SYMBOLIC REGRESSION WITH SELF-SUPERVISED HEURISTIC BEAM SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Symbolic Regression (SR) aims to discover simple and interpretable mathematical expressions that explain observed data, making it a powerful tool for scientific discovery. In this work, we introduce a conceptually simple SR method that is both sample-efficient with respect to observed data points and self-supervised on large-scale synthetic data. By design, our approach favors parsimony, yielding interpretable and concise expressions. We focus on problems with exact solutions, evaluating our method on datasets containing physical laws and dynamical equations. Our results demonstrate that combining beam search with a learned heuristic achieves competitive performance compared to existing methods in SRBench. Additionally, our approach effectively handles expressions with constants, a common challenge in the SR field. Finally, we provide a comprehensive scalability analysis across four key dimensions: (i) expression length, (ii) number of variables, (iii) number of domains, and (iv) number of observed data points.

1 INTRODUCTION

In Machine Learning, many models are designed to achieve low training error and perform well in unseen but similar data. Yet, fitness to data is not the only important attribute. Some applications require interpretability: models must be meaningful in terms of familiar constructs. Another desirable quality is to have Out-Of-Distribution (OOD) generalization. In this context, **Symbolic Regression** (SR) is the task of finding mathematical expressions that fit the data and are as simple as possible. In Physics and other natural sciences, interpretability is commonly accompanied by OOD generalization, as laws of nature have been widely tested. This makes SR a good candidate for finding scientific insight from data. Other areas that can benefit from SR include medicine and finance (Jobin et al., 2019; Rudin, 2019), which are critical and high-stakes.

Formally, given a *domain* set of data points $\mathcal{D} := \{(\mathbf{x}_i,\, v_i)\}_{1 \leq i \leq n}$ consisting of paired features $\mathbf{x}_i$ and target values v_i, the goal of SR is to find a mathematical expression E such that $E(\mathbf{x}_i) \approx v_i$ and E is as simple as possible (e.g. it has a small number of symbols from a pre-defined vocabulary). In the case where an *exact* solution F exists, it is required that $E \equiv F$ up to some tolerance on constant values that may appear (e.g. $1.5x \cdot x + 2.0001$ and $1.4999x^2 + 2$ may be considered equivalent).

In this paper, we present **HTSSR: HeurisTic beam Search Symbolic Regression**, a new method for SR that learns, in a self-supervised way, a precedence relation among expressions to guide a beam search algorithm. We detail key design choices that make our results possible, investigate the scalability of the search and its sample efficiency, and compare HTSSR against existing methods on SRBench (Cava et al., 2021).

2 RELATED WORK

Genetic Programming (GP) was the first note-worthy way to approach SR and many SR methods fall into this category. Early works include (Koza, 1989; 1990), which deal with Program Synthesis, a superclass of SR in a sense. More recent applications of GP to SR are (Keijzer, 2003; Vladislavleva et al., 2009; Schmidt & Lipson, 2009; Korns, 2011; Uy et al., 2011; Jin et al., 2020). GP techniques are known to be easily parallelized and have high parallelism, allowing for the evaluation of a high number of expressions. One downside of GP methods is that they are not robust in cases involving hyperparameters (Petersen et al., 2021). Hybrid approaches, like those proposed in (Mundhenk et al., 2021; Kamienny et al., 2023), combine Deep Learning and GP by letting one or more learned models perform sub-tasks of the GP search, like population seeding, mutation, and selection. (Mundhenk et al., 2021) combines GP with Deep Learning by seeding the GP search with expressions from the learned model. In principle, the learned models help guide GP to more promising regions in search space. Similarly to (Petersen et al., 2021), the model is trained with Reinforcement Learning with the reward signal based on the fitness to data. A clear disadvantage is that a supervision/reward signal based on numerical fit means very different things depending on the context. For instance, the same numerical error may come from a candidate solution that is very close or very far in the space of discrete expressions. In contrast, the supervision of our heuristic model is a simple binary value indicating a “precedence” relation between pairs of elements, having simple optimization and using well-stabilished binary cross-entropy loss.

When it comes to **datasets and benchmarks**, possibly the most well-known effort to standardize SR evaluation is SRBench (Cava et al., 2021). It contains more than 250 problems with and without ground-truth formulas. At least 14 methods have already been tested and compared (Makke & Chawla, 2024). SRBench includes the Strogatz (Strogatz, 2024) and Feynman (Feynman et al., 2011; Udrescu & Tegmark, 2020) problem sets, the latter having some of the original physical laws removed. Other problem sets for SR with ground-truth are available in (Keijzer, 2003; Vladislavleva et al., 2009; Uy et al., 2011; Korns, 2011; Petersen et al., 2021), but they are not physics-related.

The application of **Deep Learning** to SR has earlier examples like (Kusner et al., 2017; Sahoo et al., 2018; Alaa & van der Schaar, 2019). The work (Udrescu & Tegmark, 2020) is possibly the first to show notable progress of Deep Learning in SR. It approaches Symbolic Regression mostly by simplifying a problem into subproblems. (Cranmer et al., 2020; Bendinelli et al., 2023) also allows for the inclusion of simplifying assumptions or prior knowledge. Even though problem simplification should be used in expression discovery, it needs domain-specific knowledge and even so there is always some remaining search space of possible solutions. Instead, we focus on the **search guidance** approach and let problem simplification for further study.

Under the umbrella of Deep Learning, some more recent works employ **Self-Attention** (Vaswani et al., 2017) architecture (d’Ascoli et al., 2024; Shojaee et al., 2023; Kamienny et al., 2023; Lalande et al., 2023; Valipour et al., 2021). Even though we do use Self-Attention layers to process expressions as sequences, we do not use those as a generative model.

(Hayes et al., 2025) introduces a framework based on **neural guided search** where the core model can be trained (and fine-tuned) in different ways, including with Reinforcement Learning, Expert Imitation Learning, and pre-training with synthetic data. The guiding generative model outputs a distribution over tokens, while in our method there is no explicit distribution over tokens. Instead, the output of the model is a score that can be used to prioritize elements in the search. Also, the expression generation in our method is independent of any parametric model: it happens by applying pre-defined grammar-like generation rules and is very fast by means of its simplicity.

Most of the methods try to solve SR in two steps: (i) finding “skeletons” of expressions and (ii) optimizing for the constants. For instance, (Biggio et al., 2021) applies self-supervised training to sample **expression skeletons** with beam-search (not to be confused with the beam-search of our method, where each element is an entire expression instead of a token) and then solves for the constants. Differently from this approach, (Kamienny et al., 2022) tries to infer both **expression structure and constants** directly with a Transformer-based model, using the inner optimization just for small adjustments. Notice that depending on the choice and configuration of optimizer for the inner constant fitting, the whole search process may be severely affected.

The reporting of SR results still needs adherence to standardization. For instance, in (Biggio et al., 2021; Kamienny et al., 2022) authors report metrics based on $R^2 > 0.99$ as a proxy for symbolic solution on the Feynman problem subset from SRBench. As pointed out in (Matsubara et al., 2023), R^2-based accuracy does not take expression interpretability into account and is vulnerable to the use of “dummy” variables. Also, the R^2-criteria changes from work to work, sometimes being $R^2 > 0.9$ (d’Ascoli et al., 2024), $R^2 > 0.99$ (Kamienny et al., 2022), (Kamienny et al., 2023), (Shojaee et al., 2023), while SRBench requires $R^2 > 0.999$. We stick to the Symbolic Solution Rate (SSR) defined in SRBench (Cava et al., 2021) as the default metric.

Table 1: Example of primitives with the respective generation rules.

Symbol	Rule
$\square$	$x \mapsto \square$
y	$x \mapsto y$
$+$	$x \mapsto +xx$
$-$	$x \mapsto -xx$
$\cdot$	$x \mapsto \cdot xx$
$\sqrt{}$	$x \mapsto \sqrt{}x$

Table 2: Expressions and respective prefix forms.

Expression	Prefix Form
$x + \sqrt{y}$	$+x\sqrt{}y$
$\square + x \cdot y$	$+\square \cdot xy$
$x - (y + \square)$	$-x + y\square$

3 HTSSR: HEURISTIC BEAM SEARCH SYMBOLIC REGRESSION

Understanding the following components of our method is necessary for its comprehension. The basic constructs are the set of primitives and the generation rules. Then, expressions can be generated or randomly sampled with the rollout strategy. This generation procedure is at the core of the training data synthesis. That given, some care needs to be taken when evaluating the expressions numerically and feeding the heuristic neural network with such values.

3.1 PRIMITIVES AND GENERATION RULES

The mathematical expressions in this study are a combination of symbols, namely operators (unary and binary), variables, and constants. We call the set of all symbols the *primitives* set. Optionally, that set can be enriched with complexity constraints that tell the maximum allowed occurrences of a symbol under another symbol (e.g., at most 0 cosine operations inside a cosine). This controls the appearance of bizarre expressions and reduces the search space size. All considered expressions have a syntactical tree structure and are implemented using prefix notation. This choice of implementation allows for the fast generation, evaluation, and automatic differentiation of expressions.

Generation rules are defined in terms of the primitive symbols and their arities. One of the variables, x, is considered to be the special symbol used for rule applications. The generation rules have one of three forms: $x \mapsto o_2xx$, $x \mapsto o_1x$, and $x \mapsto o_0$. The o_i indicate an operator with arity i. o_0 can be a variable name, including x, or the constant placeholder, $\square$. Multiple appearances of $\square$ represent independent constants. Tables 1 and 2 show examples of primitive sets, generation rules, and expressions with prefix forms.

3.2 EXPRESSION ROLLOUTS AND CANONICAL DATASET

Instead of working with a static dataset, we find it better to synthesize the expressions during the training of the heuristic model. The expressions are sampled in generation sequences, or *rollouts* (see Figure 1), where a source expression is first sampled from a static *canonical* dataset to then be expanded into increasingly more complex forms. This strategy gives access to a very large set of expressions, even when there are constraints for expression formation.

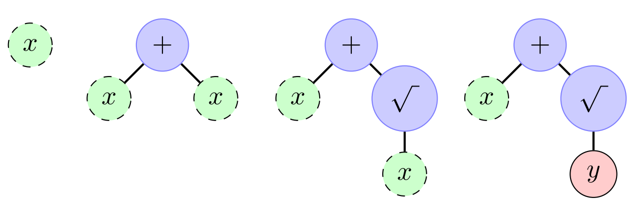


Figure 1: Example rollout from x to $x + \sqrt{y}$. After the rollout is finished, x becomes like any other variable.

(Kamienny et al., 2023) uses a mechanism similar to our "rollouts" in reverse order to generate expressions for training a mutation generative model. This model helps the main GP procedure in the search. Like our method, it is a tree-search but uses MCTS instead of beam-search. Their method combines 3 parameterized models: a mutation policy, a selection policy, and a critic network. Ours, instead, only has one self-supervised model, trained for binary classification.

The canonical dataset contains representatives of the numerical equivalence classes of expressions. The representatives are the smallest elements of a class. We define smallest as having the least number of primitive symbols and being the lexicographically smallest. If the constant placeholder is fixed, computing such a dataset and storing it on disk is possible up to some expression size. This limit also depends on the generation rules and on the primitives.

Uniformly sampling an entire set without considering complexity may underrepresent simpler expressions. We believe that such an imbalance makes the learning process harder. This is the main motivation behind the use of the rollouts. Regarding how the starting points of the rollouts are sampled, we see that sampling (uniformly on length) from canonical sets of different maximum expression lengths shows no significant difference (see Appendix A.2, Figure 9, for an ablation). However, the canonical set is important for the evaluation of the method, as is described in Section 3.6.

3.3 Numeric evaluation

We make extensive use of stack-based evaluation of the expressions in prefix form. Given the limited scope of operations and the small number of variables, this solution is easy to implement and faster than SymPy (Meurer et al., 2017) and isolated Python code calls.

The evaluation in the leaves involves variables and constants. The values attributed to the variables are the *feature domain* $\mathcal{D}[\mathbf{X}]$ - the part of the observed data $\mathcal{D}$ that is not the vector of target values $\mathcal{D}[\mathbf{v}]$. Constant placeholders are sampled from a uniform random distribution or get a fixed value. Our ablation in Appendix A.2, Figure 8, suggests that both choices result in very similar results. Operators get the result of being applied to their arguments. This happens until the top operation is computed.

The numeric results of expressions can easily get out of hand. Common problems are nondetermined (nan), overflow, underflow, and infinite values. To deal with values with large magnitude or that are infinite, we clip at a fake infinite (e.g. $\pm 10^{10}$). Overflows, underflows, and nondetermined results are avoided by the design of safe operators. For instance, a safe division attributes a floating-point number even if the result is not determined in the regular division. When the input domain is well behaved, the safe operators give the exact same results as the regular ones.

When performing prefix-order evaluation, there is a choice between keeping just the final result and also keeping the intermediary results of subexpressions. The last naturally distinguishes different expressions that have equal final values. The first needs some extra information for the distinction, like expression embeddings. We find that training with the first option converges with fewer iterations.

3.4 Constant optimization

The small number of numeric constants that might appear in the expressions works well with second-order optimization methods like Levenberg-Marquadt, taking between 4 and 12 iterations when converging. This is considerably faster than using first-order gradient methods like those based on SGD (Ruder, 2017). Using tools like Pytorch's *autograd* (Paszke et al., 2019), performing such inner optimizations is feasible. Because we implement all the evaluation processes, we can differentiate it with PyTorch.

3.5 The heuristic model

Given an expression E and the observed data $\mathcal{D}$, the heuristic models the probability that there is an expression F such that $V(F)$, the evaluation of F, matches $\mathcal{D}[\mathbf{v}]$ and there is a rollout from E to F. In other words, the heuristic tries to tell if a given expression is in the way of generating (or *precedes*) one expression that fits the data.

The basic architecture (see Figure 2) has two parts: (i) an *encoder* that takes numeric values and outputs latent representations, and (ii) a binary classification module that takes a pair of outputs from the first module (plus some additional information about the potentially preceding expression) to predict the probability that one element precedes the other. The **Sort-Diff** and **Digit** transforms introduced in the following sections are performed in this order, before the parametric part of the encoder.

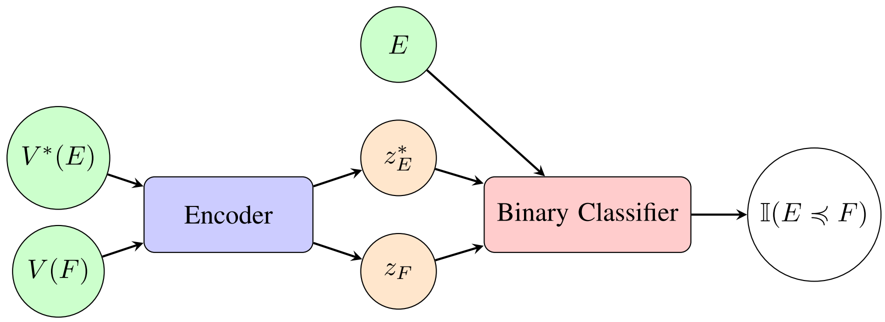


Figure 2: The generic form of the neural networks for the heuristic model. E represents the potentially preceding expression, while $V(F)$ mocks the observed target data $\mathcal{D}[\mathbf{v}]$. The inputs to the encoder are processed independently.

"Sort-Diff" transform. Motivated by the idea that information about the derivatives of the expression value with respect to input variables is helpful to learn the heuristic task, we introduce the Sort-Diff features. Those features consist of sorting $V(E)$ with respect to each input variable and then performing a "diff" operation on the sorted vector. This is supposed to be a surrogate for differentiation and can be applied to data that are not homogeneously sampled (e.g. there is no single step size). Notice that the observed data $\mathcal{D}$ cannot be automatically differentiated. The transformed evaluation vectors are concatenated with the original in a single vector. Equations 1 and 2 define the transform. We get better results when using Sort-Diff, but it still not clear whether it is *because* Sort-Diff or if just the increase in parameters (see Appendix A.2, Figure 7 for an ablation).

$$\mathbf{Diff}(V) := \{V_{i+1} - V_i\}_{0 \,\leq\, i \,<\, |V|} \tag{1}$$

$$\mathbf{SortDiff}(V, x) := \mathbf{Diff}(\{V_i\}_{i \in \mathbf{ArgSort}(x)}) \tag{2}$$

Digit transform. It is known that having high differences in value ranges from feature to feature affects the stability and convergence of optimization during training. Because expression evaluations in SR do suffer from such differences in range, we introduce a transformation that, for every single number, outputs a vector. This vector contains what would be "digits" in a base b representation. For a suitable value of b, each input feature can have a standardized and optimization-friendly range. Equation 3 defines the transform. Here, $a\%b := a - \lfloor a/b \rfloor \cdot b$.

$$\mathbf{DigitTransform}(v) := (v \cdot b^{[-d, -d+1, \ldots, d]})\ \%\ b \tag{3}$$

Common normalization techniques like Min-Max and Mean-Std lose scale information, which is fundamental for the SR task. Transformations that try to make high values more amenable, like taking the logarithm, might "squeeze" values from higher ranges into smaller intervals, making their representations less useful.

Binary classifier. Each pair of outputs from the encoder can be combined in different ways before entering the classifier. In our experiments, the best approach was to take the difference between the latent representations and then add positional encodings and expression embeddings. Because the precedence relation is antisymmetric, the classifier must distinguish different orderings of the input pair. That is why subtracting the latent vectors works better than adding them. The loss function is the Binary Cross-Entropy.

Training with "all-pairs" mini-batches. During training, a set of rollouts is sampled such that the starting points have an equal chance of having any length from 1 to the maximum length of the canonical set. Only starting points are guaranteed to not have a smaller form, up to simplification of constant sub-expressions. Then, when the collection of rollouts reaches a certain number of expressions (e.g., 32), the binary labels (precedes or not) are computed for all ordered pairs of elements. It is easy to do that for pairs of the same rollout, as the expressions that appear first precede the ones that appear later. For pairs of different rollouts, the syntactic trees are compared. We use the convention that any expression precedes itself.

3.6 Synthetic held-out problem sets

For each number of variables $n_{var} \in \{1, 2, 3, 4\}$, a set with 30 expressions for each expression length fom 5 to 10 is created (except for 4 variables, which require at least 7 symbols). Each expression is sampled from the canonical set created with the respective number of variables, but keeping the rest of primitives the same. Unlike rollouts, this sampling is uniform given the number of variables and length. Also, cases where an expression simplifies to a simpler one only happen when the canonical expression has a subexpression of "composite" constants (e.g. $\square \cdot e^{\square}$). When evaluating on these held-out sets, expressions that simplify are counted as having the shorter length. Check Appendix A.9 to see the held-outs.

3.7 Beam Search

The search starts from x and keeps creating new expressions by expanding leaf nodes with x. These expansions are exhaustive: for each combination of x leaf and generation rule, a new expression is formed. It uses the same set of primitives and generation rules used to train the heuristic model.

Each expression is numerically evaluated and fed to the heuristic model. Then, a priority queue receives the expressions with their respective priorities. Whenever an expression without constants (purely operators and variables) is taken from the queue, it is evaluated and compared to the observed values. If all values differ by less than some threshold (e.g. 10^{-8}), it returns the solution. In case the expression has at least one placeholder for constants, a subroutine optimizes for the constants and, if converging, returns the parameter values. The main routine then applies the same acceptance criteria. If a maximum number of expressions is visited, the search stops.

The acceptance criterion is defined in terms of a relative tolerance and the relative squared error between the target $\mathbf{v}$ and the expression evaluation $\hat{\mathbf{v}}$:

$$RSE(\mathbf{v}, \hat{\mathbf{v}}) := \frac{\sum (v_i - \hat{v}_i)^2}{\sum v_i^2} \tag{4}$$

Algorithm 1 synthesizes the high-level workings of the beam-search, given a trained heuristic h_Θ.

4 Experiments

Next, we first analyze HTSSR with respect to sample efficiency and scalability (Sections 4.1 4.2 4.3). Then we show the results of HTSSR on the Feynman and Strogatz problem sets from SRBench (Section 4.4). The scalability experiments show how the Symbolic Solution Rate (SSR) changes given expression length and some other aspects, which are the number of variables and the number of domains $\mathcal{D}$. All experiments have the same set of primitives with 3 variables, differing only in the number of variables when needed (Section 4.2).

The evaluation problem sets used in Sections 4.1 4.2 4.3 are the same described in Section 3.6 and are integrally shown in Appendix A.9. The default domain is `feynman_I_34_1`, appearing other domains only in the experiment from Section 4.3. More details about configuration can be found in Appendices A.5 A.6.

4.1 Sample efficiency

We investigate how search performance changes when changing the availability of data points. The results in Figure 3 support the idea that, under similar conditions, more data points produce better results. Increasing one order of magnitude from 10^2 to 10^3 data points shows little to no gain, while

Algorithm 1 HTSSR, based on beam-search.

$Q \leftarrow [(0, x)]$ ▷ Initialize priority queue
$V \leftarrow \{\ \}$ ▷ Set of visited states
while $length(V) \leq m$ **do** ▷ Maximum of m visited states
  if $length(Q) = 0$ **then**
    return ▷ No solution found
  end if
  $B \leftarrow [Q.pop(), ..., Q.pop()]$ ▷ Beam size pops while not empty
  for $s, E \in B$ **do** ▷ Iterate through priority-expression pairs
    if $\square \notin E$ and $Accept(Eval(E), \mathcal{D})$ **then**
      return E ▷ Constant-free solution found
    else if $\square \in E$ **then**
      $\xi \leftarrow LM(E, \mathcal{D})$ ▷ Run Levenberg-Marquadt optimization
      if $Accept(Eval(E), \mathcal{D}, \xi)$ **then**
        return E, ξ ▷ Solution with constant(s) found
      end if
    end if
    $C \leftarrow Expand(E)$ ▷ Get the set of children expressions
    $S \leftarrow 1 - \sigma(h_\Theta(C, \mathcal{D}))$ ▷ Attribute priority scores with the learned heuristic, h_Θ
    $Q.push((S, C))$ ▷ Update the priority queue
    $V.add(E)$
  end for
end while

increasing from 10^1 to 10^2 shows clear gains. Importantly, HTSSR can find solutions with as few as 10 points, which supports the idea that the method has potential in a data-scarce scenario. The *dummy* baseline shows the raw "brute force" capacity of the search, where the heuristic is clueless but still can find some simple expressions under the imposed conditions.

As in the case of Sort-Diff, the increase in SSR is not necessarily *because* of more data. Having more data points makes a larger input layer in the numeric encoder and this also needs to be taken into account. For example, from 10^2 to 10^3 data points, the model increases 176% in the number of parameters.

The fact that this evaluation is over the single `feynman_I_34_1` domain raises the question of how different the behavior shown in Figure 3 would be with different domains. Regarding this, we think that because the overall difficulty of the task depends both on the data and on the expressions to be discovered, this experiment covers an important part of the investigation.

Other question that can be raised is about what happens after length 10. To that, under the same resource constraints, the tendency is indeed to fall to zero SSR shortly after 10. Repeating this experiment with more time, compute, and more degrees of magnitude in data points would certainly improve the analysis.

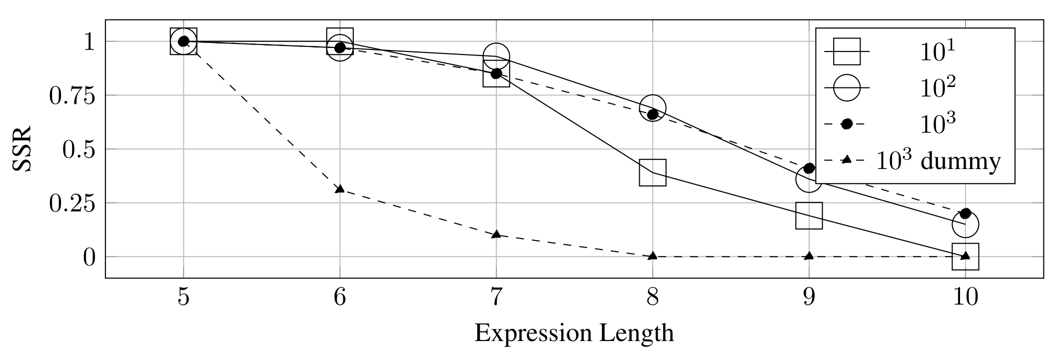


Figure 3: Symbolic Solution Rate (SSR) versus expression length for sample sizes 10^1, 10^2, 10^3, and a non-trained baseline. $n_{var} = 3$.

4.2 SCALABILITY: NUMBER OF VARIABLES

Now we investigate the impact that n_{var} has on the SSR. From Figure 4, it looks like the expression length plays a more important role in the decay of the SSR than the number of variables. Only for $n_{var} = 4$ versus $n_{var} < 4$ is there a clear sign of degradation for expression length greater than 7. Furthermore, it seems that the decay of SSR for $n_{var} = 1$ is "slower" at larger lengths. It could be that for $n = 1$ it is possible to find solutions larger than 10 symbols somewhat frequently.

Given that the lines for each n_{var} look very entangled, a larger sample of expressions in each combination of length and n_{var} might produce clearer tendencies. Although we sampled 30 expressions for each length and n_{var}, the cases where "composite" constants form and simplify expressions reduce the number of expressions that really have higher lengths.

Still, the "hardness" of finding expressions is being evaluated under a fixed set of primitives. Changing the set of primitives may produce considerably different results, even if the total number of elements in the primitives is kept the same.

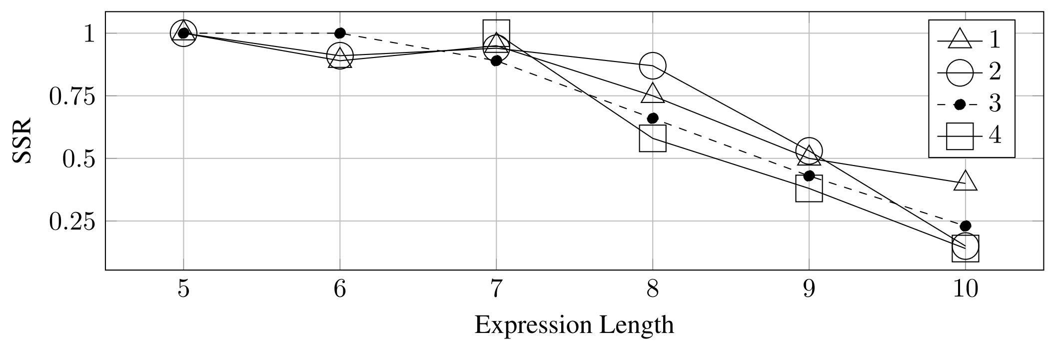


Figure 4: Symbolic Solution Rate (SSR) versus expression length for $1 \leq n_{var} \leq 4$.

4.3 SCALABILITY: NUMBER OF DOMAINS

Figure 5 indicates that increasing the number of domains in which a single heuristic model is trained (using learnable domain embeddings) degrades its quality on the evaluation domain `feynman_I_34_1`, also seen during training, at least for $n_{var} = 1$ versus $n_{var} > 1$. However, among $n_{var} > 1$, the degradation of SRR is relatively small, if any. This might indicate potential for reusability of the heuristic model, as one single model could be used for many data domains. A better way to test potential for reusability would be to let variables assume any set of values, but we cannot find such a general model at the present moment. It would need to be invariant to the order of input samples and robust to small differences in the set of samples (e.g. sets differing by a small number of elements), all of this while keeping relevant information.

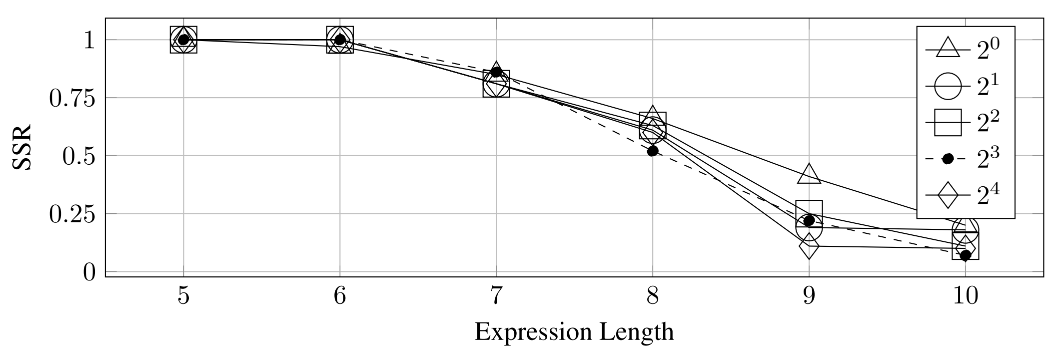


Figure 5: Symbolic Solution Rate (SSR) versus expression length for 1 to 16 domains in the same heuristic model. $n_{var} = 3$.

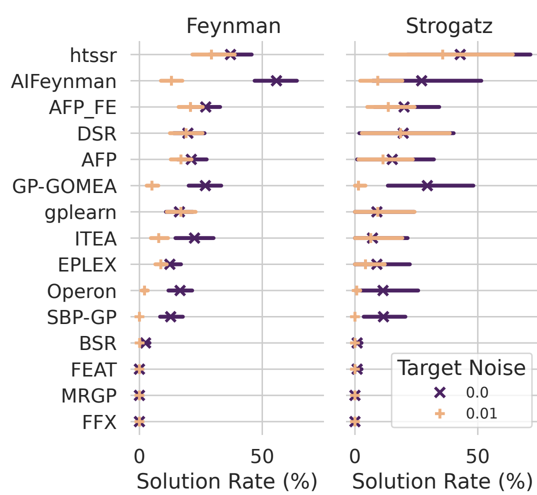
4.4 SRBench: Feynman and Strogatz problem sets

We run HTSSR on the Feynman and Strogatz problem sets under the constraints of SRBench for "ground-truth problems". There are training time limits of 36000 and 3600 seconds for each problem in Feynman and Strogatz, respectively. Within the training time budget, model checkpoints at different epochs are used to search. Expressions that do not meet HTSSR's acceptance criteria are not returned as approximate solutions. See Appendix A.6 for details and Appendix A.8 for the lists of solutions.

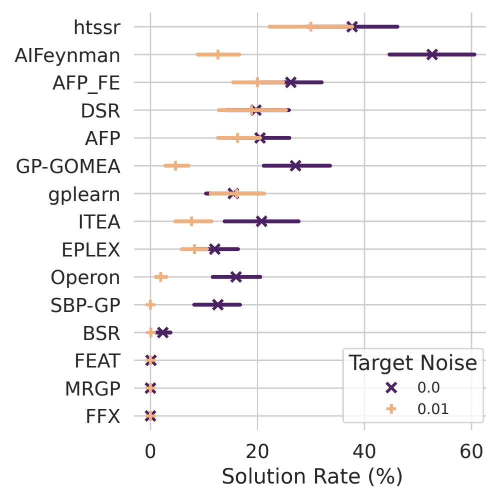
Unlike the other methods in Figure 6, HTSSR has only one trial per combination of problem and noise level, not 10. Therefore, the confidence intervals (95%) suggested by the horizontal bars tend to be narrower for the other methods. The noise levels considered are $[0.0, 0.01]$. Refer to (Cava et al., 2021) for the exact meaning of the noise levels in SRBench.

Even considering the confidence intervals, HTSSR is likely among the top methods and is possibly the best at the Strogatz set. Compared with the other methods, it has a small degradation in performance when the noise level increases. On Strogatz, HTSSR with 0.01 noise level surpasses the other methods with 0.0 noise (using the middle of CIs as reference). Also, the performance of HTSSR is consistent when changing problem sets, as it does not make specific assumptions about the problems. In principle, HTSSR could score higher if helped with problem simplification or a "divide-and-conquer" approach, where a problem is decomposed into sub-problems.

In this experiment, the same primitives are used for both Feynman and Strogatz problems, but the expressions from Strogatz use a smaller set of symbols. So, in principle, HTSSR applied to the least sufficient set could reach more intricate expressions from Strogatz, but this would have to be checked empirically.



(a) SSR grouped by problem set.



(b) Aggregated SSR results.

Figure 6: Comparison with the SRBench results of other methods.

5 Conclusion

This paper presents a new and simple method for SR with key innovations, making a shift from common approaches in the literature that use fitness to data as training signal or that explore the space of solutions on a token-by-token manner. It finds solutions with desired properties, such as exactness and simplicity, while being competitive with existing methods and less affected by noise. We also analyze some aspects of the scalability and sample efficiency of the algorithm, getting insight into further investigations and improvements. We find that the major factors that affect the effectiveness of the method seem to be the length of expressions and the number of data points.

6 REPRODUCIBILITY

We plan to soon release a refactored version of the code and instructions to the public. As of now, code and instructions are available as suplementary material for the reviewers in the reviewing platform. Each experiment ran on a NVIDIA A100-80GB GPU with single process at a maximum 2.2 GHz processor core.

REFERENCES

Ahmed M. Alaa and Mihaela van der Schaar. Demystifying black-box models with symbolic metamodels. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/567b8f5f423af15818a068235807edc0-Paper.pdf`.

Tommaso Bendinelli, Luca Biggio, and Pierre-Alexandre Kamienny. Controllable neural symbolic regression. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Luca Biggio, Tommaso Bendinelli, Alexander Neitz, Aurelien Lucchi, and Giambattista Parascandolo. Neural symbolic regression that scales. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 936–945. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/biggio21a.html`.

W. L. Cava, Patryk Orzechowski, Bogdan Burlacu, Fabr'icio Olivetti de Francca, M. Virgolin, Ying Jin, Michael Kommenda, and Jason H. Moore. Contemporary symbolic regression methods and their relative performance. *Advances in neural information processing systems*, 2021 DB1:1–16, 2021. URL `https://api.semanticscholar.org/CorpusID:236635250`.

Miles Cranmer, Alvaro Sanchez-Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Stéphane d'Ascoli, Sören Becker, Philippe Schwaller, Alexander Mathis, and Niki Kilbertus. ODEFormer: Symbolic regression of dynamical systems with transformers. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=TzoHLiGVMo`.

Richard P. Feynman, Robert B. Leighton, and Matthew Sands. *The Feynman Lectures on Physics, Vol. I: The New Millennium Edition: Mainly Mechanics, Radiation, and Heat*. Basic Books, 2011. ISBN 978-0465024933.

Conor F. Hayes, Felipe Leno Da Silva, Jiachen Yang, T. Nathan Mundhenk, Chak Shing Lee, Jacob F. Pettit, Claudio Santiago, Sookyung Kim, Joanne T. Kim, Ignacio Aravena Solis, Ruben Glatt, Andre R. Goncalves, Alexander Ladd, Ahmet Can Solak, Thomas Desautels, Daniel Faissol, Brenden K. Petersen, and Mikel Landajuela. Deep symbolic optimization: Reinforcement learning for symbolic mathematics, 2025. URL `https://arxiv.org/abs/2505.10762`.

Ying Jin, Weilin Fu, Jian Kang, Jiadong Guo, and Jian Guo. Bayesian symbolic regression, 2020. URL `https://arxiv.org/abs/1910.08892`.

Anna Jobin, Marcello Ienca, and Effy Vayena. The global landscape of ai ethics guidelines. *Nature Machine Intelligence*, 1, 09 2019. doi: 10.1038/s42256-019-0088-2.

Pierre-Alexandre Kamienny, Stéphane d'Ascoli, Guillaume Lample, and François Charton. End-to-end symbolic regression with transformers. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Pierre-Alexandre Kamienny, Guillaume Lample, Sylvain Lamprier, and Marco Virgolin. Deep generative symbolic regression with monte-carlo-tree-search. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Maarten Keijzer. Improving symbolic regression with interval arithmetic and linear scaling. In Conor Ryan, Terence Soule, Maarten Keijzer, Edward Tsang, Riccardo Poli, and Ernesto Costa (eds.), *Genetic Programming*, pp. 70–82, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-36599-0.

Michael F. Korns. *Accuracy in Symbolic Regression*, pp. 129–151. Springer New York, New York, NY, 2011. ISBN 978-1-4614-1770-5. doi: 10.1007/978-1-4614-1770-5_8. URL https://doi.org/10.1007/978-1-4614-1770-5_8.

John R. Koza. Hierarchical genetic algorithms operating on populations of computer programs. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'89, pp. 768–774, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

J.R. Koza. Genetically breeding populations of computer programs to solve problems in artificial intelligence. In *[1990] Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence*, pp. 819–827, 1990. doi: 10.1109/TAI.1990.130444.

Matt J. Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1945–1954. JMLR.org, 2017.

Florian Lalande, Yoshitomo Matsubara, Naoya Chiba, Tatsunori Taniai, Ryo Igarashi, and Yoshitaka Ushiku. A transformer model for symbolic regression towards scientific discovery. In *NeurIPS 2023 AI for Science Workshop*, 2023. URL https://openreview.net/forum?id=AIfqWNHKjo.

Nour Makke and Sanjay Chawla. Interpretable scientific discovery with symbolic regression: a review. *Artif. Intell. Rev.*, 57(1), January 2024. ISSN 0269-2821. doi: 10.1007/s10462-023-10622-0. URL https://doi.org/10.1007/s10462-023-10622-0.

Yoshitomo Matsubara, Naoya Chiba, Ryo Igarashi, Tatsunori Taniai, and Yoshitaka Ushiku. Rethinking symbolic regression datasets and benchmarks for scientific discovery, 2023. URL https://openreview.net/forum?id=i2e2wqt0nAI.

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, January 2017. ISSN 2376-5992. doi: 10.7717/peerj-cs.103. URL https://doi.org/10.7717/peerj-cs.103.

T. Nathan Mundhenk, Mikel Landajuela, Ruben Glatt, Claudio P. Santiago, Daniel M. Faissol, and Brenden K. Petersen. Symbolic regression via neural-guided genetic programming population seeding. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA, 2019.

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.

Sebastian Ruder. An overview of gradient descent optimization algorithms, 2017. URL https://arxiv.org/abs/1609.04747.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, May 2019. doi: 10.1038/s42256-019-0048-x. Epub 2019 May 13.

Subham Sahoo, Christoph Lampert, and Georg Martius. Learning equations for extrapolation and control. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4442–4450. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/sahoo18a.html.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009. doi: 10.1126/science.1165893. URL https://www.science.org/doi/abs/10.1126/science.1165893.

Parshin Shojaee, Kazem Meidani, Amir Barati Farimani, and Chandan K. Reddy. Transformer-based planning for symbolic regression. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=0rVXQEeFEL.

S.H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. CRC Press, 2024. ISBN 9780429676284. URL https://books.google.com.br/books?id=1wrsEAAAQBAJ.

Silviu-Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020. doi: 10.1126/sciadv.aay2631. URL https://www.science.org/doi/abs/10.1126/sciadv.aay2631.

Nguyen Quang Uy, Nguyen Xuan Hoai, Michael O'Neill, R. I. Mckay, and Edgar Galván-López. Semantically-based crossover in genetic programming: application to real-valued symbolic regression. *Genetic Programming and Evolvable Machines*, 12(2):91–119, June 2011. ISSN 1389-2576. doi: 10.1007/s10710-010-9121-2. URL https://doi.org/10.1007/s10710-010-9121-2.

Mojtaba Valipour, Bowen You, Maysum Panju, and Ali Ghodsi. Symbolicgpt: A generative transformer model for symbolic regression. *ArXiv*, abs/2106.14131, 2021. URL https://api.semanticscholar.org/CorpusID:235658383.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Ekaterina J. Vladislavleva, Guido F. Smits, and Dick den Hertog. Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. *IEEE Transactions on Evolutionary Computation*, 13(2):333–349, 2009. doi: 10.1109/TEVC.2008.926486.

A Appendix

A.1 LLM Usage

In this work, LLMs helped to find typos and suggested words in a minority of cases.

A.2 Ablations

Sort-Diff. In Figure 7 there is a clear pattern that shows the superiority of applying the Sort-Diff transform to input features versus not. However, it is true that with Sort-Diff the input layer of the numeric encoder is larger and therefore the model has more parameters. That difference only occurs at the first layer and, if we take $n_{var} = 3$, the addition in number of parameters is in the order of $3 \cdot m \cdot n$, where m and n are the dimensions of the input (without Sort-Diff) and output of the first layer, respectively. This is around 412 million parameters **increase** over the original 194 million **total**, or roughly 213% increase. This suggests that the number of parameters may play a more important role in this case than the Sort-Diff transform itself.

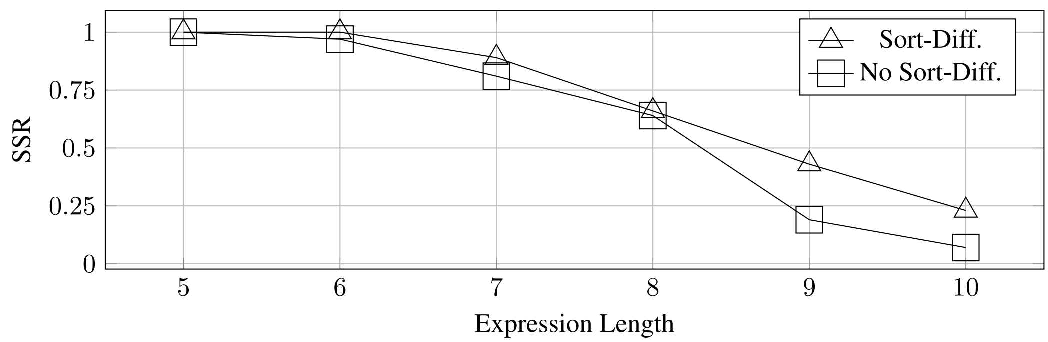


Figure 7: Symbolic Solution Rate (SSR) versus expression length for model with and without the $\mathbf{SortDiff}$ transform. $n_{var} = 3$.

Evaluation of the constant placeholder. Figure 8 shows very close tendencies when comparing the SSR resulting from heuristics trained with a fixed value $v_\square$ versus the sampled value $v_\square + U(-0.1, 0.1)$. The motivation behind this experiment is to see if sampling $\square$ improves the ability of the heuristic model to perform well for expressions with constants that are not seen during training. The results have only small, opposite differences at the lengths 9 and 10 and suggest that no difference is revealed.

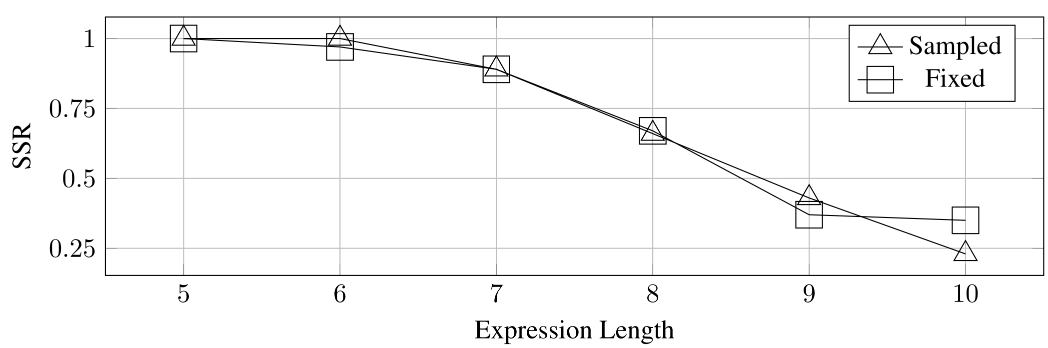


Figure 8: Symbolic Solution Rate (SSR) versus expression length for $\square$ sampled versus fixed during training. $n_{var} = 3$.

Maximum size in the canonical set. In Figure 9 there is a comparison between the SSRs resulting from heuristics trained by sampling the starting points of rollouts from canonical datasets of different sizes. The idea of using canonical datasets to "anchor" the sampling is that it would make the mini-batches more balanced with respect to expression length. This in turn would result in better heuristic models. However, the results do not indicate such improvement. In part, this could be because the rollouts naturally create expressions with varying complexities, and the expressions that simplify are not sufficient to impact the representation of larger expressions negatively. On the other hand, the larger number of longer expressions do not affect the representation of smaller ones because of the nature of rollouts.

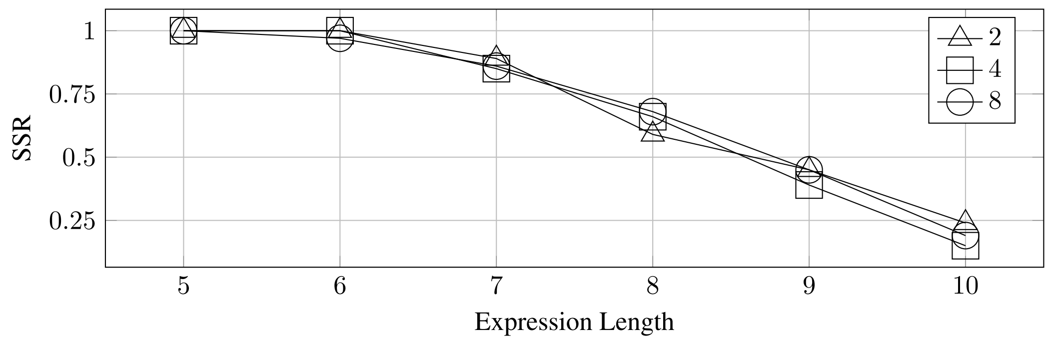


Figure 9: Symbolic Solution Rate (SSR) versus expression length for canonical sets with maximum expression lengths 2, 4, and 8. $n_{var} = 3$.

A.3 Primitives Set

Table 3 shows the set of primitive symbols. Table 4 shows the constraints used in the formation of expressions for the experiments.

Table 3: Constants, variables, and operators used in the experiments. $\arcsin$ was used only in the SRBench experiment (Section 4.4).

Symbol	$\square$	x	y	z	w	$+$	$-$	$\cdot$	$/$	$\cdot^2$	$\sqrt{\cdot}$	sin	cos	$e^{\cdot}$	arcsin
Arity	0	0	0	0	0	2	2	2	2	1	1	1	1	1	1

Table 4: Constraints for the formation of expressions. Row elements can appear up to the specified number of times under the column element in the expression syntactic tree. Empty cells indicate no constraint.

	$+$	$-$	$\cdot$	$/$	$\cdot^2$	$\sqrt{\cdot}$	sin	cos	$e^{\cdot}$	arcsin
$\square$	2	2	2	2	2	2	2	2	2	2
x										
y	2	2	2	2	2	2	2	2	2	2
z	2	2	2	2	2	2	2	2	2	2
w	2	2	2	2	2	2	2	2	2	2
$+$										
$-$										
$\cdot$										
$/$										
$\cdot^2$					0	0	0	0	1	0
$\sqrt{\cdot}$						0	0	0	0	0
sin							0	0	0	
cos							0	0	0	0
$e^{\cdot}$							0	0	0	0
arcsin										0

A.4 Algorithm for Creating Canonical Sets

Algorithm 2 is a simplified version of the implementation for creating canonical sets of expressions.

Algorithm 2 Creation of canonical set of expressions up to length n.

$S \leftarrow O_0$ ▷ Initialize canon set with zero-ary elements.
$V \leftarrow \{ \}$ ▷ Initialize visited values.
for $2 \leq l \leq n$ **do** ▷ Iterate from lengths 2 to n.
  for $o \in O_1$ **do** ▷ For each unary operator
    for $F \in S_{l-1}$ **do** ▷ For each expression in S with length $l-1$
      $E \leftarrow o(F)$ ▷ Create new expression of length l
      if $\textbf{Eval}(E) \notin V$ **then** ▷ Add only if a smaller one is not equivalent.
        $S.append(E)$
        $V.add(\textbf{Eval}(E))$
      end if
    end for
  end for
  for $o \in O_2$ **do** ▷ For each binary operator
    for $1 \leq l^L \leq n-2$ **do** ▷ For each length of the left subtree
      $l^R \leftarrow n - 1 - l^L$
      for $F^L \in S_{l^L}$ **do**
        for $F^R \in S_{l^R}$ **do**
          $E \leftarrow o_2(F^L, F^R)$
          if $\textbf{Eval}(E) \notin V$ **then**
            $S.append(E)$
            $V.add(\textbf{Eval}(E))$
          end if
        end for
      end for
    end for
  end for
end for
return S

A.5 Setups for the scalability and sample efficiency experiments

Every heuristic model in that part of the experiments was trained for 1000 "epochs" of 50 iterations each. The mini-batches were "all-pairs" of size 32×32. The beam search window is 128 and the limit of visited states is 10240. Except for the multi-domain experiment, the default domain used is from the problem `feynman_I_34_1`, with the extra fourth variable being sampled from $U(1, 5)$. $\mathcal{D}$ is randomly sub-sampled from 10^5 to 10^3 data points (and to $\{10^2, 10^1\}$ in Section 4.1). Details about the neural net configuration are in Appendix A.7.

A.6 Setups for the SRBench experiment

The general process for searching for a solution of a given problem starts by training the heuristic model. Training is interrupted at defined epochs ($[599, 999]$ fpr Strogatz and $[99, 199, 399, 599, 799]$ for Feynman) so that the current checkpoint is used by HTSSR to search. The beam of the search is 16384 for Strogatz and 8192 for Feynman. The limit of visited states is 102400 for both. If no solution is found but there is still time remaining, the checkpoint goes back to training. The search ends if a solution is found or if time is out.

The relative tolerance to accept a candidate solution and stop the search is 10^{-3}. We choose this threshold in order to conform to both levels of noise, $[0.0, 0.01]$, but it can be orders of magnitude lower (e.g. 10^{-8}) in the absence of noise.

Other configuration and neural net structure are described in Appendix A.7.

A.7 Neural Net Architecture

Table 5 shows the main neural net configuration used across experiments. The main difference between experiments is at the first layer, as the number of input units is different between problem sets ($10, 100, 1000$ for Feynman, 300 for Strogatz). In the Self-Attention layers, $d_{model} = 1024$ for all experiments except for the SRBench experiment, where $d_{model} = 768$. The "Linear" layers in the numerical encoder have standard 2048 hidden-layer "width", with final layer width being d_{model}. The exception is for the SRBench experiment, where those hidden Linear layers have width 1024. In the Digit Transform, all experiments use 67 "digits" in base 2, with position values from 2^{-33} to 2^{33}.

Table 5: General Neural Net Configuration for the Experiments.

Module	**Submodules**
Numeric Encoder	SortDiff (optional) Digit Transform $3\times$ (Linear, RMSNorm, GELU)
Source-Target aggregation	$-$ (difference) Final result or "all-tree" results
Positional Encoding Positional Encoding (parent symbol) Expression Embeddings Domain Embeddings	$+$ (padded to length 15) $+$ (optional) $+$ (optional) $+$ (optional)
Classification	$4\times$ Self-Attention (4 heads) Sequence aggregation (mean) Linear

A.8 Symbolic Solutions

Tables 6 7 8 9 list solutions found for the SRBench experiment.

A.9 Expression held-outs

The following Tables 10 11 12 13 14 15 16 17 contain the held-out problem sets used in the experiments (except SRBench). Those are grouped by n_{var}.

Dataset	Training Time (s)	Solution	Ground-Truth
strogatz_barmag2	3489	sin(y - x) * 0.5 - sin(y)	0.5 * sin(y - x) - sin(y)
strogatz_glider1	703	x**2 / (-20.0) - sin(y)	-0.05 * x**2 - sin(y)
strogatz_glider2	43	x - cos(y) / x	x - cos(y) / x
strogatz_lv2	630	((2.0 - x) - y) * y	2 * y - x * y - y ** 2
strogatz_shearflow1	566	(cos(x) * cos(y)) / sin(y)	cot(y) * cos(x)
strogatz_vdp2	276	(-0.1) * x	-(1/10) * x

Table 6: Ran on Strogatz with 0.0 noise level under 3600 seconds for each problem.

Dataset	Training Time (s)	Solution	Ground-Truth
strogatz_glider1	2004	((x * x) * (-0.0499866)) - sin(y)	-0.05 * x**2 - sin(y)
strogatz_glider2	1604	x - cos(y) / x	x - cos(y) / x
strogatz_lv2	2554	(2.0004959 - (x + y)) * y	2 * y - x * y - y ** 2
strogatz_shearflow1	2083	(cos(x) * cos(y)) / sin(y)	cot(y) * cos(x)
strogatz_vdp2	570	x * (-0.0999554)	-(1/10) * x

Table 7: Ran on Strogatz with 0.01 noise level under 3600 seconds for each problem.

Dataset	Time (s)	Solution	Ground-Truth
I_25_13	27	q / C	q / C
I_12_5	27	q2 * Ef	q2 * Ef
I_12_1	27	mu * Nn	mu*Nn
I_29_4	27	omega / c	omega/c
I_34_27	45	(h * omega) / 6.2831855	(h/(2*pi))*omega
I_39_1	49	pr * V * 1.5	3/2*pr*V
I_14_3	59	m * g * z	m*g*z
I_43_31	59	mob * T * kb	mob*kb*T
III_12_43	61	h / (6.2831855 / n)	n*(h/(2*pi))
I_6_2a	108	0.3989423 / sqrt(exp(theta**2))	exp(-theta**2/2)/sqrt(2*pi)
I_18_12	117	r * sin(theta) * F	r*F*sin(theta)
I_14_4	130	(k_spring * x)**2 / (k_spring + k_spring)	1/2*k_spring*x**2
II_8_31	131	(epsilon * Ef)**2 / (epsilon + epsilon)	epsilon*Ef**2/2
II_3_24	145	(Pwr * 0.0795775) / r**2	Pwr/(4*pi*r**2)
II_27_16	164	c * Ef**2 * epsilon	epsilon*c*Ef**2
II_37_1	167	(mom * chi + mom) * B	mom*(1+chi)*B
I_47_23	172	sqrt(gamma * pr / rho)	sqrt(gamma*pr/rho)
II_38_3	179	Y * A * x / d	Y*A*x/d
II_10_9	221	sigma_den / (epsilon + (epsilon * chi))	sigma_den/epsilon*1/(1+chi)
I_27_6	223	d2 / ((d2 / d1) + n)	1/(1/d1+n/d2)
I_34_14	**225**	**omega_0 * (exp(v / c))**	**(1+v/c)/sqrt(1-v**2/c**2)*omega_0**
II_34_2	290	r / ((2.0 / v) / q)	q*v*r/2
II_34_2a	294	q / ((r / 0.1591549) / v)	q*v/(2*pi*r)
II_34_29a	301	(q * h / m) * 0.0795775	q*h/(4*pi*m)
I_34_1	301	omega_0 / (1.0 - v / c)	omega_0/(1-v/c)
III_15_27	304	((6.2831855 / n) / d) * alpha	2*pi*alpha/(n*d)
III_17_37	304	beta * alpha * cos(theta) + beta	beta*(1+alpha*cos(theta))
II_27_18	309	epsilon * Ef**2	epsilon*Ef**2
I_34_8	314	B * q * v / p	q*v*B/p
II_38_14	320	Y / (sigma + 2.0 + sigma)	Y/(2*(1+sigma))
I_39_22	340	n / (V / (T * kb))	n*kb*T/V
II_4_23	342	(q / epsilon) * (0.0795775 / r)	q/(4*pi*epsilon*r)
I_43_16	344	(q / d) * mu_drift * Volt	mu_drift*q*Volt/d
III_7_38	359	(mom / h) * B * 12.5663710	2*mom*B/(h/(2*pi))
II_15_5	434	-0.0000001 - p_d * cos(theta) * Ef	-p_d*Ef*cos(theta)
II_15_4	445	0.0000001 - (B * cos(theta) * mom)	-mom*B*cos(theta)
I_18_14	453	m * sin(theta) * r * v	m*r*v*sin(theta)
II_8_7	544	0.0477465 * q**2 / (epsilon * d)	3/5*q**2/(4*pi*epsilon*d)
III_15_14	746	(h / d)**2 / (E_n / 0.0126651)	(h/(2*pi))**2/(2*E_n*d**2)
I_12_4	1046	(q1 * 0.0795775 / epsilon) / r**2	q1*r/(4*pi*epsilon*r**3)
I_39_11	3014	(pr * V) / (gamma - gamma / gamma)	1/(gamma-1)*pr*V
II_34_11	3507	q / ((m + m) / (g_ * B))	g_*q*B/(2*m)
I_43_43	3686	kb * v / (gamma * A - A)	1/(gamma-1)*kb*v/A
III_21_20	7591	(0.0 - A_vec * rho_c_0 / m) * q	-rho_c_0*q*A_vec/m

Table 8: Ran on Feynman with 0.0 noise level under 36000 seconds for each problem.

Dataset	Time (s)	Solution	Ground-Truth
I_25_13	20	q / C	q / C
I_12_5	20	q2 * Ef	q2 * Ef
I_12_1	20	mu * Nn	mu*Nn
I_29_4	20	omega / c	omega/c
I_34_27	78	(h / 6.2837) * omega	(h/(2*pi))*omega
I_39_1	**47**	**(pr + sqrt(pr)) * V**	**3/2*pr*V**
I_14_3	78	z * m * g	m*g*z
I_43_31	81	mob * T * kb	mob*kb*T
III_12_43	65	h * n / 6.2822762	n*(h/(2*pi))
I_6_2a	**52**	**exp(theta - (theta * 2.4011037))**	**exp(-theta**2/2)/sqrt(2*pi)**
I_18_12	237	sin(theta) * r * F	r*F*sin(theta)
I_14_4	190	(k_spring * x)**2 / (k_spring + k_spring)	1/2*k_spring*x**2
II_8_31	**192**	**(sqrt(exp(Ef) - epsilon)) * epsilon**	**epsilon*Ef**2/2**
II_3_24	224	Pwr / (r * (-3.5453365))**2	Pwr/(4*pi*r**2)
II_27_16	231	epsilon * c * Ef**2	epsilon*c*Ef**2
II_37_1	230	(mom * chi + mom) * B	mom*(1+chi)*B
I_47_23	241	sqrt(gamma * pr / rho)	sqrt(gamma*pr/rho)
II_38_3	409	Y * A * x / d	Y*A*x/d
II_10_9	348	sigma_den / (epsilon * chi + epsilon)	sigma_den/epsilon*1/(1+chi)
I_27_6	337	d2 / (n + (d2 / d1))	1/(1/d1+n/d2)
II_34_2	2632	0.5001777 * q * v * r	q*v*r/2
II_34_2a	**1623**	**(sqrt(v * q) - 1.4519717) / r**	**q*v/(2*pi*r)**
II_34_29a	1635	(q / m) / (exp(2.5306423) / h)	q*h/(4*pi*m)
I_34_1	**219**	**exp(v / c) * omega_0**	**omega_0/(1-v/c)**
III_15_27	**1846**	**(alpha / n) * exp(2.0399628 / d)**	**2*pi*alpha/(n*d)**
III_17_37	1414	beta + (alpha * beta * cos(theta))	beta*(1+alpha*cos(theta))
II_27_18	22	Ef**2 * epsilon	epsilon*Ef**2
I_34_8	406	(q / (p / B)) * v	q*v*B/p
II_38_14	**248**	**(Y / sqrt(sigma)) * 0.2274444**	**Y/(2*(1+sigma))**
I_39_22	340	n * T * kb / V	n*kb*T/V
II_4_23	1607	((q / r) / 12.5702) / epsilon	q/(4*pi*epsilon*r)
I_43_16	1425	mu_drift * (q / (d / Volt))	mu_drift*q*Volt/d
III_7_38	**1586**	**(((mom + 3.0326998)**2) / h) * B**	**2*mom*B/(h/(2*pi))**
II_15_5	480	p_d - p_d - (p_d * Ef * cos(theta))	-p_d*Ef*cos(theta)
II_15_4	1584	mom - mom - (cos(theta) * B * mom)	-mom*B*cos(theta)
I_18_14	1867	v * sin(theta) * m * r	m*r*v*sin(theta)
II_8_7	1882	(q**2 / exp(3.0413795)) / (epsilon * d)	3/5*q**2/(4*pi*epsilon*d)
III_15_14	11162	(((cos(-4.8251600) * h) / d)**2) / E_n	(h/(2*pi))**2/(2*E_n*d**2)
I_12_4	1713	(q1 / epsilon) / (r / cos(-1.8567311))**2	q1*r/(4*pi*epsilon*r**3)
I_39_11	1503	(pr * V) / (gamma - gamma / gamma)	1/(gamma-1)*pr*V
II_34_11	1928	B * ((g_ * q) / (m + m))	g_*q*B/(2*m)
I_43_43	1839	kb / ((gamma * A - A) / v)	1/(gamma-1)*kb*v/A
III_21_20	**1967**	**(q / m) * (sin(rho_c_0) - rho_c_0 * A_vec)**	**-rho_c_0*q*A_vec/m**

Table 9: Ran on Feynman with 0.01 noise level under 36000 seconds for each problem.

Table 10: Held-out expressions for $n_{var} = 1$. Part 1.

$(e^x - \sqrt{\square})$	$(x \cdot \sqrt{\sin(x)})$	$((x - \cos(\square)))^2$
$(\sqrt{\sin(\square)} - x)$	$((\square + \sqrt{x}))^2$	$\frac{\frac{(x-\square)}{\square}}{(\cos(x))^2}$
$\sqrt{\sin((\square - x))}$	$(\cos(\square) - (x)^2)$	$\frac{x}{\sqrt{\sin(x)}}$
$\frac{x}{(e^x)^2}$	$\frac{\sqrt{e^x}}{\square}$	$((x \cdot \sin(\square)))^2$
$\frac{\square}{\sqrt{(x)^2}}$	$\frac{e^x}{\sin(x)}$	$(x + \cos((\square \cdot x)))$
$(e^{(\square \cdot x)} - x)$	$((x + \frac{x}{\square}))^2$	$(\square + \frac{\square}{\sin(x)})$
$\frac{x}{(\square + e^{\square})}$	$(x \cdot \cos((\square \cdot x)))$	$(\square + ((x)^2 - x))$
$\frac{(\square - x)}{\sqrt{\square}}$	$\sqrt{(\sin(\square) - e^x)}$	$(\sqrt{(x)^2} \cdot e^{\square})$
$(\sqrt{\frac{x}{\sin(x)}})^2$	$((\square)^2 - \sqrt{\sin(x)})$	$(e^x - \sqrt{\sin(\square)})$
$\frac{(x-\square)}{\cos(\square)}$	$\frac{\square}{(\square + e^x)}$	$(\sin(\frac{\square}{x}) + \cos(\square))$
$(x \cdot \frac{\cos(\square)}{e^x})$	$(x - \frac{e^x}{\cos(\square)})$	$\frac{\square}{\sqrt{((x)^2 - \square)}}$
$(\frac{\sin(x)}{\square} - e^x)$	$((e^{\square})^2 - \frac{\square}{x})$	$(e^{(\square + x)} - \sin(x))$
$(\cos(x) - \frac{x}{\sin(x)})$	$(\square + (x + (e^{\square})^2))$	$\frac{\sqrt{x}}{((x)^2 - \square)}$
$(x - \frac{\sqrt{\cos(x)}}{\square})$	$(\square \cdot \sqrt{(\square \cdot (x)^2)})$	$\frac{e^{\square}}{(\square - \sqrt{x})}$
$((x \cdot \sqrt{\square}) - \sqrt{x})$	$\sqrt{(\square \cdot (x + \sin(\square)))}$	$((\frac{x}{\cos(x)} + e^{\square}))^2$
$(\square \cdot \sqrt{(\sin(x) + e^x)})$	$(\sqrt{\sin(\square)} - \frac{x}{\cos(\square)})$	$\sqrt{(\sin(\square) - \sin((\square \cdot x)))}$
$\frac{\frac{(x)^2}{\sqrt{\square}}}{\cos(\square)}$	$(\cos(x) - \frac{\square}{(\sin(\square))^2})$	$(\frac{\sin((\square + \square))}{\sqrt{x}})^2$
$(\sqrt{(\frac{x}{\cos(\square)} - \square)})^2$	$\sqrt{\frac{\square}{(\cos(\frac{x}{\square}))^2}}$	$\frac{x}{((\square + x) \cdot \sqrt{\square})}$
$((\frac{\square}{x} + \frac{x}{\square}))^2$	$\frac{(e^x - (\cos(\square))^2)}{\square}$	$(\frac{\square}{\cos(x)} + (\sqrt{x})^2)$
$\sqrt{(\square \cdot \frac{\cos(x)}{\sin(x)})}$	$(x \cdot \sin((\square - (\square \cdot x))))$	$((\frac{x}{(x+x)} - \sqrt{x}))^2$
$(\sqrt{x} + \cos(\frac{x}{(x+x)}))$	$\frac{(\cos(\square))^2}{(\square + (\sqrt{x})^2)}$	$(((x + \sqrt{\sin(\square)}) \cdot \sqrt{x}))^2$
$(((\square + x) \cdot e^{\square}) + \cos(x))$	$(\square \cdot \sqrt{(e^{\frac{\square}{x}} - x)})$	$(x + (x + (x - \sqrt{\sin(\square)})))$
$(((\sin(\square))^2 - \sin(x)) - \cos(x))$	$(x - \frac{(x + \sqrt{\square})}{\sin(x)})$	$(x \cdot (\sqrt{x} - \sin((x + x))))$
$(\frac{x}{(\cos(\square))^2} + \sqrt{e^x})$	$(((x)^2 \cdot \sin(\square)) + e^{(x)^2})$	$(x + (x \cdot (\sqrt{\square} + e^x)))$
$\frac{\sqrt{\square}}{(\sin((x - \square)) - x)}$	$(\frac{\frac{x}{\sqrt{x}}}{x} - e^{\square})$	$((\square - (x + \sqrt{(x)^2})) \cdot \cos(x))$
$(\frac{\square}{(\square - \sin(x))} + (e^x)^2)$	$\frac{\sqrt{(\square - (\sin(x) + \cos(\square)))}}{x}$	$(\square \cdot (\square - (\sqrt{(\cos(x) - x)})^2))$
$((\sqrt{x} - \square) \cdot (\frac{\sqrt{x}}{\square})^2)$	$\frac{(\square - ((x \cdot \sin(x)))^2)}{e^x}$	$\frac{\sqrt{((\frac{x}{\square})^2 + e^x)}}{\square}$
$\frac{\square}{(x \cdot \sqrt{(x - (\square \cdot x))})}$	$(\square \cdot (x - (\sqrt{x} + \sqrt{e^x})))$	$(((\square \cdot x))^2 + (e^{\frac{\square}{x}})^2)$
$(\sqrt{\frac{((x+x))^2}{\cos(x)}} - \square)$	$\frac{(e^{(\square)^2} - \sin(x))}{(\sqrt{x})^2}$	$(((x + x) \cdot (\sin(\square) - \square)) - x)$
$\frac{(x - \sqrt{(x + e^{\square})})}{\sin(x)}$	$\frac{(\cos(x) + \cos((x+x)))}{e^{\square}}$	

Table 11: Held-out expressions for $n_{var} = 1$. Part 2.

$(\cos(\square) - \sqrt{x})$	$(\frac{\cos(x)}{x})^2$	$(\square - \sqrt{e^x})$
$((\square)^2 \cdot e^x)$	$\frac{\sin(x)}{\cos(x)}$	$(\cos(x) + e^{\square})$
$((x)^2 \cdot \sin(\square))$	$(\frac{x}{\square} - x)$	$(\square + \frac{\square}{x})$
$(x \cdot \sqrt{\cos(x)})$	$\sqrt{(\square \cdot \cos(x))}$	$(\square \cdot \sqrt{(x)^2})$
$\sqrt{(x \cdot e^{\square})}$	$\sqrt{(x - \cos(\square))}$	$((\square - \cos(x)))^2$
$(\square + \frac{\cos(x)}{\square})$	$((\sin(\square) - \cos(x)))^2$	$((\cos(x))^2 + \sin(x))$
$\sqrt{(\square \cdot (x + x))}$	$\frac{(e^x)^2}{e^{\square}}$	$\frac{\sin(\square)}{(\square - x)}$
$((\square \cdot x) - \sin(x))$	$(x \cdot (\cos(x) - x))$	$\frac{(\square - \sin(x))}{x}$
$(x + (x - \sqrt{x}))$	$((x - \square) \cdot e^{\square})$	$\frac{(\square + x)}{\cos(x)}$
$\frac{(\square - \sqrt{x})}{x}$	$\frac{\square}{(x + e^x)}$	$(\frac{\square}{x} - (\square)^2)$
$(\square + \sqrt{(x - \cos(\square))})$	$(\frac{x}{\sqrt{\square}} - \cos(x))$	$(\sqrt{(\square - x)} \cdot \sin(x))$
$\frac{(\square)^2}{\sqrt{(\square - x)}}$	$\frac{(\cos(\square) - \sin(\square))}{x}$	$(\frac{(x - \cos(\square))}{x})^2$
$\frac{(e^x - \sin(x))}{x}$	$\frac{\sqrt{\square}}{(x + (x)^2)}$	$(\square - \sqrt{\frac{\cos(\square)}{x}})$
$\frac{\cos(\square)}{\sqrt{(x - \square)}}$	$(x + (x - \frac{x}{\square}))$	$(\square + \sqrt{\frac{(x)^2}{\square}})$
$\frac{\sqrt{e^{(x - \square)}}}{x}$	$(\sin((\square + \square)) + e^x)$	$\frac{(\sin(x) - (\square)^2)}{x}$
$((\square - x) \cdot \sqrt{(x + x)})$	$(\sqrt{e^{\square}} - \frac{e^x}{\square})$	$(x - ((\cos(x))^2 + \sin(x)))$
$(x \cdot (x + \sin((x + x))))$	$\frac{\frac{x}{\sqrt{e^{\square}}}}{e^x}$	$((\sin(\square) \cdot e^x) - e^{\square})$
$\frac{(\square - \sin(\frac{x}{\square}))}{x}$	$\frac{\sqrt{(e^x)^2}}{(\sin(\square))^2}$	$((e^{\square})^2 - \sin((\square + x)))$
$(x \cdot (\square + \sqrt{(e^x)^2}))$	$\frac{\square}{(\square \cdot ((x)^2 - x))}$	$\frac{e^{(\frac{\square}{x})^2}}{\sqrt{x}}$
$\frac{\square}{(\square + (x \cdot \sin(x)))}$	$((\sin(x) \cdot e^x) - \sin(\square))$	$\frac{e^{(\square + x)}}{(\sin(\square))^2}$
$(\square - \sqrt{((\cos(\square))^2 + \cos(x))})$	$\frac{(\frac{x}{\sin(x)} - (x)^2)}{\square}$	$((\sqrt{(x + \sin(x))})^2 + \sqrt{\square})$
$\frac{(\sqrt{\square} - \sqrt{\sin(\square)})}{e^x}$	$\sqrt{((\cos(x))^2 - \frac{x}{e^{\square}})}$	$\frac{x}{(\sqrt{\square} \cdot \cos((\square + x)))}$
$\sqrt{\frac{\square}{((e^{\square})^2 - \cos(x))}}$	$\frac{((\square \cdot x))^2}{\cos(\frac{\square}{x})}$	$\frac{e^{\square}}{(\square + \cos((x + x)))}$
$((e^x - \sin(\square)) - e^{(\square)^2})$	$\sqrt{(((e^{\square})^2 - x) \cdot (x)^2)}$	$\frac{(\cos(x) + e^{(\square)^2})}{(x)^2}$
$(x \cdot \frac{((x)^2 + \sqrt{x})}{\square})$	$(x + \sqrt{\frac{x}{(\sin(\square) - \square)}})$	$(\square - (x + (x \cdot \sqrt{\sin(x)})))$
$\frac{(x - (\sqrt{x} \cdot e^x))}{\cos(\square)}$	$\frac{x}{((\sqrt{\square} - \cos(x)) \cdot \cos(x))}$	$(\sqrt{((\cos(x))^2 - x)} - (\sqrt{x})^2)$
$(\frac{x}{\sin(\square)} - \sqrt{\frac{e^x}{\square}})$	$((\square \cdot \frac{e^{\square}}{\sin(x)}) + e^x)$	$\frac{(\square + \sqrt{(x - \sin(x))})}{\cos(x)}$
$((\square + \frac{\square}{\sqrt{\cos((x + x))}}))^2$	$(x - ((\square + (\sqrt{x})^2) \cdot \cos(\square)))$	$(\frac{\frac{x}{\sqrt{x}}}{x} + \sqrt{(x)^2})$
$(\sqrt{(\square - e^x)} - \frac{\sin(x)}{x})$	$(\sqrt{e^{\frac{(\square - x)}{x}}} - (x)^2)$	$((x + (x)^2) \cdot \frac{\sin(x)}{\cos(x)})$
$(((\square)^2 \cdot e^x) + \sin((\square \cdot x)))$	$(\sqrt{((\sin(x) - \cos(\square)))^2} - (\square)^2)$	$(\frac{(\frac{\sqrt{x}}{\square})^2}{x} - e^x)$

Table 12: Held-out expressions for $n_{var} = 2$. Part 1.

$(\cos(y) - e^{x})$	$\sqrt{\cos(\frac{y}{x})}$	$\frac{\sqrt{e^{y}}}{x}$
$((y)^{2} - \sin(x))$	$(\square + \frac{x}{y})$	$(e^{x} - \sqrt{y})$
$\sqrt{e^{\frac{x}{y}}}$	$(\sqrt{\cos(x)} - y)$	$(y - (\square \cdot x))$
$((\square \cdot x) - y)$	$(y - \sqrt{\cos(x)})$	$(y + \sqrt{\cos(x)})$
$(\sqrt{\sin(x)} - y)$	$\frac{y}{(\sin(x))^{2}}$	$\frac{(x-y)}{y}$
$(x \cdot \sqrt{(y + y)})$	$(y - (x \cdot \sin(y)))$	$\frac{(x-y)}{\sqrt{\square}}$
$\frac{x}{((\square-y))^{2}}$	$(\sqrt{x} + \sqrt{\sin(y)})$	$(\sqrt{y} + \sqrt{e^{x}})$
$(y \cdot \frac{\cos(x)}{x})$	$(x - \frac{y}{\cos(\square)})$	$\frac{\sqrt{x}}{(\sin(y))^{2}}$
$(y + \sqrt{(x + x)})$	$\cos(\frac{(\square+y)}{x})$	$((x - y) \cdot \cos(\square))$
$(\square - (y + \cos(x)))$	$(\frac{y}{x} - e^{y})$	$\frac{x}{(\sin(x)-y)}$
$(\sin(x) - (y \cdot \sqrt{x}))$	$\sqrt{(x + \cos(\frac{y}{\square}))}$	$(\square - \frac{\sin(y)}{\sqrt{x}})$
$(y + (x \cdot \sqrt{\sin(\square)}))$	$(x - \frac{\sin(x)}{\sqrt{y}})$	$(\sqrt{(y)^{2}} \cdot \sqrt{e^{x}})$
$(\frac{\sin(x)}{\sin(y)} - x)$	$((\square + (y \cdot \sin(x))))^{2}$	$\frac{x}{(\square+\frac{y}{x})}$
$\frac{((y)^{2}-x)}{e^{x}}$	$(y - \frac{\sin(x)}{\sin(\square)})$	$\frac{e^{\frac{x}{(y)^{2}}}}{x}$
$((y \cdot (\square + x)) - y)$	$\frac{y}{(\sin(x)+\sin(x))}$	$((x)^{2} + \sin((y - \square)))$
$(\sqrt{e^{(x)^{2}}} - (\sqrt{y})^{2})$	$((y + (\sqrt{\square} - e^{x})))^{2}$	$(x \cdot (x - (\frac{y}{\square})^{2}))$
$((\frac{\sqrt{x}}{x} - \sqrt{y}))^{2}$	$\frac{\sqrt{y}}{((x)^{2}+\cos(y))}$	$(e^{y} - \frac{(\square+x)}{\square})$
$\frac{(\sin(\square)-\square)}{(x+y)}$	$(((\square \cdot \sqrt{y}) + \sqrt{x}))^{2}$	$(\sqrt{(y - x)} - \frac{\square}{x})$
$(\cos((y \cdot (\square + x))) - \square)$	$\frac{\sqrt{((\square)^{2}-\sin(y))}}{x}$	$(\square + (x + \sin((\square + y))))$
$((y + \sqrt{(x)^{2}}) \cdot e^{\square})$	$\frac{x}{((\square \cdot y)-\sin(\square))}$	$\frac{(x-\frac{\cos(x)}{y})}{x}$
$\frac{\sqrt{(\sin(y)+\cos(\square))}}{(x)^{2}}$	$(x \cdot (\square - (\sin((\square + y)))^{2}))$	$\frac{\sqrt{(y-\cos(x))}}{\sqrt{\sin(y)}}$
$\frac{\frac{\sin((\square \cdot x))}{\sqrt{y}}}{y}$	$(\sqrt{(\square \cdot y)} - \sqrt{\frac{x}{\square}})$	$(\frac{(\square+\sqrt{\square})}{y} + e^{x})$
$\sqrt{\frac{(\sin(\square)+\cos(y))}{\sin(x)}}$	$\frac{(\frac{y}{\sin(x)}+\cos(x))}{\square}$	$\frac{((x)^{2}+\sin(y))}{(\sqrt{x})^{2}}$
$(\square \cdot (\frac{(x+\sin(\square))}{y})^{2})$	$((\square \cdot \sqrt{\frac{\square}{x}}) + \sin(y))$	$(\frac{x}{\sqrt{\square}} - \frac{\sin(x)}{y})$
$((y \cdot (y - \square)) - e^{(x)^{2}})$	$\sqrt{((x + y) \cdot \frac{\sin(x)}{x})}$	$\frac{\square}{\sqrt{\frac{\cos(y)}{(x+y)}}}$
$\frac{(x-\frac{e^{\frac{\square}{y}}}{x})}{x}$	$(x - (y \cdot \cos((\square + (y - x)))))$	$(((y - e^{y}) \cdot \sin(x)) + e^{\square})$
$\frac{\frac{e^{\frac{x}{x}}}{(\sqrt{y}-\square)}}{(y)^{2}}$	$\frac{y}{((x \cdot (\sin(\square))^{2})-\cos(x))}$	$(\sqrt{\frac{(\cos(y))^{2}}{x}} + \sqrt{(y)^{2}})$
$\frac{\frac{\sqrt{y}}{\square}}{((y)^{2}+\cos(x))}$	$(x + \frac{\cos(y)}{(\sqrt{x}+\sin(x))})$	$((\square + \square) \cdot (\frac{y}{\sin(y)} - x))$
$\frac{y}{((\sin(\square)-(x)^{2})-\sin(y))}$	$\frac{x}{((\square)^{2}-\sqrt{(y+\cos(y))})}$	$(x + (x + \sqrt{\frac{y}{(\sin(y))^{2}}}))$
$\sqrt{(\frac{(x+y)}{\sin(\square)} + (\square)^{2})}$	$\frac{\sqrt{x}}{(x-(\sqrt{x}+\sin(y)))}$	$(\square + (y \cdot (\sqrt{\sin(\frac{\square}{x})})^{2}))$

Table 13: Held-out expressions for $n_{var} = 2$. Part 2.

$(y - \frac{x}{y})$	$\sqrt{\frac{\cos(y)}{x}}$	$((x)^2 \cdot \sin(y))$
$(\sqrt{x} \cdot \sqrt{y})$	$\sqrt{\frac{\sin(x)}{y}}$	$\sqrt{\frac{e^x}{y}}$
$\frac{y}{\sqrt{\sin(x)}}$	$e^{\frac{y}{(x)^2}}$	$((e^x)^2 - y)$
$\frac{e^x}{\sin(y)}$	$(\sqrt{x} - (y)^2)$	$(y \cdot (x - \square))$
$\frac{\cos(y)}{\cos(x)}$	$(x + \sqrt{(y)^2})$	$\frac{x}{(\square+y)}$
$\frac{y}{\sqrt{(\square+x)}}$	$\frac{(\square-x)}{\sin(y)}$	$(x \cdot \sin(\frac{x}{y}))$
$\frac{(y-\cos(x))}{\square}$	$((x \cdot e^{\square}) - y)$	$\frac{\cos((\square+x))}{y}$
$\frac{x}{(\sin(y)-y)}$	$\sqrt{((y \cdot \sin(x)))^2}$	$\frac{x}{(y+\sin(\square))}$
$\cos((x - \frac{y}{\square}))$	$\frac{(\cos(y)-y)}{x}$	$(\sin(y) - \sqrt{\sin(x)})$
$(x \cdot \cos(\frac{\square}{y}))$	$(((x + y))^2 - x)$	$(x + \cos(\frac{\square}{y}))$
$\sqrt{(y - \cos((\square + x)))}$	$((y + (\sin(x) - \square)))^2$	$(y \cdot \frac{(x)^2}{\sin(x)})$
$(x - ((\square \cdot \sqrt{y}))^2)$	$\frac{\sqrt{(\square+y)}}{(x)^2}$	$(\sqrt{\square} - \frac{(y)^2}{x})$
$\sqrt{\frac{x}{(y+\cos(y))}}$	$\frac{(x+e^x)}{(y)^2}$	$(\sqrt{x} - \frac{y}{\sqrt{\square}})$
$\sqrt{\cos((x \cdot (y + y)))}$	$(y + \sqrt{\frac{\cos(x)}{\square}})$	$((\square + e^{(x-y)}))^2$
$\frac{((x)^2-(y)^2)}{\square}$	$(\frac{x}{e^{\square}} + (y)^2)$	$((\square)^2 - \frac{y}{(x)^2})$
$\frac{y}{\sqrt{(\sin(y) \cdot e^x)}}$	$\frac{\frac{(\frac{y}{\square})^2}{\square}}{x}$	$\frac{\frac{\cos(x)}{y}}{\sqrt{e^y}}$
$((x \cdot y) + \cos(\frac{\square}{\square}))$	$(\square \cdot ((x)^2 \cdot \sqrt{(y)^2}))$	$\frac{(\square+e^x)}{(x \cdot y)}$
$(\frac{\sqrt{e^y}}{x} - (y)^2)$	$(\frac{e^{(y-x)}}{\sin(\square)})^2$	$(x + \frac{\frac{y}{\sqrt{x}}}{y})$
$(\frac{(x)^2}{\sin(x)} - \sin(y))$	$(y + (\frac{y}{x} + (\square)^2))$	$\frac{\sin(x)}{\sqrt{(x-\sin(y))}}$
$\frac{((\sqrt{y}-\cos(x)))^2}{x}$	$((\cos(x))^2 - (x \cdot \sqrt{y}))$	$(y - (y \cdot \frac{\sin(x)}{x}))$
$(\sqrt{\frac{x}{(e^x)^2}} - \sqrt{y})$	$((x - \sqrt{\frac{x}{(\square+y)}}))^2$	$\frac{(\cos(y)-\square)}{\sin((\square+x))}$
$((x \cdot y) - (\sqrt{x} \cdot \sin(y)))$	$((\sqrt{(y)^2} - (\square)^2) - \cos(x))$	$(((\square + \square) \cdot e^{\frac{x}{y}}))^2$
$(\sqrt{\frac{\square}{e^y}} + \sqrt{e^x})$	$\frac{(x-\sqrt{y})}{\cos((\square+x))}$	$((y \cdot (x - y)) - (\sin(\square))^2)$
$(\frac{e^x}{\square} - (y \cdot \sqrt{y}))$	$(\square \cdot (e^{(\square-y)} - e^x))$	$((\sqrt{x} + \sqrt{\sin(y)}) + \sin(\square))$
$(((\sin(x) - y) - y) - e^{\square})$	$((y + (\frac{\cos(y)}{x} - x)))^2$	$\sqrt{\frac{(\cos(y)-(y)^2)}{\sin(x)}}$
$\frac{x}{(y+(\sin((\square \cdot x))-\square))}$	$\frac{\sqrt{x}}{(y-\sqrt{\frac{x}{e^{\square}}})}$	$(\frac{\square}{(x)^2} + \sqrt{(y - \cos(x))})$
$(x + (x + ((x \cdot y) - e^{\square})))$	$(\sin((y - x)) - \frac{\cos(x)}{\sqrt{y}})$	$((\sqrt{\frac{e^x}{y}} - x) - \cos(y))$
$((\square \cdot \frac{(x)^2}{\sqrt{y}}) \cdot \cos(\square))$	$\frac{y}{((y-(\sin(\square))^2) \cdot (x)^2)}$	$(y + \frac{\square}{\sin(\frac{x}{(\square \cdot y)})})$
$(x + ((\frac{\square}{\sin(y)})^2 \cdot \sin(\square)))$	$((x - (e^y + e^y)) \cdot \sin(x))$	$\frac{((x \cdot \sqrt{\sin(y)})-e^x)}{y}$
$((\sin((x \cdot y)) \cdot e^{\square}) - \cos(x))$	$(((\square - \frac{x}{\sin(y)}))^2 + \cos(y))$	$((\frac{\cos(y)}{\square})^2 + \sqrt{(\square \cdot x)})$

Table 14: Held-out expressions for $n_{var} = 3$. Part 1.

$(y + (z - x))$	$((x \cdot z) - y)$	$\frac{y}{(x+z)}$
$(x + \frac{y}{z})$	$\frac{(y-z)}{x}$	$(\frac{z}{y} - x)$
$(x + (y \cdot z))$	$(y \cdot (x - z))$	$(y \cdot \frac{z}{x})$
$(z - (x \cdot y))$	$\frac{(y+z)}{x}$	$\frac{(x-z)}{y}$
$\frac{x}{(y+z)}$	$\frac{z}{(x-y)}$	$\frac{z}{(x \cdot y)}$
$e^{\frac{(y+z)}{x}}$	$\frac{x}{(\sqrt{y}-z)}$	$(z \cdot \sin(\frac{y}{x}))$
$(z - \sin((x + y)))$	$((\frac{x}{y})^2 - z)$	$((y \cdot \sin(x)) - z)$
$(y \cdot e^{(x-z)})$	$\frac{\sin((x \cdot z))}{y}$	$\frac{z}{(y \cdot \cos(x))}$
$\frac{y}{e^{\frac{x}{z}}}$	$((y \cdot e^x) - z)$	$(\frac{(x+y)}{z})^2$
$\sin((x + \frac{z}{y}))$	$(x - (y + \sin(z)))$	$((y + z) \cdot \cos(x))$
$\frac{(\sin(z)-y)}{\sqrt{x}}$	$\frac{z}{(x+(x-y))}$	$(\sin(\frac{y}{x}) - \sin(z))$
$(\sqrt{x} \cdot \sqrt{(z - y)})$	$(x \cdot \frac{(x+y)}{z})$	$(\sqrt{(z + e^x)} - y)$
$\frac{(y-\sqrt{\cos(x)})}{z}$	$((z - y) - e^{(x)^2})$	$\frac{(y+e^x)}{\sqrt{z}}$
$(x + (z + \sqrt{\cos(y)}))$	$\frac{(\sin(y))^2}{(z-x)}$	$\frac{(x+z)}{(x \cdot y)}$
$((y \cdot \frac{\sqrt{x}}{z}))^2$	$(\frac{\sqrt{y}}{z} - \cos(x))$	$\frac{x}{(y \cdot (x-z))}$
$\frac{z}{\cos((y \cdot (x-y)))}$	$(x - (z + \cos((y + y))))$	$\sqrt{(\frac{x}{(y)^2} - \cos(z))}$
$\frac{e^{(z \cdot (x-z))}}{y}$	$((z)^2 + ((x - \cos(y)))^2)$	$(e^x - \frac{x}{(y \cdot z)})$
$\frac{z}{(\sin(\frac{x}{y})-x)}$	$\sqrt{\frac{(z-(x+y))}{\square}}$	$(z + ((\square + x) \cdot (y)^2))$
$(\sqrt{y} - (\cos((x + z)))^2)$	$\frac{x}{(y \cdot \sin((y+z)))}$	$(\sqrt{x} + \sqrt{((y - z))^2})$
$\frac{z}{(\cos((x+z))-y)}$	$\frac{x}{((z \cdot \sin(y))-z)}$	$\frac{(\square+x)}{\cos((y-z))}$
$(\frac{z}{(\cos(\frac{x}{y}))^2} - z)$	$(x \cdot \frac{e^{(z-y)}}{(y)^2})$	$(\sqrt{(x - z)} - (x \cdot (y)^2))$
$\frac{((\sqrt{y} \cdot \sin(z))-x)}{z}$	$((z \cdot (y + \sin(\frac{x}{\square}))))^2$	$((z \cdot \cos((\square - x))) + \sin(y))$
$((x \cdot \frac{z}{\sqrt{y}}) + e^z)$	$(\frac{y}{\sin(z)} + \sqrt{(x - y)})$	$(\square \cdot (\sqrt{(\cos(y) - x)} - z))$
$(x + (e^{\frac{x}{z}} - \cos(y)))$	$(y - (\sqrt{(x - z)} \cdot e^{\square}))$	$\frac{\square}{(y \cdot e^{((x-z))^2})}$
$\frac{(y-z)}{(\sqrt{x}-(z)^2)}$	$(x \cdot (\cos(z) - \frac{y}{\cos(\square)}))$	$(x - \frac{\square}{(z+(\sin(y))^2)})$
$((x + (y \cdot \sqrt{\sin(\square)})) - e^z)$	$(e^{(x-\square)} - \sqrt{(y + \cos(z))})$	$(((e^x - \square) + e^y) \cdot \sqrt{z})$
$\frac{\sin(z)}{(\cos(y)-\frac{z}{\sin(x)})}$	$(((x)^2 - \sin(z)) + \sin((\square + y)))$	$\frac{z}{((e^{(z-y)})^2+\sin(x))}$
$\frac{\sqrt{\sin(x)}}{(x+\sqrt{(z-y)})}$	$((x + \frac{(\square)^2}{z}) \cdot (x - y))$	$\frac{\frac{e^y}{z}}{\sqrt{(y+(x)^2)}}$
$(x + (\square \cdot \frac{\sin(\frac{z}{y})}{y}))$	$(y - \frac{\sin((x \cdot z))}{e^{(\square)^2}})$	$\frac{(\sqrt{y}+e^{(\square \cdot z)})}{\sqrt{x}}$
$\sqrt{\frac{((y-\frac{z}{\sin(x)}))^2}{z}}$	$((((y)^2 - z) + \cos(x)) \cdot \sqrt{x})$	$(y \cdot \frac{z}{\cos((\square+(x+z)))})$

Table 15: Held-out expressions for $n_{var} = 3$. Part 2.

$(\frac{x}{y} - z)$	$(y + \frac{z}{x})$	$(x \cdot (z - y))$
$(x \cdot (y - z))$	$(y - (x \cdot z))$	$(x \cdot \frac{y}{z})$
$(x - (y + z))$	$(x \cdot (y \cdot z))$	$(z - \frac{x}{y})$
$(y - \frac{z}{x})$	$\frac{x}{(z-y)}$	$(x - (y \cdot z))$
$\frac{x}{(y-z)}$	$(y - \frac{x}{z})$	$(z - (x + y))$
$(x + (z - \sin(y)))$	$(\frac{(y)^2}{z} - x)$	$((x - \frac{z}{y}))^2$
$(\frac{z}{x} - \sqrt{y})$	$((x \cdot z) + e^y)$	$\frac{(z-x)}{(y)^2}$
$e^{(x - \frac{z}{y})}$	$\frac{\sin((y-x))}{z}$	$((x \cdot \sqrt{z}) - y)$
$\frac{x}{(z+\sqrt{y})}$	$((z)^2 - \frac{x}{y})$	$((z - y) \cdot (x)^2)$
$((y \cdot z) + \sin(x))$	$(y \cdot \frac{\sin(x)}{z})$	$\frac{z}{(y-\cos(x))}$
$(x + \frac{z}{(\square-y)})$	$\frac{\sqrt{(x+e^z)}}{y}$	$(x - (z + \frac{y}{x}))$
$(x - (\sin(y) + \cos(z)))$	$(e^z \cdot e^{\frac{x}{y}})$	$\frac{(z+z)}{(x \cdot y)}$
$(x - (\sqrt{(y - z)})^2)$	$\frac{(z+\frac{y}{x})}{\square}$	$\frac{(x+\sin(z))}{\sin(y)}$
$(\frac{(y-x)}{x} - z)$	$((y \cdot \sqrt{(z - x)}))^2$	$\frac{y}{e^{\frac{z}{(x)^2}}}$
$(e^{(y-x)} - e^z)$	$\frac{\sin(z)}{\sqrt{(y-x)}}$	$(((x)^2 - y) \cdot (z)^2)$
$(\frac{(\sin(y))^2}{\sqrt{z}} - x)$	$(z \cdot (\frac{x}{(x-y)})^2)$	$((y + \frac{\cos(z)}{\cos(x)}))^2$
$((z)^2 + \sqrt{\frac{x}{\cos(y)}})$	$\frac{(\cos((\square \cdot y)) - x)}{z}$	$(\square \cdot \frac{(x-y)}{e^z})$
$(\sqrt{\cos(y)} - \sqrt{(x \cdot z)})$	$\sqrt{((x + \sin(y)) \cdot \sin(z))}$	$(x + (\sqrt{(\square + y)} - z))$
$\sqrt{(\frac{(x)^2}{y} + \sin(z))}$	$((y \cdot z) - \frac{\cos(y)}{x})$	$(\square - \frac{z}{(x+(y)^2)})$
$((z \cdot (z + \sin(x))) - y)$	$(((\square + z) \cdot e^x) - y)$	$\frac{(\square + \frac{\cos(z)}{x})}{y}$
$\frac{(x)^2}{(\frac{y}{z} + \sin(y))}$	$(x - \frac{\frac{\sqrt{e^x}}{y}}{z})$	$\sqrt{\cos(\frac{x}{(y \cdot (x+z))})}$
$(((\cos(x))^2 - \sin(y)) \cdot (z)^2)$	$\sqrt{\frac{((x)^2 - \frac{x}{y})}{z}}$	$\frac{y}{(x \cdot \sqrt{\frac{\cos(\square)}{z}})}$
$((x \cdot ((y)^2 - y)) + (z)^2)$	$(\frac{\sqrt{(x \cdot \sin(y))}}{x} - z)$	$\frac{(\square + x)}{\sqrt{\frac{e^z}{y}}}$
$(y \cdot ((x + z) \cdot (\sin(y))^2))$	$\sqrt{(z - (x \cdot ((\square + y))^2))}$	$(z - (y \cdot (z + (\cos(x))^2)))$
$(((\frac{y}{z})^2 \cdot \cos(y)) - x)$	$(\frac{(y-\cos(z))}{(x-z)})^2$	$((x - (y \cdot \cos(\frac{z}{x}))))^2$
$(z \cdot (\sqrt{x} - (y \cdot (y + z))))$	$(y - \frac{\frac{(z)^2}{\sqrt{x}}}{(z)^2})$	$(e^{\frac{x}{z}} - (y \cdot \sqrt{\sin(\square)}))$
$(x + \frac{z}{((\sqrt{y}+\sin(z)))^2})$	$\frac{(y-(\square+(z \cdot \sin(y))))}{x}$	$(\frac{\sqrt{\sin(x)}}{y} + (\frac{z}{y})^2)$
$(\square \cdot (x + (x \cdot \frac{\cos(y)}{z})))$	$\frac{x}{((\sin(x))^2 + \sqrt{\frac{z}{y}})}$	$(y + \frac{\sin(z)}{(x - \frac{y}{x})})$
$(\sqrt{(x + \frac{(x-z)}{x})} - y)$	$(y - (z \cdot (\square + (x \cdot \sqrt{y}))))$	$(\sqrt{z} - (x \cdot \sqrt{\frac{x}{e^y}}))$
$\frac{x}{((z)^2 \cdot \sqrt{((\square+y))^2})}$	$(z + (\frac{(y-x)}{\square} + \sin(y)))$	$(\sqrt{\cos(x)} + \sin((y \cdot (x - z))))$

Table 16: Held-out expressions for $n_{var} = 4$. Part 1.

$\frac{z}{(w-\frac{y}{x})}$	$((w \cdot (x+y)) - z)$	$\frac{x}{((y \cdot w)-z)}$
$(\frac{x}{w} - \frac{z}{y})$	$((z \cdot \frac{w}{y}) - x)$	$\frac{x}{(w-\frac{y}{z})}$
$\frac{(x-\frac{w}{z})}{y}$	$\frac{z}{((y \cdot w)-x)}$	$(x \cdot \frac{y}{(z-w)})$
$(\frac{w}{(y+z)} - x)$	$(z \cdot (x - \frac{y}{w}))$	$(\frac{w}{(x \cdot z)} - y)$
$\frac{w}{(x+(y \cdot z))}$	$\frac{(y+z)}{(x-w)}$	$(y \cdot (z - (x+w)))$
$(\frac{y}{z} - (w \cdot e^{x}))$	$((w \cdot \cos(z)) + \frac{x}{y})$	$(e^{(w \cdot (x-z))} - y)$
$(\frac{\frac{\cos(z)}{x}}{y} - w)$	$(z + \frac{(w+\sin(y))}{x})$	$\sqrt{(x \cdot (z + \frac{y}{w}))}$
$e^{(x \cdot \frac{(y-w)}{z})}$	$\frac{y}{(x-(w+\sin(z)))}$	$((w \cdot ((y)^2 - z)) - x)$
$(x + \frac{(z+\sin(w))}{y})$	$(y \cdot ((z-w) \cdot \sqrt{x}))$	$(y \cdot \frac{(z)^2}{(w-x)})$
$(w \cdot (x + \cos((y-z))))$	$((z \cdot ((x-w))^2) - y)$	$(z + e^{\frac{x}{(w-y)}})$
$(w - ((\frac{z}{x} - \cos(y)))^2)$	$\frac{(\sin(\frac{z}{w})-x)}{e^{y}}$	$((x \cdot \sqrt{e^{z}}) + (y \cdot w))$
$\frac{\sqrt{(x \cdot (y+(w)^2))}}{z}$	$(\frac{y}{(w)^2} + \frac{\sqrt{x}}{z})$	$(x \cdot \sqrt{(y \cdot \frac{\sin(w)}{z})})$
$(\frac{(y-x)}{\sqrt{z}} - \sqrt{w})$	$((\cos(z) - y) - ((x-w))^2)$	$\frac{(\frac{w}{\cos(y)}-x)}{\cos(z)}$
$(y \cdot (z + e^{((x-w))^2}))$	$\frac{(w+\sin(y))}{(x-\cos(z))}$	$(((y-w) \cdot (\cos(x))^2) - z)$
$(w - ((x - \sqrt{(y+z)}))^2)$	$\sqrt{\frac{((z+\frac{w}{y}))^2}{x}}$	$((\sqrt{w} - x) \cdot ((y+z))^2)$
$\frac{(\frac{(y-\square)}{w}+\cos(x))}{z}$	$((y \cdot (x+w)) + e^{(z-w)})$	$(x - e^{\frac{(y-z)}{(w-z)}})$
$\frac{\frac{y}{e^{z}}}{e^{(\frac{x}{w})^2}}$	$(\frac{(\frac{z}{\cos(y)})^2}{x} - (w)^2)$	$(z + e^{\frac{w}{(\frac{x}{y}-y)}})$
$(\square \cdot \frac{(z+\cos(x))}{(y-w)})$	$\sqrt{\frac{((y)^2+((z \cdot w))^2)}{x}}$	$(y + (\sqrt{(x \cdot w)} - (\square \cdot z)))$
$(y + ((z \cdot w) - \frac{\sin(x)}{z}))$	$(\frac{\sqrt{w}}{z} - \frac{y}{\sqrt{(x)^2}})$	$((z - (x \cdot (\sqrt{w})^2)) \cdot \cos(y))$
$\frac{(\sin(\frac{y}{(x-z)})-w)}{z}$	$((\sqrt{z} - (y)^2) \cdot ((x \cdot w))^2)$	$((x+z) \cdot \frac{e^{(y-w)}}{z})$

Table 17: Held-out expressions for $n_{var} = 4$. Part 2.

$(w \cdot (y + \frac{z}{x}))$	$((y \cdot (z - x)) - w)$	$((y \cdot z) - \frac{x}{w})$
$(y - \frac{(x+w)}{z})$	$(\frac{y}{(x \cdot w)} - z)$	$(x - (y \cdot \frac{z}{w}))$
$\frac{(x-y)}{(z \cdot w)}$	$\frac{(\frac{w}{y} - z)}{x}$	$(y \cdot (\frac{x}{z} - w))$
$(\frac{w}{(x+z)} - y)$	$(x \cdot (\frac{y}{w} - z))$	$(w - (x + (y \cdot z)))$
$\frac{(z+(x \cdot y))}{w}$	$\frac{w}{(\frac{z}{x} - y)}$	$(w \cdot (x - \frac{y}{z}))$
$\sin(\frac{(x-z)}{(y+w)})$	$\frac{z}{(y \cdot (w \cdot (x)^2))}$	$(w \cdot (y + (\sin(x) - z)))$
$\frac{(w + \frac{z}{\sin(x)})}{y}$	$((w \cdot (e^y - x)) - z)$	$(e^{\frac{x}{(y \cdot z)}} - w)$
$(y \cdot (z + e^{(x-w)}))$	$\sqrt{(x + (y \cdot (w - z)))}$	$\frac{(x+(\cos(y)-w))}{z}$
$\sqrt{\frac{y}{(w-(x \cdot z))}}$	$\frac{(z+\sqrt{w})}{(x \cdot y)}$	$\frac{(x+((y \cdot w))^2)}{z}$
$\frac{(y-w)}{(x-(z)^2)}$	$(((y \cdot z) - x) - \cos(w))$	$(y + \frac{x}{(\sqrt{w}-z)})$
$(x + (w + \frac{(\cos(y))^2}{z}))$	$(\frac{x}{(y-(z \cdot w))} - y)$	$\frac{((\sqrt{y}-z)-(w)^2)}{x}$
$(\frac{(w-\sin(x))}{(y)^2} - z)$	$\frac{\sqrt{y}}{(w+\sin(\frac{x}{z}))}$	$((w \cdot (y - \frac{x}{z})) - y)$
$(\frac{(\frac{x}{w})^2}{z} \cdot \sin(y))$	$(w \cdot (\frac{x}{y} - \frac{\square}{z}))$	$\frac{x}{(\frac{y}{\sqrt{w}}+\sin(z))}$
$(x \cdot (((y)^2 \cdot \cos(z)) - w))$	$(w + e^{\frac{y}{((x)^2-z)}})$	$((y \cdot \cos(z)) - ((x \cdot w))^2)$
$\frac{(w-\cos(x))}{(y+\cos(z))}$	$\frac{z}{((\sin(w)-x)+\cos(y))}$	$((x + (\cos(y) - \frac{w}{z})))^2$
$((((x - \cos(y)))^2 \cdot \sin(z)) - w)$	$((\square - z) \cdot \cos(\frac{x}{(y+w)}))$	$(z - \frac{\cos((\square+(x+y)))}{w})$
$((z)^2 \cdot \sqrt{(w + \frac{(x)^2}{y})})$	$((\frac{\sqrt{y}}{(w)^2} - z) + \sqrt{x})$	$(\frac{(\sin((y+w)))^2}{z} + \sin(x))$
$(\sqrt{(y + ((z \cdot w) - x))} - z)$	$\frac{w}{(\square - \frac{z}{\cos(\frac{x}{y})})}$	$\frac{x}{(\frac{e^z}{\sin(w)} - \cos(y))}$
$\frac{\sqrt{(\frac{e^w}{y} - \sin(z))}}{x}$	$(((y \cdot \frac{\sin(w)}{x}))^2 \cdot \sqrt{z})$	$(z - \frac{\sqrt{\frac{y}{(z-x)}}}{w})$
$((y \cdot (\frac{y}{x})^2) - \frac{z}{w})$	$((\square - (z \cdot \sqrt{x})) \cdot (y - w))$	$((\frac{w}{y} + \sqrt{(x + e^z)}))^2$