PHILOSOPHICAL TRANSACTIONS A

royalsocietypublishing.org/journal/rsta

Research

Article submitted to journal

Subject Areas:
xxxxx, xxxxx, xxxx

Keywords:
Symbolic Regression, Bayesian Statistics, Sequential Monte Carlo

Author for correspondence:
G. F. Bomarito
e-mail: geoffrey.f.bomarito@nasa.gov

THE ROYAL SOCIETY PUBLISHING

Bayesian Symbolic Regression via Posterior Sampling

G. F. Bomarito[1] and P. E. Leser[1]

[1]NASA Langley Research Center, Hampton, VA, USA

Symbolic regression is a powerful tool for discovering governing equations directly from data, but its sensitivity to noise hinders its broader application. This paper introduces a Sequential Monte Carlo (SMC) framework for Bayesian symbolic regression that approximates the posterior distribution over symbolic expressions, enhancing robustness and enabling uncertainty quantification for symbolic regression in the presence of noise. Differing from traditional genetic programming approaches, the SMC-based algorithm combines probabilistic selection, adaptive annealing, and the use of normalized marginal likelihood to efficiently explore the search space of symbolic expressions, yielding parsimonious expressions with improved generalization. When compared to standard genetic programming baselines, the proposed method better deals with challenging, noisy benchmark datasets. The reduced tendency to overfit and enhanced ability to discover accurate and interpretable equations paves the way for more robust symbolic regression in scientific discovery and engineering design applications.

1. Introduction

Symbolic regression (SR) stands out as a machine learning technique that promises compact, human-interpretable representations and the potential to uncover valuable physical insights [1,2]. Unlike traditional regression methods that require a pre-defined model structure, SR aims to discover the underlying mathematical expression that best describes relationships within a dataset. This capability has led to its successful application in diverse fields, including the identification of physical laws [3–6], discovery of governing equations [2,7–12], and the development of predictive regressors [1,2,13–15]. The appeal of SR lies in its ability to automatically learn both the form and parameters of a model directly from data, offering a powerful tool for scientific discovery and engineering design.

A significant challenge hindering the more widespread adoption of SR is its sensitivity to noise and limited data availability. These are often the very scenarios where SR's unique ability to extract interpretable relationships from data would be most beneficial. Recent benchmark studies have demonstrated that the effectiveness of many SR algorithms on system identification and predictive regression tasks is severely reduced when even nominal levels of noise are introduced into training datasets [16,17]. Thus, the propensity to overfit noisy data, coupled with the vast search space of possible symbolic expressions, makes the robustness of SR a critical area for improvement.

In this work, we pursue robust SR through the approximation of the distribution representing the probability of symbolic expressions given our observed dataset: *i.e.,* the Bayesian posterior distribution over symbolic expressions. This approach aims to produce a family of symbolic expressions, each associated with a probability density reflecting its relative likelihood given the observed data. By characterizing the posterior distribution, we ultimately obtain (1) a maximum *a posteriori* (MAP) expression, representing the most probable (*i.e.*, maximally predictive) expression, and (2) the means to quantify and express uncertainty in predictions, both in the form of the expression and in its parameters. This uncertainty quantification is crucial for assessing the robustness and reliability of SR models, especially in noisy environments.

To efficiently explore the vast space of possible symbolic expressions and approximate the Bayesian posterior, we leverage the power of Sequential Monte Carlo (SMC) [18]. SMC provides a framework for iteratively refining a population of symbolic expressions, guiding the search towards regions of non-zero posterior probability density. By maintaining a diverse set of candidate models and adapting the population based on evidence from the data, SMC enables us to effectively handle the challenges posed by noise and limited data. Furthermore, the population-based nature of SMC readily allows for the estimation of multimodal posteriors, which are common in SR applications [19].

In a related work [20], Bayesian SR has been performed using Markov Chain Monte Carlo (MCMC) [21] for posterior sampling; however, MCMC has a few notable drawbacks in this application. First, MCMC is less scalable than SMC, requiring long chains of calculations that must be performed serially. Secondly, MCMC can struggle to efficiently explore complex and high-dimensional domains such as that of symbolic expressions; this is particularly true when the posterior distribution is multi-modal. The difficulty of search could potentially be addressed by carefully tuning a transition kernel to a given problem or by using more advanced, adaptive MCMC algorithms, but methodology for this adaptation in the domain of SR is still lacking. On the other hand, the effectiveness of population-based algorithms for SR is underscored by the fact that genetic-programming-based SR (GPSR) methods currently achieve state-of-the-art performance on many benchmark datasets [16,22]. Thus, we posit that because SMC is population-based method, it provides a more natural and effective framework for performing Bayesian inference in the complex landscape of symbolic expressions.

Other Bayesian approaches, specifically Bayesian model selection, have previously been incorporated into GPSR [23–25]. These works have illustrated benefits to GPSR including reducing bloat, increasing generalizabity, and introduction of physical knowledge through definition of a prior. Here, we aim to expand on these works by maintaining these benefits while incorporating a principled application of Bayesian inference. This addition ensures that our final population of expressions converges to an approximation of the Bayesian posterior.

This paper introduces a novel SMC-based algorithm for Bayesian SR, addressing the limitations of existing MCMC-based methods and expanding on previous Bayesian GPSR efforts. Our primary contributions include:

(i) The development of a robust SMC framework for effectively exploring the space of symbolic expressions and approximating the Bayesian posterior,
(ii) A thorough evaluation on benchmark datasets, highlighting the superior performance (better accuracy and generalizability) of the proposed method over GPSR in noisy environments,

(iii) A discussion of likely reasons for improved performance with supporting analysis.

The remainder of this paper details the methodological details of our approach (Section 2), including the specific SMC implementation used and comparison with other methods. Subsequently (Section 3), we present experimental results demonstrating the efficacy of our method on a range of benchmark datasets, specifically highlighting its robustness to noise. Finally, in Sections 4 and 5, we discuss the implications and nuances associated with these findings and outline potential directions for future research.

2. Methods

The Bayesian approach to SR, henceforth referred to as Baysian SR, involves estimating the joint posterior distribution over models, $\mathcal{M}$, and parameters, $\boldsymbol{\theta}$, given observed data, $\mathcal{D}$:

$$\underbrace{\pi(\boldsymbol{\theta}, \mathcal{M}|\mathcal{D})}_{\text{Joint Posterior}} = \frac{\overbrace{\pi(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M})}^{\text{Likelihood}}\, \pi(\boldsymbol{\theta}|\mathcal{M})\pi(\mathcal{M})}{\pi(\mathcal{D})}. \tag{2.1}$$

Here, $\mathcal{M}$ is a function of $\boldsymbol{\theta} \in \mathbb{R}^{N_\theta}$ and independent variables $\mathbf{x} \in \mathbb{R}^{N_x}$, *i.e.*, $\mathcal{M}(\mathbf{x}, \boldsymbol{\theta}) \approx y$, and the dataset $\mathcal{D}$ contains N_d observations of $(\mathbf{x}, y)$ pairs. The domain of $\mathcal{M}$ is discrete while the domain of $\boldsymbol{\theta}$ is continuous with dimension N_θ varying for a given $\mathcal{M}$. Exploration of the trans-dimensional space of models and parameters is difficult. Jin et al. [20] used reversible jump MCMC to estimate an approximation of $\pi(\boldsymbol{\theta}, \mathcal{M}|\mathcal{D})$; however, their method still required a number of simplifications and constraints on the model structure to address inefficient sampling.

Given that the ultimate goal in SR is to identify models matching the data, (2.1) can be marginalized over the parameter dimension, resulting in the posterior over models,

$$\underbrace{\pi(\mathcal{M}|\mathcal{D})}_{\text{Model Posterior}} = \int_{\mathbb{R}^{N_\theta}} \pi(\boldsymbol{\theta}, \mathcal{M}|\mathcal{D}) d\boldsymbol{\theta} \quad \propto \quad \underbrace{\pi(\mathcal{M})}_{\text{Model Prior}} \underbrace{\int_{\mathbb{R}^{N_\theta}} \pi(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M})\pi(\boldsymbol{\theta}|\mathcal{M}) d\boldsymbol{\theta}}_{\text{Marginal Likelihood}}. \tag{2.2}$$

The primary benefit of marginalization is that the model space can be explored without requiring trans-dimensional jumps in $\boldsymbol{\theta}$, leading to more efficient approximation of the posterior with standard sampling algorithms (*e.g.*, MCMC and SMC). The drawback of the marginalization is that the integral on the right hand side, referred to as the marginal likelihood, must be computed for every proposed $\mathcal{M}$, resulting in a computationally-intensive double loop. Each iteration of the inner loop is equivalent to solving for the normalizing constant of the more classic, fixed-model Bayesian inference problem. In other words, we gain efficiency in navigation between models at the cost of a higher computation needed for each model.

The efficient approximation of the marginal likelihood in a SR framework is the subject of previous works [23–25]. It was noted that little prior information on $\boldsymbol{\theta}$ is available in SR since the model form is unknown *a priori*. Therefore, the prior is often chosen to be an uninformative, improper uniform distribution, *i.e.*, $\pi(\boldsymbol{\theta}|\mathcal{M}) \propto 1$, which results in an indeterminate constant appearing in the marginal likelihood. Originally proposed in [26], the use of a normalized marginal likelihood (NML) has been used to alleviate this issue in SR [19,23–25],

$$q = \frac{\int \pi(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M})\pi(\boldsymbol{\theta}|\mathcal{M})}{\int \pi(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M})^\gamma \pi(\boldsymbol{\theta}|\mathcal{M})}, \tag{2.3}$$

where the empirically motivated choice of $\gamma = 1/\sqrt{N_d}$ is used [26].

The NML can be estimated directly with SMC [23,24] or approximated using the Laplace approximation [25],

$$q \approx \gamma^{N_\theta/2} \pi(\mathcal{D}|\boldsymbol{\theta}^*, \mathcal{M})^{(1-\gamma)}. \tag{2.4}$$

Here, $\boldsymbol{\theta}^*$ is the maximum *a posteriori* estimate,which is equivalent to the maximum likelihood estimate given the improper uniform prior. In practice, $\boldsymbol{\theta}^*$ is estimated using a gradient-based optimization. The Laplace approximation is orders of magnitude faster than using SMC to estimate NML but is less accurate for many common SR models [19]. We use the Laplace approximation to estimate q in this work as it was assumed that an inner loop implementation of SMC would be computationally intractable.

Given the NML estimator and a choice of prior over models, $\pi(\mathcal{M})$, MCMC or SMC can be used to draw samples from the posterior (2.2) as a means of performing Bayesian SR. In this work, we choose SMC for the following reasons: (1) MCMC requires a carefully tuned proposal or can suffer from very low acceptance rates whereas SMC, a global, population-based algorithm, is generally more robust to proposal selection; (2) MCMC has strict ergodicity requirements to ensure that the stationary distribution of the Markov chain is equivalent to $\pi(\mathcal{M}|\mathcal{D})$, whereas SMC can relax these requirements; and (3) MCMC is fundamentally serial whereas SMC can be parallelized to improve computational efficiency.

(a) Posterior Sampling with Sequential Monte Carlo

SMC-based SR (SMC-SR) works by evolving a population of weighted symbolic expressions, $P = \{\mathcal{M}_1, \ldots, \mathcal{M}_{N_P}\}$, through a series of target distributions using sequential importance sampling and resampling. Using a process called likelihood-tempering, the t^{th} target distribution is defined as $p_t(\mathcal{M}) \propto \pi(\mathcal{M}) \left[\int_{\mathbb{R}^{\mathbb{N}_\theta}} \pi(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M})\pi(\boldsymbol{\theta}|\mathcal{M})d\boldsymbol{\theta}\right]^{\phi_t}$ with $0 = \phi_0 < \ldots < \phi_t < \ldots < \phi_T = 1$ such that target distributions transition smoothly from the prior to the posterior. In this way, the initial population can be sampled directly from $\pi(\mathcal{M})$, which is known.

At each step, ϕ_t is chosen adaptively [27] to maintain a user-specified effective sample size (ESS) defined as $1/\sum_{i=1}^{N_P}(W_i)^2$ where W_i is the normalized weight associated with the i^{th} expression. The ESS can vary between 1 and N_P, where ESS is correlated with the amount of information contained in the population. To avoid degeneracy (*i.e.*, most model weights are near zero and ESS is low), the expressions are reweighted according to the new target and resampled with replacement, where resampling probabilities are generally proportional to W_i. While a number of resampling strategies exist, stratified resampling [28] was used in this work to encourage expression diversity. It is often the case that $p_0 = \pi(\mathcal{M})$ differs significantly from the final target $p_T = \pi(\mathcal{M}|\mathcal{D})$ leading to degeneracy despite reweighting. To combat this, the SMC algorithm employs a short run of a forward MCMC kernel to move the population toward the current target prior to each update of ϕ_t, promoting diversity and global exploration of posterior modes. For more general information on SMC, see [18].

Algorithm 1 provides a summary of the proposed SMC-SR algorithm and contrasts it with GPSR, which is summarized in Algorithm 2. It is clear from the comparison that the primary differences are the SMC-specific steps (*e.g.*, iterating through target distributions, reweighting, resampling, and the MCMC forward kernel) and the definition of the acceptance probability, α. For the SMC approach, the acceptance probability is the classic random walk Metropolis step whereas traditional GPSR performs a binary accept/reject based on whether fitness has improved or not. The key ingredients for the MCMC kernel are α and proposal distribution, $h(O|P)$, used to generate offspring, O, from parents in P. Standard GPSR variation operations are used for the proposal, *i.e.*, population-based crossover and mutation [29]; details are included in Appendix A.

It is worth noting that our proposal strategy is asymmetric, meaning $h(O_i|P_i) \neq h(P_i|O_i)$. This can be detrimental to MCMC due to its strict ergodicity requirement. We instead rely on the fact that our MCMC kernel is implemented within the broader SMC algorithm where ergodicity improves sampling but is not strictly required as long as the kernel is effectively rejuvenating the population. The combination of global exploration, less sensitivity to proposal selection, relaxation of ergodicity requirements, and computational efficiency made SMC a natural choice over MCMC for this initial study. A more detailed comparison of SMC and MCMC for Bayesian SR is included in Appendix B.

Algorithm 1 Sequential Monte Carlo	**Algorithm 2** Genetic Programming
given a dataset, $\mathcal{D}$	**given** a dataset, $\mathcal{D}$
define $q(\cdot)$ as the NML (2.4) given $\mathcal{D}$	**define** $e(\cdot)$ as a loss function given $\mathcal{D}$
Generate a set of symbolic expressions P	Generate a set of symbolic expressions P
Evaluate $q(P_i)$ for P_i in P	Evaluate $e(P_i)$ for P_i in P
Initialize $\phi_t = 0$	·
Set uniform weights $W_i = 1/N_p$	·
repeat	**repeat**
Update ϕ_t	·
Update weights W	·
Resample P with replacement	·
repeat N_{MCMC} **times**	·
Generate offspring O of P	Generate offspring O of P
Evaluate $q(O_i)$ for O_i in O	Evaluate $e(O_i)$ for O_i in O
Pair parent P_i with offspring O_i	Pair each parent P_i with its offspring O_i
$\alpha = \min(1, (q(O_i)/q(P_i))^{\phi_t})$	$\alpha = 1$ if $e(O_i) \leq e(P_i)$ else 0
Replace P_i with O_i with probability α	Replace P_i with O_i with probability α
end	·
until $\phi = 1$	**until** convergence or repetition limit

A number of additional choices were made that are briefly summarized here for repeatability. First, throughout this work it is assumed that noise in the data is independent and identically distributed (iid) such that, for a given model, $y_j = \mathcal{M}(\mathbf{x}_j, \boldsymbol{\theta}) + \epsilon_j$ where $\epsilon_j \overset{\text{iid}}{\sim} N(0, \sigma)$ and σ could be estimated as part of the inference. The prior over models is assumed to be uniform, *i.e.*, $\pi(\mathcal{M}) \propto 1$. However, it should be noted that this was a choice and not a requirement of the method. Informative priors such as those proposed by Bartlett et al. [25] could be implemented, and simply impart a users' belief upon the likely model structure. Here, expressions were randomly generated for the initial population P until a unique set of expressions of desired size was achieved.[1] Though this generation process likely imparts some nonuniformity, it is an approximation of samples of the uniform prior $\pi(\mathcal{M}) \propto 1$.

A handful of hyperparameters are required for the SMC algorithm. They are listed here and will remain constant for the remainder of this work (unless otherwise noted). The total number of expressions in the population was $N_P = 2000$. To ensure the forward kernel provided good mixing (generally measured as the percentage of final offspring that differ from the original parent, where higher percentages are better), N_{MCMC} was set to 10. The target ESS was 1900, or 95% of the population size. Higher percentages generally result in smaller ϕ_t updates and thus more iterations. No effort was made to optimize hyperparameters (though future work may find this fruitful), besides observing that relatively large values of N_P are preferred to smaller sizes. Although larger populations impart a larger computational burden, we posit that larger population sizes allow for more expression diversity early on in sampling, which in turn leads to more efficient exploration of intermediate targets and, ultimately, the posterior.

(b) Demonstration

The key aspects of SMC-SR are demonstrated here using a simple case study. A dataset was generated with $y_j = \frac{1}{2\pi} e^{-\frac{(x_j)^2}{2}} + \epsilon_j$ where $x_j \sim \mathcal{U}(-3, 3)$ and $\epsilon_j \sim \mathcal{N}(0, 0.16)$. The dataset comprised 25 datapoints. SMC-SR was applied to this dataset using the operators $[+, -, \times]$ in

[1] Equality testing of expressions occurs after symbolic simplification and reorganization into a canonical form. However, these processes are imperfect and may not identify all symbolic equivalences.

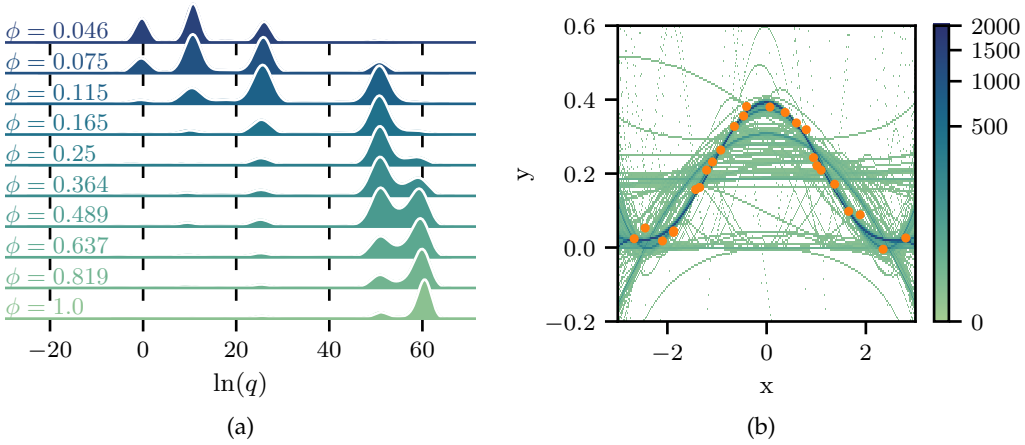

Figure 1: Demonstration of SMC-SR. (a) The effect of likelihood-tempering on the distribution of NML. (b) The model posterior viewed as a histogram. Points indicate the training data.

order to make the regression more difficult since SMC could quickly identify the true expression given an exponential operator.

Due to difficulties visualizing models sampled from the posterior $\pi(\mathcal{M}|D)$, we instead show two alternatives in Figure 1. In Figure 1a the distribution of natural log of NML, $\ln(q)$, is shown which allows for the visualization of the posterior in a single dimension. The distribution is also shown for some intermediate target distributions (*i.e.*, at intermediate ϕ_t values). The figure illustrates how likelihood-tempering in SMC corresponds to early exploration (larger spread in distribution at $\phi = 0.046$) followed by gradual convergence toward a multimodal posterior. Figure 1b depicts the model posterior as a two-dimensional histogram in the space of the data. Over most of the domain, there is very high, unimodal density in the areas between the training data points with histogram frequency of nearly 2000 (the population size used). Areas with disagreement between models (which look more multimodal) indicate two things: an area where caution might be exercised when making predictions and also an area where additional training data might be especially insightful.

Note that Figure 1b is an effort to illustrate the posterior $\pi(\mathcal{M}|D)$, but it should not be confused with the joint posterior $\pi(\boldsymbol{\theta}, \mathcal{M}|\mathcal{D})$ which is commonly used to create credible intervals. However, the process is straightforward to approximate $\pi(\boldsymbol{\theta}, \mathcal{M}|\mathcal{D})$ from our approximation of $\pi(\mathcal{M}|D)$: using the final P produced with SMC-SR and recalibrating the $\boldsymbol{\theta}$ in each model (using *e.g.*, standard MCMC or SMC tools).

3. Results

The efficacy of SMC-SR was quantified using a set of 12 benchmark problems. These benchmark problems were modelled after equations seen in Richard Feynman's physics lectures [5] and have been incorporated into the SR benchmarking suite SRBench [16,22]. Previous results on the Feynman datasets have illustrated that the addition of even 1% noise into the datasets drastically reduces performance of most SR algorithms [17]; thus, noisy versions of the Feynman datasets represent a pertinent challenge for Bayesian SR methods. The 12 specific datasets chosen for this work are chosen based on the most difficult subset of Feynman datasets [30] that are scheduled to be incorporated into the upcoming revision of SRBench.[2] Training on the datasets are repeated 20 times in this study.

The datasets each consist of 10,000 data points that are obtained by evaluating a known physical equation over a range of its input parameter space. The dimension of the input, N_x,

[2] https://github.com/cavalab/srbench/discussions/174#discussioncomment-10285133

ranged from 3 to 8. Training subsets with size $N_d = 500$ data points were extracted, and the remaining data points were withheld as test sets. Gaussian noise was added to the training set with standard deviation of 10% of the magnitude of the dataset, *i.e.*, $0.1||Y||$. The magnitude was calculated as $||Y|| = \text{median}(|y|)$ for the datasets because many contain data points very close to asymptotes that could skew other norms. Normalized root mean squared error (NRMSE) will be used when discussing results; here the standard root mean squared error is normalized by $||Y||$. Thus, a NRMSE of 0.1 corresponds to achieving an accuracy level comparable to the added noise. We define here NRMSE-train and NRMSE-test as two metrics corresponding to NRMSE calculated using the training dataset and testing dataset, respectively.

SMC-SR is compared to 3 GPSR baselines. The GPSR baselines have minimal variation from SMC-SR (see Algorithm 2) to make as direct a comparison as possible. For instance, the population size in GPSR and SMC-SR are both set to a fixed value of $N_P = 2000$. The GPSR baselines vary based on their loss function and selection algorithm as indicated below:

	Loss Function	Selection Algorithm
GP-MSE	MSE	deterministic crowding (less aggressive)
GP-NML	NML	deterministic crowding (less aggressive)
GP-agg	MSE	tournament (more aggressive)

Since the SMC algorithm is adaptive, T, the total number of ϕ_t updates, is not fixed *a priori*. In order to preserve a consistent level of compute, the SMC-SR benchmarks are performed first and the GPSR benchmarks are performed subsequently with a number of generations equal to $T \times N_{MCMC}$. Repetitions of the algorithms on the same dataset may or may not have the same number of SMC steps; hence, a normalized compute metric $[0 - 1]$ is used to indicate algorithm progress in further figures and discussions.

The effectiveness of SMC-SR on two select Feynman datasets is illustrated in Figure 2. For dataset I-32-17, lower levels of NRMSE-train is achieved with SMC-SR than any GPSR method. By comparing GP-MSE and GP-NML we can see that the addition of the regularization provided by the NML loss is partly responsible for the improved performance of SMC-SR. For dataset I-36-38, GP-agg achieves a low level of NRMSE-train quicker than SMC-SR, indicating that a high selection pressure (provided by resampling in SMC-SR and by tournament selection in GP-agg) leads to more rapid fitting of the training data. However, the high values of NRMSE-test for GP-agg indicate that the higher selection pressure leads GP-agg to overfitting while SMC-SR does not. This poor performance of GP-agg indicates that higher selection pressure cannot be solely the cause for improved performance in SMC-SR. The minimum NRMSE-test over the population P is shown to illustrate the presence of highly-fit expressions in the population. These highly-fit expressions are encountered more quickly with SMC-SR. The practical ability to identify the highly-fit expressions without access to a large test dataset, especially with a population size of 2000 expressions, is discussed later in this section.

The effect of the SMC algorithm on parametric complexity (N_θ) and model form complexity (number of nodes needed to represent an expression as an acyclic graph) is illustrated in Figure 3. Early on in SMC-SR, while in a state of exploration with $\phi \ll 1$ the complexity rises very quickly, at approximately the rate of GP-MSE. But as the NML is tempered and $\phi \to 1$, regularization appears to take hold and complexities approach that of GP-NML. In other words, SMC-SR first explores and bloats but eventually identifies a region of high posterior density and homes in on those expressions in a way that reduces complexity. The rapid complexity growth for GP-agg illustrates another pitfall seen when blindly increasing selection pressure.

Looking at the results for all the datasets (Figure 4) elucidates more insights into the performance behaviours of the SMC algorithm. The SMC algorithm is much less likely to underfit the datasets, *i.e.*, have a NRMSE-train greater than the noise level, see the top of Figure 4. The middle of Figure 4 illustrates the NRMSE-test of the expressions with best training loss (NML in SMC-SR and explicitly defined above for GPSSR methods). The bars in the plot that land above their counterparts in the top of Figure 4 indicate overfitting. Overfitting is seen in almost all datasets for the GPSR approaches. SMC-SR, however, only overfits on 5 of the 12 datasets,

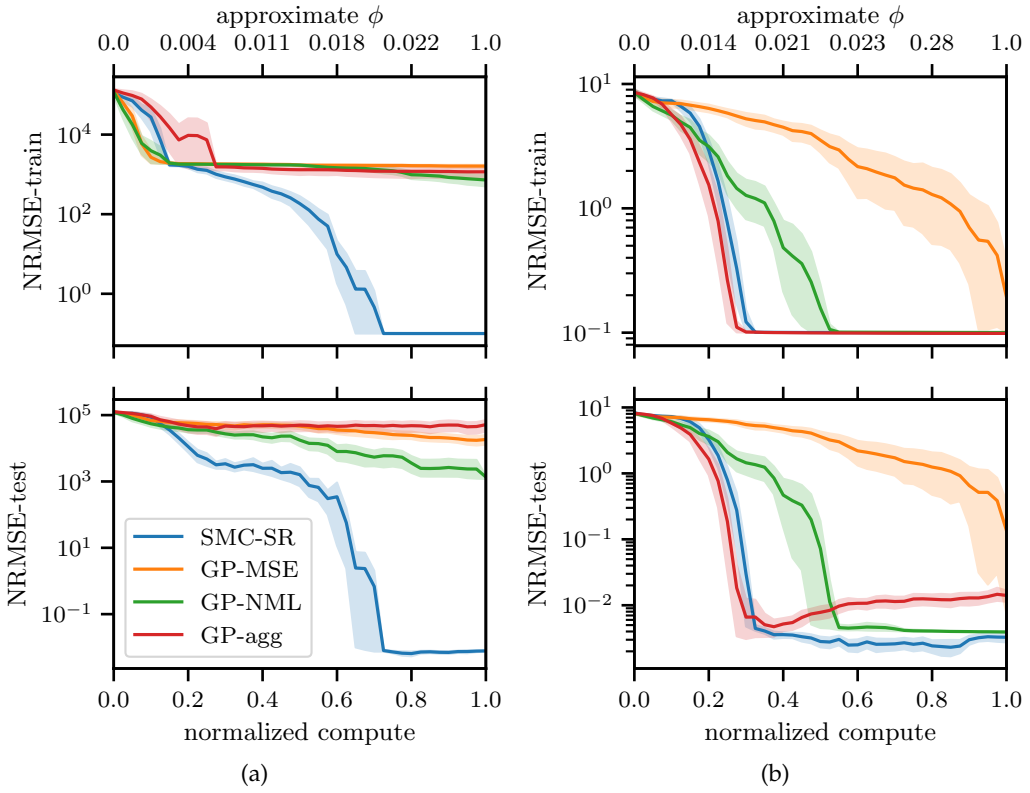

Figure 2: Training results for Feynman dataset (a) I-32-17 and (b) II-36-38. Figures show the lowest (top) NRMSE-train and (bottom) NRMSE-test in the populations.

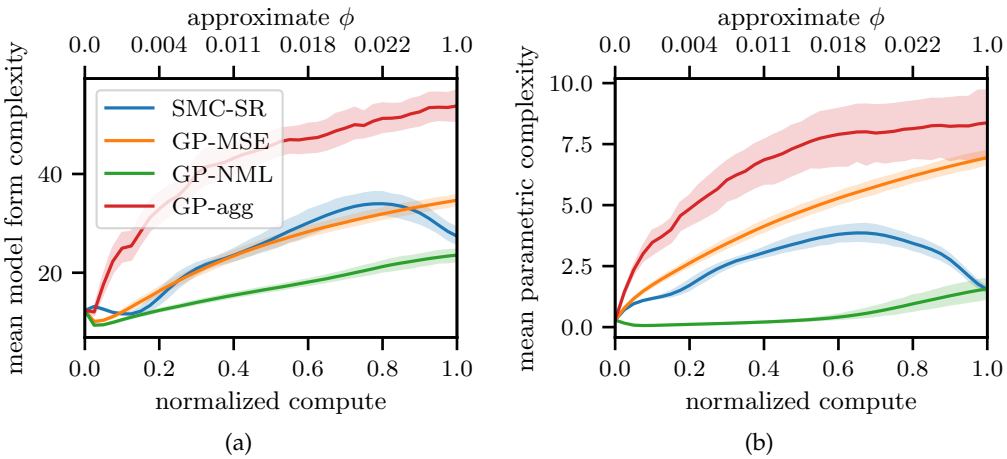

Figure 3: Complexity of models seen when fitting to Feynman dataset I-17-32.

which indicates the potential for improved generalizability with the SMC-SR. The bottom of Figure 4 illustrates the minimum NRMSE-test of the final populations, again highlighting the ability to encounter highly-fit expressions. SMC-SR has most frequent and lowest values in this figure which indicates that it is more likely to encounter these highly-fit expressions.

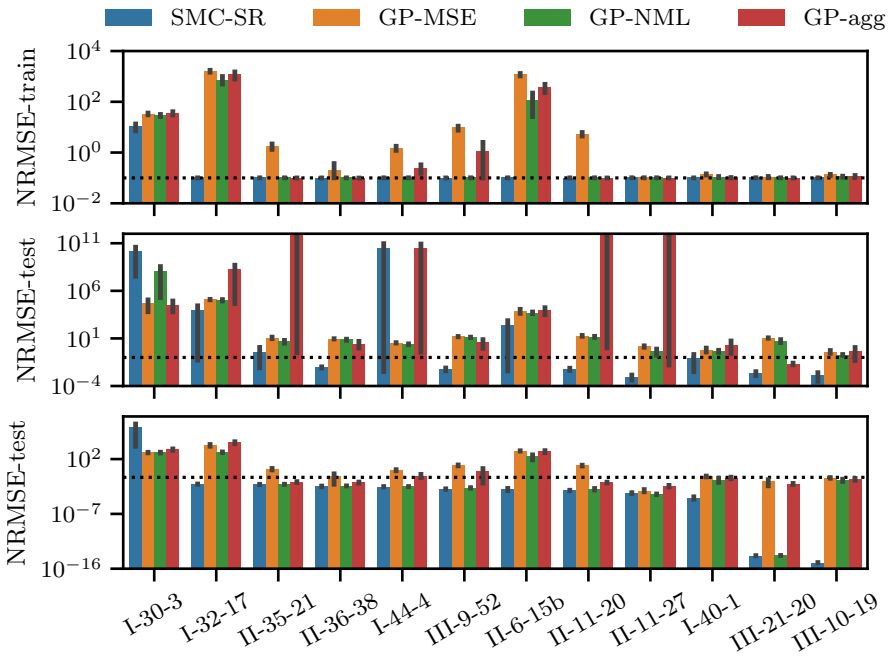

Figure 4: Results on Feynman datasets. (top) NRMSE-train for best models in final population. (middle) NRMSE-test for the same models as top. (bottom) Minimal values of NRMSE-test in final population. The black dotted line indicates the level of noise of 0.1 added to the training data. Note: some GP-agg bars are truncated in the middle plot.

Comparison of the bottom two plots in Figure 4 indicate that apt model selection is a manner of significantly improving SR results, especially in cases with large pools of expressions to select from. We investigate three manners of model selection with respect to SMC-SR: (i) the expression with best training error, which is an approximation of the MAP expression given an uninformative prior, (ii) the expression which has the best error on a cross-validation dataset, 20% the size of the training dataset, and (iii) the mode of the expressions in the final population P. Here the mode is another approximation of the MAP expression, but without assumption on the prior. The top of Figure 5 illustrates that both cross validation (cross val.) and selection of the mode (mode) are improvements over model selection by best training loss (max NML). Interestingly, we see that selection of the mode performs better than cross validation selection despite the lack of additional data. The same trend is present when considering the ability to correctly identify the ground-truth expression, as seen in Figure 5. Here ground-truth identification rate is quantified by refitting numerical constants $\boldsymbol{\theta}$ in a given expression using the test dataset and identifying if the refit NRMSE-test is less than 1e-10.

4. Discussion

The four major differences between SMC-SR and GPSR are the use of NML, resampling, probabilistic selection, and likelihood-tempering. Comparisons of GP-NML and GP-MSE in the previous section indicate that the use of NMLL contributes modestly to the success of SMC-SR. Resampling increases selection pressure, but the poor performance of GP-agg indicates that this alone is insufficient for successful SR. We theorize that the remaining two differences, probabilistic selection and likelihood-tempering (either by themselves or in conjunction with the other two) are responsible for the majority of the success of SMC-SR.

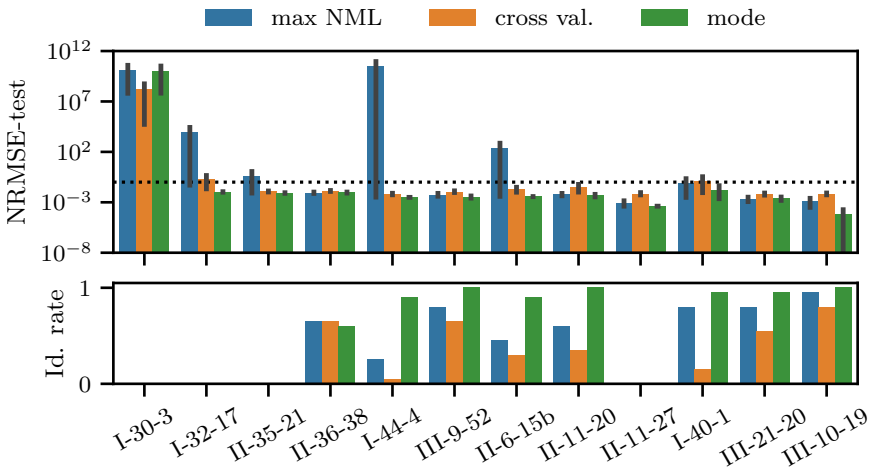

Figure 5: Effect of model selection methodology on (top) NRMSE-test and (bottom) ground-truth identification rate for the Feynman datasets.

It has been found that GPSR tends to struggle in efficiently exploring the domain of expressions and tends to revisit the same expressions frequently [31]. This has given rise to a few techniques that increase emphasis on novelty of expressions in SR [15,32]. Probabilistic selection and likelihood-tempering both promote novelty and allow for less restricted exploration of expressions. However, in tracking the total number of unique expressions encountered, we find that SMC-SR usually has about 50% of the number of unique expressions encountered compared to the GP-based methods (See Figure 6a for an example). While this result could stem in part from the imperfect metric,[3] we posit that pure novelty alone is not what benefits SR. Rather, novelty only in the region of high posterior probability is what provides benefit.

Figure 6b illustrates that, despite fewer total unique models, SMC-SR has a much more dynamic population and accepts more unique models into P. Thus, combining likelihood-tempering with probabilistic selection provides an effective mechanism for *targeted* novelty. It does not, however, ensure that the same expressions are not revisited: restricting the population in such a manner would mean the produced P would no longer represent the posterior. Though, in cases where the full posterior is not important, the combination of likelihood-tempering and/or probabilistic selection with the novelty methods cited earlier could prove fruitful.

The rates of the adaptive likelihood-tempering are illustrated in Figure 7a, where we can see that the typical progression of ϕ_t proceeds through two phases, an exploration phase with $\phi_t \ll 1$ followed by an exploitation phase where ϕ_t rapidly approaches 1. The population dynamics illustrate a similar story in Figure 7b, wherein the number of unique models is larger for SMC than GP early on (exploration) but then flips near the end of computation (exploitation). In the exploitation phase, one of the benefits of duplicated expressions is that each expression can have an attempt at finding optimal $\boldsymbol{\theta}^*$ values as part of the NML calculation. In our work, the optimization of $\boldsymbol{\theta}$ occurs once for each occurrence of an expression and is randomly initialized. In future works, storing $\boldsymbol{\theta}$ values with the expression and using them for future optimizations could add additional efficiency [33].

5. Conclusions

This work introduced a novel Sequential Monte Carlo framework for Bayesian symbolic regression (dubbed SMC-SR) designed to enhance robustness to noise and provide built-in

[3] $\mathcal{M} = \theta_0(1 + \theta_1)$ is identified as a different expression than $\mathcal{M} = (\theta_0 + \theta_1)$ despite the fact that they are the same expression given optimal values of the constants.

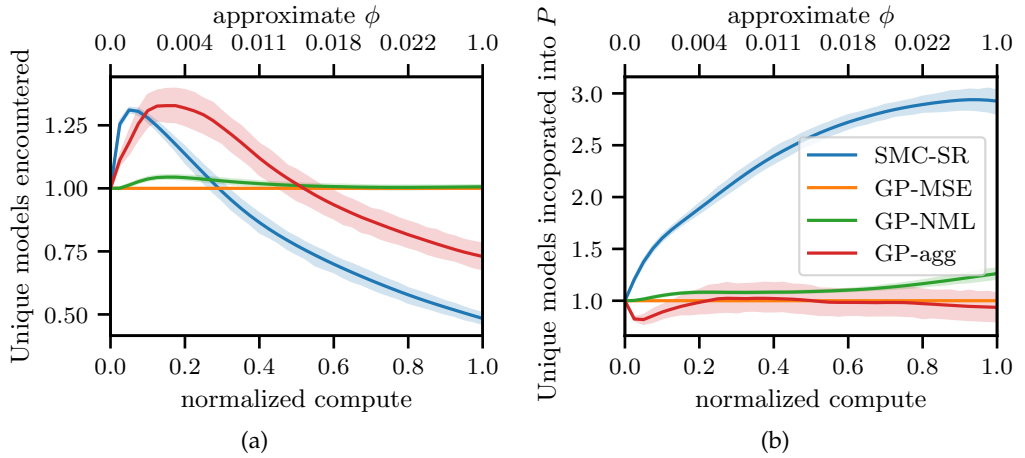

Figure 6: Diversity of populations for Feynman dataset I-32-17. Both are normalized by GP-MSE.

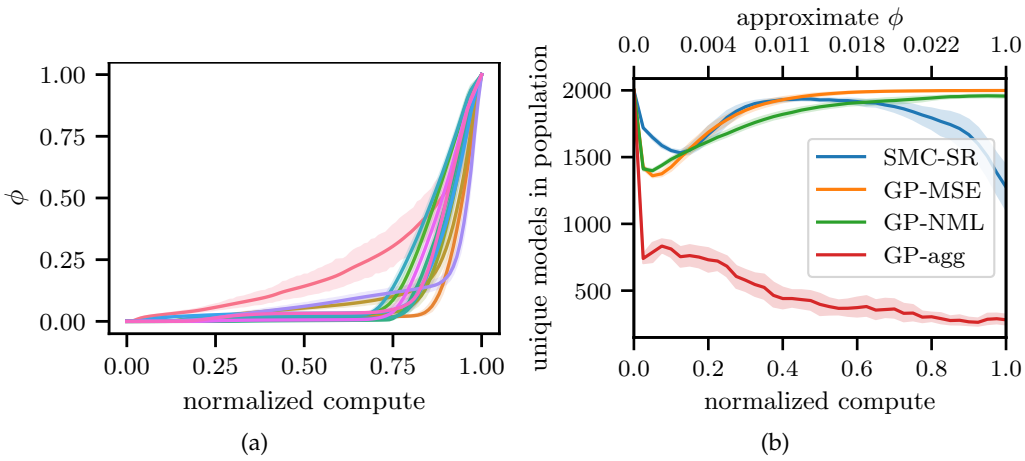

Figure 7: Exploration-exploitation trends of SMC-SR. (a) Adaptively set ϕ for Feynman datasets; each colour is a different dataset. (b) Population diversity for dataset I-32-17.

quantification of uncertainty in model form. Addressing the limitations of existing MCMC-based approaches and expanding on previous Bayesian GPSR efforts, SMC-SR aims to approximate the Bayesian posterior distribution over symbolic expressions. The results demonstrate that SMC-SR outperforms traditional GPSR baselines, particularly in noisy environments, exhibiting a reduced propensity for overfitting and an improved ability to identify highly-fit expressions. Furthermore, the method provides a means to quantify uncertainty in both predictions and equation form, offering a more reliable assessment of the produced expressions.

Our experiments on a challenging subset of the Feynman benchmark datasets revealed that SMC-SR achieves lower training errors more rapidly and, critically, generalizes better to unseen data. Our analysis suggests that the success of SMC-SR is attributable not only to the use of NML and increased selection pressure via resampling, but also to the combination of probabilistic selection and likelihood-tempering. These elements promote *targeted* novelty and exploration of the search space, enabling the algorithm to efficiently identify regions of high posterior probability without being hampered by premature convergence or the inefficiency of pure novelty-seeking.

Since the choice of prior distributions over symbolic expressions has been shown to play a crucial role in Bayesian SR, future work could explore the impact of different priors – particularly those informed by domain knowledge – on the algorithm's performance and the interpretability of the resulting models. Additionally, the incorporation of more accurate methods for calculating NML, such as replacing the Laplace approximation with SMC in the inner loop, should be investigated.

References

1. Koza JR. 1992 *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press.
2. Kronberger G, Burlacu B, Kommenda M, Winkler SM, Affenzeller M. 2024 *Symbolic Regression*. CRC Press.
3. Schmidt M, Lipson H. 2009 Distilling Free-Form Natural Laws from Experimental Data. *Science* **324**, 81–85. (10.1126/science.1165893)
4. Hills DJ, Grütter AM, Hudson JJ. 2015 An Algorithm for Discovering Lagrangians Automatically from Data. *PeerJ Comput. Sci.* **1**, e31. (10.7717/peerj-cs.31)
5. Udrescu SM, Tegmark M. 2020 AI Feynman: A Physics-Inspired Method for Symbolic Regression. *Science Advances* **6**, eaay2631. (10.1126/sciadv.aay2631)
6. Oh H, Amici R, Bomarito G, Zhe S, Kirby RM, Hochhalter J. 2024 Inherently Interpretable Machine Learning Solutions to Differential Equations. *Engineering with Computers* **40**, 2349–2361. (10.1007/s00366-023-01915-7)
7. Brunton SL, Proctor JL, Kutz JN. 2016 Discovering Governing Equations from Data by Sparse Identification of Nonlinear Dynamical Systems. *Proc. Natl. Acad. Sci.* **113**, 3932–3937. (10.1073/pnas.1517384113)
8. Cranmer M, Sanchez Gonzalez A, Battaglia P, Xu R, Cranmer K, Spergel D, Ho S. 2020 Discovering Symbolic Models from Deep Learning with Inductive Biases. In *Adv. Neural Inf. Process. Syst.* vol. 33 pp. 17429–17442. Curran Associates, Inc.
9. Bomarito GF, Townsend TS, Stewart KM, Esham KV, Emery JM, Hochhalter JD. 2021 Development of Interpretable, Data-Driven Plasticity Models with Symbolic Regression. *Computers & Structures* **252**, 106557. (10.1016/j.compstruc.2021.106557)
10. Cranmer M. 2023 Interpretable Machine Learning for Science with PySR and SymbolicRegression.Jl. (10.48550/arXiv.2305.01582)
11. Birky D, Garbrecht K, Emery J, Alleman C, Bomarito G, Hochhalter J. 2023 Generalizing the Gurson Model Using Symbolic Regression and Transfer Learning to Relax Inherent Assumptions. *Modelling Simul. Mater. Sci. Eng.* **31**, 085005. (10.1088/1361-651X/acfe28)
12. Russeil E, de Franca FO, Malanchev K, Burlacu B, Ishida E, Leroux M, Michelin C, Moinard G, Gangler E. 2024 Multiview Symbolic Regression. In *Proc. Genet. Evol. Comput. Conf.* GECCO '24 pp. 961–970 New York, NY, USA. Association for Computing Machinery. (10.1145/3638529.3654087)
13. La Cava WG, Lee PC, Ajmal I, Ding X, Solanki P, Cohen JB, Moore JH, Herman DS. 2023 A Flexible Symbolic Regression Method for Constructing Interpretable Clinical Prediction Models. *npj Digit. Med.* **6**, 1–14. (10.1038/s41746-023-00833-8)
14. Merrell J, Emery J, Kirby RM, Hochhalter J. 2024 Stress Intensity Factor Models Using Mechanics-Guided Decomposition and Symbolic Regression. *Engineering Fracture Mechanics* **310**, 110432. (10.1016/j.engfracmech.2024.110432)
15. Bartlett DJ, Desmond H, Ferreira PG. 2024 Exhaustive Symbolic Regression. *IEEE Trans. Evol. Comput.* **28**, 950–964. (10.1109/TEVC.2023.3280250)
16. de Franca FO, Virgolin M, Kommenda M, Majumder MS, Cranmer M, Espada G, Ingelse L, Fonseca A, Landajuela M, Petersen B, Glatt R, Mundhenk N, Lee CS, Hochhalter JD, Randall DL, Kamienny P, Zhang H, Dick G, Simon A, Burlacu B, Kasak J, Machado M, Wilstrup C, Cavaz WGL. 2024 SRBench++: Principled Benchmarking of Symbolic Regression With Domain-Expert Interpretation. *IEEE Transactions on Evolutionary Computation* pp. 1–1. (10.1109/TEVC.2024.3423681)
17. La Cava W. 2022 SRBENCH Results. https://cavalab.org/srbench/srbench/results/.
18. Del Moral P, Doucet A, Jasra A. 2006 Sequential Monte Carlo Samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **68**, 411–436. (10.1111/j.1467-9868.2006.00553.x)

19. Leser P, Bomarito G, Kronberger G, Olivetti De França F. 2024 Comparing Methods for Estimating Marginal Likelihood in Symbolic Regression. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* GECCO '24 Companion pp. 2058–2066 New York, NY, USA. Association for Computing Machinery. (10.1145/3638530.3664142)
20. Jin Y, Fu W, Kang J, Guo J, Guo J. 2020 Bayesian Symbolic Regression. (10.48550/arXiv.1910.08892)
21. Smith RC. 2024 *Uncertainty quantification: theory, implementation, and applications*. SIAM.
22. La Cava W, Burlacu B, Virgolin M, Kommenda M, Orzechowski P, de França FO, Jin Y, Moore JH. 2021 Contemporary Symbolic Regression Methods and Their Relative Performance. *Adv Neural Inf Process Syst* **2021**, 1–16.
23. Bomarito GF, Leser PE, Strauss NCM, Garbrecht KM, Hochhalter JD. 2022 Bayesian Model Selection for Reducing Bloat and Overfitting in Genetic Programming for Symbolic Regression. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* GECCO '22 pp. 526–529 New York, NY, USA. Association for Computing Machinery. (10.1145/3520304.3528899)
24. Bomarito GF, Leser PE, Strauss NCM, Garbrecht KM, Hochhalter JD. 2023 Automated Learning of Interpretable Models with Quantified Uncertainty. *Computer Methods in Applied Mechanics and Engineering* **403**, 115732. (10.1016/j.cma.2022.115732)
25. Bartlett DJ, Desmond H, Ferreira PG. 2023 Priors for Symbolic Regression. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation* pp. 2402–2411. (10.1145/3583133.3596327)
26. O'Hagan A. 1995 Fractional Bayes Factors for Model Comparison. *Journal of the Royal Statistical Society. Series B (Methodological)* **57**, 99–138.
27. Buchholz A, Chopin N, Jacob PE. 2021 Adaptive Tuning of Hamiltonian Monte Carlo Within Sequential Monte Carlo. *Bayesian Anal.* **16**, 745–771. (10.1214/20-BA1222)
28. Hol JD, Schon TB, Gustafsson F. 2006 On Resampling Algorithms for Particle Filters. In *2006 IEEE nonlinear statistical signal processing workshop* pp. 79–82. IEEE.
29. Randall DL, Townsend TS, Hochhalter JD, Bomarito GF. 2022 Bingo: A Customizable Framework for Symbolic Regression with Genetic Programming. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* GECCO '22 pp. 2282–2288 New York, NY, USA. Association for Computing Machinery. (10.1145/3520304.3534031)
30. Matsubara Y, Chiba N, Igarashi R, Ushiku Y. 2022 Rethinking Symbolic Regression Datasets and Benchmarks for Scientific Discovery. https://arxiv.org/abs/2206.10540v5.
31. Kronberger G, Olivetti de Franca F, Desmond H, Bartlett DJ, Kammerer L. 2024 The Inefficiency of Genetic Programming for Symbolic Regression. In Affenzeller M, Winkler SM, Kononova AV, Trautmann H, Tušar T, Machado P, Bäck T, editors, *Parallel Problem Solving from Nature – PPSN XVIII* pp. 273–289 Cham. Springer Nature Switzerland. (10.1007/978-3-031-70055-2_17)
32. de Franca FO, Kronberger G. 2025 Improving Genetic Programming for Symbolic Regression with Equality Graphs. (10.48550/arXiv.2501.17848)
33. Burlacu B, Winkler SM, Affenzeller M. 2025 Revisiting Gradient-Based Local Search in Symbolic Regression. *Genet. Program. Theory Pract. XXI* **13**, 259.

Appendices

A. Proposal Distribution

Mutation and crossover operations are used to define $h(O|P)$. An offspring O_i is generated from P_i by first applying crossover with another randomly selected parent P_j and then performing mutation. Crossover and mutation are applied with probabilities 0.25, and 0.75, respectively. A linear-stack with single-point crossover is used. Mutation is performed by randomly choosing one of five equally probable mutation types: operator mutation, parameter mutation, operator + parameter mutation, prune mutation, and branch mutation [29]. Variation is repeated until $O_i \neq P_i$. Note that acyclic graph representations of expressions are used in our work, but our results should be unaffected by choice of expression representation.

A challenge with our proposal strategy is that it is asymmetric, meaning $h(O_i|P_i) \neq h(P_i|O_i)$. To ensure detailed balance and facilitate ergodicity under these conditions, a proper MCMC algorithm must incorporate a Metropolis-Hastings step, *i.e.*, $\alpha = \min(1, \frac{q(O_i)^{\phi_t} h(O_i|P_i)}{q(P_i)^{\phi_t} h(P_i|O_i)})$, or suffer from biased sampling. While we expect estimating $h(\cdot|\cdot)$ to be feasible, the added complexity is left to future work, and a basic Metropolis step was employed within the forward MCMC kernel in Algorithm 1. We expect, reweighting and resampling at each step to combat the associated bias as a result of this simplification within SMC.

B. Comparison to MCMC

To further support the choice of SMC over MCMC, the two algorithms were compared on the case study from Section 2(b). For reference, the MCMC algorithm is, after initialization of P with $N_p = 1$, equivalent to the inner loop of Algorithm 1 where N_{MCMC} now represents the total length of the chain prior to burn-in samples being discarded.

SR was performed on the dataset 5 times for both SMC and MCMC. Expressions were restricted to using the operators $[+, -, \times, \div]$ in order to make the regression more difficult since it was found that both SMC and MCMC could quickly identify the true expression given an exponential operator. The total number of expressions in the population was $N_P = 500$ for SMC ($N_P = 1$ for MCMC). For SMC, N_{MCMC} was set to 10 and the target ESS was 475 (95%).

To facilitate a fair comparison, the total number of expression evaluations was kept constant between SMC and MCMC. Replicate simulations were performed, and the exact number of evaluations in each replicate varied due to the adaptive nature of the SMC algorithm. MCMC was run after each SMC replicate to match total evaluations. The average number of expressions evaluated was 186,000. SMC and MCMC were run on the same computational hardware; however, SMC is trivially parallelizable across the population of expressions and utilized multiprocessing. Even with a rudimentary parallelism scheme, where only NML calculations were parallelized, SMC was able to achieve a $15\times$ speed up over MCMC. The ideal speed up of $40\times$ (based on the hardware used) could be better realized with further optimizations and/or larger datasets.

Both algorithms used the same proposal $h(O|P)$ and Metropolis acceptance step. However, SMC is generally more robust to selection of the proposal than MCMC. In this example, the average acceptance rate for MCMC was 4.3%. Tracing the history of each index of the SMC population within the forward kernel, the approximate acceptance ratio for SMC was 15.1%, although the average percentage of offspring that were different from the original parents (*i.e.*, a measure of mixing) was 81% after each set of 10 MCMC steps

Figure B.1 illustrates how the likelihood tempering in SMC leads to a multimodal posterior. In contrast, MCMC only identifies a single mode, corresponding to approximately the highest-density mode estimated by SMC, but slightly offset. It is worth noting again that both algorithms ignore the asymmetry of the proposal distribution. We believe this is more detrimental to MCMC due to its stricter ergodicity requirement. In contrast, SMC may correct for some associated bias with reweighting and resampling. Additionally, it should be noted that the MCMC algorithm proposed here was basic; the MCMC literature is rich with sophisticated variations that could potentially outperform both the MCMC and SMC implementations presented here.

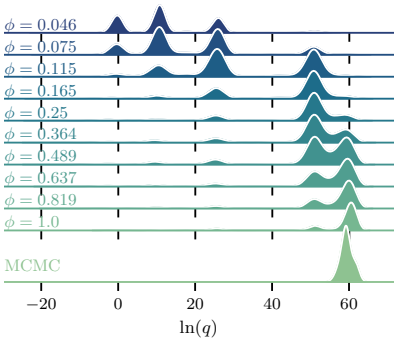

Figure B.1: Illustration Likelihood-tempering of SMC and comparison to MCMC.