arXiv:2601.15167v1 [cs.PL] 21 Jan 2026

DeGAS：无需采样的概率程序基于梯度的优化

Francesca Randone[1][0009−0002−3489−9600]、Romina Doz[2][0009−0008−0210−6673]、Mirco Tribastone[3][0000−0002−6018−5989]和Luca Bortolussi[2][0000−0001−8874−4001]

[1] 维也纳工业大学，奥地利维也纳 francesca.randone@tuwien.ac.at[2] 的里雅斯特大学，意大利的里雅斯特[3] 卢卡高等研究IMT学院，意大利卢卡

摘要。我们提出 DeGAS，这是一种适用于无循环概率程序的可微高斯近似语义，可在同时包含连续分量和离散分量的模型中实现无需采样的基于梯度的优化。DeGAS 在高斯混合语义下对程序求值，并通过渐消平滑替代零测集谓词和离散分支，从而得到后验概率和路径概率的闭式表达式。我们证明这些量关于程序参数可微，因此能够通过标准自动微分进行端到端优化，无需使用蒙特卡罗估计量。在十三个基准程序上，DeGAS 的准确度和运行时间可与变分推断和马尔可夫链蒙特卡罗相媲美。重要的是，对于因涉及连续变量的条件化而导致基于采样的基线方法无法收敛的优化问题，DeGAS 也能可靠地加以解决。

关键词： 概率编程 · 高斯混合 · 可微语义 · 无采样优化

1 引言

概率编程语言（PPL）通过概率原语扩展通用语言。近年来，这类语言受到广泛关注，这主要得益于它们能够对随机化算法、 [18], 概率网络物理系统 [23], 以及机器学习模型等本质上具有随机性的系统进行建模和分析 [8]。

概率程序（PPs）通常会暴露一些必须调节的参数，以实现各项目标：最大化似然 [10, 5], 满足安全约束 [31], 校准不确定性 [13], 或优化定制效用函数 [29]。当前的实践依赖采样——无论是马尔可夫链蒙特卡罗（MCMC） [2]还是变分推断（VI） [7]。尽管这些方法往往有效，但对于由分支和硬条件化导致的不连续目标函数，它们可能产生高方差的梯度估计，并表现出脆弱性，因此常常需要针对特定问题采用重新参数化技巧 [34, 28, 1, 14, 26]。

本文提出以下问题：我们能否在不进行采样的情况下，获得概率程序目标函数的梯度？我们的答案是肯定的。我们提出了 DeGAS，这是一种可微的高斯混合近似语义，可实现端到端的可微评估，从而能够使用现成的自动微分（AD）进行优化，且无需蒙特卡罗估计量。其核心思想是对离散结构和零测度结构（例如点质量和布尔谓词）进行有原则的渐消平滑，从而使整个程序执行过程中的密度和路径概率都可微。这使得目标函数值及其关于程序参数的梯度均可确定性地计算。

DeGAS 建立在一类无循环概率程序的 SOGA（二阶高斯近似）语义之上 [30]。在 SOGA 中，每个程序状态都表示为高斯混合（GM），程序构造通过解析方式更新混合分布（仿射变换、乘积以及通过截断进行条件化），从而近似程序变量的后验分布。DeGAS 通过在需要的位置注入微小的高斯噪声，并相应地放宽谓词，平滑分支和条件化过程中产生的不连续性，同时保持对混合操作的闭合性。我们证明了后验分布和路径概率关于参数的可微性，并表明当平滑参数趋于零时，在温和的正则性条件下，可微语义会收敛到未经平滑的 SOGA 语义。

我们在 PyTorch 中实现了 DeGAS，利用其自动微分能力提供一种无需采样的基于梯度的优化器。数值实验表明，在十三个基准程序上，DeGAS 的准确性和运行时间均可与 VI 和马尔可夫链蒙特卡罗相媲美。更重要的是，我们展示了 DeGAS 能够解决这样一类优化问题实例：由于存在依赖于连续变量的条件语句等构造，基于采样的基线方法无法收敛。

论文结构。 第 2 节回顾相关工作。第 3 节介绍语法并定义可微语义。第 4 节陈述并证明我们的主要性质（可微性和收敛性）。第 5 节报告实验结果。第 6 节总结全文。

2 相关工作

据我们所知，这是首个无需采样即可计算概率程序梯度的工作。对于确定性程序，通过平滑来实现基于梯度自由方法的无采样优化这一思想，已在 [11, 12], 中得到探索，并在 [20] 中逐步适用于自动微分。然而，正如第 4 节所示，对确定性程序进行平滑并不能简单地推广到概率程序。

有人尝试将平滑扩展到 Leios [22] 中的概率程序。Leios 接受离散或混合离散-连续概率程序作为输入，并通过平滑将其近似为完全连续的概率程序。作者注意到，必须调整松弛参数，以避免将任何概率事件压缩到

零。例如，一个最初服从伯努利分布的变量经过平滑后，会变成由两个分量构成的混合分布，其均值分别为 0 和 1，标准差都很小。这意味着事件 $x == 0$ 的概率从 0.5 变为 0。为避免这种情况，Leios 通过求解一个优化问题，替换所有依赖于平滑变量的布尔谓词。从理念上说，我们的思路与之相近，都是通过受控平滑来处理不连续性。然而，我们的目标截然不同：与前面提到的工作类似，Leios 是为了辅助基于采样的推断；而我们则致力于实现无采样优化。

对于概率程序，大多数工作都集中于将自动微分与基于采样的推断相结合，为不连续程序设计梯度估计器——要么在全局范围内改进马尔可夫链蒙特卡罗和变分推断 [34, 28, 1, 14, 26]，要么仅改进不连续部分 [35, 21, 24]。这些方法仍然通过采样来估计梯度，并用于更快速或更稳定地运行推断。相比之下，我们通过计算可微程序语义来求得确定性梯度；优化循环中不会出现蒙特卡罗。这使我们能够直接优化似然和非似然概率目标（例如可达性），而不仅仅是采样器下的期望值。

一些方法从输入程序出发，计算出第二个程序，该程序的期望返回值是输入程序期望返回值的导数 [3, 27]；随后可以对新程序进行采样，以获得梯度估计。我们的目标和机制都与之不同：我们的目标是优化，而不是为推断进行导数估计；在平滑处理中，我们利用 SOGA/GM语义的解析性质 [30], ，为每个节点生成闭式截断矩和累积分布函数，然后再以解析方式通过反向传播进行计算。

3 语法与语义

SOGA 于 [30] 被提出，作为一种无循环概率编程语言的语义，用于操作服从高斯混合的输入变量。SOGA 使该语言在高斯混合下封闭，并提供后验分布的解析表示，从而能够高效计算矩、累积分布函数（CDF）和概率密度函数（PDF）。这种语义是基于高斯混合的一系列近似语义中的二阶近似；该系列语义保证收敛到真实语义，并已在各种程序中展现出很高的准确性。在此基础上，我们加入了指定用于优化的参数的功能。

记号。 对于分布 D，令 f_D 表示其概率密度函数。均值为 μ、协方差矩阵为 Σ的正态随机变量记作 $\mathcal{N}(\mu,\ \Sigma)$。高斯混合（GM）是高斯分布的加权和，$G_{\pi,\ \mu,\ \Sigma} = \sum_{i=1}^{c} \pi_i \mathcal{N}(\mu_i,\ \Sigma_i)$，其中 c是分量的数量，$(\pi_i)_{\{i=1,\ldots,c\}}$ 是权重向量，满足 $0 \leq \pi_i \leq 1$ 和 $\sum_{i=1}^{c} \pi_i = 1$。以 x_0 为中心的狄拉克δ函数记作 δ_{x_0}。令 $x[x_i \to e]$表示一个向量，其中 e是第 i个分量，其余分量均等于

x_j。令 Marg$_y(D)$ 表示分布 D 关于子向量 y的边缘分布。我们用$\mathbf{I}_n$ 表示维度为 n 的单位矩阵，并用 $\mathbb{I}_A$ 表示集合 A的特征函数。我们用 $\otimes$ 表示概率测度的乘积：若 D_1，D_2 是定义在 $\mathbb{R}$上的分布，$D_1 \otimes D_2$ 是定义在 $\mathbb{R}^2$ 上的分布，其第一坐标上的边缘分布为 D_1，第二坐标上的边缘分布为 D_2。对于谓词 b，$P_D(b)$表示在分布D下 b 的概率。$D_{|b}$ 表示在 b条件下的分布 D。

3.1 语法

变量。我们考虑定义在实变量向量 $x = (x_1, \ldots, x_n)$ 以及参数集合 Θ 上的程序，$\Theta = \{\theta_i = d_i\}_{i=1,\ldots,p}$，其中 θ_i 是参数的名称，$d_i \in \mathbb{R}$ 是初始值。此外，为每个参数指定一个区间域。例如，我们可以将标准差（std）指定为名为 σ 的参数，其初始值为 1，区间域为 $(0, +\infty)$。

表达式。程序变量上的表达式可以是两个变量的乘积，也可以是程序变量上的线性表达式，其中变量的系数或零阶项可以是常数或参数：

$$\mathbb{E} = \{x_i x_j \mid i, j = 1 \ldots n\} \cup \{c_1 x_1 + \ldots + c_n x_n + c_0 \mid c_i \in \mathbb{R} \cup \Theta\} \ .$$

布尔谓词。布尔谓词可以是真 或假，也可以是变量与实数之间的不等式，其中实数可以是常数或参数：

$$\mathbb{B} = \{true, false\} \cup \{x_i \bowtie c \mid c \in \mathbb{R} \cup \Theta, \bowtie \in \{<, \leq, ==, \geq, >\}\}$$

条件（if）语句守卫中的布尔谓词仅限于$\mathbb{B}_{\text{if}}$，且需满足 $\bowtie \in \{<, \leq, \geq, >\}$。

对表达式和布尔谓词的限制也适用于 SOGA，其动机是确保高斯混合语义在所支持的运算下保持封闭。不过，我们的语法表达能力足以编码一般的多项式赋值和守卫。

语言。我们的语法使用原语 $x \sim gm(\pi, \mu, \sigma)$ 来进行高斯混合的随机赋值，其中 $\pi = (\pi_1, \ldots, \pi_c)$、$\mu = (\mu_1, \ldots, \mu_c)$ 和 $\sigma = (\sigma_1, \ldots, \sigma_c)$ 分别是权重、均值和标准差向量，而 π_i, μ_i, σ_i 可以是实数常量或实数参数，但需满足限制：$\pi_i, \sigma_i \geq 0$ 和 $\sum_i \pi_i = 1$。我们的编程语言的完整语法如下：

$$\begin{aligned}
P ::= \ & \texttt{skip} \mid x \ \texttt{:=}\ e && \text{(deterministic assignments)} \\
& x \sim gm(\pi, \mu, \sigma) && \text{(random assignment)} \\
& \texttt{if}\ b\ \{\ P\ \}\ \texttt{else}\ \{\ P\ \} && \text{(test)} \\
& \texttt{observe}\ b && \text{(observe)} \\
& P;\ P && \text{(sequential composition)}
\end{aligned}$$

observe b 构造表示硬观测，它根据 b 指定的事件对程序的分布进行条件化，从而将所得分布的支撑集截断为满足 b 的状态。

3.2 语义

我们用 $[\![\cdot]\!]$ 表示程序的精确语义 算子，其含义遵循 Kozen 语义 2 [19] 的经典定义；而 $[\![\cdot]\!]^S$ 表示二阶高斯近似（*SOGA*）语义算子，其定义见 [30]（为完整起见，两者均在附录 A 中给出），该算子使语言对高斯混合封闭。具体而言，对于精确语义中不是高斯混合的分布（例如分支导致条件化时），我们用高斯混合替代，从而实现闭包。替换由一个专用算子矩匹配算子完成，记为 $\mathcal{T}_{\mathrm{gm}}$；该算子通过匹配密度函数的前两阶矩，以高斯混合对其进行近似。具体来说，当 $\mathcal{T}_{\mathrm{gm}}$ 作用于混合分布时，它会作用于每个分量；当它作用于单分量混合分布时，则将其映射为一个高斯分布，使其均值和协方差矩阵与原分布匹配，记为 μ_D, Σ_D。形式化地：

$$\mathcal{T}_{\mathrm{gm}}(D) = \begin{cases} \pi_1 \mathcal{T}_{\mathrm{gm}}(D_1) + \ldots + \pi_c \mathcal{T}_{\mathrm{gm}}(D_c) & \text{if } D = \sum_{i=1}^{c} \pi_i D_i \\ \mathcal{N}(\mu_D, \Sigma_D) & \text{else.} \end{cases} . \tag{1}$$

值得注意的是，在作用于高斯混合时，$\mathcal{T}_{\mathrm{gm}}$ 不会改变它们 [30]。

SOGA 语义的不可微性。 通过将语义对高斯混合封闭，可以证明，依赖于参数 Θ 的程序所产生的分布的一般概率密度函数具有如下的一般形式

$$f_\Theta(x) = \sum_{i=1}^{C} \frac{\pi_i(\Theta)}{(2\pi)^{n/2}|\Sigma_i(\Theta)|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_i(\Theta))^T \Sigma_i^{-1}(\Theta)(x-\mu_i(\Theta))\right).$$

假设 $\pi(\Theta)$ $\mu(\Theta)$ 和 $\Sigma(\Theta)$ 关于 Θ 可微，则 f_Θ 仅当协方差矩阵是奇异的，即 $|\Sigma_i(\Theta)| = 0$ 时，才会在 Θ 处不可微。

图 1 展示了可能发生这种情况的四个最小示例。在图 1a 中，y 被赋予一个常数值 c，使其方差为 0；在图 1b 中，y 被赋予一个离散的（伯努利）分布，再次得到方差为 0；在图 1c 中，y 被赋予 x 的一个确定性函数，因此 y 的分布完全由 x 的分布决定；这反映在协方差矩阵的奇异性上。在图 1d 中，观测到 y 取一个真实值，其效果类似于常数赋值。

为了一致地处理这类情况，我们对语义进行平滑。平滑过程取决于一个超参数 $\epsilon > 0$，我们假设该参数在本节其余部分中保持固定。直观地说，ϵ表示对程序进行平滑的程度，对应于施加到平滑变量上的高斯扰动的标准差。

接下来，我们分别讨论如何平滑布尔谓词和整体语义。在本节中，我们用 V 表示已被平滑的程序变量集合。

```
x = 高斯混合([1.], [0], [1]);
y =c;
```

$\pi = 1, \quad \mu = \begin{bmatrix} 0 \\ c \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$

(a) 常量赋值。

```
x = 高斯混合([1.], [0], [1]);
y = 高斯混合([0.5, 0.5], [0, 1], [0, 0]);
```

$\pi_1 = 0.5, \ \mu_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$

$\pi_2 = 0.5, \ \mu_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$

(b) 离散分布赋值。

```
x = 高斯混合([1.], [0], [1]);
y = x + 1;
```

$\pi = 1, \quad \mu = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

(c) 确定性函数。

```
x = 高斯混合([1.], [0], [1]);
y = 高斯混合([1.], [0], [1]);
观测(y == 0.5);
```

$\pi = 1, \quad \mu = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$

(d) 对一个常量值的观测。

图 1：SOGA 语义的不可微性。导致协方差矩阵奇异的示例。

布尔谓词的平滑算子。我们定义如下用于布尔谓词的平滑算子，它以谓词 b 和一组平滑变量 V 为输入，并返回一个新谓词。

$$\mathcal{B}_\epsilon(x \bowtie c, V) = \begin{cases} x > c + \delta_\epsilon & \text{if } \bowtie \text{ is } >, x \in V \\ x > c - \delta_\epsilon & \text{if } \bowtie \text{ is } \geq, x \in V \\ x < c + \delta_\epsilon & \text{if } \bowtie \text{ is } \leq, x \in V \\ x < c - \delta_\epsilon & \text{if } \bowtie \text{ is } <, x \in V \\ (x > c - \delta_\epsilon) \wedge (x < c + \delta_\epsilon) & \text{if } \bowtie \text{ is } == t, x \in V \\ x \bowtie c & \text{else} \end{cases} \tag{2}$$

目前，我们允许 δ_ϵ 随 ϵ 变化，唯一的假设是 $\lim_{\epsilon \to 0} \delta_\epsilon = 0$。下一节中，我们将引入一项额外限制，以保证收敛到 SOGA 语义。

DeGAS。与 [30],一样，我们通过程序的控制流图（控制流图）来定义其语义。图中的每个节点都表示一条程序指令。我们考虑六种节点类型：入口、det、随机节点、测试、观测 和出口，并使用记号 v：类型 表示 v 的类型为类型。类型为 det 的节点标记为 skip

或标记为确定性赋值 x_i := e。类型为随机节点的节点标记为随机赋值 $x_i \sim gm(\pi, \mu, \sigma)$。类型为测试 的节点标记为 $\mathbb{B}_{\text{if}}$ 中的布尔谓词，而观测节点则标记为 $\mathbb{B}$ 中的布尔谓词。函数arg 定义在所有带标记的节点 v 上，返回 v 的标记。此外，函数条件 定义在 det 和随机节点节点集合上，返回 $true, false$ 或$none$ ，且当且仅当 v 的父节点不是测试节点时，条件$(v) = none$ 。算术表达式和布尔谓词均按标准方式解释。表达式 $e \in \mathbb{E}$ 表示映射e: $\mathbb{R}^n \to \mathbb{R}$。布尔谓词 $b \in \mathbb{B}$ 表示满足该谓词的向量 $x \in \mathbb{R}^n$ 集合。

对于一个程序 P 我们令 Ω^P 表示其控制流图中的路径集合。对于每条路径 $\omega \in \Omega^P$ 函数 $s_\omega(v)$ 将 $v \in \omega$ 与其在 ω中的后继相关联。路径 ω的语义，记作 $[\![\omega]\!]^{\Delta, \epsilon}$ 定义为一个不接受输入、返回一对(p, D) 的函数，其中 $p \in \mathbb{R}_{\geq 0}$ 且 D 是 $\mathbb{R}^n$上的一个高斯混合。路径 $\omega = v_0..v_n$ 的语义定义为该路径中每个节点语义的复合： $[\![\omega]\!]^{\Delta, \epsilon} = [\![v_n]\!]^{\Delta, \epsilon}_{\omega} \circ \ldots \circ [\![v_0]\!]^{\Delta, \omega, \epsilon}$.

如前所述，在计算节点的语义时，除了p 和 D之外，我们还需要跟踪平滑变量，因此将变量集合 V加入语义中，其中包含在精确语义中为离散型的变量。节点 v 的语义是一个函数，其输入为三元组 (p, D, V)，并返回同类型的三元组 (p', D', V')。唯一的例外是入口节点，它不接受输入；以及出口节点，它只输出二元组 (p, D)。每种节点类型的语义分别定义。

– 如果 v: 入口，那么 $[\![v]\!]^{\Delta, \epsilon}_{\omega} = (1, \mathcal{N}(0, \mathbb{I}_n)\ \emptyset)$。– 如果 v: det, 令 D' 表示 $x[x_i \to e]$ 的分布， D^ϵ 表示 $x[x_i \to e + z]$的分布，其中 z 是一个新的程序变量，其分布为 $\mathcal{N}(0, \epsilon)$。若 e中的所有变量都在 V中，则令 V'为 $V \cup \{x_i\}$，否则令其为$V' = V$。那么：

$$[\![v]\!]^{\Delta, \epsilon}_{\omega}(p, D, V) = \begin{cases} (p, D, V) & arg(v) = \texttt{skip} \\ (p, D^\epsilon, V \cup \{x_i\}) & arg(v) = x_i \texttt{ := } c \\ (p, D^\epsilon, V') & arg(v) = x_i \texttt{ := } e, e \text{ linear}, x_i \text{ not in } e \\ (p, \mathcal{T}_{\text{gm}}(D'), V') & \text{else.} \end{cases}$$

– 如果 v: 随机节点 且 $arg(v) = x_i \sim gm(\pi, \mu, \sigma)$，若 $\sigma_j \neq 0$，则令 σ'为 $\sigma'_j = \sigma_j$，否则为 $\sigma'_j = \epsilon$。那么：

$$[\![v]\!]^{\Delta, \epsilon}_{\omega}(p, D, V) = \begin{cases} (p, \text{Marg}_{x \setminus x_i}(D) \otimes G_{\pi, \mu, \sigma'}, V \cup \{x_i\}) & \sigma_j = 0 \text{ for some } j \\ (p, \text{Marg}_{x \setminus x_i}(D) \otimes G_{\pi, \mu, \sigma}, V) & \text{else.} \end{cases}$$

– 若 v: 测试，令 $arg(v) = b$和 $b' = \mathcal{B}_\epsilon(b, V)$:

$$[\![v]\!]^{\Delta, \epsilon}_{\omega}(p, D, V) = \begin{cases} (p \cdot P_D(b'), \mathcal{T}_{\text{gm}}(D_{|b'}), V) & cond(s_\omega(v)) = true \\ (p \cdot P_D(\neg b'), \mathcal{T}_{\text{gm}}(D_{|\neg b'}), V) & cond(s_\omega(v)) = false. \end{cases}$$

– 若 v：观测，令 $arg(v) = b$, $I = \int_{\mathbb{R}^{n-1}} f_D(x[x_i \to c])d(x \setminus x_i)$ 和 $b' = \mathcal{B}_\epsilon(b, V)$：

$$[\![v]\!]^{\Delta,\epsilon}_\omega(p, D, V) = \begin{cases} (p \cdot I, \mathrm{Marg}_{x \setminus x_i}(D_{|b}) \otimes \mathcal{N}(c, \epsilon), V \cup \{x_i\}) & \\ & b \text{ is } x_i == c, x_i \notin V \\ (p \cdot P_D(b'), \mathcal{T}_{\mathrm{gm}}(D_{|b'}), V) & \text{else} \end{cases}$$

– 如果 v：出口，$[\![v]\!]^{\Delta,\epsilon}_\omega(p, D, V) = (p, D)$。

如果至少存在一条路径 $\omega \in \Omega^P$ $[\![\omega]\!] = (p, D)$ 满足 $p > 0$，则称程序 P 为有效程序。有效程序 P 的语义定义为其各条路径语义的混合分布：

$$[\![P]\!]^{\Delta,\epsilon} = \sum_{\substack{\omega \in \Omega^P \\ [\![\omega]\!]^{\Delta,\epsilon} = (p,D)}} \frac{p \cdot D}{\sum_{\substack{\omega' \in \Omega^P \\ [\![\omega']\!]^{\Delta,\epsilon} = (p',D')}} p'}.$$

让我们回到图 1，简要讨论 $[\![\cdot]\!]^{\Delta,\epsilon}$ 如何避免不连续性。对于图 1a 中的程序，$[\![\cdot]\!]^{\Delta,\epsilon}$ 实际上会将 y 赋值为 $c + z$，其中 z 是一个新定义的变量，服从均值为 0、标准差为 ϵ 的高斯分布。这等价于将 y 赋值为 $\mathcal{N}(c, \epsilon)$，并得到一个非退化的协方差矩阵 $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon^2 \end{bmatrix}$。对于图 1b 中的程序，$[\![\cdot]\!]^{\Delta,\epsilon}$ 会用标准差为 ϵ 的非退化分量替换标准差为零的分量。因此，实际执行的赋值为 $y = gm([0.5, 0.5], [0, 1], [\epsilon, \epsilon])$，从而得到 $\Sigma_1 = \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon^2 \end{bmatrix}$。对于图 1c 中的程序，$[\![\cdot]\!]^{\Delta,\epsilon}$ 的作用与第一种情况类似：y 被赋值为 $x + 1 + z$，其中 z 是一个新定义的高斯变量，均值为 0、标准差为 ϵ。此时协方差矩阵变为 $\Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 1 + \epsilon^2 \end{bmatrix}$。最后，对于图 1d 中的程序，观测后会校正 y 的方差，并将其设为 ϵ，从而得到 $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon^2 \end{bmatrix}$。

为避免平滑导致概率质量的行为发生意外变化，当布尔谓词涉及平滑变量时，$[\![\cdot]\!]^{\Delta,\epsilon}$ 会使用算子 $\mathcal{B}_\epsilon$ 对其进行不同的解释，这体现在 $test$ 和 $observe$ 节点的语义定义中。这样便能避免对零概率事件进行条件化，因为 $[\![\cdot]\!]^{\Delta,\epsilon}$ 始终作用于非退化分布，而 $\mathcal{B}_\epsilon(b, V)$ 始终返回开集。

在下一节中，我们将证明，$[\![\cdot]\!]^{\Delta,\epsilon}$ 计算出的语义确实关于程序参数可微；并且通过在应用 $\mathcal{B}_\epsilon$ $[\![\cdot]\!]^{\Delta,\epsilon}$ 时恰当选择 δ_ϵ 的值，将收敛到 $[\![\cdot]\!]^S$。

例 *1*。考虑程序 P 及其在图 2 中的控制流图，其中 θ 和 σ 是参数，且 $\theta \in (-\infty, +\infty)$ 和 $\sigma \in (0, +\infty)$。其精确语义为：

$$[\![P]\!] = \Phi(\theta/\sigma) \cdot (\mathcal{N}(0, \sigma)_{|x<\theta} \otimes \delta_{-1}) + (1 - \Phi(\theta/\sigma)) \cdot (\mathcal{N}(0, \sigma)_{|x \geq \theta} \otimes \delta_1)$$

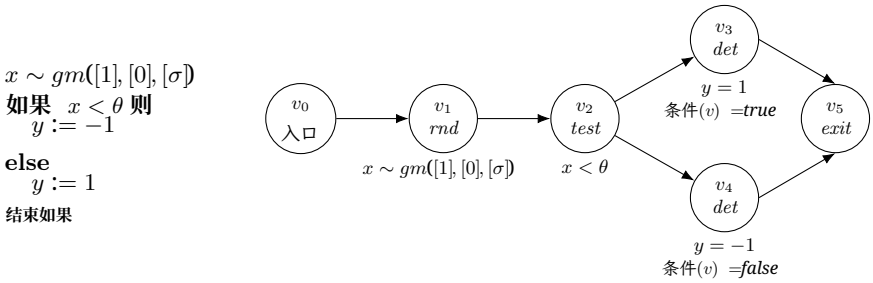


图 2：示例程序 P（左）及其控制流图（右）。

具体而言，关于 y 的精确边缘分布是一个离散分布，将 $\Phi(\theta/\sigma)$ 的概率质量赋予 -1，并将 $1-\Phi(\theta/\sigma)$ 的概率质量赋予 1。因此，关于y 的精确边缘分布关于这些参数是不可微的（事实上是不连续的）。

使用对高斯混合封闭的 SOGA 语义，可得:

$$[\![P]\!]^S = \Phi(\theta/\sigma)\cdot \mathcal{T}_{\mathrm{gm}}(\mathcal{N}(0,\sigma)_{|x<\theta}\otimes\delta_{-1}) + (1-\Phi(\theta/\sigma))\cdot \mathcal{T}_{\mathrm{gm}}(\mathcal{N}(0,\sigma)_{|x\geq\theta}\otimes\delta_1)$$

然而，关于 y 的边缘分布保持不变，因此仍然不可微。在计算可微语义时，不仅应用了矩匹配算子，使语义相对于高斯混合保持闭合，还通过平滑 y的赋值来避免退化。这得到:

$$[\![P]\!]^{\Delta,\epsilon} = \Phi(\theta/\sigma)\cdot \mathcal{T}_{\mathrm{gm}}(\mathcal{N}(0,\sigma)_{|x<\theta}\otimes\mathcal{N}(-1,\epsilon)) + \\ + (1-\Phi(\theta/\sigma))\cdot \mathcal{T}_{\mathrm{gm}}(\mathcal{N}(0,\sigma)_{|x\geq\theta}\otimes\mathcal{N}(1,\epsilon))$$

对于任意 $\sigma \in (0, +\infty)$，关于 y 的可微边缘分布为

$$\Phi(\theta/\sigma)\cdot\mathcal{N}(-1,\epsilon) + (1-\Phi(\theta/\sigma))\cdot\mathcal{N}(1,\epsilon).$$

针对 $\sigma = 1$ 和 $\theta = 0$计算得到的三种语义的边缘密度如图 3 所示，其中混合分布各分量的密度以橙色突出显示；对于 DeGAS，我们固定了 $\epsilon = 0.05$。

4 $[\![\cdot]\!]^{\Delta,\epsilon}$的性质

本节将证明 DeGAS 语义算子 $[\![\cdot]\!]^{\Delta,\epsilon}$的两个性质，即 $[\![P]\!]^{\Delta,\epsilon}$ 关于参数的可微性，以及当 $\epsilon \to 0$时，$[\![P]\!]^{\Delta,\epsilon}$收敛到 $[\![P]\!]^S$。

可微性。 首先，我们关注 DeGAS 语义关于 Θ 的可微性。因此，在本小节中，我们使用记号 $P(\Theta)$，明确表示程序对参数的依赖。例如，例 1 中的程序 P可记为 $P(\theta, \sigma)$。我们假设参数具有指定的初始值和区间域，且

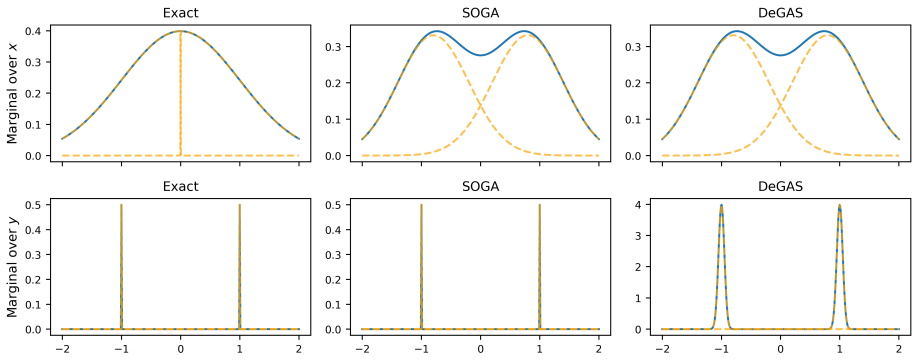


图 3：程序 P 在不同语义下的边缘后验密度。橙色线条突出显示了混合分布中的不同分量。

它们的取值始终位于区间域内。例如，对于 $P(\theta,\ \sigma)$，我们可以指定初始值 $d_\theta = 0$ 和 $d_\sigma = 1$，以及区间域$\theta \in (-\infty,\ +\infty)$、$\sigma \in (0,\ +\infty)$。如果分布 $D(\Theta)$ 的概率密度函数可微，我们就称其关于 Θ 可微。例如，如果 $D(\Theta)$ 是高斯混合，若对于 $i = 1, \ldots\ldots, c$，(i) $\pi_i(\Theta)$、$\mu_i(\Theta)$ 和 $\Sigma_i(\Theta)$ 关于 Θ 可微；(ii) 对于每个满足约束的 Θ 取值，$\Sigma_i(\Theta)$ 都是非奇异的，则 $D(\Theta)$ 关于 Θ 可微。

定理 1. 对于任意有效程序 $P(\Theta)$，$[\![P(\Theta)]\!]^{\Delta,\ \epsilon}$返回 $(p'(\Theta)\ D'(\Theta)\ V')$，使得 $p'(\Theta)$和 $D'(\Theta)$在 Θ 上可微。

这是本文的核心理论结果（证明见附录），因为它使得基于梯度的优化方法能够安全地应用于概率程序。

收敛性。 DeGAS 通过高斯扰动，在 SOGA 已近似的语义 $[\![\cdot]\!]^S$ 之上又增加了一层近似。我们证明，当扰动变得非常小时，即在 $\epsilon \to 0$ 的极限下，这一额外近似趋于消失。结合 SOGA 收敛到真实语义的结果 $[\![\cdot]\!]$，这为一类概率程序获取无采样梯度提供了有理论依据的方法。

DeGAS 的收敛性证明（见附录）出人意料地微妙，值得给出直观解释。对谓词进行平滑时，δ_ϵ 对 ϵ 的依赖可能会危及收敛。下面的例子说明了引入一项额外要求的原因，这项要求称为 一致性，它是保证收敛所必需的。

例 *1.* 我们考虑两个示例程序：

$$P_1 \colon x := 0; y \sim gm([1.],[0.],[1.]); \texttt{observe}\ x \geq 0,$$
$$P_2 \colon x := 0; y \sim gm([1.],[0.],[1.]); \texttt{observe}\ x > 0.$$

在 SOGA 中，我们有 $[\![P_1]\!]^S = (1, \delta_0 \times \mathcal{N}(0,1))$ 和 $[\![P_2]\!]^S = (0, \delta_0 \times \mathcal{N}(0,1))$。在 DeGAS 中，对于这两个程序，在 observe 语句之前，我们有输出三元组 $(1, D_\epsilon, \{x\})$，其中 $D_\epsilon = \mathcal{N}(0,\epsilon) \times \mathcal{N}(0,1)$。不过，观测节点的谓词经过了不同的平滑处理：

$$\mathcal{B}_\epsilon(x \geq 0, \{x\}) = x > -\delta_\epsilon, \qquad \mathcal{B}_\epsilon(x > 0, \{x\}) = x > \delta_\epsilon.$$

考虑 P_1。要计算观测节点的可微语义，我们需要计算 $P_{D_\epsilon}(x > -\delta_\epsilon)$ 和 $(D_\epsilon)_{|x>\delta_\epsilon}$ 的矩，并希望它们分别趋近于 1 和 $\delta_0 \times \mathcal{N}(0,1)$ 的矩。这里我们重点关注概率，因为矩的收敛可用类似的论证证明。令 Φ 表示标准高斯分布的累积分布函数，则 $P_{D_\epsilon}(x > -\delta_\epsilon) = 1 - \Phi\left(-\frac{\delta_\epsilon}{\epsilon}\right)$。要使该量趋近于 1，我们需要 $\frac{\delta_\epsilon}{\epsilon} \to \infty$。直观地说，这个极限要求 δ_ϵ 的减小速度慢于 x 的方差（此处为 ϵ），这样当 ϵ 趋近于 0 时，D_ϵ 中越来越多的概率质量就会落在 $\{x > -\delta_\epsilon\}$ 内。此外，显然，如果 δ_ϵ 是 ϵ 的线性函数，或者其减小速度比 ϵ 更快，这个极限就无法成立。

类似地，考虑 P_2，其中 $P_{D_\epsilon}(x > \delta_\epsilon) = 1 - \Phi(\frac{\delta_\epsilon}{\epsilon})$。要使其趋于 0，我们仍然需要 $\frac{\delta_\epsilon}{\epsilon} \to \infty$。这一次，$\delta_\epsilon$ 趋于 0 的速度较慢，因此 D_ϵ 的较大部分概率质量可以留在 $\{x > \delta_\epsilon\}$之外。

前面的例子引出了以下定义和定理（完整证明见附录）。

定义 1. 若对每个测试节点和观测节点，只要其参数为 $(v) = x_i \bowtie c$，在应用 $\mathcal{B}_\epsilon$ 计算 $[\![v]\!]^{\Delta,\epsilon}(p, D, V)$ 时，$x_i \in V$，δ 的选取满足以下条件，则称算子 $\mathcal{B}_\epsilon$ 的应用是一致的：

$$\lim_{\epsilon \to 0} \delta_\epsilon(\epsilon) = 0 \qquad and \qquad \lim_{\epsilon \to 0} \frac{\delta_\epsilon(\epsilon)}{\sqrt{\Sigma_{x_i,x_i}(\epsilon)}} = +\infty$$

对于 D 的某个分量的每个 $\Sigma(\epsilon)$ 协方差矩阵。Σ 对 ϵ 的依赖性由以下事实保证：$x_i \in V$。

定理 2. 假设 P 是一个有效程序，并且在计算 $[\![P]\!]^{\Delta,\epsilon}$ 时一致地应用了 $\mathcal{B}_\epsilon$。那么 $\lim_{\epsilon \to 0} [\![P]\!]^{\Delta,\epsilon} = [\![P]\!]^S$（这里，分布的收敛是指依分布收敛）。

我们要指出的是，尽管从理论上讲，可以让 ϵ 趋近于 0，从而使额外误差消失，但在实践中，我们希望使用足够大的 ϵ，以实现高效的基于梯度的优化，下一节将对此进行讨论。

5 实验评估

设置。 我们从两个方面评估 DeGAS：（i）优化取自文献中的基准概率程序的参数，并与变分推断（VI）和马尔可夫链蒙特卡罗（MCMC）进行比较；（ii）我们评估

DeGAS 在含有与连续变量相关的不连续性的程序上的表现。正如我们将要展示的，这类不连续性会使标准优化方法不再适用。在这种情况下，我们考虑信息物理系统（CPS）的经典模型，通过综合其参数来优化轨迹似然或可达性性质。

我们使用 PyTorch 实现了 DeGAS，利用自动微分计算梯度（参见附录 C），并采用 Adam 作为优化器。所有实验均设定 $\delta_\epsilon = \sqrt{\epsilon}$，并报告 $\epsilon = 0.001$ 的结果。选择 δ_ϵ 符合一致性假设。此外，如附录 D.1 所述，只要 ϵ 保持在 0。1 以下，改变其取值并未导致相较于 SOGA 近似的精度发生显著变化。附录 D.6 中报告的结果进一步证实了这一点，表明误差的主要来源是 SOGA 引入的误差，而对于所考虑的取值，ϵ 的经验影响可以忽略不计。关于推断任务中 SOGA 误差的专门分析，包括提高高斯近似阶数的讨论，请参阅 [30]。所有实验均在一台配备 Apple M2 处理器的机器上运行，该处理器有 8 个核心，运行频率为 3.49 GHz，内存为 8 GB。所有实验的超时阈值均设为 600 秒。

与 VI 和 MCMC 的比较。 我们首先在取自概率编程语言文献（[10, 15,16]）的 13 个案例研究上，将 DeGAS 与 Pyro [5] 中实现的变分推断（VI）和马尔可夫链蒙特卡罗（MCMC）进行比较。作为标准的基准优化任务，我们还考察了比例-积分-微分（PID）控制器，其描述见附录 D.2。

对于每个案例研究，一些重要变量被设为可优化参数，其余变量则设为观测量。对于这些观测量，我们根据参数的真实值生成大小为 $N = 1000$ 的数据集 D。在 PID 案例研究中，数据对应于稳定动态的理想化表示，使系统保持在平衡点。对于每个程序，使用 DeGAS 计算出的后验密度来计算数据集 D 的负对数似然 $l(\Theta) = -\log f_\Theta(x)$；随后，我们进行优化以求得 $\Theta^* = \text{argmin}_\Theta l(\Theta)$。

对于 VI 和 DeGAS，我们报告在 $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.2\}$ 中选取的、能够达到最小误差的学习率对应的结果。优化在容差为 10^{-8}且耐心值为 30 次迭代时达到收敛后停止；DeGAS 最多运行 500 步（PID 模型除外，最多 1000 步），VI 最多运行 1000 步。对于 VI，我们使用 Trace_ELBO 损失[4]。对于马尔可夫链蒙特卡罗，我们使用 4 条链运行 NUTS 核，首先以 500 个样本和 50 个预热步数运行推断 ˆ。如果 R 超过 1.05，则重复该过程：当总步数在 2000 到 5000 之间时，样本数增加 500，预热步数增加 50；此后每次增加 100，直至总步数达到最多 6000。所有超参数值均列于附录 D.3。

表1汇总了结果，其中“时间”指10次执行的平均运行时间（单位：秒），“误差”指程序所有可优化参数的优化值与真实值之间的平均相对差异

[4] https://docs.pyro.ai/en/dev/inference_算法.html

模型	# 参数	# 变量	VI		MCMC		DeGAS	
			时间	误差	时间	误差	时间	误差
伯努利	1	2	0.204 (		10.968 (		1.986	0.001
入室盗窃	2	7	3.244	0.913	13.014	0.141	14.347	1.686
ClickGraph	1	6	1。635					
临床试验	3	6	1.055				17.070 (	
草	4	6	3。077	0。306	58。151	0。015	317。582	0。036
谋杀之谜	1	1	0.792				2.508	0.203
SurveyUnbiased	2	2	0.794					
TrueSkills	3	6	2.056 (		t.o.	t.o.	1.664	0.003
TwoCoins	2	1	0.095				3.410	0.688
AlterMu	3	5	1.007 :		t.o.	t.o.	0.640	0.312
AlterMu2	2	3	0.718 (				0.618	0.096
正态混合	3	3	1.061	0.909	345.409	0.053	2.941	0.386
PID	2	55	10.778	–	nc	nc	213	–

表1：各模型上推断方法的比较。灰色突出显示的模型表示在这些情况下，DeGAS 在准确度或执行时间方面优于另外两种方法。对于最后一个模型 PID，由于真实参数未知，我们无法量化误差，但在图7中展示了优化结果。对于此案例研究，马尔可夫链蒙特卡罗无法收敛（$R \gg 1$。）

。如表所示，我们的方法通常能达到与基线方法相当的性能，并且在若干基准测试中，准确度优于基线方法。对 DeGAS 而言，Grass 和入室盗窃这两个模型尤其具有挑战性。这两个模型都包含大量嵌套的 if 语句，导致 GM近似中的分量数量庞大。SOGA 也存在这一已知问题，可以通过剪枝策略缓解 [30], ，但该策略的调优不在本文讨论范围内。

许多基准程序包含针对离散变量的布尔守卫条件和条件语句。当这些条件可以枚举时，两种变分推断
马尔可夫链蒙特卡罗可以通过枚举正确执行推断。[5] 然而，当条件取决于连续变量时，生成的模型会变得不连续且不可微，从而破坏基于梯度的变分推断所要求的平滑性假设，并影响马尔可夫链蒙特卡罗的收敛。这种现象在右侧插图中的程序中清晰可见，该程序改编自 [25], ，其中

$v \sim gm([1],[\mu_1],[5])$**如果** $v > 0$ **则**$y \sim gm([1],[\mu_2],[1])$**否则** $y \sim gm([1],[-2],[1])$**结束如果**

μ_1 和 μ_2 是可优化参数，其真实值分别为 0。5 和 1。图 4 显示，VI 的损失值呈振荡状态（同样地，MCMC 未能达到 $R < 1.05$的值，表明未收敛），而 DeGAS 则成功完成了优化。这凸显了 DeGAS 与现有方法的互补性：

[5] https://pyro.ai/examples/enumeration.html

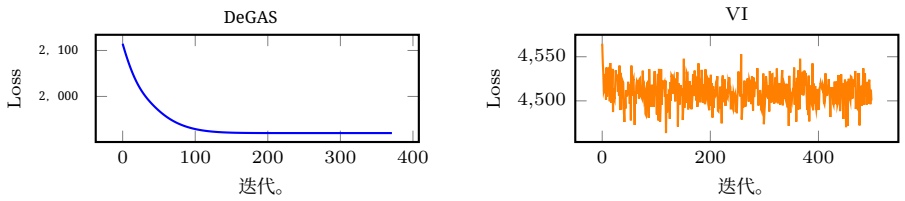


图 4：第5节中程序的 DeGAS 与 VI 损失曲线对比。

尽管在某些基准测试中可能需要更长的运行时间，但在具有挑战性的非连续设置下，它也能实现高效优化，而 VI 和马尔可夫链蒙特卡罗已知在此类情况下表现不佳。

信息物理系统的参数综合。 现在我们考虑信息物理系统（CPS）的模型，其中控制流分支取决于连续随机变量，因此会出现前文所述的不可微性问题。相比之下，DeGAS 提供了损失函数的解析形式：我们展示了数据似然和复杂可达性目标的示例。我们考虑以下取自 [32, 11] 且在附录 D.4 中完整描述的模型：

1.**恒温器** 表示一种恒温器，它通过在温度越过阈值t_{on} 和 t_{off} （可优化参数）时开启或关闭加热器来维持恒定温度。温度按照一阶衰减过程演化；加热器开启时会增加一个加热项，同时还会受到噪声造成的随机波动影响。

2. **变速箱** 表示汽车的传动系统。我们考虑一个有三个挡位的模型。控制器决定何时从一个挡位换到下一个挡位，并且只允许连续挡位之间的转换。阈值 s_i （可优化参数）规定了释放挡位 i 和接合挡位 $i+1$时的速度。此外，换挡并非瞬时完成：在我们的模型中，每次转换持续 0.3 秒，在此期间变速箱处于脱开状态。挡位接合时，速度线性增长；挡位脱开时，速度按二次方规律下降。两种情况下，速度都会受到随机波动的影响。

3. **受控弹跳球**表示一个落到带有弹簧和阻尼器的平台上的球。弹簧会以由柔顺性 C 决定的力将球推回，而阻尼器则根据系数 R （二者均为可优化参数）使球减速。当球到达平台时，系统会切换到反弹模式；一旦高度大于零，便恢复自由落体。

基于数据似然的参数综合。与前面的案例研究一样，优化的目标是针对使用真实参数值从模型生成的 100 条轨迹，最大化负对数似然。表2和图5显示，DeGAS始终能够恢复出接近真实值的参数值。优化后的平均轨迹（红色）准确地

表2：基于轨迹的CPS优化结果

案例研究	参数	学习率	步数	时间	Init	目标值	结果
恒温器	t_{ON}, t_{OFF}	0.1	40	61.964	(15,22)	(17, 20)	(16.78, 19.97)
变速箱	s_1, s_2	0.15	200	167.643	(8, 12)	(10，20)	(8。69，19。24)
弹跳球	$R, 1/C$	0.8	100	500.621	$(-1, 450)$	(7,400)	(5.27,408.99)

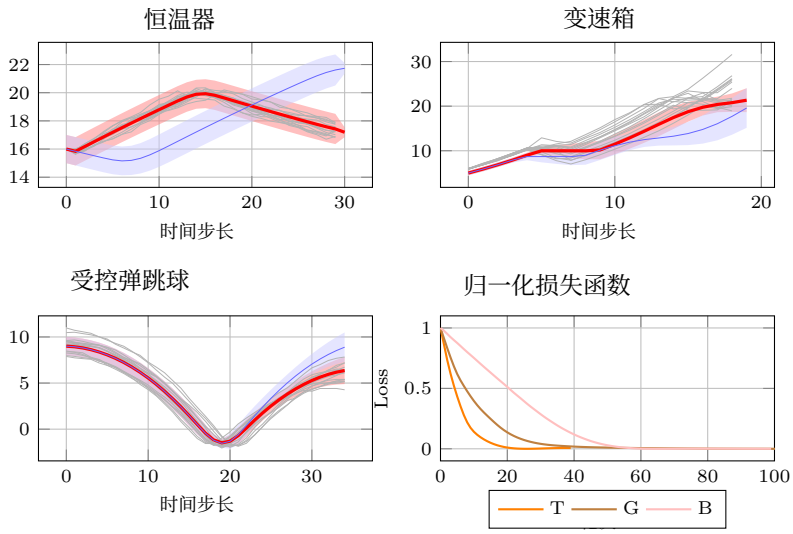


图 5：使用优化后的参数得到的均值轨迹及一个标准差范围（红色），与使用初始参数生成的轨迹（蓝色）进行比较。灰色轨迹表示用于优化的样本。

捕捉了随机模拟（灰色）的行为，即使初始轨迹（蓝色）与正确轨迹相差甚远。归一化损失曲线证实了这三个案例均已收敛。

基于可达性的综合 一种互补的综合目标是优化信息物理系统的参数，以最大化到达特定目标状态或满足某些行为约束的概率。我们考虑三种此类情况。恒温器：目标是最大化在不同时间点（$\tau_1 = 0.6$，$\tau_2 = 1.8$ 和 $\tau_3 = 2.4$）处于模式“ON”下到达区域 $T(t) \in [T_{min}, T_{max}] = [19.9, 20.1]$ 的概率；变速箱：我们最大化在整个轨迹中保持速度低于 16 的概率；弹跳球：我们希望最大化球在弹跳一次后坠落时到达 $H \geq 7$ 的概率。可优化参数与上一节中的相同。

表 3：基于可达性性质的 CPS优化结果

案例研究	学习率	步数	时间（秒）	Init	初始损失	结果	最终损失
恒温器	0.1	100	2770	(16.5, 22)	$-2e^{-12}$	(19.04, 21.30)	-5695
1变速箱	0.5	40	52.642	(10, 20)	1.000	(1.92, −1.57)	0.966
弹跳球 0.2		50	184.264	(7, 200)	−0.002	(−3.87, 210.14)	−0.979

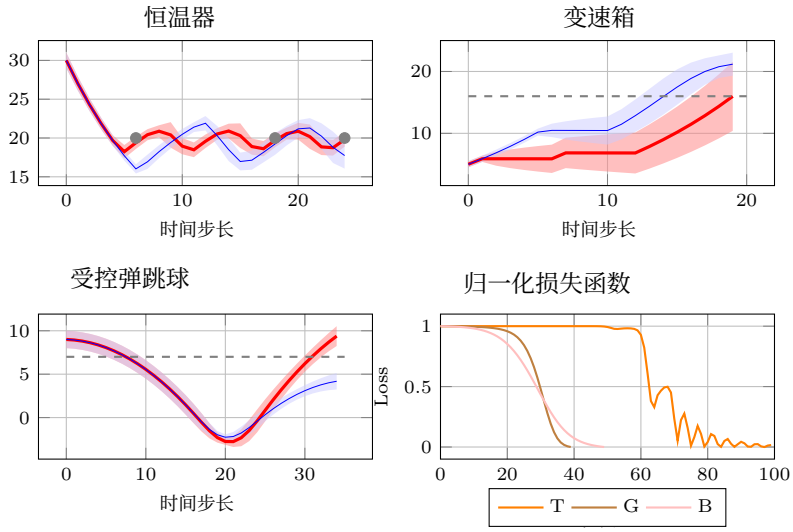


图6：通过优化灰色表示的目标值获得的平均轨迹及一个标准差范围（红色），与初始轨迹（蓝色）进行比较

我们在表 3 和图 6 中报告结果。由于真实的最优参数未知，表中报告了初始和最终损失值。如图所示，对于所有模型，优化后的轨迹（红色）都满足所需的行为约束，而初始轨迹（蓝色）则不满足。优化曲线进一步证实了这三个模型均已收敛。

6 结论

我们提出了 DeGAS，这是一种通过可微程序语义实现概率程序无采样优化的方法。DeGAS 对 SOGA 进行了光滑化。SOGA 是一种语义，它通过高斯分布混合更新（仿射映射、乘积和截断矩）来表示程序的演化，从而能够以解析形式表示后验分布，但这种表示通常是不连续的。我们展示了 DeGAS 在解决综合任务方面的潜力；由于不连续性会阻碍收敛，这些任务无法直接通过变分推断、马尔可夫链蒙特卡罗等经典优化方法解决。

原则上，DeGAS 背后的主要思想可应用于任何满足以下条件的语义：(i) 以易于处理的形式表示程序状态；(ii) 支持对变换和条件化进行闭式（或确定性求积）更新；(iii) 能够对离散构造或测度为零的构造进行光滑松弛，且其参数 $\varepsilon \to 0$ 能够还原原始语义。我们推测，在这些条件下，关于可微性的所有结论都适用于 $\varepsilon > 0$，并且通过与 SOGA 类似的论证，可以证明其收敛回未平滑的语义。因此，未来一个值得探索的方向是研究其他语义，例如概率电路、指数分布混合或一般的符号概率程序，它们能够提供分布的解析或半解析表示。

数据可用性声明

DeGAS 可在 GitHub 上公开获取：https://github.com/frarandone/DeGAS。本文附有用于复现实验结果的材料，可在以下网址获取：https://zenodo.org/records/18197807。

致谢

本项目获得欧盟地平线2020研究与创新计划资助，资助依据为玛丽·斯克沃多夫斯卡-居里资助协议第 101205923 号（MSCA 博士后奖学金 EMBEr）。

参考文献

1. Afshar, H.M., Domke, J.：反射、折射与哈密顿蒙特卡罗。载于：Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R.（编）《神经信息处理系统进展28：神经信息处理系统2015年度会议》，2015年12月7—12日，蒙特利尔，魁北克，加拿大。第3007–3015页（2015年） 2. Andrieu, C., De Freitas, N., Doucet, A., Jordan, M.I.：机器学习中的马尔可夫链蒙特卡罗简介。《机器学习》**50**(1)，5–43（2003年） 3. Arya, G., Schauer, M., Schäfer, F., Rackauckas, C.：对具有离散随机性的程序进行自动微分。载于：Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A.（编）《神经信息处理系统进展35：神经信息处理系统2022年度会议》，神经信息处理系统大会2022，美国路易斯安那州新奥尔良，2022年11月28日—12月9日（2022年） 4. Billingsley, P.：《概率测度的收敛》。John Wiley & Sons（2013年） 5. Bingham, E., Chen, J.P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., Goodman, N.D.：Pyro：深度通用概率编程。《机器学习研究杂志》**20**(28)，1–6（2019年） 6. Bishop, C.M., Nasrabadi, N.M.：《模式识别与机器学习》，第4卷。Springer（2006年）

7. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.：变分推断：统计学家综述。《美国统计学会会刊》**112**(518)，859–877（2017年） 8. Borgström, J., Gordon, A.D., Greenberg, M., Margetson, J., Van Gael, J.：贝叶斯机器学习的测度变换器语义。载于：欧洲编程研讨会。第77–96页。Springer（2011年） 9. Boyd, S.P., Vandenberghe, L.：《凸优化》。剑桥大学出版社（2004年） 10. Carpenter, B., Gelman, A., Hoffman, M.D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., Riddell, A.：Stan：一种概率编程语言。《统计软件杂志》**76**，1–32（2017年） 11. Chaudhuri, S., Solar-Lezama, A.：平滑解释。《ACM Sigplan Notices》**45**(6)，279–291（2010年） 12. Chaudhuri, S., Solar-Lezama, A.：以可靠且稳健的方式平滑程序。载于：计算机辅助验证国际会议。第277–292页。Springer（2011年） 13. Chong, A., Menberg, K.：建筑能耗模型贝叶斯校准指南。《能源与建筑》**174**，527–547（2018年） 14. Dinh, V., Bilge, A., Zhang, C., IV, F.A.M.：概率路径哈密顿蒙特卡罗。载于：Precup, D., Teh, Y.W.（编）《第34届国际机器学习会议论文集，ICML 2017》，澳大利亚新南威尔士州悉尼，2017年8月6—11日。《机器学习研究论文集》，第70卷，第1009–1018页。PMLR（2017年），http://proceedings.mlr.press/v70/dinh17a.html 15. Gehr, T., Misailovic, S., Vechev, M.：Psi：概率程序的精确符号推断。载于：计算机辅助验证国际会议。第62–83页。Springer（2016年） 16. Huang, Z., Dutta, S., Misailovic, S.：Aqua：概率程序的自动量化推断。载于：自动化验证与分析技术国际研讨会。第229–246页。Springer（2021年） 17. Kan, R., Robotti, C.：折叠与截断多元正态分布的矩。《计算与图形统计杂志》**26**(4)，930–934（2017年） 18. Karp, R.M.：随机化算法简介。《离散应用数学》**34**(1-3)，165–201（1991年） 19. Kozen, D.：概率程序语义。载于：第20届计算机科学基础年度研讨会（SFCS 1979）。第101–114页。IEEE（1979年） 20. Kreikemeyer, J.N., Andelfinger, P.：跨条件分支自动微分的平滑方法。《IEEE Access》**11**，143190–143211（2023年） 21. van Krieken, E., Tomczak, J.M., ten Teije, A.：Storchastic：通用随机自动微分框架。载于：Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W.（编）《神经信息处理系统进展34：神经信息处理系统2021年度会议》，神经信息处理系统大会2021，2021年12月6—14日，线上举行。第7574–7587页（2021年） 22. Laurel, J., Misailovic, S.：带校正的概率程序连续化。载于：Müller, P.（编）《编程语言与系统——第29届欧洲编程研讨会，ESOP 2020，作为软件理论与实践欧洲联合会议ETAPS 2020的一部分举行》，爱尔兰都柏林，2020年4月25—30日，论文集。《计算机科学讲义》，第12075卷，第366–393页。Springer（2020年）。https://doi.org/10.1007/978-3-030-44914- 8_14, https://doi.org/10.1007/978-3-030-44914-8_14

23. Lee, E.A., Seshia, S.A.：《嵌入式系统导论：一种信息物理系统方法》。麻省理工学院出版社（2016年） 24. Lee, W., Rival, X., Yang, H.：概率程序平滑性分析及其在优化变分推断中的应用。《ACM 编程语言会议论文集》**7**(POPL)，335–366（2023年）。https://doi.org/10.1145/3571205, https://doi.org/10 3571205 25 1145. / . Lee, W., Yu, H., Rival, X., Yang, H.：迈向概率程序的经验证随机变分推断。《ACM 编程语言》**4**(POPL)，1–33（2019年） 26. Lee, W., Yu, H., Yang, H.：不可微模型的重参数化梯度。《神经信息处理系统进展》**31** （2018年） 27. Lew, A.K., Huot, M., Staton, S., Mansinghka, V.K.：ADEV：对概率程序期望值进行可靠的自动微分。《ACM 编程语言会议论文集》**7**(POPL)，121–153（2023年）。https://doi.org/10.1145/3571198, https://doi.org/10 3571198 28 1145. / . Pakman, A., Paninski, L.：用于二元分布的辅助变量精确哈密顿蒙特卡罗采样器。载于：Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q.（编）《神经信息处理系统进展 26：2013年第27届神经信息处理系统年度会议》。会议论文集，会议于2013年12月5日至8日在美国内华达州太浩湖举行。第2490–2498页（2013年） 29. Rainforth, T., Le, T.A., van de Meent, J.W., Osborne, M.A., Wood, F.：概率程序的贝叶斯优化。《神经信息处理系统进展》**29** （2016年） 30. Randone, F., Bortolussi, L., Incerto, E., Tribastone, M.：使用矩匹配高斯混合进行概率程序推断。《ACM 编程语言会议论文集》**8**(POPL)，1882–1912（2024年）。https://doi.org/10.1145/3632905, https://doi.org/10 3632905 31 1145. / . Sato, T., Aguirre, A., Barthe, G., Gaboardi, M., Garg, D., Hsu, J.：高阶概率程序的形式化验证：关于近似、收敛、贝叶斯推断和优化的推理。《ACM 编程语言》**3**(POPL)，1–30（2019年） 32. Shmarov, F., Zuliani, P.：Probreach：随机混合系统的经验证概率δ可达性（2015 8060 33 1410 ），https://arxiv.org/abs/ 。. Wick, G.C.：碰撞矩阵的求值。《物理评论》**80**(2)，268（1950年） 34. Zhang, Y., Sutton, C., Storkey, A.J., Ghahramani, Z.：离散哈密顿蒙特卡罗的连续松弛。载于：Bartlett, P.L., Pereira, F.C.N., Burges, C.J.C., Bottou, L., Weinberger, K.Q.（编）《神经信息处理系统进展 25：2012年第26届神经信息处理系统年度会议》。会议论文集，会议于2012年12月3日至6日在美国内华达州太浩湖举行。第3203–3211页（2012年） 35. Zhou, Y., Gram-Hansen, B.J., Kohn, T., Rainforth, T., Yang, H., Wood, F.：LF-PPL：面向不可微模型的低级一阶概率编程语言。载于：Chaudhuri, K., Sugiyama, M.（编）《第22届人工智能与统计国际会议，AISTATS 2019》，2019年4月16日至18日，日本冲绳那霸。机器学习研究论文集，第89卷，第148–157页。PMLR（2019年），http://proceedings.mlr.press/v89/zhou19b.html

附录

A 精确语义与 SOGA 语义

A.1 SOGA 语义的定义

为了定义 SOGA 语义，如 [30],中所述，首先介绍精确的控制流图（cfg）语义。

我们使用与第 3 节相同的标记。

节点 v 的精确语义是一个函数，它以一对 $(p,\ D)$ 为输入，并返回一对 $(p',\ D')$，两者类型相同。唯一的例外是入口节点，它不接受输入。我们将各节点类型的语义定义如下。

– 如果 v：入口，则 $[\![v]\!]_\omega = (1,\ \mathcal{N}(0,\ \mathbb{I}_n))$。– 如果 v：det，那么，令 D' 表示 $x[x_i \to e]$的分布，

$$[\![v]\!]_\omega(p, D) = \begin{cases} (\pi, D) & \text{if } arg(v) = \texttt{skip} \\ (p, D') & \text{if } arg(v) = x_i \texttt{ := } e. \end{cases}$$

– 如果 v：随机节点 且 $arg(v) = x_i \sim gm(\pi,\ \mu,\ \sigma)$，则

$$[\![v]\!]_\omega(p, D) = (p, \mathrm{Marg}_{x \setminus x_i}(D) \otimes G_{\pi,\mu,\sigma}).$$

– 如果 v：测试，令 $arg(v) = b$：

$$[\![v]\!]_\omega(p, D) = \begin{cases} (p \cdot P_D(b), D_{|b}) & cond(s_\omega(v)) = true \text{ and } P_D(b) \neq 0 \\ (p \cdot P_D(\neg b), D_{|\neg b}) & cond(s_\omega(v)) = false \text{ and } P_D(\neg b) \neq 0 \\ (0, D) & \text{else.} \end{cases}$$

– 如果 v：观测，令 $arg(v) = b$ 和 $I = \int_{\mathbb{R}^{n-1}} f_D(x[x_i \to c])d(x \setminus x_i)$：

$$[\![v]\!]_\omega(p, D) = \begin{cases} (p \cdot I, D_{|b}) & b \text{ is } x_i == c \\ (p \cdot P_D(b), D_{|b}) & b \text{ is } x_i \bowtie c \text{ and } P_D(b) \neq 0 \\ (0, D) & \text{else.} \end{cases}$$

– 如果 v：出口，$[\![v]\!]_\omega(p,\ D) = (p,\ D)$。

路径和有效程序的语义定义与可微语义相同。

A.2 SOGA 语义

随后，使用第 3 节中定义的矩匹配算子 $\mathcal{T}_{\mathrm{gm}}$ 为每个节点定义 SOGA 语义

$$[\![v]\!]^S_\omega = \begin{cases} [\![v]\!]_\omega & v\colon entry, exit \\ (\mathrm{Id}, \mathcal{T}_{\mathrm{gm}}) \circ [\![v]\!]_\omega & \text{else.} \end{cases}$$

路径和程序的语义定义与前述情况相同。

B 补充证明

B.1 定理1的证明

定理 1。 对于任意有效程序 $P(\Theta)$， $[\![P(\Theta)]\!]^{\Delta,\epsilon}$返回 $(p'(\Theta)\ D'(\Theta))$，使得 $p'(\Theta)$和 $D'(\Theta)$关于 Θ 可微。

证明。 由于有效程序的语义定义为

$$[\![P]\!]^{\Delta,\epsilon} = \sum_{\substack{\omega\in\Omega^P \\ [\![\omega]\!]^{\Delta,\epsilon}=(p,D)}} \frac{p\cdot D}{\sum_{\substack{\omega'\in\Omega^P \\ [\![\omega']\!]^{\Delta,\epsilon}=(p',D')}} p'}$$

且每条路径的语义定义为

$$[\![\omega]\!]^{\Delta,\epsilon} = [\![v_n]\!]^{\Delta,\epsilon}_{\omega} \circ \ldots \circ [\![v_0]\!]^{\Delta,\epsilon}_{\omega}$$

我们可以证明，对于每个节点 v，如果 $(p(\Theta)\ D(\Theta))$ 在 Θ 中可微，那么$[\![v]\!]^{\Delta,\epsilon}_{\omega}(p(\Theta)\ D(\Theta)\ V)$ 也可微。

不失一般性，假设 $D(\Theta)$是一个高斯分布，其均值为 $\mu(\Theta)$，协方差矩阵为 $\Sigma(\Theta)$，其中 μ和 Σ 是关于 Θ 的可微函数，并且对每个 Θ， Σ 都是非奇异的。在节点 v处，$[\![v]\!]^{\Delta,\epsilon}_{\omega}$输出一个$p'(\Theta)$和 $D'(\Theta)$，其中 D' 可能是一个高斯混合。在这种情况下，我们用 $\mu'(\Theta)$、$\Sigma'(\Theta)$表示 $D'(\theta)$任意分量的均值和协方差矩阵。对于每种节点类型，我们证明了 $p'(\Theta)$、 $\mu'(\Theta)$和 $\Sigma'(\Theta)$关于 Θ 可微，并且 $\Sigma'(\Theta)$是非奇异的。若一般情形下 $D(\Theta)$（分别地， $D'(\Theta)$）是高斯混合，则可通过对各分量应用相同的论证得出。

- $v:=det$。如果$arg(v) =$ 跳过，则可微性可直接由 $p(\Theta)$和 $D(\Theta)$的可微性得出。如果 $\mathsf{arg}(v) = x_i := e$，那么 $[\![v]\!]^{\Delta,\epsilon}_{\omega}(p,\ D,\ V) = (p,\ D',\ V)$，因此我们只需证明 D'的可微性。我们区分三种情况。
  - e是线性的， x_i 不属于 e。
    令 x' 表示子向量 $x \setminus x_i$，并考虑向量（x'， x_i， z）。赋值之前，该向量服从高斯分布，

$$\mu(\Theta) = \begin{bmatrix} \mu(\Theta)_{x'} \\ \mu(\Theta)_{x_i} \\ 0 \end{bmatrix}, \quad \Sigma(\Theta) = \begin{bmatrix} \Sigma_{x',x'} & \Sigma_{x',x_i} & 0 \\ \Sigma_{x_i,x'} & \Sigma_{x_i,x_i} & 0 \\ 0 & 0 & \epsilon^2 \end{bmatrix}.$$

    令 $\alpha(\Theta)$ 为表示 e 中线性组合的行向量，$\alpha_0(\Theta)$ 为零阶项。由于系数要么是常数，要么是参数本身， $\alpha(\Theta)$ 和 $\alpha_0(\Theta)$ 都关于 Θ 可微。令 $A(\theta)$为分块矩阵

$$A(\Theta) = \begin{bmatrix} \mathbb{I}_{n-1} & 0 & 0 \\ \alpha(\Theta) & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

赋值 $x_i := e + z$ 后，程序变量上的分布为高斯分布，其均值为 $\mu^*(\Theta)$，协方差矩阵 $\Sigma^*(\Theta)$ 由下式给出（为简洁起见，右侧省略了对 Θ 的依赖）：

$$\mu^*(\Theta) = \begin{bmatrix} \mu_{x'} \\ \alpha\mu + \alpha_0 \\ 0 \end{bmatrix}, \Sigma^*(\Theta) = A\Sigma A^T = \begin{bmatrix} \Sigma_{x',x'} & \Sigma_{x',x_i}\alpha^T & 0 \\ \alpha\Sigma_{x_i,x'} & \alpha\Sigma_{x',x'}\alpha^T + \epsilon^2 & \epsilon^2 \\ 0 & 0 & \epsilon^2 \end{bmatrix}.$$

考虑关于 x 的边缘分布时，根据高斯分布[6], 的性质，只需考虑由 $x =$ 索引的子矩阵（x'，x_i）。由于 μ，Σ，α 和 α_0 关于 Θ 可微，因此 μ^*_x 和 $\Sigma^*_{x,\ x}$ 也可微。不过，要证明所得概率密度函数关于 Θ 可微，我们仍需证明 $\Sigma^*_{x,\ x}$ 非奇异。为此，可以利用舒尔补 [9] 计算行列式：

$$|\Sigma^*_{x,x}| = \alpha\Sigma_{x',x'}\alpha^T + \epsilon^2 - \alpha\Sigma_{x_i,x'}\Sigma^{-1}_{x',x'}\Sigma_{x',x_i}\alpha^T = \epsilon^2 > 0.$$

* e 是关于 e 的线性函数，其中 x_i。对于这种情况，计算过程与上文类似，只是不需要变量 z。
* e 是非线性的，即 $x_i := x_j x_k$。赋值后，均值和协方差矩阵关于 Θ 的可微性，源于这些量可通过伊瑟利斯定理 [33] 计算为 μ（Θ）和 Σ（Θ）的多项式函数。同样，要证明概率密度函数的可微性，我们需要证明新的协方差矩阵 Σ^*（Θ）非奇异。令 x' 表示子向量 $x' = x \setminus \{x_i, x_j, x_k\}$。我们证明 $\tilde{x} =$（x'，$x_j x_k$）的协方差矩阵非奇异。对于向量（x_j，x_k，$x_k x_k$），证明过程类似。这两点共同证明 Σ^* 非奇异。我们采用反证法，假设 $\tilde{}$ 协方差矩阵 Σ 的 $\tilde{x}$ 是奇异的。于是存在一个非零向量 c，使得 $c^T\tilde{\Sigma}c = 0$。这意味着 c^T（$\tilde{x} - \mathbb{E}[\tilde{x}]$）的方差为 0。但这意味着 c^T（$\tilde{x} - \mathbb{E}[\tilde{x}]$）是确定性的，尤其是等于其均值 0。因此，可以将 c 划分为 a、b，使得：

$$a(x_j x_k - \mathbb{E}[x_j x_k]) + b^T(x' - \mathbb{E}[x']) = 0. \tag{3}$$

现在我们利用高斯向量的一个已知性质：给定其他分量时，某个分量的条件期望是一个仿射函数 [6]。这意味着，对于某些向量 α，β，γ，有 $\mathbb{E}[x'|x_j, x_k] = \alpha x_j + \beta x_k + \gamma$。因此，将 (3) 条件化于 x_j 和 x_k，并取均值，我们发现存在常数 α，β，γ，δ，使得：

$$\alpha x_j x_k + \beta x_j + \gamma x_k + \delta = 0.$$

如果 $\alpha = 0$ 等式化简为 $\beta x_j + \gamma x_k + \delta = 0$，这将意味着原协方差矩阵 Σ退化，从而导致矛盾。如果 $\alpha \neq 0$ 我们可以改写为 $x_j = \frac{-\gamma x_k - \delta}{\alpha x_k + \beta}$。由于根据假设，$x_k$ 非退化，因此分母几乎处处不为零。但根据

高斯向量的性质， $\mathbb{E}[x_j|x_k]$ 必须是仿射函数，而这与 x_j 是 x_k的有理函数这一事实相矛盾。因此， Σ^*不可能是奇异的。

- v: 随机节点。在这种情况下，可微性由 $D(\Theta)$ 的可微性以及 $gm(\pi, \mu, \sigma)$ 的可微性保证，后者的分量均具有非零标准差。
- v: 观测。我们区分三种情况。
  - b 是 $x_i == c$ 和 $x_i \notin V$。可微性源于 $\mathcal{N}(c, \epsilon)$ 的概率密度函数以及 $D_{|b}$ 的可微性；这一点由其矩是 $\mu(\Theta)$ 和 $\Sigma(\Theta)$ 的多项式这一事实保证。此外，如果 $\Sigma(\Theta)$ 非退化，那么 $D_{|b}$ 的协方差矩阵也非退化。
  - b 在所有其他情况下都是 $x_i \bowtie c$ 。在这种情况下，谓词被转换为 $b' = \mathcal{B}_\epsilon(x_i \bowtie c, V)$。由于 D 非退化，且 $\mathcal{B}_\epsilon(b, V)$ 定义的所有条件都具有正测度，因此 $P_D(b')$ 可微且不为 0（它使用 D 的累积分布函数计算，而该函数可微）。截断分布 $dist_{|b'}$ 的矩根据 [17] 中的公式计算，作为 $\mu(\Theta)$、 $\Sigma(\Theta)$ 以及截断区间端点（除 x_i 之外所有变量对应的端点为 $-\infty$ 和 $+\infty$，而 x_i 的端点涉及 c）的函数。由于 D 非退化， [17] 中的公式关于其参数可微。因此，$dist_{|b'}$ 的矩关于 Θ 可微。此外，我们还需要证明新的协方差矩阵是非奇异的；这可由初始协方差矩阵非奇异且截断是在正测度集合上进行这一事实推出。

- v: 测试。我们令$arg(v) = b$ and $b' = \mathcal{B}_\epsilon(b, V)$。条件分布$D_{|b'}$ （分别为 $\neg b'$）的概率 $P_D(b')$（分别为 $\neg b'$）及其矩的计算，使用了与观测 节点情形相同的公式 [17], 。因此，概率和截断矩关于 Θ均可微

- v: 入口 和 v: 出口 结论显然成立。

B.2 定理 2 的证明

定理 2。 假设 P 是一个有效程序，并且在计算 $[\![P]\!]^{\Delta,\epsilon}$时始终一致地应用 $[\![P]\!]^{\Delta,\epsilon}\mathcal{B}_\epsilon$ 。那么$\lim_{\epsilon\to 0}[\![P]\!]^{\Delta,\epsilon} = [\![P]\!]^S$ （此处所说的分布收敛是指依分布收敛）。

证明。 回顾一下，对于 $[\![\cdot]\!]^{\Delta,\epsilon}$ 和 $[\![\cdot]\!]^S$ ， P 的语义都定义为 Ω^P中各条路径语义的混合分布。由于我们只考虑有限程序，程序语义的收敛可由路径语义的收敛推出。首先，我们只考虑满足 $[\![\omega]\!]^S = (p, D)$ 且 $p > 0$ 的路径 $\omega \in \Omega^P$ 。考虑一条这样的路径，其可微语义为 $[\![\omega]\!]^{\Delta,\epsilon} = [\![v_n]\!]^{\Delta,\epsilon}_\omega \circ \ldots \circ [\![v_0]\!]^{\Delta,\epsilon}_\omega$。我们通过对 n 作归纳法，证明其收敛到 $[\![\omega]\!]^S = [\![v_n]\!]^S_\omega \circ \ldots \circ [\![v_0]\!]^S_\omega$。

对于 $n = 1$ ，我们只有一个入口 和一个出口 节点，因此结论显然成立。

假设在节点 v_{i-1} 之前，有 $[\![v_{i-1}]\!]^{\Delta,\epsilon}_{\omega} \circ \ldots \circ [\![v_0]\!]^{\Delta,\epsilon} = (p_\epsilon, D_\epsilon, V)$ 和 $[\![v_{i-1}]\!]^{S}_{\omega} \circ \ldots \circ [\![v_0]\!]^{S}_{\omega} = (p, D)$，且 $\lim_{\epsilon\to 0}(p_\epsilon, D_\epsilon) = (p, D)$。我们证明存在 $[\![v_i]\!]^{\Delta,\epsilon}_{\omega}(p_\epsilon, D_\epsilon, V) = (p'_\epsilon, D'_\epsilon, V')$，使得 $\lim_{\epsilon\to 0}(p'_\epsilon, D'_\epsilon) = (p', D') = [\![v_i]\!]^{S}_{\omega}(p, D)$。不失一般性，假设 D_ϵ（分别为 D）是均值为 μ_ϵ（分别为 μ）、协方差矩阵为 Σ_ϵ（分别为 Σ）的高斯分布。作用于高斯分布 D_ϵ时，$[\![v]\!]^{\Delta}$和 $[\![v]\!]^{S}$ 通常会产生具有固定分量数的高斯混合；对于所有 ϵ 的取值，这在两种语义中都是其语义。在这种情况下，我们用 μ'_ϵ（分别为 μ）和 Σ'_ϵ（分别为 Σ'）表示 D'_ϵ（分别为 D'）任一分量的均值和协方差矩阵，并假定在每个分布中选择对应的分量。通过逐分量应用相同的论证，证明可推广到高斯混合。注意，如果 D'_ϵ和 D 都是高斯分布，只需证明 $\mu'_\epsilon \to \mu$ 和 $\Sigma'_\epsilon \to \Sigma$，因为根据特征函数的收敛性，这意味着依分布收敛。

我们分别考虑 v_i 的可能类型。

– v_i: det。如果 $arg(v) =$ 跳过，则结论显然成立。如果 $arg(v) = x_i :=e$，我们只需证明 D'_ϵ 收敛到 D'。假设 e 是线性的，其系数向量为 α，零阶项为 α_0。令 x' 表示向量 $x \setminus x_i$，这样 x 可表示为 (x', x_i)。于是：

$$\mu'_\epsilon = \begin{bmatrix} [\mu_\epsilon]_{x'} \\ \alpha\mu_\epsilon + \alpha_0 \end{bmatrix}$$

$$\Sigma'_\epsilon = A\Sigma_\epsilon A^T = \begin{cases} \begin{bmatrix} [\Sigma_\epsilon]_{x',x'} & [\Sigma_\epsilon]_{x',x_i}\alpha^T \\ \alpha[\Sigma_\epsilon]_{x_i,x'} & \alpha[\Sigma_\epsilon]_{x',x'}\alpha^T + \epsilon^2 \end{bmatrix} & x_i \text{ not in } e \\ \begin{bmatrix} [\Sigma_\epsilon]_{x',x'} & [\Sigma_\epsilon]_{x',x_i}\alpha^T \\ \alpha[\Sigma_\epsilon]_{x_i,x'} & \alpha[\Sigma_\epsilon]_{x',x'}\alpha^T \end{bmatrix} & x_i \text{ in } e \end{cases}$$

对于 $\epsilon \to 0$，显然可以验证 μ'_ϵ 和 Σ'_ϵ 收敛到 μ'、Σ'。如果 e 是非线性的，则 μ'_ϵ 和 Σ'_ϵ 可利用伊瑟利斯定理 [33] 得到，它们是关于 μ_ϵ 和 Σ_ϵ 的多项式函数。由于多项式是连续的，极限得以保持。

– v_i: 随机节点。同样，我们只需证明 D_ϵ 收敛，但这可直接由 $G_{\pi,\mu,\sigma'}$ 收敛到 $G_{\pi,\mu,\sigma}$ 得出（回想一下，当 $\sigma_j > 0$ 且 ϵ 时为 $\sigma'_j = \sigma_j$，否则则为.....）。

– v_i: 观测。如果 $\text{arg}(v) = x_i == c$ 且 $x_i \notin V$，则由 $\lim_{\epsilon\to 0}\mathcal{N}(c, \epsilon) = \mathcal{N}(c, 0)$ 可立即得出收敛。令 b 为剩余情形之一中的 $x_i \bowtie c$。我们考虑两种不同的情况。若 Σ_{x_i}，$_{x_i} \neq 0$，则有 $P_D(\partial b) = 0$，其中 ∂b 表示集合 b的边界。在这种情况下，b 称为 D的连续集。此外，由于 V仅包含在原始程序 $x_i \notin V$中为离散的变量，且 $\mathcal{B}_\epsilon(b, V) = b$。因此，收敛问题归结为证明 $P_{D_\epsilon}(b) \to P_D(b)$ 且 $(D_\epsilon)_{|b} \to D_{|b}$。由于 b 是 D 的连续集，根据 Portmanteau 引理 [4]，这两个极限都成立。若 Σ_{x_i}，$_{x_i} = 0$，则 $\text{Marg}_{x_i}(D)$ 在 x_i^0处有一个点质量。此外，由于我们是在假设 $[\![\omega]\!]^{\Delta}$的条件下进行讨论，$_\epsilon = (p_\omega, D_\omega)$，其中 $p_\omega > 0$，那么 $P_D(b) > 0$。因此，

$x_i^0 \in (b)_i$，其中 $(b)_i$ 是 b 在第 i 个坐标上的投影。因此，$P_D(b) = 1$ 且 $D_{|b} = D$。对于 D_ϵ，有 $(\Sigma_\epsilon)_{x_i,x_i} \to 0$ 且 $(\mu_\epsilon)_{x_i} \to x_i^0$。现在令 $b_\epsilon = \mathcal{B}_\epsilon(b, V)$，并令 $(b_\epsilon)_i$ 为 b_ϵ 在第 i 个坐标上的投影。那么 $x_i^0 \in (b_\epsilon)_i$。此外，由于 $\mathcal{B}_\epsilon$ 被一致地应用，$\delta_\epsilon(\epsilon)/\sqrt{\Sigma_\epsilon} \to +\infty$，这意味着 $\lim_{\epsilon\to 0} P_{\mathrm{Marg}_{x_i}(D_\epsilon)}((b_\epsilon)_i) = 1$。但于是 $\lim_{\epsilon\to 0} P_{D_\epsilon}(b_\epsilon) = 1 = P_D(b)$。对于分布，我们可以使用由弱收敛诱导的度量来证明该极限，该度量称为列维-普罗霍罗夫度量，记作 $d_{LP}(\cdot,\cdot)$。然后，

$$d_{LP}((D_\epsilon)_{|b_\epsilon}, D) \le d_{LP}((D_\epsilon)_{|b_\epsilon}, D_\epsilon) + d_{LP}(D_\epsilon, D).$$

右侧的第一项趋于零，因为 $\delta_\epsilon(\epsilon)/\sqrt{\Sigma_\epsilon} \to +\infty$；而第二项根据假设趋于零。

– v_i：测试。在这种情况下，证明过程与观测情形类似。

我们现在已经证明，对于概率为正的路径，$[\![\omega]\!]^{\Delta,\epsilon} \to [\![\omega]\!]^S$。对于满足 $[\![\omega]\!]^S = (0, D)$ 的路径，我们只需证明 $p_\epsilon \to 0$。设 v_i 为路径中第一个满足 $[\![v_i]\!]^S(p, D) = (p', D')$ 且 $p' = 0$ 的节点。那么截至该节点，可微语义给出的输出三元组为 $(p_\epsilon, D_\epsilon, V)$，满足 $p_\epsilon \to p$, $D_\epsilon \to D$，我们要证明 $p'_\epsilon \to 0$。我们知道 v_i 必定是观测节点或 test 节点。假设它是观测节点（另一种情况可用相同方式处理）。令 $\mathrm{arg}(v_i) = b = x_i \bowtie c$。我们可以排除 $\bowtie$ 是 $==$ 且 $x_i \notin V$ 的情况，因为这种情况不可能产生概率 0。在其他情况下，我们可以再次区分两种情形。若 $\Sigma_{x_i,x_i} \neq 0$，则 b 是 $D, \mathcal{B}_\epsilon(\exp V) = b$ 的连续集，并且由 Portmanteau 引理可保证收敛。若 $\Sigma_{x_i,x_i} = 0$，边缘分布 $\mathrm{Marg}_{x_i}(D)$ 是点质量 $x_i^0 \notin (b)_i$。而 D_ϵ 则是将其概率质量集中在 x_i^0 附近的高斯分布。再次利用算子 $\mathcal{B}_\epsilon$ 的一致性，我们可以再次证明 $P_{D_\epsilon}(b_\epsilon) \to 0$。

C DeGAS

本节概述工具 DeGAS。该工具实现了前文介绍的可微语义。DeGAS 使用 PyTorch 实现，可通过 Python 编程调用。原型实现可见于 url-anonymized。

DeGAS 实现的算法如算法 1 所示。它以一个 .soga 文件 输入_文件作为输入，该文件按照第 3.1 节的语法对程序进行编码；以一个字典 参数_字典 作为输入，其键为参数名称（也出现在程序中，并带有下划线前缀），值为 requires_grad $= True$ 的 PyTorch 张量；以一个损失函数 损失_函数 作为输入，用于将分布映射到真实值；以平滑参数 $\epsilon > 0$ 作为输入；以及以优化步数 步数作为输入。如有需要，可以在优化循环中直接将各个参数限制在其相应的区间域内，从而指定对单个参数的约束。

算法 1 DeGAS(输入_文件，参数_字典，损失_函数 ϵ，步数):

1: cfg = 平滑_*cfg(*输入_文件 ϵ*) 2:* 优化器 = 导入_优化器_来自_*torch() 3:* 对 _ 在 *range(*步数*)* 中执行 *4:* 优化器*.zero_grad() 5:* 当前_分布 = 计算_*degas(cfg,*参数_字典*) 6:* 损失 = 损失_函数*(*当前_分布*) 7:* 损失*. backward() 8:* 优化器*.step() 9:* 结束循环 *10:* 返回 参数_字典

算法首先通过 smooth_cfg 构建程序的控制流图（控制流图）。这一步还会准备计算可微语义所需的信息：例如，赋值语句 x := c 会转换为 x := $c + gm([1], [0], [\epsilon])$，其他指令也会进行类似的改写。此外，还会预先计算每个节点上的平滑变量集合 V。生成的对象 控制流图 编码了程序的控制流图，并包含评估 $[\![\cdot]\!]^{\Delta}$, $^{\epsilon}$所需的全部信息。

构建控制流图后，初始化一个来自 PyTorch 的基于梯度的优化器（第 2 行），并开始优化循环（第 3 行）。每一步中，compute_degas 会计算后验分布（第 5 行），随后评估损失（第 6 行），并通过反向传播更新参数 _ 字典中的参数（第 7–8 行）。所有迭代完成后，返回优化参数。

DeGAS 的计算成本。 在计算程序的语义时，DeGAS 与 SOGA 共享相同的控制流图结构。为保证可微性而引入的操作不会改变渐近计算成本，因为计算程序的可微语义涉及的基本操作和函数调用与 SOGA 中描述的相同 [30]。因此，总体复杂度仍为 $O(|V|2^{|T|}n^4)$，其中 $|V|$ 表示控制流图中的节点数， $|T|$ 表示测试 节点和观测 节点的数量， n 是程序中的变量数；这里假设随机节点中使用的所有分布都已移至初始分布中。

与 SOGA 类似，可以在 DeGAS 中引入剪枝策略，以限制混合成分的数量并控制计算开销。不过，在本文展示的案例研究中，这并非必要。

D 实验评估细节

D.1 收敛实验

为了展示收敛在实际中的表现，我们考虑以下三个程序：

$$P_1\colon x := 0; \texttt{observe}\ x \geq 0;$$
$$P_2\colon x := 0; \texttt{observe}\ x > 0;$$
$$P_3\colon x \sim gm([0.5, 0.5], [0, 1], [0, 0]); \texttt{observe}\ x == 0;$$

在下表中，我们报告了当 ϵ 减小时，持续应用 $\mathcal{B}_\epsilon$ 并选择 $\delta_\epsilon = \sqrt{\epsilon}$ 时 p、μ 和 σ 的结果。对于 $\epsilon = 10^{-4}$，三个输出的值均与 SOGA 的值相同，精确到小数点后三位。

	P_1			P_2			P_3		
	p	μ	σ	p	μ	σ	p	μ	σ
$\epsilon = 10^{-1}$	0.9992	0.0002	0.0995	0.0008	0.3434	0.0256	0.4992	0.0000	0.0991
$\epsilon = 10^{-2}$	1.0000	0.0000	0.0100	0.0000	0.0000	0.0100	0.5000	0.0000	0.0100
$\epsilon = 10^{-3}$	1.0000	0.0000	0.0010	0.0000	0.0000	0.0010	0.5000	0.0000	0.0010
$\epsilon = 10^{-4}$	1.0000	0.0000	0.0001	0.0000	0.0000	0.0001	0.5000	0.0000	0.0001
SOGA	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.5000	0.0000	0.0000

表 4：随着 ϵ 的值减小，DeGAS 和 SOGA 的结果。

通过测试 DeGAS 在改变 ϵ 值时对概率程序参数进行综合，我们得到了相符的结果；但在某些以灰色突出显示的情况下，$\epsilon = 0.1$ 除外。这些程序更为复杂，涉及多个变量和嵌套的条件分支。

模型	$\epsilon = 0.1$	$\epsilon = 0.01$	$\epsilon = 0.001$	$\epsilon = 0.0001$
伯努利	0.001	0.001	0.001	0.001
入室盗窃	1.6857	1.6857	1.6857	1.6857
ClickGraph	0.375	0.085	0.086	0.086
临床试验	3.463	0.659	0.657	0.657
草	0.695	0.036	0.036	0.036
谋杀之谜	0.203	0.203	0.203	0.203
SurveyUnbiased	0.008	0.008	0.008	0.008
TrueSkills	0.003	0.003	0.003	0.003
TwoCoins	0.688	0.688	0.688	0.688
AlterMu	0.312	0.312	0.312	0.312
AlterMu2	0.096	0.096	0.096	0.096
NormalMixtures	0.386	0.386	0.386	0.386

表5：不同 ϵ 值下程序参数综合的性能

D.2 PID

比例-积分-微分（PID）由以下动力学描述。

$$
\begin{aligned}
d_k &= \pi - \theta_k \\
T_k &= s_0 d_k + s_1 \omega_k + s_2 I_k \\
I_{k+1} &= 0.9 I_k + d_k \Delta t \\
\omega_{k+1} &= \omega_k + \frac{\Delta t}{10} T_k + \mathcal{N}(0, \sigma_\omega^2) \\
\theta_{k+1} &= \theta_k + \frac{\Delta t}{2}(\omega_k + \omega_{k+1}) + \mathcal{N}(0, \sigma_\theta^2)
\end{aligned}
$$

DeGAS 优化的结果如图7所示。

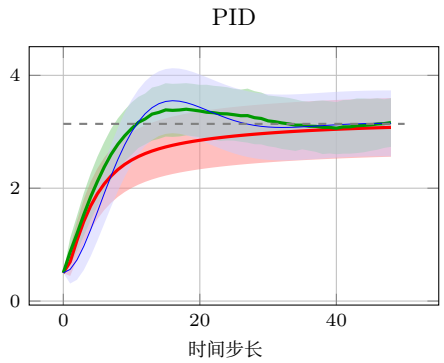


图7：从理想轨迹进行 PID 优化，同时保持稳定动态，使用$\theta = 3.14$得到红色轨迹，起点为蓝色轨迹。绿色轨迹为通过变分推断获得的轨迹。

D.3 超参数程序优化

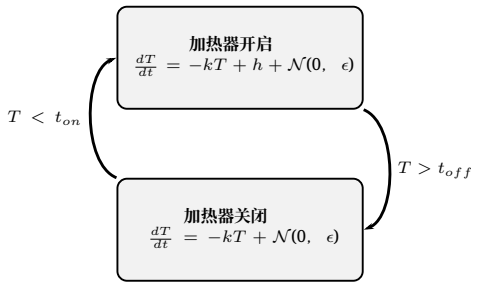
在表6中，我们报告了在通过数据似然优化进行概率程序参数综合的实验中，经调优选出的超参数。

模型	VI		MCMC				DeGAS	
	学习率	步数	样本预热	R-hat统计量	有效样本量		学习率	步数
伯努利	0.05	247	500	50	1.005	802	0.01	248
入室盗窃	0.05	1000	500	50	1.003	1008	0.005	295
ClickGraph	0.001	1000	500	50	1.018	333	0.2	216
临床试验	0.2	1000	500	50	1.021	146	0.001	500
草	0.01	1000	500	50	1.004	1444	0.1	500
谋杀之谜	0.001	1000	500	50	1.001	1036	0.01	239
SurveyUnbiased	0.01	675	500	50	1.001	1382	0.01	500
TrueSkills	0.2	1000	t.o.	t.o.	t.o.	t.o.	0.2	367
TwoCoins	0.05	1000	500	50	1.015	279	0.001	337
AlterMu	0.0005	1000	t.o.	t.o.	t.o.	t.o.	0.05	235
AlterMu2	0.001	1000	1500	50	1.040	81	0.2	228
正态混合	0.0005	1000	500	50	1.018	457	0.01	500
PID	0.05	1000	n.c.	n.c.	n.c.	n.c.	0.2	1000

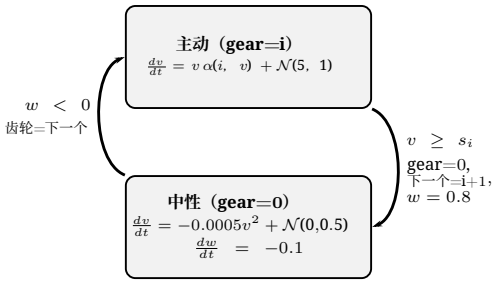
表6：用于概率程序参数综合的调优超参数（参见第5节）。

D.4 信息物理系统

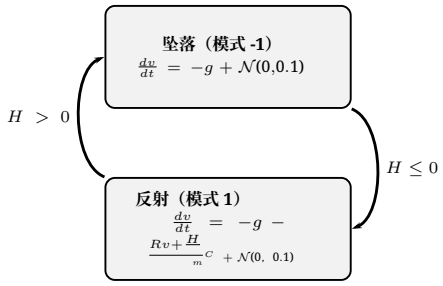
我们展示了研究中考察的三个信息物理系统，呈现其依赖模式的动态、切换条件及关键模型参数。



恒温器 对于相对于轨迹的优化，我们考虑 $k = 0.01$、 $h = 0.5$，初始值为 $T = 16$。在可达性优化中（其中 $k = 1$、 $h =$为30，初始值为 $T \sim \mathcal{N}(30,\ 1)$），在假设不同时间步长上的事件相互独立的情况下，损失函数变为：在三个时间步长上计算的、处于感兴趣区域内的累积分布函数的乘积，再乘以处于“ON”模式的概率密度函数。



变速箱动力学定义为 $\alpha(i,\ v) = \frac{1}{1+\frac{(v-p_i)^2}{25}}$，其中$p_i$为(5，15，25，40，60)。在可达性优化中，损失函数被设为边缘 $CDF(v_i \leq 16)\ \forall i$ ，以最大化汽车始终保持低于16的速度的概率。



受控弹跳球。质量为$m = 7$的球的位置根据$\frac{dH}{dt} = v + \mathcal{N}(0, \epsilon)$相应变化，初始高度为 $H_0 \sim \mathcal{N}(9,\ 1)$，其中$g = 9.81$、$m = 7$。可达性优化的目标表示为 $1 - CDF(H_i \geq 7)$，其中 H_i 是球弹跳后达到的最大高度。

D.5 可达性性质分析

在可达性性质框架下，优化的目标是根据 DeGAS 语义计算的。为了定性了解这与真实系统及 SOGA 语义的关系，以及误差如何在较长的执行轨迹中累积，我们考虑迭代次数最多的示例——受控弹跳球，并绘制 DeGAS 和 SOGA 计算的平均轨迹，以及从真实系统中随机采样得到的轨迹。图 8 显示，即使经过大量迭代，平滑也不会造成显著差异。此外，尽管高斯近似会引入误差，SOGA 和 DeGAS 估计的平均轨迹与模拟轨迹仍高度一致。

D.6 与 SOGA 的比较

对于表1中的每个程序，我们比较了在参数值固定时，DeGAS 和 SOGA 计算得到的目标变量均值，以评估平滑引入的误差。表7所报告的结果表明，对于我们的实验设置所采用的平滑超参数（ϵ =$1e-3$，$\delta_\epsilon = \sqrt{\epsilon}$），误差在小数点后三位以内为零。 –

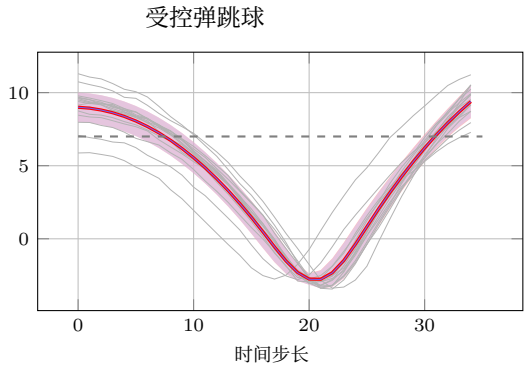


图 8：DeGAS 语义（红色）与 SOGA 语义（蓝色）及真实系统（灰色轨迹）的对比，均采用通过 DeGAS 优化可达性性质获得的优化参数

模型	目标值	DeGAS	SOGA	参数
伯努利	y	0.810	0.810	$p = 0.794$
入室盗窃	$called$	0.237	0.237	$pe = 0.063; pb = 0.132$
ClickGraph	sim	0.617	0.617	$p = 0.646$
临床试验	$ytreated$	0.789	0.789	$pe = 0.054; pc = 0.797; pt = 0.778$
草	$wetGrass$	0.666	0.666	$pcloudy = 0.546;\ p1 = 0.705; p2 = 0.910; p3 = 0.864$
谋杀之谜	$withGun$	0.716	0.716	$palice = 0.138$
SurveyUnbiased	$ansb2$	0.713	0.713	$bias1 = 0.778; bias2 = 0.723$
TrueSkills	$perfC$	119.021	119.021	$pa = 105.990;\ pb = 89.931; pc = 119.021$
TwoCoins	$both$	0.158	0.158	$first = 0.367; second = 0.367$
AlterMu	y	4.530	4.530	$p1 = 0.617; p2 = 0.617; p3 = -3.388$
AlterMu2	y	3.553	3.553	$muy = 3.553; vary = 2.838$
正态混合	y	−3.192	−3.192	$theta = 0.718; p1 = -4.788; p2 = 0.627$
PID	v	−0.008	−0.008	$s0 = 46.500; s1 = -23.496; s2 = -0.495$

表7：对于表1中的程序和固定参数值，比较由 DeGAS 和 SOGA 计算得到的所选目标变量的均值。