
VaSST: Variational Inference for Symbolic Regression using Soft Symbolic Trees

Somjit Roy^{*1}

Pritam Dey¹

Bani K. Mallick¹

¹ Department of Statistics, Texas A&M University, College Station, Texas, USA

Abstract

Symbolic regression (SR) has gained recent traction in AI-driven scientific discovery for learning closed-form physical laws. Yet existing methods are dominated by heuristic search or data-intensive approaches that often assume low-noise regimes and lack principled uncertainty quantification, while fully probabilistic SR formulations remain scarce. We introduce a scalable probabilistic framework for SR, **VaSST**, based on *variational inference*. **VaSST** uses *soft symbolic trees*, a continuous relaxation of symbolic expression trees in which discrete operator and feature assignments are replaced by probability distributions over allowable components. This transforms combinatorial symbolic search through an astronomically large expression space into efficient gradient-based optimization while preserving a coherent probabilistic interpretation. The learned soft representations induce posterior distributions over symbolic structures, enabling uncertainty quantification across plausible symbolic forms through posterior-aware symbolic model selection. On simulated experiments and the *Feynman Symbolic Regression Database*, **VaSST** achieves strong structural recovery and predictive accuracy compared to state-of-the-art competing SR methods.

1 INTRODUCTION

Symbolic regression for scientific discovery. *Scientific machine learning* has increasingly emerged as a powerful paradigm for integrating domain knowledge with data-driven modeling, enabling advances across materials science [Butler et al., 2018], climate and weather prediction [Zhang et al., 2025], biology [Boadu et al., 2025], and

physics [Raissi et al., 2019]. A central objective in these domains is the discovery of explicit governing equations that encode mechanistic structure rather than merely predictive relationships. In this setting, *symbolic regression* (SR) plays a pivotal role [Makke and Chawla, 2024]. Unlike classical prediction-centric regression methods [Tibshirani, 1996, Rasmussen and Williams, 2006, Chipman et al., 2010], SR operates directly over functional forms to recover concise, interpretable mathematical expressions from experimental data. By identifying explicit equations underlying complex scientific phenomena, SR has enabled sparse discovery of nonlinear dynamical systems [Brunton et al., 2016], accelerated materials design [Wang et al., 2024], and uncovered fundamental scientific laws [Schmidt and Lipson, 2009].

Related work. Existing SR methods can be broadly categorized into evolutionary, machine learning-based, and Bayesian approaches. Classical SR is dominated by genetic programming and related heuristic search algorithms [Fortin et al., 2012, Stephens, 2016, Burlacu et al., 2020, Cranmer, 2023], which explore expressions through stochastic evolution but often suffer from high computational complexity, sensitivity to initialization, and the generation of overly complex formulas [Korns, 2011]. Machine learning-driven methods cast SR as a sequential decision-making problem, learning to generate grammar rules, tree traversals, or executable symbolic strings via neural architectures [Udrescu and Tegmark, 2020, Petersen et al., 2021, Broløs et al., 2021, Kamienny et al., 2022, Xu et al., 2024]. Although these methods enhance scalability and predictive performance, they remain inherently search-driven and computationally challenging due to the NP-hard nature of SR [Virgolin and Pissis, 2022]. Moreover, their effectiveness is often contingent on large training datasets and low-noise regimes, as demonstrated in Section 5. More recent machine learning-centric SR modules use pretrained large language models (LLMs) to guide symbolic expression generation, refinement, or scientific reasoning [Grayeli et al., 2024, Shojaee et al., 2025, Wang et al., 2025, Ying et al., 2025]. While these approaches provide a flexible interface for incorporating

^{*}Corresponding author: <sroy_123@tamu.edu>.

domain knowledge, they can be highly sensitive to prompt design and model choice, and may produce syntactically invalid or unstable expressions; see [Appendix P](#) for a representative example. The preceding review highlights the necessity for fully probabilistic SR formulations.

Scientific expressions possess an inherent hierarchical and compositional structure that aligns naturally with tree representations. It is important, however, to distinguish such representations from *decision trees* which recursively partition the covariate space using split rules and assign local predictions to terminal regions [Breiman et al., 1984, Chipman et al., 1998, Denison et al., 1998, Chipman et al., 2010]. In contrast, tree representations of scientific expressions encode recursive compositions of mathematical operators (e.g., $+$, $\times$, $\exp$, $\sin$; assigned to internal nodes) with primitive data features (assigned to leaves) [Bartlett et al., 2023, Roy et al., 2025]; see [Figure 1](#). Among existing tree-based SR formulations, Bayesian Machine Scientist (BMS) [Guimera et al., 2020] represents an important step toward model-based equation discovery. However, it employs an ad hoc structural prior built from corpus-parsing based on a priori knowledge, and performs inference using Metropolis-Hastings (MH) proposals over discrete symbolic tree structures. Given the highly multimodal and combinatorial posterior landscape, such local MH updates can exhibit poor mixing in complex discrete spaces [Bhamidi et al., 2008, Łatuszyński et al., 2025], leading to slow convergence and inefficient exploration of the symbolic expression space. Similarly, Bayesian Symbolic Regression (BSR) [Jin et al., 2020] adopts a tree-based partial Bayesian formulation, where it uses plug-in ordinary least squares estimates for the outer regression coefficients. This incomplete parameter uncertainty propagation, combined with local stochastic MH structural updates, further impedes effective traversal of the symbolic expression space and frequently produces overly complicated output symbolic structures, as evidenced in [Section 5](#). Finally, the reliance of these existing Bayesian SR approaches on Markov chain Monte Carlo-based discrete structural exploration can limit their scalability beyond small- to moderate-sized data settings.

Our contributions. Motivated by these limitations, we propose *Variational Inference for Symbolic Regression using Soft Symbolic Trees* (VaSST), a *variational inference* framework for SR that combines principled Bayesian modeling with improved computational scalability over existing probabilistic SR methods. Variational inference recasts Bayesian inference as an optimization problem [Blei et al., 2017], offering a scalable alternative to traditional Monte Carlo-based approaches for modern data-intensive settings [Jordan et al., 1999, Wainwright and Jordan, 2008, Graves, 2011]. However, a naïve application of variational inference over the discrete structural space of symbolic expressions results in a combinatorial optimization problem that negates these scalability benefits [Williams, 1992, Koza, 1994].

To overcome this challenge, VaSST uses a novel latent representation of symbolic expressions using *soft symbolic trees*, in which discrete operator and feature assignments are replaced by probability distributions over all allowable operators and features. This relaxation transforms the discrete structural search into a continuous optimization problem, enabling efficient gradient-based exploration of the symbolic expression space through black-box variational inference [Ranganath et al., 2014, Kucukelbir et al., 2017, Giordano et al., 2024]. Although this relaxation is similar in spirit to the soft tree formulation used in *Variational Regression Trees* (VaRT) [Salazar, 2023], the relaxed objects and inferential goals here are fundamentally different. While VaRT performs soft routing over data-partitioning trees, thus relaxing discrete decision splits, VaSST relaxes discrete assignments of operators and features defining a tree-structured scientific expression. Thus, its soft tree representation serves as a differentiable surrogate for symbolic expression evaluation, rather than as a probabilistic partitioning rule. For a detailed comparison of VaRT and VaSST, refer to [Appendix M](#).

The learned soft symbolic trees naturally define probability distributions over symbolic expressions. Thus, post-optimization, structurally interpretable expressions are sampled from these learned representations and ranked using a posterior-aware evidence score. The top candidates are then retained yielding an Occam’s window-based [Madigan and Raftery, 1994] posterior summary of high-support symbolic explanations. Finally, a primary goal in scientific equation discovery is to favor structural parsimony in accordance to the Occam’s razor principle [Jefferys and Berger, 1992]. VaSST achieves this by employing a depth-dependent regularizing tree prior that penalizes overly complex sampled symbolic expressions.

Through extensive experiments, we demonstrate the superior performance of VaSST relative to a range of state-of-the-art SR methods from SRBench [La Cava et al., 2021, Imai Aldeia et al., 2025, La Cava, 2026], where it effectively balances structural discovery and predictive accuracy while maintaining computational stability. The Python implementation of VaSST is available at [Roy and Dey \[2026\]](#).

2 SCIENTIFIC EXPRESSIONS AS SYMBOLIC TREES

Scientific expressions can be recursively constructed by combining primary features (e.g., x_1, x_2) and elementary mathematical operators (e.g., $\exp, \sin, +, \times$). We denote by $\mathbf{x} = (x_1, \dots, x_p)^\top$, the vector of p primary features and by $\mathbf{O} = \mathbf{O}_u \cup \mathbf{O}_b$, the set of allowed mathematical operators, where $\mathbf{O}_u$ and $\mathbf{O}_b$ contain the unary and binary mathematical operators, respectively. Typical choices in scientific modeling include $\mathbf{O}_u = \{\sin, \cos, \exp, \log, ^2, ^3\}$ and $\mathbf{O}_b = \{+, \times, -, /\}$ [Udrescu and Tegmark, 2020].

Let $\mathcal{S}$ denote the set of admissible feature-operator compositions. Any symbolic expression $f \in \mathcal{S}$ is defined recursively as one of, (a) *primitive expression*: $f(\mathbf{x}) = x_m$, where $m \in \{1, \dots, p\}$; (b) *binary composition*: $f(\mathbf{x}) = b(f_1(\mathbf{x}), f_2(\mathbf{x}))$, where $f_1, f_2 \in \mathcal{S}$ and $b \in \mathbf{O}_b$, e.g., $x_1 + x_2$; and (c) *unary composition*: $f(\mathbf{x}) = u(\tilde{f}(\mathbf{x}))$, where $\tilde{f} \in \mathcal{S}$ and $u \in \mathbf{O}_u$, e.g., $\exp(x_1)$. This set of characterizations allows representations of $f \in \mathcal{S}$ by a symbolic tree structure $T(f)$, where internal (nonterminal) nodes either represent unary (with 1 child node) or binary (with 2 children nodes) operators, while leaves (terminal nodes) correspond to primary features; see Figure 1. Such tree representations of symbolic expressions are not necessarily unique, e.g., $x_1(x_2 + x_3) = x_1x_2 + x_1x_3$. With this setup, now in the sequel we formally present the **VaSST** modeling framework.

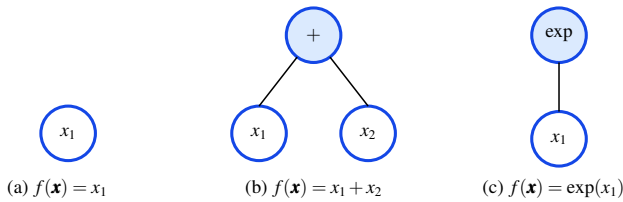


Figure 1: Symbolic tree representation.

3 THE VaSST MODEL

We outline the **VaSST** model which comprises two major components: (i) a symbolic ensemble consisting of an affine (linear) combination of K symbolic trees and (ii) a hierarchical prior specification over the model regression coefficients, the model noise variance, and the symbolic tree structures.

3.1 THE SYMBOLIC ENSEMBLE COMPONENT

Let $\mathbb{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be the collection of observed data units. **VaSST** models and structurally learns the hidden symbolic relationship between the responses $y_i \in \mathbb{R}$ and primary feature vector $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,p})^\top \in \mathbb{R}^p$ using K symbolic expressions $\{f_j\}_{j=1}^K \subset \mathcal{S}$ evaluated at $\mathbf{x}_i$, as:

$$y_i = \beta_0 + \sum_{j=1}^K \beta_j f_j(\mathbf{x}_i) + \varepsilon_i, \quad i = 1, \dots, n, \quad (1)$$

where $\boldsymbol{\beta} = (\beta_0, \dots, \beta_K)^\top \in \mathbb{R}^{K+1}$ is the model regression coefficient vector and $\varepsilon_i \in \mathbb{R}$ is the model noise with $\varepsilon_i \sim \mathcal{N}_1(0, \sigma^2)$ independently. From (1), we obtain the vector representation of the symbolic ensemble component:

$$\mathbf{y} = \mathbf{T}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\mathbf{0}_n, \sigma^2 \mathbf{I}_n), \quad (2)$$

where $\mathbf{T} = (f_j(\mathbf{x}_i))_{1 \leq i \leq n, 0 \leq j \leq K} \in \mathbb{R}^{n \times \overline{K+1}}$ is the expression design matrix, $f_0(\mathbf{x}_i) = 1$ for all $i = 1, \dots, n$, $\mathbf{y} = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$ is the response vector, and $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_n)^\top \in \mathbb{R}^n$ is the model noise vector.

The model regression parameters $(\boldsymbol{\beta}, \sigma^2)$ in (2) jointly are endowed upon with the conjugate Normal Inverse-Gamma (NIG) prior viz., $\boldsymbol{\beta} \mid \sigma^2 \sim \mathcal{N}_{K+1}(\boldsymbol{\mu}_0, \sigma^2 \boldsymbol{\Sigma}_0)$ and $\sigma^2 \sim \text{IG}(a_0, b_0)$, where $\boldsymbol{\mu}_0 \in \mathbb{R}^{K+1}$, $\boldsymbol{\Sigma}_0$ (positive definite matrix of order $K+1$), and $a_0, b_0 > 0$ are the hyperparameters of the Normal and Inverse-Gamma prior distributions, respectively. We complete the **VaSST** model specification by describing the symbolic tree prior in Section 3.2.

3.2 THE VaSST SYMBOLIC TREE PRIOR

The **VaSST** symbolic tree prior for each $T(f_j)$ is constructed in two stages: (a) a probabilistic specification over a maximal binary tree skeleton and (b) a deterministic pruning operation that maps the skeleton tree to a valid symbolic tree, as depicted in Figure 2.

Full binary tree skeleton. To decouple structural decisions from operator and feature assignments, we embed each symbolic tree $T(f_j)$ into a full binary tree skeleton, denoted by S_j , of fixed maximum depth $D \in \mathbb{N}$. Let $Z_D = \{0, 1, \dots, N-1\}$ index the skeleton tree nodes in *heap order*, where 0 is the root and the left and right children of a skeleton node $\zeta \in Z_D$ are given by $L(\zeta) = 2\zeta + 1$ and $R(\zeta) = 2\zeta + 2$, respectively. The total number of nodes is $N = 2^{D+1} - 1$. Also, let $d_\zeta \in \{0, \dots, D\}$ denote the depth of node ζ . For each skeleton node $\zeta \in Z_D$, we introduce the following, (a) expansion indicator: $e_{j\zeta} \in \{0, 1\}$, where $e_{j\zeta} = 1$ or $e_{j\zeta} = 0$ indicates that ζ is an internal node or a leaf of $T(f_j)$, respectively; (b) operator assignment: $o_{j\zeta} \in \mathbf{O}$, specifies the operator assigned to ζ if $e_{j\zeta} = 1$; and (c) feature assignment: $h_{j\zeta}$, specifies the primary feature assigned to ζ if $e_{j\zeta} = 0$. Thus, the skeleton tree S_j can be encoded by the structural variables $\{e_{j\zeta}, o_{j\zeta}, h_{j\zeta} : \zeta \in Z_D\}$.

Valid symbolic tree via deterministic pruning. The skeleton S_j is an ambient representation. We obtain the corresponding valid symbolic tree $T(f_j)$ by applying a deterministic pruning operator $\mathfrak{p}$ which proceeds as follows, (a) terminal pruning: if $e_{j\zeta} = 0$, then all descendants of ζ in the skeleton are removed, making it a leaf and (b) unary operator pruning: if $e_{j\zeta} = 1$ and $o_{j\zeta} \in \mathbf{O}_u$, then the right subtree rooted at $R(\zeta)$ is removed. Thus, the resulting symbolic tree is $T(f_j) = \mathfrak{p}(S_j)$ and the corresponding pruned node set is $Z(f_j)$.

Prior over full binary tree skeleton. The prior specifications over $\{e_{j\zeta}, o_{j\zeta}, h_{j\zeta} : \zeta \in Z_D\}$ and hence S_j is:

$$\begin{aligned} \Pi(e_{j\zeta}) &\equiv \text{Ber}(e_{j\zeta}; p_\zeta = \alpha(1 + d_\zeta)^{-\delta}), \\ \Pi(o_{j\zeta} \mid \mathbf{w}_{\text{op}}) &\equiv \text{Cat}(o_{j\zeta}; \mathbf{w}_{\text{op}}), \quad \Pi(h_{j\zeta} \mid \mathbf{w}_{\text{ft}}) \equiv \text{Cat}(h_{j\zeta}; \mathbf{w}_{\text{ft}}), \\ \Pi(S_j \mid \mathbf{w}_{\text{op}}, \mathbf{w}_{\text{ft}}) &= \prod_{\zeta \in Z_D} \Pi(e_{j\zeta}) \Pi(o_{j\zeta} \mid \mathbf{w}_{\text{op}}) \Pi(h_{j\zeta} \mid \mathbf{w}_{\text{ft}}), \end{aligned} \quad (3)$$

for $j = 1, \dots, K$. In (3), $\alpha, \delta > 0$ and $\mathbf{w}_{\text{op}} \in \Delta^{|\mathbf{O}|}, \mathbf{w}_{\text{ft}} \in \Delta^p$ are the operator and feature weight vectors, where $\Delta^m = \{\mathbf{z} \in [0, \infty)^m \mid \mathbf{1}_m^\top \mathbf{z} = 1\}$ is the m -dimensional simplex. Also, $\text{Ber}(\cdot)$ and $\text{Cat}(\cdot)$ denote Bernoulli and Categorical distributions, respectively. Note that, the prior over S_j in (3) induces a prior over $T(f_j)$, as:

$$\begin{aligned} \Pi(T(f_j) \mid \mathbf{w}_{\text{op}}, \mathbf{w}_{\text{ft}}) \\ = \prod_{\zeta \in Z(f_j)} \Pi(e_{j\zeta}) \Pi(o_{j\zeta} \mid \mathbf{w}_{\text{op}}) \Pi(h_{j\zeta} \mid \mathbf{w}_{\text{ft}}). \end{aligned} \quad (4)$$

Further, the priors over $\mathbf{w}_{\text{op}}$ and $\mathbf{w}_{\text{ft}}$ are:

$$\Pi(\mathbf{w}_{\text{op}}) \equiv \text{Dir}(\mathbf{w}_{\text{op}}; \boldsymbol{\eta}_{\text{op}}), \quad \Pi(\mathbf{w}_{\text{ft}}) \equiv \text{Dir}(\mathbf{w}_{\text{ft}}; \boldsymbol{\eta}_{\text{ft}}), \quad (5)$$

where $\boldsymbol{\eta}_{\text{op}} \in (0, \infty)^{|\mathbf{O}|}$ and $\boldsymbol{\eta}_{\text{ft}} \in (0, \infty)^p$ are concentration hyperparameters of Dirichlet distributions denoted by $\text{Dir}(\cdot)$. Therefore, (3) and (5) define the complete prior specification over $\Theta = (\{S_j\}_{j=1}^K, \mathbf{w}_{\text{op}}, \mathbf{w}_{\text{ft}})$.

We conclude with Remark 1 emphasizing the role of the depth-dependent split probability $p_\zeta = \alpha(1 + d_\zeta)^{-\delta}$ as a mechanism for controlling symbolic expression complexity.

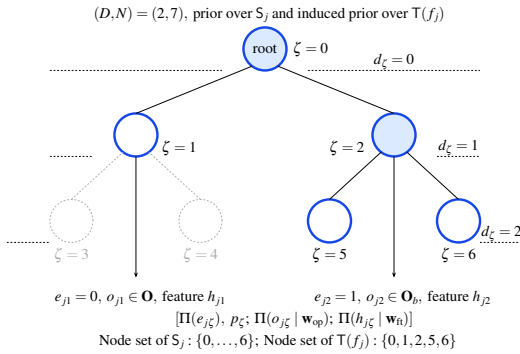


Figure 2: A live representation of the prior over S_j and the induced prior over $T(f_j)$. In S_j , the children nodes $\zeta = 3, 4$ of $\zeta = 1$ are pruned using $\mathbf{p}$ as $e_{j1} = 0$ to obtain the valid symbolic tree $T(f_j)$.

Remark 1 (Depth-dependent split probability). *Guided by the Occam’s razor principle [Jefferys and Berger, 1992], we aim to learn interpretable and parsimonious symbolic expressions which adequately captures the underlying scientific mechanism. This is achieved by the depth-dependent split probability p_ζ in (3) which imparts a regularizing effect on the individual tree depths [Chipman et al., 1998].*

4 VARIATIONAL INFERENCE FOR VaSST

Combining the data likelihood $p(\mathbf{y} \mid \mathbf{T}, \boldsymbol{\beta}, \sigma^2)$, the joint prior over the model regression parameters in Section 3.1, and the

VaSST symbolic tree prior in Section 3.2, the joint posterior distribution induced over all unknowns $(\Theta, \boldsymbol{\beta}, \sigma^2)$ is:

$$\begin{aligned} \Pi(\Theta, \boldsymbol{\beta}, \sigma^2 \mid \mathbb{D}_n) &\propto p(\mathbf{y} \mid \mathbf{T}, \boldsymbol{\beta}, \sigma^2) \Pi(\boldsymbol{\beta}, \sigma^2) \Pi(\mathbf{w}_{\text{op}}) \\ &\Pi(\mathbf{w}_{\text{ft}}) \prod_{j=1}^K \prod_{\zeta \in Z_D} \Pi(e_{j\zeta}) \Pi(o_{j\zeta} \mid \mathbf{w}_{\text{op}}) \Pi(h_{j\zeta} \mid \mathbf{w}_{\text{ft}}). \end{aligned} \quad (6)$$

Marginalization over model regression parameters. The conjugate NIG prior over $(\boldsymbol{\beta}, \sigma^2)$ enables marginalization in (6) and results into $\Pi(\Theta \mid \mathbb{D}_n) \propto p(\mathbf{y} \mid \mathbf{T}) \Pi(\Theta)$, where $p(\mathbf{y} \mid \mathbf{T})$ up to constants is:

$$\log p(\mathbf{y} \mid \mathbf{T}) = \frac{1}{2} \log |\boldsymbol{\Sigma}_n| - a_n \log b_n + \log \Gamma(a_n), \quad (7)$$

where:

$$\begin{aligned} \boldsymbol{\Sigma}_n^{-1} &= \boldsymbol{\Sigma}_0^{-1} + \mathbf{T}^\top \mathbf{T}, \quad \boldsymbol{\mu}_n = \boldsymbol{\Sigma}_n (\boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0 + \mathbf{T}^\top \mathbf{y}), \\ a_n &= a_0 + \frac{n}{2}, \\ b_n &= b_0 + \frac{1}{2} (\mathbf{y}^\top \mathbf{y} + \boldsymbol{\mu}_0^\top \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0 - \boldsymbol{\mu}_n^\top \boldsymbol{\Sigma}_n^{-1} \boldsymbol{\mu}_n), \end{aligned} \quad (8)$$

are the hyperparameters of the posterior NIG distribution. Also, $\Gamma(\cdot)$ and $|\boldsymbol{\Sigma}_n|$ are the Gamma function and the determinant of the matrix $\boldsymbol{\Sigma}_n$, respectively. See Appendix C.1 for complete derivations of (7) and (8). Therefore, it remains to draw inference from $\Pi(\Theta \mid \mathbb{D}_n)$.

Variational family. To conduct scalable and efficient inference, we adopt a variational inference routine for approximating the posterior $\Pi(\Theta \mid \mathbb{D}_n)$ with a tractable mean-field approximation [Jordan et al., 1999, Blei et al., 2017] $q \in \mathbb{Q}$ detailed in (9) below:

$$\begin{aligned} q(\Theta) &= q(\mathbf{w}_{\text{op}}) q(\mathbf{w}_{\text{ft}}) \prod_{j=1}^K \prod_{\zeta \in Z_D} q(e_{j\zeta}) q(o_{j\zeta}) q(h_{j\zeta}), \\ q(e_{j\zeta}) &\equiv \text{Ber}(e_{j\zeta}; \tilde{p}_{j\zeta}), \\ q(o_{j\zeta}) &\equiv \text{Cat}(o_{j\zeta}; \tilde{\pi}_{j\zeta}^{\text{op}}), \quad q(h_{j\zeta}) \equiv \text{Cat}(h_{j\zeta}; \tilde{\pi}_{j\zeta}^{\text{ft}}), \\ q(\mathbf{w}_{\text{op}}) &\equiv \text{Dir}(\mathbf{w}_{\text{op}}; \tilde{\boldsymbol{\eta}}_{\text{op}}), \quad q(\mathbf{w}_{\text{ft}}) \equiv \text{Dir}(\mathbf{w}_{\text{ft}}; \tilde{\boldsymbol{\eta}}_{\text{ft}}), \end{aligned} \quad (9)$$

for $j = 1, \dots, K$ and $\zeta \in Z_D$, where $\tilde{p}_{j\zeta} = \sigma(\ell_{j\zeta})$, $\tilde{\pi}_{j\zeta}^{\text{op}} = \text{smax}(\mathbf{a}_{j\zeta}^{\text{op}})$, and $\tilde{\pi}_{j\zeta}^{\text{ft}} = \text{smax}(\mathbf{a}_{j\zeta}^{\text{ft}})$; with $\sigma(\cdot)$ and $\text{smax}(\cdot)$ denoting the sigmoid and softmax functions, respectively, and $\ell_{j\zeta} \in \mathbb{R}$, $\mathbf{a}_{j\zeta}^{\text{op}} \in \mathbb{R}^{|\mathbf{O}|}$, and $\mathbf{a}_{j\zeta}^{\text{ft}} \in \mathbb{R}^p$. Hence, q is parameterized by:

$$\phi = (\tilde{\boldsymbol{\eta}}_{\text{op}}, \tilde{\boldsymbol{\eta}}_{\text{ft}}, \{\ell_{j\zeta}, \mathbf{a}_{j\zeta}^{\text{op}}, \mathbf{a}_{j\zeta}^{\text{ft}}\}_{j=1, \dots, K; \zeta \in Z_D}). \quad (10)$$

Evidence lower bound. In theory, one can obtain the optimal value of the variational parameters ϕ in (10) by maximizing the evidence lower bound (ELBO):

$$\mathcal{E}(\phi) = \mathbb{E}_q[\log p(\mathbf{y} \mid \mathbf{T})] - \text{KL}[q(\Theta) \parallel \Pi(\Theta)], \quad (11)$$

where $\text{KL}[q(\Theta) \parallel \Pi(\Theta)] = \mathbb{E}_q[\log q(\Theta) - \log \Pi(\Theta)]$ is the Kullback-Leibler (KL) divergence and $\log p(\mathbf{y} \mid \mathbf{T})$ is as in (7). By (9), $\text{KL}[q(\Theta) \parallel \Pi(\Theta)]$ splits as:

$$\begin{aligned} & \text{KL}[q(\Theta) \parallel \Pi(\Theta)] \\ &= \text{KL}[q(\mathbf{w}_{\text{op}}) \parallel \Pi(\mathbf{w}_{\text{op}})] + \text{KL}[q(\mathbf{w}_{\text{ft}}) \parallel \Pi(\mathbf{w}_{\text{ft}})] \\ &+ \sum_{j=1}^K \sum_{\zeta \in Z_D} \left[\text{KL}[q(e_{j\zeta}) \parallel \Pi(e_{j\zeta})] \right. \\ &\quad + \mathbb{E}_{q(\mathbf{w}_{\text{ft}})} \text{KL}[q(h_{j\zeta}) \parallel \Pi(h_{j\zeta} \mid \mathbf{w}_{\text{ft}})] \\ &\quad \left. + \mathbb{E}_{q(\mathbf{w}_{\text{op}})} \text{KL}[q(o_{j\zeta}) \parallel \Pi(o_{j\zeta} \mid \mathbf{w}_{\text{op}})] \right]. \end{aligned} \quad (12)$$

The individual terms in (12) are given by:

$$\begin{aligned} & \text{KL}[q(e_{j\zeta}) \parallel \Pi(e_{j\zeta})] \\ &= \tilde{p}_{j\zeta} \log \frac{\tilde{p}_{j\zeta}}{p_{j\zeta}} + (1 - \tilde{p}_{j\zeta}) \log \frac{1 - \tilde{p}_{j\zeta}}{1 - p_{j\zeta}}, \\ & \text{KL}[q(\mathbf{w}_{\text{op}}) \parallel \Pi(\mathbf{w}_{\text{op}})] = \log \frac{\mathcal{B}(\boldsymbol{\eta}_{\text{op}})}{\mathcal{B}(\tilde{\boldsymbol{\eta}}_{\text{op}})} \\ &+ \sum_{k=1}^{|\mathbf{O}|} (\tilde{\eta}_{\text{op},k} - \eta_{\text{op},k}) (\Psi(\tilde{\eta}_{\text{op},k}) - \Psi(\mathbf{1}_{|\mathbf{O}|}^\top \tilde{\boldsymbol{\eta}}_{\text{op}})), \\ & \mathbb{E}_{q(\mathbf{w}_{\text{op}})} \text{KL}[q(o_{j\zeta}) \parallel \Pi(o_{j\zeta} \mid \mathbf{w}_{\text{op}})] \\ &= \sum_{k=1}^{|\mathbf{O}|} \tilde{\pi}_{j\zeta,k}^{\text{op}} \left[\log(\tilde{\pi}_{j\zeta,k}^{\text{op}}) - \Psi(\tilde{\eta}_{\text{op},k}) + \Psi(\mathbf{1}_{|\mathbf{O}|}^\top \tilde{\boldsymbol{\eta}}_{\text{op}}) \right], \end{aligned}$$

where $\mathcal{B}(\cdot)$ and $\Psi(\cdot)$ are the multivariate Beta and Digamma functions, respectively. $\text{KL}[q(\mathbf{w}_{\text{ft}}) \parallel \Pi(\mathbf{w}_{\text{ft}})]$ and $\mathbb{E}_{q(\mathbf{w}_{\text{ft}})} \text{KL}[q(h_{j\zeta}) \parallel \Pi(h_{j\zeta} \mid \mathbf{w}_{\text{ft}})]$ are given analogously; see Appendix C.2 for details.

The term $\mathbb{E}_q[\log p(\mathbf{y} \mid \mathbf{T})]$ in (11) does not admit an analytical form. A direct stochastic optimization is computationally prohibitive due to the combinatorial nature of the skeleton tree structural variables $\{e_{j\zeta}, o_{j\zeta}, h_{j\zeta} : \zeta \in Z_D\}_{j=1}^K$. The induced discrete model space grows exponentially with both tree depth (D) and ensemble size (K) having $\mathcal{O}((2p|\mathbf{O}|)^{N \cdot K})$ possible configurations, where $\mathcal{O}$ is the big-O notation. Combinatorial search over this astronomically large space is therefore infeasible. Moreover, gradient-based optimization over discrete variables would require high-variance score-function estimators [Williams, 1992, Ranganath et al., 2014] or specialized structured search procedures [Koza, 1994, Schmidt and Lipson, 2009], both of which scale poorly and become impractical even for moderate values of D and K . To overcome this computational bottleneck, VaSST introduces a differentiable relaxation of symbolic trees, regarded as *soft symbolic trees*, transforming discrete structural and labeling assignments into probability distributions [Maddison et al., 2017, Liu et al., 2019].

Soft symbolic trees via continuous relaxations. To construct soft symbolic trees, i.e., $S_j \rightarrow S_j^{\text{soft}}$ for $j = 1, \dots, K$, we implement the Binary-Concrete [Maddison et al., 2017]

and Gumbel-Softmax [Jang et al., 2017] continuous relaxations of the structural variables $\{e_{j\zeta}, o_{j\zeta}, h_{j\zeta} : \zeta \in Z_D\}$ of the skeleton tree S_j for $j = 1, \dots, K$, as:

$$\begin{aligned} & \text{Binary-Concrete} : \tilde{e}_{j\zeta} = \sigma \left(\frac{\ell_{j\zeta} + \log u_{j\zeta}^{\text{ex}} - \log(1 - u_{j\zeta}^{\text{ex}})}{\tau_{\text{ex}}} \right), \\ & \text{Gumbel-Softmax} : \tilde{\mathbf{o}}_{j\zeta} = \text{smax} \left(\frac{\mathbf{a}_{j\zeta}^{\text{op}} + \mathbf{g}_{j\zeta}^{\text{op}}}{\tau_{\text{op}}} \right), \\ & \tilde{\mathbf{h}}_{j\zeta} = \text{smax} \left(\frac{\mathbf{a}_{j\zeta}^{\text{ft}} + \mathbf{g}_{j\zeta}^{\text{ft}}}{\tau_{\text{ft}}} \right), \end{aligned} \quad (13)$$

where $u_{j\zeta}^{\text{ex}} \sim \text{Unif}(0, 1)$, $\mathbf{g}_{j\zeta}^{\text{op}} = \{-\log(-\log u_{j\zeta,r}^{\text{op}})\}_{r=1}^{|\mathbf{O}|}$, $\mathbf{g}_{j\zeta}^{\text{ft}} = \{-\log(-\log u_{j\zeta,k}^{\text{ft}})\}_{k=1}^p$, $u_{j\zeta,r}^{\text{op}} \sim \text{Unif}(0, 1)$, and $u_{j\zeta,k}^{\text{ft}} \sim \text{Unif}(0, 1)$. Also, all uniform random variables are mutually independent. Observe that, $\tilde{e}_{j\zeta} \in \mathbb{R}$, $\tilde{\mathbf{o}}_{j\zeta} \in \mathbb{R}^{|\mathbf{O}|}$, and $\tilde{\mathbf{h}}_{j\zeta} \in \mathbb{R}^p$ can be interpreted as the soft one-hot encoding of the corresponding original discrete structural variables. In (13), $\boldsymbol{\tau} = (\tau_{\text{ex}}, \tau_{\text{op}}, \tau_{\text{ft}})$ are temperature parameters controlling the sharpness of the Binary-Concrete and Gumbel-Softmax relaxations. Particularly, higher temperatures yield smooth mixtures of structural configurations, whereas for smaller temperatures, the relaxed variables concentrate toward discrete structures. A careful annealing of these parameters allows for a balance between exploration and structural learning of symbolic expressions. Thus, the resulting soft symbolic tree S_j^{soft} is encoded by $\{\tilde{e}_{j\zeta}, \tilde{\mathbf{o}}_{j\zeta}, \tilde{\mathbf{h}}_{j\zeta} : \zeta \in Z_D\}$.

Evaluation of soft symbolic trees. Given the soft symbolic trees S_j^{soft} for $j = 1, \dots, K$, as discussed above, we now outline an algorithm for evaluation of S_j^{soft} at a given feature vector instance $\mathbf{x}_i \in \mathbb{R}^p$. This evaluation is done recursively over the nodes of S_j^{soft} . For a given node $\zeta \in Z_D$, the soft terminal (leaf) contribution is computed as a convex combination of the input features, $\tilde{\mathbf{h}}_{j\zeta}^\top \mathbf{x}_i$, which corresponds to a soft feature evaluation. If the node acts as nonterminal (internal), its output is obtained by a weighted mixture of unary and binary operations over its children. Specifically, the unary contribution aggregates $\tilde{o}_{j\zeta,u} \cdot u(S_j^{\text{soft}}(\mathbf{x}_i; L(\zeta)))$, over $u \in \mathbf{O}_u$, while the binary contribution aggregates $\tilde{o}_{j\zeta,b} \cdot b(S_j^{\text{soft}}(\mathbf{x}_i; L(\zeta)), S_j^{\text{soft}}(\mathbf{x}_i; R(\zeta)))$, over $b \in \mathbf{O}_b$. The complete node evaluation is then obtained via *soft gating*:

$$\begin{aligned} & S_j^{\text{soft}}(\mathbf{x}_i; \zeta) = (1 - \tilde{e}_{j\zeta}) \tilde{\mathbf{h}}_{j\zeta}^\top \mathbf{x}_i \\ &+ \tilde{e}_{j\zeta} \sum_{u \in \mathbf{O}_u} \tilde{o}_{j\zeta,u} \cdot u(S_j^{\text{soft}}(\mathbf{x}_i; L(\zeta))) \\ &+ \tilde{e}_{j\zeta} \sum_{b \in \mathbf{O}_b} \tilde{o}_{j\zeta,b} \cdot b(S_j^{\text{soft}}(\mathbf{x}_i; L(\zeta)), S_j^{\text{soft}}(\mathbf{x}_i; R(\zeta))), \end{aligned} \quad (14)$$

thus smoothly interpolating over all possible combinations of operators and features; see **SOFT EVAL AT NODE Algorithm 1** in Appendix D. Consequently, evaluating a soft symbolic tree corresponds to computing $S_j^{\text{soft}}(\mathbf{x}_i; \zeta)$ from (14) at the root node $\zeta = 0$. Repeating this procedure for the collection of K soft symbolic trees and all observations yields the

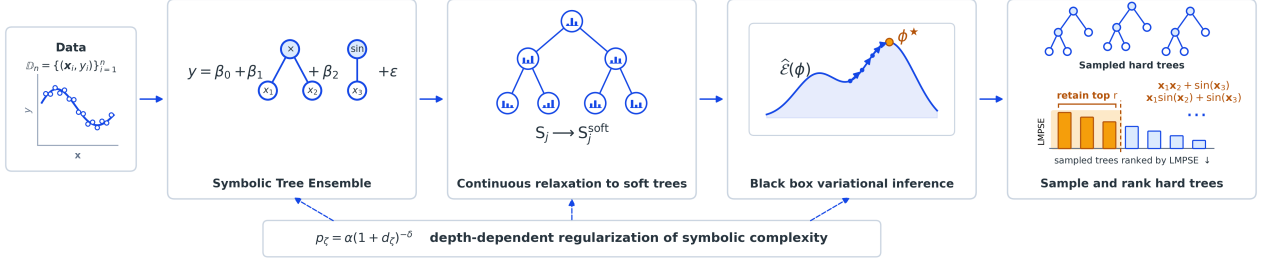


Figure 3: The complete VaSST pipeline.

soft design matrix $\mathbf{T}_{\text{soft}} \in \mathbb{R}^{n \times \overline{K}+1}$, which includes the intercept column; see [SOFTEVAL Algorithm 2](#) in [Appendix D](#).

Stochastic approximation of $\mathcal{E}(\phi)$. The ELBO objective $\mathcal{E}(\phi)$ in (11) involves the analytically intractable term $\mathbb{E}_q[\log p(\mathbf{y} | \mathbf{T})]$, since $\mathbf{T}$ exhibits nonlinear dependence on the soft symbolic trees. We therefore approximate this term using Monte Carlo (MC) sampling from the distribution induced over $\{\tilde{e}_{j\zeta}, \tilde{o}_{j\zeta}, \tilde{h}_{j\zeta} : \zeta \in \mathbb{Z}_D\}_{j=1}^K$ in (13), denoted as $\mathcal{P}_{\phi, \tau}^{\text{soft}}$. In particular, consider S MC samples from $\mathcal{P}_{\phi, \tau}^{\text{soft}}$ and invoking the [SOFTEVAL Algorithm 2](#) in [Appendix D](#), we compute the soft design matrices $\mathbf{T}_{\text{soft}}^{(s)}$, and hence $\log p(\mathbf{y} | \mathbf{T}_{\text{soft}}^{(s)})$ using (7), for $s = 1, \dots, S$. Finally, the MC approximation of $\mathcal{E}(\phi)$ is obtained as:

$$\hat{\mathcal{E}}(\phi) = \frac{1}{S} \sum_{s=1}^S \log p(\mathbf{y} | \mathbf{T}_{\text{soft}}^{(s)}) - \text{KL}[q(\Theta) \| \Pi(\Theta)], \quad (15)$$

where $\text{KL}[q_\phi(\Theta) \| \Pi(\Theta)]$ is given by (12). [APPROXELBO Algorithm 3](#) in [Appendix E](#) evaluates the stochastic approximation of $\mathcal{E}(\phi)$ in (11) by $\hat{\mathcal{E}}(\phi)$ in (15) and [Appendix H](#) discusses the corresponding approximation gap. Importantly, $\hat{\mathcal{E}}(\phi)$ is differentiable with respect to the variational parameters in ϕ , enabling efficient gradient-based optimization.

Black-box optimization. To obtain the optimal variational parameter vector ϕ^* , we maximize $\hat{\mathcal{E}}(\phi)$ in (15) using gradient-based black-box variational inference [[Ranganath et al., 2014](#), [Kucukelbir et al., 2017](#)], leveraging automatic differentiation [[Rall, 1981](#)] to compute $\nabla_\phi \hat{\mathcal{E}}(\phi)$. The resulting objective is optimized using the AdamW optimizer [[Loshchilov and Hutter, 2019](#)]. To progressively sharpen the continuous relaxations in (13), we employ an annealing schedule which linearly decreases the temperature across iterations, gradually transitioning from smooth structural mixtures to near-discrete symbolic trees; refer to [VaSST Algorithm 4](#) in [Appendix F](#).

Interpretable hard symbolic ensembles. While soft symbolic trees S_j^{soft} serve as tractable computational intermediaries, to recover interpretable symbolic expressions we

sample *hard symbolic trees* using ϕ^* . Concretely, we draw $H \in \mathbb{N}$ samples, where for each draw $s = 1, \dots, H$ and each tree $j = 1, \dots, K$, we sample: $\hat{e}_{j\zeta}^{(s)} \sim \text{Ber}(\sigma(\ell_{j\zeta}^*))$, $\hat{o}_{j\zeta}^{(s)} \sim \text{Cat}(\text{smax}((\mathbf{a}_{j\zeta}^{\text{op}})^*))$, and $\hat{h}_{j\zeta}^{(s)} \sim \text{Cat}(\text{smax}((\mathbf{a}_{j\zeta}^{\text{ft}})^*))$, respectively for all $\zeta \in \mathbb{Z}_D$. This sampling defines a full depth- D hard binary tree skeleton $\hat{S}_j^{(s)}$, which is then deterministically pruned to obtain $\mathbf{T}(f_j^{(s)}) = \mathbf{p}(\hat{S}_j^{(s)})$ and hence expressions $f_j^{(s)} \in \mathcal{S}$. The expression design matrix $\mathbf{T}^{(s)}$ is computed by evaluating $f_j^{(s)}$ at instances $\{\mathbf{x}_i\}_{i=1}^n$. Conditioned on $\mathbf{T}^{(s)}$ and $\mathbf{y}$, the posterior mean estimates $\boldsymbol{\beta}_{\text{PM}}^{(s)} = \mathbb{E}[\boldsymbol{\beta} | \sigma^2, \mathbf{T}^{(s)}, \mathbf{y}]$ and $(\sigma_{\text{PM}}^2)^{(s)} = \mathbb{E}[\sigma^2 | \mathbf{T}^{(s)}, \mathbf{y}]$ are computed using (8). See [SAMPLEHARD Algorithm 5](#) in [Appendix G](#) for details of obtaining hard symbolic expressions.

Posterior-aware ranking of hard symbolic ensembles.

We complete the VaSST pipeline by ranking the above H sampled hard symbolic ensembles using a posterior-aware selection criterion. For any ensemble $\{\mathbf{T}(f_j)\}_{j=1}^K$, the log-marginal posterior over symbolic ensembles (LMPSE) is:

$$\begin{aligned} \text{LMPSE}(\{\mathbf{T}(f_j)\}_{j=1}^K) &= \log \Pi(\{\mathbf{T}(f_j)\}_{j=1}^K | \mathbb{D}_n) \\ &= \log p(\mathbf{y} | \mathbf{T}) + \log \Pi(\{\mathbf{T}(f_j)\}_{j=1}^K), \end{aligned} \quad (16)$$

where $p(\mathbf{y} | \mathbf{T})$ is as in (7) and $\Pi(\{\mathbf{T}(f_j)\}_{j=1}^K)$ is:

$$\Pi(\{\mathbf{T}(f_j)\}_{j=1}^K) = \mathbb{E}_{\mathbf{w}_{\text{op}}, \mathbf{w}_{\text{ft}}} \left[\prod_{j=1}^K \Pi(\mathbf{T}(f_j) | \mathbf{w}_{\text{op}}, \mathbf{w}_{\text{ft}}) \right],$$

with $\Pi(\mathbf{T}(f_j) | \mathbf{w}_{\text{op}}, \mathbf{w}_{\text{ft}})$ given in (4). [Appendix C.3](#) gives the analytical formula and derivation of LMPSE. Note that, LMPSE in (16) quantifies an interpretability-accuracy trade-off by balancing data fit, through the marginal likelihood, with structural parsimony, through the marginal prior over the symbolic ensemble. Ranking hard symbolic ensembles by LMPSE and consequently retaining the top $r \in \mathbb{N}$ ensembles with the largest LMPSE values provides an Occam’s window-based posterior summary [[Madigan and Raftery, 1994](#)] of competing symbolic structures which account for posterior uncertainty in symbolic model selection. A complete overview of the VaSST pipeline is in [Figure 3](#).

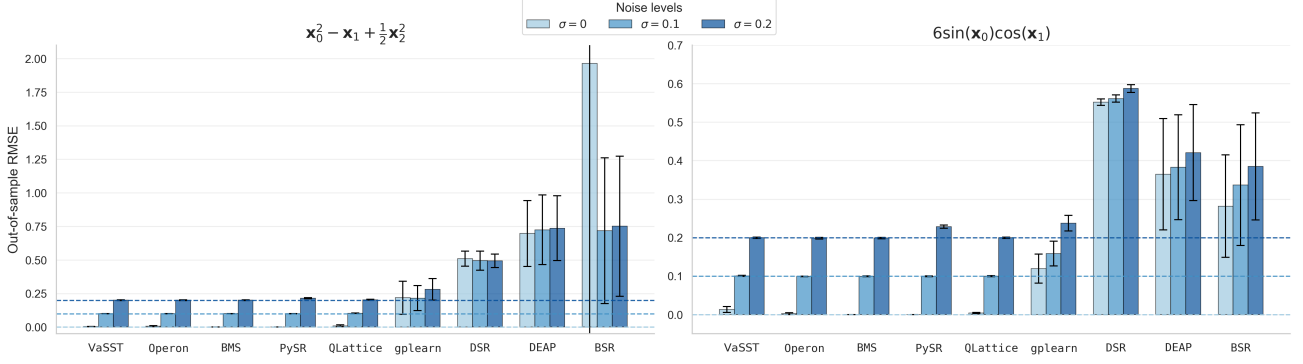


Figure 4: Out-of-sample RMSEs computed on a 10% held-out test set of **VaSST** and competitors for learning (17) and (18) across 10 repetitions under varying noise levels.

5 VaSST IN ACTION

We evaluate **VaSST** through simulation studies and a suite of canonical Feynman equations, assessing: (a) structural learning of symbolic expressions, (b) predictive accuracy, (c) stability under experimental noise, and (d) computational scalability relative to existing Bayesian SR methods. **VaSST** is compared with state-of-the-art SR modules from SRBench [La Cava, 2026], including Distributed Evolutionary Algorithms in Python (DEAP) [Fortin et al., 2012], gplearn [Stephens, 2016], operon [Burlacu et al., 2020], PySR [Cranmer, 2023], Deep Symbolic Regression (DSR) [Petersen et al., 2021], QLattice [Broløs et al., 2021], BMS [Guimerà et al., 2020], and BSR [Jin et al., 2020]. Across all experiments, we implement **VaSST** with $(K, D) = (3, 3)$ and use a 90/10 train-test split for measuring predictive accuracy by computing out-of-sample root mean square error (RMSE) on the held-out test set. For **VaSST**, after variational optimization, we sample $H = 2000$ hard symbolic ensembles from the learned soft representation and rank them using LMPSE in (16). Throughout, we use the top ranked symbolic model of **VaSST** for empirical comparisons, while the top $r = 10$ ranked models summarize posterior uncertainty over different symbolic structures. For budget equalization, we use common operator sets across methods whenever possible; see Appendix I for detailed configurations of all methods. Experiments were conducted in Python on a MacBook Air with an Apple M2 chip and 8GB RAM.

5.1 SIMULATION EXPERIMENTS

We consider the following symbolic expressions of varying levels of structural complexity:

$$\mathbf{y} = \mathbf{x}_0^2 - \mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2^2 + \boldsymbol{\varepsilon}, \quad (17)$$

$$\mathbf{y} = 6\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) + \boldsymbol{\varepsilon}, \quad (18)$$

where $\mathbf{x}_j = (x_{1,j}, \dots, x_{n,j})$ denotes the observed values of the j th feature, with all operations on these vectors interpreted

coordinate-wise. We simulate $x_{i,j} \sim \text{Unif}(2j, 2j+1)$, independently for $i = 1, \dots, n$ with $n = 2000$ and $j = 0, \dots, p-1$. Note that, $p = 3$ for (17) and $p = 2$ for (18). Two experimental regimes are studied: (a) a noiseless setting, $\boldsymbol{\varepsilon} = \mathbf{0}_n$ and (b) noisy settings, where $\boldsymbol{\varepsilon} \sim \mathcal{N}_n(\mathbf{0}_n, \sigma^2 \mathbf{I}_n)$ with $\sigma^2 \in \{0.1^2, 0.2^2\}$.

Figure 4 reports the out-of-sample RMSEs across 10 repetitions for both simulation settings, with detailed numerical summaries provided in Appendix J.2. For (17) and (18), **VaSST** achieves superior predictive accuracy with RMSEs that closely track the corresponding noise floors. Several competing methods, including operon, BMS, PySR, QLattice, and gplearn, also achieve competitive RMSEs. Conversely, DSR, DEAP, and BSR exhibit degradation in accuracy with substantially larger errors, where BSR is unstable particularly for (17), while DSR produces high RMSE values across both simulations.

	$\mathbf{x}_0^2 - \mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2^2$			$6\sin(\mathbf{x}_0)\cos(\mathbf{x}_1)$		
VaSST	10 / 10	10 / 10	10 / 10	10 / 10	10 / 10	10 / 10
BMS	10 / 10	10 / 10	10 / 10	10 / 10	10 / 10	10 / 10
BSR	0 / 10	0 / 10	0 / 10	1 / 10	1 / 10	1 / 10
DEAP	0 / 10	0 / 10	0 / 10	0 / 10	0 / 10	0 / 10
DSR	0 / 10	0 / 10	0 / 10	0 / 10	0 / 10	0 / 10
gplearn	0 / 10	0 / 10	0 / 10	0 / 10	0 / 10	0 / 10
operon	2 / 10	3 / 10	2 / 10	0 / 10	0 / 10	0 / 10
PySR	10 / 10	10 / 10	10 / 10	10 / 10	10 / 10	10 / 10
QLattice	4 / 10	3 / 10	1 / 10	0 / 10	0 / 10	0 / 10
	0	0.1	0.2	0	0.1	0.2
	Noise SD (σ)					

Figure 5: Expression recovery frequency of **VaSST** and competitors for learning (17) and (18).

We next examine whether predictive accuracy translates into structural recovery. Figure 5 summarizes the expression recovery frequencies over same 10 repetitions. **VaSST**, BMS, and PySR recover the correct structure in all repetitions and across all noise levels while learning (17) and (18), whereas BSR, DEAP, DSR, gplearn, operon, and QLattice either fail to recover the target expressions or do so inconsistently across repetitions and noise levels. The symbolic expressions learned by each method reported in Appendix J.1

further explain the distinction between accuracy and structural recovery. The LMPSE ranked expressions selected by **VaSST** retain the data-generating components of (17) and (18), while methods such as *operon*, *QLattice*, and *gplearn* often achieve low RMSEs through structurally indirect formulas. In particular, *operon* and *QLattice* occasionally recover the polynomial structure in (17), but via unwieldy and complex expressions with multiple high-order powers or nested trigonometric transformations; this recovery also deteriorates under noise and does not transfer to the trigonometric model in (18). Thus, Figures 4-5 and Tables J.1–J.6 in Appendix J.1 collectively show that **VaSST** balances strong predictive performance with stable symbolic recovery through structurally parsimonious expressions, which is driven by the depth-dependent split probability in (3) penalizing complex symbolic tree forms.

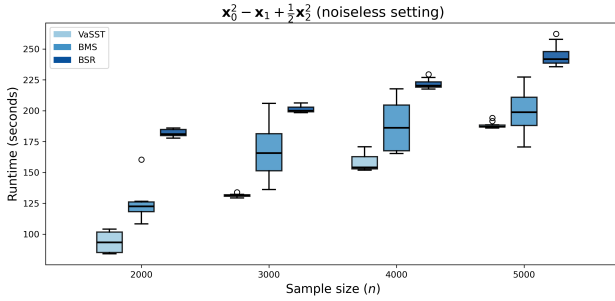


Figure 6: Computational scalability of **VaSST**, BMS, BSR.

Furthermore, in learning (17) under the noiseless setting, Figure 6 compares the single-run compute time of **VaSST** with the Bayesian baselines BMS and BSR (maintained at similar configurations to that of **VaSST**). For increasing sample sizes $n \in \{2000, 3000, 4000, 5000\}$ with 10 repetitions per n , **VaSST** invariably records the lowest runtime, demonstrating improved scalability. Finally, the top $r = 10$ hard symbolic ensembles in Tables J.9–J.14 of Appendix J.3, ranked by LMPSE, provide a posterior-aware summary of uncertainty over competing symbolic expressions. These ranked candidates show how **VaSST** concentrates posterior support on expressions with the correct symbolic structure while retaining structurally similar variants as low rank alternatives.

5.2 APPLICATION TO FEYNMAN EQUATIONS

To further assess symbolic discovery in scientific settings, we apply **VaSST** and the competing methods to benchmark SR problems from the *Feynman Symbolic Regression Database* (FSReD) [Udrescu and Tegmark, 2020], which contains over 100 equations from the Feynman Lectures on Physics [Feynman et al., 1963], each with 10^5 observations. We consider five representative laws with varying symbolic complexity viz., Coulomb’s law (CL), change in gravitational potential energy (CPE), Lorentz force on a

moving charge in an electromagnetic field (FCE), Fourier’s law of thermal conduction (FTC), and angular distribution of scattering with forward-backward asymmetry (ADA):

$$F = 0.08 \frac{q_1 q_2}{\epsilon r^2}, \quad (\text{CL})$$

$$\Delta U = G m_1 m_2 \left(\frac{1}{r_2} - \frac{1}{r_1} \right), \quad (\text{CPE})$$

$$F = q (E_f + v B \sin \theta), \quad (\text{FCE})$$

$$P = \frac{\kappa A (T_2 - T_1)}{d}, \quad (\text{FTC})$$

$$f = \beta^\dagger (1 + \alpha^\dagger \cos \theta), \quad (\text{ADA})$$

where the input features, responses, and constants are described in Appendix K. For each equation, we randomly subsample $n = 2000$ observations and study the same regimes as in Section 5.1: the original noiseless response and noisy regimes obtained by adding Gaussian noise with variance $\sigma^2 \in \{0.1^2, 0.2^2\}$ to the response variable, mimicking measurement error and/or experimental perturbations.

	CL			CPE			FCE			FTC			ADA		
VaSST	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			Y	Y	Y
BMS	Y	Y	Y	Y	Y	Y	Y	Y	Y				Y	Y	Y
BSR															
DEAP	Y														
DSR				Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
gplearn	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			Y	Y	Y
operon	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
PySR	Y			Y	Y	Y	Y	Y	Y	Y			Y	Y	Y
QLattice	Y	Y	Y	Y											
	0	0.1	0.2	0	0.1	0.2	0	0.1	0.2	0	0.1	0.2	0	0.1	0.2
	Noise SD (σ)														
	Captured														
	Not captured														

Figure 7: Expression recovery of **VaSST** and competitors for learning the Feynman equations.

Figure 7 summarizes symbolic recovery across the five Feynman equations and varying noise levels, with full expression-level results reported in Appendix L.1 and a representative example for (CPE) under $\sigma = 0.2$ shown in Table 1. **VaSST** recovers the target structure for (CL), (CPE), (FCE), and (ADA) across noise regimes. For (FTC), it learns the exact symbolic form in the noiseless setting and retains dominant thermal conduction components under the noisy regimes. Across these settings, the out-of-sample RMSEs of **VaSST** closely track the corresponding noise levels, indicating that its symbolic recovery is accompanied by accurate prediction. The genetic programming modules are competitive on several equations. PySR and gplearn recover exact or near-exact formulas with near-noise-floor RMSE values. operon also achieves accurate prediction in many cases, although its learned formulas are sometimes more algebraically elaborate, involving split terms, fitted constants, or indirect rearrangements of the target law. BMS is structurally accurate for (CL), (CPE), (FCE), and (ADA), but is less successful in recovering the full difference structure of (FTC). In contrast, QLattice, BSR, and DEAP return unwieldy expressions involving multiple high-order interactions, consistent with their behavior in Section 5.1.

Table 1: Expressions recovered by **VaSST** and competitors for learning **(CPE)** under $\sigma = 0.2$.

Method	Symbolic Expression Learned	RMSE
VaSST	$1.014765 \frac{Gm_1 m_2}{r_2} - 1.000981 \frac{Gm_1 m_2}{r_1} - 0.006813 \frac{r_2}{r_1} - 0.055216$	0.199507
operon	$-0.010000 - 0.332 \left[\frac{0.707 Gm_1^2/r_1 - 6.275451 Gm_1 m_2/r_1}{2.031 r_2/r_1} + 3.007296 \frac{Gm_1 m_2}{r_1} + \frac{0.670 Gm_1^2/r_1}{3.388 Gm_1^2/r_1 + 2.718552 m_2^2/m_1^2} \right]$	0.199207
PySR	$Gm_1 m_2 \left(\frac{1}{r_2} - \frac{1}{r_1} \right)$	0.199569
gplearn	$Gm_1 m_2 \left(\frac{1}{r_2} - \frac{1}{r_1} \right)$	0.199569
DEAP	$-\frac{Gm_1 m_2}{r_1} - \frac{Gm_1^2}{r_1} + \frac{m_2}{m_1} + \frac{2.8770}{-Gm_1^2/r_1 + m_2/m_1} - 4.5144$	0.589860
BSR	$8.54074 + 0.827664 \frac{m_2 r_2}{m_1 r_1} + 0.253507 \left(\frac{Gm_1 m_2 r_2}{r_1^2} - \frac{Gm_1^2}{r_1} \right) + 0.379881 \frac{m_2}{m_1^2}$	1.107579
BMS	$-\frac{Gm_1 m_2}{r_1} + \frac{a_0 Gm_1 m_2}{2r_2}$	0.199557
QLattice	$2.11181 \left(-0.270397 \frac{Gm_1^2}{r_1} - 0.00788965 \right) \left(-0.0392111 \frac{r_2}{r_1} + 1.45354 + 1.86691 \exp\left\{0.281573 \frac{r_2}{r_1}\right\} \right) \sqrt{\left(\frac{m_2}{m_1} - 0.0186931\right)^2} + 0.238388$	0.246633
DSR	$Gm_1 m_2 \left(\frac{1}{r_2} - \frac{1}{r_1} \right)$	0.199569

Note: a_0 is a constant learned by BMS. The RMSE values correspond to the out-of-sample RMSEs computed on a 10% held-out test set.

When these methods attain competitive RMSEs, the fit is typically achieved through complex surrogate structures rather than clean recovery of the underlying physical law. Overall, the Feynman results show that **VaSST** accurately learns the underlying symbolic structure of the benchmark equations while providing a posterior-aware summary of uncertainty over competing symbolic forms through the top $r = 10$ LMPSE ranked hard symbolic ensembles reported in [Tables L.17-L.31 of Appendix L.3](#). As in [Section 5.1](#), **VaSST** also exhibits substantial runtime gains over BMS and BSR, with detailed runtime comparisons provided in [Table L.16 of Appendix L.2](#).

We conclude with additional diagnostics that further characterize the robustness and computational behavior of **VaSST**. Specifically, [Appendix N](#) presents ablation studies for **VaSST** over K and D , providing practical guideline for choosing the symbolic ensemble size and tree depth, and showing that $(K, D) = (3, 3)$ balances symbolic expressivity, recovery accuracy, and computational cost in the applications considered above. Memory footprint and gradient variability analyses in [Appendix O](#) indicate that the variational optimization in **VaSST** remains computationally stable as the symbolic search space grows. Finally, [Appendix P](#) compares **VaSST** with recent machine learning-based SR architectures, including LLM-SR [\[Shojaee et al., 2025\]](#) for learning **(ADA)** under $\sigma = 0.2$. The results highlight a complementary distinction between prompt-driven and probabilistic approaches to SR. While prompt-driven methods can leverage the generative capacity of language models, the LLM-SR experiments show sensitivity to initialization, language model choice, and prompt specification. In particular, weaker prompts often yield incomplete or non-executable symbolic programs, while more structured prompts improve execution but may still produce overly complex or structurally invalid expressions. In contrast, **VaSST** frames symbolic discovery as posterior inference over expression structures, enabling efficient optimization and posterior-aware

uncertainty summaries across competing symbolic forms.

6 CONCLUSION

We presented **VaSST**, a fully probabilistic framework for SR that leverages variational inference with continuous relaxations of symbolic trees. It enables scalable gradient-based optimization while preserving interpretability equipped with posterior-aware uncertainty quantification. Empirically, **VaSST** demonstrates strong structural recovery, competitive predictive accuracy, stability under noise, and notable computational gains over existing Bayesian SR methods.

This work opens promising future avenues for fully probabilistic and scalable variational inference-based SR. First, extending **VaSST** to infer internal numerical constants would broaden its applicability in learning scientific laws with unknown physical coefficients. Although the current formulation can represent simple constants compositionally, such as $\sin(2x)$ through $\sin(x+x)$, it does not directly learn free constants inside nonlinear terms such as $\exp(-\frac{1}{2}x^2)$. Second, because algebraically equivalent expressions are naturally pervasive in SR, future work could incorporate equivalence-aware symbolic model comparison or posterior summaries that group semantically identical expression forms. Finally, more structured optimization strategies for the variational objective may further improve scalability relative to modern machine learning-based SR approaches while preserving the probabilistic interpretation of **VaSST**.

Author Contributions

SR and PD were responsible for the conceptualization of **VaSST**, the development of the associated methodology, the design, code, and implementation of the experiments, and the writing of the article. BKM provided supervision and guidance throughout the course of the research.

Acknowledgements

Research reported in this article was supported by National Institute of General Medical Sciences of the National Institutes of Health under award number R01GM163238. Experiments in [Appendix P](#) were conducted with the computing resources provided by Texas A&M Department of Statistics Arseven Computing Cluster. The authors are grateful to the anonymous reviewers for their constructive comments, which helped improve the quality and clarity of the article.

References

- Deaglan Bartlett, Harry Desmond, and Pedro Ferreira. Priors for symbolic regression. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, 2023. URL <https://doi.org/10.1145/3583133.3596327>.
- Shankar Bhamidi, Guy Bresler, and Allan Sly. Mixing time of exponential random graphs. In *49th Annual IEEE Symposium on Foundations of Computer Science*, 2008. URL <https://doi.org/10.1109/FOCS.2008.75>.
- David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: a review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. URL <https://doi.org/10.1080/01621459.2017.1285773>.
- Frimpong Boadu, Ahhyun Lee, and Jianlin Cheng. Deep learning methods for protein function prediction. *Proteomics*, 25(1-2):2300471, 2025. URL <https://doi.org/10.1002/pmic.202300471>.
- Leo Breiman, Jerome Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees (1st ed.)*. Chapman and Hall/CRC, 1984. URL <https://doi.org/10.1201/9781315139470>.
- Kevin René Broløs, Meera Vieira Machado, Chris Cave, Jaan Kasak, Valdemar Stentoft-Hansen, Victor Galindo Batanero, Tom Jelen, and Casper Wilstrup. An approach to symbolic regression using feyn, 2021. URL <https://arxiv.org/abs/2104.05417>.
- Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. URL <https://doi.org/10.1073/pnas.1517384113>.
- Bogdan Burlacu, Gabriel Kronberger, and Michael Komenda. Operon C++: an efficient genetic programming framework for symbolic regression. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, 2020. URL <https://doi.org/10.1145/3377929.3398099>.
- Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018. URL <https://doi.org/10.1038/s41586-018-0337-2>.
- Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. Bayesian CART model search. *Journal of the American Statistical Association*, 93(443):935–948, 1998. URL <https://doi.org/10.1080/01621459.1998.10473750>.
- Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010. URL <https://doi.org/10.1214/09-AOAS285>.
- Miles Cranmer. Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl, 2023. URL <https://arxiv.org/abs/2305.01582>.
- David G. T. Denison, Bani K. Mallick, and Adrian F. M. Smith. A Bayesian CART algorithm. *Biometrika*, 85(2):363–377, 1998. URL <https://doi.org/10.1093/biomet/85.2.363>.
- Richard P. Feynman, Robert B. Leighton, and Matthew Sands. *The Feynman Lectures on Physics, Vol. I: Mainly Mechanics, Radiation, and Heat*. Addison-Wesley, 1963. URL <https://books.google.com/books>.
- Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13(70):2171–2175, 2012. URL <http://jmlr.org/papers/v13/fortin12a.html>.
- Ryan Giordano, Martin Ingram, and Tamara Broderick. Black box variational inference with a deterministic objective: faster, more accurate, and even more black box. *Journal of Machine Learning Research*, 25(18):1–39, 2024. URL <http://jmlr.org/papers/v25/23-1015.html>.
- Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, 2011. URL <https://proceedings.neurips.cc>.
- Arya Grayeli, Atharva Sehgal, Omar Costilla-Reyes, Miles Cranmer, and Swarat Chaudhuri. Symbolic regression with a learned concept library. In *Advances in Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=B7S4jJG1vl>.

- Roger Guimerà, Ignasi Reichardt, Antoni Aguilar-Mogas, Francesco A Massucci, Manuel Miranda, Jordi Pallarès, and Marta Sales-Pardo. A Bayesian machine scientist to aid in the solution of challenging scientific problems. *Science advances*, 6(5):eaav6971, 2020. URL <https://doi.org/10.1126/sciadv.aav6971>.
- Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, Yunlong Feng, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. Qwen2.5-coder technical report, 2024. URL <https://arxiv.org/abs/2409.12186>.
- Guilherme Seidyo Imai Aldeia, Hengzhe Zhang, Geoffrey Bomarito, Miles Cranmer, Alcides Fonseca, Bogdan Burlacu, William G La Cava, and Fabrício Olivetti de França. Call for action: towards the next generation of symbolic regression benchmark. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2025. URL <https://doi.org/10.1145/3712255.3734309>.
- Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=rkE3y85ee>.
- William H. Jefferys and James O. Berger. Ockham’s razor and Bayesian analysis. *American Scientist*, 80(1):64–72, 1992. URL <https://www.jstor.org/stable/29774559>.
- Ying Jin, Weilin Fu, Jian Kang, Jiadong Guo, and Jian Guo. Bayesian Symbolic Regression, 2020. URL <https://arxiv.org/abs/1910.08892>.
- Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999. URL <https://doi.org/10.1023/A:1007665907178>.
- Pierre-Alexandre Kamienny, Stéphane d’Ascoli, Guillaume Lample, and François Charton. End-to-end symbolic regression with transformers. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=GoOuIrDHG_Y.
- Michael F. Korn. Accuracy in symbolic regression. In *Genetic Programming Theory and Practice IX*, 2011. URL https://doi.org/10.1007/978-1-4614-1770-5_8.
- John R Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and computing*, 4(2):87–112, 1994. URL <https://doi.org/10.1007/BF00175355>.
- Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automatic differentiation variational inference. *Journal of machine learning research*, 18(14):1–45, 2017. URL <https://jmlr.org/papers/v18/16-107.html>.
- William La Cava. SRBench: a living benchmark for symbolic regression, 2026. URL <https://cavalab.org/srbench/>.
- William La Cava, Bogdan Burlacu, Marco Virgolin, Michael Kommenda, Patryk Orzechowski, Fabrício Olivetti de França, Ying Jin, and Jason H Moore. Contemporary symbolic regression methods and their relative performance. In *Advances in neural information processing systems*, 2021. URL <https://openreview.net/forum?id=xVQMrDLyGst>.
- Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: differentiable architecture search. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=S1eYHoC5FX>.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: a continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=S1je5L5gl>.
- David Madigan and Adrian E Raftery. Model selection and accounting for model uncertainty in graphical models using Occam’s window. *Journal of the American Statistical Association*, 89(428):1535–1546, 1994. URL <https://doi.org/10.1080/01621459.1994.10476894>.
- Nour Makke and Sanjay Chawla. Interpretable scientific discovery with symbolic regression: a review. *Artificial Intelligence Review*, 57(1):2, 2024. URL <https://doi.org/10.1007/s10462-023-10622-0>.
- Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006. URL <https://doi.org/10.1007/978-0-387-40065-5>.
- Brenden K Petersen, Mikel Landajuela Larma, Terrell N. Mundhenk, Claudio Prata Santiago, Soo Kyung Kim, and Joanne Taery Kim. Deep symbolic regression: recovering mathematical expressions from data via risk-seeking policy gradients. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=m5Qsh0kBQG>.

- Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. URL <https://doi.org/10.1016/j.jcp.2018.10.045>.
- Louis B Rall. *Automatic Differentiation: Techniques and Applications*. Springer, 1981. URL <https://doi.org/10.1007/3-540-10861-0>.
- Rajesh Ranganath, Sean Gerrish, and David M. Blei. Black box variational inference. In *International Conference on Artificial Intelligence and Statistics*, 2014. URL <https://proceedings.mlr.press/v33/ranganath14.html>.
- Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT press, 2006. URL <https://doi.org/10.7551/mitpress/3206.001.0001>.
- Somjit Roy and Pritam Dey. VaSST: variational inference for symbolic regression using soft symbolic trees, 2026. URL <https://github.com/Roy-SR-007/VaSST>. Github Repository.
- Somjit Roy, Pritam Dey, Debdeep Pati, and Bani K. Mallick. Hierarchical Bayesian operator-induced symbolic regression trees for structural learning of scientific expressions, 2025. URL <https://arxiv.org/abs/2509.19710>.
- Sebastian Salazar. VaRT: variational regression trees. In *Advances in Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=qJrlz3SucN>.
- Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923): 81–85, 2009. URL <https://doi.org/10.1126/science.1165893>.
- Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K. Reddy. LLM-SR: scientific equation discovery via programming with large language models. In *International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=m2nmp8P5in>.
- Trevor Stephens. Genetic programming in python, with a scikit-learn inspired api: gplearn, 2016. URL <https://gplearn.readthedocs.io/en/latest/>.
- Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. URL <https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>.
- Silviu-Marian Udrescu and Max Tegmark. AI Feynman: a physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020. URL <https://doi.org/10.1126/sciadv.aay2631>.
- Marco Virgolin and Solon P Pissis. Symbolic regression is NP-hard. *Transactions on Machine Learning Research*, 2022. URL <https://openreview.net/forum?id=LTiaPxqe2e>.
- Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2): 1–305, 2008. URL <https://doi.org/10.1561/2200000001>.
- Guanjie Wang, Erpeng Wang, Zefeng Li, Jian Zhou, and Zhimei Sun. Exploring the mathematic equations behind the materials science data using interpretable symbolic regression. *Interdisciplinary Materials*, 3(5):637–657, 2024. URL <https://doi.org/10.1002/idm2.12180>.
- Runxiang Wang, Boxiao Wang, Kai Li, Yifan Zhang, and Jian Cheng. DrSR: LLM based Scientific Equation Discovery with Dual Reasoning from Data and Experience, 2025. URL <https://arxiv.org/abs/2506.04282>.
- Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992. URL <https://doi.org/10.1007/BF00992696>.
- Yilong Xu, Yang Liu, and Hao Sun. Reinforcement symbolic regression machine. In *International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=PJVUWpPnZC>.
- Jie Ying, Haowei Lin, Chao Yue, Yajie Chen, Chao Xiao, Quanqi Shi, Yitao Liang, Shing-Tung Yau, Yuan Zhou, and Jianzhu Ma. A neural symbolic model for space physics. *Nature Machine Intelligence*, 7(10):1726–1741, 2025. URL <https://doi.org/10.1038/s42256-025-01126-3>.
- Huijun Zhang, Yaxin Liu, Chongyu Zhang, and Ningyun Li. Machine learning methods for weather forecasting: a survey. *Atmosphere*, 16(1), 2025. URL <https://doi.org/10.3390/atmos16010082>.
- Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. TinyLlama: an open-source small language model, 2024. URL <https://arxiv.org/abs/2401.02385>.
- Krzysztof Łatuszyński, Matthew T. Moores, and Timothée Stumpf-Fétizon. MCMC for multi-modal distributions, 2025. URL <https://arxiv.org/abs/2501.05908>.

Supplementary Materials for VaSST: Variational Inference for Symbolic Regression using Soft Symbolic Trees

CONTENTS

A	Notations	15
B	Parameterization Details of Distribution Families used in VaSST	16
C	Derivations of Key Quantities	17
C.1	Marginalization of the Model Regression Parameters	17
C.2	Derivation of the Analytical Kullback-Leibler Divergence Terms	17
C.3	Derivation of LMPSE	19
D	Soft Evaluation Algorithms	21
D.1	Soft Evaluation at a Node	21
D.2	Complete Soft Evaluation	21
E	Algorithm for Approximation of $\mathcal{E}(\phi)$	22
F	The VaSST Algorithm	23
G	Algorithm for Sampling Hard Symbolic Trees	24
H	Discussion on the Gap induced by ELBO Approximation	25
I	Configurations of VaSST and Competing Methods	27
I.1	Operator Set	27
I.2	Experimental Settings of VaSST	27
I.3	Experimental Settings of <code>operon</code> , <code>gplearn</code> , <code>DEAP</code> , and <code>PySR</code>	28
I.4	Experimental Settings of <code>BMS</code> and <code>BSR</code>	28
I.5	Experimental Settings of <code>QLattice</code> and <code>DSR</code>	29
J	Additional Results for Simulation Experiments	30
J.1	Symbolic Expressions Learned by VaSST and Competing Methods	30
J.2	Out-of-sample RMSE for VaSST and Competing Methods	32
J.3	Top Ranked Symbolic Expressions Learned by VaSST with respect to LMPSE	33

K	Description of Feynman Equations	35
L	Additional Results for Feynman Equations	36
L.1	Symbolic Expressions Learned by VaSST and Competing Methods	36
L.2	Compute Runtimes for VaSST and Competing Bayesian SR Methods	41
L.3	Top Ranked Symbolic Expressions Learned by VaSST with respect to LMPSE	42
M	Distinction between VaSST and VaRT	48
N	Ablation Studies for VaSST	49
O	Computational Scaling and Gradient Variance Diagnostics for VaSST	51
O.1	Scaling of the Relaxed Symbolic Tree Parameterization	51
O.2	Gradient Variance Diagnostic	51
O.3	Results	51
P	Additional Experiments with Large Language Models	54

A NOTATIONS

Table A.1: Special functions and other notations.

Symbol	Name	Definition
$\mathbb{N}$	Natural numbers	Set $\{1, 2, 3, \dots\}$
$\mathbb{R}$	Real numbers	Set of real numbers
$\mathbb{R}^p$	p -dimensional Euclidean space	$\{\mathbf{x} = (x_1, \dots, x_p)^\top : x_i \in \mathbb{R}\}$
$\mathbb{R}^{m \times n}$	Real matrix space	Space of real $m \times n$ matrices
$ A $	Cardinality	Number of elements in a finite set A
$\Gamma(t)$	Gamma function	$\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} dx$
$\Psi(t)$	Digamma function	$\Psi(t) = \frac{d}{dt} \log \Gamma(t)$
$\mathcal{B}(\boldsymbol{\eta})$	Multivariate Beta function	$\mathcal{B}(\boldsymbol{\eta}) = \frac{\prod_{k=1}^m \Gamma(\eta_k)}{\Gamma(\sum_{k=1}^m \eta_k)}$
$\mathbf{1}_m$	Vector of ones	$(1, \dots, 1)^\top \in \mathbb{R}^m$
$\mathbf{0}_m$	Vector of zeros	$(0, \dots, 0)^\top \in \mathbb{R}^m$
$\mathbf{I}_m$	Identity matrix of order m	$(\mathbf{I}_m)_{ij} = \mathbf{1}\{i = j\}, \quad i, j = 1, \dots, m$
$ \mathbf{A} $	Determinant	Determinant of the matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$
Δ^m	m -dimensional simplex	$\{\mathbf{w} \in [0, \infty)^m : \mathbf{1}_m^\top \mathbf{w} = 1\}$
$\text{softmax}(\mathbf{z})$	Softmax function	$\text{softmax}(\mathbf{z})_k = \frac{\exp(z_k)}{\sum_{\ell=1}^m \exp(z_\ell)}, \quad k = 1, \dots, m$
$\sigma(t)$	Sigmoid function	$\sigma(t) = \frac{1}{1 + \exp(-t)}$
$O(g(n))$	Order notation	$f(n) = O(g(n))$ if $ f(n) \leq C g(n) $ for some constant $C > 0$ and sufficiently large n

Note: Notations listed here are used throughout the article.

B PARAMETERIZATION DETAILS OF DISTRIBUTION FAMILIES USED IN VaSST

The **VaSST** framework utilizes several standard probability distributions for model specification and prior construction. Although these distributions are widely used in the machine learning literature, multiple parameterizations are common for some of them across various sources. For clarity and reproducibility, we explicitly document the notation and parameterizations adopted in this work.

Normal Inverse-Gamma. We use the conjugate Normal Inverse-Gamma (NIG) prior for Gaussian linear regression parameters:

$$\boldsymbol{\beta} \mid \sigma^2 \sim \mathcal{N}_d(\boldsymbol{\mu}_0, \sigma^2 \boldsymbol{\Sigma}_0), \quad \sigma^2 \sim \text{IG}(a_0, b_0), \quad (\text{B.1})$$

where $d = K + 1$, $\boldsymbol{\mu}_0 \in \mathbb{R}^d$, $\boldsymbol{\Sigma}_0 \in \mathbb{R}^{d \times d}$ is symmetric positive definite, and $a_0, b_0 > 0$. Our Inverse-Gamma parameterization, i.e., $\text{IG}(a, b)$, uses the density:

$$\Pi(\sigma^2) = \frac{b^a}{\Gamma(a)} (\sigma^2)^{-(a+1)} \exp\left(-\frac{b}{\sigma^2}\right), \quad \sigma^2 > 0, \quad (\text{B.2})$$

where $\Gamma(\cdot)$ is the Gamma function as in [Table A.1](#). The conditional Gaussian density in [\(B.1\)](#) is:

$$\Pi(\boldsymbol{\beta} \mid \sigma^2) = (2\pi)^{-\frac{d}{2}} (\sigma^2)^{-\frac{d}{2}} |\boldsymbol{\Sigma}_0|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2\sigma^2} (\boldsymbol{\beta} - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_0^{-1} (\boldsymbol{\beta} - \boldsymbol{\mu}_0)\right\}. \quad (\text{B.3})$$

Equivalently, the joint prior is $\Pi(\boldsymbol{\beta}, \sigma^2) = \Pi(\boldsymbol{\beta} \mid \sigma^2) \Pi(\sigma^2)$.

Bernoulli. A Bernoulli random variable $E \sim \text{Ber}(p)$, with $p \in (0, 1)$, has support $\{0, 1\}$ and probability mass function:

$$\Pi(e) = p^e (1 - p)^{1-e}, \quad e \in \{0, 1\}. \quad (\text{B.4})$$

In our tree prior, the expansion indicator follows $e_{j\zeta} \sim \text{Ber}(p_\zeta)$ with $p_\zeta = \alpha(1 + d_\zeta)^{-\delta}$, where $\alpha \in (0, 1)$ and $\delta > 0$.

Dirichlet. For $m \geq 2$, a vector $\mathbf{w}$ follows a Dirichlet distribution with concentration vector $\boldsymbol{\eta} = (\eta_1, \dots, \eta_m)^\top \in (0, \infty)^m$, i.e., $\mathbf{w} \sim \text{Dir}(\boldsymbol{\eta})$, having density:

$$p(\mathbf{w} \mid \boldsymbol{\eta}) = \frac{1}{\mathcal{B}(\boldsymbol{\eta})} \prod_{k=1}^m w_k^{\eta_k - 1}, \quad \mathbf{w} \in \Delta^m, \quad (\text{B.5})$$

where $\mathcal{B}(\cdot)$ is the multivariate Beta function and Δ^m is the m -dimensional simplex, as defined in [Table A.1](#). In **VaSST**, the operator and feature weight vectors follow $\mathbf{w}_{\text{op}} \sim \text{Dir}(\boldsymbol{\eta}_{\text{op}})$ and $\mathbf{w}_{\text{ft}} \sim \text{Dir}(\boldsymbol{\eta}_{\text{ft}})$, where $\mathbf{w}_{\text{op}} \in \Delta^{|\mathbf{O}|}$ and $\mathbf{w}_{\text{ft}} \in \Delta^p$, respectively.

Categorical. Let $\mathbf{w} \in \Delta^m$. A Categorical random variable, $c \sim \text{Cat}(\mathbf{w})$, takes values in $\{1, \dots, m\}$ with probability mass function:

$$\Pi(c = j \mid \mathbf{w}) = w_j, \quad j \in \{1, \dots, m\} \quad (\text{B.6})$$

In **VaSST**, conditional on weights, we assign operators and features via:

$$o_{j\zeta} \mid \mathbf{w}_{\text{op}} \sim \text{Cat}(\mathbf{w}_{\text{op}}), \quad h_{j\zeta} \mid \mathbf{w}_{\text{ft}} \sim \text{Cat}(\mathbf{w}_{\text{ft}}), \quad o_{j\zeta} \in \mathbf{O}, \text{ feature } h_{j\zeta}. \quad (\text{B.7})$$

When needed, we identify the operator set $\mathbf{O}$ and the set of features with index sets of sizes $|\mathbf{O}|$ and p , respectively, so that [\(B.6\)](#) applies directly.

Uniform. A continuous random variable $u \sim \text{Unif}(a, b)$, with $a < b$, has support on the interval (a, b) and probability density function:

$$\Pi(u) = \frac{1}{b-a}, \quad a < u < b, \quad (\text{B.8})$$

and $\Pi(u) = 0$ otherwise. In the Binary-Concrete and Gumbel-Softmax continuous relaxations in [Section 4](#), the $\text{Unif}(0, 1)$ distribution was used to reparameterize the discrete structural variables. Further, in simulation experiments in [Section 5.1](#), uniform distributions were used to generate the input variables.

C DERIVATIONS OF KEY QUANTITIES

C.1 MARGINALIZATION OF THE MODEL REGRESSION PARAMETERS

The full joint posterior distribution as stated in (6) in Section 4 is:

$$\Pi(\Theta, \boldsymbol{\beta}, \sigma^2 \mid \mathbb{D}_n) \propto p(\mathbf{y} \mid \mathbf{T}, \boldsymbol{\beta}, \sigma^2) \Pi(\boldsymbol{\beta}, \sigma^2) \Pi(\mathbf{w}_{\text{op}}) \Pi(\mathbf{w}_{\text{ft}}) \prod_{j=1}^K \prod_{\zeta \in \mathbb{Z}_D} \Pi(e_{j\zeta}) \Pi(o_{j\zeta} \mid \mathbf{w}_{\text{op}}) \Pi(h_{j\zeta} \mid \mathbf{w}_{\text{ft}}). \quad (\text{C.1})$$

Under the Gaussian likelihood and the conjugate NIG prior:

$$\mathbf{y} \mid \boldsymbol{\beta}, \sigma^2, \mathbf{T} \sim \mathcal{N}(\mathbf{T}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n), \quad \boldsymbol{\beta} \mid \sigma^2 \sim \mathcal{N}_{K+1}(\boldsymbol{\mu}_0, \sigma^2 \boldsymbol{\Sigma}_0), \quad \sigma^2 \sim \text{IG}(a_0, b_0),$$

the posterior remains NIG:

$$\boldsymbol{\beta} \mid \sigma^2, \mathbf{T}, \mathbf{y} \sim \mathcal{N}_{K+1}(\boldsymbol{\mu}_n, \sigma^2 \boldsymbol{\Sigma}_n), \quad \sigma^2 \mid \mathbf{T}, \mathbf{y} \sim \text{IG}(a_n, b_n), \quad (\text{C.2})$$

where the posterior hyperparameters are:

$$\boldsymbol{\Sigma}_n^{-1} = \boldsymbol{\Sigma}_0^{-1} + \mathbf{T}^\top \mathbf{T}, \quad \boldsymbol{\mu}_n = \boldsymbol{\Sigma}_n (\boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0 + \mathbf{T}^\top \mathbf{y}), \quad a_n = a_0 + \frac{n}{2}, \quad b_n = b_0 + \frac{1}{2} \left(\mathbf{y}^\top \mathbf{y} + \boldsymbol{\mu}_0^\top \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0 - \boldsymbol{\mu}_n^\top \boldsymbol{\Sigma}_n^{-1} \boldsymbol{\mu}_n \right).$$

Integrating out $(\boldsymbol{\beta}, \sigma^2)$ yields the marginal likelihood:

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{T}) &= \int p(\mathbf{y} \mid \mathbf{T}, \boldsymbol{\beta}, \sigma^2) \Pi(\boldsymbol{\beta}, \sigma^2) d\boldsymbol{\beta} d\sigma^2 \\ &= (2\pi)^{-\frac{n}{2}} \frac{\Gamma(a_n)}{\Gamma(a_0)} \frac{|\boldsymbol{\Sigma}_n|^{\frac{1}{2}} b_0^{a_0}}{|\boldsymbol{\Sigma}_0|^{\frac{1}{2}} b_n^{a_n}}, \end{aligned} \quad (\text{C.3})$$

where $\Gamma(\cdot)$ is the Gamma function as in Table A.1. Up to additive constants independent of $\mathbf{T}$, the log-marginal likelihood is therefore:

$$\log p(\mathbf{y} \mid \mathbf{T}) = \frac{1}{2} \log |\boldsymbol{\Sigma}_n| - a_n \log b_n + \log \Gamma(a_n) + C, \quad (\text{C.4})$$

where C collects terms independent of $\mathbf{T}$. Consequently, the marginal posterior over the tree parameters Θ becomes:

$$\Pi(\Theta \mid \mathbb{D}_n) \propto p(\mathbf{y} \mid \mathbf{T}) \Pi(\mathbf{w}_{\text{op}}) \Pi(\mathbf{w}_{\text{ft}}) \prod_{j=1}^K \prod_{\zeta \in \mathbb{Z}_D} \Pi(e_{j\zeta}) \Pi(o_{j\zeta} \mid \mathbf{w}_{\text{op}}) \Pi(h_{j\zeta} \mid \mathbf{w}_{\text{ft}}), \quad (\text{C.5})$$

which is the quantity of interest we approximate by optimizing the variational objective.

C.2 DERIVATION OF THE ANALYTICAL KULLBACK-LEIBLER DIVERGENCE TERMS

In this section, we provide a detailed derivation of the Kullback-Leibler (KL) term in (11) of Section 3.2. Specifically, this term corresponds to the KL divergence between the variational posterior $q(\Theta)$ in (9) and the prior $\Pi(\Theta)$ over the tree parameters defined by (3) and (5).

$$\begin{aligned} \text{KL}[q(\Theta) \parallel \Pi(\Theta)] &= \mathbb{E}_q \left[\log \frac{q(\Theta)}{\Pi(\Theta)} \right] \\ &= \mathbb{E}_q \left[\log \frac{q(\mathbf{w}_{\text{op}}) q(\mathbf{w}_{\text{ft}}) \prod_{j=1}^K \prod_{\zeta \in \mathbb{Z}_D} q(e_{j\zeta}) q(o_{j\zeta}) q(h_{j\zeta})}{\Pi(\mathbf{w}_{\text{op}}) \Pi(\mathbf{w}_{\text{ft}}) \prod_{j=1}^K \prod_{\zeta \in \mathbb{Z}_D} \Pi(e_{j\zeta}) \Pi(o_{j\zeta} \mid \mathbf{w}_{\text{op}}) \Pi(h_{j\zeta} \mid \mathbf{w}_{\text{ft}})} \right] \\ &= \mathbb{E}_q \left[\log \frac{q(\mathbf{w}_{\text{op}})}{\Pi(\mathbf{w}_{\text{op}})} + \log \frac{q(\mathbf{w}_{\text{ft}})}{\Pi(\mathbf{w}_{\text{ft}})} + \sum_{j=1}^K \sum_{\zeta \in \mathbb{Z}_D} \left(\log \frac{q(e_{j\zeta})}{\Pi(e_{j\zeta})} + \log \frac{q(o_{j\zeta})}{\Pi(o_{j\zeta} \mid \mathbf{w}_{\text{op}})} + \log \frac{q(h_{j\zeta})}{\Pi(h_{j\zeta} \mid \mathbf{w}_{\text{ft}})} \right) \right] \\ &= \text{KL}[q(\mathbf{w}_{\text{op}}) \parallel \Pi(\mathbf{w}_{\text{op}})] + \text{KL}[q(\mathbf{w}_{\text{ft}}) \parallel \Pi(\mathbf{w}_{\text{ft}})] + \sum_{j=1}^K \sum_{\zeta \in \mathbb{Z}_D} \left[\text{KL}[q(e_{j\zeta}) \parallel \Pi(e_{j\zeta})] \right. \\ &\quad \left. + \mathbb{E}_{q(\mathbf{w}_{\text{op}})} \text{KL}[q(o_{j\zeta}) \parallel \Pi(o_{j\zeta} \mid \mathbf{w}_{\text{op}})] + \mathbb{E}_{q(\mathbf{w}_{\text{ft}})} \text{KL}[q(h_{j\zeta}) \parallel \Pi(h_{j\zeta} \mid \mathbf{w}_{\text{ft}})] \right]. \end{aligned} \quad (\text{C.6})$$

Now, we derive the analytic forms of each of the terms in the final sum in (C.6).

Lemma C.1 (KL divergence between Dirichlet distributions). *Let $q \equiv \text{Dir}(\tilde{\boldsymbol{\eta}})$ and $\pi \equiv \text{Dir}(\boldsymbol{\eta})$ on Δ^m , where $\tilde{\boldsymbol{\eta}}, \boldsymbol{\eta} \in (0, \infty)^m$. Define $\tilde{\eta}_0 = \sum_{k=1}^m \tilde{\eta}_k$. Then:*

$$\text{KL}[\text{Dir}(\tilde{\boldsymbol{\eta}}) \parallel \text{Dir}(\boldsymbol{\eta})] = \log \frac{\mathcal{B}(\boldsymbol{\eta})}{\mathcal{B}(\tilde{\boldsymbol{\eta}})} + \sum_{k=1}^m (\tilde{\eta}_k - \eta_k) [\Psi(\tilde{\eta}_k) - \Psi(\tilde{\eta}_0)], \quad (\text{C.7})$$

where $\mathcal{B}(\cdot)$ and $\Psi(\cdot)$ are the multivariate Beta and Digamma functions, respectively, as defined in [Table A.1](#).

Proof. Let $\mathbf{w} \in \Delta^m$ such that $\mathbf{w} \sim q$. From the definition of KL divergence, we conclude:

$$\begin{aligned} \text{KL}[\text{Dir}(\tilde{\boldsymbol{\eta}}) \parallel \text{Dir}(\boldsymbol{\eta})] &= \mathbb{E}_q \left[\log \frac{q(\mathbf{w})}{\pi(\mathbf{w})} \right] = \mathbb{E}_q \left[\log \frac{\mathcal{B}(\boldsymbol{\eta})}{\mathcal{B}(\tilde{\boldsymbol{\eta}})} + \sum_{k=1}^m (\tilde{\eta}_k - \eta_k) \log w_k \right] \\ &= \log \frac{\mathcal{B}(\boldsymbol{\eta})}{\mathcal{B}(\tilde{\boldsymbol{\eta}})} + \sum_{k=1}^m (\tilde{\eta}_k - \eta_k) \mathbb{E}_q[\log w_k]. \end{aligned} \quad (\text{C.8})$$

It remains to compute $\mathbb{E}_q[\log w_k]$ under $q \equiv \text{Dir}(\tilde{\boldsymbol{\eta}})$. For any $k \in \{1, \dots, m\}$, using differentiation under the integral sign:

$$0 = \frac{\partial}{\partial \tilde{\eta}_k} \int_{\Delta^m} q(\mathbf{w}) d\mathbf{w} = \int_{\Delta^m} \frac{\partial}{\partial \tilde{\eta}_k} q(\mathbf{w}) d\mathbf{w}. \quad (\text{C.9})$$

Moreover:

$$\begin{aligned} \log q(\mathbf{w}) &= -\log \mathcal{B}(\tilde{\boldsymbol{\eta}}) + \sum_{r=1}^m (\tilde{\eta}_r - 1) \log w_r \\ \Rightarrow \frac{\partial}{\partial \tilde{\eta}_k} \log q(\mathbf{w}) &= -\frac{\partial}{\partial \tilde{\eta}_k} \log \mathcal{B}(\tilde{\boldsymbol{\eta}}) + \log w_k \\ \Rightarrow \frac{\partial}{\partial \tilde{\eta}_k} q(\mathbf{w}) &= q(\mathbf{w}) \left(-\frac{\partial}{\partial \tilde{\eta}_k} \log \mathcal{B}(\tilde{\boldsymbol{\eta}}) + \log w_k \right). \end{aligned} \quad (\text{C.10})$$

Substituting (C.10) into (C.9) and noting that $\int_{\Delta^m} q(\mathbf{w}) d\mathbf{w} = 1$ gives:

$$0 = -\frac{\partial}{\partial \tilde{\eta}_k} \log \mathcal{B}(\tilde{\boldsymbol{\eta}}) + \int_{\Delta^m} q(\mathbf{w}) \log w_k d\mathbf{w} = -\frac{\partial}{\partial \tilde{\eta}_k} \log \mathcal{B}(\tilde{\boldsymbol{\eta}}) + \mathbb{E}_q[\log w_k], \quad (\text{C.11})$$

and therefore:

$$\mathbb{E}_q[\log w_k] = \frac{\partial}{\partial \tilde{\eta}_k} \log \mathcal{B}(\tilde{\boldsymbol{\eta}}) = \frac{\partial}{\partial \tilde{\eta}_k} \left[\sum_{r=1}^m \log \Gamma(\tilde{\eta}_r) - \log \Gamma(\tilde{\eta}_0) \right] = \Psi(\tilde{\eta}_k) - \Psi(\tilde{\eta}_0). \quad (\text{C.12})$$

Substituting this expression into (C.8) gives (C.7), completing the proof. $\square$

Now, an application of [Lemma C.1](#) yields the following:

$$\text{KL}[q(\mathbf{w}_{\text{op}}) \parallel \Pi(\mathbf{w}_{\text{op}})] = \log \frac{\mathcal{B}(\boldsymbol{\eta}_{\text{op}})}{\mathcal{B}(\tilde{\boldsymbol{\eta}}_{\text{op}})} + \sum_{k=1}^{|\mathbf{O}|} (\tilde{\eta}_{\text{op},k} - \eta_{\text{op},k}) [\Psi(\tilde{\eta}_{\text{op},k}) - \Psi(\mathbf{1}_{|\mathbf{O}|}^\top \tilde{\boldsymbol{\eta}}_{\text{op}})], \quad (\text{C.13})$$

$$\text{KL}[q(\mathbf{w}_{\text{ft}}) \parallel \Pi(\mathbf{w}_{\text{ft}})] = \log \frac{\mathcal{B}(\boldsymbol{\eta}_{\text{ft}})}{\mathcal{B}(\tilde{\boldsymbol{\eta}}_{\text{ft}})} + \sum_{k=1}^p (\tilde{\eta}_{\text{ft},k} - \eta_{\text{ft},k}) [\Psi(\tilde{\eta}_{\text{ft},k}) - \Psi(\mathbf{1}_p^\top \tilde{\boldsymbol{\eta}}_{\text{ft}})], \quad (\text{C.14})$$

where $\mathbf{1}_m$ is a m -dimensional vector of ones as in [Table A.1](#). By definition, the KL divergence between two Bernoulli distributions $\text{Ber}(\tilde{p})$ and $\text{Ber}(p)$ is:

$$\text{KL}[\text{Ber}(\tilde{p}) \parallel \text{Ber}(p)] = \tilde{p} \log \frac{\tilde{p}}{p} + (1 - \tilde{p}) \log \frac{1 - \tilde{p}}{1 - p}. \quad (\text{C.15})$$

Consequently using (C.15):

$$\text{KL}[q(e_{j\zeta}) \parallel \Pi(e_{j\zeta})] = \tilde{p}_{j\zeta} \log \frac{\tilde{p}_{j\zeta}}{p_{j\zeta}} + (1 - \tilde{p}_{j\zeta}) \log \frac{1 - \tilde{p}_{j\zeta}}{1 - p_{j\zeta}}. \quad (\text{C.16})$$

Similarly, the KL divergence between two categorical distributions $\text{Cat}(\tilde{\boldsymbol{\pi}})$ and $\text{Cat}(\mathbf{w})$ is:

$$\text{KL}[\text{Cat}(\tilde{\boldsymbol{\pi}}) \parallel \text{Cat}(\mathbf{w})] = \sum_{k=1}^m \tilde{\pi}_k \log \frac{\tilde{\pi}_k}{w_k} = \sum_{k=1}^m \tilde{\pi}_k [\log \tilde{\pi}_k - \log w_k], \quad (\text{C.17})$$

which implies:

$$\begin{aligned} \mathbb{E}_{q(\mathbf{w}_{\text{op}})} \text{KL}[q(o_{j\zeta}) \parallel \Pi(o_{j\zeta} \mid \mathbf{w}_{\text{op}})] &= \sum_{k=1}^{|\mathbf{O}|} \tilde{\pi}_{j\zeta,k}^{\text{op}} \left[\log(\tilde{\pi}_{j\zeta,k}^{\text{op}}) - \mathbb{E}_{q(\mathbf{w}_{\text{op}})}[\log w_k] \right] \\ &= \sum_{k=1}^{|\mathbf{O}|} \tilde{\pi}_{j\zeta,k}^{\text{op}} \left[\log(\tilde{\pi}_{j\zeta,k}^{\text{op}}) - \Psi(\tilde{\boldsymbol{\eta}}_{\text{op},k}) + \Psi(\mathbf{1}_{|\mathbf{O}|}^\top \tilde{\boldsymbol{\eta}}_{\text{op}}) \right], \end{aligned} \quad (\text{C.18})$$

where the last equality follows from (C.12) in the proof of Lemma C.1. Analogously:

$$\mathbb{E}_{q(\mathbf{w}_{\text{ft}})} \text{KL}[q(h_{j\zeta}) \parallel \Pi(h_{j\zeta} \mid \mathbf{w}_{\text{ft}})] = \sum_{k=1}^p \tilde{\pi}_{j\zeta,k}^{\text{ft}} \left[\log(\tilde{\pi}_{j\zeta,k}^{\text{ft}}) - \Psi(\tilde{\boldsymbol{\eta}}_{\text{ft},k}) + \Psi(\mathbf{1}_p^\top \tilde{\boldsymbol{\eta}}_{\text{ft}}) \right]. \quad (\text{C.19})$$

Putting together (C.13), (C.14), (C.16), (C.18), and (C.19) in (C.6), we get the following composite expression for the KL component:

$$\begin{aligned} \text{KL}[q(\boldsymbol{\Theta}) \parallel \Pi(\boldsymbol{\Theta})] &= \log \frac{\mathcal{B}(\boldsymbol{\eta}_{\text{op}})}{\mathcal{B}(\tilde{\boldsymbol{\eta}}_{\text{op}})} + \sum_{k=1}^{|\mathbf{O}|} (\tilde{\boldsymbol{\eta}}_{\text{op},k} - \boldsymbol{\eta}_{\text{op},k}) \left[\Psi(\tilde{\boldsymbol{\eta}}_{\text{op},k}) - \Psi(\mathbf{1}_{|\mathbf{O}|}^\top \tilde{\boldsymbol{\eta}}_{\text{op}}) \right] + \log \frac{\mathcal{B}(\boldsymbol{\eta}_{\text{ft}})}{\mathcal{B}(\tilde{\boldsymbol{\eta}}_{\text{ft}})} \\ &\quad + \sum_{k=1}^p (\tilde{\boldsymbol{\eta}}_{\text{ft},k} - \boldsymbol{\eta}_{\text{ft},k}) \left[\Psi(\tilde{\boldsymbol{\eta}}_{\text{ft},k}) - \Psi(\mathbf{1}_p^\top \tilde{\boldsymbol{\eta}}_{\text{ft}}) \right] + \sum_{j=1}^K \sum_{\zeta \in \mathbf{Z}_D} \left[\tilde{p}_{j\zeta} \log \frac{\tilde{p}_{j\zeta}}{p_{j\zeta}} + (1 - \tilde{p}_{j\zeta}) \log \frac{1 - \tilde{p}_{j\zeta}}{1 - p_{j\zeta}} \right. \\ &\quad \left. + \sum_{k=1}^{|\mathbf{O}|} \tilde{\pi}_{j\zeta,k}^{\text{op}} \left[\log(\tilde{\pi}_{j\zeta,k}^{\text{op}}) - \Psi(\tilde{\boldsymbol{\eta}}_{\text{op},k}) + \Psi(\mathbf{1}_{|\mathbf{O}|}^\top \tilde{\boldsymbol{\eta}}_{\text{op}}) \right] + \sum_{k=1}^p \tilde{\pi}_{j\zeta,k}^{\text{ft}} \left[\log(\tilde{\pi}_{j\zeta,k}^{\text{ft}}) - \Psi(\tilde{\boldsymbol{\eta}}_{\text{ft},k}) + \Psi(\mathbf{1}_p^\top \tilde{\boldsymbol{\eta}}_{\text{ft}}) \right] \right]. \end{aligned} \quad (\text{C.20})$$

C.3 DERIVATION OF LMPSE

From (1) and (4), the joint posterior distribution induced over the tree structures $\{\mathbf{T}(f_j)\}_{j=1}^K$, coefficient vector $\boldsymbol{\beta}$, noise variance σ^2 , and operator and feature weights $\mathbf{w}_{\text{op}}$ and $\mathbf{w}_{\text{ft}}$ is:

$$\Pi(\{\mathbf{T}(f_j)\}_{j=1}^K, \boldsymbol{\beta}, \sigma^2, \mathbf{w}_{\text{op}}, \mathbf{w}_{\text{ft}} \mid \mathbb{D}_n) \propto p(\mathbf{y} \mid \mathbf{T}, \boldsymbol{\beta}, \sigma^2) \Pi(\boldsymbol{\beta}, \sigma^2) \Pi(\mathbf{w}_{\text{op}}) \Pi(\mathbf{w}_{\text{ft}}) \prod_{j=1}^K \prod_{\zeta \in \mathbf{Z}(f_j)} \Pi(e_{j\zeta}) \Pi(o_{j\zeta} \mid \mathbf{w}_{\text{op}}) \Pi(h_{j\zeta} \mid \mathbf{w}_{\text{ft}}). \quad (\text{C.21})$$

The marginal posterior over $\{\mathbf{T}(f_j)\}_{j=1}^K$ is:

$$\begin{aligned} \Pi(\{\mathbf{T}(f_j)\}_{j=1}^K \mid \mathbb{D}_n) &\propto \mathbb{E}_{\boldsymbol{\beta}, \sigma^2} [p(\mathbf{y} \mid \mathbf{T}, \boldsymbol{\beta}, \sigma^2)] \mathbb{E}_{\mathbf{w}_{\text{op}}, \mathbf{w}_{\text{ft}}} \left[\prod_{j=1}^K \prod_{\zeta \in \mathbf{Z}(f_j)} \Pi(e_{j\zeta}) \Pi(o_{j\zeta} \mid \mathbf{w}_{\text{op}}) \Pi(h_{j\zeta} \mid \mathbf{w}_{\text{ft}}) \right] \\ &= p(\mathbf{y} \mid \mathbf{T}) \times \underbrace{\prod_{d=0}^D [\text{ps}(d)^{\text{nonterm}(d)} (1 - \text{ps}(d))^{\text{term}(d)}]}_{\prod_{j=1}^K \prod_{\zeta \in \mathbf{Z}(f_j)} \Pi(e_{j\zeta})} \\ &\quad \times \underbrace{\int \int \prod_{\ell=1}^{|\mathbf{O}|} w_{\text{op},\ell}^{\gamma_\ell}}_{\prod_{j=1}^K \prod_{\zeta \in \mathbf{Z}(f_j)} \Pi(o_{j\zeta})} \times \underbrace{\prod_{m=1}^p w_{\text{ft},m}^{\rho_m}}_{\prod_{j=1}^K \prod_{\zeta \in \mathbf{Z}(f_j)} \Pi(h_{j\zeta})} \times \underbrace{\frac{\prod_{\ell=1}^{|\mathbf{O}|} w_{\text{op},\ell}^{\eta_{\text{op},\ell}-1}}{\mathcal{B}(\boldsymbol{\eta}_{\text{op}})}}_{\mathbf{w}_{\text{op}} \sim \text{Dir}(\boldsymbol{\eta}_{\text{op}})} \times \underbrace{\frac{\prod_{m=1}^p w_{\text{ft},m}^{\eta_{\text{ft},m}-1}}{\mathcal{B}(\boldsymbol{\eta}_{\text{ft}})}}_{\mathbf{w}_{\text{ft}} \sim \text{Dir}(\boldsymbol{\eta}_{\text{ft}})} d\mathbf{w}_{\text{op}} d\mathbf{w}_{\text{ft}}, \end{aligned} \quad (\text{C.22})$$

where $\mathbf{w}_{\text{op}} = (w_{\text{op},1}, \dots, w_{\text{op},|\mathbf{O}|})^\top$, $\mathbf{w}_{\text{ft}} = (w_{\text{ft},1}, \dots, w_{\text{ft},p})^\top$, $\boldsymbol{\eta}_{\text{op}} = (\eta_{\text{op},1}, \dots, \eta_{\text{op},|\mathbf{O}|})^\top$, $\boldsymbol{\eta}_{\text{ft}} = (\eta_{\text{ft},1}, \dots, \eta_{\text{ft},p})^\top$, γ_ℓ and ρ_m denote the total number of occurrences of the ℓ th operator and the m th feature among the K symbolic trees $\{\mathbf{T}(f_j)\}_{j=1}^K$,

$\text{ps}(d) = \alpha(1+d)^{-\delta}$ is the split probability at depth d , $\text{term}(d)$ and $\text{nonterm}(d)$ denote the total number of terminal and nonterminal nodes at depth d among $\{\mathbb{T}(f_j)\}_{j=1}^K$. Define $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_{|\mathbf{O}|})^\top$ and $\boldsymbol{\rho} = (\rho_1, \dots, \rho_p)^\top$. Then the marginal posterior from (C.22) is:

$$\Pi(\{\mathbb{T}(f_j)\}_{j=1}^K \mid \mathbb{D}_n) \propto p(\mathbf{y} \mid \mathbf{T}) \times \prod_{d=0}^D \left[\text{ps}(d)^{\text{nonterm}(d)} (1 - \text{ps}(d))^{\text{term}(d)} \right] \frac{\mathcal{B}(\boldsymbol{\eta}_{\text{op}} + \boldsymbol{\gamma})}{\mathcal{B}(\boldsymbol{\eta}_{\text{op}})} \frac{\mathcal{B}(\boldsymbol{\eta}_{\text{ft}} + \boldsymbol{\rho})}{\mathcal{B}(\boldsymbol{\eta}_{\text{ft}})}, \quad (\text{C.23})$$

Therefore the LMPSE is:

$$\begin{aligned} \text{LMPSE}(\{\mathbb{T}(f_j)\}_{j=1}^K) &= \log \Pi(\{\mathbb{T}(f_j)\}_{j=1}^K \mid \mathbb{D}_n) \\ &= \log p(\mathbf{y} \mid \mathbf{T}) + \sum_{d=0}^D \left[\text{nonterm}(d) \log \text{ps}(d) + \text{term}(d) \log(1 - \text{ps}(d)) \right] + \log \frac{\mathcal{B}(\boldsymbol{\eta}_{\text{op}} + \boldsymbol{\gamma})}{\mathcal{B}(\boldsymbol{\eta}_{\text{op}})} + \log \frac{\mathcal{B}(\boldsymbol{\eta}_{\text{ft}} + \boldsymbol{\rho})}{\mathcal{B}(\boldsymbol{\eta}_{\text{ft}})}. \end{aligned}$$

D SOFT EVALUATION ALGORITHMS

D.1 SOFT EVALUATION AT A NODE

Algorithm 1: The $\text{SOFTVALATNODE}(\zeta', \mathbf{x}_i, j, \mathbf{O}, \{\tilde{e}_{j\zeta}, \tilde{\mathbf{o}}_{j\zeta}, \tilde{\mathbf{h}}_{j\zeta} : \zeta \in Z_D\})$ algorithm for evaluating a soft symbolic subtree rooted at a given node ζ' .

Input: A given node: ζ' ; feature vector: $\mathbf{x}_i \in \mathbb{R}^p$; soft symbolic tree index: $j \in \{1, \dots, K\}$; the operator set:

$\mathbf{O} = \mathbf{O}_u \cup \mathbf{O}_b$; soft relaxations: $\{\tilde{e}_{j\zeta}, \tilde{\mathbf{o}}_{j\zeta}, \tilde{\mathbf{h}}_{j\zeta} : \zeta \in Z_D\}$.

Output: $S_j^{\text{soft}}(\mathbf{x}_i; \zeta') \in \mathbb{R}$.

▷ *Soft feature evaluation*

$\text{Term}(\mathbf{x}_i, j, \zeta') \leftarrow \tilde{\mathbf{h}}_{j\zeta'}^\top \mathbf{x}_i$

▷ *Soft operator evaluation*

// $L(\zeta')$ and $R(\zeta')$ are the left and right children of ζ' , respectively

$\text{NontermUnary}(\mathbf{x}_i, j, \zeta') \leftarrow \sum_{u \in \mathbf{O}_u} \tilde{o}_{j\zeta', u} \cdot u(S_j^{\text{soft}}(\mathbf{x}_i, L(\zeta')))$

$\text{NontermBinary}(\mathbf{x}_i, j, \zeta') \leftarrow \sum_{b \in \mathbf{O}_b} \tilde{o}_{j\zeta', b} \cdot b(S_j^{\text{soft}}(\mathbf{x}_i; L(\zeta')), S_j^{\text{soft}}(\mathbf{x}_i; R(\zeta')))$

$\text{Nonterm}(\mathbf{x}_i, j, \zeta') \leftarrow \text{NontermUnary}(\mathbf{x}_i, j, \zeta') + \text{NontermBinary}(\mathbf{x}_i, j, \zeta')$

▷ *Complete soft evaluation at ζ'*

$S_j^{\text{soft}}(\mathbf{x}_i; \zeta') \leftarrow \tilde{e}_{j\zeta'} \text{Nonterm}(\mathbf{x}_i, j, \zeta') + (1 - \tilde{e}_{j\zeta'}) \text{Term}(\mathbf{x}_i, j, \zeta')$

return $S_j^{\text{soft}}(\mathbf{x}_i; \zeta')$

D.2 COMPLETE SOFT EVALUATION

Algorithm 2: The $\text{SOFTVAL}(\{\mathbf{x}_i \in \mathbb{R}^p\}_{i=1}^n, \mathbf{O}, \{\tilde{e}_{j\zeta}, \tilde{\mathbf{o}}_{j\zeta}, \tilde{\mathbf{h}}_{j\zeta} : \zeta \in Z_D\}_{j=1}^K)$ algorithm for evaluating an ensemble of K soft symbolic trees.

Input: Feature vector instances: $\{\mathbf{x}_i \in \mathbb{R}^p\}_{i=1}^n$; the operator set: $\mathbf{O} = \mathbf{O}_u \cup \mathbf{O}_b$; soft relaxations:

$\{\tilde{e}_{j\zeta}, \tilde{\mathbf{o}}_{j\zeta}, \tilde{\mathbf{h}}_{j\zeta} : \zeta \in Z_D\}_{j=1}^K$.

Output: $\mathbf{T}_{\text{soft}} \in \mathbb{R}^{n \times K+1}$.

for $i \leftarrow 1$ **to** n **do**

 ▷ *Intercept term set to 1*

$t_{i0} \leftarrow 1$

for $j \leftarrow 1$ **to** K **do**

 ▷ *j th soft symbolic tree evaluated at root node for $\mathbf{x}_i$*

$t_{ij} = S_j^{\text{soft}}(\mathbf{x}_i; \zeta' = 0)$ from Algorithm 1: SOFTVALATNODE

$\mathbf{T}_{\text{soft}} \leftarrow (t_{ij})_{1 \leq i \leq n, 0 \leq j \leq K}$

return $\mathbf{T}_{\text{soft}}$

E ALGORITHM FOR APPROXIMATION OF $\mathcal{E}(\phi)$

Algorithm 3: The APPROXELBO($\mathbb{D}_n, (\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, a_0, b_0, \boldsymbol{\eta}_{\text{op}}, \boldsymbol{\eta}_{\text{ft}}, \alpha, \delta), \mathbf{O}, \phi, \boldsymbol{\tau} = (\tau_{\text{ex}}, \tau_{\text{op}}, \tau_{\text{ft}}), S$) algorithm for computing Monte Carlo approximation of $\mathcal{E}(\phi)$.

Input: Data: $\mathbb{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$; full set of prior hyperparameters: $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, a_0, b_0, \boldsymbol{\eta}_{\text{op}}, \boldsymbol{\eta}_{\text{ft}}, \alpha, \delta)$; the operator set: $\mathbf{O} = \mathbf{O}_u \cup \mathbf{O}_b$; the variational parameter vector: ϕ ; the temperature parameters: $\boldsymbol{\tau} = (\tau_{\text{ex}}, \tau_{\text{op}}, \tau_{\text{ft}})$; the Monte Carlo sample size: S .

Output: $\widehat{\mathcal{E}}(\phi)$.

▷ Compute the Kullback-Leibler term

$K \leftarrow \text{KL}(q_\phi(\Theta) \parallel \Pi(\Theta))$, using (12)

for $s \leftarrow 1$ **to** S **do**

▷ Sample soft tree structural parameters from $\mathcal{P}_{\phi, \boldsymbol{\tau}}^{\text{soft}}$

$\{\tilde{e}_{j\zeta}^{(s)}, \tilde{\mathbf{o}}_{j\zeta}^{(s)}, \tilde{\mathbf{h}}_{j\zeta}^{(s)} : \zeta \in Z_D\}_{j=1}^K \sim \mathcal{P}_{\phi, \boldsymbol{\tau}}^{\text{soft}}$, using the temperature parameters

▷ Computing soft symbolic design matrix

$\mathbf{T}_{\text{soft}}^{(s)} \leftarrow \text{SOFT-EVAL}(\{\mathbf{x}_i \in \mathbb{R}^p\}_{i=1}^n, \mathbf{O}, \{\tilde{e}_{j\zeta}^{(s)}, \tilde{\mathbf{o}}_{j\zeta}^{(s)}, \tilde{\mathbf{h}}_{j\zeta}^{(s)} : \zeta \in Z_D\}_{j=1}^K)$

▷ Computing the marginal likelihood at $\mathbf{T}_{\text{soft}}^{(s)}$

$\text{ML}^{(s)} \leftarrow \log p(\mathbf{y} \mid \mathbf{T}_{\text{soft}}^{(s)})$, using (7)

$\widehat{\mathcal{E}}(\phi) \leftarrow \frac{1}{S} \sum_{s=1}^S \text{ML}^{(s)} - K$

return $\widehat{\mathcal{E}}(\phi)$

F THE VaSST ALGORITHM

Algorithm 4: The $\text{VaSST}(\mathbb{D}_n, (\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, a_0, b_0, \boldsymbol{\eta}_{\text{op}}, \boldsymbol{\eta}_{\text{ft}}, \alpha, \delta), \mathbf{O}, \phi^{\text{init}}, T, (\tau_{\text{start}}, \tau_{\text{end}}, T_\tau), S, \gamma, c)$ algorithm for SR using automatic differentiation-based black-box variational inference.

Input: Data: $\mathbb{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$; full set of prior hyperparameters: $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, a_0, b_0, \boldsymbol{\eta}_{\text{op}}, \boldsymbol{\eta}_{\text{ft}}, \alpha, \delta)$; the operator set:

$\mathbf{O} = \mathbf{O}_u \cup \mathbf{O}_b$; initial variational parameter vector: ϕ^{init} ; number of steps: T ; temperature schedule:

$(\tau_{\text{start}}, \tau_{\text{end}}, T_\tau)$; Monte Carlo sample size: S ; learning rate: γ ; gradient clipping threshold: c .

Output: Optimized variational parameter vector ϕ^* .

Set $\phi_0 \leftarrow \phi^{\text{init}}$

Initialize AdamW optimizer with learning rate γ

for $t \leftarrow 1$ **to** T **do**

$\triangleright$ Anneal temperature parameters

$\tau_t \leftarrow \tau_{\text{start}} + (\tau_{\text{end}} - \tau_{\text{start}}) \cdot \min\{\frac{t}{T_\tau}, 1\}$

 Set $(\tau_{\text{ex}}, \tau_{\text{op}}, \tau_{\text{ft}}) \leftarrow (\tau_t, \tau_t, \tau_t)$

 Set all gradients to zero

$\triangleright$ Compute stochastic ELBO estimate

$\hat{\mathcal{E}}(\phi) \leftarrow \text{APPROXELBO}(\mathbb{D}_n, (\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, a_0, b_0, \boldsymbol{\eta}_{\text{op}}, \boldsymbol{\eta}_{\text{ft}}, \alpha, \delta), \mathbf{O}, \phi, (\tau_{\text{ex}}, \tau_{\text{op}}, \tau_{\text{ft}}), S)$

$\triangleright$ minimize negative objective

$J \leftarrow -\hat{\mathcal{E}}(\phi)$

if J is not finite **then**

continue

 Backpropagate: compute $\nabla_\phi J$

if any component of $\nabla_\phi J$ is not finite **then**

 Set all gradients to zero; **continue**

 Clip gradients: $\|\nabla_\phi J\| \leq c$

$\phi_t \leftarrow \text{AdamW update step on } \phi$

Set $\phi^* \leftarrow \phi_T$

return ϕ^*

G ALGORITHM FOR SAMPLING HARD SYMBOLIC TREES

Algorithm 5: The $\text{SAMPLEHARD}(\mathbb{D}_n, (\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, a_0, b_0, \boldsymbol{\eta}_{\text{op}}, \boldsymbol{\eta}_{\text{ft}}, \alpha, \delta), \phi^*, \mathbf{O}, H)$ algorithm to draw samples of hard symbolic trees from learned soft structural representations.

Input: Data: $\mathbb{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$; full set of prior hyperparameters: $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, a_0, b_0, \boldsymbol{\eta}_{\text{op}}, \boldsymbol{\eta}_{\text{ft}}, \alpha, \delta)$; optimized variational parameter vector: ϕ^* ; the operator set $\mathbf{O} = \mathbf{O}_u \cup \mathbf{O}_b$; the number of hard symbolic tree samples: H .

Output: A collection of H sampled hard symbolic ensembles and their corresponding posterior mean estimates of the model regression parameters: $\{\{\mathbf{T}(f_j^{(s)})\}_{j=1}^K, \boldsymbol{\beta}_{\text{PM}}^{(s)}, (\sigma^2)_{\text{PM}}^{(s)} : s = 1, \dots, H\}$.

```

for  $s \leftarrow 1$  to  $H$  do
  ▷ Sample one ensemble of  $K$  hard symbolic trees
  for  $j \leftarrow 1$  to  $K$  do
    ▷ (i) sample expansion indicators
     $\hat{e}_{j\zeta}^{(s)} \sim \text{Ber}(\sigma(\ell_{j\zeta}^*))$  for all  $\zeta \in Z_D$ 
    Enforce leaf constraint:  $\hat{e}_{j\zeta}^{(s)} \leftarrow 0$  for all  $\zeta$  at depth  $D$ 
    ▷ (ii) sample operator and feature labels
     $\hat{o}_{j\zeta}^{(s)} \sim \text{Cat}(\text{smax}((\mathbf{a}_{j\zeta}^{\text{op}})^*))$ 
     $\hat{h}_{j\zeta}^{(s)} \sim \text{Cat}(\text{smax}((\mathbf{a}_{j\zeta}^{\text{ft}})^*))$ 
    ▷ (iii) Construct hard symbolic tree skeleton
     $(\hat{\mathbf{S}}_j^{(D)})^{(s)} \leftarrow \{\hat{e}_{j\zeta}^{(s)}, \hat{o}_{j\zeta}^{(s)}, \hat{h}_{j\zeta}^{(s)} : \zeta \in Z_D\}$ 
    ▷ (iv) symbolic trees via deterministic pruning of skeleton
     $\mathbf{T}(f_j^{(s)}) \leftarrow \mathbf{p}\left((\hat{\mathbf{S}}_j^{(D)})^{(s)}\right)$ 
    for  $i \leftarrow 1$  to  $n$  do
      ▷ Evaluate at sample  $\mathbf{x}_i$ 
       $t_{ij}^{(s)} \leftarrow g(\mathbf{x}_i; \mathbf{T}(f_j^{(s)}))$ 
    ▷ Construct expression design matrix
     $t_{i0}^{(s)} \leftarrow 1$  for all  $i = 1, \dots, n$ 
     $\mathbf{T}^{(s)} \leftarrow ((t_{ij}^{(s)}))_{1 \leq i \leq n, 0 \leq j \leq K}$ 
    ▷ Compute posterior means of model regression parameters
     $\boldsymbol{\beta}_{\text{PM}}^{(s)} \leftarrow \mathbb{E}[\boldsymbol{\beta} \mid \sigma^2, \mathbf{T}^{(s)}, \mathbf{y}]$  and  $(\sigma^2)_{\text{PM}}^{(s)} \leftarrow \mathbb{E}[\sigma^2 \mid \mathbf{T}^{(s)}, \mathbf{y}]$ , using (C.2)
  return  $\{\{\mathbf{T}(f_j^{(s)})\}_{j=1}^K, \boldsymbol{\beta}_{\text{PM}}^{(s)}, (\sigma^2)_{\text{PM}}^{(s)} : s = 1, \dots, H\}$ 

```

H DISCUSSION ON THE GAP INDUCED BY ELBO APPROXIMATION

As discussed in Section 4, the exact discrete ELBO in (11) for VaSST is:

$$\mathcal{E}(\phi) = \mathbb{E}_q[\log p(\mathbf{y} \mid \mathbf{T})] - \text{KL}[q(\Theta) \parallel \Pi(\Theta)], \quad (\text{H.1})$$

where $\mathbf{T}$ is the hard symbolic design matrix induced by discrete expansion, operator, and feature assignments. Direct optimization of (H.1), however, is computationally prohibitive because the expectation is over a combinatorial space of symbolic tree structures. Consequently, VaSST replaces the hard symbolic trees by temperature-controlled soft symbolic trees. Specifically, the Bernoulli expansion indicators $e_{j\zeta}$ are relaxed using Binary-Concrete random variables $\tilde{e}_{j\zeta}$, while the categorical operator and feature assignments are relaxed using Gumbel-Softmax random vectors $\tilde{\mathbf{o}}_{j\zeta}$ and $\tilde{\mathbf{h}}_{j\zeta}$, respectively. These relaxations induce a distribution over soft symbolic tree structures, denoted by $\mathcal{P}_{\phi, \tau}^{\text{soft}}$, where $\tau = (\tau_{\text{ex}}, \tau_{\text{op}}, \tau_{\text{ft}})$ is the vector of relaxation temperatures. The corresponding temperature-controlled relaxed ELBO objective is therefore:

$$\mathcal{E}_{\tau}(\phi) = \mathbb{E}_{\mathcal{P}_{\phi, \tau}^{\text{soft}}}[\log p(\mathbf{y} \mid \mathbf{T}_{\text{soft}})] - \text{KL}[q(\Theta) \parallel \Pi(\Theta)], \quad (\text{H.2})$$

where $\mathbf{T}_{\text{soft}}$ is an instance of the soft symbolic design matrix induced by the recursive soft symbolic tree evaluation in (14). However, the expectation in (H.2) is still analytically intractable because $\log p(\mathbf{y} \mid \mathbf{T}_{\text{soft}})$ depends nonlinearly on the recursively evaluated soft symbolic design matrix. Therefore, VaSST uses a Monte Carlo (MC) approximation. Given S MC samples from $\mathcal{P}_{\phi, \tau}^{\text{soft}}$, let $\mathbf{T}_{\text{soft}}^{(s)}$ denote the corresponding soft symbolic design matrix for $s = 1, \dots, S$. The stochastic objective optimized by VaSST is:

$$\hat{\mathcal{E}}(\phi) = \hat{\mathcal{E}}_{\tau, S}(\phi) = \frac{1}{S} \sum_{s=1}^S \log p(\mathbf{y} \mid \mathbf{T}_{\text{soft}}^{(s)}) - \text{KL}[q(\Theta) \parallel \Pi(\Theta)], \quad (\text{H.3})$$

It is important to note that, $\hat{\mathcal{E}}(\phi)$ in (H.3) is an unbiased MC estimator of $\mathcal{E}_{\tau}(\phi)$ in (H.2), not of the exact discrete ELBO $\mathcal{E}(\phi)$ in (H.1). This distinction leads to following decomposition of the induced approximation gap:

$$\mathcal{E}(\phi) - \hat{\mathcal{E}}_{\tau, S}(\phi) = \underbrace{\mathcal{E}(\phi) - \mathcal{E}_{\tau}(\phi)}_{\text{relaxation bias}} + \underbrace{\mathcal{E}_{\tau}(\phi) - \hat{\mathcal{E}}_{\tau, S}(\phi)}_{\text{Monte Carlo (MC) error}}. \quad (\text{H.4})$$

Since the same analytic Kullback-Leibler (KL) penalty is used in both objectives, the relaxation bias arises entirely from replacing the hard tree expected marginal likelihood by the soft tree expected marginal likelihood:

$$\mathcal{E}(\phi) - \mathcal{E}_{\tau}(\phi) = \mathbb{E}_q[\log p(\mathbf{y} \mid \mathbf{T})] - \mathbb{E}_{\mathcal{P}_{\phi, \tau}^{\text{soft}}}[\log p(\mathbf{y} \mid \mathbf{T}_{\text{soft}})], \quad (\text{H.5})$$

which is non-zero in general since soft symbolic evaluation and marginal likelihood evaluation do not commute. This non-commutativity is especially important in VaSST, since $\mathbf{T}_{\text{soft}}$ is produced by recursive mixtures of nonlinear symbolic operators, while $\log p(\mathbf{y} \mid \mathbf{T}_{\text{soft}})$ depends on $\mathbf{T}_{\text{soft}}$ through the collapsed Normal Inverse-Gamma marginal likelihood in (7). Hence, at finite temperature, the relaxed objective targets a smoothed symbolic model that is close in spirit, but not identical to, the original discrete symbolic tree model.

The second term in (H.4) is standard stochastic approximation error. For fixed (ϕ, τ) , it has mean zero. If:

$$V_{\phi, \tau} = \text{Var}_{\mathcal{P}_{\phi, \tau}^{\text{soft}}}[\log p(\mathbf{y} \mid \mathbf{T}_{\text{soft}})] < \infty,$$

then $\text{Var}_{\mathcal{P}_{\phi, \tau}^{\text{soft}}}[\hat{\mathcal{E}}_{\tau, S}(\phi)] = S^{-1}V_{\phi, \tau}$, because the KL term is deterministic for fixed ϕ . Thus, increasing S reduces the MC component of the gap at the usual $S^{-1/2}$ rate, whereas it does not remove the relaxation bias. The relaxation bias is controlled instead through the annealing schedule: as the temperatures decrease, the Binary-Concrete and Gumbel-Softmax variables become increasingly concentrated near Bernoulli and Categorical draws. Under numerically stable symbolic evaluation, this progressively sharpens the soft symbolic trees toward hard symbolic structures.

In light of these observations, this approximation is best interpreted as an algorithmic device for scalable posterior learning over symbolic structures. Direct optimization of the discrete objective in (H.1) would require either score-function estimators over symbolic structures or explicit combinatorial search over an exponentially large expression space. Such approaches may be high-variance or computationally prohibitive in the SR setting. The relaxed objective in (H.2), together with the MC

approximation in (H.3), provides a differentiable reparameterized surrogate with low-variance gradients, enabling black-box gradient-based optimization.

Finally, the distinction between the relaxed optimization objective and the original discrete symbolic model is also reflected in the post-optimization inference step of VaSST, where the optimized soft representations are not reported as the final outcomes. Instead, post-optimization, hard symbolic ensembles are sampled from the learned variational distributions and then evaluated using the discrete symbolic design matrix $\mathbf{T}$. These hard symbolic ensembles are ranked by the posterior-aware LMPSE criterion in (16), which combines the collapsed marginal likelihood under the hard design matrix with the symbolic tree prior. Thus, the relaxation is used to make posterior exploration computationally tractable, while the final reported symbolic summaries are obtained by returning to the hard symbolic tree space.

I CONFIGURATIONS OF VaSST AND COMPETING METHODS

I.1 OPERATOR SET

Following are the operator sets tabulated in Table I.1, used across all experiments in Sections 5.1 and 5.2. For budget equalization, we try to maintain a common operator specification across VaSST and competing SR methods.

Table I.1: Operator set **O** used across all experiments in Sections 5.1 and 5.2.

Method	(17)	(18)	(CL)	(CPE)	(FCE)	(FTC)	(ADA)
VaSST	$\{+, -, \times, /, ^2\}$	$\{\times, \sin, \cos\}$	$\{\times, /, ^2\}$	$\{+, -, \times, /\}$	$\{+, \times, /, \sin\}$	$\{+, -, \times, /, \sqrt{}\}$	$\{+, \times, \cos\}$
operon	$\{+, -, \times, /, ^2\}$	$\{\times, \sin, \cos\}$	$\{\times, /, ^2\}$	$\{+, -, \times, /\}$	$\{+, \times, /, \sin\}$	$\{+, -, \times, /, \sqrt{}\}$	$\{+, \times, \cos\}$
BMS	$\{+, \times, /, \text{neg}, ^2\}$	$\{\times, \sin, \cos\}$	$\{\times, /, ^2\}$	$\{+, \text{neg}, \times, /\}$	$\{+, \times, /, \sin\}$	$\{+, \text{neg}, \times, /, \sqrt{}\}$	$\{+, \times, \cos\}$
BSR	$\{+, -, \times, \text{inv}, ^2\}$	$\{\times, \sin, \cos\}$	$\{\times, \text{inv}, ^2\}$	$\{+, -, \times, \text{inv}\}$	$\{+, \times, \text{inv}, \sin\}$	$\{+, -, \times, \text{inv}, \sqrt{}\}$	$\{+, \times, \cos\}$
QLattice	$\{+, \times, ^2, \text{lin}, \text{inv}\}$	default	default	default	default	default	default
PySR	$\{+, -, \times, /, ^2\}$	$\{\times, \sin, \cos\}$	$\{\times, /, ^2\}$	$\{+, -, \times, /\}$	$\{+, \times, /, \sin\}$	$\{+, -, \times, /, \sqrt{}\}$	$\{+, \times, \cos\}$
gplearn	$\{+, -, \times, /, ^2\}$	$\{\times, \sin, \cos\}$	$\{\times, /, ^2\}$	$\{+, -, \times, /\}$	$\{+, \times, /, \sin\}$	$\{+, -, \times, /, \sqrt{}\}$	$\{+, \times, \cos\}$
DEAP	$\{+, -, \times, /, ^2\}$	$\{\times, \sin, \cos\}$	$\{\times, /, ^2\}$	$\{+, -, \times, /\}$	$\{+, \times, /, \sin\}$	$\{+, -, \times, /, \sqrt{}\}$	$\{+, \times, \cos\}$
DSR	$\{+, -, \times, /, \text{n2}\}$	$\{\times, \sin, \cos\}$	$\{\times, /, \text{n2}\}$	$\{+, -, \times, /\}$	$\{+, \times, /, \sin\}$	$\{+, -, \times, /, \sqrt{}\}$	$\{+, \times, \cos\}$

Note: operon (<https://github.com/heal-research/pyoperon>) and DEAP (<https://github.com/DEAP/deap>) additionally uses variables and ephemeral constants. The operators `neg` denotes the unary negation operator, $\text{neg}(z) = -z$; `inv` denotes the unary reciprocal operator, $\text{inv}(z) = \frac{1}{z}$; `lin` denotes the linear operator, $\text{lin}(z) = az + b$; and `n2` denotes the unary square operator, $\text{n2}(z) = z^2$. For QLattice (<https://docs.abzu.ai/>), `default` indicates the use of default operator set.

I.2 EXPERIMENTAL SETTINGS OF VaSST

Across all experiments in Sections 5.1 and 5.2, VaSST was implemented using the common experimental settings tabulated in Table I.2.

Table I.2: Common experimental settings of VaSST used in Sections 5.1 and 5.2.

Component	Hyperparameter	Value
Symbolic ensemble	Number of trees, K	3
	Maximum tree depth, D	3
Symbolic tree prior	Split probability parameters, (α, δ)	(0.95, 2.00)
	Operator Dirichlet parameter, η_{op}	$\mathbf{1}_{ \mathcal{O} }$
	Feature Dirichlet parameter, η_{ft}	$\mathbf{1}_p$
Outer regression coefficients	Prior mean, μ_0	$\mathbf{0}_{K+1}$
	Prior covariance, Σ_0	$10\mathbf{I}_{K+1}$
Model noise variance	Inverse-Gamma shape, a_0	2
	Inverse-Gamma scale, b_0	2
Variational optimization	Optimizer	AdamW
	Learning rate, γ	5×10^{-5}
	AdamW momentum parameters	(0.90, 0.99)
	AdamW weight decay	0
	Number of optimization steps, T	2000
	Monte Carlo samples per ELBO step, S	8
	Gradient clipping threshold, c	1.0
Temperature annealing	Initial temperature, τ_{start}	1.0
	Final temperature, τ_{end}	0.5
	Annealing horizon, T_{e}	1500 steps
Numerical stabilization	Soft symbolic tree output clipping threshold	<code>value_clip = 200</code>
	Hyperbolic tangent clipping	<code>use_tanh_clip = True</code>
	Variational logit clipping threshold	<code>logits_clip = 10.0</code>
Hard symbolic ensembles	Number of sampled hard symbolic ensembles, H	2000
	Ranking criterion	Using LMPSE in (16)
	Number of top ranked symbolic expressions reported, r	10

Continued on next page

Table I.2 continued from previous page.

Component	Hyperparameter	Value
Evaluation	Training split	90%
	Held-out test split for out-of-sample RMSE	10%

Note: The Binary-Concrete and Gumbel-Softmax temperature parameters τ_{ex} , τ_{op} , and τ_{ft} , respectively, are linearly annealed from $\tau_{\text{start}} = 1.0$ to $\tau_{\text{end}} = 0.5$ over $T_r = 1500$ steps. Out-of-sample RMSE is computed on a 10% held-out test set with the LMPSE selected symbolic model.

I.3 EXPERIMENTAL SETTINGS OF OPERON, GPLEARN, DEAP, AND PYSR

operon. We run operon [Burlacu et al., 2020] using the SymbolicRegressor implementation from PyOperon (<https://github.com/heal-research/pyoperon>). For each simulation experiments in (17)-(18) from Section 5.1 and the Feynman equations in (CL)-(ADA) from Section 5.2, the operator sets used are reported in Table I.1 and operon was implemented with `generations = 2000`, `population_size = 1000`, `max_length = 20`, and `n_threads = 1`. The number of generations (`generations = 2000`) specifies the total number of evolutionary (e.g., mutation, recombination, and selection) iterations used to update the population of candidate symbolic expressions, while the population size (`population_size = 1000`) determines the number of expressions maintained at each iteration. The maximum length (`max_length = 20`) parameter bounds the size of the symbolic expression tree. Finally, `n_threads = 1` enforces a single-threaded execution. After fitting, the best symbolic expression returned by operon is extracted from the fitted symbolic model with respect to the in-sample RMSE.

gplearn. We employ the gplearn [Stephens, 2016] SR framework (<https://github.com/trevorstephens/gplearn>) implemented in Python and outline its experimental configuration for learning the symbolic expressions in (17)-(18) from Section 5.1 and the Feynman equations in (CL)-(ADA) from Section 5.2. The SymbolicRegressor is configured with a population size of 2000 and evolved for 20 generations using tournament selection with tournament size 20. The operator sets used are reported in Table I.1. The fitness metric is set to RMSE, and a parsimony coefficient of 10^{-4} is used to penalize overly complex expressions.

Distributed Evolutionary algorithms in Python (DEAP). We implement genetic programming for SR using the DEAP [Fortin et al., 2012] framework (<https://github.com/DEAP/deap>), a general-purpose evolutionary computation library in Python. The experimental configurations of DEAP used for learning the symbolic expressions in (17)-(18) from Section 5.1 and the Feynman equations in (CL)-(ADA) from Section 5.2 are as follows. Symbolic expressions are evolved using a $(\mu + \lambda)$ evolutionary strategy with $\mu = 200$ parent individuals and $\lambda = 200$ offspring per generation, for 20 generations. The crossover and mutation probabilities are set to 0.5 and 0.2, respectively. The operator sets used are reported in Table I.1 along with ephemeral random constants sampled from $\text{Unif}(-5, 5)$. Fitness is measured using RMSE, and the best individual is selected from the Hall-of-Fame archive after evolution.

PySR. We use the PySRRegressor implementation (<https://github.com/MilesCranmer/pysr>) for PySR [Cranmer, 2023], which provides a Python interface to the SymbolicRegression.jl search engine. PySR searches over symbolic expressions that optimize a user-specified objective. For learning the expressions in (17)-(18) from Section 5.1 and the Feynman equations in (CL)-(ADA) from Section 5.2, we configure the operator sets as in Table I.1 and set `niterations = 200`, `populations = 10`, `population_size = 50`, and `max_size = 20`. We use squared-error loss and `model_selection = "best"`, where the final symbolic expression being extracted using `get_best()` corresponds to the best accuracy-complexity trade-off among the discovered Pareto-optimal expressions stored in `equations_`.

I.4 EXPERIMENTAL SETTINGS OF BMS AND BSR

Bayesian Machine Scientist (BMS). For each simulation experiments in (17)-(18) from Section 5.1 and the Feynman equations in (CL)-(ADA) from Section 5.2, we use the AutoRA implementation (<https://github.com/AutoResearch/autora-theorist-bms>) of BMS [Guimerà et al., 2020] with the operator sets reported in Table I.1 and uniform prior operator weights. Model discovery is performed using parallel-tempered Markov chain Monte Carlo (MCMC) over the symbolic grammar space for 2000 epochs. The final fitted symbolic expression returned by BMS is the one having the minimum description length during the MCMC run.

Bayesian Symbolic Regression (BSR). For each simulation experiments in (17)-(18) from Section 5.1 and the Feynman equations in (CL)-(ADA) from Section 5.2, we use the AutoRA implementation (<https://github.com/AutoResearch/autora-theorist-bsr>) of BSR [Jin et al., 2020] with the number of trees aligned to the ensemble size used in VaSST; see Table I.2. Uniform prior operator weights were maintained as in BMS. The number of MCMC iterations is set to 2000 and the operator sets used are reported in Table I.1. To ensure fairness in comparison, the split probability configuration for node expansion is matched to that used in VaSST, as specified in Table I.2. The outer regression coefficients are estimated using ordinary least squares and the final symbolic expression corresponds to the last fitted ensemble by the MCMC.

I.5 EXPERIMENTAL SETTINGS OF QLATTICE AND DSR

QLattice. We implement QLattice [Broløs et al., 2021] via the Feyn Python (<https://docs.abzu.ai/>), a supervised machine learning framework designed to perform SR. The default configuration of QLattice with 25 training epochs were used for learning the symbolic expressions in (17)-(18) from Section 5.1 and the Feynman equations in (CL)-(ADA) from Section 5.2. The operator sets, referred to as *interactions* in the QLattice-Feyn interface, are configured as in Table I.1.

Deep Symbolic Regression (DSR). We use the DeepSymbolicRegressor implementation (<https://github.com/dso-org/deep-symbolic-optimization>) of DSR [Petersen et al., 2021] for learning the symbolic expressions in (17)-(18) from Section 5.1 and the Feynman equations in (CL)-(ADA) from Section 5.2. DSR parameterizes a policy over symbolic expressions using an autoregressive recurrent neural network (RNN), which sequentially samples symbolic programs from a prescribed function set. The operator sets are configured as in Table I.1. The regression task is run with protected operators enabled. We maintain `n_samples = 200000`, `batch_size = 500`, and `max_length = 20` across all experiments in Sections 5.1 and 5.2. Here, `n_samples = 200000` denotes the total symbolic program sampling budget, `batch_size = 500` is the number of candidate programs sampled per RNN policy gradient update, and `max_length = 20` bounds the maximum length of each sampled symbolic expression. These settings correspond to approximately 400 RNN policy gradient updates, unless early stopping occurs. During training, DSR retains the best-reward symbolic program encountered over the search. The final recovered expression is extracted from the fitted estimator through the attribute `program_`.

J ADDITIONAL RESULTS FOR SIMULATION EXPERIMENTS

J.1 SYMBOLIC EXPRESSIONS LEARNED BY VaSST AND COMPETING METHODS

For $\mathbf{x}_0^2 - \mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2^2$ in Section 5.1.

Table J.1: Expressions learned in a single run of VaSST and competing methods for learning (17) under the noiseless setting.

Method	Symbolic Expression Learned	RMSE
VaSST	$0.9987\mathbf{x}_0^2 - 1.0018\mathbf{x}_1 + 0.5125\mathbf{x}_2^2 - 0.0031\mathbf{x}_2 - 0.2283$	0.003526
operon	$0.9451\mathbf{x}_0^2 + 0.0881\mathbf{x}_0 - 0.9999\mathbf{x}_1 + 0.4997\mathbf{x}_2^2 - 0.0680 - \frac{0.2716\mathbf{x}_2^2}{\Delta}$	0.000636
BMS	$\mathbf{x}_0^2 - \mathbf{x}_1 + 0.5\mathbf{x}_2^2$	0.000000
BSR	$5.4 \times 10^{-6} + 6.6 \times 10^{-6}\mathbf{x}_0^{-1}\mathbf{x}_1^{-2} + 1.7 \times 10^{-4}\mathbf{x}_2^2 + 2.0 \times 10^{-7}(\mathbf{x}_2^2 + \mathbf{x}_0^{-2}\mathbf{x}_2^{-1})$	13.084523
QLattice	$0.9987\mathbf{x}_0^2 - 1.0016\mathbf{x}_1 + 0.5028\mathbf{x}_2^2 - 0.0285\mathbf{x}_2 + 0.0733$	0.002887
PySR	$\mathbf{x}_0^2 - \mathbf{x}_1 + 0.5\mathbf{x}_2^2$	0.000000
gplearn	$0.7890\mathbf{x}_0^4 + 0.4260\mathbf{x}_2^2$	0.394263
DEAP	$1.1724(\mathbf{x}_0 + 1.9821)(\mathbf{x}_2 - 4.7760) + 2\mathbf{x}_2$	0.375525
DSR	$-\mathbf{x}_1 + 0.5\mathbf{x}_2^2$	0.453799

Note: For operon, $\Delta = 3.2569\mathbf{x}_0(20.4510\mathbf{x}_0^4 - 2.0478\mathbf{x}_2^2) + 1.0080\mathbf{x}_1^2 + 0.3566\mathbf{x}_1\mathbf{x}_2 - 4.0955\mathbf{x}_2^2 - 1.0550\mathbf{x}_2$. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

Table J.2: Expressions learned in a single run of VaSST and competing methods for learning (17) under $\sigma = 0.1$.

Method	Symbolic Expression Learned	RMSE
VaSST	$1.0439\mathbf{x}_0^2 - 1.0105\mathbf{x}_1 + 0.5219\mathbf{x}_2^2 - 0.0289\mathbf{x}_0 + 0.0289\mathbf{x}_2 - 0.5386$	0.101862
operon	$0.0001\mathbf{x}_0^{16} + 0.0197\mathbf{x}_0^9 + 1.0230\mathbf{x}_0^2 - 1.1390\mathbf{x}_1 + 0.5120\mathbf{x}_2^2 + 0.3510 + \frac{0.0385\mathbf{x}_2^2}{\Delta}$	0.101284
BMS	$\mathbf{x}_0^2 - \mathbf{x}_1 + a_0^{-1}\mathbf{x}_2^2$	0.101690
BSR	$-26.6898 - 0.4150\mathbf{x}_0^2 - 0.4150\mathbf{x}_1^2 + 4.4860\mathbf{x}_1 + 5.3160\mathbf{x}_2 + 68.8933\mathbf{x}_1^{-4}$	0.490357
QLattice	$0.0756\mathbf{x}_0^2\mathbf{x}_2 + 0.6128\mathbf{x}_0^2 - 0.0029\mathbf{x}_0\mathbf{x}_2 - 0.0232\mathbf{x}_0 - 0.0747\mathbf{x}_1\mathbf{x}_2 - 0.6059\mathbf{x}_1 + 0.3953\mathbf{x}_2^2 + 1.3280\mathbf{x}_2 - 4.0760$	0.102598
PySR	$\mathbf{x}_0^2 - \mathbf{x}_1 + 0.4999\mathbf{x}_2^2$	0.101690
gplearn	$0.7890\mathbf{x}_0^4 + 0.4264\mathbf{x}_2^2$	0.411029
DEAP	$1.1724\mathbf{x}_0\mathbf{x}_2 - 5.5995\mathbf{x}_0 + 4.3239\mathbf{x}_2 - 11.0990$	0.391945
DSR	$\frac{\mathbf{x}_2^3}{\mathbf{x}_1 + 2\mathbf{x}_2 - 1}$	0.489848

Note: For operon, $\Delta = 0.3842\mathbf{x}_0 - 1.1130\mathbf{x}_1 - \frac{2.5270\mathbf{x}_2}{-0.3346\mathbf{x}_2^2 + 5.9040\mathbf{x}_2 + 0.0313\mathbf{x}_1/\mathbf{x}_0}$. a_0 is a constant learned by BMS. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

Table J.3: Expressions learned in a single run of VaSST and competing methods for learning (17) under $\sigma = 0.2$.

Method	Symbolic Expression Learned	RMSE
VaSST	$1.0425\mathbf{x}_0^2 - 1.0203\mathbf{x}_1 + 0.5212\mathbf{x}_2^2 - 0.0378\mathbf{x}_0 + 0.0378\mathbf{x}_2 - 0.5396$	0.203331
operon	$1.0125\mathbf{x}_0^2 - 0.9759\mathbf{x}_1 + 0.4987\mathbf{x}_2^2 - 0.0454 + \frac{-0.8539\mathbf{x}_1 + 0.4504\mathbf{x}_2}{\Delta}$	0.202378
BMS	$\mathbf{x}_0^2 - \mathbf{x}_1 + a_0\mathbf{x}_2^2$	0.203380
BSR	$-0.4211\mathbf{x}_0^2 - 0.4211\mathbf{x}_1^2 + 4.4754\mathbf{x}_1 + 5.3177\mathbf{x}_2 - 26.6139 + 68.2127\mathbf{x}_1^{-4}$	0.520460
QLattice	$1.0382\mathbf{x}_0^2 - 0.0673\mathbf{x}_0 - 1.0220\mathbf{x}_1 + 5.5120\mathbf{x}_2 - 15.0740$	0.206777
PySR	$\mathbf{x}_0 - \mathbf{x}_1 + 0.4944\mathbf{x}_2^2$	0.218974
gplearn	$0.7890\mathbf{x}_0^4 + 0.4264\mathbf{x}_2^2$	0.448464
DEAP	$1.1724\mathbf{x}_0\mathbf{x}_2 - 5.5995\mathbf{x}_0 + 4.3239\mathbf{x}_2 - 11.0988$	0.432334
DSR	$\frac{0.5\mathbf{x}_2^3}{\mathbf{x}_2 + 1}$	0.504085

Note: For operon, $\Delta = 54.196\mathbf{x}_0^2 + 1.2928\mathbf{x}_1^2 - 10.808\mathbf{x}_1 + 1.4352\mathbf{x}_2^2 - 2.8313\mathbf{x}_2 - 0.4920$. a_0 is a constant learned by BMS. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

For $6\sin(\mathbf{x}_0)\cos(\mathbf{x}_1)$ in Section 5.1.

Table J.4: Expressions recovered in a single run of **VaSST** and competing methods for learning (18) under the noiseless setting.

Method	Symbolic Expression Learned	RMSE
VaSST	$5.6818\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) + 0.0787\mathbf{x}_0\mathbf{x}_1\cos(\mathbf{x}_1) - 0.0281\cos(\mathbf{x}_1)\cos(\mathbf{x}_0\mathbf{x}_1) - 0.0397$	0.009577
operon	$-0.002 + 6.527\cos(-0.982\mathbf{x}_1)\sin(1.026\mathbf{x}_0)\cos(A(\mathbf{x})B(\mathbf{x}))$	0.003086
BMS	$6\sin(\mathbf{x}_0)\cos(\mathbf{x}_1)$	0.000000
BSR	$-103.575 + 1.9976\sin(\mathbf{x}_1) + 101.883\cos(\mathbf{x}_0) + 43.7129\mathbf{x}_0^2$	0.340724
QLattice	$-2.011(3.114\mathbf{x}_0 + 0.002)\exp\{-0.185(0.0482 - \mathbf{x}_0)^2 - 2.891(0.289\mathbf{x}_1 + 0.832L(\mathbf{x}) - 1)^2 - 0.0004\}$	0.004568
PySR	$6\sin(\mathbf{x}_0)\cos(\mathbf{x}_1)$	0.000000
gplearn	$-0.6513\mathbf{x}_0\mathbf{x}_1^2$	0.172140
DEAP	$-1.9881\mathbf{x}_1\sin\{\sin(\mathbf{x}_0)\}$	0.279846
DSR	$\mathbf{x}_0\mathbf{x}_1^2\cos(\mathbf{x}_0)\cos(\mathbf{x}_1)$	0.548577

Note: For operon, $A(\mathbf{x}) = \cos(\cos[\cos(-1.142\mathbf{x}_1)\cos(-1.142\mathbf{x}_1)])\cos(\cos[1.686\mathbf{x}_0\cos(0.258\mathbf{x}_1)])\cos(\cos(-1.134\mathbf{x}_1))$ and $B(\mathbf{x}) = \cos[-0.295\cos(-3.325\mathbf{x}_1)\sin(1.351\mathbf{x}_0)]\sin(-0.159\mathbf{x}_1)$. For QLattice, $L(\mathbf{x}) = -0.002(-0.201\mathbf{x}_0 - 1)^2 - 1.832(0.349\mathbf{x}_1 - 1)^2$. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

Table J.5: Expressions recovered in a single run of **VaSST** and competing methods for learning (18) under $\sigma = 0.1$.

Method	Symbolic Expression Learned	RMSE
VaSST	$5.9931\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0023\mathbf{x}_1 - 0.0009\mathbf{x}_1\cos\{\cos(\mathbf{x}_1)\} + 0.0023$	0.099728
operon	$-0.001 + 1.949\cos\{0.318\mathbf{x}_0\sin(-0.894\mathbf{x}_0)\}\{1.353\mathbf{x}_1\sin(0.917\mathbf{x}_0)\cos(-1.127\mathbf{x}_1)\}A(\mathbf{x})B(\mathbf{x})$	0.099540
BMS	$a_0\sin(\mathbf{x}_0)\cos(\mathbf{x}_1)$	0.099723
BSR	$-9.5947 + 2.6779\cos(\mathbf{x}_1) + 10.3204\cos\{\mathbf{x}_0\sin(\mathbf{x}_0)\}$	0.652872
QLattice	$-3.647(1.665\mathbf{x}_0 + 0.003)\exp\{-0.235(0.182 - \mathbf{x}_0)^2 - 2\exp\{-1.544(1 - 0.866\mathbf{x}_1)^2\} + 0.001$	0.100564
PySR	$6.0049\sin(\mathbf{x}_0)\cos(\mathbf{x}_1)$	0.099723
gplearn	$-0.6513\mathbf{x}_0\mathbf{x}_1^2$	0.202208
DEAP	$-1.9881\mathbf{x}_1\sin\{\sin(\mathbf{x}_0)\}$	0.296343
DSR	$\mathbf{x}_0\mathbf{x}_1^2\cos(\mathbf{x}_0)\cos(\mathbf{x}_1)$	0.556807

Note: For operon, $A(\mathbf{x}) = \cos[\sin\{\cos(\sin(1.353\mathbf{x}_1))1.353\mathbf{x}_1\}\sin(1.352\mathbf{x}_1)\sin(-2.031\mathbf{x}_1)]$ and $B(\mathbf{x}) = \sin[\cos\{-0.474\sin[\sin\{\sin(1.353\mathbf{x}_1)\}]\sin(-2.001\mathbf{x}_1)\}]$. a_0 is a constant learned by BMS. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

Table J.6: Expressions recovered in a single run of **VaSST** and competing methods for learning (18) under $\sigma = 0.2$.

Method	Symbolic Expression Learned	RMSE
VaSST	$5.9933\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0031\mathbf{x}_1 - 0.0039\mathbf{x}_1\cos\{\cos(\mathbf{x}_1)\} + 0.0074$	0.199438
operon	$88.541A(\mathbf{x})B(\mathbf{x})C(\mathbf{x})$	0.199028
BMS	$a_0\sin(\mathbf{x}_0)\cos(\mathbf{x}_1)$	0.199446
BSR	$-9.5709 + 2.6796\cos(\mathbf{x}_1) + 10.2934\cos\{\mathbf{x}_0\sin(\mathbf{x}_0)\}$	0.679095
QLattice	$-8.4212\tanh(0.8270\mathbf{x}_0 - 0.0052)\tanh(1.0045\mathbf{x}_1 - 1.6330) + 0.0021$	0.201085
PySR	$5.4158\mathbf{x}_0\cos(\mathbf{x}_1)$	0.235483
gplearn	$-0.7930\mathbf{x}_0\mathbf{x}_1^2\cos(0.7930\mathbf{x}_0)$	0.262413
DEAP	$-1.9881\mathbf{x}_1\sin\{\sin(\mathbf{x}_0)\}$	0.342378
DSR	$\mathbf{x}_0\mathbf{x}_1^2\cos(\mathbf{x}_0)\cos(\mathbf{x}_1)$	0.582262

Note: For operon, $A(\mathbf{x}) = \cos[\cos\{\cos(1.134\mathbf{x}_1)\cos[\cos(1.134\mathbf{x}_1)]\}]\cos\{0.2027\mathbf{x}_0^2\sin^2(0.863\mathbf{x}_1)\}$, $B(\mathbf{x}) = \cos\{\cos(0.079\mathbf{x}_1)\}\cos[\cos\{\cos(0.079\mathbf{x}_1)\}\cos(0.079\mathbf{x}_1)]$, and $C(\mathbf{x}) = 0.079\mathbf{x}_1\sin(0.917\mathbf{x}_0)\cos\{0.3141\mathbf{x}_0^2\sin(0.917\mathbf{x}_0)\}\cos(1.134\mathbf{x}_1)$. a_0 is a constant learned by BMS. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

J.2 OUT-OF-SAMPLE RMSE FOR VaSST AND COMPETING METHODS

Table J.7: Out-of-sample RMSEs (computed on a 10% held-out test set) of VaSST and competing methods over 10 repetitions (mean $\pm$ standard deviation) for learning $\mathbf{x}_0^2 - \mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2^2$ in (17) of Section 5.1 across all noise settings.

Noiseless		$\sigma = 0.1$		$\sigma = 0.2$	
Method	RMSE, mean $\pm$ sd	Method	RMSE, mean $\pm$ sd	Method	RMSE, mean $\pm$ sd
VaSST	0.00412 $\pm$ 0.00133	VaSST	0.10036 $\pm$ 0.00158	VaSST	0.20061 $\pm$ 0.00309
operon	0.00644 $\pm$ 0.00394	operon	0.10040 $\pm$ 0.00152	operon	0.20025 $\pm$ 0.00309
BMS	$7.74 \times 10^{-17} \pm 2.45 \times 10^{-16}$	BMS	0.10032 $\pm$ 0.00152	BMS	0.20063 $\pm$ 0.00305
BSR	1.96367 $\pm$ 3.94562	BSR	0.71729 $\pm$ 0.54286	BSR	0.75153 $\pm$ 0.52180
QLattice	0.00975 $\pm$ 0.00684	QLattice	0.10219 $\pm$ 0.00199	QLattice	0.20347 $\pm$ 0.00319
PySR	0.00000 $\pm$ 0.00000	PySR	0.10032 $\pm$ 0.00152	PySR	0.21481 $\pm$ 0.00351
gplearn	0.21830 $\pm$ 0.12335	gplearn	0.21616 $\pm$ 0.09241	gplearn	0.28095 $\pm$ 0.08016
DEAP	0.69742 $\pm$ 0.24570	DEAP	0.72482 $\pm$ 0.25997	DEAP	0.73682 $\pm$ 0.24150
DSR	0.50942 $\pm$ 0.05589	DSR	0.49501 $\pm$ 0.07044	DSR	0.49416 $\pm$ 0.04973

Table J.8: Out-of-sample RMSEs (computed on a 10% held-out test set) of VaSST and competing methods over 10 repetitions (mean $\pm$ standard deviation) for learning $6\sin(\mathbf{x}_0)\cos(\mathbf{x}_1)$ in (18) of Section 5.1 across all noise settings.

Noiseless		$\sigma = 0.1$		$\sigma = 0.2$	
Method	RMSE, mean $\pm$ sd	Method	RMSE, mean $\pm$ sd	Method	RMSE, mean $\pm$ sd
VaSST	0.01306 $\pm$ 0.00767	VaSST	0.10054 $\pm$ 0.00149	VaSST	0.19951 $\pm$ 0.00170
operon	0.00244 $\pm$ 0.00241	operon	0.09904 $\pm$ 0.00095	operon	0.19828 $\pm$ 0.00189
BMS	0.00000 $\pm$ 0.00000	BMS	0.09941 $\pm$ 0.00091	BMS	0.19881 $\pm$ 0.00181
BSR	0.28186 $\pm$ 0.13300	BSR	0.33652 $\pm$ 0.15671	BSR	0.38499 $\pm$ 0.13881
QLattice	0.00400 $\pm$ 0.00127	QLattice	0.09994 $\pm$ 0.00123	QLattice	0.19971 $\pm$ 0.00173
PySR	0.00000 $\pm$ 0.00000	PySR	0.09941 $\pm$ 0.00091	PySR	0.22836 $\pm$ 0.00429
gplearn	0.11964 $\pm$ 0.03772	gplearn	0.15850 $\pm$ 0.03238	gplearn	0.23735 $\pm$ 0.02038
DEAP	0.36454 $\pm$ 0.14455	DEAP	0.38290 $\pm$ 0.13653	DEAP	0.42092 $\pm$ 0.12471
DSR	0.55193 $\pm$ 0.00870	DSR	0.56119 $\pm$ 0.00939	DSR	0.58739 $\pm$ 0.01000

J.3 TOP RANKED SYMBOLIC EXPRESSIONS LEARNED BY VaSST WITH RESPECT TO LMPSE

For $\mathbf{x}_0^2 - \mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2^2$ in Section 5.1.

Table J.9: Top $r = 10$ ranked symbolic expressions recovered by a single run of VaSST under the noiseless setting for learning (17). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	LMPSE	RMSE
1	$0.9987\mathbf{x}_0^2 - 1.0018\mathbf{x}_1 + 0.5125\mathbf{x}_2^2 - 0.0031\mathbf{x}_2 - 0.2283$	3296.45	0.003526
2	$1.0009\mathbf{x}_0^2 - 1.0009\mathbf{x}_1 + 0.5331\mathbf{x}_2^2 - 0.2247\mathbf{x}_2 + 0.3650$	3285.09	0.002155
3	$0.9996\mathbf{x}_0^2 - 0.9996\mathbf{x}_1 + 0.5122\mathbf{x}_2^2 - 0.0001\mathbf{x}_1^8 - 0.2236$	3276.92	0.003551
4	$0.9993\mathbf{x}_0^2 - 1.0001\mathbf{x}_1 + 0.5122\mathbf{x}_2^2 - 0.2409$	3272.47	0.003524
5	$0.9993\mathbf{x}_0^2 - 1.0014\mathbf{x}_1 + 0.5122\mathbf{x}_2^2 - 0.2384$	3270.59	0.003529
6	$0.9997\mathbf{x}_0^2 - 0.9987\mathbf{x}_1 + 0.5737\mathbf{x}_2^2 - 0.5737\mathbf{x}_2 + 1.1346$	3268.11	0.002609
7	$0.9991\mathbf{x}_0^2 - 1.0388\mathbf{x}_1 + 0.5194\mathbf{x}_2^2 + 0.1120\mathbf{x}_2 - 0.8127$	3239.00	0.010701
8	$1.0013\mathbf{x}_0^2 + 0.0114\mathbf{x}_0 - 1.0013\mathbf{x}_1 + 0.5426\mathbf{x}_2^2 - 0.8243$	3205.50	0.011400
9	$0.8460\mathbf{x}_0^2 + 0.1669\mathbf{x}_0 - 0.1669\mathbf{x}_1^2 - 0.1669\mathbf{x}_1 + 0.5473\mathbf{x}_2^2 - 0.3785\mathbf{x}_2 - 0.2803$	3184.82	0.017184
10	$-0.0719\mathbf{x}_0^2 + 0.1438\mathbf{x}_0^2\mathbf{x}_1 + 0.1438\mathbf{x}_0^2\mathbf{x}_2 - 0.0018\mathbf{x}_0 - 0.0719\mathbf{x}_1^2 - 0.1438\mathbf{x}_1\mathbf{x}_2 + 0.0006\mathbf{x}_1 + 0.5476\mathbf{x}_2^2 - 0.0006\mathbf{x}_2 - 1.3782$	3153.72	0.014262

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

Table J.10: Top $r = 10$ ranked symbolic expressions recovered by a single run of VaSST under $\sigma = 0.1$ for learning (17). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	LMPSE	RMSE
1	$1.0439\mathbf{x}_0^2 - 1.0105\mathbf{x}_1 + 0.5219\mathbf{x}_2^2 - 0.0289\mathbf{x}_0 + 0.0289\mathbf{x}_2 - 0.5386$	1506.54	0.101862
2	$0.9914\mathbf{x}_0^2 - 1.0107\mathbf{x}_1 + 0.5129\mathbf{x}_2^2 - 0.2253$	1503.94	0.101688
3	$0.9912\mathbf{x}_0^2 - 1.0099\mathbf{x}_1 + 0.5124\mathbf{x}_2^2 - 0.2196$	1502.74	0.101685
4	$1.0124\mathbf{x}_0^2 - 0.0345\mathbf{x}_0 - 1.0124\mathbf{x}_1 + 0.5238\mathbf{x}_2^2 - 0.4311$	1498.05	0.101826
5	$1.0804\mathbf{x}_0^2 - 0.0941\mathbf{x}_0 - 0.0941\mathbf{x}_1^2 - 0.5402\mathbf{x}_1 + 0.5123\mathbf{x}_2^2 - 0.7809$	1495.14	0.101806
6	$0.9813\mathbf{x}_0^2 - 1.0114\mathbf{x}_1 + 0.5361\mathbf{x}_2^2 - 0.0909\mathbf{x}_2 - 0.2825$	1490.35	0.101942
7	$0.9907\mathbf{x}_0^2 - 1.0088\mathbf{x}_1 + 0.4684\mathbf{x}_2^2 + 0.5404\mathbf{x}_2 - 1.8103$	1489.77	0.102035
8	$0.9912\mathbf{x}_0^2 - 1.0099\mathbf{x}_1 + 0.5129\mathbf{x}_2^2 - 0.2270$	1489.37	0.101688
9	$1.0113\mathbf{x}_0^2 + 0.0224\mathbf{x}_0 - 1.0113\mathbf{x}_1 + 0.5257\mathbf{x}_2^2 - 0.4911$	1486.86	0.101895
10	$0.9915\mathbf{x}_0^2 + 0.0147\mathbf{x}_1^2 - 1.0913\mathbf{x}_1 + 0.5809\mathbf{x}_2^2 - 0.5809\mathbf{x}_2 + 1.1513$	1484.55	0.101748

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

Table J.11: Top $r = 10$ ranked symbolic expressions recovered by a single run of VaSST under $\sigma = 0.2$ for learning (17). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	LMPSE	RMSE
1	$1.0425\mathbf{x}_0^2 - 1.0203\mathbf{x}_1 + 0.5212\mathbf{x}_2^2 - 0.0378\mathbf{x}_0 + 0.0378\mathbf{x}_2 - 0.5396$	255.26	0.203331
2	$0.9826\mathbf{x}_0^2 - 1.0187\mathbf{x}_1 + 0.4675\mathbf{x}_2^2 + 0.5512\mathbf{x}_2 - 1.8176$	251.41	0.203457
3	$0.9832\mathbf{x}_0^2 - 1.0198\mathbf{x}_1 + 0.5125\mathbf{x}_2^2 - 0.1982$	251.27	0.203272
4	$0.9833\mathbf{x}_0^2 - 1.0205\mathbf{x}_1 + 0.5130\mathbf{x}_2^2 - 0.2039$	249.55	0.203274
5	$0.9841\mathbf{x}_0^2 - 0.2033\mathbf{x}_1^2 + 0.6355\mathbf{x}_2^2 - 1.2709\mathbf{x}_2 + 1.8591$	248.64	0.203483
6	$1.0234\mathbf{x}_0^2 - 0.0561\mathbf{x}_0 - 1.0234\mathbf{x}_1 + 0.5239\mathbf{x}_2^2 - 0.4025$	244.71	0.203299
7	$1.0745\mathbf{x}_0^2 - 0.0968\mathbf{x}_0 - 0.0968\mathbf{x}_1^2 - 0.5373\mathbf{x}_1 + 0.5125\mathbf{x}_2^2 - 0.7737$	243.29	0.203169
8	$1.0224\mathbf{x}_0^2 + 0.0008\mathbf{x}_0 - 1.0224\mathbf{x}_1 + 0.5258\mathbf{x}_2^2 - 0.4624$	241.63	0.203338
9	$0.9827\mathbf{x}_0^2 - 1.0124\mathbf{x}_1 + 0.6067\mathbf{x}_2^2 - 1.0124\mathbf{x}_2 + 2.5075$	239.59	0.203264
10	$0.9733\mathbf{x}_0^2 - 1.0212\mathbf{x}_1 + 0.5372\mathbf{x}_2^2 - 0.1011\mathbf{x}_2 - 0.2343$	238.96	0.203410

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

For $6\sin(\mathbf{x}_0)\cos(\mathbf{x}_1)$ in [Section 5.1](#).

Table J.12: Top $r = 10$ ranked symbolic expressions recovered by a single run of **VaSST** under the noiseless setting for learning (18). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	LMPSE	RMSE
1	$0.0787\mathbf{x}_0\mathbf{x}_1\cos(\mathbf{x}_1) + 5.6818\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0281\cos(\mathbf{x}_1)\cos(\mathbf{x}_0\mathbf{x}_1) - 0.0397$	2699.67	0.009557
2	$-0.0951\mathbf{x}_0 + 5.8568\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) + 0.0977\cos\{\cos(\mathbf{x}_1)\} - 0.0725$	2693.37	0.006417
3	$-0.0178\mathbf{x}_1 + 5.9318\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.1651\sin\{\cos\{\cos(\mathbf{x}_0)\}\} + 0.1208$	2687.91	0.004291
4	$0.0021\mathbf{x}_1\cos\{\cos(\mathbf{x}_1)\} - 0.0014\mathbf{x}_1 + 5.9929\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0028$	2687.49	0.001527
5	$5.9249\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) + 0.0343\cos(\mathbf{x}_1) + 0.1327\cos\{\sin(\mathbf{x}_0)\} - 0.1156$	2681.23	0.004575
6	$5.9704\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0567\cos\{\cos(\mathbf{x}_0)\} + 0.0267$	2680.73	0.002973
7	$0.0019\mathbf{x}_1\cos(\mathbf{x}_1) + 5.9929\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0028\sin(\mathbf{x}_1) + 0.0026$	2677.57	0.001530
8	$5.9709\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) + 0.0443\sin\{\cos(\mathbf{x}_0)\}\cos(\mathbf{x}_0) - 0.0359\cos\{\cos\{\cos(\mathbf{x}_0)\}\} - 0.0104$	2676.48	0.002941
9	$0.0270\sin\{\cos(\mathbf{x}_0\mathbf{x}_1)\} + 5.9225\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0011\mathbf{x}_1\cos(\mathbf{x}_1)\cos(\mathbf{x}_0\mathbf{x}_1) - 0.0378$	2664.00	0.004804
10	$6.3825\cos(\mathbf{x}_1)\sin\{\sin(\mathbf{x}_0)\} + 0.9289\sin\{\cos(\mathbf{x}_0)\} - 0.2152\sin\{\sin(\mathbf{x}_1\mathbf{x}_0)\} - 0.9499$	2568.17	0.009784

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

Table J.13: Top $r = 10$ ranked symbolic expressions recovered by a single run of **VaSST** under $\sigma = 0.1$ for learning (18). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	LMPSE	RMSE
1	$5.9931\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0023\mathbf{x}_1 - 0.0009\mathbf{x}_1\cos\{\cos(\mathbf{x}_1)\} + 0.0023$	1405.78	0.099728
2	$5.8878\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0735\mathbf{x}_0 + 0.0778\cos\{\cos(\mathbf{x}_1)\} - 0.0606$	1402.42	0.099916
3	$5.9670\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0706\sin\{\cos\{\cos(\mathbf{x}_0)\}\} - 0.0088\mathbf{x}_1 + 0.0508$	1401.50	0.099788
4	$5.9895\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0112\cos\{\cos(\mathbf{x}_0)\} + 0.0011$	1397.16	0.099739
5	$5.7657\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0181\cos(\mathbf{x}_0\mathbf{x}_1)\cos(\mathbf{x}_1) + 0.0593\mathbf{x}_0\mathbf{x}_1\cos(\mathbf{x}_1) - 0.0299$	1396.37	0.100150
6	$5.9927\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) + 0.0109\mathbf{x}_1\cos(\mathbf{x}_1) - 0.0279\sin(\mathbf{x}_1) + 0.0324$	1395.93	0.099724
7	$5.9616\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) + 0.0188\cos(\mathbf{x}_1) + 0.0608\cos\{\sin(\mathbf{x}_0)\} - 0.0545$	1394.34	0.099799
8	$5.9902\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) + 0.0578\sin\{\cos(\mathbf{x}_0)\}\cos(\mathbf{x}_0) - 0.1480\cos\{\cos\{\cos(\mathbf{x}_0)\}\} + 0.0736$	1393.06	0.099722
9	$0.0166\sin\{\cos(\mathbf{x}_0\mathbf{x}_1)\} + 5.9499\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0007\mathbf{x}_1\cos(\mathbf{x}_1)\cos(\mathbf{x}_0\mathbf{x}_1) - 0.0240$	1377.14	0.099818
10	$6.3906\sin\{\sin(\mathbf{x}_0)\}\cos(\mathbf{x}_1) + 0.8903\cos\{\sin(\mathbf{x}_0)\} + 0.1985\sin\{\sin(\mathbf{x}_0\mathbf{x}_1)\} - 0.9055$	1365.28	0.100016

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

Table J.14: Top $r = 10$ ranked symbolic expressions recovered by a single run of **VaSST** under $\sigma = 0.2$ for learning (18). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	LMPSE	RMSE
1	$5.9933\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0031\mathbf{x}_1 - 0.0039\mathbf{x}_1\cos\{\cos(\mathbf{x}_1)\} + 0.0074$	251.27	0.199438
2	$6.0022\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) + 0.0240\sin\{\cos\{\cos(\mathbf{x}_0)\}\} + 0.0002\mathbf{x}_1 - 0.0193$	246.74	0.199424
3	$5.9187\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0518\mathbf{x}_0 + 0.0578\cos\{\cos(\mathbf{x}_1)\} - 0.0486$	246.36	0.199518
4	$6.0085\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) + 0.0342\cos\{\cos(\mathbf{x}_0)\} - 0.0245$	242.70	0.199409
5	$5.9926\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) + 0.0199\mathbf{x}_1\cos(\mathbf{x}_1) - 0.0530\sin(\mathbf{x}_1) + 0.0621$	241.43	0.199430
6	$5.9982\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) + 0.0033\cos(\mathbf{x}_1) - 0.0110\cos\{\sin(\mathbf{x}_0)\} + 0.0066$	239.50	0.199429
7	$6.0095\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) + 0.0712\sin\{\cos(\mathbf{x}_0)\}\cos(\mathbf{x}_0) - 0.2601\cos\{\cos\{\cos(\mathbf{x}_0)\}\} + 0.1577$	238.64	0.199379
8	$5.8496\sin(\mathbf{x}_0)\cos(\mathbf{x}_1) - 0.0080\cos(\mathbf{x}_0\mathbf{x}_1)\cos(\mathbf{x}_1) + 0.0399\mathbf{x}_0\mathbf{x}_1\cos(\mathbf{x}_1) - 0.0200$	238.59	0.199601
9	$5.3764\mathbf{x}_0\cos(\mathbf{x}_1) - 2.6272\sin\{\cos(\mathbf{x}_0)\} - 0.7567\sin(\mathbf{x}_0) + 2.2304$	236.51	0.201138
10	$5.3674\mathbf{x}_0\cos(\mathbf{x}_1) - 0.7371\sin(\mathbf{x}_0)\cos(\mathbf{x}_0) - 1.6969\sin\{\cos(\mathbf{x}_0)\} + 1.4529$	236.17	0.201004

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

K DESCRIPTION OF FEYNMAN EQUATIONS

Table K.1: Description of the response variables and input features for the Feynman equations in [Section 5.2](#).

Label	Feynman Equation	Variable	Type	Description
(CL)	Coulomb's law	$F = 0.08 \frac{q_1 q_2}{\epsilon r^2}$	Response	Electrostatic force between two charges
		q_1, q_2	Input	Charges of two particles
		r	Input	Distance between the two charges
		ϵ	Constant	Permittivity of the medium
(CPE)	Change in gravitational potential energy	$\Delta U = G m_1 m_2 (\frac{1}{r_2} - \frac{1}{r_1})$	Response	Change in gravitational potential energy
		m_1, m_2	Input	Masses of two objects
		r_1, r_2	Input	Initial and final separation distances
		G	Constant	Gravitational constant
(FCE)	Force on a charge in electromagnetic field	$F = q(E_f + vB \sin \theta)$	Response	Force on a moving charge in electromagnetic field
		q	Input	Electric charge of the particle
		E_f	Input	Electric field strength
		v	Input	Particle velocity magnitude
		B	Input	Magnetic field strength
		θ	Input	Angle between velocity and magnetic field
(FTC)	Fourier's law of thermal conduction	$P = \frac{\kappa A (T_2 - T_1)}{d}$	Response	Rate of heat transfer or thermal power
		A	Input	Cross-sectional area for heat flow
		T_1, T_2	Input	Temperatures at the two boundaries
		d	Input	Distance between the two temperature points
		κ	Constant	Thermal conductivity of the material
(ADA)	Angular distribution asymmetry	$f = \beta^\dagger (1 + \alpha^\dagger \cos \theta)$	Response	Angular distribution as a function of θ
		$\beta^\dagger$	Input	Baseline scale parameter of the angular distribution
		$\alpha^\dagger$	Input	Angular asymmetry parameter
		θ	Input	Observation angle

L ADDITIONAL RESULTS FOR FEYNMAN EQUATIONS

L.1 SYMBOLIC EXPRESSIONS LEARNED BY VaSST AND COMPETING METHODS

For learning (CL): $F = 0.08 \frac{q_1 q_2}{\epsilon r^2}$ in Section 5.2.

Table L.1: Expressions recovered by VaSST and competing methods for learning (CL) under the noiseless setting.

Method	Symbolic Expression Learned	RMSE
VaSST	$0.0797 \frac{q_1 q_2}{\epsilon r^2} - 0.0001 \frac{q_1^2}{\epsilon r^2} - 0.0003$	0.000138
operon	$0.0799 \frac{q_1 q_2}{\epsilon r^2}$	1.08×10^{-7}
PySR	$0.0796 \frac{q_1 q_2}{\epsilon r^2}$	1.77×10^{-8}
gplearn	$0.0793 \frac{q_1 q_2}{\epsilon r^2}$	0.000554
DEAP	$0.2055 \frac{q_1^2}{\epsilon r^2}$	0.046139
BSR	$0.1469 - 0.0058 \frac{q_1^2 q_2^2}{\epsilon^3 r^6} + 0.0043 \frac{q_1 q_2^3}{\epsilon^2 r^4} + 0.0122 \frac{q_1^5 q_2}{\epsilon^3 r^6}$	0.0264
BMS	$a_0 \frac{q_1 q_2}{\epsilon r^2}$	3.70×10^{-17}
QLattice	$-0.7418 \left(0.5132 \frac{q_1^2}{\epsilon r^2} + 4.1738 \times 10^{-6} \right) \left(-0.2091 \frac{q_2}{q_1} - 1.9028 \times 10^{-6} \right) - 1.5291 \times 10^{-6}$	4.64×10^{-7}
DSR	$\frac{q_2^2}{\epsilon r^2}$	13.248890

Note: a_0 is a constant learned by BMS. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

Table L.2: Expressions recovered by VaSST and competing methods for learning (CL) under $\sigma = 0.1$.

Method	Symbolic Expression Learned	RMSE
VaSST	$0.080738 \frac{q_1 q_2}{\epsilon r^2} + 0.001896 \frac{q_1^2}{\epsilon r^2} - 0.008712$	0.099706
operon	$0.081301 \frac{q_1 q_2}{\epsilon r^2} - 0.008000$	0.099706
PySR	$0.211932 \frac{q_1^2}{\epsilon r^2}$	0.109963
gplearn	$0.079296 \frac{q_1 q_2}{\epsilon r^2}$	0.099793
DEAP	$0.190037 \frac{q_1^2}{\epsilon r^2}$	0.117293
BSR	$0.143050 - 0.005868 \frac{q_1^4 q_2^2}{\epsilon^3 r^6} + 0.004369 \frac{q_1 q_2^3}{\epsilon^2 r^4} + 0.012211 \frac{q_1^5 q_2}{\epsilon^3 r^6}$	0.103577
BMS	$a_0 \frac{q_1 q_2}{\epsilon r^2}$	0.099776
QLattice	$-0.612279 \left(0.121789 - 0.711697 \frac{q_1^2}{\epsilon r^2} \right) \left(0.200641 \frac{q_2}{q_1} - 0.0385421 \right) + 0.0278299$	0.099699
DSR	$\frac{q_2^2}{\epsilon r^2}$	13.247801

Note: a_0 is a constant learned by BMS. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

Table L.3: Expressions recovered by VaSST and competing methods for learning (CL) under $\sigma = 0.2$.

Method	Symbolic Expression Learned	RMSE
VaSST	$0.081740 \frac{q_1 q_2}{\epsilon r^2} + 0.003941 \frac{q_1^2}{\epsilon r^2} - 0.017138$	0.199411
operon	$0.082857 \frac{q_1 q_2}{\epsilon r^2} - 0.016000$	0.199412
PySR	$0.212564 \frac{q_1^2}{\epsilon r^2}$	0.205145
gplearn	$0.080850 \frac{q_1 q_2}{\epsilon r^2}$	0.199569
DEAP	$0.190037 \frac{q_1^2}{\epsilon r^2}$	0.209396
BSR	$0.139186 - 0.005879 \frac{q_1^4 q_2^2}{\epsilon^3 r^6} + 0.004428 \frac{q_1 q_2^3}{\epsilon^2 r^4} + 0.012213 \frac{q_1^5 q_2}{\epsilon^3 r^6}$	0.201605
BMS	$a_0 \frac{q_1 q_2}{\epsilon r^2}$	0.199551
QLattice	$-0.714214 \left(0.138143 - 0.685898 \frac{q_1^2}{\epsilon r^2} \right) \left(0.182712 \frac{q_2}{q_1} - 0.0355843 \right) + 0.0228361$	0.199442
DSR	$\frac{q_2^2}{\epsilon r^2}$	13.247464

Note: a_0 is a constant learned by BMS. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

For learning **(CPE)**: $\Delta U = Gm_1m_2(\frac{1}{r_2} - \frac{1}{r_1})$ in Section 5.2.

Table L.4: Expressions recovered by VaSST and competing methods for learning **(CPE)** under the noiseless setting.

Method	Symbolic Expression Learned	RMSE
VaSST	$1.014839 \frac{Gm_1m_2}{r_2} - 0.998513 \frac{Gm_1m_2}{r_1} - 0.003078 \frac{r_2^2}{r_1} - 0.021220$	0.003817
operon	$0.998287 \frac{Gm_1m_2}{r_2} - 0.999754 \frac{Gm_1m_2}{r_1}$	0.000155
PySR	$Gm_1m_2 \left(\frac{1}{r_2} - \frac{1}{r_1} \right)$	8.57×10^{-16}
gplearn	$Gm_1m_2 \left(\frac{1}{r_2} - \frac{1}{r_1} \right)$	0.000000
DEAP	$-\frac{Gm_1m_2}{r_1} - \frac{Gm_1^2}{r_1} + \frac{m_2}{m_1} + \frac{2.8770}{-Gm_1^2/r_1 + m_2/m_1} - 4.5144$	0.556532
BSR	$8.54998 + 0.826934 \frac{m_2r_2}{m_1r_1} + 0.253004 \left(\frac{Gm_1m_2r_2}{r_1^2} - \frac{Gm_1^2}{r_1} \right) + 0.379514 \frac{m_2^2}{m_1^2}$	1.090235
BMS	$-\frac{Gm_1m_2}{r_1} + \frac{Gm_1m_2}{a_0r_2}$	0.000000
QLattice	$15.5019 \left(-0.511191 \frac{Gm_1^2}{r_1} - 0.0180004 \right) \left(0.200553 \frac{m_2}{m_1} - 0.0060554 \right) \left(-0.0263949 \frac{r_2^2}{r_1} - 1.70128 \right) \{ -0.104696 \frac{r_2^2}{r_1} - \tanh(-0.114106 \frac{r_2^2}{r_1} - 1.02825) - 1.43374 \} + 0.161139$	0.072571
DSR	$Gm_1m_2 \left(\frac{1}{r_2} - \frac{1}{r_1} \right)$	8.58×10^{-16}

Note: a_0 is a constant learned by BMS. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

Table L.5: Expressions recovered by VaSST and competing methods for learning **(CPE)** under $\sigma = 0.1$.

Method	Symbolic Expression Learned	RMSE
VaSST	$1.014802 \frac{Gm_1m_2}{r_2} - 0.999747 \frac{Gm_1m_2}{r_1} - 0.004945 \frac{r_2^2}{r_1} - 0.038218$	0.099843
operon	$0.997823 \frac{Gm_1m_2}{r_2} - 1.003906 \frac{Gm_1m_2}{r_1} - 0.009000$	0.099712
PySR	$Gm_1m_2 \left(\frac{1}{r_2} - \frac{1}{r_1} \right)$	0.099784
gplearn	$Gm_1m_2 \left(\frac{1}{r_2} - \frac{1}{r_1} \right)$	0.099784
DEAP	$-\frac{Gm_1m_2}{r_1} - \frac{Gm_1^2}{r_1} + \frac{m_2}{m_1} + \frac{2.8770}{-Gm_1^2/r_1 + m_2/m_1} - 4.5144$	0.564690
BSR	$8.54536 + 0.827299 \frac{m_2r_2}{m_1r_1} + 0.253255 \left(\frac{Gm_1m_2r_2}{r_1^2} - \frac{Gm_1^2}{r_1} \right) + 0.379697 \frac{m_2^2}{m_1^2}$	1.094409
BMS	$-\frac{Gm_1m_2}{r_1} + \frac{a_0Gm_1m_2}{2r_2}$	0.099778
QLattice	$15.5533 \left(-0.230750 \frac{Gm_1^2}{r_1} + 0.00218109 \right) \left(0.355623 \frac{m_2}{m_1} - 0.144215 + 1.54939 \exp\{0.638983 \frac{r_2^2}{r_1}\} \right) + 15.5533 \exp[-0.734346(0.428116 \frac{m_2}{m_1} + 1)^2 - 0.747608(-0.00406677 \frac{r_2^2}{r_1} + 1)^2] - 0.100993$	0.126651
DSR	$Gm_1m_2 \left(\frac{1}{r_2} - \frac{1}{r_1} \right)$	0.099784

Note: a_0 is a constant learned by BMS. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

Table L.6: Expressions recovered by VaSST and competing methods for learning **(CPE)** under $\sigma = 0.2$.

Method	Symbolic Expression Learned	RMSE
VaSST	$1.014765 \frac{Gm_1m_2}{r_2} - 1.000981 \frac{Gm_1m_2}{r_1} - 0.006813 \frac{r_2^2}{r_1} - 0.055216$	0.199507
operon	$-0.010000 - 0.332 \left[\frac{0.707 Gm_1^2/r_1 - 6.275451 Gm_1m_2/r_1}{2.031 r_2/r_1} + 3.007296 \frac{Gm_1m_2}{r_1} + \frac{0.670 Gm_1^2/r_1}{3.388 Gm_1^2/r_1 + 2.718552 m_2^2/m_1^2} \right]$	0.199207
PySR	$Gm_1m_2 \left(\frac{1}{r_2} - \frac{1}{r_1} \right)$	0.199569
gplearn	$Gm_1m_2 \left(\frac{1}{r_2} - \frac{1}{r_1} \right)$	0.199569
DEAP	$-\frac{Gm_1m_2}{r_1} - \frac{Gm_1^2}{r_1} + \frac{m_2}{m_1} + \frac{2.8770}{-Gm_1^2/r_1 + m_2/m_1} - 4.5144$	0.589860
BSR	$8.54074 + 0.827664 \frac{m_2r_2}{m_1r_1} + 0.253507 \left(\frac{Gm_1m_2r_2}{r_1^2} - \frac{Gm_1^2}{r_1} \right) + 0.379881 \frac{m_2^2}{m_1^2}$	1.107579
BMS	$-\frac{Gm_1m_2}{r_1} + \frac{a_0Gm_1m_2}{2r_2}$	0.199557
QLattice	$2.11181 \left(-0.270397 \frac{Gm_1^2}{r_1} - 0.00788965 \right) \left(-0.0392111 \frac{r_2^2}{r_1} + 1.45354 + 1.86691 \exp\{0.281573 \frac{r_2^2}{r_1}\} \right) \sqrt{\left(\frac{m_2}{m_1} - 0.0186931 \right)^2} + 0.238388$	0.246633
DSR	$Gm_1m_2 \left(\frac{1}{r_2} - \frac{1}{r_1} \right)$	0.199569

Note: a_0 is a constant learned by BMS. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

For learning **(FCE)**: $F = q(E_f + Bv \sin \theta)$ in Section 5.2.

Table L.7: Expressions recovered by VaSST and competing methods for learning **(FCE)** under the noiseless setting.

Method	Symbolic Expression Learned	RMSE
VaSST	$0.979592qE_f + 1.066203qBv \sin \theta - 0.002652\theta + 0.001420$	0.001574
operon	$0.988416qE_f + 0.439639qBv \sin(0.982\theta) + 0.565508qBv \sin(1.014\theta) + 0.074$	0.277625
PySR	$qE_f + qBv \sin \theta$	0.000000
gplearn	$qE_f + qBv \sin \theta$	0.000000
DEAP	$\{qBv + 4 \sin \theta + \sin(\sin \theta) + \sin(\sin(\sin \theta))\} \sin \theta$	5.511325
BSR	$14.9902 + 0.245458 \left[\left(\frac{Bv}{E_f} + \theta^2 \right) \sin \theta + \sin \left(\frac{Bv}{E_f} \right) \right] + 0.182367 \frac{Bv}{E_f} \sin(qE_f \theta) - 2.18703 \left\{ \frac{Bv}{E_f} + \sin \left(\frac{Bv \theta}{E_f} \right) \right\}$	19.116390
BMS	$qBv \sin \theta + a_0^2 qE_f$	0.000000
QLattice	$95.8349(0.0317161qE_f - 1.20835) \exp[-2 \exp\{-1.83685(0.0751253 \frac{Bv}{E_f} - 1)^2 - 23.7706(0.125944\theta - 1)^2\} - 2 \exp\{-1.96799(0.634099\theta - 1)^2\}] + 74.3577$	13.875759
DSR	$qE_f + qBv \sin \theta$	0.000000

Note: a_0 is a constant learned by BMS. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

Table L.8: Expressions recovered by VaSST and competing methods for learning **(FCE)** under $\sigma = 0.1$.

Method	Symbolic Expression Learned	RMSE
VaSST	$0.979179qE_f + 1.066295qBv \sin \theta - 0.002220\theta + 0.001432$	0.101577
operon	$1.003148qE_f + 1.002258qBv \sin \theta - 0.043937 \sin(0.883\theta) - 0.012000$	0.101138
PySR	$qE_f + qBv \sin \theta$	0.098647
gplearn	$qE_f + qBv \sin \theta$	0.098647
DEAP	$\{qBv + 4 \sin \theta + \sin(\sin \theta) + \sin(\sin(\sin \theta))\} \sin \theta$	5.513179
BSR	$14.9833 + 0.245459 \left[\left(\frac{Bv}{E_f} + \theta^2 \right) \sin \theta + \sin \left(\frac{Bv}{E_f} \right) \right] + 0.182854 \frac{Bv}{E_f} \sin(qE_f \theta) - 2.18523 \left\{ \frac{Bv}{E_f} + \sin \left(\frac{Bv \theta}{E_f} \right) \right\}$	19.119963
BMS	$qE_f + qBv \sin \theta$	0.098647
QLattice	$255.963 \exp[-5.76788 \{0.446894 - \exp[-2.16672(1 - 0.213868\theta)^2]\}^2 - 1.78239(1 - 0.0236962qE_f)^2] - 35.3727$	14.287933
DSR	$qE_f + qBv \sin \theta$	0.098647

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

Table L.9: Expressions recovered by VaSST and competing methods for learning **(FCE)** under $\sigma = 0.2$.

Method	Symbolic Expression Learned	RMSE
VaSST	$0.978765qE_f + 1.066388qBv \sin \theta - 0.001789\theta + 0.001444$	0.201587
operon	$0.995742qE_f + 0.427023qBv \sin(0.989\theta) + 0.576095qBv \sin(1.008\theta) + 0.033000$	0.216361
PySR	$1.000169qE_f + qBv \sin \theta$	0.197290
gplearn	$qE_f + qBv \sin \theta$	0.197294
DEAP	$\{qBv + 4 \sin \theta + \sin(\sin \theta) + \sin(\sin(\sin \theta))\} \sin \theta$	5.516797
BSR	$14.9764 + 0.245460 \left[\left(\frac{Bv}{E_f} + \theta^2 \right) \sin \theta + \sin \left(\frac{Bv}{E_f} \right) \right] + 0.183341 \frac{Bv}{E_f} \sin(qE_f \theta) - 2.18344 \left\{ \frac{Bv}{E_f} + \sin \left(\frac{Bv \theta}{E_f} \right) \right\}$	19.124043
BMS	$qE_f + qBv \sin \theta$	0.197294
QLattice	$71.7853 - 111.307 \exp[-2\{0.0512725\theta + \exp[-0.887081(1 - 0.0224396qE_f)^2 - 5.71614(-0.022938 \frac{Bv}{E_f} + 0.150368\theta - 1)^2] - 0.225009\}^2 - 2 \exp\{-1.30897(0.61342\theta - 1)^2\}]$	13.574922
DSR	$qE_f + qBv \sin \theta$	0.197294

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

For learning **(FTC)**: $P = \frac{\kappa A(T_2 - T_1)}{d}$ in Section 5.2.

Table L.10: Expressions recovered by **VaSST** and competing methods for learning **(FTC)** under the noiseless setting.

Method	Symbolic Expression Learned	RMSE
VaSST	$-0.805371 \frac{AT_1 \kappa}{d} + 0.998520 \frac{A\kappa T_2}{d} - 0.000254 \left\{ \frac{1}{\sqrt{A} T_1 \kappa} - \frac{d/\sqrt{A} - T_2/T_1}{\sqrt{A} T_1 \kappa - T_2/T_1} \right\} + 0.014782$	0.012602
operon	$6.66730 \frac{AT_1 \kappa}{d} \left(0.150000 \frac{T_2}{T_1} - 0.149937 \right)$	6.30×10^{-5}
PySR	$\frac{A\kappa(T_2 - T_1)}{d}$	1.59×10^{-16}
gplearn	$\frac{A\kappa(T_2 - T_1)}{d}$	1.71×10^{-16}
DEAP	$0.675078 \frac{A\kappa T_2}{d}$	0.199012
BSR	$1.06626 - 0.113972 \frac{dT_2}{\sqrt{A} T_1} + 0.031160 \left(\frac{d\kappa T_2}{\sqrt{A}} - \sqrt{A} T_1 \kappa \right) + 0.069105 \frac{\sqrt{A} T_2^2}{dT_1^2}$	0.178899
BMS	$a_0 \frac{A\kappa T_2}{d}$	0.074917
QLattice	$-3.01307 \left(0.262817 - 0.263207 \frac{T_2}{T_1} \right) \left(-0.597221 \sqrt{A} T_1 \kappa - 0.00029899 \right) \frac{0.000501386d/\sqrt{A} - 0.584919}{0.274851d/\sqrt{A} + 0.00563991} - 0.000457$	0.000277
DSR	$\frac{A\kappa(T_2 - T_1)}{d}$	2.18×10^{-16}

Note: a_0 is a constant learned by BMS. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

Table L.11: Expressions recovered by **VaSST** and competing methods for learning **(FTC)** under $\sigma = 0.1$.

Method	Symbolic Expression Learned	RMSE
VaSST	$0.871347 \frac{A\kappa T_2}{d} - 0.019212 \sqrt{A} T_1 \kappa - 0.046585 \left(\sqrt{A} T_1 \kappa - 1 - \frac{T_2}{T_1} \right) - 0.293006$	0.101254
operon	$1.007879 \frac{A\kappa T_2}{d} - 1.015839 \frac{AT_1 \kappa}{d} + 0.249325 \frac{\sqrt{A}}{d} - 0.063014$	0.099538
PySR	$\frac{A\kappa(T_2 - 1.001694T_1)}{d}$	0.099782
gplearn	$0.900000 \frac{A\kappa(T_2 - 0.720000T_1)}{d}$	0.111707
DEAP	$0.675078 \frac{A\kappa T_2}{d}$	0.223923
BSR	$1.09040 - 0.117204 \frac{dT_2}{\sqrt{A} T_1} + 0.031340 \left(\frac{d\kappa T_2}{\sqrt{A}} - \sqrt{A} T_1 \kappa \right) + 0.070823 \frac{\sqrt{A} T_2^2}{dT_1^2}$	0.206912
BMS	$a_0 \frac{A\kappa T_2}{d}$	0.126948
QLattice	$-1.19133 \left(0.409023 - 0.399587 \frac{T_2}{T_1} \right) \left(1.18272 \sqrt{A} T_1 \kappa - 0.0111254 \right) \exp \left(-0.203901 \frac{d}{\sqrt{A}} \right) + 0.0143714$	0.101341
DSR	$\frac{A\kappa(T_2 - T_1)}{d}$	0.099784

Note: a_0 is a constant learned by BMS. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

Table L.12: Expressions recovered by **VaSST** and competing methods for learning **(FTC)** under $\sigma = 0.2$.

Method	Symbolic Expression Learned	RMSE
VaSST	$0.903072 \frac{A\kappa T_2}{d} - 0.094430 \sqrt{A} T_1 \kappa - 0.661969 \frac{T_1}{T_2} + 0.112071$	0.199821
operon	$-0.021000 + 0.100 \left[2.053 \frac{\sqrt{A}}{d} \left\{ 1.071212 \frac{T_1}{T_2} + 6.962264 \sqrt{A} T_1 \kappa \right\} \left(0.707 \frac{T_2}{T_1} - 0.693 \right) - 0.002139 \left(\frac{d}{\sqrt{A}} \right)^3 \right]$	0.198948
PySR	$\frac{A\kappa(T_2 - 1.003386T_1)}{d}$	0.199565
gplearn	$\frac{A\kappa(T_2 - 0.900000T_1)}{d}$	0.203313
DEAP	$0.675078 \frac{A\kappa T_2}{d}$	0.283886
BSR	$1.11454 - 0.120437 \frac{dT_2}{\sqrt{A} T_1} + 0.031521 \left(\frac{d\kappa T_2}{\sqrt{A}} - \sqrt{A} T_1 \kappa \right) + 0.072541 \frac{\sqrt{A} T_2^2}{dT_1^2}$	0.271019
BMS	$\frac{A\kappa T_2}{a_0 d}$	0.215714
QLattice	$2.2079(0.36293 \frac{T_2}{T_1} - 0.419621)[0.167743 \sqrt{A} T_1 \kappa + (0.0940613 \frac{d}{\sqrt{A}} - 0.639193)(-0.581647 \sqrt{A} T_1 \kappa + 0.248409 \frac{d}{\sqrt{A}} - 0.775808) + 0.0568153] + 0.0549773$	0.200905
DSR	$\frac{A\kappa(T_2 - T_1)}{d}$	0.199569

Note: a_0 is a constant learned by BMS. RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

For learning (ADA): $f = \beta^\dagger(1 + \alpha^\dagger \cos \theta)$ in Section 5.2.

Table L.13: Expressions recovered by VaSST and competing methods for learning (ADA) under the noiseless setting.

Method	Symbolic Expression Learned	RMSE
VaSST	$1.023842\beta^\dagger\{1 + \alpha^\dagger \cos \theta\} + 0.001029\theta + 0.000125\cos(2\alpha^\dagger) - 0.000899$	0.001349
operon	$1.91884\beta^\dagger\{0.520\alpha^\dagger \cos(0.994\theta) + 0.002\theta + 0.366\} + 0.417217\beta^\dagger \cos(0.594\theta) \cos(0.583\theta) + 0.004$	0.133614
PySR	$\beta^\dagger\{1 + \alpha^\dagger \cos \theta\}$	0.000000
gplearn	$\beta^\dagger + \alpha^\dagger \beta^\dagger \cos \theta$	1.39×10^{-15}
DEAP	$\cos \theta \{ \alpha^\dagger \beta^\dagger + 3 \cos \theta + \cos(1.020634\theta) \}$	2.220867
BSR	$88.5628 - 11.4517\theta + 0.643483\cos(\alpha^\dagger) + 0.065444\{\cos((\beta^\dagger)^2) + 2\beta^\dagger\theta\}$	7.863577
BMS	$\beta^\dagger + \alpha^\dagger \beta^\dagger \cos \theta$	1.80×10^{-15}
QLattice	$-225.967[-0.0466202\theta - 0.307711 + \exp\{-2\exp(-0.302799(0.0450206\alpha^\dagger + 1)^2 - 11.5544(0.164397\theta - 1)^2)\}\exp\{-2.16269(0.14257\beta^\dagger - 1)^2 - 2.24866(0.0880658\alpha^\dagger - 1)^2\} + 0.330437$	0.478315
DSR	$\beta^\dagger + \alpha^\dagger \beta^\dagger \cos \theta$	1.41×10^{-15}

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

Table L.14: Expressions recovered by VaSST and competing methods for learning (ADA) under $\sigma = 0.1$.

Method	Symbolic Expression Learned	RMSE
VaSST	$1.024106\beta^\dagger\{1 + \alpha^\dagger \cos \theta\} + 0.000106\theta + 0.000302\cos(2\alpha^\dagger) - 0.000092$	0.108279
operon	$1.015367\alpha^\dagger \beta^\dagger \cos(0.996\theta) + 1.46096\beta^\dagger \cos[\cos\{-0.872 + 0.607\theta + \cos(0.291\theta)\}] + 0.015$	0.158603
PySR	$\beta^\dagger\{\alpha^\dagger \cos \theta + 0.999743\}$	0.099780
gplearn	$\beta^\dagger + \alpha^\dagger \beta^\dagger \cos \theta$	0.099784
DEAP	$\cos \theta \{ \alpha^\dagger \beta^\dagger + 3 \cos \theta + \cos(1.020634\theta) \}$	2.225102
BSR	$88.5585 - 11.4506\theta + 0.646516\cos(\alpha^\dagger) + 0.065331\{\cos((\beta^\dagger)^2) + 2\beta^\dagger\theta\}$	7.866517
BMS	$\beta^\dagger + \alpha^\dagger \beta^\dagger \cos \theta$	0.099784
QLattice	$-39.2975(0.00653208 - 0.317349\beta^\dagger)(0.108825 - 0.269264\alpha^\dagger)[-0.00439472\beta^\dagger - 0.678794 + \exp\{-2\exp[-15.4029(1 - 0.159096\theta)^2 - 0.419709(0.0349345\alpha^\dagger + 1)^2]\}] + 1.02986$	0.467908
DSR	$\beta^\dagger + \alpha^\dagger \beta^\dagger \cos \theta$	0.099784

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

Table L.15: Expressions recovered by VaSST and competing methods for learning (ADA) under $\sigma = 0.2$.

Method	Symbolic Expression Learned	RMSE
VaSST	$1.024369\beta^\dagger\{1 + \alpha^\dagger \cos \theta\} + 0.001094\theta + 0.00048\cos(2\alpha^\dagger) - 0.000952$	0.200414
operon	$0.995938\beta^\dagger + 1.000753\alpha^\dagger \beta^\dagger \cos \theta + 0.007000$	0.199415
PySR	$\beta^\dagger\{\alpha^\dagger \cos \theta + 0.999471\}$	0.199560
gplearn	$\beta^\dagger + \alpha^\dagger \beta^\dagger \cos \theta$	0.199569
DEAP	$\cos \theta \{ \alpha^\dagger \beta^\dagger + 3 \cos \theta + \cos(1.020634\theta) \}$	2.233791
BSR	$88.5542 - 11.4494\theta + 0.649548\cos(\alpha^\dagger) + 0.065219\{\cos((\beta^\dagger)^2) + 2\beta^\dagger\theta\}$	7.870720
BMS	$\beta^\dagger + \alpha^\dagger \beta^\dagger \cos \theta$	0.199569
QLattice	$-41.1561(0.691287\beta^\dagger - 0.000452184)[-0.000217418\beta^\dagger - 0.285239\theta + \tanh\{(6.8735 - 0.874459\theta)(-0.0417498\alpha^\dagger - 0.324422)\}] + 2.20758] + 0.0668375$	0.416285
DSR	$\beta^\dagger + \alpha^\dagger \beta^\dagger \cos \theta$	0.199569

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set.

L.2 COMPUTE RUNTIMES FOR VaSST AND COMPETING BAYESIAN SR METHODS

Table L.16: Single-run compute runtimes (in seconds) of **VaSST** and competing Bayesian methods for learning the Feynman equations in [Section 5.2](#) under different noise settings.

Feynman Equation	Method	Noiseless	$\sigma^2 = 0.1^2$	$\sigma^2 = 0.2^2$
(CL) : $F = 0.08 \frac{q_1 q_2}{\epsilon r^2}$	VaSST	96.41	94.45	96.04
	BMS	143.55	144.23	144.48
	BSR	319.96	315.43	328.92
(CPE) : $\Delta U = Gm_1 m_2 (\frac{1}{r_2} - \frac{1}{r_1})$	VaSST	129.45	108.75	109.66
	BMS	155.50	224.50	232.54
	BSR	242.06	240.68	244.30
(FCE) : $F = q(E_f + vB \sin \theta)$	VaSST	121.54	114.00	109.91
	BMS	184.79	222.07	340.54
	BSR	246.79	252.23	256.57
(FTC) : $P = \frac{\kappa A(T_2 - T_1)}{d}$	VaSST	108.87	117.82	115.23
	BMS	143.06	141.42	144.24
	BSR	236.09	249.89	246.77
(ADA) : $f = \beta^\dagger (1 + \alpha^\dagger \cos \theta)$	VaSST	104.72	116.34	107.88
	BMS	261.41	268.45	264.55
	BSR	252.54	257.96	259.78

Note: Runtimes may vary depending on the machine architecture and computational environment. The reported runtimes were obtained on a MacBook Air with an Apple M2 chip and 8GB RAM.

L.3 TOP RANKED SYMBOLIC EXPRESSIONS LEARNED BY VaSST WITH RESPECT TO LMPSE

For learning (CL): $F = 0.08 \frac{q_1 q_2}{\epsilon r^2}$ in Section 5.2.

Table L.17: Top $r = 10$ ranked symbolic expressions recovered by a single run of VaSST under the noiseless setting for learning (CL). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	RMSE	LMPSE
1	$-0.0003 - \frac{0.0001q_1^2}{\epsilon r^2} + \frac{0.0797q_1 q_2}{\epsilon r^2}$	0.0001	3365.2575
2	$-0.0003 - \frac{2.4261 \times 10^{-5} q_2}{q_1} + \frac{0.0797q_1 q_2}{\epsilon r^2}$	0.0001	3364.8864
3	$-0.0003 - \frac{0.0001q_1^2}{\epsilon r^2} + \frac{0.0797q_1 q_2}{\epsilon r^2}$	0.0001	3363.8731
4	$-0.0003 - \frac{2.2908 \times 10^{-5} q_2}{q_1} + \frac{0.0797q_1 q_2}{\epsilon r^2}$	0.0001	3363.5020
5	$-0.0003 - \frac{2.2908 \times 10^{-5} q_2}{q_1} + \frac{0.0797q_1 q_2}{\epsilon r^2}$	0.0001	3363.5020
6	$-0.0003 + \frac{0.0796q_1 q_2}{\epsilon r^2} + \frac{2.8973 \times 10^{-5} q_2^2}{\epsilon r^2}$	0.0001	3362.8026
7	$0.0003 - \frac{8.6888 \times 10^{-5} q_2^2}{q_1^2} - \frac{0.0007q_1^2}{\epsilon r^2} + \frac{0.0799q_1 q_2}{\epsilon r^2}$	0.0001	3360.9947
8	$-0.0003 + \frac{0.0061q_1^2}{\epsilon r^2} + \frac{0.0749q_1 q_2}{\epsilon r^2} + \frac{0.0009q_2^2}{\epsilon r^2}$	0.0002	3360.1902
9	$0.0012 - \frac{0.0006q_2}{q_1} + \frac{0.0793q_1 q_2}{\epsilon r^2} + \frac{0.0001q_2^2}{\epsilon r^2}$	0.0001	3358.6486
10	$-9.3659 \times 10^{-5} + \frac{0.0796q_1 q_2}{\epsilon r^2} + \frac{5.2089 \times 10^{-6} q_1^8}{\epsilon^4 r^8}$	0.0001	3358.2041

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

Table L.18: Top $r = 10$ ranked symbolic expressions recovered by a single run of VaSST under $\sigma = 0.1$ for learning (CL). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	RMSE	LMPSE
1	$\frac{-0.0087\epsilon r^2 + 0.0019q_1^2 + 0.0807q_1 q_2}{\epsilon r^2}$	0.0997	1574.9014
2	$0.0018 - \frac{0.0043q_2}{q_1} + \frac{0.0816q_1 q_2}{\epsilon r^2}$	0.0997	1574.6367
3	$\frac{-0.0087\epsilon r^2 + 0.0019q_1^2 + 0.0807q_1 q_2}{\epsilon r^2}$	0.0997	1573.5842
4	$0.0018 - \frac{0.0043q_2}{q_1} + \frac{0.0816q_1 q_2}{\epsilon r^2}$	0.0997	1573.3191
5	$0.0018 - \frac{0.0043q_2}{q_1} + \frac{0.0816q_1 q_2}{\epsilon r^2}$	0.0997	1573.3191
6	$\frac{-0.0086\epsilon r^2 + 0.0819q_1 q_2 - 0.0002q_2^2}{\epsilon r^2}$	0.0997	1572.4406
7	$0.0127 - \frac{0.0031q_2^2}{q_1^2} - \frac{0.0186q_1^2}{\epsilon r^2} + \frac{0.0885q_1 q_2}{\epsilon r^2}$	0.0997	1571.0526
8	$\frac{-0.0091\epsilon r^2 + 0.0421q_1^2 + 0.0499q_1 q_2 + 0.0059q_2^2}{\epsilon r^2}$	0.0997	1569.9638
9	$\frac{-0.0097\epsilon r^2 + 0.1073q_1^2 + 0.0154q_2^2}{\epsilon r^2}$	0.0997	1569.3446
10	$0.0253 - \frac{0.0131q_2}{q_1} + \frac{0.0757q_1 q_2}{\epsilon r^2} + \frac{0.0022q_2^2}{\epsilon r^2}$	0.0997	1568.6428

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

Table L.19: Top $r = 10$ ranked symbolic expressions recovered by a single run of VaSST under $\sigma = 0.2$ for learning (CL). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	RMSE	LMPSE
1	$\frac{-0.0171\epsilon r^2 + 0.0039q_1^2 + 0.0817q_1 q_2}{\epsilon r^2}$	0.1994	320.3584
2	$0.0039 - \frac{0.0085q_2}{q_1} + \frac{0.0835q_1 q_2}{\epsilon r^2}$	0.1994	320.1072
3	$\frac{-0.0171\epsilon r^2 + 0.0040q_1^2 + 0.0817q_1 q_2}{\epsilon r^2}$	0.1994	319.0511
4	$0.0039 - \frac{0.0085q_2}{q_1} + \frac{0.0835q_1 q_2}{\epsilon r^2}$	0.1994	318.7996
5	$0.0039 - \frac{0.0085q_2}{q_1} + \frac{0.0835q_1 q_2}{\epsilon r^2}$	0.1994	318.7996

Continued on next page

Table L.19 continued from previous page.

Rank	VaSST Expression Learned	RMSE	LMPSE
6	$\frac{-0.0169\epsilon r^2 + 0.0843q_1 q_2 - 0.0004q_2^2}{\epsilon r^2}$	0.1994	317.8968
7	$0.0251 - \frac{0.0061q_2^2}{q_1^2} - \frac{0.0366q_1^2}{\epsilon r^2} + \frac{0.0971q_1 q_2}{\epsilon r^2}$	0.1994	316.5585
8	$\frac{-0.0178\epsilon r^2 + 0.0780q_1^2 + 0.0249q_1 q_2 + 0.0108q_2^2}{\epsilon r^2}$	0.1994	315.4286
9	$\frac{-0.0181\epsilon r^2 + 0.1105q_1^2 + 0.0156q_2^2}{\epsilon r^2}$	0.1994	314.9703
10	$0.0494 - \frac{0.0255q_2}{q_1} + \frac{0.0721q_1 q_2}{\epsilon r^2} + \frac{0.0042q_2^2}{\epsilon r^2}$	0.1994	314.1365

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

For learning (CPE): $\Delta U = Gm_1 m_2 (\frac{1}{r_2} - \frac{1}{r_1})$ in Section 5.2.

Table L.20: Top $r = 10$ ranked symbolic expressions recovered by a single run of VaSST under the noiseless setting for learning (CPE). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	RMSE	LMPSE
1	$\frac{1.0148Gm_1 m_2}{r_2} - \frac{0.9985Gm_1 m_2}{r_1} - 0.0212 - \frac{0.0031r_2}{r_1}$	0.0038	3278.4007
2	$\frac{0.0009G^2 m_1^3 m_2}{r_1^2} + \frac{1.0107Gm_1 m_2}{r_2} - \frac{0.9965Gm_1 m_2}{r_1} - 0.0007$	0.0037	3199.2274
3	$\frac{0.0064G^2 m_1^5}{m_2 r_1 r_2} + \frac{0.0064Gm_1^3}{m_2 r_1} - \frac{0.0010Gm_1^2 \{2Gm_1^2/r_1 - 1\}}{r_1} - \frac{1.0073Gm_1^2 \{-m_2 r_1/(m_1 r_2) + m_2/m_1\}}{r_1} - 0.0086 - \frac{0.0064m_2}{m_1}$	0.0152	3174.0205
4	$\frac{2.2506G^2 m_1^3 m_2}{r_1^2 \{-2Gm_1^2/r_1 + 1\}} + \frac{0.0873Gm_1 m_2 \{-Gm_1^2/r_1 + m_2/m_1\}}{r_2} + 0.9087 - \frac{0.1176m_2 r_1 \{-Gm_1^2/r_1 + m_2/m_1\}}{m_1 r_2}$	0.1501	770.4722
5	$\frac{1.0486Gm_1^2}{r_1} - \frac{1.8288Gm_1 m_2}{r_1} + 1.0821 + \frac{0.6100r_2 \{Gm_1 m_2/r_1 + m_2 r_1/(m_1 r_2)\}}{r_1 \{Gm_1^2/r_1 + r_2/r_1\}}$	0.1915	357.5249
6	$\frac{0.6894Gm_1^2}{r_1} - \frac{1.4571Gm_1 m_2}{r_1} + 1.5451 + \frac{0.0898r_2}{r_1} - \frac{1.4571m_2 r_1}{m_1 r_2} - \frac{0.6894m_2}{m_1}$	0.2020	245.0738
7	$\frac{0.0361Gm_1^2}{r_1} - \frac{1.2526Gm_1 m_2}{r_1} + \frac{17.2393r_2^2}{r_2} - 0.6275 + \frac{0.0542r_2}{r_1} - \frac{0.0181m_2}{m_1}$	0.1932	13.0193
8	$-\frac{0.8731Gm_1^2 \{-r_2/r_1 + m_2/m_1\}}{r_1} - \frac{0.7120Gm_1^2 r_2}{r_1^2} - 0.8223 + \frac{0.8731m_2 r_1 \{Gm_1^2/r_1 + r_2/r_1\}}{m_1 r_2} + \frac{0.1539m_2 r_2}{m_1 r_1}$	0.2330	-31.8476
9	$\frac{1.0196Gm_1^2}{r_1} - \frac{1.4805Gm_1 m_2}{r_1} + 2.4189 + \frac{0.1999r_2}{r_1} - \frac{0.3998m_2}{m_1}$	0.2354	-48.6053
10	$\frac{0.0025Gm_1^2}{r_2} + \frac{0.8171Gm_1 m_2 \{-r_2/r_1 + m_2/m_1\}}{r_1 \{r_2/r_1 + m_2 r_1/(Gm_1^3)\}} + 0.0206 - \frac{0.0051r_2}{r_1} + \frac{0.0245m_2 \{-2r_2/r_1 + 2m_2/m_1\}}{m_1} + \frac{0.0025m_2}{m_1}$	0.2363	-66.4971

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

Table L.21: Top $r = 10$ ranked symbolic expressions recovered by a single run of VaSST under $\sigma = 0.1$ for learning (CPE). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	RMSE	LMPSE
1	$\frac{1.0148Gm_1 m_2}{r_2} - \frac{0.9997Gm_1 m_2}{r_1} - 0.0382 - \frac{0.0049r_2}{r_1}$	0.0998	1534.4676
2	$\frac{0.0011G^2 m_1^3 m_2}{r_1^2} + \frac{1.0077Gm_1 m_2}{r_2} - \frac{0.9980Gm_1 m_2}{r_1} - 0.0077$	0.0998	1513.0846
3	$\frac{0.0046G^2 m_1^5}{m_2 r_1 r_2} - \frac{0.0009G^2 m_1^4}{r_1^2} + \frac{0.0046Gm_1^3}{m_2 r_1} + \frac{0.0004Gm_1^2}{r_1} + \frac{1.0071Gm_1 m_2}{r_2} - \frac{1.0071Gm_1 m_2}{r_1} - 0.0194 - \frac{0.0046m_2}{m_1}$	0.1010	1489.7953
4	$\frac{2.2538G^2 m_1^3 m_2 r_2 - 0.9018m_1^2 r_1 r_2 \{2Gm_1^2 - r_1\} + m_2 \{0.0872Gm_1^2 - 0.1176r_1\} \{2Gm_1^2 - r_1\} \{Gm_1^3 - m_2 r_1\}}{m_1^2 r_1 r_2 \{2Gm_1^2 - r_1\}}$	0.1781	455.0898
5	$\frac{-0.6088Gm_1^2 m_2 r_2 + m_1 \{Gm_1^2 + r_2\} \{-1.0444Gm_1^2 + 1.8285Gm_1 m_2 - 1.0711r_1\} - 0.6088m_2 r_1^2}{-m_1 r_1 \{Gm_1^2 + r_2\}}$	0.2157	130.8494
6	$\frac{0.6888Gm_1^2}{r_1} - \frac{1.4581Gm_1 m_2}{r_1} + 1.5241 + \frac{0.0876r_2}{r_1} - \frac{1.4581m_2 r_1}{m_1 r_2} - \frac{0.6888m_2}{m_1}$	0.2255	35.3900
7	$\frac{0.0354Gm_1^2}{r_1} - \frac{1.2537Gm_1 m_2}{r_1} + \frac{17.1818r_2^2}{r_2} - 0.6402 + \frac{0.0531r_2}{r_1} - \frac{0.0177m_2}{m_1}$	0.2187	-161.2716
8	$\frac{0.1616Gm_1^2 r_2}{r_1^2} + \frac{0.8735Gm_1 m_2}{r_2} - \frac{0.8735Gm_1 m_2}{r_1} - 0.8325 + \frac{0.8735m_2}{m_1} + \frac{0.1539m_2 r_2}{m_1 r_1}$	0.2529	-190.5952
9	$\frac{1.0151Gm_1^2}{r_1} - \frac{1.4807Gm_1 m_2}{r_1} + 2.4027 + \frac{0.1993r_2}{r_1} - \frac{0.3986m_2}{m_1}$	0.2555	-205.9995
10	$\frac{0.8178G^2 m_1^5 m_2 r_2 \{m_1 r_2 - m_2 r_1\} - 0.0038Gm_1^4 r_1 \{Gm_1^3 r_2 + m_2 r_1^2\} + r_2 \{Gm_1^3 r_2 + m_2 r_1^2\} \{0.0018m_1^2 r_1 + 0.0038m_1 r_2^2 - 0.0245m_2 r_1 \{2m_1 r_2 - m_2 r_1\}\}}{-r_1 r_2 \{Gm_1^3 r_2 + m_2 r_1^2\}}$	0.2569	-228.4333

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

Table L.22: Top $r = 10$ ranked symbolic expressions recovered by a single run of **VaSST** under $\sigma = 0.2$ for learning **(CPE)**. Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	RMSE	LMPSE
1	$\frac{1.0148Gm_1m_2}{r_2} - \frac{1.0010Gm_1m_2}{r_1} - 0.0552 - \frac{0.0068r_2}{r_1}$	0.1995	287.4354
2	$\frac{0.0012G^2m_1^2m_2}{r_1^2} + \frac{1.0048Gm_1m_2}{r_2} - \frac{0.9994Gm_1m_2}{r_1} - 0.0148$	0.1995	275.0103
3	$\frac{0.0028G^2m_1^5}{m_2r_1r_2} + \frac{0.0002G^2m_1^4}{r_1^2} + \frac{0.0028Gm_1^3}{m_2r_1} - \frac{0.0001Gm_1^2}{r_1} + \frac{1.0068Gm_1m_2}{r_2} - \frac{1.0068Gm_1m_2}{r_1} - 0.0302 - \frac{0.0028m_2}{m_1}$	0.2002	252.9171
4	$\frac{2.2570G^2m_1^5m_2r_2 - 0.8948m_1^2r_1r_2\{2Gm_1^2 - r_1\} + m_2\{0.0871Gm_1^2 - 0.1175r_1\}\{2Gm_1^2 - r_1\}\{Gm_1^3 - m_2r_1\}}{-m_1^2r_1r_2\{2Gm_1^2 - r_1\}}$	0.2466	-163.8244
5	$\frac{-m_1\{0.6077Gm_1m_2r_2 - (Gm_1^2 + r_2)\}(-1.0402Gm_1^2 + 1.8282Gm_1m_2 - 1.0601r_1) - 0.6077m_2r_1^2}{-m_1r_1\{Gm_1^2 + r_2\}}$	0.2762	-346.8534
6	$\frac{0.6883Gm_1^2}{r_1} - \frac{1.4591Gm_1m_2}{r_1} + 1.5032 + \frac{0.0854r_2}{r_1} - \frac{1.4591m_2r_1}{m_1r_2} - \frac{0.6883m_2}{m_1}$	0.2842	-412.0230
7	$\frac{0.0346Gm_1^2}{r_1} - \frac{1.2549Gm_1m_2}{r_1} + \frac{17.1240r_1^2}{r_2^2} - 0.6530 + \frac{0.0519r_2}{r_1} - \frac{0.0173m_2}{m_1}$	0.2796	-544.3299
8	$\frac{0.1622Gm_1^2r_2}{r_1^2} + \frac{0.8740Gm_1m_2}{r_2} - \frac{0.8740Gm_1m_2}{r_1} - 0.8426 + \frac{0.8740m_2}{m_1} + \frac{0.1540m_2r_2}{m_1r_1}$	0.3058	-561.1996
9	$\frac{1.0107Gm_1^2}{r_1} - \frac{1.4810Gm_1m_2}{r_1} + 2.3864 + \frac{0.1987r_2}{r_1} - \frac{0.3974m_2}{m_1}$	0.3083	-570.1379
10	$\frac{0.8185G^2m_1^5m_2r_2\{m_1r_2 - m_2r_1\} - 0.0051Gm_1^4r_1\{Gm_1^3r_2 + m_2r_1^2\}}{-r_1r_2\{Gm_1^3r_2 + m_2r_1^2\}} + \frac{r_2\{Gm_1^3r_2 + m_2r_1^2\}\{+0.0241m_1^2r_1 + 0.0102m_1^2r_2 - 0.0051m_1m_2r_1 - 0.0490m_2^2r_1\}}{-r_1r_2\{Gm_1^3r_2 + m_2r_1^2\}}$	0.3099	-594.8235

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

For learning **(FCE)**: $F = q(E_f + Bv \sin \theta)$ in Section 5.2.

Table L.23: Top $r = 10$ ranked symbolic expressions recovered by a single run of **VaSST** under the noiseless setting for learning **(FCE)**. Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	RMSE	LMPSE
1	$1.0662Bvq \sin \theta + 0.9796E_fq - 0.0027\theta + 0.001420$	0.0016	-3800.7159
2	$0.1920Bvq\theta - \frac{0.1550Bv\theta(E_fq + \theta)}{E_f} + 0.8769(Bvq + E_fq) \sin \theta + 0.1920 \sin\{\sin(E_fq)\} + 0.0121$	0.0055	-4807.8071
3	$1.0511Bvq \sin \theta + 0.1952Bvq + \frac{0.1952Bv\theta}{E_f} + \frac{1.0511Bv \sin(Bv/E_f)}{E_f} + \frac{0.1952Bv}{E_f} + 0.3903\theta - 0.0395(Bv/E_f + 2\theta)(Bv/E_f + \theta^2) + 0.0001$	0.0065	-4896.5806
4	$1.0389Bvq \sin \theta + 0.0979\theta + 1.6425 \sin(E_fq) + 0.0063$	0.0080	-4954.2483
5	$1.1034Bvq \sin \theta - \frac{0.3719Bv}{E_f} - 3.7914 \sin\{\sin \theta\} + 0.097$	0.0081	-4966.7859
6	$0.9973Bvq \sin \theta + 0.0990\theta + 0.0084(E_fq + \theta)^2 \{\sin \theta + \sin(E_fq)\} + 0.0990 \sin(Bv/E_f) + 0.0211$	0.0086	-5032.7066
7	$1.0184Bvq \sin \theta - 0.0215E_fq\theta \{\theta + \sin(Bv/E_f)\} + 0.1353 \sin(Bv/E_f) + 0.0051$	0.0106	-5045.5640
8	$1.0134Bvq \sin \theta + 0.1554\theta + 0.1554 \sin \theta - 0.0688 \sin(2\theta) \sin\{\sin(E_fq)\} + 0.1554 \sin(E_fq) + 0.0071$	0.0101	-5048.4944
9	$0.0950BvE_fq^2 + \frac{0.0950Bv(E_fq + \theta)}{E_f} + 1.0087(Bvq + E_fq) \sin\{\sin \theta\} - 0.8010 \sin\{E_fq^2(Bv/E_f + \theta)\} - 0.0073$	0.0104	-5072.3435
10	$1.1751Bvq \sin\{\sin \theta\} - 0.1238\theta \{\sin(Bv/E_f) + \sin(E_fq)\} \sin(E_fq) + 0.1329 \sin(Bv/E_f) + 0.0995$	0.0128	-5201.4007

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

Table L.24: Top $r = 10$ ranked symbolic expressions recovered by a single run of **VaSST** under $\sigma = 0.1$ for learning **(FCE)**. Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	RMSE	LMPSE
1	$1.0663Bvq \sin(\theta) + 0.9792E_fq - 0.0022\theta + 0.0014$	0.1016	-3805.6060
2	$0.0370Bvq\theta + 0.8770Bvq \sin(\theta) - \frac{0.1549Bv\theta^2}{E_f} + 0.8770E_fq \sin(\theta) + 0.1919 \sin(\sin(E_fq)) + 0.0051$	0.1021	-4810.8994
3	$-\frac{0.0395Bv^2}{E_f^2} + 1.0512Bvq \sin(\theta) + 0.1954Bvq - \frac{0.0395Bv\theta^2}{E_f} + \frac{0.1163Bv\theta}{E_f} + \frac{1.0512Bv \sin(\frac{Bv}{E_f})}{E_f} + 0.0908\theta + 0.030$	0.1037	-4899.0665
4	$1.0390Bvq \sin(\theta) + 0.0983\theta + 1.6438 \sin(E_fq) + 0.0091$	0.1028	-4954.6377
5	$1.1036Bvq \sin(\theta) - \frac{0.3703Bv}{E_f} - 3.7934 \sin(\sin(\theta)) + 0.0001$	0.1045	-4967.3706
6	$0.9974Bvq \sin(\theta) + 0.0084E_f^2q^2 \sin(\theta) + 0.0084E_f^2q^2 \sin(E_fq) + 0.0168E_fq\theta \sin(\theta) + 0.0168E_fq\theta \sin(E_fq) + 0.0084\theta^2 \sin(\theta) + 0.0084\theta^2 \sin(E_fq) + 0.0992\theta + 0.0992 \sin(\frac{Bv}{E_f}) + 0.5121$	0.1074	-5033.1722
7	$1.0185Bvq \sin(\theta) - 0.0214E_fq\theta^2 - 0.0214E_fq\theta \sin(\frac{Bv}{E_f}) + 0.1274 \sin(\frac{Bv}{E_f}) + 0.1023$	0.1334	-5046.4801
8	$1.0135Bvq \sin(\theta) + 0.1558\theta + 0.1558 \sin(\theta) - 0.0744 \sin(2\theta) \sin(\sin(E_fq)) + 0.1558 \sin(E_fq) + 6.0358$	0.1903	-5048.9328

Continued on next page

Table L.24 continued from previous page.

Rank	VaSST Expression Learned	RMSE	LMPSE
9	$0.0950BvE_fq^2 + 1.0088Bvq \sin(\sin(\theta)) + 0.0950Bvq + \frac{0.0950Bv\theta}{E_f} + 1.0088E_fq \sin(\sin(\theta)) - 0.7980 \sin\left(E_fq \left(\frac{Bv}{E_f} + \theta\right)\right) - 1.1405$	0.2031	-5073.3466
10	$1.1752Bvq \sin(\sin(\theta)) - 0.1240\theta \sin\left(\frac{Bv}{E_f}\right) \sin(E_fq) - 0.1240\theta \sin^2(E_fq) + 0.1251 \sin\left(\frac{Bv}{E_f}\right) + 1.6583$	0.1949	-5201.4565

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

Table L.25: Top $r = 10$ ranked symbolic expressions recovered by a single run of VaSST under $\sigma = 0.2$ for learning (FCE). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	RMSE	LMPSE
1	$1.0664Bqv \sin \theta + 0.9788E_fq - 0.0018\theta + 0.0014$	0.2016	-3818.2221
2	$0.0370Bqv \sin \theta + 0.8771Bqv \sin \theta - \frac{0.1549Bv\theta^2}{E_f} + 0.8771E_fq \sin \theta + 0.1919 \sin\{\sin(E_fq)\} + 0.0021$	0.2034	-4816.9309
3	$-\frac{0.0395B^2v^2}{E_f^2} + 1.0513Bqv \sin \theta + 0.1956Bqv - \frac{0.0395Bv\theta^2}{E_f} + \frac{0.1166Bv\theta}{E_f} + \frac{1.0513Bv \sin(Bv/E_f)}{E_f} + \frac{0.1956Bv}{E_f} - 0.0790\theta^3 + 0.3912\theta + 0.0048$	0.2032	-4904.2802
4	$1.0391Bqv \sin \theta + 0.0987\theta + 1.6451 \sin(E_fq) + 0.0714$	0.2050	-4957.4885
5	$1.1037Bqv \sin \theta - \frac{0.3686Bv}{E_f} - 3.7954 \sin\{\sin \theta\} + 0.1075$	0.2049	-4970.4218
6	$0.9974Bqv \sin \theta + 0.0084E_f^2q^2 \sin \theta + 0.0084E_f^2q^2 \sin(E_fq) + 0.0168E_fq \sin \theta + 0.0168E_fq \sin(E_fq) + 0.0084\theta^2 \sin \theta + 0.0084\theta^2 \sin(E_fq) + 0.0994\theta + 0.0994 \sin(Bv/E_f) + 1.0122$	0.2293	-5036.0061
7	$1.0186Bqv \sin \theta - 0.0213E_fq\theta^2 - 0.0213E_fq \sin(Bv/E_f) + 0.1195 \sin(Bv/E_f) + 0.9636$	0.2531	-5049.6806
8	$1.0135Bqv \sin \theta + 0.1562\theta + 0.1562 \sin \theta - 0.0800 \sin(2\theta) \sin\{\sin(E_fq)\} + 0.1562 \sin(E_fq) + 1.0345$	0.2714	-5051.6859
9	$0.0951BE_fq^2v + 1.0089Bqv \sin\{\sin \theta\} + 0.0951Bqv + \frac{0.0951Bv\theta}{E_f} + 1.0089E_fq \sin\{\sin \theta\} - 0.7951 \sin(BE_fq^2v + E_f^2q^2\theta) - 1.0005$	0.2612	-5076.6401
10	$1.1753Bqv \sin\{\sin \theta\} - 0.1241\theta \sin^2(E_fq) - 0.1241\theta \sin(E_fq) \sin(Bv/E_f) + 0.1174 \sin(Bv/E_f) + 1.9158$	0.3121	-5203.5019

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

For learning (FTC): $P = \frac{\kappa A(T_2 - T_1)}{d}$ in Section 5.2.

Table L.26: Top $r = 10$ ranked symbolic expressions recovered by a single run of VaSST under the noiseless setting for learning (FTC). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	RMSE	LMPSE
1	$-0.805371 \frac{AT_1\kappa}{d} + 0.998520 \frac{AT_2}{d} - 0.000254 \left\{ \frac{1}{\sqrt{A}T_1\kappa} - \frac{d/\sqrt{A}-T_2/T_1}{\sqrt{A}T_1\kappa-T_2/T_1} \right\} + 0.014782$	0.0126	3205.1951
2	$-0.0484\sqrt{A}T_1\kappa - \frac{0.8506\sqrt{A}(-\sqrt{A}T_2\kappa+d/\sqrt{A})}{d} + 0.5893 + \frac{0.0508T_2}{T_1}$	0.0207	3125.1013
3	$\frac{2.0449A^{3/2}T_2\kappa}{d^2} - 0.2527\sqrt{A}T_1\kappa + 0.1263\sqrt{A}T_2\kappa - 0.1748 + \frac{0.0409d}{\sqrt{A}}$	0.0173	3111.3952
4	$0.0032AT_1^2\kappa^2(-\sqrt{A}T_1\kappa+T_2/T_1) + \frac{0.7742AT_2\kappa}{d} - 0.3208 + \frac{0.0679T_2}{T_1}$	0.0224	3097.3072
5	$0.8039\sqrt{A}T_1\kappa - \frac{0.8675AT_1\kappa(-T_2/T_1+d/\sqrt{A})}{d} - \frac{0.7636T_1}{T_2} + 0.1411$	0.0226	3078.9767
6	$0.0047\sqrt{A}T_1\kappa + \frac{0.0047\sqrt{A}T_2}{T_1d} - 0.0047AT_1^2\kappa^2 + \frac{0.8076AT_2\kappa}{d} - 0.3331 + \frac{0.0645T_2}{T_1}$	0.0237	3075.3877
7	$-0.0045AT_1^2\kappa^2 - \frac{0.8146AT_2\kappa(-T_2/T_1+d/\sqrt{A})}{d(T_2/T_1-d/\sqrt{A})} - 0.3277 + \frac{0.0637T_2}{T_1}$	0.0241	3055.1463
8	$\frac{0.8197AT_2\kappa}{d} - 0.0010T_2d\kappa - 0.3395 + \frac{0.0679T_2}{T_1} + \frac{0.0010d}{\sqrt{A}}$	0.0241	3050.3108
9	$-0.0005AT_1T_2\kappa^2 + \frac{0.8000AT_2\kappa}{d} - 0.3437 + \frac{0.0699T_2}{T_1} + \frac{0.0005d}{\sqrt{A}}$	0.0248	3049.2081
10	$0.0034\sqrt{A}T_1\kappa - \frac{0.7927\sqrt{A}T_2(-\sqrt{A}T_1\kappa+d/\sqrt{A})}{T_1d} - 0.3544 + \frac{0.8633T_2}{T_1} - \frac{0.0034(\sqrt{A}T_1\kappa-d/\sqrt{A})}{\sqrt{A}T_1\kappa}$	0.0248	3029.5726

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

Table L.27: Top $r = 10$ ranked symbolic expressions recovered by a single run of VaSST under $\sigma = 0.1$ for learning (FTC). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	RMSE	LMPSE
1	$-0.0658\sqrt{A}T_1\kappa + \frac{0.8713AT_2\kappa}{d} - 0.2464 + \frac{0.0466T_2}{T_1}$	0.1013	1519.3407
2	$-0.0790\sqrt{A}T_1\kappa + \frac{0.8853AT_2\kappa}{d} - \frac{0.7128T_1}{T_2} + 0.1266$	0.1015	1519.0129

Continued on next page

Table L.27 continued from previous page.

Rank	VaSST Expression Learned	RMSE	LMPSE
3	$\frac{2.0411A^{3/2}T_2\kappa}{d^2} - 0.2565\sqrt{A}T_1\kappa + 0.1283\sqrt{A}T_2\kappa - 0.1242 + \frac{0.0285d}{\sqrt{A}}$	0.1012	1508.1725
4	$-0.0041A^{3/2}T_1^3\kappa^3 + 0.0041AT_1T_2\kappa^2 + \frac{0.7700AT_2\kappa}{d} - 0.3306 + \frac{0.0702T_2}{T_1}$	0.1023	1499.4539
5	$0.0076\sqrt{A}T_1\kappa + \frac{0.0076\sqrt{A}T_2}{T_1d} - 0.0076AT_1^2\kappa^2 + \frac{0.8186AT_2\kappa}{d} - 0.3429 + \frac{0.0633T_2}{T_1}$	0.1024	1499.0290
6	$-\frac{0.8347\sqrt{A}T_2\kappa}{-T_2/T_1+T_2d/(\sqrt{A}T_1)} + \frac{1.0066AT_2\kappa}{d} - \frac{0.0001T_2}{\sqrt{A}T_1^2\kappa-T_2} + \frac{0.0001d}{AT_1\kappa-\sqrt{A}T_2/T_1} + 0.0148$	0.1013	1496.6708
7	$\frac{0.8381AT_2\kappa}{d} - 0.0017T_2d\kappa - 0.3522 + \frac{0.0688T_2}{T_1} + \frac{0.0017d}{\sqrt{A}}$	0.1024	1495.2980
8	$-0.0071AT_1^2\kappa^2 + \frac{0.8274AT_2\kappa}{d} - 0.3337 + \frac{0.0625T_2}{T_1}$	0.1025	1486.1853
9	$-0.0012AT_1T_2\kappa^2 + \frac{0.8147AT_2\kappa}{d} - 0.3606 + \frac{0.0700T_2}{T_1} + \frac{0.0012d}{\sqrt{A}}$	0.1030	1485.0374
10	$-0.0837\sqrt{A}T_1\kappa + \frac{0.0837\sqrt{A}T_2}{T_1d} - \frac{0.0837AT_1\kappa}{d} + \frac{0.9207AT_2\kappa}{d} - 0.1414$	0.1024	1478.0320

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

Table L.28: Top $r = 10$ ranked symbolic expressions recovered by a single run of VaSST under $\sigma = 0.2$ for learning (FTC). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	RMSE	LMPSE
1	$-0.0944\sqrt{A}T_1\kappa + \frac{0.9031AT_2\kappa}{d} - \frac{0.6620T_1}{T_2} + 0.1121$	0.1998	296.1263
2	$-0.0832\sqrt{A}T_1\kappa + \frac{0.8921AT_2\kappa}{d} - 0.2315 + \frac{0.0424T_2}{T_1}$	0.1998	290.2211
3	$\frac{2.0374A^{3/2}T_2\kappa}{d^2} - 0.2604\sqrt{A}T_1\kappa + 0.1302\sqrt{A}T_2\kappa - 0.0737 + \frac{0.0160d}{\sqrt{A}}$	0.2000	285.3669
4	$0.0105\sqrt{A}T_1\kappa + \frac{0.0105\sqrt{A}T_2}{T_1d} - 0.0105AT_1^2\kappa^2 + \frac{0.8296AT_2\kappa}{d} - 0.3527 + \frac{0.0620T_2}{T_1}$	0.2007	279.9495
5	$\frac{0.8566AT_2\kappa}{d} - 0.0024T_2d\kappa - 0.3648 + \frac{0.0697T_2}{T_1} + \frac{0.0024d}{\sqrt{A}}$	0.2005	279.1597
6	$0.9642\left(\frac{AT_2\kappa}{d} - \sqrt{A}T_1\kappa\right) + 0.8266\sqrt{A}T_1\kappa - 0.0003\left\{\frac{\sqrt{A}T_1\kappa}{\sqrt{A}T_1\kappa - \sqrt{A}T_1\kappa}\right\}$	0.2009	277.5255
7	$-0.0050A^{3/2}T_1^3\kappa^3 + 0.0050AT_1T_2\kappa^2 + \frac{0.7657AT_2\kappa}{d} - 0.3404 + \frac{0.0726T_2}{T_1}$	0.2007	276.9519
8	$-0.0019AT_1T_2\kappa^2 + \frac{0.8293AT_2\kappa}{d} - 0.3775 + \frac{0.0701T_2}{T_1} + \frac{0.0019d}{\sqrt{A}}$	0.2011	270.7425
9	$-0.0096AT_1^2\kappa^2 + \frac{0.8401AT_2\kappa}{d} - 0.3396 + \frac{0.0612T_2}{T_1}$	0.2007	267.8984
10	$-0.0911\sqrt{A}T_1\kappa + \frac{0.0911\sqrt{A}T_2}{T_1d} - \frac{0.0911AT_1\kappa}{d} + \frac{0.9303AT_2\kappa}{d} - 0.1489$	0.2001	263.8642

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

For learning (ADA): $f = \beta^\dagger(1 + \alpha^\dagger \cos \theta)$ in Section 5.2.

Table L.29: Top $r = 10$ ranked symbolic expressions recovered by a single run of VaSST under the noiseless setting for learning (ADA). Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	RMSE	LMPSE
1	$1.0238\alpha^\dagger\beta^\dagger \cos \theta + 1.0238\beta^\dagger + 0.00103\theta + 0.0012\cos(2\alpha^\dagger) - 0.0009$	0.0013	1000.8680
2	$0.0975\alpha^\dagger + 0.0505\beta^\dagger(\beta^\dagger + 2\theta) + 0.0325\theta + 0.8895(\alpha^\dagger\beta^\dagger + \beta^\dagger)\cos \theta + 0.0325\cos(\beta^\dagger) - 0.5612$	0.5480	-1720.7743
3	$0.0440\alpha^\dagger\beta^\dagger(\beta^\dagger + \theta) - 0.1722\beta^\dagger \cos(\alpha^\dagger) + 1.0081\{\alpha^\dagger\beta^\dagger + \cos(\alpha^\dagger)\}\cos \theta + 0.0440\cos(\beta^\dagger + \theta) - 1.6680$	0.7250	-2283.3134
4	$0.8978\alpha^\dagger\beta^\dagger \cos \theta + 0.0826\alpha^\dagger + 0.0476\beta^\dagger\theta^2 + 0.8978\beta^\dagger \cos \theta + 0.0476\theta - 2.9743$	0.7862	-2428.7997
5	$1.0678\alpha^\dagger\beta^\dagger \cos \theta + 0.1879\alpha^\dagger\theta - 1.1075\alpha^\dagger + 0.5636\beta^\dagger - 0.5537\cos(\beta^\dagger) - 0.0448$	0.8996	-2700.6837
6	$1.0092\alpha^\dagger\beta^\dagger \cos \theta + 0.2084\cos(\alpha^\dagger) + 0.2084\cos(2\alpha^\dagger) + 0.2084\cos(\beta^\dagger) - 0.0643\cos(2\alpha^\dagger\beta^\dagger\theta) + 3.4579$	1.1653	-3228.7870
7	$1.0434\alpha^\dagger\beta^\dagger \cos \theta - 0.1611\beta^\dagger \cos \theta - 0.0155\cos\{(\alpha^\dagger)^2\} - 0.1611\cos(2\beta^\dagger) + 3.4235$	1.1933	-3266.0467
8	$1.1812\beta^\dagger + 4.3937(\alpha^\dagger + \beta^\dagger)\cos \theta - 20.0205\cos \theta + 4.3937\cos(\cos \theta) - 3.9709$	1.7949	-4090.8640
9	$-0.1459\alpha^\dagger + 0.8872\beta^\dagger\{\alpha^\dagger + \cos(\alpha^\dagger)\}\cos \theta - 0.0613\theta^2 - 0.0613\theta - 0.0613\cos\{\cos(\alpha^\dagger)\} + 8.1100$	1.8462	-4140.0795
10	$0.5482\beta^\dagger(\alpha^\dagger + \beta^\dagger)\cos \theta + 0.8815\beta^\dagger \cos(\cos \theta) - 1.0354\theta - 1.0354\cos(\cos \theta) + 9.9864$	1.9516	-4253.3533

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

Table L.30: Top $r = 10$ ranked symbolic expressions recovered by a single run of **VaSST** under $\sigma = 0.1$ for learning **(ADA)**. Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	RMSE	LMPSE
1	$1.0241\alpha^\dagger\beta^\dagger\cos\theta + 1.0241\beta^\dagger + 0.0001\theta + 0.0003\cos(2\alpha^\dagger) - 0.0001$	0.1083	596.2956
2	$0.8896\alpha^\dagger\beta^\dagger\cos\theta + 0.0999\alpha^\dagger + 0.0504(\beta^\dagger)^2 + 0.1009\beta^\dagger\theta + 0.8896\beta^\dagger\cos\theta + 0.0333\theta + 0.0333\cos(\beta^\dagger) - 0.5769$	0.5599	-1763.2871
3	$0.0440\alpha^\dagger(\beta^\dagger)^2 + 0.0440\alpha^\dagger\beta^\dagger\theta + 1.0081\alpha^\dagger\beta^\dagger\cos\theta - 0.1715\beta^\dagger\cos(\alpha^\dagger) + 1.0081\cos(\alpha^\dagger)\cos\theta + 0.0440\cos(\beta^\dagger + \theta) - 1.6715$	0.7327	-2297.4057
4	$0.8978\alpha^\dagger\beta^\dagger\cos\theta + 0.0847\alpha^\dagger + 0.0476\beta^\dagger\theta^2 + 0.8978\beta^\dagger\cos\theta + 0.0476\theta - 2.9871$	0.7872	-2433.3839
5	$1.0679\alpha^\dagger\beta^\dagger\cos\theta + 0.1878\alpha^\dagger\theta - 1.1050\alpha^\dagger + 0.5634\beta^\dagger - 0.5525\cos(\beta^\dagger) - 0.0551$	0.9001	-2704.0075
6	$1.0092\alpha^\dagger\beta^\dagger\cos\theta + 0.2101\cos(\alpha^\dagger) + 0.2101\cos(2\alpha^\dagger) + 0.2101\cos(\beta^\dagger) - 0.0647\cos(2\alpha^\dagger\beta^\dagger\theta) + 3.4477$	1.1641	-3226.8548
7	$1.0438\alpha^\dagger\beta^\dagger\cos\theta - 0.1627\beta^\dagger\cos\theta - 0.0180\cos\{(\alpha^\dagger)^2\} - 0.1627\cos(2\beta^\dagger) + 3.4140$	1.1915	-3263.6318
8	$4.3980\alpha^\dagger\cos\theta + 4.3980\beta^\dagger\cos\theta + 1.1795\beta^\dagger - 20.0600\cos\theta + 4.3980\cos(\cos\theta) - 3.9615$	1.7949	-4091.1167
9	$0.8875\alpha^\dagger\beta^\dagger\cos\theta - 0.1441\alpha^\dagger + 0.8875\beta^\dagger\cos(\alpha^\dagger)\cos\theta - 0.0610\theta^2 - 0.0610\theta - 0.0610\cos\{\cos(\alpha^\dagger)\} + 8.0903$	1.8461	-4140.0206
10	$0.5484\alpha^\dagger\beta^\dagger\cos\theta + 0.5484(\beta^\dagger)^2\cos\theta + 0.8827\beta^\dagger\cos(\cos\theta) - 1.0310\theta - 1.0310\cos(\cos\theta) + 9.9696$	1.9507	-4252.5163

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

Table L.31: Top $r = 10$ ranked symbolic expressions recovered by a single run of **VaSST** under $\sigma = 0.2$ for learning **(ADA)**. Expressions are ranked by LMPSE among $H = 2000$ sampled hard symbolic trees.

Rank	VaSST Expression Learned	RMSE	LMPSE
1	$1.0244\alpha^\dagger\beta^\dagger\cos\theta + 1.0244\beta^\dagger + 0.0011\theta + 0.0005\cos(2\alpha^\dagger) - 0.0009$	0.2004	-73.3202
2	$0.8896\alpha^\dagger\beta^\dagger\cos\theta + 0.1024\alpha^\dagger + 0.0504(\beta^\dagger)^2 + 0.1007\beta^\dagger\theta + 0.8896\beta^\dagger\cos\theta + 0.0341\theta + 0.0341\cos(\beta^\dagger) - 0.5925$	0.5797	-1871.4716
3	$0.0440\alpha^\dagger(\beta^\dagger)^2 + 0.0440\alpha^\dagger\beta^\dagger\theta + 1.0082\alpha^\dagger\beta^\dagger\cos\theta - 0.1708\beta^\dagger\cos(\alpha^\dagger) + 1.0082\cos(\alpha^\dagger)\cos\theta + 0.0440\cos(\beta^\dagger + \theta) - 1.6750$	0.7474	-2343.8105
4	$0.8979\alpha^\dagger\beta^\dagger\cos\theta + 0.0867\alpha^\dagger + 0.0477\beta^\dagger\theta^2 + 0.8979\beta^\dagger\cos\theta + 0.0477\theta - 2.9999$	0.7891	-2443.9352
5	$1.0679\alpha^\dagger\beta^\dagger\cos\theta + 0.1877\alpha^\dagger\theta - 1.1024\alpha^\dagger + 0.5631\beta^\dagger - 0.5512\cos(\beta^\dagger) - 0.0654$	0.9011	-2711.0094
6	$1.0093\alpha^\dagger\beta^\dagger\cos\theta + 0.2119\cos(\alpha^\dagger) + 0.2119\cos(2\alpha^\dagger) + 0.2119\cos(\beta^\dagger) - 0.0523\cos(2\alpha^\dagger\beta^\dagger\theta) + 3.4568$	1.1614	-3217.6691
7	$1.0441\alpha^\dagger\beta^\dagger\cos\theta - 0.1643\beta^\dagger\cos\theta - 0.0206\cos\{(\alpha^\dagger)^2\} - 0.1643\cos(2\beta^\dagger) + 3.4217$	1.1877	-3257.2784
8	$4.4023\alpha^\dagger\cos\theta + 4.4023\beta^\dagger\cos\theta + 1.1778\beta^\dagger - 20.1000\cos\theta + 4.4023\cos(\cos\theta) - 3.9658$	1.7950	-4092.0921
9	$0.8878\alpha^\dagger\beta^\dagger\cos\theta - 0.1423\alpha^\dagger + 0.8878\beta^\dagger\cos(\alpha^\dagger)\cos\theta - 0.0607\theta^2 - 0.0607\theta - 0.0607\cos\{\cos(\alpha^\dagger)\} + 8.0502$	1.8460	-4139.9095
10	$0.5487\alpha^\dagger\beta^\dagger\cos\theta + 0.5487(\beta^\dagger)^2\cos\theta + 0.8838\beta^\dagger\cos(\cos\theta) - 1.0266\theta - 1.0266\cos(\cos\theta) + 9.9047$	1.9489	-4249.8368

Note: RMSE corresponds to the out-of-sample RMSE computed on a 10% held-out test set with the LMPSE selected symbolic model.

M DISTINCTION BETWEEN VaSST AND VaRT

Although **VaSST** and **VaRT** [Salazar, 2023] both use variational inference over relaxed tree-structured objects, the underlying statistical objects, computational relaxations, and inferential targets are fundamentally different. **VaRT** is designed for variational inference in decision trees, which consist of recursive partitions of the covariate space followed by prediction through routing of observations to terminal regions. In contrast, **VaSST** addresses **SR**, where trees encode compositional mathematical expressions; see [Section 2](#). This distinction changes the meaning of every node of the underlying tree structures, the form of the forward evaluation, the nature of the posterior distribution, and the role of uncertainty quantification, as summarized below.

First, as noted in [Sections 1 and 2](#), **VaSST** is based on symbolic tree representations. Internal nodes in a symbolic tree represent mathematical operators such as $+$, $-$, $\times$, $/$, $\sin$, $\cos$, $\log$, $\exp$, and powers, while terminal nodes represent primitive features. Thus, a symbolic tree evaluates to an analytical expression. This is fundamentally different from a decision tree, where internal nodes represent split rules and the tree defines a partition of the covariate space rather than a closed-form scientific law.

Second, **VaSST** requires a new *soft symbolic evaluation* mechanism; see [SOFTEVAL Algorithm 2](#) of [Appendix D](#). In **VaRT**, relaxation is achieved through soft routing, where an observation is probabilistically routed through a decision tree. In **VaSST**, there is no routing interpretation. Instead, the relaxation must propagate mixtures over operator and feature choices through a compositional expression tree. The resulting soft tree is therefore a differentiable surrogate for symbolic expression evaluation, not a probabilistic partitioning rule.

Third, **VaSST** uses reparameterized relaxations tailored to symbolic structures. In [\(13\)](#), Binary-Concrete variables are used for expansion decisions, while Gumbel-Softmax relaxations are used for operator and feature assignments. These relaxations must respect the compositional nature of **SR**, where invalid or numerically unstable operations, such as division by near-zero quantities or undefined logarithmic evaluations, require careful treatment. Such issues do not arise in standard soft decision tree routing.

Fourth, **VaSST** incorporates a collapsed Bayesian linear layer that is absent in the **VaRT** modeling framework. Specifically, **VaSST** places a conjugate Normal Inverse-Gamma prior on the outer regression coefficients and noise variance, and analytically marginalizes these quantities. This yields a collapsed structural objective over symbolic ensembles, allowing the variational optimization to focus on expression structure while preserving uncertainty over the regression layer. This collapsed formulation is central to the underlying Bayesian model of **VaSST**.

Finally, the posterior object in **VaSST** is a distribution over symbolic expressions, not a distribution over predictive partitions. This enables posterior-aware sampling, ranking, and selection of candidate scientific laws after optimization. Consequently, uncertainty in **VaSST** is uncertainty over competing analytical explanations, whereas uncertainty in **VaRT** concerns alternative tree partitions and their associated predictions. Thus, **VaSST** is not a direct application of **VaRT** to **SR**; it introduces a distinct probabilistic framework for differentiable, uncertainty-aware discovery of compositional scientific expressions.

N ABLATION STUDIES FOR VaSST

Using ablation studies, we investigate the sensitivity of **VaSST** with respect to the number of symbolic trees K and the maximum tree depth D . These two architectural hyperparameters control complementary aspects of the symbolic representation. The number of trees K determines the number of additive symbolic components in the final ensemble, whereas the depth D controls the maximum compositional complexity allowed within each symbolic tree.

We consider two representative Feynman equations [Udrescu and Tegmark, 2020] from Section 5.2:

$$\text{(CPE)} : \Delta U = Gm_1m_2 \left(\frac{1}{r_2} - \frac{1}{r_1} \right), \quad \text{(FTC)} : P = \frac{\kappa A(T_2 - T_1)}{d}.$$

For each equation in the preceding display, $n = 2000$ observations are subsampled from 10^5 observations, and Gaussian noise with variance $\sigma^2 = 0.1^2$ is added to the response. We then run **VaSST** over the grid $K \in \{2, 3, 4, 5\}$ and $D \in \{2, 3, 4, 5\}$, while keeping all other prior, relaxation, and optimization settings configured to be the same as in Appendix I.2. The operator sets used for **(CPE)** and **(FTC)** are $\mathbf{O} = \{+, \times, /\}$ and $\mathbf{O} = \{\times, -, /\}$, respectively.

Table N.1: Runtime and out-of-sample RMSE (computed on a 10% held-out test set) of **VaSST** for **(CPE)** : $\Delta U = Gm_1m_2 \left(\frac{1}{r_2} - \frac{1}{r_1} \right)$ across different choices of K and D .

Runtime					RMSE				
$K \backslash D$	2	3	4	5	$K \backslash D$	2	3	4	5
2	54.7	98.8	192.4	399.6	2	0.2330	0.1012	0.1959	0.2349
3	61.5	112.0	221.4	491.3	3	0.1002	0.1019	0.1051	0.1918
4	67.9	123.1	251.8	600.9	4	0.1168	0.1002	0.1005	0.1012
5	72.3	134.8	286.4	800.5	5	0.1002	0.1005	0.1004	0.1010

Table N.2: Runtime and out-of-sample RMSE (computed on a 10% held-out test set) of **VaSST** for **(FTC)** : $P = \frac{\kappa A(T_2 - T_1)}{d}$ across different choices of K and D .

Runtime					RMSE				
$K \backslash D$	2	3	4	5	$K \backslash D$	2	3	4	5
2	50.6	97.2	191.3	401.6	2	0.1019	0.1002	0.1002	0.1070
3	62.1	112.0	221.8	498.5	3	0.1000	0.1011	0.1012	0.1019
4	72.2	125.0	249.8	628.2	4	0.1000	0.0999	0.1000	0.1000
5	83.1	146.0	299.4	803.4	5	0.0999	0.0999	0.0998	0.1007

Tables N.1 and N.2 report the runtimes and out-of-sample RMSEs of **VaSST** across the chosen (K, D) grid while learning **(CPE)** and **(FTC)**. The key findings are summarized as follows.

Performance stabilizes with model complexity. For both the Feynman equations, out-of-sample RMSE improves significantly and tracks the noise level $\sigma^2 = 0.1^2$ closely when moving from $K = 2$ to $K \geq 3$, and stabilizes for $K \geq 4$ and $D \geq 3$. This indicates that **VaSST** does not require large model sizes to achieve optimal predictive accuracy.

Low capacity regimes can limit symbolic recovery. As expected, insufficient model capacity leads to occasional failures in recovering the true symbolic law. For **(FTC)**, failure occurs at $(K = 2, D = 5)$, and for **(CPE)**, failure occurs at $(K = 2, D \in \{2, 4, 5\})$ and $(K = 3, D = 5)$. This highlights that both sufficient number of trees K and appropriate depth D are necessary for capturing the underlying compositional symbolic structure.

Runtime behavior with K and D . For fixed depth D , runtime increases approximately linearly in K and for fixed K , runtime increases much more rapidly with D , reflecting the exponential growth in the number of nodes in a symbolic tree. Importantly, moderate configurations, e.g., $(K = 3, D = 3)$, already achieve near-optimal RMSE value, offering a favorable accuracy-efficiency trade-off.

Principled heuristics for choosing K and D . Based on the above ablation studies, we prescribe the following practical guideline for choosing K and D . Since [VaSST](#) is designed to recover simple and interpretable scientific expressions, our recommendation follows an Occam’s razor principle [[Jefferys and Berger, 1992](#)]: choose the smallest model capacity (K, D) that attains stable out-of-sample RMSE and symbolic recovery. Empirically, $K \in \{3, 4\}$ and $D \in \{3, 4\}$ provide a favorable accuracy-interpretability-runtime trade-off in the applications considered in [Section 5](#), while larger depths increase runtime exponentially and may lead to unnecessarily complex symbolic expressions without substantial predictive gains.

O COMPUTATIONAL SCALING AND GRADIENT VARIANCE DIAGNOSTICS FOR VaSST

In this section, we examine how the computational footprint and stochastic gradient behavior of VaSST scale with the maximum symbolic tree depth D . This diagnostic complements the ablation studies in Appendix N by isolating the effect of depth on memory, runtime, and gradient variability in the relaxed symbolic tree parameterization.

O.1 SCALING OF THE RELAXED SYMBOLIC TREE PARAMETERIZATION

A full binary tree skeleton S_j of maximum depth D contains $N = 2^{D+1} - 1$ nodes. For each soft symbolic tree $\{S_j^{\text{soft}}\}_{j=1}^K$, VaSST maintains relaxed logits for node expansion ($\ell_{j\zeta}$), operator assignment ($\mathbf{a}_{j\zeta}^{\text{op}}$), and feature assignment ($\mathbf{a}_{j\zeta}^{\text{ft}}$) for every node $\zeta \in Z_D = \{0, 1, \dots, N-1\}$, along with the variational concentration hyperparameters $\tilde{\boldsymbol{\eta}}_{\text{op}}$ and $\tilde{\boldsymbol{\eta}}_{\text{ft}}$. Therefore, if $|\mathbf{O}|$ denotes the number of operators and p denotes the input dimension, then the total number of variational parameters in $\phi = ((\ell_{j\zeta}, \mathbf{a}_{j\zeta}^{\text{op}}, \mathbf{a}_{j\zeta}^{\text{ft}})_{j=1, \dots, K; \zeta \in Z_D}, \tilde{\boldsymbol{\eta}}_{\text{op}}, \tilde{\boldsymbol{\eta}}_{\text{ft}})$ is $d_{\text{relax}}(K, D) = |\mathbf{O}| + p + KN(1 + |\mathbf{O}| + p)$. Thus, for fixed $|\mathbf{O}|$ and p , the number of variational parameters grows as $d_{\text{relax}}(K, D) = O(K2^D)$.

This scaling gives the dominant depth-dependent memory footprint for the structural relaxation. In addition, during Monte Carlo (MC) ELBO evaluation in the APPROXELBO Algorithm 3 in Appendix E, the soft tree forward pass stores node-level evaluations and stochastic relaxation samples. With S MC samples and n training observations, the activation-level memory is of order $O(SnKN)$ up to operator-specific constants. Hence, the memory footprint grows approximately linearly in K , S , and n , but exponentially in D .

The same node count scaling also determines runtime. A stochastic forward-backward pass evaluates all relaxed nodes and backpropagates through the Binary-Concrete and Gumbel-Softmax relaxations. Therefore, the per-iteration computational cost is approximately $O(SnKNC_{\mathbf{O}})$, where $C_{\mathbf{O}}$ denotes the average cost of evaluating the operator mixture. Since $N = 2^{D+1} - 1$, the runtime is expected to increase exponentially with depth D .

O.2 GRADIENT VARIANCE DIAGNOSTIC

We next quantify how the stochastic gradient variability changes with depth D . For one draw of the relaxed tree variables, the stochastic gradient of the negative ELBO is $g(\phi; \omega) = \nabla_{\phi} \{-\hat{\mathcal{E}}(\phi; \omega)\}$, where ω encodes the randomness owing to the Binary-Concrete and Gumbel-Softmax relaxations. We estimate the coordinate-wise gradient variance using R independent stochastic gradients $g^{(1)}, \dots, g^{(R)}$:

$$\hat{\mathbb{V}}_j(g) = \frac{1}{R-1} \sum_{r=1}^R (g_j^{(r)} - \bar{g}_j)^2, \quad \bar{g}_j = \frac{1}{R} \sum_{r=1}^R g_j^{(r)}. \quad (\text{O.1})$$

We summarize the gradient variability in (O.1) using:

$$\hat{V}_{\text{coord}} = \frac{1}{d_{\text{relax}}} \sum_{j=1}^{d_{\text{relax}}} \hat{\mathbb{V}}_j(g), \quad \hat{V}_{\text{total}} = \sum_{j=1}^{d_{\text{relax}}} \hat{\mathbb{V}}_j(g), \quad (\text{O.2})$$

which are the mean coordinate-wise gradient variance and the total gradient variance across all relaxed structural logits, respectively. The first quantity in (O.2) measures the average stochastic fluctuation per relaxed coordinate, while the second quantity in (O.2) measures the aggregate stochastic fluctuation over the full relaxed symbolic tree parameterization.

O.3 RESULTS

We run the diagnostic experiment on the Feynman equation (CPE): $\Delta U = Gm_1m_2(\frac{1}{r_2} - \frac{1}{r_1})$ from Section 5.2 for which VaSST is implemented with $K = 3$ symbolic trees and depths $D \in \{2, 3, 4, 5\}$. The diagnostic experiment is run on CPU with $n = 2000$ subsampled training observations and Gaussian noise level $\sigma^2 = 0.2^2$ added to the response. The operator set is configured as $\mathbf{O} = \{+, \times, /\}$. All remaining settings follow similarly from Appendix I.2. In this experiment, $K = 3$, $|\mathbf{O}| = 3$, and $p = 3$. Therefore, the size of the relaxed structural parameterization grows as shown in Table O.1.

Table O.1: Growth of the relaxed symbolic parameterization for $K = 3$, $|\mathbf{O}| = 3$, and $p = 3$.

Depth D	Nodes per tree (N)	Total nodes (KN)	Variational Parameters ($d_{\text{relax}}(K, D)$)
2	7	21	153
3	15	45	321
4	31	93	657
5	63	189	1329

For each depth D , we measure the runtime of one stochastic forward-backward pass using $S = 8$ MC samples in the ELBO. For the gradient variance diagnostic, we generate $R = 100$ independent stochastic gradients using one MC sample per gradient. Gradients are collected only with respect ϕ . Each diagnostic is repeated 5 times, and the median value is reported. We compute these quantities before training and after 2000 training iterations.

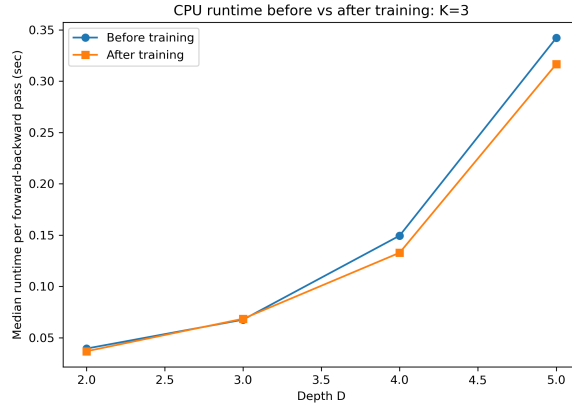
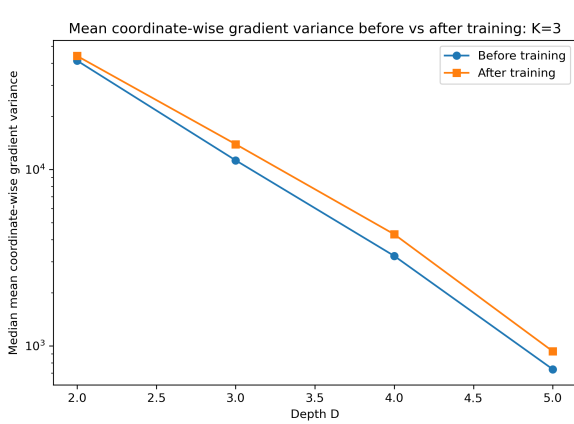
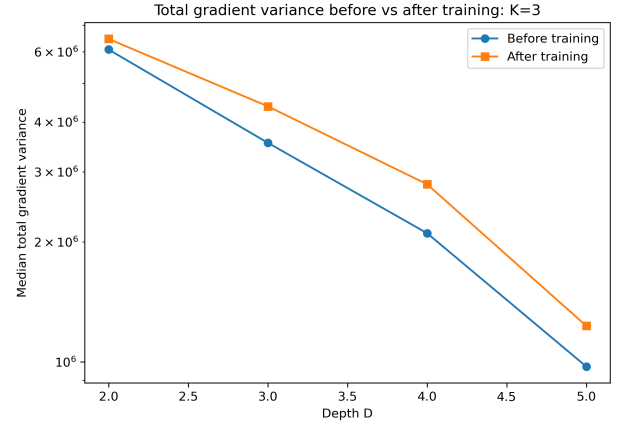


Figure O.1: Median (over 5 repetitions) CPU runtime per forward-backward pass against depth D , before and after training for (CPE): $\Delta U = Gm_1m_2(\frac{1}{r_2} - \frac{1}{r_1})$ with $K = 3$ and $\sigma^2 = 0.2^2$.



(a) Median (over 5 repetitions) of $\hat{V}_{\text{coord}}$ in (O.2).



(b) Median (over 5 repetitions) of $\hat{V}_{\text{total}}$ in (O.2).

Figure O.2: Gradient variance diagnostic against depth D , before and after training for $\Delta U = Gm_1m_2(\frac{1}{r_2} - \frac{1}{r_1})$ with $K = 3$ and $\sigma^2 = 0.2^2$.

The runtime behavior in Figure O.1 confirm the theoretical scaling as noted in Table O.1. The median CPU time per forward-backward pass increases monotonically with depth D . This growth is consistent with the expansion of the full binary tree skeleton S_j from $N = 2^{2+1} - 1 = 7$ to $N = 2^{5+1} - 1 = 63$ nodes per tree. The before-training and after-training curves are nearly identical, indicating that computational cost is governed primarily by the size of the relaxed tree skeleton rather than by the learned parameter values.

The mean coordinate-wise gradient variance ($\hat{V}_{\text{coord}}$) in Figure O.2a provide a useful clarification. Increasing the depth D does not lead to an explosion of the gradient variance. Instead, $\hat{V}_{\text{coord}}$ decreases as D increases. This behavior is consistent with the depth-dependent split prior in Section 3.2: deeper nodes have smaller probability of being active, therefore many deep expansion, operator, and feature logits receive weaker effective gradients and contribute smaller coordinate-wise fluctuations. In other words, although increasing D adds multiple relaxed coordinates, many of these coordinates correspond to deeper nodes that are appropriately penalized for the learning of simple and interpretable scientific expressions [Jefferys and Berger, 1992]. The total gradient variance ($\hat{V}_{\text{total}}$) in Figure O.2b also decreases over the range of depths D . After training, $\hat{V}_{\text{total}}$ is slightly larger than before training, but the same decreasing trend with D is preserved. Thus, for this Feynman equation (CPE), increasing depth D mainly increases the deterministic computation and memory burden through the exponential growth of the full relaxed tree skeleton; it does not destabilize optimization through exploding stochastic gradient variance.

These diagnostics further justify the use of moderate depth values in the applications considered in Section 5. Even though larger D expands the symbolic expression search space and allows for more complex structures, the memory and runtime scale as $O(K2^D)$. Moreover, the ablation studies performed in Appendix N shows that $D = 3$ already suffices to recover the relevant scientific structures for the Feynman equations viz., (CPE) and (FTC). Therefore, an appropriate choice of (K, D) (which is $(K = 3, D = 3)$ in our case) offers a principled accuracy-interpretability-efficiency trade-off, while larger depths can be used when the underlying scientific problem requires more compositional complexity.

P ADDITIONAL EXPERIMENTS WITH LARGE LANGUAGE MODELS

For comparing **VaSST** with more recent machine learning-centric SR modules, we conduct additional experiments with LLM-SR [Shojaee et al., 2025], which casts SR as a large language model (LLM)-guided program synthesis task, iteratively generating and evaluating executable symbolic expressions from data. We consider the Feynman equation (**ADA**): $f = \beta^\dagger(1 + \alpha^\dagger \cos \theta)$ from Section 5.2 with $n = 2000$ subsampled training observations under additive Gaussian noise level $\sigma^2 = 0.2^2$.

LLM-SR configurations and prompts. Two open-source language models were used as the LLM backend: TinyLlama-1.1B-Chat [Zhang et al., 2024] and Qwen2.5-Coder-7B-Instruct [Hui et al., 2024], configured as in Table P.1. Candidate expressions are scored by negative mean squared error after fitting the free continuous parameters using BFGS [Nocedal and Wright, 2006]. We execute three prompting regimes of increasing informativeness, as summarized below, ranging from weak syntactic guidance to strongly structured scientific prompting.

Table P.1: LLM-SR configurations for learning (**ADA**) under $\sigma = 0.2$.

LLM	Prompt	Initialization	MAX_NPARAMS	BFGS maxiter	Compute Resources
TinyLlama-1.1B-Chat	Prompt 1	$c_0\beta^\dagger$	3	100	1 GPU, 16 GB memory
TinyLlama-1.1B-Chat	Prompt 2	$c_0\beta^\dagger + c_1\alpha^\dagger$	3	100	1 GPU, 16 GB memory
TinyLlama-1.1B-Chat	Prompt 3	$c_0\beta^\dagger(c_1 + c_2\alpha^\dagger \cos \theta)$	3	100	1 GPU, 16 GB memory
Qwen2.5-Coder-7B-Instruct	Prompt 1	$c_0\beta^\dagger$	3	100	1 GPU, 64 GB memory
Qwen2.5-Coder-7B-Instruct	Prompt 2	$c_0\beta^\dagger + c_1\alpha^\dagger$	3	100	1 GPU, 64 GB memory
Qwen2.5-Coder-7B-Instruct	Prompt 3	$c_0\beta^\dagger(c_1 + c_2\alpha^\dagger \cos \theta)$	3	100	1 GPU, 64 GB memory

Note: c_0 , c_1 , and c_2 are free continuous parameters to be learned. GPU partition uses NVIDIA A30 GPU.

LLM-SR prompt design for TinyLlama-1.1B-Chat while learning (**ADA**) under $\sigma = 0.2$.

Prompt 1: Weak syntactic guidance.

- Search target: a simple vectorized NumPy expression mapping $(\beta^\dagger, \alpha^\dagger, \theta)$ to f .
- Allowed objects: $\beta^\dagger, \alpha^\dagger, \theta$, number of free constants are 3, and elementary trigonometric operations are sin and cos.
- Constraints: exactly two executable lines, no auxiliary variables, reassignment, or invalid output shapes.
- Initialization: $c_0\beta^\dagger$.

Prompt 2: Scientist-guided moderate prompt.

- Adds domain context: the response is an angular distribution with possible forward-backward asymmetry.
- Describes the roles of the variables: $\beta^\dagger$ acts as an overall scale, $\alpha^\dagger$ controls asymmetry, and θ is an angular variable.
- Encourages compact expressions with low complexity and cosine-type angular dependence.
- Initialization: $c_0\beta^\dagger + c_1\alpha^\dagger$.

Prompt 3: Strongly structured scientific prompt.

- States the additive Gaussian noise model and gives the expected scientific form: response = scale $\times$ (baseline + asymmetry $\times$ angular factor).
- Specifies that $\beta^\dagger$ is the scale, $\alpha^\dagger$ controls asymmetry, and the angular factor θ is cosine-like.
- Initialization: $c_0\beta^\dagger(c_1 + c_2\alpha^\dagger \cos \theta)$.

LLM-SR prompt design for Qwen2.5-Coder-7B-Instruct while learning (**ADA**) under $\sigma = 0.2$.

Prompt 1: Weak / fair prompt.

- Search target: a compact vectorized NumPy expression learned from data.
- Guidance level: minimal scientific information; only syntactic and execution constraints are imposed.
- Constraints: two executable lines, no reassignment, valid output shape, and restricted elementary operations.
- Initialization: $c_0\beta^\dagger$.

Prompt 2: Scientist-guided prompt.

- Provides the scientific target structure $\beta^\dagger(1 + \alpha^\dagger \cos \theta)$.
- Explains variable roles: $\beta^\dagger$ is a scale parameter, $\alpha^\dagger$ controls asymmetry, and θ is the angular coordinate.
- Encourages recovery of the interaction term $\beta^\dagger \alpha^\dagger \cos \theta$, while discouraging unnecessary noise-fitting terms.
- Initialization: $c_0\beta^\dagger + c_1\alpha^\dagger$.

Prompt 3: Strongly structured scientific prompt.

LLM-SR prompt design continued from previous page

- Fully specifies the expected multiplicative scientific structure: $\text{scale} \times (\text{baseline} + \text{angular modulation})$.
- Introduces a strong inductive bias toward a cosine-based interaction between α and θ .
- Initialization: $c_0\beta^\dagger(c_1 + c_2\alpha^\dagger \cos \theta)$.

Results. The results below show that LLM-SR is highly sensitive to prompt specification and initialization across both TinyLlama-1.1B-Chat and Qwen2.5-Coder-7B-Instruct. Under weak or moderately informative prompts, the generated candidates are often incomplete or non-executable, typically consisting only of docstrings or function signatures, and are therefore rejected by the evaluator. Even with strongly structured prompts that describe the underlying scientific mechanism, the generated expressions may only partially recover the interaction $\beta^\dagger(\cdot + \alpha^\dagger \cos \theta)$, while introducing extraneous transformations or violating execution constraints. This leads to degraded predictive performance relative to the noise level $\sigma = 0.2$. These findings suggest that LLM-based SR methods are powerful and complementary, but their practical reliability can depend strongly on prompt design, initialization, model hosting, and filtering of invalid generated expressions.

In contrast, VaSST provides a computationally efficient probabilistic framework for symbolic structure learning. It directly optimizes a differentiable soft symbolic tree representation and quantifies uncertainty over competing symbolic forms, without relying on prompt engineering or large-scale generative sampling. As reported in Appendix L, VaSST exactly recovers the symbolic structure of (ADA) across the considered noise levels, and its out-of-sample RMSEs on a 10% held-out test set closely track the corresponding injected noise levels.

LLM-SR results for TinyLlama-1.1B-Chat while learning (ADA) under $\sigma = 0.2$.

Prompt 1: Weak syntactic guidance.

- Generated output: only a function signature and descriptive text were produced; no executable symbolic body was returned.
- Interpretation: invalid/incomplete candidate.
- Score: -10^{20} .

Prompt 2: Scientist-guided moderate prompt.

- Generated output: only a function signature and descriptive text were produced; no executable symbolic body was returned.
- Interpretation: invalid/incomplete candidate.
- Score: -10^{20} .

Prompt 3: Strongly structured scientific prompt.

- Generated output: the evolved candidate was equivalent to $\hat{f} = 1 + c_0\beta^\dagger(c_1 + \alpha^\dagger \cos \theta)$.
- Interpretation: the candidate partially captures the angular interaction $\beta^\dagger(\cdot + \alpha^\dagger \cos \theta)$, but introduces an extraneous additive shift and violates the required execution format.
- Out-of-sample RMSE computed on a 10% held-out test set: 0.3748.

Summary. TinyLlama-1.1B-Chat is highly sensitive to prompt specification. Prompts 1-2 fail to produce executable symbolic expressions, while Prompt 3 recovers part of the correct angular structure but adds an incorrect transformation. The initialized Prompt 3 structure $c_0\beta^\dagger(c_1 + c_2\alpha^\dagger \cos \theta)$ achieves out-of-sample RMSE ≈ 0.2012 after parameter fitting, close to the injected noise level.

LLM-SR results for Qwen2.5-Coder-7B-Instruct while learning (ADA) under $\sigma = 0.2$.

Prompt 1: Weak / fair prompt.

- Generated output: only a function signature and descriptive text were produced; no executable symbolic body was returned.
- Interpretation: invalid/incomplete candidate.
- Score: -10^{20} .

Prompt 2: Scientist-guided prompt.

- Generated output: only a function signature and descriptive text were produced, despite the additional scientific guidance.
- Interpretation: invalid/incomplete candidate.
- Score: -10^{20} .

Prompt 3: Strongly structured scientific prompt.

- Generated output: the model produced a descriptive docstring with an extended explanation and an additional “improved version” section, but no valid executable symbolic body.
- Interpretation: invalid/incomplete candidate under the strict execution constraints.
- Score: -10^{20} .

Summary. Across all prompting regimes, Qwen2.5-Coder-7B-Instruct frequently generates descriptive rather than executable symbolic programs under the strict LLM-SR formatting constraints. Nevertheless, the structured initialized form $c_0\beta^\dagger(c_1 + c_2\alpha^\dagger \cos \theta)$ achieves out-of-sample RMSE ≈ 0.2012 after parameter fitting, indicating that the target expression is representable within the search space but is not reliably recovered through the generated samples.