DISENTANGLING INVARIANT STRUCTURES FROM PARAMETER VARIATIONS FOR PARAMETERIZED EQUATION DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

While Symbolic Regression (SR) excels at discovering mathematical expressions from data, it traditionally treats numerical coefficients as fixed, recovering isolated mathematical instances. However, many scientific applications require **P**arameterized **E**quation **D**iscovery (PED): identifying an invariant symbolic skeleton alongside instance-specific parameters across multiple parameter-varying environments. A fundamental challenge in PED is parameter-induced structural ambiguity, where parameter variations can drastically alter the data distribution of a single skeleton or cause distinct skeletons to become observationally aliased. Existing multi-environment SR methods rely heavily on heuristic search and post-hoc aggregate fitness evaluation, lacking a data-driven prior to separate structural invariants from parameter variations. In this work, we propose ***DisPV (Disentangling invariant structures from Parameter Variations)***, the first pre-training framework specifically designed for PED. DisPV designs a dual-stream architecture that explicitly disentangles universal mathematical structures from environment-specific constants at the representation level. Through a cross-environment contrastive learning strategy, the skeleton stream isolates invariant structural features, while the parallel constant stream predicts instance-specific parameters. By learning structural invariance directly from parameter-varying data representations, DisPV provides a highly accurate structural prior. Extensive experiments on synthetic and physics-inspired benchmarks demonstrate that DisPV, coupled with genetic programming refinement, significantly outperforms state-of-the-art search-based baselines with an average $R^2 = 0.9$.

1 INTRODUCTION

Scientific laws rarely describe isolated observations; rather, they characterize universal mathematical relationships whose specific manifestations vary across physical environments. While Symbolic Regression (SR) has emerged as a powerful tool for discovering mathematical expressions from data (Biggio et al., 2021; Petersen et al., 2021), conventional SR typically treats coefficients as fixed constants, aiming to recover a single mathematical instance (e.g., $y = 1.2x^2 + 0.4$) from a single dataset. In scientific practice, a more fundamental objective is **Parameterized Equation Discovery (PED)** (Sen Fong & Motani, 2024). PED shifts the paradigm from single-instance curve fitting to identifying an invariant symbolic skeleton f alongside instance-specific parameters θ from multiple heterogeneous datasets, formulating the governing law as $y = f(\mathbf{x}; \theta)$.

However, discovering a universal skeleton across multiple environments introduces a critical challenge we term **parameter-induced structural ambiguity**, which manifests in two coupled ways (Fig. 1). Consider an underdamped mass-spring-damper system governed by $y = a\, e^{-bx} \cos(cx)$, where the task-specific parameters $\theta = (a, b, c)$ vary across experimental setups. First, *different parameter configurations of the same skeleton induce vastly different data distributions*. For instance, a low-friction setup (Env A) exhibits rapid oscillations, whereas near-critical damping (Env B) mimics a monotonic decay (Fig. 1(a)); a model trained in isolation might erroneously infer two entirely different governing mechanisms. Second, *under specific parameter regimes, distinct symbolic structures become observationally aliased*. In Env B, the damped frequency $c = 0.22$ is so small that $\cos(cx) \approx 1$, rendering the true trajectory numerically indistinguishable from a structurally in-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

complete, spurious pure-exponential skeleton $y = \alpha e^{-\beta x}$ (Fig. 1(b)). Without cross-environment constraints to reveal that this pure-exponential form fails catastrophically on Env A, a conventional SR model is highly susceptible to collapsing into this aliased local optimum.

To address these challenges, existing multi-dataset SR methods, such as adapted PySR (Cranmer, 2023) or Class Symbolic Regression via PhySO (Tenachi et al., 2024), rely heavily on heuristic search. These methods incorporate cross-environment consistency primarily through the optimization objective—evaluating the aggregate fitness of a generated candidate across all datasets. While effective, they do not learn a latent representation of the mathematical structures. Every search starts from scratch, lacking a data-driven prior to distinguish structural invariants from parameter-dependent variations.

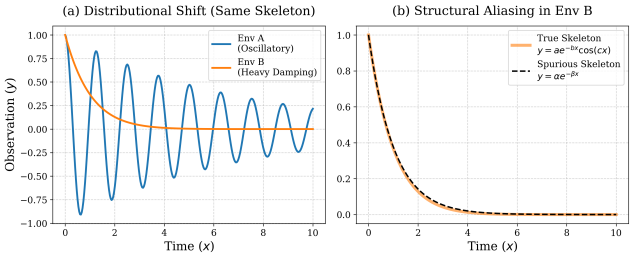


Figure 1: Parameter-induced structural ambiguity. The parameteried equation of exampled underdampled mass-spring-damper system is $y = a\,e^{-bx}\cos(cx)$, where a, b, c are mass, stiffness, and damping, respectively. (a) Distributional shift: The same underlying symbolic skeleton generates fundamentally different observations across setups—rapid oscillations in Env A ($b = 0.15, c = 5.0$) versus near-monotonic decay in Env B ($b = 0.975, c = 0.22$). (b) Structural Aliasing: In Env B, the true low-frequency oscillatory system $y = 1.0e^{-0.975x}\cos(0.22x)$ becomes numerically indistinguishable (MSE$< 1.0 \times 10^{-5}$) from a spurious, structurally incomplete pure-exponential equation ($y = 1.0e^{-0.975x}$), creating a deceptive local optimum if evaluated in isolation.

Recently, the standard paradigm in single-task SR has shifted toward hybrid frameworks (Landajuela et al., 2022; Petersen et al., 2021), where powerful neural pre-trained models provide strong structural priors that are subsequently refined by genetic programming (Mundhenk et al., 2021; Meidani et al., 2023), reinforcement learning (Holt et al., 2023; Liu et al., 2023), or Montle Carlo Tree Search (Kamienny et al., 2023; Shojaee et al., 2023). We argue that PED can profoundly benefit from this paradigm. Rather than repeatedly generating and evaluating expressions via search to enforce consistency, a pre-trained model can directly learn to disentangle invariant structures from varying parameters in the latent feature space.

To this end, we propose ***D**isentangling **i**nvariant **s**tructures from **P**arameter **V**ariations (**DisPV**)*, the first pre-training framework specifically designed for PED. DisPV is an end-to-end dual-stream architecture. It takes diverse observations from multiple environments and processes them through a shared SetTransformer (Lee et al., 2019) encoder. The **Skeleton Stream** isolates environment-invariant structural features via contrastive learning—pulling together diverse parameter manifestations of the same skeleton while pushing apart aliased structures—and autoregressively decodes the universal mathematical skeleton. Concurrently, the **Constant Stream** estimates the environment-specific parameters. By disentangling these features during pre-training, DisPV provides a highly accurate structural prior, which is then seamlessly integrated with a GP module for final refinement, following the modern SR paradigm.

To summarize, our main contributions are summarized as follows:

- We identify and formalize the core challenge of PED as **parameter-induced structural ambiguity**, characterized by high inter-parameter distributional variance and cross-structural aliasing.
- We introduce **DisPV**, the first pre-training framework for PED. By employing a novel dual-stream architecture, DisPV explicitly disentangles universal symbolic skeletons from instance-specific parameters at the representation level.
- Extensive experiments on synthetic and physics-inspired benchmarks demonstrate that DisPV paired with GP refinement vastly outperforms existing search-based PED baselines.

2 Related Work and Preliminaries

2.1 Symbolic Regression

Symbolic Regression (SR) seeks to discover a closed-form mathematical expression $f(\mathbf{x})$ that best characterizes the relationship within a dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$. Current state-of-the-art SR methods can be categorized into three paradigms based on their search strategy and utilization of priors.

Iterative Search-based Methods: These methods treat SR as a discrete optimization problem, employing Genetic Programming (McConaghy, 2011; Lu et al., 2022; Zhang et al., 2022), Reinforcement Learning (Petersen et al., 2021; Liu et al., 2024), Gradients Optimization (Kim et al., 2021), or Exhaustive Search with divide and conquer strategy (Udrescu & Tegmark, 2020; Udrescu et al., 2020). While highly flexible and adaptable to varying dimensions, they suffer from the "curse of dimensionality" as the search space expands exponentially with the number of operators and variables, leading to significant computational overhead for each individual task.

Pre-trained Neural Solvers: Inspired by the success of Large Language Models, these approaches (Valipour et al., 2021; Luca et al., 2021; Li et al., 2023; Kamienny et al., 2022; Wu et al., 2023) frame SR as a sequence-to-sequence translation task. By pre-training on massive synthetic data-expression pairs, they learn a neural prior that maps numerical distributions to symbolic structures. These models enable near-instantaneous inference but are often constrained by the distribution of the pre-training corpus and struggle with out-of-distribution (OOD) functional forms.

Hybrid Refinement Models: To bridge the gap between efficiency and accuracy, hybrid frameworks utilize pre-trained models to provide a warm-start or structural skeleton, which is subsequently refined via GP (Meidani et al., 2023), Reinforcement Learning (Kamienny et al., 2023; Liu et al., 2023; Shojaee et al., 2023; Mundhenk et al., 2021; Xu et al., 2024), or Montle Carlo Tree Search (Mundhenk et al., 2021; Landajuela et al., 2022). This paradigm achieves a superior trade-off between search efficiency and fitting precision.

2.2 Parameterized Equation Discovery

Unlike SR, PED addresses the challenge of discovering a shared governing law across multiple related environments. Formally, given a mathematical skeleton f, we observe a set of m dataset instances $\mathcal{D} = \{D_1, D_2, \ldots, D_m\}$, where each instance $D_i = \{(\mathbf{x}_j, y_j)\}_{j=1}^n \in \mathbb{R}^{n\times(d+1)}$ is generated using the same skeleton τ but with distinct parameter instantiations θ_i. The goal is to recover the discrete skeleton τ and the continuous parameters $\theta = \{\theta_1, \ldots, \theta_m\}$.

Direct approaches to PED are relatively sparse. Evolutionary aggregation methods, such as MvSR (Russeil et al., 2024) and MSR (Sen Fong & Motani, 2024), adapt GP-based engines by modifying the fitness function to aggregate performance (e.g., via max or harmonic mean) across multiple views. Similarly, general-purpose frameworks like PySR (Cranmer, 2023) and PhySO (Tenachi et al., 2024) can be adapted for PED through multi-dataset error averaging. However, these *instance-specific* methods treat each task family as an isolated optimization problem, requiring an exhaustive search from scratch and failing to exploit the latent structural similarities across different physical domains.

Another research line involves differentiable PED, notably EQL-style methods (Zhang et al., 2023) like SEQL and HEQL. By employing sparsity-promoting regularizations (e.g., L_0 or group sparsity) on neural network weights with symbolic activation functions, they attempt to "prune" a dense network into a symbolic form. While offering a gradient-based perspective, their search space is restricted by the predefined network depth, and they are notoriously sensitive to hyperparameters, often leading to sub-optimal structural recovery in noisy scenarios.

The research gap: Search-based methods treat skeletons as latent variables discovered via iterative fitness optimization, whereas DisPV directly infers them from a structural prior. We present the ***first pre-trained neural solver for PED***, which learns this prior by disentangling invariant skeletons from instance-specific constants across diverse environments.

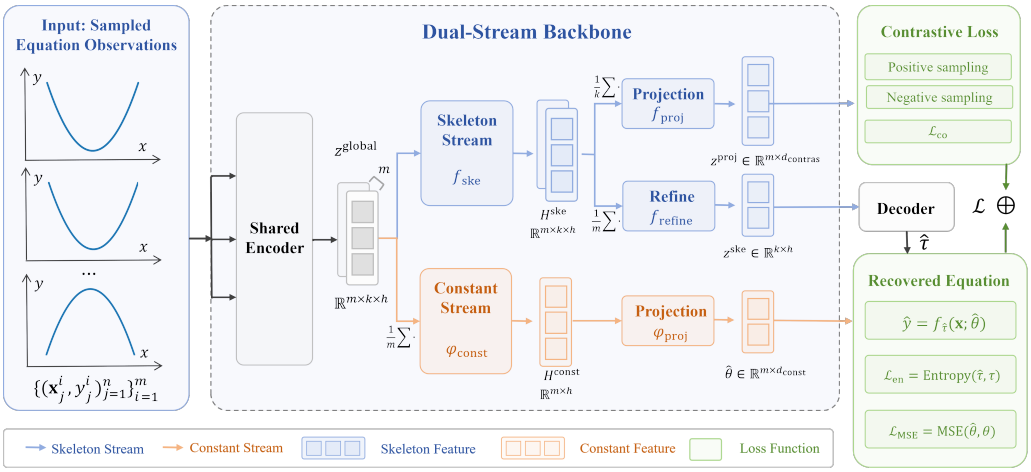


Figure 2: Overview of DisPV for PED. Given m sub-datasets generated by sampling the same parameterized equation under different parameter configurations, a shared encoder extracts a global representation, which is disentangled into skeleton and constant streams. The skeleton stream learns environment-invariant representations via contrastive learning and cross-instance fusion, while the constant stream predicts instance-specific parameters. The two streams are jointly optimized for skeleton generation and parameter reconstruction.

3 METHOD

To address the challenge of parameter-induced structural ambiguity in PED, we propose a dual-stream disentanglement framework. As illustrated in Fig. 2, our architecture explicitly separates the extraction of topological skeletons from the regression of numerical constants. By leveraging instance-level contrastive learning and cross-instance fusion, the model discovers invariant mathematical structures while maintaining parameter estimation.

3.1 GLOBAL FEATURE EXTRACTION

First, a shared neural encoder processes the input instances to extract a unified global representation $z^{\text{global}} = \text{Encoder}(\mathcal{D}) \in \mathbb{R}^{m\times k\times h}$, where the encoder is a SetTransformer Lee et al. (2019) which is permutation invariant for the order of the input. And k is the hyperapameter of SetTransformer's indicateing the number of seed vectors of PMA, and h is the hidden dimension. z^{global} contains highly entangled structural and parametric information, which is subsequently bifurcated into two orthogonal pathways.

3.2 SKELETON STREAM

The upper stream focuses exclusively on identifying the discrete mathematical structure. The global feature is first mapped into a dedicated skeleton feature space via a multi-layer perceptron f_{ske}: $H^{\text{ske}} = f_{\text{ske}}(z^{\text{global}}) \in \mathbb{R}^{m\times k\times h}$. To ensure that H^{ske} captures only the structural invariants and discards parameter-specific fluctuations, the skeleton stream is further split into two distinct mechanisms: Topology Anchoring and Cross-Instance Fusion.

Topology Anchoring via Contrastive Learning: To explicitly combat structural ambiguity, we apply a readout operation by averaging H^{ske} over the sequence dimension k. This distills the sequence into an instance-level macroscopic topological fingerprint, which is then mapped to a contrastive projection space:

$$z^{\text{pro}} = f_{\text{proj}}\left(\frac{1}{k}\sum_{j=1}^{k} H^{\text{ske}}_{:,j,:}\right) \in \mathbb{R}^{m\times d_{\text{contras}}}, \tag{1}$$

where d_{contras} is a hyperparameter which determines the size of feature dimension. For a training batch containing B different skeletons, each with m instances, we construct a Supervised Con-

trastive Loss ($\mathcal{L}_{co}$). The m instances sharing the same skeleton τ are treated as positive pairs, while instances originating from different skeletons act as hard negatives. Let I be the set of indices in a batch, and $P(i)$ be the set of indices of positive samples for instance i:

$$\mathcal{L}_{co} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i^{\text{pro}} \cdot z_p^{\text{pro}} / \Gamma)}{\sum_{a \in I \setminus \{i\}} \exp(z_i^{\text{pro}} \cdot z_a^{\text{pro}} / \Gamma)}, \tag{2}$$

where Γ is a temperature hyperparameter. This objective forces the network to map numerically dissimilar datasets (e.g., $1.2x^2$ and $0.5x^2$) to the same topological anchor in the feature space.

Cross-Instance Fusion and Decoding: To generate the final symbolic expression, we must aggregate the features from all m instances to filter out instance-specific information. We average H^{ske} over the instance dimension m and pass it through a self-attention refinement module f_{refine}:

$$z^{\text{ske}} = f_{\text{refine}} \left(\frac{1}{m} \sum_{i=1}^{m} H_{i,:,:}^{\text{ske}} \right) \in \mathbb{R}^{k \times h}. \tag{3}$$

The distilled structural feature z^{ske} is fed into the Decoder (a Transformer) to auto-regressively generate the predicted skeleton sequence $\hat{\tau} \in \mathbb{R}^L$, optimized via standard Cross-Entropy loss ($\mathcal{L}_{\text{en}}$), where L is the maximum length of the equation's traversal.

3.3 Parameter Stream

Parallel to the skeleton stream, the lower pathway is dedicated to recovering the continuous physical constants θ_i for each specific instance. We project z^{global} into the parameter space via a MLP φ_{const}. Since parameters are global properties of the equation and independent of the latent sequence length, we apply average pooling over the k dimension: $H^{\text{const}} = \frac{1}{k} \sum_{j=1}^{k} \varphi_{\text{const}}(Z^{\text{global}})_{:,j,:} \in \mathbb{R}^{m \times h}$.

Unlike the skeleton stream, we strictly preserve the instance dimension m here, as each dataset D_i possesses unique parameters. A linear prediction head φ_{proj} decodes these features into the predicted constants $\hat{\theta} \in \mathbb{R}^{m \times d_{\text{const}}}$. This prediction is supervised by a Mean Squared Error (MSE) loss against the ground-truth parameters: $\mathcal{L}_{\text{MSE}} = \frac{1}{m} \sum_{i=1}^{m} ||\hat{\theta}_i - \theta_i||_2^2$.

3.4 Overall Objective

The entire dual-stream architecture is trained end-to-end. The full objective function is a weighted sum of the sequence generation loss, the topology anchoring contrastive loss, and the parameter reconstruction loss:

$$\mathcal{L} = \mathcal{L}_{\text{en}} + \lambda_1 \mathcal{L}_{co} + \lambda_2 \mathcal{L}_{\text{MSE}}. \tag{4}$$

where λ_1 and λ_2 are scaling coefficients. This joint optimization mathematically guarantees that the feature space is strictly disentangled, yielding a framework capable of exact parameterized equation discovery.

3.5 DisPV-Guided Structural Refinement

While the pre-trained DisPV provides high-fidelity high-quality skeleton candidates, extreme Out-Of-Distribution (OOD) tasks may require fine-grained adjustments. Unlike traditional GP that searches from a random population, we utilize DisPV's top-ϵ beam search results as the **initial population** for a lightweight Genetic Programming (GP) refinement.

To adapt to the PED setting, we define the multi-environment fitness function as: $\text{Fitness}(\tau) = \frac{1}{m} \sum_{i=1}^{m} R^2(D_i, \tau; \hat{\theta}_i)$, where $\hat{\theta}_i$ are the task-specific constants optimized via L-BFGS for each dataset. Crucially, because DisPV provides a near-optimal structural starting point, this refinement process converges in orders of magnitude fewer generations than traditional SR, maintaining the overall efficiency of the framework.

4 EXPERIMENTS

4.1 SETTINGS

Dataset. Following NeSymReS (Luca et al., 2021), we generate 5M training and 1k validation equations by sampling m parameter sets per skeleton from $[-2, 2]$, with up to 3 parameters per equation. The two sets are non-overlapping. For evaluation, we test on 36 parameterized equations from public benchmarks, sampling m parameter groups per equation. Details of the benchmarks and experimental settings are in Appendix D and B, respectively.

Baselines. To comprehensively evaluate the effectiveness of DisPV, we benchmarked it against five state-of-the-art PED methods, HEQL (Zhang et al., 2023), SEQL (Zhang et al., 2023), MSR (Sen Fong & Motani, 2024), PySR (Cranmer, 2023), and PhySO (Tenachi et al., 2024). Details of these methods please refer to Appendix A.

Evaluating Metric. We evaluate each discovered equation by its average coefficient of determination (R^2) across all m sub-datasets:

$$\bar{R}^2 = \frac{1}{m}\sum_{j=1}^{m} R^2(y^j, f_{\hat{\tau}}(\mathbf{x}^j; \hat{\theta}_i)), \tag{5}$$

where $f_{\hat{\tau}}$ is the predicted expression, $\hat{\theta}_j$ are optimized via L-BFGS for each sub-dataset D_i, and R^2 is defined as

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}. \tag{6}$$

Here $R^2 > 0$ indicates predictions superior to the mean; $R^2 = 1$ is a perfect fit; $R^2 < 0$ denotes failure. Each experiment is repeated over 10 independent runs and the maximum $\bar{R}^2$ are reported. Additionally, the equation complexity of the best-performed run are reported which measured by the number of basic operators and terminal nodes.

4.2 STRUCTURAL CONSISTENCY ANALYSIS

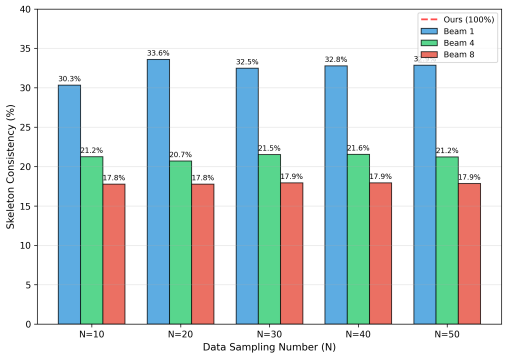


Figure 3: Skeleton consistency under varying sampling numbers and beam sizes.

A naive approach to PED applies traditional pre-trained SR models (e.g., NeSymReS (Biggio et al., 2021)) independently to each environment, attempting to consolidate the predictions post-hoc via voting. However, this *instance-wise inference* is highly vulnerable to parameter-induced structural ambiguity. To quantify this, we test the structural consistency of NeSymReS across 50 PED task families (protocol and equivalence verification detailed in Appendix C).

As Fig. 3 illustrates, NeSymReS rarely maintains semantic consistency across environments (17%–35% success). Crucially, we observe a beam search paradox: as beam size increases from 1 to 8, skeleton consistency consistently *declines*. Without cross-environment constraints, a wider search space simply allows the model to overfit local distributional quirks induced by specific θ values, favoring spurious skeletons over the true invariant. In contrast, **DisPV achieves 100% consistency across all benchmarks**. By aggregating all sub-dataset features into a single latent structural representation, DisPV performs joint inference over the entire task family. This underscores that PED necessitates a collective structural representation rather than repeated, isolated SR.

4.3 Overall Evaluation

R^2 and complexity analysis. Fig. 4 displays the average R^2 and complexity across all benchmarks for 6 different methods. The closer the dot is to the upper left corner, the better the trade-off. PySR, PhySO, and DisPV (Ours) achieve significantly higher R^2 than MSR, SEQL, and HEQL, despite their higher complexity. Further, DisPV achieves a state-of-the-art R^2 of 0.900 with a complexity of 11 operators, outperforming PySR (0.827) by 0.073 in R^2 while maintaining comparable complexity. Traditional search-based methods like PySR and PhySO achieve lower R^2 here because their "averaging fitness" strategy provides insufficient pressure to decouple invariant structures across different environments, often converging to oversimplified or spurious forms. In contrast, the differentiable methods HEQL and SEQL perform poorly, as the heterogeneous activation functions and sparse regularizations often lead to vanishing gradients or local optima during the weight-sharing optimization process. MSR is constrained by its pre-defined search space thus has limited generalization.

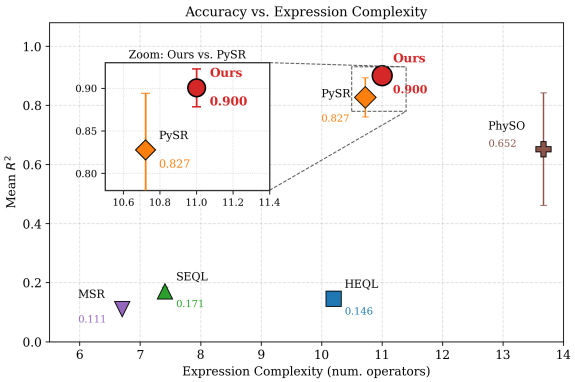


Figure 4: R^2 **performance vs. equation complexity.** DisPV maintains high symbolic fidelity with the trade of acceptable additional complexity.

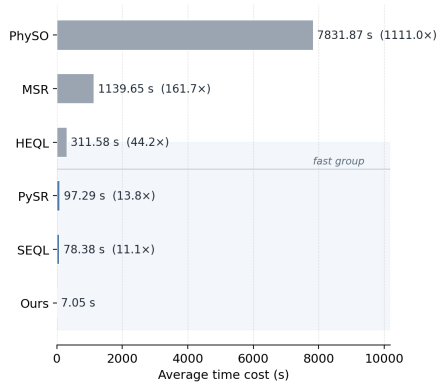


Figure 5: **Time cost.** DisPV achieves significant time advance across 5 methods.

Time cost. Training DisPV requires about 14 hours on a cluster of 4 RTX 5090 GPUs (30GB) using our 5M pre-training dataset. While the training phase is computationally intensive, once converged, the model operates with high inference efficiency. As illustrated in Fig. 5, the pure neural inference of DisPV takes an average of 7.05 seconds per problem, achieving over a $10\times$ speedup compared to heuristic search baselines like PySR (97.29 s) and SEQL (78.38 s).

It is worth noting that while our full pipeline supports optional genetic programming (GP) refinement for edge cases, DisPV's high-fidelity skeleton candidatas provide an exceptional "warm-start" initialization. Unlike traditional methods that search from scratch, DisPV directly places the evolutionary search near the global optimum, allowing the optional GP refiner to converge in a fraction of the iterations and maintaining an overall computational overhead vastly lower than traditional exhaustive search.

Generalization on Out-of-Distribution (OOD) Parameters. To evaluate robustness against parameter shifts, we compare the exact recovery rate of DisPV against PySR across in-domain ($[-2, 2]$) and out-of-domain ($[2, 6]$) parameter intervals (Table 1). Under in-domain regimes, DisPV substantially outperforms PySR (58.3% vs. 42.7%), demonstrating the decisive advantage of our pre-trained structural prior. For search-from-scratch algorithms like PySR, OOD parameter shifts pose no inherent distributional hurdle because optimization is conducted independently per instance. Conversely, while pre-trained neural networks typically suffer performance degradation under distributional shifts, DisPV maintains a competitive recovery rate of 41.6% (15 of 36 recovered) in the OOD regime, closely trailing PySR (44.4%) (16 of 36 recovered). This re-

Table 1: The recovery rate on in-domain and out-of-domain datasets.

Interval	PySR	Ours
$[-2, 2]$	42.7%	58.3%
$[2, 6]$	44.4%	41.6%

silience confirms that our dual-stream architecture captures genuine structural invariants rather than overfitting to specific parameter domains, while the integrated GP refinement module effectively acts as a safety net to rectify residual parameter discrepancies.

4.4 Representation Disentanglement Analysis

Skeleton Structural Invariance. To verify whether the skeleton stream reliably extracts the underlying mathematical law across different environmental observations, we evaluate its latent representation space across 14 public parameterized benchmarks. For each benchmark equation, we sample 50 distinct parameter configurations, each inducing substantially divergent numerical responses. As visualized via t-SNE in Fig. 6, observations originating from the same governing skeleton form tight, cohesive clusters, completely separated by clean structural boundaries despite extreme within-class parameter variations.

This qualitative observation is further corroborated by the Representation Similarity Analysis (RSA) on the `nguyen1_c` benchmark (Fig. 7, left). Across all 50 different parameter realizations, the pairwise cosine similarities of the extracted skeleton embeddings remain uniformly near 1.0 (mean similarity > 0.98).

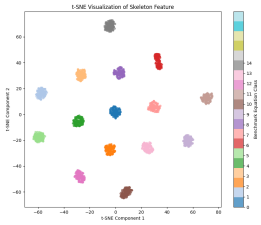


Figure 6: t-SNE visualization of 14 skeleton representations.

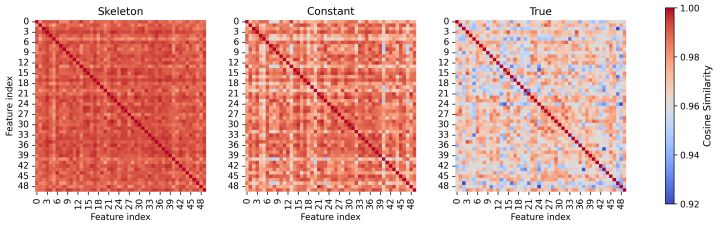


Figure 7: Representation Similarity Analysis (RSA) matrices comparing cosine similarities of extracted features with ground-truth parameter vectors.

Constant-Specific Variation and Disentanglement. Conversely, the constant stream must retain environmental specificity rather than collapse into an invariant representation. As shown in Fig. 7 (middle), the pairwise similarity matrix of the constant features exhibits pronounced textural variation across environments, visually matching the geometric structure of the raw parameter ground-truth matrix (Fig. 7, right).

Considering the severe dimensionality gap between the high-dimensional latent space and the raw parameter manifold, this alignment confirms that the constant stream actively captures the continuous topology of instance-specific coefficients. Synthesized with the parameter-agnostic properties observed in the skeleton stream, these results provide concrete evidence that DisPV successfully achieves representation-level disentanglement: isolating the global mathematical skeleton in one branch while faithfully delegating environmental parametrization to the other.

4.5 Ablation Study

Table 2: Ablation study evaluating cross-environment aggregation strategies and the dual-stream components of DisPV.

Model	> 0.9999	> 0.999	> 0.99	> 0.9	> 0.8	R^2 (mean)
Pure GP	5	10	18	32	41	0.4718
Early-Concat	56	62	70	95	114	0.6876
Feature-Pooling	68	74	85	109	126	0.7716
DisPV (w/o CL)	70	75	91	124	142	0.8098
DisPV (Zero-shot)	76	85	98	121	134	0.7948
DisPV + GP (Ours)	**97**	**115**	**131**	**159**	**166**	**0.9133**

To isolate the contributions of our proposed architecture, we perform an ablation study comparing different cross-environment aggregation strategies and structural components (Table 2). We evaluate Early-Concat, which naively flattens sub-datasets across environments prior to encoding, and Feature-Pooling, which encodes environments independently before aggregating latent features. Feature-Pooling significantly outperforms Early-Concat (R^2: 0.77 vs. 0.68), demonstrating that explicitly maintaining environmental boundaries during initial encoding prevents the model from conflating heterogeneous data distributions. Furthermore, our dual-stream DisPV surpasses Feature-Pooling by explicitly disentangling the structural skeleton from instance-specific parameters. Adding the contrastive objective improves the high-fidelity recovery rates ($R^2 > 0.99$ increases from 91 to 98). Although DisPV (w/o CL) attains a slightly higher mean R^2 due to an accumulation of moderately fitting approximations ($0.8 < R^2 < 0.9$), these candidates frequently represent spurious aliased skeletons that overfit local data without capturing the true invariant. By enforcing explicit structural separation, DisPV effectively suppresses such deceptive local optima, directing the model toward exact physical skeleton recovery. Finally, DisPV + GP achieves the highest performance (R^2: 0.9133), validating our core hypothesis: the pre-trained DisPV provides a highly accurate, invariant structural prior that dramatically accelerates and stabilizes the genetic programming refinement process, overcoming the limitations of searching from scratch (Pure GP).

4.6 Sensitivity Analysis

To investigate how environmental diversity impacts representation learning, we evaluate the zero-shot recovery performance of DisPV (without GP refinement) trained under varying candidate pool sizes $m \in \{10, 20, 30\}$. While each equation family contains m sub-datasets, we randomly sample a subset of 5 environments per batch iteration during training to prevent out-of-memory overhead while maximizing stochastic combinatorial exposure across epochs.

As shown in Figure 8, the number of valid samples successfully recovered across all R^2 thresholds exhibits an inverted U-shaped pattern, with $m = 20$ consistently outperforming both $m = 10$ and $m = 30$. Although expanding m from 10 to 20 effectively breaks parameter-induced structural ambiguity by providing richer cross-environment variations, performance degrades when m increases further to 30. We attribute this decline to an optimization trade-off: scaling the candidate pool to $m = 30$ exponentially expands the combinatorial sampling space ($\binom{30}{5} \gg \binom{20}{5}$), significantly increasing the variance of multi-environment latent distributions. Under a fixed training budget (identical epochs), the fixed-capacity encoder struggles to fully converge on this vastly enlarged configuration space, indicating that $m = 20$ provides the optimal equilibrium between structural diversity and optimization stability.

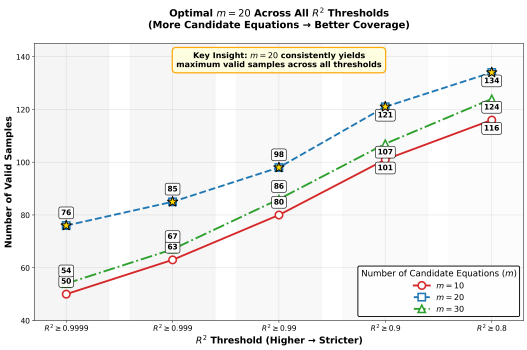


Figure 8: Parameter setting selection.

5 Conclusion

In this paper, we propose DisPV, an end-to-end dual-stream framework for Parameterized Equation Discovery (PED) that explicitly disentangles universal symbolic skeletons from instance-specific parameter variations. By combining cross-environment contrastive learning with parallel parameter estimation, DisPV extracts robust, environment-invariant structural representations while preserving instance-level coefficients. When paired with genetic programming refinement, DisPV achieves state-of-the-art R^2, while cutting inference latency by an order of magnitude compared to traditional search-based baselines. Future work will investigate large-scale symbolic pre-training and adaptive search strategies to further expand the structural out-of-distribution (OOD) generalization of the framework.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and proofreading of the manuscript. We have not used generative AI tools for research ideation, experiment design, or code generation, and the use of generative AI for data creation, figure generation, or co-authorship are not applicable to this work. Additionally, we used generative AI tools for improving the clarity and readability of the written text. We have reviewed all AI-assisted work. Specifically, all LLM-suggested language improvements were carefully checked by the authors to ensure academic accuracy, proper technical terminology, and the absence of factual errors or hallucinated references. The authors manually verified that the AI-assisted text accurately reflects the original research content and made necessary corrections where needed. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have made concrete efforts to ensure the reproducibility of this work. Detailed hyperparameter configurations for the Genetic Programming refinement are provided in Appendix B. The dataset generation protocol (including the sampling ranges and equation templates) and the complete list of 36 benchmark equations are described in Appendix D, respectively. Upon acceptance, we will publicly release the source code for DisPV, including all training and evaluation scripts, to facilitate future research. All theoretical claims and algorithmic steps are thoroughly explained in the main text to enable independent verification.

REFERENCES

Luca Biggio, Tommaso Bendinelli, Alexander Neitz, Aurelien Lucchi, and Giambattista Parascandolo. Neural symbolic regression that scales. In *International Conference on Machine Learning*, pp. 936–945. Pmlr, 2021.

Miles Cranmer. Interpretable machine learning for science with pysr and symbolicregression.jl, 2023. URL https://arxiv.org/abs/2305.01582.

Kei Sen Fong and Mehul Motani. Distilsr: A distilled version of gene expression programming symbolic regression. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, pp. 567–570, 2023.

Samuel Holt, Zhaozhi Qian, and Mihaela van der Schaar. Deep generative symbolic regression. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=o7koEEMA1bR.

Pierre-Alexandre Kamienny, Stéphane d'Ascoli, Guillaume Lample, and François Charton. End-to-end symbolic regression with transformers. *Advances in Neural Information Processing Systems*, 35:10269–10281, 2022.

Pierre-Alexandre Kamienny, Guillaume Lample, Sylvain Lamprier, and Marco Virgolin. Deep generative symbolic regression with monte-carlo-tree-search. In *International Conference on Machine Learning*, pp. 15655–15668. PMLR, 2023.

Samuel Kim, Peter Y. Lu, Srijon Mukherjee, Michael Gilbert, Li Jing, Vladimir Čeperić, and Marin Soljačić. Integration of neural network-based symbolic regression in deep learning for scientific discovery. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9):4166–4177, 2021. doi: 10.1109/TNNLS.2020.3017010.

Mikel Landajuela, Chak Shing Lee, Jiachen Yang, Ruben Glatt, Cláudio P. Santiago, Ignacio Aravena, Terrell Nathan Mundhenk, Garrett Mulcahy, and Brenden K. Petersen. A unified framework for deep symbolic regression. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/dbca58f35bddc6e4003b2dd80e42f838-Abstract-Conference.html.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Wenqiang Li, Weijun Li, Linjun Sun, Min Wu, Lina Yu, Jingyi Liu, Yanjie Li, and Songsong Tian. Transformer-based model for symbolic regression via joint supervised learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ULzyv9M1j5.

Jingyi Liu, Weijun Li, Lina Yu, Min Wu, Linjun Sun, Wenqiang Li, and Yanjie Li. Snr: Symbolic network-based rectifiable learning framework for symbolic regression. *Neural Networks*, 165: 1021–1034, 2023.

Jingyi Liu, Min Wu, Lina Yu, Weijun Li, Wenqiang Li, Yanjie Li, Meilan Hao, Yusong Deng, and Wei Shu. Camo: Capturing the modularity by end-to-end models for symbolic regression. *Knowledge-Based Systems*, pp. 112747, 2024.

Qiang Lu, Congwen Xu, Jake Luo, and Zhiguang Wang. Ab-gep: Adversarial bandit gene expression programming for symbolic regression. *Swarm and Evolutionary Computation*, 75:101197, 2022.

Biggio Luca, Bendinelli Tommaso, Neitz Alexander, Lucchi Aurelien, and Parascandolo Giambattista. Neural symbolic regression that scales. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 936–945. PMLR, 2021.

Trent McConaghy. Ffx: Fast, scalable, deterministic symbolic regression technology. *Genetic Programming Theory and Practice IX*, pp. 235–260, 2011.

Kazem Meidani, Parshin Shojaee, Chandan K Reddy, and Amir Barati Farimani. Snip: Bridging mathematical symbolic and numeric realms with unified pre-training. In *NeurIPS 2023 AI for Science Workshop*, 2023.

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*, pp. 24912–24923, 2021.

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.

Etienne Russeil, Fabrício Olivetti de França, Konstantin Malanchev, Bogdan Burlacu, Emille Ishida, Marion Leroux, Clément Michelin, Guillaume Moinard, and Emmanuel Gangler. Multiview symbolic regression. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 961–970, 2024.

Kei Sen Fong and Mehul Motani. Multi-level symbolic regression: Function structure learning for multi-level data. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 2890–2898. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/sen-fong24a.html.

Parshin Shojaee, Kazem Meidani, Amir Barati Farimani, and Chandan Reddy. Transformer-based planning for symbolic regression. *Advances in Neural Information Processing Systems*, 36: 45907–45919, 2023.

Wassim Tenachi, Rodrigo Ibata, Thibaut L. François, and Foivos I. Diakogiannis. Class Symbolic Regression: Gotta Fit 'Em All. *The Astrophysical Journal Letters*, 969(2):arXiv:2312.01816, jul 2024. doi: 10.3847/2041-8213/ad5970.

Silviu-Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

Silviu-Marian Udrescu, Andrew Tan, Jiahai Feng, Orisvaldo Neto, Tailin Wu, and Max Tegmark. Ai feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. *Advances in Neural Information Processing Systems*, 33:4860–4871, 2020.

Mojtaba Valipour, Bowen You, Maysum Panju, and Ali Ghodsi. Symbolicgpt: A generative transformer model for symbolic regression. *arXiv preprint arXiv:2106.14131*, 2021.

Min Wu, Weijun Li, Lina Yu, Linjun Sun, Jingyi Liu, and Wenqiang Li. Discovering mathematical expressions through deepsymnet: A classification-based symbolic regression framework. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023. doi: 10.1109/TNNLS.2023.3332400.

Yilong Xu, Yang Liu, and Hao Sun. Reinforcement symbolic regression machine. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=PJVUWpPnZC.

Hengzhe Zhang, Aimin Zhou, Hong Qian, and Hu Zhang. Ps-tree: A piecewise symbolic regression tree. *Swarm and Evolutionary Computation*, 71:101061, 2022.

Michael Zhang, Samuel Kim, Peter Y Lu, and Marin Soljačić. Deep learning and symbolic regression for discovering parametric equations. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

A Details of Compared Methods

We compare with 5 state-of-the-art methods, they are:

- **HEQL** (Zhang et al., 2023): Incorporates a meta-weight unit (MWU) that, using t as input, outputs the EQL network's weights, akin to a hypernetwork.
- **SEQL** (Zhang et al., 2023): Utilizes a separate EQL network for each "time step" (varying dimension), with shared L_0 regularization to promote group sparsity.
- **MSR** (Sen Fong & Motani, 2024): Employs DistilSR (Fong & Motani, 2023) to search for expression skeletons within a defined constrained space, adapting the fitness function to the harmonic mean of MSE across different datasets.
- **PySR** (Cranmer, 2023): A leading open-source High-Performance Computing (HPC) framework for symbolic regression which has the interface to handle PED tasks. It employs a highly optimized evolutionary algorithm and serves as a strong non-neural baseline.
- **PhySO** (Tenachi et al., 2024): Its symbolic regression module uses deep reinforcement learning to infer analytical physical laws that fit data points, searching in the space of functional forms.

B Details of Hyperparameter Settings

B.1 Reproducibility.

We implemented our GP search using the `DEAP`[1] package. For baseline comparisons, HEQL and SEQL[2], and MSR[3], we use the standard hyperparameters provided in their official repositories. Each algorithm is executed in 10 independent runs, with the maximum R^2, average computational time, and expression complexity reported. The configuration for `PySR`[4] and `PhySO`[5] please see Table B.1 and B.2.

Table B.1: Detailed Hyper-parameters for PySR Experiments.

Category	Parameter Value
Experimental Setup	
Independent Runs	10
Sub-datasets per Task (m)	5
Points per Sub-dataset (n)	100
Operator Set	
Binary Operators	$\{+, -, \times, /\}$
Unary Operators	$\{\sin, \cos, \exp, \log\}$
Search Constraints	
Max Expression Size	20
Objective Metric	Mean R^2 across tasks
Data Generation	
Input Range	$[-10, 10]$ (benchmark default)
Constant Optimization	L-BFGS (PySR default)

B.2 DisPV Hybrid Initialization (Warm-start)

The Genetic Programming (GP) component in our framework serves as a local structural refiner to polish the skeletons predicted by the pre-trained Transformer. We implement this using the `DEAP`

[1] https://github.com/DEAP/deap
[2] https://github.com/samuelkim314/DeepSymReg
[3] https://github.com/kentridgeai/MSR-Supplementary
[4] https://github.com/MilesCranmer/PySR
[5] https://physo.readthedocs.io/en/latest/index.html

Table B.2: Detailed Hyper-parameters for PhySO Experiments.

Category	Parameter Value
Search Engine	
Function Class	`physo.ClassSR`
Run Configuration	`config0b` (RL-based)
Number of Epochs	10
Stop Reward	$1.0 - 10^{-3}$
Operator Set	
Binary Operators	$\{\times, +, -, /\}$
Unary Operators	$\{\cdot^2, \sqrt{\cdot}, \cos, \sin, \exp, \log\}$
Constant Handling	
Task-specific Constants	`spe_free_consts_names` (c_{names})
Physical Constraints	
Input Units ($\mathbf{x}$)	X_{units} (problem-specific)
Output Units (y)	$[0, 0, 0]$ (dimensionless)
Execution Details	
Parallel Mode	True
CPU Cores	8

library. Unlike traditional GP that searches from a random population, our approach utilizes a hybrid initialization strategy to leverage the neural structural prior. The initial population is composed of two parts:

- **Neural Prior (20%):** Skeletons generated by the DisPV decoder via beam search. These individuals are parsed from symbolic strings into `PrimitiveTree` structures, with structural integrity checks to ensure arity consistency.
- **Stochastic Diversity (80%):** Individuals generated using the "Half-and-Half" method to maintain population diversity and explore novel functional forms.

GP Hyper-parameters. The configuration for the evolutionary process is detailed in Table B.3.

Table B.3: Hyper-parameters for GP Structural Refinement.

Parameter	Setting / Value
Evolutionary Control	
Population Size	100
Number of Generations	100
Crossover Probability	0.3
Mutation Probability	0.5
Selection Method	Tournament (size = 3)
Primitive Set	
Binary Operators	$\{+, -, \times, /, \text{pow}\}$
Unary Operators	$\{\sin, \cos, \exp, \log\}$
Terminals	$\{x_1, \ldots, x_d, c_0, \ldots, c_3\}$
Structural Constraints	
Max Tree Height	12
Initialization Method	Half-and-Half (depth 1–2)
Early Stopping Threshold	1×10^{-3} (MSE)
Numerical Optimizer	
Algorithm	L-BFGS-B

C Details of the Symbolic Equivalence Checker

To rigorously evaluate the Structural Consistency of the discovered skeletons across different sub-datasets, we developed a multi-tier symbolic equivalence checker. Since Parameterized Equation Discovery (PED) focuses on the invariance of the functional form rather than specific numerical coefficients, our checker treats all task-specific constants as generic placeholders.

C.1 Skeleton Preprocessing

Before comparison, each predicted skeleton $\hat{\tau}$ is neutralized by replacing all numerical constants $\{c_0, c_1, \ldots, c_9\}$ with a unified placeholder C. This ensures that the checker focuses exclusively on the mathematical structure (e.g., $C \cdot \sin(x)$ is considered equivalent to $2.5 \cdot \sin(x)$).

C.2 Three-Tier Verification Protocol

To handle the potential complexity of symbolic expressions where standard string matching fails, we implement a tiered verification logic:

1. Symbolic Expansion and Simplification: We utilize the `sympy` library to perform algebraic expansion and simplification. Each expression is transformed into a canonical form using `sp.simplify()` and `sp.expand()`.
2. Algebraic Identity Check: We employ the `equals()` method in `sympy`, which tests if the difference between two expressions is identically zero across their domain. This is the most reliable method for identifying identities such as $\sin^2(x) + \cos^2(x) \equiv 1$.
3. Randomized Numerical Probing: In cases where symbolic simplification is computationally inconclusive, we implement a randomized numerical probe as a fallback. We evaluate the expressions at $n = 20$ random sampling points $(\mathbf{x}, C)$ within the domain $[-5, 5]$. Two skeletons are deemed equivalent only if the maximum absolute difference across all sampling points is below a strict tolerance of $\epsilon = 10^{-8}$.

C.3 Consistency Ratio Calculation

The Consistency Ratio for a task family $\mathcal{E}$ is defined as the proportion of successful runs where the model returns semantically identical skeletons across all m sub-datasets. We calculate this over various sampling intervals $N \in \{10, 20, 30, 40, 50\}$ to observe the stability of the algorithms under different data regimes. This metric provides a direct measure of the model's ability to "see through" distributional shifts and converge on a universal physical law.

D Details of Tested Benchmarks

The details of the tested benchmarks are listed in Table D.4. We sample the parameter values from interval $[-2, 2]$. The independent variable are sampled from interval $[-10, 10]$. If the independent variable has physical meaning like time t, we sample it from interval $[0, 10]$. The total number of sampled data points are 100.

Table D.4: Public datasets. x, y, z are independent variables, a, b, c are the parameters in the equation.

Benchmark	Equation	Parameters	Description
Nguyen1_c	$ax^3 + bx^2 + cx$	a, b, c	Public equation from SR
Nguyen2_c	$ax^4 + bx^3 + cx^2 + x$	a, b, c	Public equation from SR
Nguyen5_c	$\sin(x^2)\cos(x) - a$	a	Public equation from SR
Nguyen7_c	$\log(x + a) + \log(x^2 + b)$	a, b	Public equation from SR
Nguyen8_c	$\sqrt{ax}$	a	Public equation from SR
Nguyen10_c	$\sin(ax)\cos(by)$	a, b	Public equation from SR
Constant4	ax^y	a	Public equation from SR
Constant6	x^a	a	Public equation from SR
GrammarVAE1	$a + x + \sin(x)$	a	Public equation from SR
Jin2	$ax^2 + by^3 + c$	a, b, c	Public equation from SR
Jin4	$ae^x + b\cos(y)$	a, b	Public equation from SR
Jin5	$a\sin(x)\cos(y)$	a	Public equation from SR
Nguyend3_1c	$ax^3 + y^2 + z$	a	Synthetic equation
Trigfunc3c	$\cos(x)\sin(y) + az^2$	a	Synthetic equation
Expfunc3c	$ae^{(x+y+z)}$	a	Synthetic equation
Physical1	$s = v_0 + \frac{1}{2}gt^2$	v_0, g	Free fall displacement equation
Physical2	$x(t) = A\cos(\omega t)$	A, ω	Simple harmonic vibration displacement equation (without initial phase)
Physical3	$x(t) = Ae^{-\gamma t}\cos(\omega t)$	A, γ, ω	Dampled vibration (no initial phase)
Physical4	$V = IR$	I	Ohm's law
Physical5	$Q = Q_0 e^{-\frac{t}{RC}}$	Q_0, R, C	Capacitor discharge formula
Physical6	$A = \epsilon cl$	ϵ	Lambert-beer law
Physical7	$x = \frac{\lambda L}{d}m$	λ, d, L	The position of the interference fringes of light
Physical8	$v = -eu$	e	Expression of velocity after elastic collision (Small ball and wall)
Physical9	$B = \mu H$	μ	Primary approximation of hysteresis loop (magnetic induction intensity varying with magnetic field, linear region)
Physical10	$Ax + By + C$	A, B, C	Two dimensional stedy-state solution of heat conduction (passive)
Physical11	$A\sin(kx - \omega t)$	A, k, ω	Two-dimensional plane wave
Physical12	$Ae^{(-\frac{(x-x_0)^2+(y-y_0)^2}{2})}$	A, x_0, y_0	Two-dimensional Gaussian distribution (elliptical contour lines)
Physical13	$A\cos(kr - \omega t)$	A, k, ω	Two-dimensional circular wavefront
Physical14	$\frac{A}{\sqrt{(x-x_0)^2+(y-y_0)^2}}$	A, x_0, y_0	Two-dimensional electrostatic potential point source model
Physical15	$E_0\sin(kx - \omega t)$	E_0, k, ω	The plane wave solution of electromagnetic waves in an isotropic and uniform medium
Physical16	$\frac{A}{4\pi Dt}e^{-x^2/(4Dt)}$	A, D	Two-dimensional isotropic diffusion (concentration at the diffusion front)
Physical17	$A\sin(k_x x + k_y y - \pi t)$	A, k_x, k_y	Three-dimensional plane wave
Physical18	$ax + by + cz$	a, b, c	Three-dimensional steady-state linear solution of heat conduction (passive)
Physical19	$\rho_0 e^{-\frac{x^2+y^2+z^2}{2\sigma^2}}$	ρ_0, σ	Three-dimensional isotropic Gaussian distribution
Physical20	$T_t = T_{env} + (T_0 - T_{env})e^{-kt}$	T_0, T_{env}, k	A linear model of volume resistance with respect to temperature and length
Physical21	$\rho_0(1 + \alpha T)(\frac{L}{A})$	ρ_0, α	Newton's law of cooling