# SurfaceBench: Can Self-Evolving LLMs Find the Equations of 3D Scientific Surfaces?

**Anonymous authors**
**Paper under double-blind review**

## Abstract

Equation discovery from data is a core challenge in machine learning for science, requiring the recovery of concise symbolic expressions that govern complex physical and geometric phenomena. Recent approaches with large language models (LLMs) show promise in symbolic regression, but their success often hinges on memorized formulas or overly simplified functional forms. Existing benchmarks exacerbate this limitation: they focus on scalar functions, ignore domain grounding, and rely on brittle string-matching based metrics that fail to capture scientific equivalence. We introduce SurfaceBench, the first comprehensive benchmark for symbolic surface discovery. SurfaceBench comprises *183* tasks across *15* categories of symbolic complexity, spanning explicit, implicit, and parametric equation representation forms. Each task includes ground-truth equations, variable semantics, and synthetically sampled three dimensional data. Unlike prior SR datasets, our tasks reflect surface-level structure, resist LLM memorization through novel symbolic compositions, and are grounded in scientific domains such as fluid dynamics, robotics, electromagnetics, and geometry. To evaluate equation discovery quality, we pair symbolic checks with geometry-aware metrics such as Chamfer and Hausdorff distances, capturing both algebraic fidelity and spatial reconstruction accuracy. Our experiments reveal that state-of-the-art frameworks, while occasionally successful on specific families, struggle to generalize across representation types and surface complexities. SurfaceBench thus establishes a challenging and diagnostic testbed that bridges symbolic reasoning with geometric reconstruction, enabling principled benchmarking of progress in compositional generalization, data-driven scientific induction, and geometry-aware reasoning with LLMs. The code for this paper can be found here: Code Repository

## 1 Introduction

Symbolic regression, or equation discovery, is a foundational task in computational scientific discovery. It refers to the process of recovering interpretable mathematical equations that best describe observed data. By converting raw observations into symbolic form, symbolic regression uncovers functional relationships and invariants that govern physical, biological, and engineered systems.

Existing approaches to symbolic regression have traditionally relied on domain experts to manually specify relevant variables, transformations, and functional forms (Brunton et al., 2016). This manual process is both time-consuming and prone to noise, limiting scalability and robustness in real-world scenarios. To overcome this, early methods employed genetic programming, evolutionary programming, and transformer-based planning (Cranmer, 2023; Biggio et al., 2021; Shojaee et al., 2023; Petersen et al., 2019; Landajuela et al., 2022; Kamienny et al., 2022), which have shown strong potential but remain purely data-driven and often require millions of iterations to search effectively. Recent advances leverage large language models (LLMs) as symbolic priors to guide equation discovery. Methods such as LLM-SR (Shojaee et al., 2025a), LaSR (Grayeli et al., 2024), SGA (Ma et al., 2024), and OpenEvolve (Sharma, 2025) incorporate domain knowledge in the discovery process and thereby increase the search efficiency. However, two key limitations persist: (i) LLM-based methods are prone to surface-level memorization of known equations, undermining

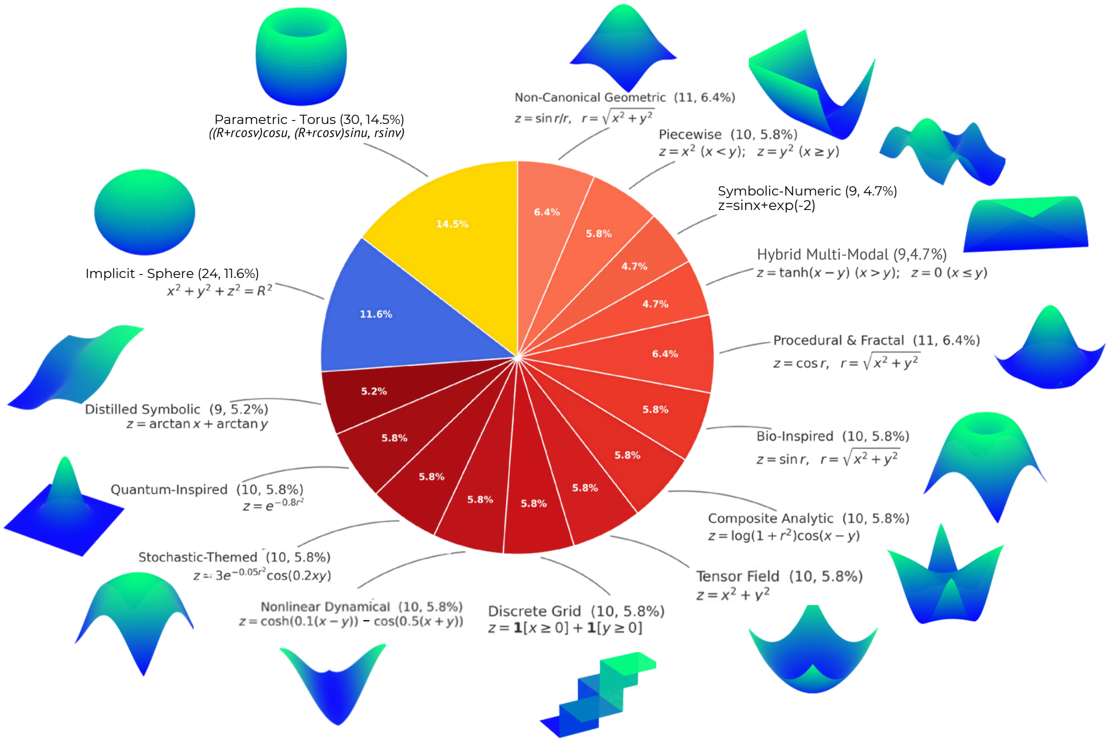


Figure 1: SurfaceBench: A benchmark suite for symbolic regression featuring 183 surface equations spanning 15 scientific domains. The benchmark covers three canonical equation representations: explicit (red), implicit (blue), and parametric (yellow), thus illustrating diverse surface structures and symbolic challenges.

true reasoning; and (ii) search remains difficult to steer without carefully designed prompting or feedback mechanisms, limiting their effectiveness in complex scientific domains.

Benchmarking symbolic regression methods remains an open challenge. Existing benchmarks are either synthetic (Shojaee et al., 2025b) or curated from canonical textbook equations (Udrescu & Tegmark, 2020), making them susceptible to memorization by LLMs and limited in their ability to represent the diversity of equations found in real scientific domains. While useful for controlled evaluation, such benchmarks do not adequately test reasoning, generalization, or equation diversity. LLMs can often reproduce known expressions through memorization, especially when benchmarks rely on well-documented equations. Moreover, current benchmarks are *predominantly scalar*, focusing on functions of the form $y = f(x)$, which *do not capture the multivariate and geometrically structured nature of equations common in scientific and engineering contexts.*

In contrast, surfaces are ubiquitous in scientific modeling and naturally span a richer set of mathematical forms (explicit, implicit, and parametric), each with distinct geometric and symbolic challenges. Recovering the governing surface equation from sampled $(x, y, z)$ data involves reasoning about multi-output couplings, latent coordinates, and representational non-uniqueness.

To address this gap, we introduce SurfaceBench (refer Figure 1), a benchmark suite for symbolic surface discovery. In contrast to traditional symbolic regression benchmarks, SurfaceBench targets surface-level reasoning, where rote memorization fails and geometric structure must be inferred from data alone. The benchmark includes surfaces with varied complexity, symmetry, and coordinate systems, thus foregrounding symbolic reasoning under structural ambiguity. Importantly, surface equations afford a unique advantage: multiple symbolic forms may yield visually or topologically equivalent geometries. This allows evaluation beyond exact string matches, using geometry-aware metrics such as Chamfer and Hausdorff distances. Together, these metrics provide a robust and semantically meaningful signal for evaluating symbolic recovery, even when the recovered expression differs from the ground truth in form. The primary contributions of our paper are:

- We introduce SurfaceBench, the first systematic and large-scale benchmark for symbolic surface discovery, comprising 183 explicit, implicit, and parametric surfaces spanning 15 scientific categories. SurfaceBench establishes a new paradigm for equation discovery that moves beyond scalar functions to structured, multi-output, geometry-aware expressions.
- We evaluate a broad spectrum of symbolic regression baselines and state-of-the-art LLM-guided discovery frameworks. Our comprehensive experiments reveal that current methods struggle to recover surface equations accurately, with equation recovery rates of only 4% for LLM-based frameworks and 6% for traditional symbolic regression methods. These results underscore the difficulty of symbolic surface discovery and the need for structure-aware reasoning frameworks.
- We provide an in-depth error taxonomy, decomposing *symbolic (structural)* versus *geometric (shape-level)* failure modes. Through targeted ablations, we analyze the impact of representation type, operator composition, and fine-tuning strategies, providing concrete failure diagnostics and generalization breakdowns that offer actionable design insights for future surface reasoning methods.

## 2 Challenges and Design of SurfaceBench

### 2.1 Benchmark Challenges and Motivation

SurfaceBench is a symbolic regression benchmark designed to evaluate symbolic reasoning capabilities that are critical for scientific equation discovery. Traditional symbolic regression tasks focus on scalar mappings such as $y = f(x)$, where evaluation is limited to one-dimensional dependencies and success is measured by numerical fit. SurfaceBench extends this paradigm to governing equations that describe full three-dimensional surfaces, which encapsulate multi-variable dependencies, invariances, and geometric constraints. The surfaces themselves are not the end goal, but serve as a structured medium through which the benchmark evaluates symbolic reasoning under complex geometric and physical constraints. By representing relationships as explicit, implicit, or parametric surfaces, it tests whether a model can recover equations that jointly capture variable interactions and geometric structure. This formulation moves symbolic regression beyond scalar curve fitting toward the more rigorous demands of scientific modeling, where systems are multi-output, coordinate-dependent, and often topologically complex. To systematically evaluate these capabilities, SurfaceBench is organized around five defining features that collectively characterize the key challenges of symbolic regression:

**(1) Multi-output coupling:** In conventional benchmarks, outputs are typically independent, allowing models to fit each dimension separately. SurfaceBench introduces multivariate targets that depend on several interacting variables—for instance, $z = \sin(x^2 + y^2)$, where changes in one variable alter curvature across the entire surface. This setup tests whether models can reason jointly over variables that together define a governing law, rather than treating them as separable regressions. Such coupling expands the symbolic search space and reflects the interdependencies typical of real-world physical systems (Moyano et al., 2021).

**(2) Latent coordinate systems and topology:** Many scientific laws admit concise symbolic forms only in transformed coordinate systems spherical, cylindrical, or other field aligned representations while observed data are typically provided in Cartesian space (Champion et al., 2019). SurfaceBench examines whether models can infer such latent transformations that reveal underlying invariances. The benchmark also includes surfaces with nontrivial topology holes, folds, or disconnected components that are best represented as implicit level sets $f(x, y, z) = 0$ or as parametric mappings $(x(u, v), y(u, v), z(u, v))$. These formulations challenge symbolic regression pipelines that assume scalar outputs and fixed templates, requiring models to infer both the algebraic law and its structural form.

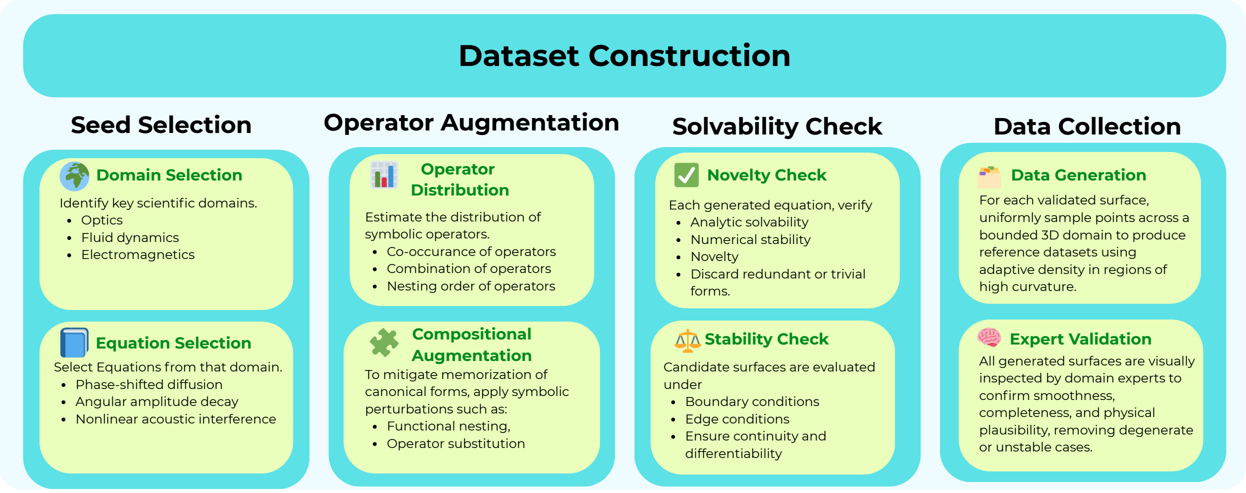
**(3) Symbolic non-uniqueness:** Multiple algebraically distinct expressions can describe identical behaviors. A sphere, for example, may appear implicitly as $x^2 + y^2 + z^2 = R^2$ or parametrically as

$(R\sin\phi\cos\theta,\ R\sin\phi\sin\theta,\ R\cos\phi)$. Trigonometric identities, affine transformations, and reparameterizations multiply the number of equivalent formulations, rendering string-level comparison unreliable. This representational non-uniqueness motivates the need for evaluation criteria that operate in geometric rather than symbolic space (Jiang et al., 2025).

**(4) Geometry-aware evaluation:** To address symbolic non-uniqueness, SurfaceBench evaluates predicted and reference equations through their induced geometry. Both are sampled as dense point clouds, aligned under a similarity transform, and scored using the **Chamfer Distance** (capturing mean geometric fidelity) and **Hausdorff Distance** (capturing worst-case deviation). This procedure quantifies functional equivalence directly in object space, rewarding models that recover the correct governing law regardless of symbolic form or parameterization (Fan et al., 2016).

**(5) Diversity and coverage:** SurfaceBench spans 15 scientifically grounded categories and 183 validated equations across explicit, implicit, and parametric formulations. The benchmark covers domains ranging from wave optics and material deformation to energy minimization and procedural geometry, combining real-world equations with controlled synthetic variants. This diversity ensures broad coverage of symbolic operators, compositional patterns, and topological structures, providing a rigorous and representative testbed for assessing symbolic reasoning across scientific domains.

## 2.2 Benchmark Construction Pipeline



Dataset Construction

Seed Selection

Domain Selection

Identify key scientific domains.
- Optics
- Fluid dynamics
- Electromagnetics

Equation Selection

Select Equations from that domain.
- Phase-shifted diffusion
- Angular amplitude decay
- Nonlinear acoustic interference

Operator Augmentation

Operator Distribution

Estimate the distribution of symbolic operators.
- Co-occurance of operators
- Combination of operators
- Nesting order of operators

Compositional Augmentation

To mitigate memorization of canonical forms, apply symbolic perturbations such as:
- Functional nesting,
- Operator substitution

Solvability Check

Novelty Check

Each generated equation, verify
- Analytic solvability
- Numerical stability
- Novelty
- Discard redundant or trivial forms.

Stability Check

Candidate surfaces are evaluated under
- Boundary conditions
- Edge conditions
- Ensure continuity and differentiability

Data Collection

Data Generation

For each validated surface, uniformly sample points across a bounded 3D domain to produce reference datasets using adaptive density in regions of high curvature.

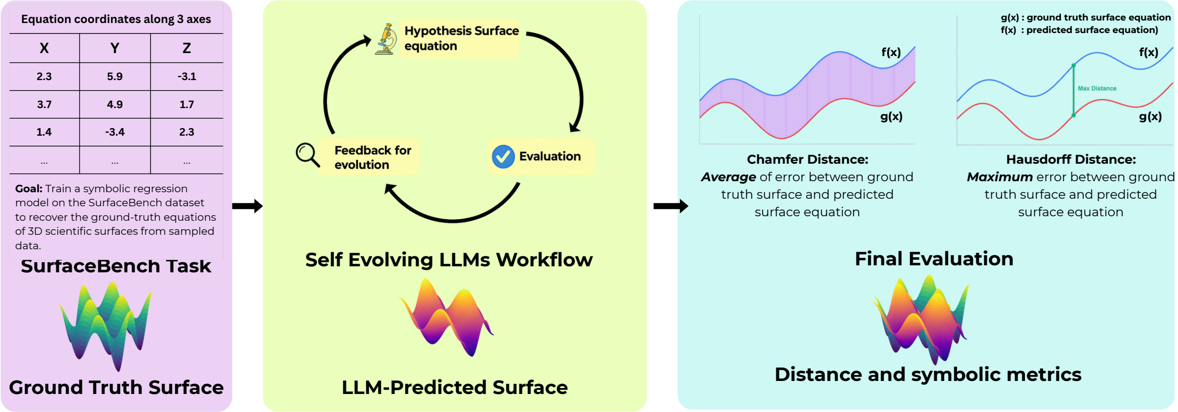
Expert Validation

All generated surfaces are visually inspected by domain experts to confirm smoothness, completeness, and physical plausibility, removing degenerate or unstable cases.

Figure 2: Dataset curation pipeline for SurfaceBench, ensuring a diverse set of seed equations, their transformation to discourage memorization, and rigorous validation through novelty and solvability checks.

The construction of SurfaceBench visualized in figure 2 follows a structured, multi-phase pipeline designed to ensure scientific grounding, analytic diversity, and robustness against memorization. **(1) Domain Selection:** We first identify key scientific domains (such as optics, fluid dynamics, electromagnetics, materials science, and robotics) that naturally give rise to continuous 3D surfaces governed by analytic equations. **(2) Equation Selection:** Within each domain, representative problems are selected and their governing equations are obtained in explicit, implicit, or parametric form, providing physically interpretable seeds for transformation. **(3) Operator Distribution:** The collected corpus is analyzed to estimate the empirical distribution of symbolic operators (e.g., trigonometric, exponential, rational, polynomial), ensuring that later augmentations reflect realistic symbolic usage patterns observed in scientific laws. **(4) Compositional Augmentation:** To mitigate memorization of canonical forms, we apply controlled symbolic perturbations inspired by LLM-SRBench (Shojaee et al., 2025b), including functional nesting (e.g., $\sin(x) \to \sin(x^2 + y^2)$), additive and multiplicative term blending, coordinate reparameterization (affine, polar, or spherical substitutions), and operator substitution (e.g., replacing $\sin(x)$ with $\tanh(x)$ or $(1 - e^{-x})$). These augmentations produce non-canonical yet analytically solvable variants that maintain interpretability while forcing models to reason compositionally rather than retrieve memorized templates. **(5) Novelty Check:** Each generated equation is symbolically simplified and verified for analytic solvability, numerical stability, and novelty relative to prior benchmarks (Feynman (Udrescu & Tegmark, 2020), SRBench (La Cava et al., 2021)), discarding redundant or trivial forms. **(6) Stability Check:** Candidate surfaces are evaluated under boundary and edge conditions (e.g., $x, y = 0$ or asymptotic limits) to ensure continuity, differentiability, and bounded evaluation domains. **(7) Data Generation:** For each validated surface, we uniformly sample points across a bounded 3D domain to produce reference datasets for explicit, implicit, and parametric formulations, using adaptive sampling density in regions of high curvature. **(8) Expert Validation:** Finally, all surfaces are visually inspected by domain experts to confirm smoothness, completeness, and physical plausibility, removing degenerate or unstable cases. This pipeline yields a benchmark of 183 rigorously validated surfaces with 3 different representations of explicit, implicit, and parametric equations, spanning 15 scientific categories.

## 3 Experimental Setup



Equation coordinates along 3 axes

| X | Y | Z |
|---|---|---|
| 2.3 | 5.9 | -3.1 |
| 3.7 | 4.9 | 1.7 |
| 1.4 | -3.4 | 2.3 |
| ... | ... | ... |

**Goal:** Train a symbolic regression model on the SurfaceBench dataset to recover the ground-truth equations of 3D scientific surfaces from sampled data.

SurfaceBench Task
Ground Truth Surface
Hypothesis Surface equation
Evaluation
Feedback for evolution
Self Evolving LLMs Workflow
LLM-Predicted Surface
f(x)
g(x)
Chamfer Distance:
Average of error between ground truth surface and predicted surface equation
g(x) : ground truth surface equation
f(x) : predicted surface equation)
Max Distance
Hausdorff Distance:
Maximum error between ground truth surface and predicted surface equation
Final Evaluation
Distance and symbolic metrics

Figure 3: The SurfaceBench evaluation pipeline integrates symbolic and geometric metrics to assess equation recovery quality. Given sampled 3D surface data, self-evolving LLM frameworks generate candidate symbolic expressions. These predictions are compared against the ground truth using three complementary evaluation modes: regression-style errors (NMSE), symbolic accuracy (via equivalence checks), and geometry-aware distance metrics, namely Chamfer and Hausdorff distances.

### 3.1 Benchmark Methods

We evaluate SurfaceBench using a representative suite of symbolic regression frameworks that capture both classical and LLM-driven approaches. Together, these baselines span evolutionary search, neural guidance, and LLM–based symbolic discovery. The overall pipeline is depicted in figure 3.

#### 3.1.1 LLM-Based Equation Discovery Methods

**LLM-SR (Shojaee et al., 2025a)** — A program-search framework that expresses candidate equations as Python function templates. It fuses the scientific prior knowledge of large language models with a multi-island evolutionary search, using data-driven feedback to refine hypotheses.

**LaSR (Grayeli et al., 2024)** — A concept-learning approach that abstracts high-level semantic descriptions of mathematical relations from previously successful equations. These concepts guide a hybrid discovery process that blends LLM-assisted search with evolutionary optimization implemented through PySR.

**SGA (Ma et al., 2024)** — A bilevel optimization method that alternates between symbolic structures generated by LLM through discrete search and parameter fitting via PyTorch-based numerical optimization, thus coupling symbolic reasoning with continuous simulation.

**OpenEvolve (Sharma, 2025)** — An open-source package implementation for the alphaevolve framework (Novikov et al., 2025). It is a general purpose LLM–guided evolutionary framework that uses LLMs to propose transformation rules, mutation operators, or symbolic search heuristics, while fitness evaluation and selection are performed externally for the given tasks. This design decouples reasoning from optimization, allowing the LLM to serve as a flexible rule generator that evolves hypotheses efficiently.

#### 3.1.2 Non-LLM-Based Equation Discovery Methods

**TPSR (Shojaee et al., 2023)** — A Transformer-driven symbolic regression model that integrates Monte Carlo Tree Search (MCTS) as a decoding strategy. By incorporating feedback and caching, it accelerates equation generation during inference.

**NeSymReS (Biggio et al., 2021)** — A neural symbolic regression model pre-trained on large synthetic corpora of equations. It maps expressions to latent embeddings and reconstructs symbolic skeletons, followed by gradient-based fitting of numerical constants.

**E2E (Kamienny et al., 2022)** — An end-to-end Transformer trained to output entire equations directly, without intermediate skeletons. Constant values are refined post-prediction using the BFGS optimization algorithm, and scalable inference is achieved through generative sampling.

**DSR (Petersen et al., 2019)** — A reinforcement-learning–based symbolic regression framework that directly optimizes expression trees, balancing exploration and parsimony via accuracy-driven reward functions.

**uDSR (Landajuela et al., 2022)** — An extension of DSR that performs variable reduction and adds linear tokens for polynomial construction. It employs large-scale pretraining while remaining rooted in genetic programming principles.

**PySR (Cranmer, 2023)** — A genetic programming engine for symbolic regression that automatically tunes hyperparameters and enforces dimensional consistency by penalizing expressions violating unit constraints.

**gplearn (Stephens, 2016)** — A scikit-learn–compatible genetic programming library using a fit/predict interface, enabling integration with existing ML workflows and hyperparameter tuning pipelines.

### 3.2 Evaluation Metrics

Evaluating symbolic regression for surface recovery poses unique challenges due to the vast hypothesis space and the existence of multiple algebraically distinct expressions that describe identical physical or geometric behaviors. Prior equation discovery benchmarks typically rely on scalar regression metrics (e.g., NMSE or symbolic string comparisons between candidate and ground-truth formulas. Such approaches are insufficient for surfaces, where functional equivalence is determined by geometric correspondence rather than textual similarity. To address this, SurfaceBench introduces a domain-specific evaluation suite that integrates *geometry-aware distances*, *symbolic equivalence checks*, and *scale-invariant regression error*.

Owing to the nature of surfaces, multiple algebraically distinct expressions can produce the same three dimensional manifold. These can be due to representation differences or localized similarity of different equations. Conventional metrics such as normalized MSE or string based similarity would incorrectly penalize such cases. To capture equivalence in *functional* rather than *symbolic* space, SurfaceBench evaluates candidate and ground-truth surfaces directly in object space. Candidate and reference surfaces are uniformly sampled into dense point clouds and aligned to remove translation, rotation, and scale differences. We adopt two standard object-space distances that together capture global and local geometric fidelity:

$$\mathrm{Chamfer}(P,Q) = \frac{1}{|P|}\sum_{p\in P}\min_{q\in Q}\|p-q\|_2^2 \;+\; \frac{1}{|Q|}\sum_{q\in Q}\min_{p\in P}\|q-p\|_2^2, \tag{1}$$

$$\mathrm{Hausdorff}(P,Q) = \max\Big\{\sup_{p\in P}\min_{q\in Q}\|p-q\|_2,\;\sup_{q\in Q}\min_{p\in P}\|q-p\|_2\Big\}. \tag{2}$$

Chamfer Distance measures the *mean geometric fidelity* between two surfaces, emphasizing smooth, global deviations and gradual warping across the shape. It provides a stable, differentiable indicator of overall shape fidelity and is robust to minor sampling noise or misalignment. Hausdorff Distance captures the *worst-case deviation*, making it sensitive to sharp discontinuities, holes, or missing components. It is characteristically less smooth and highlights rare but critical geometric failures that Chamfer distance may overlook. Together, they characterize both distributed and localized geometric errors—revealing whether discrepancies stem from global distortion or isolated structural mismatches. Their joint use ensures both global accuracy and local structural integrity in evaluating equation discovery models for surfaces.

**Symbolic Accuracy.** Following LLM-SRBench (Shojaee et al., 2025b), we measure *Symbolic Accuracy* using an LLM-based equivalence check that incorporates algebraic simplifications and parameter rescalings. This provides a principled yet flexible way to judge whether the recovered equation is symbolically equivalent to the ground truth.

**Normalized Mean Squared Error (NMSE).** To maintain comparability with scalar-function benchmarks, we include NMSE as a regression-style measure of pointwise fit:

$$\mathrm{NMSE} = \frac{\sum_{i=1}^{N_{\text{test}}} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{N_{\text{test}}} (y_i - \bar{y})^2}.$$

## 4 Results

| | Explicit | | | | Implicit | | | |
|---|---|---|---|---|---|---|---|---|
| **Base LLM** | **SA ↑** | **NMSE ↓** | **Chamfer ↓** | **Hausdorff ↓** | **SA ↑** | **NMSE ↓** | **Chamfer ↓** | **Hausdorff ↓** |
| **SGA** | | | | | | | | |
| GPT4o-mini | 0.20 | 2.86 | 8.26 | 16.53 | 0.06 | 1.38 | 2.96 | 6.72 |
| Llama-3.1-8B | 0.10 | 3.73 | 9.82 | 18.48 | 0.05 | 1.43 | 3.01 | 7.81 |
| Qwen-3-8b | 0.10 | 4.29 | 5.19 | 13.25 | 0.05 | 1.57 | 3.05 | 8.26 |
| **LaSR** | | | | | | | | |
| GPT4o-mini | 0.35 | 2.87 | 4.30 | 11.00 | 0.06 | 3.48 | 5.04 | 10.07 |
| Llama-3.1-8B | 0.30 | 3.21 | 3.68 | 14.21 | 0.10 | 2.81 | 4.67 | 9.78 |
| Qwen-3-8b | 0.30 | 2.96 | 4.18 | 12.84 | 0.06 | 3.08 | 4.92 | 10.06 |
| **LLM-SR** | | | | | | | | |
| GPT4o-mini | 0.30 | 2.57 | 7.08 | 24.17 | 0.10 | 2.54 | 2.20 | 5.25 |
| Llama-3.1-8B | 0.20 | 2.62 | 7.44 | 29.29 | 0.13 | 2.74 | 3.01 | 9.05 |
| Qwen-3-8b | 0.25 | 2.38 | 6.99 | 28.83 | 0.02 | 2.61 | 1.51 | 10.6 |
| **OpenEvolve** | | | | | | | | |
| GPT4o-mini | **0.50** | 0.98 | 2.69 | 4.88 | **0.12** | 0.71 | 1.85 | **4.96** |
| Llama-3.1-8B | 0.40 | 0.99 | 3.17 | 5.08 | 0.02 | 0.99 | 2.96 | 5.02 |
| Qwen-3-8b | 0.40 | 1.25 | 3.23 | 5.82 | 0.04 | 0.92 | 2.35 | 5.92 |
| **Non-LLM Baselines** | | | | | | | | |
| NeSymReS | 0.20 | 0.59 | 0.84 | 1.23 | 0.05 | 0.69 | 2.45 | 5.95 |
| gplearn | 0.15 | 0.38 | 0.61 | 1.09 | 0.05 | 0.82 | 1.83 | 5.13 |
| E2E | 0.30 | 0.28 | 0.34 | 0.88 | 0.05 | 0.69 | 1.83 | 5.70 |
| DSR | 0.15 | 0.32 | 0.24 | 0.87 | 0.05 | 0.64 | 1.74 | 6.06 |
| uDSR | 0.25 | 0.25 | 0.30 | 0.88 | 0.10 | **0.61** | 1.79 | 5.72 |
| TPSR | 0.25 | 0.21 | 0.34 | 0.85 | 0.05 | 0.66 | 2.52 | 6.88 |
| PySR | 0.25 | **0.18** | **0.13** | **0.41** | 0.05 | 0.60 | **1.64** | 5.53 |

Table 1: Comparison of various symbolic regression methods on SurfaceBench. Performance is reported across explicit and implicit forms using Symbolic Accuracy (SA), Normalized Mean Squared Error (NMSE), Chamfer distance, and Hausdorff distance. Higher is better for SA and lower is better for the remaining metrics.

Table 1 reports results over the surfaces by representation: explicit and implicit, with four metrics: Symbolic Accuracy, nMSE, Chamfer, and Hausdorff. We observe that explicit surfaces yield the highest Symbolic Accuracy while implicit surfaces achieve the lowest geometric distances. Notably, the explicit results reveal that models often recover the correct structural family but fail to produce geometrically tight equations. We attribute this to symbolic structures pointing in the right direction, but the equation fitting stage being under-optimized. As a result, Chamfer and Hausdorff distances remain high despite strong Symbolic Accuracy. This exposes a pipeline gap: after structure discovery, a targeted geometric calibration step is needed to refine parameters such as scale and shift, ensuring that structural discovery translates into gains in geometry-based metrics.

| Model | SA ↑ | NMSE ↓ | Chamfer ↓ | Hausdorff ↓ |
|---|---|---|---|---|
| **OpenEvolve** | | | | |
| GPT4o-mini | 0.07 | 0.71 | **1.85** | **4.96** |
| Llama-3.1-8B | 0.02 | 0.99 | 2.96 | 5.02 |
| Qwen-3-8B | 0.04 | 0.92 | 2.35 | 5.92 |
| **Non-LLM Baseline** | | | | |
| PySR | **0.10** | **0.61** | 2.52 | 5.53 |

Table 2: Evaluation of symbolic regression methods on parametric surface equations.

Secondly, the results for the implicit category show the opposite pattern. Distance-driven search brings the discovered surface equations closer to the ground truth even when the algebraic form is not exact. This yields strong Chamfer and Hausdorff performance despite lower Symbolic Accuracy. Together, these findings highlight the tension and complementarity between geometric proximity and algebraic fidelity.

Table 2 reports the results of these methods on parametric surfaces. Parametric equations remain one of the most underexplored representation types in symbolic regression. Among the methods we benchmark, only Openevolve and PySR reliably handle multiple equations, i.e., multi-output regression in a single pass. This exposes a major gap in current symbolic discovery methods that very few methods are designed to jointly learn a coupled set of equations, which is essential for modeling parametric surfaces. Finally, the results for parametric surfaces show consistently strong performance across all metrics for both non-LLM and LLM based OpenEvolve framework.

## 5 Ablations

We conduct a comprehensive ablation to understand the robustness of these methods. As noted previously, the LaSR, SGA, and LLM-SR methods lack algorithmic support for parametric equation discovery, and thus we exclude the parametric category from the ablation experiments.

### 5.1 Noise sensitivity

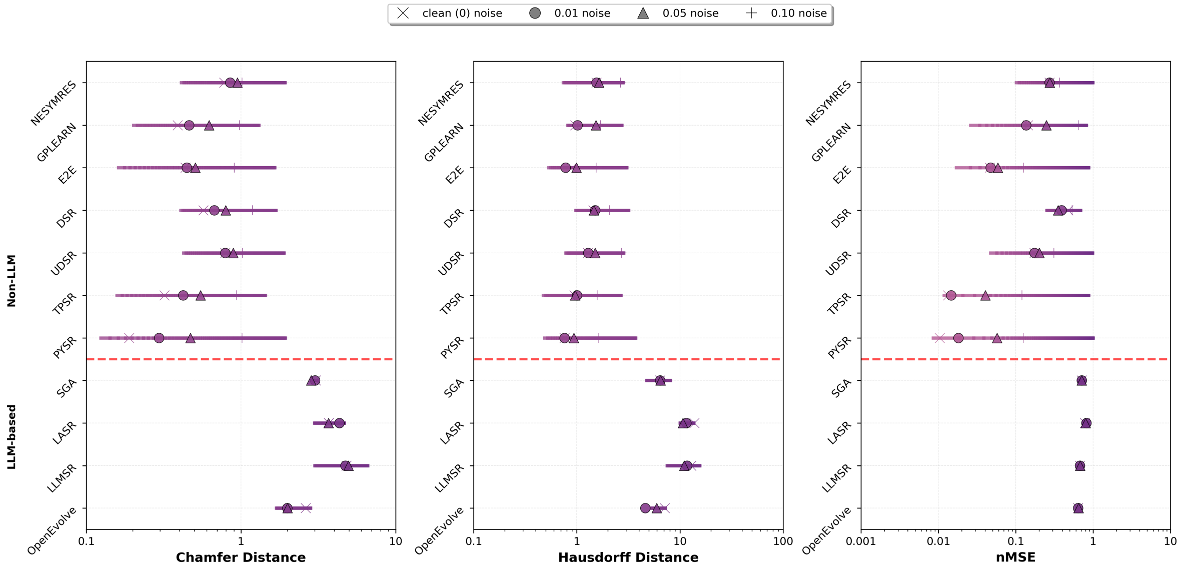
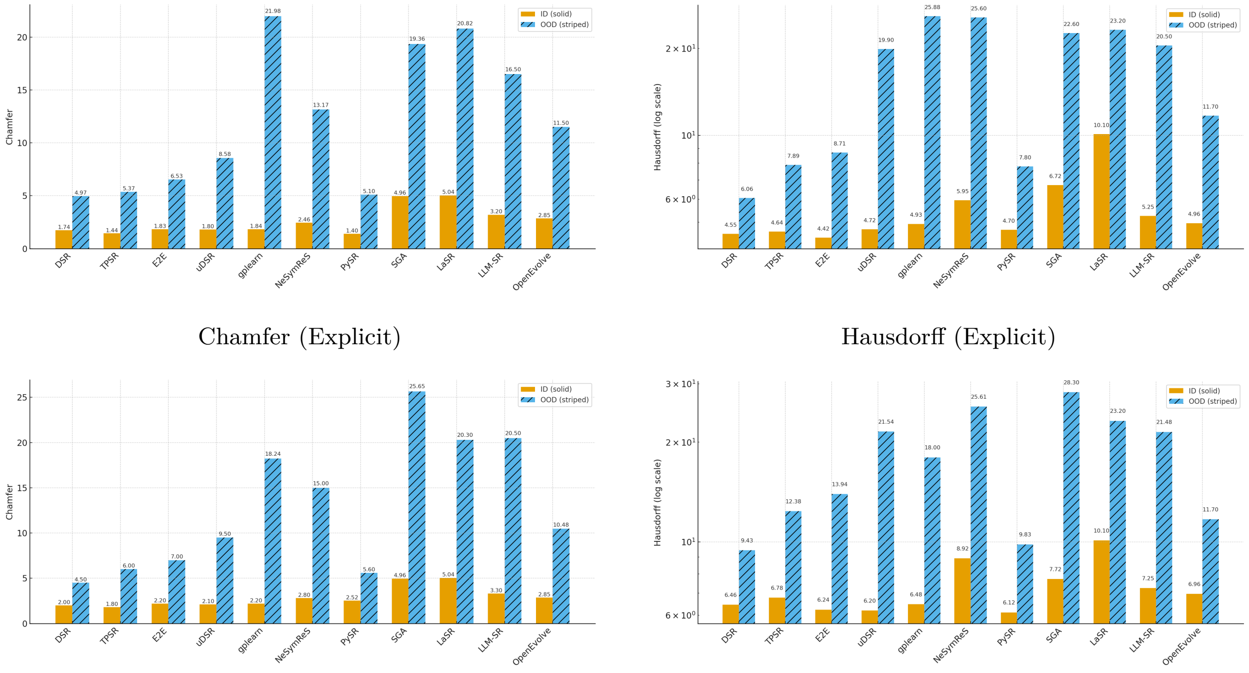


Figure 4: Noise sensitivity analysis across Chamfer Distance, Hausdorff Distance, and nMSE. Lower values indicate better performance.

To evaluate the robustness of our symbolic regression models under realistic data perturbations, we conducted a comprehensive noise sensitivity analysis across two state-of-the-art language models: GPT-4o-mini and

LLaMA-3.1-8B. The experiment was designed to systematically investigate how model performance degrades under varying levels of data corruption, simulating real-world scenarios where training data may contain measurement errors, sensor noise, or other sources of uncertainty. We selected 13 representative equations spanning diverse mathematical domains including nonlinear dynamical systems, quantum-inspired surfaces, stochastic processes, and hybrid multi-modal symbolic surfaces, ensuring broad coverage of the symbolic regression problem space. The experimental design employed a factorial approach with three noise levels (1%, 5%, and 10% Gaussian noise). Performance was evaluated using both Chamfer distance and Hausdorff distance metrics on in-domain test data, providing complementary measures of geometric fidelity between predicted and ground truth.

## 5.2 Out of Domain performance



Chamfer (Explicit)

Hausdorff (Explicit)

Chamfer (Implicit)

Hausdorff (Implicit)

Figure 5: Chamfer and Hausdorff distance metrics for both in-domain (ID) and out-of-domain (OOD) generalization across explicit and implicit forms.

We define out-of-domain strictly as a range shift in the input grid. If a model is trained on inputs sampled from [-5,5] along each axis, OOD tests use the non-overlapping exterior bands $[-10, -5] \cup [5, 10]$. This isolates extrapolation from interpolation: models must generalize learned structure beyond training support, not merely reproduce local behavior. Evaluation is conducted directly in object space, using both symbolic and geometric metrics, ensuring that performance gains stem from genuine extrapolative competence rather than token-level memorization.

## 5.3 Impact of Domain-Prior Prompts

To assess the utility of domain knowledge, we conduct an ablation by inserting domain priors in prompts serving as lightweight cues describing coordinate systems such as spherical or cylindrical charts, conservation or symmetry properties, knowledge about the scientific field of the equation such as optics, electromagnetism, etc. These prompts differ from generic discovery prompts by encoding partial structural knowledge rather than purely data-driven guidance. Although such priors can, in principle, reduce search-space ambiguity, they are not typically available in real-world experimental settings. Moreover, providing incorrect or mismatched

| Method | Explicit | | | | | | Implicit | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Chamfer | | | Hausdorff | | | Chamfer | | | Hausdorff | | |
| | w/o priors | w/ priors | Δ | w/o priors | w/ priors | Δ | w/o priors | w/ priors | Δ | w/o priors | w/ priors | Δ |
| SGA | 7.76 | 6.22 | +1.54 | 16.09 | 12.22 | +3.87 | 3.01 | 2.76 | +0.25 | 7.60 | 6.26 | +1.34 |
| LaSR | 4.05 | 4.04 | +0.01 | 12.68 | 10.04 | +2.64 | 4.88 | 4.24 | +0.64 | 9.97 | 9.24 | +0.73 |
| LLM-SR | 7.17 | 5.77 | +1.40 | 27.43 | 20.77 | +6.66 | 2.24 | 2.07 | +0.17 | 8.30 | 5.07 | +3.23 |
| OpenEvolve | 2.93 | 1.16 | +1.77 | 4.98 | 4.16 | +0.82 | 2.41 | 1.82 | +0.59 | 4.99 | 4.82 | +0.17 |

Table 3: Effect of priors measured via Chamfer and Hausdorff distances (↓ better). $\Delta$ = without priors − with priors. Positive $\Delta$ indicates improvement.

priors often leads to severe degradation in reconstruction quality, showing how overly constraining prompts can hinder rather than help discovery. Table 3 reports the results for this study.

Empirically, we observe that even when correct priors are provided, LLM-based symbolic regression methods show only marginal improvement over their baseline performance and remain inferior to non-LLM methods. This suggests that, despite a narrowed functional space, current LLM-based approaches do not yet reliably exploit structural cues during optimization. These findings motivate a deeper failure analysis, presented in the subsequent section, to identify where such methods falter in integrating structural and geometric cues.

## 5.4 Failure Analysis

Figure 6 summarizes our failure analysis, which investigates why LLM-based symbolic discovery methods underperform relative to traditional approaches. Symbolic regression requires (1) discrete structural search over symbolic expressions and (2) continuous optimization of numerical parameters. These are tightly coupled subproblems that LLM-based models attempt to solve within a single generative process, which often makes it difficult to isolate specific failure modes. To disentangle these effects, we classify errors as *search failures* and *equation fitting failures*. (i) A search failure occurs when the discovered equation includes terms from the incorrect functional family (for example, polynomials instead of trigonometric terms), indicating a breakdown in symbolic term retrieval. (ii) An equation-fitting failure, in contrast, arises when the model correctly identifies the relevant symbolic families but fails to assemble them in the correct structural order or infer accurate constants, indicating a lack of optimization quality. We evaluated LLM-SR and OpenEvolve on ten representative equations containing trigonometric or exponential terms to isolate these behaviors. We observe that LLM-based methods show strong category retrieval, narrowing the hypothesis space to relevant symbolic families, but struggle to efficiently traverse the vast search space. Non-LLM based methods perform thousands of evaluation and mutation cycles within minutes, guided by explicit loss-driven feedback. We attribute this failure to LLM-based approaches being constrained by autoregressive generation as they exhibit substantial computational latency and stagnate once optimization of discovered equations is required. We also observe that LLM-based methods succeed primarily when they identify the correct functional categories early in the search process. After ∼ 200 iterations, it is rare for these models to recover or move meaningfully toward the correct equation, suggesting limited corrective or self-refinement capability once an initial hypothesis diverges from the target function. In complete failure cases, the issue originates earlier, where the models fail to identify correct functional primitives at all, leading to collapse at the symbolic search stage rather than during optimization. While LLMs offer early efficiency through linguistic priors, their

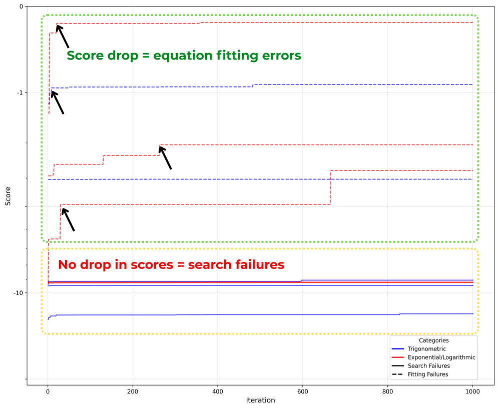


Figure 6: Failure modes of two LLM based symbolic regression methods: LLM-SR and OpenEvolve. We identify two modes of errors: (i) search space errors, where the frameworks make severe errors to find the correct functional families and (ii) equation fitting errors, where frameworks are unable to optimize the equation comprising of correct functional families.

lack of iterative optimization and feedback loops prevents robust convergence, especially under multi-output constraints or complex compositional structures. Motivated by these observations we envision future LLM based symbolic regression methods to improve their equation fitting or optimization approaches to overcome majority of errors brought forth by SurfaceBench.

## 6 Related Work

### 6.1 Symbolic Regression

Early symbolic regression (SR) research was predominantly centered around genetic programming (GP) which evolved tree-structured equations using crossover and mutation, enabling discovery of interpretable analytic forms directly from data (Koza, 1992). Later variants of GP improved efficiency through regularization strategies and lexicographic tournament selection (Schmidt & Lipson, 2009; Vladislavleva et al., 2009; Agapitos et al., 2012), while others incorporated dimensional analysis constraints (Udrescu & Tegmark, 2020). Despite success on low-dimensional problems, classical GP methods suffer from exponential search complexity and limited scalability in the presence of noise or higher-order dynamics (McConaghy, 2011; Cava et al., 2019). In physics-guided SR, partial differential and variational approaches discover governing equations by enforcing physical constraints such as smoothness or conservation; examples include SINDy (Brunton et al., 2016) and PDE-Net (Long et al., 2018), which identify sparse terms from candidate operators. Differentiable and neural SR frameworks such as DSR (Petersen et al., 2019) and NeSymReS (Biggio et al., 2021) combine neural function approximators with symbolic optimization, blending gradient-based fitting with discrete equation search to improve data efficiency. Recently, large language models (LLMs) have been incorporated into SR pipelines, with approaches such as LLM-SR (Shojaee et al., 2025a), LaSR (Grayeli et al., 2024), and OpenEvolve (Sharma, 2025). These methods use LLMs to generate symbolic hypotheses, refine them via tool-assisted feedback, and apply self-reflective reasoning loops. While they exploit pretrained scientific priors to propose meaningful candidate expressions, these methods continue to struggle with multi-equation coupling, implicit/parametric surfaces, and topology-aware reconstruction (Kazhdan et al., 2006).

### 6.2 Benchmarking Symbolic Regression

Early benchmarks such as AI Feynman (Udrescu & Tegmark, 2020), SSRD (Matsubara et al., 2024) and SRBench (La Cava et al., 2021) have been widely used to evaluate symbolic regression methods. However, these benchmarks are vulnerable to memorization, where pretrained LLMs often reproduce canonical formulas directly rather than reasoning from data (Shojaee et al., 2025b). To address this, LLM-SRBench (Shojaee et al., 2025b) introduced a large-scale set of synthetic equations designed to break memorization. While effective at increasing difficulty, these synthetic tasks are loosely connected to real scientific domains, and thus lack external validity. Additionally, these benchmarks tend to be narrow in scope, with many instances being structural variants of the same core formula.

### 6.3 Surface Equation Discovery

The task of recovering closed-form equations that define 3D surfaces has a long history across computer graphics, geometry processing, and learning-based methods. Early work focused on fitting smooth implicit functions to point clouds using radial basis functions (Carr et al., 2001) or solved Poisson-type PDEs to generate indicator fields aligned with input normals (Kazhdan et al., 2006; Kazhdan & Hoppe, 2013). These methods yield robust reconstructions, but the resulting equations are either constrained to a predefined basis or lack interpretability. Neural implicit models such as DeepSDF (Park et al., 2019) and Occupancy Networks (Mescheder et al., 2019) generalize this idea by learning continuous fields whose level sets define geometry, achieving high fidelity but producing black-box representations. Hybrid methods like AtlasNet (Groueix et al., 2018) learn parametric surface patches, offering some structural insight but still lacking symbolic abstraction.

## 7 Conclusion

We introduce SurfaceBench, the first comprehensive benchmark for LLM-driven *symbolic discovery of 3D surfaces*, encompassing *183* tasks across *15* categories and three representation types: *explicit*, *implicit*, and *parametric*. SurfaceBench provides a standardized, geometry-aware evaluation suite that combines symbolic fidelity with spatial accuracy, through metrics such as Chamfer distance and Hausdorff distances. Unlike prior symbolic regression datasets that target 1D or low-dimensional expressions, SurfaceBench elevates the challenge to surface-level reasoning, where geometric structure must be inferred from data alone. It captures a broad diversity of coordinate systems, compositional complexity, and transformation types, offering a rigorous testbed for both algebraic and geometric generalization. Overall, success on SurfaceBench demands models that can reason over multi-output coupling and latent coordinate systems, resolve symbolic non-uniqueness, and generalize across a wide range of scientific domains and representation types. Extensive experiments with state-of-the-art discovery frameworks and multiple LLM backbones reveal that, despite occasional successes on specific families, no existing approach consistently excels across all representation types or evaluation metrics. Performance remains far from saturation, underscoring the need for new advances in symbolic reasoning, geometric calibration, multi-equation coupling, and topology-aware discovery. By releasing both the curated dataset and accompanying evaluation pipeline, we hope to establish SurfaceBench as a community standard for benchmarking symbolic surface discovery. We envision that the SurfaceBench dataset and evaluation framework will catalyze progress in automated surface equation recovery, foster cross-pollination between geometry and symbolic regression, and deepen our understanding of LLMs' compositional and scientific reasoning capabilities in high-dimensional domains.

## References

Alexandros Agapitos, Anthony Brabazon, and Michael O'Neill. Controlling overfitting in symbolic regression based on a bias/variance error decomposition. In Carlos A. Coello Coello, Vincenzo Cutello, Kalyanmoy Deb, Stephanie Forrest, Giuseppe Nicosia, and Mario Pavone (eds.), *Parallel Problem Solving from Nature - PPSN XII*, pp. 438–447, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

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.

Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, March 2016. ISSN 1091-6490. doi: 10.1073/pnas.1517384113. URL `http://dx.doi.org/10.1073/pnas.1517384113`.

Jonathan C. Carr, Richard K. Beatson, Jon B. Cherrie, Tim J. Mitchell, W. Richard Fright, Bruce C. McCallum, and Tim R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of SIGGRAPH 2001*, pp. 67–76. ACM, 2001.

William La Cava, Tilak Raj Singh, James Taggart, Srinivas Suri, and Jason H. Moore. Learning concise representations for regression by evolving networks of trees, 2019. URL `https://arxiv.org/abs/1807.00981`.

Kathleen Champion, Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019. doi: 10.1073/pnas.1906995116. URL `https://www.pnas.org/doi/abs/10.1073/pnas.1906995116`.

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

Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3d object reconstruction from a single image, 2016. URL `https://arxiv.org/abs/1612.00603`.

Arya Grayeli, Atharva Sehgal, Omar Costilla Reyes, Miles Cranmer, and Swarat Chaudhuri. Symbolic regression with a learned concept library. *Advances in Neural Information Processing Systems*, 37:44678–44709, 2024.

Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 216–224, 2018.

Nan Jiang, Ziyi Wang, and Yexiang Xue. Egg-sr: Embedding symbolic equivalence into symbolic regression via equality graph. *arXiv preprint arXiv:2511.05849*, 2025.

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.

Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3), July 2013. ISSN 0730-0301. doi: 10.1145/2487228.2487237. URL https://doi.org/10.1145/2487228.2487237.

Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. SGP '06, pp. 61–70, Goslar, DEU, 2006. Eurographics Association. ISBN 3905673363.

John R. Koza. *Genetic programming: on the programming of computers by means of natural selection.* MIT Press, Cambridge, MA, USA, 1992. ISBN 0262111705.

William La Cava, Bogdan Burlacu, Marco Virgolin, Michael Kommenda, Patryk Orzechowski, Fabrício Olivetti de França, Ying Jin, and Jason H Moore. Contemporary symbolic regression methods and their relative performance. *Advances in neural information processing systems*, 2021(DB1):1, 2021.

Mikel Landajuela, Chak Lee, Jiachen Yang, Ruben Glatt, Claudio P. Santiago, Ignacio Aravena, Terrell N. Mundhenk, Garrett Mulcahy, and Brenden K. Petersen. A unified framework for deep symbolic regression. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=2FNnBhwJsHK.

Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data, 2018. URL https://arxiv.org/abs/1710.09668.

Pingchuan Ma, Tsun-Hsuan Wang, Minghao Guo, Zhiqing Sun, Joshua B. Tenenbaum, Daniela Rus, Chuang Gan, and Wojciech Matusik. LLM and simulation as bilevel optimizers: A new paradigm to advance physical scientific discovery. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=hz8cFsdz7P.

Yoshitomo Matsubara, Naoya Chiba, Ryo Igarashi, and Yoshitaka Ushiku. Rethinking symbolic regression datasets and benchmarks for scientific discovery, 2024. URL https://arxiv.org/abs/2206.10540.

Trent McConaghy. *FFX: Fast, Scalable, Deterministic Symbolic Regression Technology*, pp. 235–260. Springer New York, New York, NY, 2011. ISBN 978-1-4614-1770-5. doi: 10.1007/978-1-4614-1770-5_13. URL https://doi.org/10.1007/978-1-4614-1770-5_13.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Jose M. Moyano, Oscar Gabriel Reyes Pupo, Habib M. Fardoun, and Sebastián Ventura. Performing multi-target regression via gene expression programming-based ensemble models. *Neurocomputing*, 432:275–287, 2021. doi: 10.1016/J.NEUCOM.2020.12.060. URL https://doi.org/10.1016/j.neucom.2020.12.060.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. Alphaevolve: A coding agent for scientific and algorithmic discovery, 2025. URL `https://arxiv.org/abs/2506.13131`.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Brenden K Petersen, Mikel Landajuela, T Nathan Mundhenk, Claudio P Santiago, Soo K Kim, and Joanne T Kim. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. *arXiv preprint arXiv:1912.04871*, 2019.

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`.

Asankhaya Sharma. Openevolve: an open-source evolutionary coding agent, 2025. URL `https://github.com/codelion/openevolve`.

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.

Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K. Reddy. LLM-SR: Scientific equation discovery via programming with large language models. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL `https://openreview.net/forum?id=m2nmp8P5in`.

Parshin Shojaee, Ngoc-Hieu Nguyen, Kazem Meidani, Amir Barati Farimani, Khoa D Doan, and Chandan K. Reddy. LLM-SRBench: A new benchmark for scientific equation discovery with large language models. In *Forty-second International Conference on Machine Learning*, 2025b. URL `https://openreview.net/forum?id=SyQPiZJVWY`.

Trevor Stephens. Genetic programming in python, with a scikit-learn inspired api: gplearn. `https://github.com/trevorstephens/gplearn`, 2016.

Silviu-Marian Udrescu and Max Tegmark. Ai feynman: a physics-inspired method for symbolic regression, 2020. URL `https://arxiv.org/abs/1905.11481`.

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. *Trans. Evol. Comp*, 13(2):333–349, April 2009. ISSN 1089-778X. doi: 10.1109/TEVC.2008.926486. URL `https://doi.org/10.1109/TEVC.2008.926486`.

# A Appendix

## A. Evaluation Details

### A.1. Symbolic Accuracy

In addition to geometry-based metrics, we also report **Symbolic Accuracy** to provide a complementary view of equation discovery performance. We adopt the evaluation methodology introduced in (Shojaee et al., 2025b), which leverages GPT-4o as an automated evaluator to assess the mathematical equivalence between predicted and ground-truth hypotheses.

Traditional exact-match metrics (e.g., recovery rate, tree edit distance) are insufficient in our setting, as many surface equations admit multiple algebraically equivalent representations. The LLM-based evaluator provides a more flexible and semantically meaningful assessment of symbolic equivalence, operating in diverse formats (strings, trees, and executable forms).

We follow the same preprocessing pipeline as (Shojaee et al., 2025b), including normalization of constants and removal of auxiliary information, and rely on GPT-4o's judgement of equivalence. This ensures comparability with prior symbolic regression benchmarks while complementing our primary focus on geometric fidelity.

As shown in Figure 7, the symbolic assessment provides a complementary view of equation discovery performance.

**Prompt for Symbolic Accuracy Evaluation**

**Question:** Given the ground truth mathematical expression **A** and the hypothesis **B**, determine if there exist any constant parameter values that would make the hypothesis equivalent to the given ground truth expression.

Let's think step by step. Explain your reasoning and then provide the final answer as:

- **(A):** `sqrt(q1/(E*epsilon))/(2*sqrt(pi))`
- **(B):** Hypothesis as Program

**LLM (GPT-4o) Judgement:**
**Reasoning:** "The expressions can match if params[0] * params[1] = 1 and params[2] = 1, as this aligns both the scalar and constant factors appropriately."
**Answer:** Yes

Figure 7: Prompt used for symbolic accuracy evaluation.

## B. Implementation Details

For a comprehensive evaluation, we implement three state-of-the-art LLM-guided scientific equation discovery baselines, each tested on the SurfaceBench datasets using three different LLM backbones: an open-source model (Llama-3.1-8B-Instruct), a closed-source model (GPT-4o-mini), and a proprietary model (Qwen-2.5-7B-Instruct).

### A.1 Parameters

Table 4 presents the key implementation details for each discovery agentic method. We adopt most of the hyperparameters from the original implementation for these methods. We have only changed some hyperparameters in different baselines that affect the number of LLM calls in the search framework. This is to make sure we have a fair comparison across baseline discovery frameworks with the same access budget to LLMs. In our experiments, all baseline frameworks have 1k calls to LLMs (per problem) through the discovery process and an equivalent number of calls to the Non-LLM method.

### A.2 Dataset Statistics

The table 5 details the split of each domain by category

Table 4: Implementation details of LLM-based scientific equation discovery methods.

| Method | Parameters |
|---|---|
| OpenEvolve | Temperature $\tau = 0.8$<br>5 equation program hypotheses sampled from LLM for initial prompt<br>No access to data for data-driven refinement<br>Time limit $T = 30$s per program hypothesis execution,<br>BFGS optimizer from Scipy for parameter optimization of equation skeletons |
| SGA | PyTorch-based implementation of model and `torch.nn.Module` class<br>Mean square error loss for data-driven feedback in agentic search<br>Adam optimizer in PyTorch for differential parameter optimization of equation skeletons |
| LaSR | Iterations = 25<br>Cycles per iteration = 550<br>Populations = 10<br>Population size = 33<br>Maximum size = 30<br>Operators: $+$, $*$, $-$, $/$, $\wedge$, exp, log, sqrt, sin, cos, tan, cosh<br>LLM weights: llm_mutate =0.005, llm_crossover =0.005, llm_gen_random =0.005<br>Top-$K$ = 20 concepts from library<br>Default configuration of PySR for parameter optimization |
| LLM-SR | Temperature $\tau = 0.8$<br>Batch size $b = 4$ equation programs per prompt<br>$e = 4$ parallel evaluators<br>Time limit $T = 30$s per program hypothesis,<br>Memory limit M = 2GB<br>$m = 10$ islands for population diversity through search<br>$k = 2$ in-context examples per prompt<br>Maximum 10 parameters per equation skeleton<br>BFGS optimizer from Scipy for parameter optimization of equation skeletons |

Table 5: **SurfaceBench Dataset Statistics.** The benchmark spans 183 scientifically grounded symbolic surfaces across 15 categories and 3 representation types. Each surface includes 5k training, 500 test, and 500 OOD samples, enabling evaluation across symbolic, geometric, and regression modalities.

| Scientific Category | Count | Highlights |
|---|---|---|
| Non-Canonical 3D Geometric Surfaces | 11 | Smooth analytic shapes; trigonometric + rational forms |
| Piecewise Regime Surfaces | 10 | Conditional logic and discontinuities |
| Symbolic–Numeric Composite Surfaces | 9 | Blends analytic and numeric components |
| Hybrid Multi-Modal Symbolic Surfaces | 9 | Mixed symbolic regimes and decision-based forms |
| Procedural & Fractal Surfaces | 11 | Oscillatory and recursive functions |
| Bio-Inspired Morphological Surfaces | 10 | Continuous biomorphic surfaces |
| Complex Composite Surfaces | 10 | Mixed polynomial–trigonometric compositions |
| Tensor Field Surfaces | 10 | Linear and nonlinear tensor projections |
| Discrete Symbolic Grid Surfaces | 10 | Indexed grid surfaces with modulo/floor operators |
| Non-Linear Dynamical System Surfaces | 10 | Dynamical mappings and non-linear coupling |
| Stochastic Process Surfaces | 10 | Analytic surfaces with stochastic behavior |
| Quantum Inspired Surfaces | 10 | Gaussian-enveloped oscillatory patterns |
| Surrogate-Distilled Symbolic Approximations | 9 | Symbolic surrogates distilled from neural models |
| Algebraic Manifolds of Higher Degree | 24 | High-order polynomial and transcendental implicit forms |
| Parametric / Transformed Coordinate Surfaces | 30 | Multi-output, latent-coordinate equations |

### A.3 SurfaceBench equations for each scientific categories

Table 6: Exact equations used in SurfaceBenchafter applying transformations. This table provides the equations by categories of scientific domains.

| Category | ID | Equations |
|---|---|---|
| Non-Canonical 3D Surfaces | NCGS1 | $\frac{\sin(x^2+y^2)}{1+x^2+y^2}$ |
| | NCGS2 | $\frac{x^2-y^2}{1+x^2+y^2}$ |
| | NCGS3 | $\text{atan2}(x,y)\ \exp\big(-(x^2+y^2)\big)$ |
| | NCGS4 | $\tanh\big(\sin(xy)\big)$ |
| | NCGS5 | $\log\big(1+x^2+y^2\big)\ \sin(x-y)$ |
| | NCGS6 | $\exp\big(\sin(x^2+y^2)\big)$ |
| | NCGS7 | $\frac{\cos(x^2+y)}{1+\lvert xy\rvert}$ |
| | NCGS8 | $\sinh(xy)\ \exp(-y^2)$ |
| | NCGS9 | $\frac{\sin\big(\sqrt{x^2+y^2}\big)}{\log(1+x^2)}$ |
| | NCGS10 | $x\ \exp\big(-x^2-y^2\big)\ \cos y$ |
| | NCGS11 | $\frac{\sin(xy)}{1+x^2+y^2}$ |
| Piecewise Regime Surfaces | PRS1 | $x^2$ if $x<y$ else $y^2$ |
| | PRS2 | $\sin(x)$ if $x<0$ else $\exp(y)$ |
| | PRS3 | $xy$ if $xy>0$ else $-xy$ |
| | PRS4 | $x^2+y^2$ if $x<y$ else $x^2-y^2$ |
| | PRS5 | $\cos(x)$ if $\lvert x\rvert<1$ else $\exp(-y^2)$ |
| | PRS6 | $x^3$ if $y>0$ else $-y^3$ |
| | PRS7 | $\lvert x-y\rvert+\sin(x)$ |
| | PRS8 | $\sin(x+y)$ if $x^2+y^2<1$ else $0$ |
| | PRS9 | $\tanh(x)$ if $x>y$ else $\cos(y)$ |
| | PRS10 | $xy$ if $\lvert x-y\rvert<0.5$ else $\sin(x-y)$ |
| Symbolic-Numeric Composite Surfaces | SNCS1 | $\sin(x)+\exp(-y^2)$ |
| | SNCS2 | $\tanh(xy)+x^2$ |
| | SNCS3 | $\exp(-x^2-y^2)+\cos(3x)$ |
| | SNCS4 | $\alpha\sin(\beta x)+\gamma\log(1+y^2)$ |

*Continued on next page*

*Table 6 - continued from previous page*

| **Category** | **Equation** | **Real-world Domain** |
|---|---|---|
| | SNCS5 | $\sinh(x) - \tanh(y)$ |
| | SNCS6 | $\sin(x^2 + y^2)\ \exp(-\sqrt{x^2 + y^2})$ |
| | SNCS7 | $\tanh(x)\ \log(1 + y^2)$ |
| | SNCS8 | $\cos(xy) + \exp(-x^2 + y)$ |
| | SNCS9 | $\beta\cos(x) + \gamma\sin(y^2)$ |
| Hybrid Multi-Modal Symbolic Surfaces | HMMSS1 | $x^2$ if $x < 0$ else $\sin(y)$ |
| | HMMSS2 | $\log\big(1 + \lvert x\rvert\big)$ if $y < 0$ else $\exp(-y^2)$ |
| | HMMSS3 | $x^2 + \sin(y)$ if $xy > 0$ else $\big(-x^2 - \cos y\big)$ |
| | HMMSS4 | $\tanh(x - y)$ if $x > y$ else $0$ |
| | HMMSS5 | $\lvert xy\rvert + \sin(x - y)$ |
| | HMMSS6 | $x^2$ if $y > 0$ else $\cos(y^2)$ |
| | HMMSS7 | $\sin(xy)$ if $x^2 + y^2 < 1$ else $\log(1 + x^2)$ |
| | HMMSS8 | $\tanh(x + y)$ if $xy < 0$ else $\sin(x - y)$ |
| | HMMSS9 | $x$ if $x > y$ else $y^2 + \sin(x)$ |
| Procedural & Fractal Surfaces | PFS1 | $\sin(5x)\cos(5y)$ |
| | PFS2 | $\cos(x^2y^2){+}0.2\ \sin\big(5\sqrt{\lvert x\rvert + \lvert y\rvert}\big)$ |
| | PFS3 | $\sin(xy) + 0.5\ \sin(3x + 5y)$ |
| | PFS4 | $e^{-0.1(x^2+y^2)}\ \sin(xy)$ |
| | PFS5 | $\sin(x^3 + y^3)$ |
| | PFS6 | $xy\ \cos\big(\sqrt{x^2 + y^2}\big)$ |
| | PFS7 | $\sin(2^x x)\ \cos(2^y y)$ |
| | PFS8 | $e^{-\lvert x-y\rvert}\ \sin\big(3(x + y)\big)$ |
| | PFS9 | $\sum_{i=1}^{4} \frac{\sin(2^i x)}{i}$ |
| | PFS10 | $\tanh(xy)\ \cos\big(\sqrt{x^2 + y^2}\big)$ |
| | PFS11 | $\sin(2x) + \alpha\exp(-y^2)$ |

*Continued on next page*

*Table 6 - continued from previous page*

| **Category** | **Equation** | **Real-world Domain** |
|---|---|---|
| Bio-Inspired Morphological Surfaces | BIMS1 | $\sin\big(\sqrt{x^2+y^2}\big)$ |
| | BIMS2 | $\exp(-x^2-y^2)\,\cos(3x)$ |
| | BIMS3 | $\tanh(x+y)\,\sin(xy)$ |
| | BIMS4 | $\log(1+x^2+y^2)\,\cos(xy)$ |
| | BIMS5 | $x^2+y^2-\sin(2x+2y)$ |
| | BIMS6 | $\cos(2x)\,\cos(2y)$ |
| | BIMS7 | $\frac{\sin(x^2+y^2)}{1+x^2+y^2}$ |
| | BIMS8 | $\tanh(x^2-y^2)$ |
| | BIMS9 | $\exp\big(-\lvert xy\rvert\big)\,\sin(x+y)$ |
| | BIMS10 | $\cos(xy)+0.1\,(x^2+y^2)$ |
| Complex Composite Surfaces | CSS1 | $\log\big(1+x^2+y^2\big)\,\cos(x-y)$ |
| | CSS2 | $\frac{\sin(x)+\cos(y)}{1+x^2+y^2}$ |
| | CSS3 | $e^{-0.1\lvert xy\rvert}\,\tanh(x+y)$ |
| | CSS4 | $\frac{x^2y-y^2}{1+x^2}$ |
| | CSS5 | $\sqrt{1+x^2+y^2}\,\sin(xy)$ |
| | CSS6 | $\frac{e^x+e^{-y}}{1+\lvert x-y\rvert}$ |
| | CSS7 | $\begin{cases} x^2+y^2, & \text{if } x+y<0, \\ \sin(x+y), & \text{if } x+y\geq 0 \end{cases}$ |
| | CSS8 | $\frac{\cos\big(\sqrt{x^2+y^2}\big)}{1+e^{-xy}}$ |
| | CSS9 | $\sinh(x^2-y^2)\,e^{-0.1(x+y)^2}$ |
| | CSS10 | $\arctan(xy)+0.2\,e^{-x^2-y^2}$ |
| Tensor Field Surfaces | TFS1 | $x^2+y^2$ |
| | TFS2 | $\sin(x)\,\cos(y)$ |
| | TFS3 | $\exp(-x^2-y^2)$ |
| | TFS4 | $x\,y$ |
| | TFS5 | $\tanh(x+y)$ |
| | TFS6 | $\cos(x^2+y^2)$ |

*Continued on next page*

*Table 6 - continued from previous page*

| **Category** | **Equation** | **Real-world Domain** |
|---|---|---|
| | TFS7 | $\log(1+x^2+y^2)$ |
| | TFS8 | $x^2-y^2$ |
| | TFS9 | $\sin(x\,y)$ |
| | TFS10 | $\exp\big(-\lvert x-y\rvert\big)$ |
| Discrete Symbolic Grid Surfaces | DSGS1 | $\sin(i)+\cos(j)$ |
| | DSGS2 | $(-1)^i\,(-1)^j$ |
| | DSGS3 | $\mathrm{mod}(i,3)+\mathrm{mod}(j,2)$ |
| | DSGS4 | $\left\lfloor\sqrt{i^2+j^2}\right\rfloor$ |
| | DSGS5 | $\sin(ij)+i-j$ |
| | DSGS6 | $\cos(i+j)$ |
| | DSGS7 | $\mathrm{mod}(i^2+j^2,\,5)$ |
| | DSGS8 | $\tanh(i-j)$ |
| | DSGS9 | $\lfloor\sin(i^2+j^2)\rfloor$ |
| | DSGS10 | $\mathrm{mod}(ij,\,4)$ |
| Non-Linear Dynamical System Surfaces | NLDSS1 | $\cosh\big(0.1(x-y)\big)-\cos\big(0.5(x+y)\big)$ |
| | NLDSS2 | $e^{-0.05(x^2+y^2)}\,(x^2-y)\cos(y)$ |
| | NLDSS3 | $\log(1+x^2)\sin(y)-\log(1+y^2)\cos(x)$ |
| | NLDSS4 | $\sqrt{1+0.1(x^2+y^2)}\,\sin\big(0.5(x-y)\big)$ |
| | NLDSS5 | $\tanh\big(0.2(x^2-y^2)\big)$ |
| | NLDSS6 | $0.3(xy)-0.2\sin(x+y)\,e^{-0.05(x^2+y^2)}$ |
| | NLDSS7 | $\dfrac{x^2\sin(y)}{1+0.2y^2}$ |
| | NLDSS8 | $\sinh(0.2x)\,e^{-0.1y^2}$ |
| | NLDSS9 | $\arctan(xy)-0.3\sin(x-y)$ |
| Stochastic Process Surfaces | SPS1 | $3\,e^{-0.05(x^2+y^2)}\cos(0.2xy)+0.1x$ |
| | SPS2 | $2.2\sin(0.3x+0.2y)\,\big(1-e^{-0.1x^2}\big)$ |

*Continued on next page*

*Table 6 - continued from previous page*

| **Category** | **Equation** | **Real-world Domain** |
|---|---|---|
| | SPS3 | $1.8\cos(0.4xy)\,e^{-0.1x^2} + 0.3y^2$ |
| | SPS4 | $2\sin(0.7x)\,e^{-0.05y^2} + 0.5xy$ |
| | SPS5 | $3\,(1 - e^{-0.15x^2})\cos(0.3y) + 0.2x$ |
| | SPS6 | $2.5\tanh(0.2xy) + 0.4\sin(0.5x + y)$ |
| | SPS7 | $1.5e^{-0.1(x^2+y^2)}\sin(0.6x) + 0.3y$ |
| | SPS8 | $4\cos(0.4x)\left(1 - e^{-0.05y^2}\right) + 0.1x^2$ |
| | SPS9 | $2x^2e^{-0.2\lvert y\rvert} + 1.5\sin(0.3xy)$ |
| | SPS10 | $3\sin(0.5x)\,e^{-0.1y^2} + 0.2xy\cos(y)$ |
| Quantum Inspired Surfaces | QIS1 | $e^{-0.8(x^2+y^2)}$ |
| | QIS2 | $x^2\,e^{-(x^2+y^2)}$ |
| | QIS3 | $(x^2 + y^2)\,e^{-0.9(x^2+y^2)}$ |
| | QIS4 | $e^{-0.4(x^2+y^2)}\big(1.1 + \cos(5x)\big)$ |
| | QIS5 | $\big(\cos(1.5x)\cos(1.5y)\big)^2$ |
| | QIS6 | $\sin^2\!\big(3\arctan(\frac{y}{x})\big)\,e^{-\sqrt{x^2+y^2}}$ |
| | QIS7 | $1 - \tanh(x^2 + y^2 - 4)$ |
| | QIS8 | $(x^2 - y^2)^2\,e^{-0.7(x^2+y^2)}$ |
| | QIS9 | $\sin^2(x + y)\ \cos^2(x - y)$ |
| | QIS10 | $\frac{1+x^2}{1+(x^2+y^2)^2}$ |
| Surrogate-Distilled Approximations | SDSA1 | $x^3 + y^3 - 3xy + \sin(x)$ |
| | SDSA2 | $\log\big(1 + x^2 + y^2\big) - \tanh(x - y)$ |
| | SDSA3 | $e^{-x^2-y^2}\ \sin(2x + y)$ |
| | SDSA4 | $\arctan(x) + \arctan(y)$ |
| | SDSA5 | $\sin(x)\cos(y) + 0.1\,xy$ |
| | SDSA6 | $3\ \sin(0.4x)\,e^{-0.05y^2} + 0.2\,xy$ |
| | SDSA7 | $2\ \sin(x + y)\,e^{-0.5x^2} + y^2$ |
| | SDSA8 | $\tanh(xy) + 0.5\ \sin(0.5x)\,y$ |
| | SDSA9 | $1.5\,x^2\cos(0.2y) + 0.3\,e^{-0.1x^2}$ |

*Continued on next page*

*Table 6 - continued from previous page*

| Category | Equation | Real-world Domain |
|---|---|---|
| Implicit Surfaces | AMHD1 | $x^3 + y^3 + z^3 - 3xyz = 0$ |
|  | AMHD2 | $x^3y + y^3z + z^3x = 0$ |
|  | AMHD3 | $x^5 + y^5 + z^5 - xyz = 0$ |
|  | AMHD4 | $x^4y - z^6 + \sin(xz) = 1$ |
|  | AMHD5 | $z^5 + x^3y^4 - e^y = 0$ |
|  | AMHD6 | $x^6 - y^4z^2 + \tan(z) = 2$ |
|  | AMHD7 | $z^3 + x^4y^3 - \cos(x) = -1$ |
|  | AMHD8 | $x^3y^2 - z^5 + \sin(yz) = 0$ |
|  | AMHD9 | $x^2y^3z - z^4 + \sin(x) = -1$ |
|  | AMHD10 | $z^3 + x^5y - e^z + xy^2 = 0$ |
|  | AMHD11 | $x^4 - y^2z^5 + \tan(z) = 2$ |
|  | AMHD12 | $z^5 + x^3y^4 - \cos(y) = 1$ |
|  | AMHD13 | $x^6y^2 - z^3 + e^x = 0$ |
|  | AMHD14 | $z^4 - x^4y + \sin(xz) = -2$ |
|  | AMHD15 | $x^3 + y^4z^2 - e^y + \cos(x) = 1$ |
|  | AMHD16 | $x^5y - z^4 + \sin(yz) = 2$ |
|  | AMHD17 | $z^3 - x^3y + e^z + 2xy = -1$ |
|  | AMHD18 | $x^4 + y^5z - \cos(xz) = 0$ |
|  | AMHD19 | $x^6 - y^3z^2 + \tan(z) = 2$ |
|  | AMHD20 | $x^2y^2z - z^5 + \sin(xz) = 1$ |
|  | AMHD21 | $z^3 + x^4y - 2e^z + xy^2 = 0$ |
|  | AMHD22 | $x^5 - y^2z^3 + \cos(xy) = -1$ |
|  | AMHD23 | $x^3 + y^4z - \tan(x) + 1 = 0$ |
|  | AMHD24 | $z^5 + x^3y^2 - 2z^2x + \sin(y) = 1$ |
| Parametric Surfaces | TCS1 | $(\sin(u^2 + v)\, e^{-v},\ \cos(uv)\, \log(1 + \lvert v \rvert),\ \frac{\sin(uv^2)}{1+u^2})$ |
|  | TCS2 | $(\cosh(u + v^2),\ \sinh(uv),\ \tanh(u^2 - v) \cos v)$ |

*Continued on next page*

*Table 6 - continued from previous page*

| **Category** | **Equation** | **Real-world Domain** |
|---|---|---|
| | TCS3 | $(e^{u-v^2}\sin u,\ u^2\cos v,\ \log(1+u^2+v^2))$ |
| | TCS4 | $(\sin(u^2)\,v,\ \cos(v^2)\,u,\ u\,e^{-v})$ |
| | TCS5 | $(u^3-v^2,\ \cos(uv^2),\ \tanh(u-v)\log(1+u^2))$ |
| | TCS6 | $((u^2+v^2)\sin u,\ (u^2+v^2)\cos v,\ \sqrt{u^2+v^2}\,\cos\sqrt{u^2+v^2})$ |
| | TCS7 | $(\log(1+u^2)\cos v,\ \sin(u+v^2),\ u^2\tanh v)$ |
| | TCS8 | $(\tanh(u^2)\sin v,\ u\,e^{-v^2},\ \frac{\cos(uv^2)}{1+u^2})$ |
| | TCS9 | $(\cos(u^2+v)\,e^{u/5},\ \sin(v^2-u),\ u\,\log(1+v^2)\tanh u)$ |
| | TCS10 | $(u\cos(v^2),\ u\sin v,\ e^{(u-v^2)/5})$ |
| | HDPS1 | $(\sinh(u/5),\ \cosh(uv/10),\ \sin(u+v)\log(1+v^2))$ |
| | HDPS2 | $(u^2\cos v,\ v^2\sin u,\ \tanh(uv))$ |
| | HDPS3 | $(e^{(u^2-v)/10}\sin v,\ \cos(uv),\ u^2+v^2)$ |
| | HDPS4 | $(\tanh(u+v^2),\ \sin(u^2v),\ \cos(u-v^2)\log(1+u^2))$ |
| | HDPS5 | $(u\cos(v^2),\ u\sin v,\ \sin(u^2+v))$ |
| | HDPS6 | $(\sinh(uv/5),\ \cos(u-v^2),\ e^{-u^2/10}\tanh v)$ |
| | HDPS7 | $(u\sin(v^2),\ v\cos u,\ \log(1+u^2+v^2)\sin u)$ |
| | HDPS8 | $(\tanh(u^2+v)\cos v,\ \sin(uv^2),\ u^2e^{-v/5})$ |
| | HDPS9 | $(e^{(u-v)/5}\sin(u^2),\ \cos(v^2-u),\ u\tanh(v^2))$ |
| | HDPS10 | $(\sin(\frac{u^2v}{10}),\ v\cos u,\ \log(1+u^2)\sinh(v/5))$ |
| | TRPS1 | $(\log(1+u^2)\cos(v^2),\ \sin(u+v),\ u\,e^{v^2/10})$ |
| | TRPS2 | $(\frac{u^3\sin v}{100},\ \cos(uv^2),\ \tanh(u-v^2))$ |

*Continued on next page*

*Table 6 - continued from previous page*

| **Category** | **Equation** | **Real-world Domain** |
|---|---|---|
| | TRPS3 | $(\sin(\frac{u^2}{v^2+1}),\ e^{-v}\cos u,\ \sinh(v^2))$ |
| | TRPS4 | $((5+v\cos(u/2))\sin u,\ (5+v\cos(u/2))\cos u,\quad v\sin(u/2))$ |
| | TRPS5 | $((5+\sin(uv))\cos u\sin v,\ (5+\sin(uv))\sin u\sin v,\ (5+\sin(uv))\cos v)$ |
| | TRPS6 | $(\cos u\sin v,\ \sin u\sin v,\ \cos v+\frac{u}{2})$ |
| | TRPS7 | $(u,\ v,\ \sin\sqrt{u^2+v^2}+\frac{\cos u\sin v}{5})$ |
| | TRPS8 | $(\cos u\,(5+\sin 3v),\ \sin u\,(5+\sin 3v),\ \cos(3v)+u/2)$ |
| | TRPS9 | $(\sin(2u)\cos^2 v, \cos(2u)\sin v, \sin u\cos v)$ |
| | TRPS10 | $(\sinh(u/5)\cos v, \cosh(u/5)\sin v, \tanh(v)\cos u)$ |