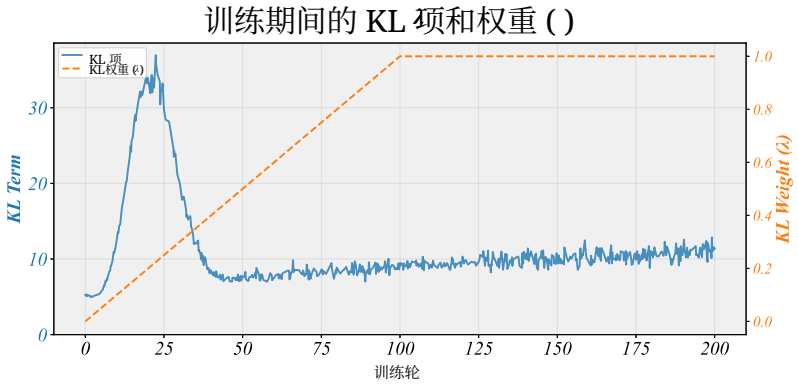


图 10：横轴表示训练轮次，左、右纵轴分别对应 KL 项和 KL权重。这里采用的 KL 权重退火策略是在前 50% 的训练轮次中将权重从 0 增至 1.0，此后保持不变。

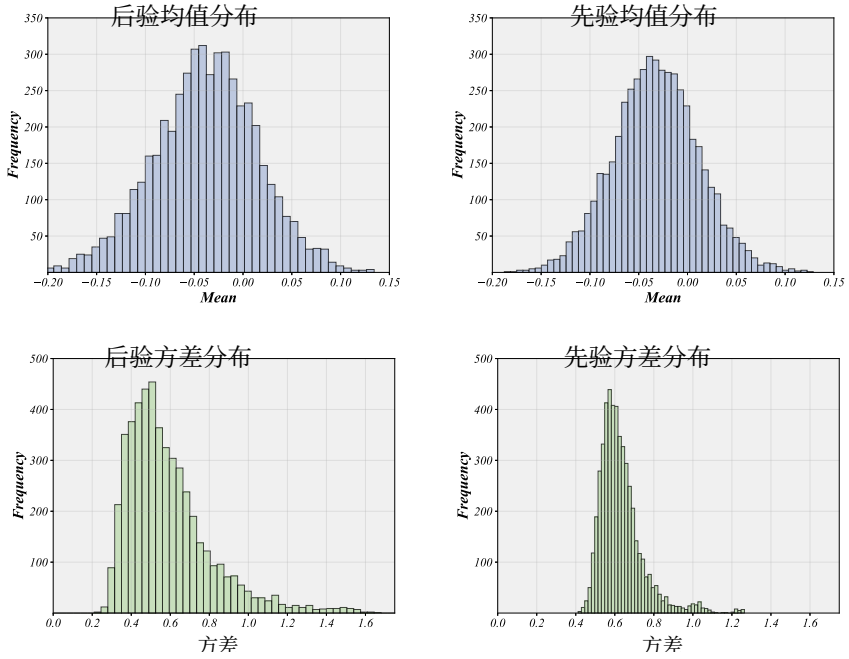


图 11：从 GenSR 中随机抽取的 5,000 个输入估计得到的先验和后验均值及方差分布。

如附录 F.2 所述，我们在训练过程中采用 KL 权重退火策略。如图 10 所示，KL权重在训练的前半阶段从 0 逐渐增加至 1，而 KL 项在后期最终稳定在相对稳定的范围内，表明没有发生坍塌。

此外，在 GenSR 中，先验分布是可学习的，而非固定为标准高斯分布。训练期间，先验的持续更新可防止后验分布简单地向先验坍塌，从而有助于降低后验坍塌的可能性。图 11 展示了使用最终预训练模型，根据 5,000 个随机生成的样本计算得到的先验和后验均值及方差分布。可以观察到，均值和方差分布均存在明显差异，这表明先验分支和后验分支并未坍塌为单一分布。如果发生了后验坍塌，后验就会趋近先验；然而，此处所示的差异表明，该模型保留了非退化且具有信息量的后验。

K 生成式与判别式潜在空间的比较

对比学习方法（例如 SNIP）构建的判别式潜在空间是碎片化的，且包含大量“空洞”，因此，在两个有效点之间采样得到的潜向量往往会解码为空字符串或其他无效字符串。相比之下，我们的 GenSR 构建了一个生成空间（连续流形），并学习方程分布——这意味着我们可以连续采样并解码出有效方程。这使得“平滑插值”成为可能，而这正是我们所执行的细粒度搜索（CMA-ES）的前提。表 K 清晰地总结了生成式与判别式

潜在空间之间的差异。此外，插值实验（附录 C.2）证实，SNIP 在搜索过程中常常陷入语义无效区域。

表 11：生成式潜在空间**与**判别式潜在空间**的比较**

属性	**生成式潜在空间（GenSR）**	**判别式潜在空间（对比式）**
学习目标	方程分布	逐点对齐
连续潜在采样	支持	不支持
适用于搜索	High	低：受无效区域限制