Symbolic Regression recent work

Present by: Junqi Qu

Symbolic Regression

$y = wx + b$

$f(x) = y$

Core Definition

$x \in \mathbb{R}^d$ x_1, x_2

- Given N data points $\{x_i, y_i\}_{i=1}^N$, discover a symbolic expression f: x → y that captures the underlying relationship.
- Unlike conventional regression (linear, polynomial, neural), the functional form itself is the unknown — SR searches the space of mathematical expressions.

SR vs. Standard Regression

Aspect	Standard Reg.	Symbolic Reg.
Form known?	✓ Yes	X No — discovered
Output	Parameters	Expression
Interpretable	Partial	Fully
Search space	Continuous	Discrete / huge
Opt. target	Loss fn.	Expression + loss

Binary tree

$\sin(x_1) + (x_2$

prefix

$- + \times \div$

sin cos log exp

$x_1 + x_2 + x_3$

logic or and

Core Challenges in Symbolic Regression

Combinatorial Explosion

- Expression space is doubly-exponential in expression length
- With k operators and d variables, trees of depth D yield O(k^(2^D)) distinct expressions
- No tractable exhaustive search — NP-hard in general

Non-Monotone Error Landscape

- Structural similarity (edit distance) ≠ numerical similarity
- MSE does not decrease monotonically as search nears the target
- No optimal substructure → heuristic search can diverge

Discrete Optimization

- Expression trees are discrete objects; gradient-based methods inapplicable directly
- Crossover/mutation in GP changes structure but not in numerically meaningful ways
- Constants within expressions add a nested continuous optimisation problem

Overfitting vs. Simplicity

- Shorter expressions generalise better (Occam's razor / MDL principle)
- But accuracy and simplicity are in direct tension
- Multi-objective Pareto optimisation required (R^2, complexity, time)

Evolution of SR Methods: From GP to Neural Approaches

1990s–2000s	2010s	2020–2023	2024–2026
Genetic Programming	**Hybrid & Physics-Guided**	**RL & Transformer**	**Latent Space & MDL**
Evolve expression trees via selection, crossover, mutation. Eureqa, GPlearn, PySR. Slow, heuristic, no gradient.	GP + domain knowledge. AI Feynman (symmetry rules). BSR (Bayesian priors). Improved but dataset-specific.	DSR (risk-seeking policy gradients). TPSR (Transformer planning). E2ESR / NeurSR: seq-to-seq. Fast inference, low recovery rate.	GENSR: generative VAE latent space. MDLFormer: MDL as search objective. Principled, theoretically grounded.
Limitation	**Limitation**	**Limitation**	**Advance**
Discrete space; MSE landscape non-monotone; no gradient signal.	Rules hand-crafted for specific datasets; noise-sensitive; not general.	Generative methods have low formula recovery rates; search still uses MSE.	Principled continuous search; MDL monotone objective; theoretically grounded.

GENSR: SYMBOLIC REGRESSION BASED ON EQUATION GENERATIVE SPACE

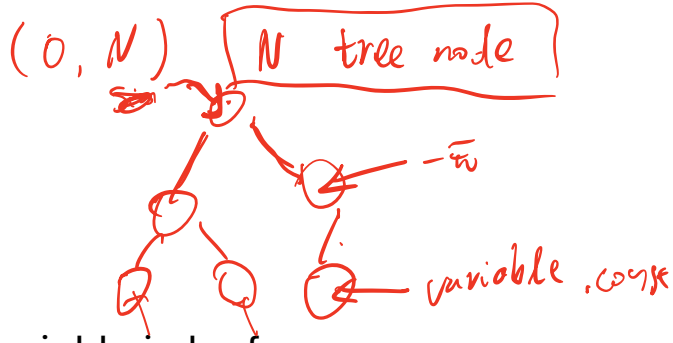
poster

Qian Li[1,2], **Yuxiao Hu**[2,3], **Juncheng Liu**[4], **Yuntian Chen**[2*]
[1]Shanghai Jiao Tong University, Shanghai, China
[2]Eastern Institute of Technology, Ningbo, China
[3]The Hong Kong Polytechnic University, Hong Kong, China
[4]Imperial College London, London, England
qianl01205@sjtu.edu.cn, yuxiao.hu@connect.polyu.hk,
junchengliu23@imperial.ac.uk, ychen@eitech.edu.cn

Motivation

- **The Problem:** Discrete equation spaces rely on structural similarity (edit distance), which is an unreliable proxy for numerical performance.
- **Generative Latent Space:** GENSR reparameterizes the discrete equation space into a continuous generative latent space.
- **The Solution (GENSR):** A framework following a "map construction -> coarse localization -> fine search" paradigm.

Preparation



- Preparation of synthetic data
  - randomly generate tree structure first
  - Then fill in operators in the nodes and variable in leafs

$$(\boldsymbol{x}, y)_{i=1}^{m} \in \mathbb{R}^{m \times (D+1)}$$

- Pre-processing of numeric samples

$$\{(\boldsymbol{x}, y)\}_{i=1}^{m} \in \mathbb{R}^{m \times 3(D+1)}$$

0.0125.

(+, 0125, E-10)

(+/-, 0000-9999, E-100, -E100)

- Pre-processing equation
  - Binary tree in prefix order
  - [<BOS>] and [<EOS>] are starting and ending tokens, padding to fixed length m

$$\boldsymbol{F} \in \mathbb{R}^{m \times d}$$

m

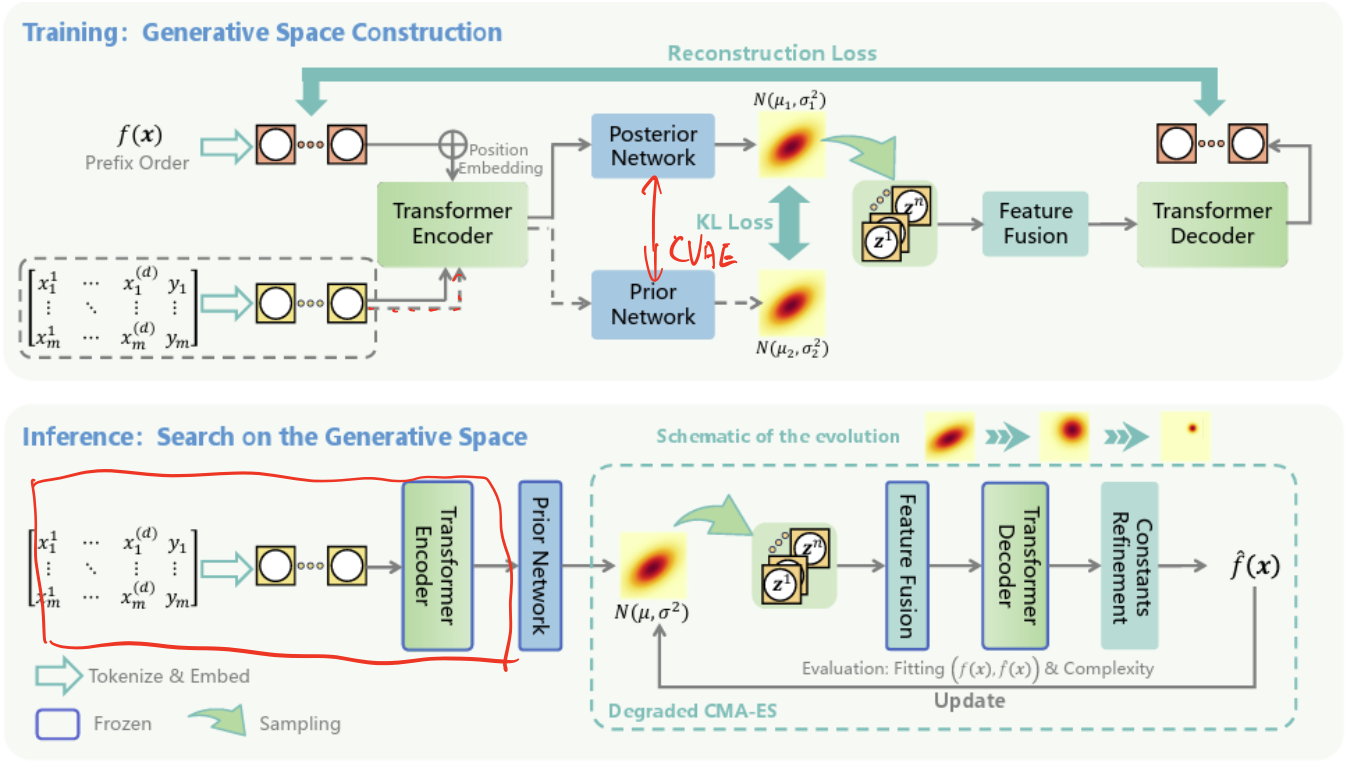


Figure 1: The overview of GenSR. During training, the dashed lines denote the prior branch, while the solid lines indicate the posterior branch. During inference, only the prior branch is used.

Latent Space

$$q(z|X,F) = \mathcal{N}(\mu_1, \sigma_1^2 I)$$

$$p(z|X) = \mathcal{N}(\mu_2, \sigma_2^2 I)$$

- Latent variable z's conditional distribution
- Training Objective

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{KL}$$

- Cross Entropy + KL divergence

$$\mathcal{L}_{CE} = -\sum_{i=1}^{m} \log p_\theta(f_i|z_i^1)$$

$$\mathcal{L}_{KL} = D_{KL}(\mathcal{N}(\mu_1, \sigma_1^2) \| \mathcal{N}(\mu_2, \sigma_2^2)) = \frac{1}{2} \sum_{j=1}^{d} \left[\frac{(\mu_{1,j} - \mu_{2,j})^2}{\sigma_{2,j}^2} + \frac{\sigma_{1,j}^2}{\sigma_{2,j}^2} - \ln \frac{\sigma_{1,j}^2}{\sigma_{2,j}^2} - 1 \right]$$

$$\lambda(t) = \begin{cases} \frac{t}{T/2} & t \leq T/2 \\ 1.0 & t > T/2 \end{cases}$$

Bayesian Perspective

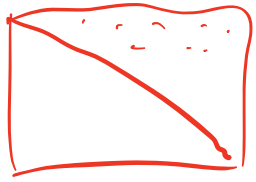
$$
\begin{aligned}
\log p(\boldsymbol{F}|\boldsymbol{X}) &= \int_z q(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F}) \log p(\boldsymbol{F}|\boldsymbol{X}) d\boldsymbol{z} \\
&\geq \int_z q(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F}) \log \frac{p(\boldsymbol{F},\boldsymbol{z}|\boldsymbol{X})}{q(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F})} d\boldsymbol{z} \\
&= \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F})} \left[\log p(\boldsymbol{F}|\boldsymbol{X},\boldsymbol{z})\right] - D_{\mathrm{KL}} \left(q(\boldsymbol{z}|\boldsymbol{X},\boldsymbol{F}) \,||\, p(\boldsymbol{z}|\boldsymbol{X})\right). \qquad (1)
\end{aligned}
$$

max

min

inference

GMM



- *Covariance Matrix Adaptation Evolution Strategy (CMAES)*

1. **Initialization**: Set the initial search distribution as $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0^2 \boldsymbol{I})$ (with $\boldsymbol{\mu}_0$ and $\boldsymbol{\sigma}_0$ are the output of the prior network).
2. **Iteration** (for each generation i):
   (a) Sample multiple latent vectors $\{\boldsymbol{z}_j\}$ from $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2 \boldsymbol{I})$.
   (b) Decode $\{\boldsymbol{z}_j\}$ into candidate equations $\{f_j(\boldsymbol{x})\}$ using the well-trained decoder.
   (c) Refine the constants of $\{f_j(\boldsymbol{x})\}$ via BFGS optimization.
   (d) Evaluate the fitness of candidate equations: Fitness $= R^2 - \omega \cdot$ complexity.
   (e) Select the top-p candidates and update the distribution parameters $\boldsymbol{\mu}_i \rightarrow \boldsymbol{\mu}_{i+1}$ and $\boldsymbol{\sigma}_i \rightarrow \boldsymbol{\sigma}_{i+1}$ using the CMA-ES update rules. This yields the updated distribution

Results (srbench benchmark)

- Srbench
  - *SRBench benchmark (La Cava et al., 2021; Cavalab, 2022)*
  - *119 Feynman equations, 14 ODE-Strogatz challenges, and 57 black-box regresssion tasks*

$R^2 \uparrow$ complexity $\downarrow$

- Against *18 baselines*

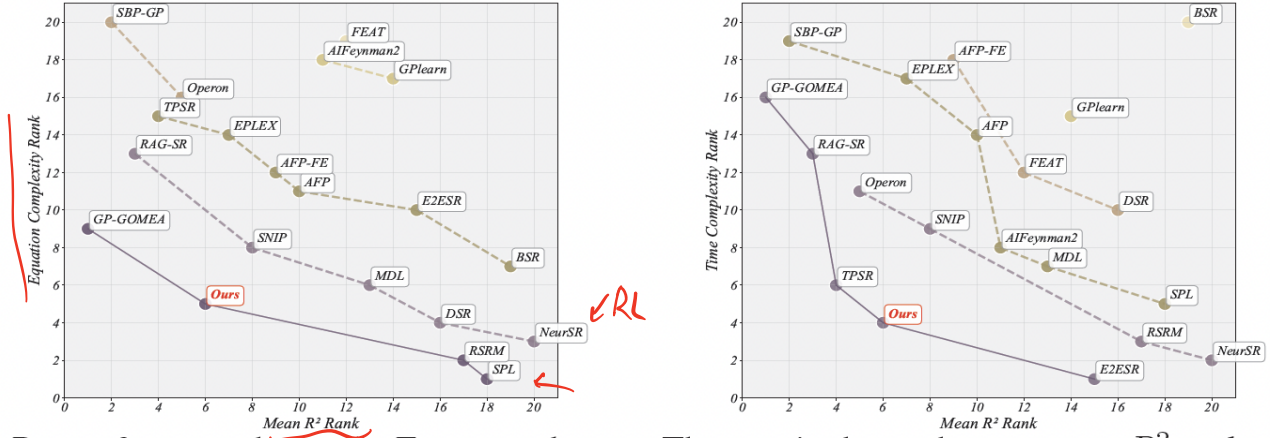


Figure 2: Pareto front results on the Feynman dataset. The x-axis shows the mean test R^2 rank, while the y-axis shows equation complexity rank (left) and time complexity rank (right). Solid lines indicate the optimal Pareto front, and dashed lines show lower-ranked fronts from bottom-left to top-right.

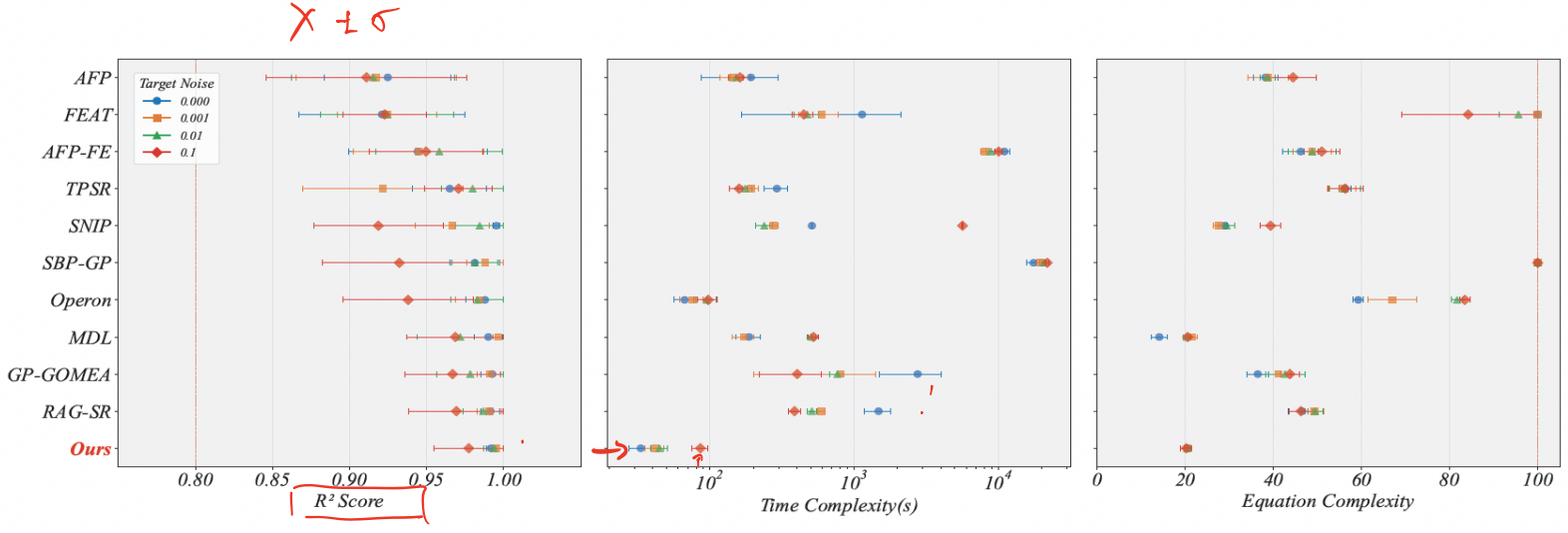


Figure 3: Comparison on the Strogatz dataset under different noise levels. Subplots (left to right) report R^2 score, time complexity (s), and equation complexity. Noise levels are represented by blue circles (0.000), orange squares (0.001), green triangles (0.01), and red diamonds (0.1), with error bars indicating standard deviations. Only methods whose mean R^2 across noise settings exceeds 0.9 are included.

Latent space visualization

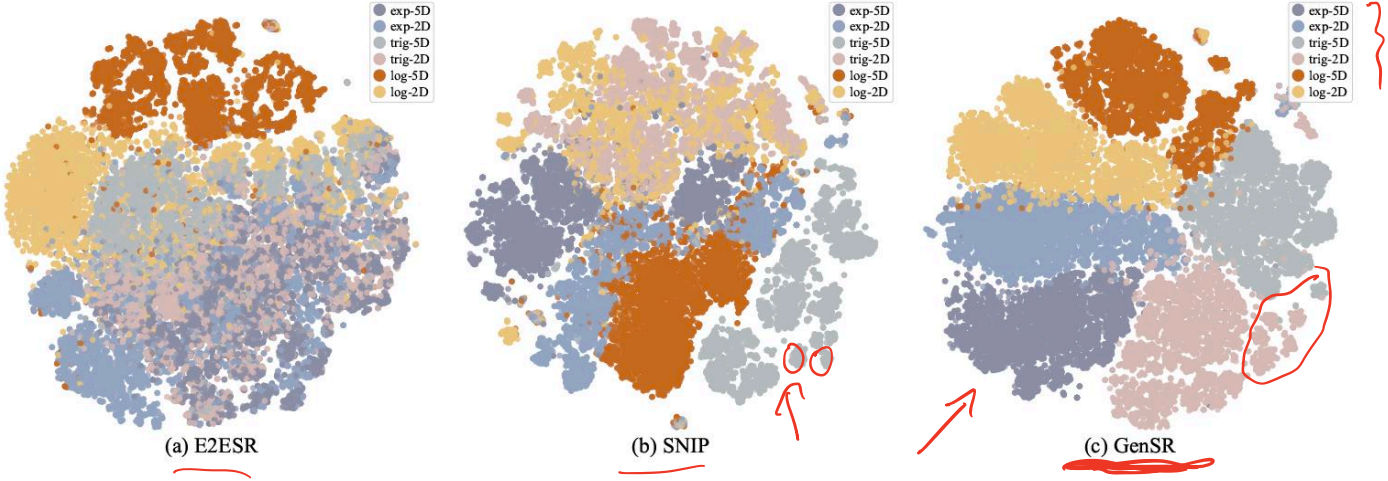


Figure 4: 2D t-SNE visualization of latent variables from E2ESR, SNIP, and GenSR. The legend distinguishes six categories, corresponding to equations from three representative function families, each evaluated under 2D and 5D input dimensionality, illustrating the clustering behavior of the learned latent spaces.

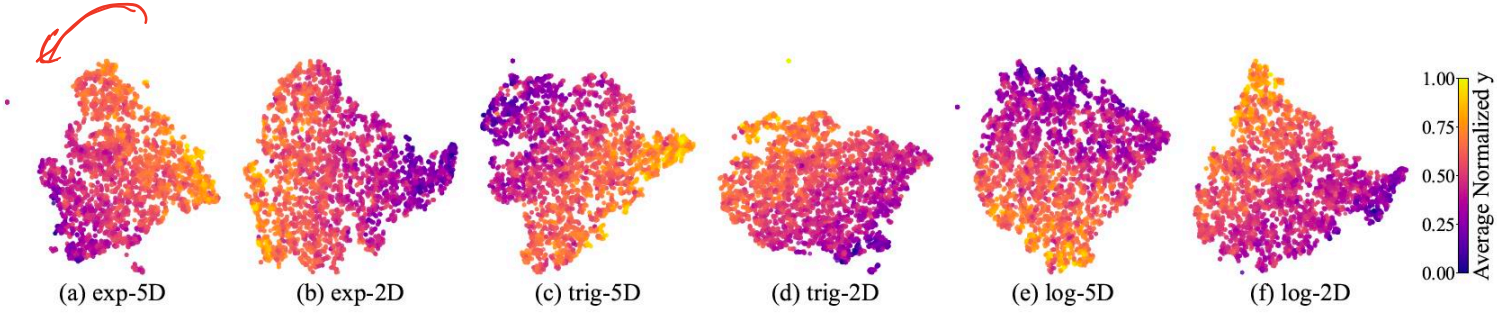


Figure 5: 2D t-SNE visualization of GenSR latent variables for equations from three function families (exponential, trigonometric, logarithmic) under 2D and 5D input settings, shown in subplots (a)–(f). Colors indicate the average of normalized y values, as displayed in the accompanying color bar.

SYMBOLIC REGRESSION VIA MDLFORMER-GUIDED SEARCH: FROM MINIMIZING PREDICTION ERROR TO MINIMIZING DESCRIPTION LENGTH

Zihan Yu
Department of Electronic Engineering, BNRist
Tsinghua University
Beijing, China

Jingtao Ding*
Department of Electronic Engineering, BNRist
Tsinghua University
Beijing, China

Yong Li*
Department of Electronic Engineering, BNRist
Tsinghua University
Beijing, China

Depeng Jin
Department of Electronic Engineering, BNRist
Tsinghua University
Beijing, China

Problem define

- Keep the discrete space but transform the *search objective* from non-monotonic Error to monotonic Complexity (MDL).

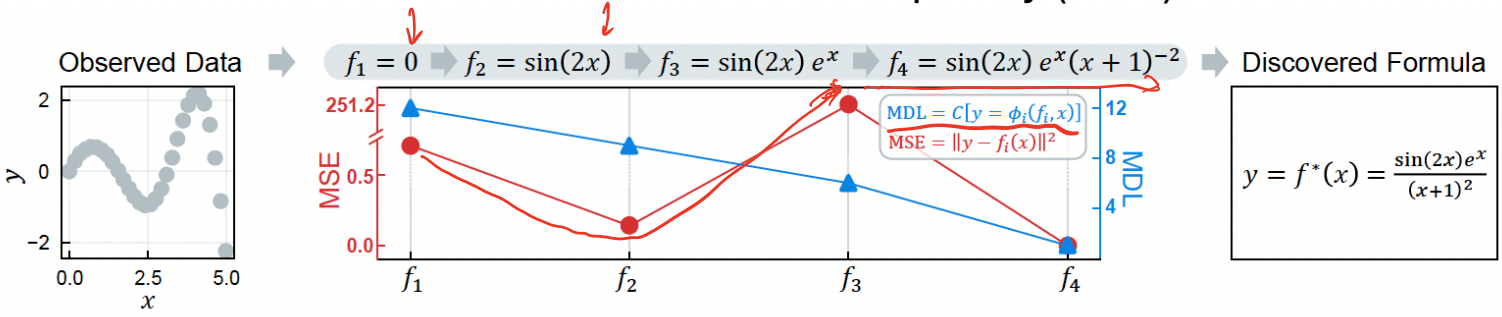


Figure 1: **Comparison of the two search objectives.** In the searching route leading to the target, f^*, the prediction error (measured by the mean square error, MSE) does not decrease monotonically as the candidate formula's form gets closer to the target one, whereas the minimum description length (MDL) does. Here, ϕ_i denotes the function $f^* = \phi_i(x, f_i)$ and $C[\phi_i]$ is its complexity.

Minimum Description Length(MDL)

MSE → MDL

- Introduce Minimum Description Length (MDL) as the ultimate monotonic guide: it strictly decreases as structural transformations correctly approach the ground truth.

$\hat{f}(x) \xrightarrow{4} f(x)$

1 x^2 3 0

$x^2 + 3x + 1$

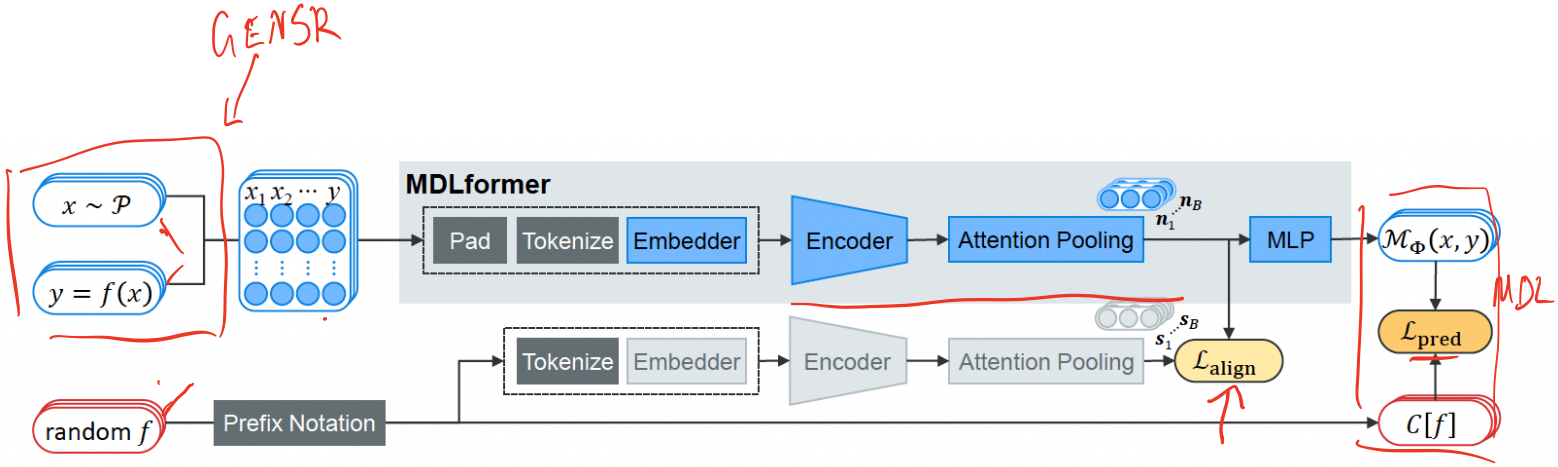


Figure 2: **Schematic diagram of the architecture and learning process of MDLformer.**

Losses(objective goal)

Primary learning objective for prediction. To train the MDLformer for predicting the corresponding minimum description length (MDL) based on numerical input, we optimize the mean square error, $\mathcal{L}_{\text{pred}}$, between MDLs estimated by the MDLformer and ground-truths:

$$\mathcal{L}_{\text{pred}} = \frac{1}{B}\sum_{i=1}^{B}\left(C[f_i] - \mathcal{M}_{\Phi}(x_i, y_i)\right)^2, \tag{1}$$

where B denotes the batch size, $C[f_i]$ is the complexity of the symbolic function f_i.

Alignment as an auxiliary learning objective. The auxiliary objective, proposed by Meidani et al. (2023), aims to facilitate a mutual understanding of both numeric and symbolic domains and thus empower better cross-modal prediction. Specifically, as suggested by Meidani et al. (2023), we introduce a symbolic encoder to map the prefix notation of f into a compact representation $\boldsymbol{s}$, which has a structure similar to the numeric encoder, as depicted in Figure 2. The latent spaces of these two encoders are aligned by optimizing a symmetric cross-entropy loss over similarity scores:

$$\mathcal{L}_{\text{align}} = -\left(\sum_{i=1}^{B}\log\frac{\exp(\boldsymbol{n}_i\cdot\boldsymbol{s}_i/\tau)}{\sum_{j=1}^{B}\exp(\boldsymbol{n}_i\cdot\boldsymbol{s}_j/\tau)} + \sum_{i=1}^{B}\log\frac{\exp(\boldsymbol{s}_i\cdot\boldsymbol{n}_i/\tau)}{\sum_{j=1}^{B}\exp(\boldsymbol{s}_i\cdot\boldsymbol{n}_j/\tau)}\right), \tag{2}$$

where B is the batch size, τ is the temperature parameter, $\boldsymbol{s}_i$ and $\boldsymbol{n}_i$ are encoded representations of i-th numeric data and symbolic function, respectively. Note that this loss is calculated per batch.

MCTS

results

SRBench

Table 1: **Recovery rate and search time of different methods in both Strogatz and Feynman datasets.** Each experiment is conducted at ten random seeds and four noise levels.

Type	Method	Strogatz (14 problems)		Feynman (119 problems)	
		R. Rate ↑	Time (s)	R. Rate ↑	Time (s)
Regression	FEAT (La Cava et al., 2019)	0.19%	636.6	0.00%	1532
Generative	NeurSR (Biggio et al., 2021)	1.79%	15.71	2.44%	24.78
	E2ESR (Kamienny et al., 2022)	3.78%	4.044	10.40%	4.576
	SNIP (Meidani et al., 2023)	6.79%	1.457	1.60%	2.196
Search-based	GPlearn (Stephens, 2016)	9.21%	966.2	16.89%	3349
	AFP (Schmidt & Lipson, 2010)	10.90%	160.7	17.51%	3845
	AFP-FE (Schmidt & Lipson, 2010)	12.86%	9532	20.80%	25138
	EPLEX (La Cava et al., 2016)	6.02%	446.8	10.10%	11548
	SBP-GP (Virgolin et al., 2019)	2.44%	20089	2.88%	28933
	GP-GOMEA (Virgolin et al., 2021)	8.46%	1100	10.32%	3456
	Operon (Burlacu et al., 2020)	4.29%	83.58	7.97%	2656
	SPL (Sun et al., 2022)	8.12%	363.7	10.48%	263.3
	DSR (Petersen et al., 2021)	18.05%	784.3	18.60%	1042
	RSRM (Xu et al., 2024)	4.43%	133.2	15.40%	127.1
	AIFeynman2 (Udrescu et al., 2020)	15.27%	241.3	27.24%	708.6
	BSR (Jin et al., 2020)	0.38%	25346	0.70%	30635
	Ours	**66.78%** (+6.82 formulas)	338.3	**33.93%** (+7.96 formulas)	660.5

Figure 4: **Recovery rate at different noise levels.** † and ‡ denote generative and regression methods, the others are search methods. The error bars depict the 95% confidence interval.

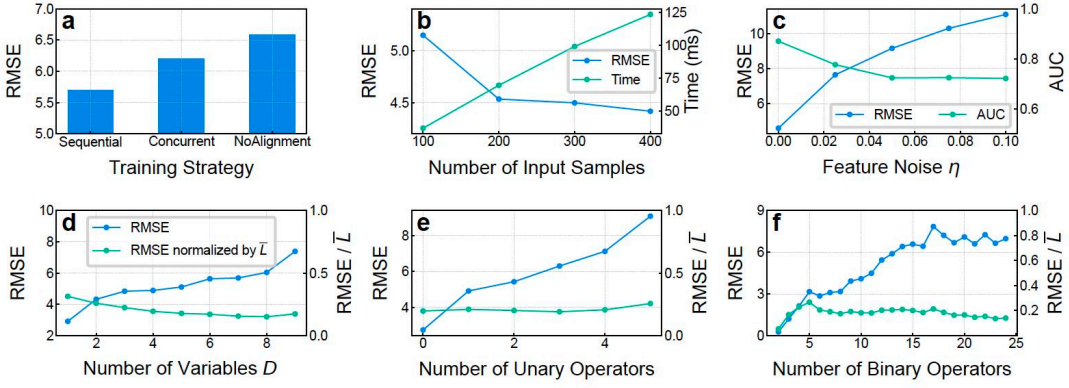
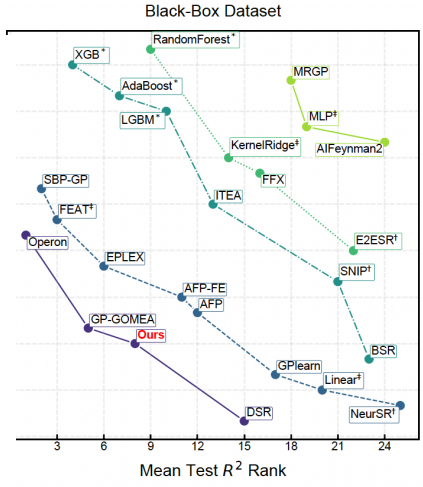


Figure 7: **Ablation Study.** The prediction performance of MDLformer with respect to **a**: trained with three alignment strategies, **b**: number of input pairs N, **c**: feature noise η, **d**: number of variables D, **e**: number of unary operators, and **f**: number of binary operators. In **d,e,f** we also plot the RMSE normalized by the average formula length $\bar{L}$.



: 5: **Pareto fronts on black-box dataset.** and ∗ denote generative, regression, and on-tree methods, respectively, the others are methods. The colored lines mark the Pareto front in different ranks, from bottom left (best) to upper right (worst).