MOD-SR: Unifying Multimodal Learning and Direct Optimization with Diffusion Model for Symbolic Regression

Chuyang Xiang* Yichen Wei* Junchi Yan†

School of Artificial Intelligence, Shanghai Jiao Tong University

ICML International Conference On Machine Learning

GitHub Code

MOTIVATION & OVERVIEW

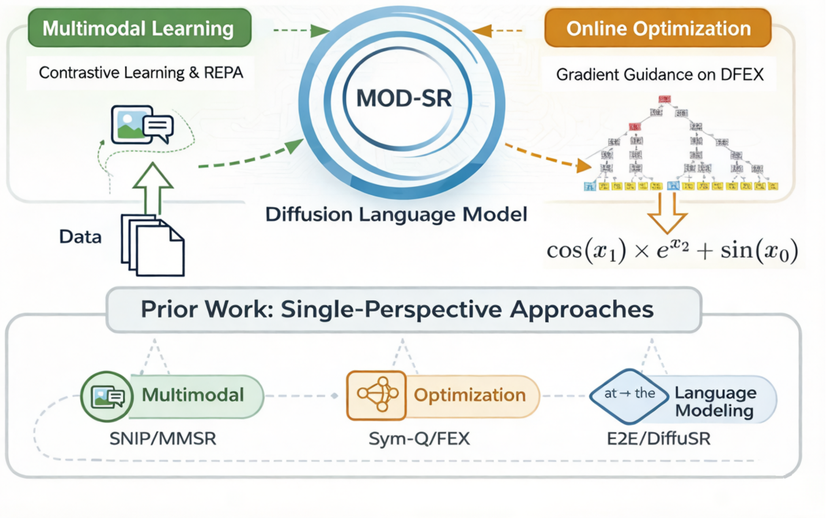

Figure 1: MOD-SR adopts gradient-guided diffusion model, being the first unified model integrating multimodal learning and direct optimization for symbolic regression.

Symbolic regression (SR) seeks to discover explicit mathematical expressions that describe observed data. Two fundamental limitations exist in prior work:

Multimodal approaches (SNIP/MMSR): Strong distribution learning but cannot directly optimize objectives during inference.

Optimization approaches (Sym-Q/FEX): Direct objective optimization but suffer from exponential slowdown with dimensionality.

MOD-SR is the first to **unify both perspectives** with one model.

PROBLEM FORMULATION

Given dataset D = $\{(x_i, y_i)\}$ where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, find f such that $y_i = f(x_i)$.

$$p(x_0|c, y^*) \quad \text{where} \quad c = Enc_{num}(D)$$

- x_0: latent expression representation in embedding space
- c: cross-modal condition from frozen numerical encoder
- y*: optimal objective score (fitting error / complexity)

This Bayesian formulation captures inherent ambiguity: multiple expressions can fit the same finite, noisy data equally well.

METHOD COMPARISON

Table 1. Comparison of different symbolic regression methods.

Method	Perspective	Distribution Learning	Optimization
E2E (Kamienny et al., 2022)	AR (Translation)	Transformer	Beam Search
MMSR (Li et al., 2025b)	AR (Multimodal)	Transformer	Beam Search
SNIP (Meidani et al., 2024)	AR (Multimodal)	Transformer + CL	Heuristic (GP)
FEX (Liang & Yang, 2025)	MDP (Optimization)	/	RL
Sym-Q (Tian et al., 2024)	MDP (Optimization)	Q-Learning	RL (REINFORCE)
DBSR (Bastiani et al., 2025)	MDP (Optimization)	/	RL (GRPO)
MetaSymNet (Li et al., 2025a)	EQL (Optimization)	/	Novel NN
Symbolic Diffusion (Tymkow et al., 2025)	Bayesian (Translation)	Discrete diffusion	/
DiffuSR (Han et al., 2025)	Bayesian (Translation)	DiffusionLM	Heuristic (GP)
GenSR (Li et al., 2026)	Bayesian (Multimodal)	Conditional VAE	Heuristic (CMA-ES)
MOD-SR (Ours)	**Bayesian (Multimodal + Optimization)**	**Gradient-guided DiffusionLM + REPA/CL**	

AR: Autoregressive, MDP: Markov Decision Process, EQL: Equation Learner, CL: Contrastive Learning, DiffusionLM: Diffusion Language Model (Li et al., 2022), REPA: Representation Alignment.

MODEL ARCHITECTURE

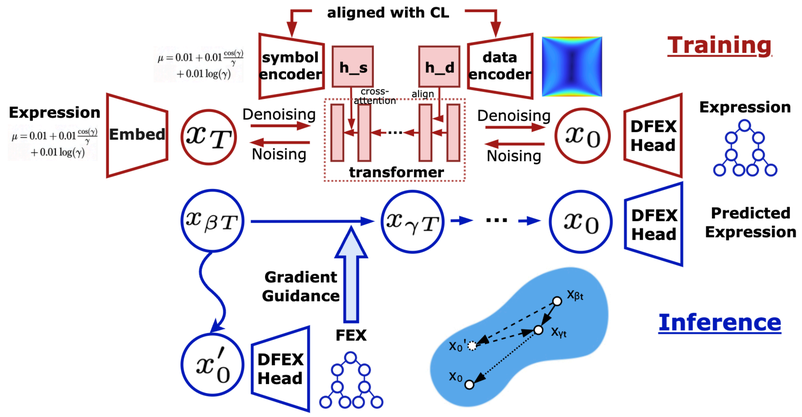

Figure 2: Overview of our method: multimodal encoders pre-trained with contrastive learning condition the diffusion model, representation alignment with the symbolic encoder from the same contrastive framework during training, DFEX relaxes trees for differentiable evaluation, enabling gradient guidance to directly optimize objectives during sampling.

DFEX: DIFFERENTIABLE FINITE EXPRESSION TREE

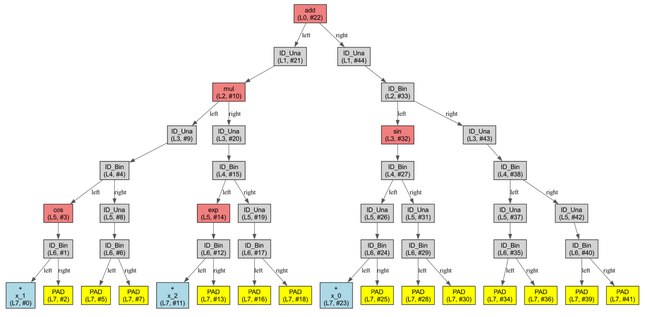

Figure 3: Example of DFEX tree structure of depth 3 encoding expression $y = \cos(x_1) \times e^{x_2} + \sin(x_0)$.

$$c = (\sum_i p_{m,i} \cdot i) \times 10^{(\sum_j p_{e,j} \cdot j)}$$
$$v = (p_+ - p_-) \times \sum_k p_{x,k} \cdot x_k$$

DFEX is a fixed-depth tree enabling gradient guidance:

Alternating binary/unary layers: Binary for ops (+,x), Unary for functions (sin,exp)

Softmax relaxation: Token probabilities via softmax

Mantissa p_m, exponent p_e for constants; sign p_+/p_-, variable p_x for variables.

GRADIENT GUIDANCE THROUGH DFEX

$$G(x_t, t) = -w(\hat{t}) \cdot \nabla_{x_t} \mathcal{L}_{fit}(\hat{x}_0(x_t); D, y^*)$$

Key Idea:

Look-ahead guidance steers the diffusion sampling process toward objectives by computing gradients of the fitting loss with respect to the noisy latent x_t. This enables direct optimization during inference without requiring separate optimization loops.

Components:

$\hat{x}_0(x_t)$: The predicted clean expression via Tweedie's formula. We decode x_t to get an estimated expression that can be evaluated on the dataset.

L_{fit}: Fitting loss measuring how well the predicted expression fits data D. Can be MSE for accuracy or node count for simplicity.

$w(\hat{t})$: Time-dependent weight schedule controlling guidance strength during sampling.

Gradient Computation via Chain Rule:

$\partial L/\partial x_t = (\partial L/\partial \hat{x}_0) \cdot (\partial \hat{x}_0/\partial x_t)$

The gradient flows back from the objective L through the predicted expression $\hat{x}_0$ to the noisy latent x_t, enabling gradient-based optimization in the embedding space.

SUBTREE BFGS

Algorithm 1 Optimize Expression via Subtree BFGS

Input: dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, DFEX tree, max iterations K, top-k size k
Output: optimized expression e^*
Initialize: logits $\ell \in \mathbb{R}^{L \times |\mathcal{V}|}$ (strong init: $\ell_{target} = 10$)
$X \leftarrow [x_1, \ldots, x_N]$, $Y \leftarrow [y_1, \ldots, y_N]$
for *iter* $= 1$ to K **do**
  Sample subtree $\mathcal{T}_k \leftarrow \textsc{SampleSubtree}(d_{sub})$ {non-ID subtree}
  Build top-k indices $\mathcal{I}_k \leftarrow \text{TopK}(\ell, k)$ {Restrict optimization space}
  $\ell_{topk} \leftarrow \text{Gather}(\ell, \mathcal{I}_k)$; $\ell_{topk} \leftarrow \text{Clip}(\ell_{topk}, -c, c)$
  $P \leftarrow \text{softmax}(\ell_{topk}, \tau)$; $M \leftarrow \textsc{Mask}(\mathcal{T}_k)$ $\{M \in \{0,1\}^{L \times k}\}$
  $\hat{y} \leftarrow \textsc{ComputeRelaxed}(P \odot M, X)$ {Alg. 3}
  $\mathcal{L}_{fit} \leftarrow \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2$
  $\mathcal{L}_{01} \leftarrow -\frac{1}{k} \sum_{i,j} (P_{ij} - 0.5)^2$
  $\mathcal{L} \leftarrow \mathcal{L}_{fit} + \lambda_{01} \mathcal{L}_{01}$
  $\ell_{topk} \leftarrow \text{BFGS}(\mathcal{L}, \ell_{topk})$ {L-BFGS-B on top-k logits}
  $\ell \leftarrow \text{Scatter}(\ell, \mathcal{I}_k, \ell_{topk})$
end for
$e^* \leftarrow \arg\max_j P_{ij}$ for each position i {discretize}
return e^*

Algorithm 1: Optimize Expression via Subtree BFGS.

Subtree BFGS Optimization:

Given a DFEX tree x and dataset D, we optimize via BFGS on subtrees to minimize fitting loss L_fit:

Top-k restriction: Focus optimization on top-k candidate tokens at each node, reducing search space.

Random subtree masking: Randomly mask subtrees during optimization.

Chain rule: Compute $\partial L/\partial x_t$ through $\hat{x}_0$ prediction, backpropagating from loss to latent.

This enables direct optimization of expression structure during diffusion sampling, guided by gradient of the fitting objective.

RESULTS (WITHOUT POST-PROCESSING)

Table 2. Performance comparison of different SR methods *without* post-processing on benchmark datasets. The table shows R^2 scores (higher is better) and average number of nodes (lower is better) for each method.

Group	Dataset	MOD-SR		DiffuSR*		SNIP		E2E		SymbolicGPT†	
		R^2	$\mathcal{C}$	R^2	$\mathcal{C}$	R^2	$\mathcal{C}$	R^2	$\mathcal{C}$	R^2	$\mathcal{C}$
Standard	Nguyen	0.984	16.6	**0.986**	8.6	0.938	15.2	0.671	19.3	0.671	21.6
	Keijzer	**0.970**	17.3	-	-	0.802	17.4	0.861	19.5	0.603	24.5
	Koza	**1.000**	10.0	-	-	**1.000**	9.5	**1.000**	19.0	0.661	29.2
	Constant	**0.983**	15.5	0.964	10.6	0.896	12.1	0.755	16.6	0.702	38.5
	Livermore	**0.989**	16.6	0.943	9.4	0.947	14.0	0.720	20.7	0.563	41.2
	Vladislavleva	**0.907**	21.1	-	-	0.769	20.6	0.743	33.8	0.541	36.6
	R	0.985	15.7	-	-	0.774	16.7	**0.991**	26.0	0.704	25.2
	Jin	**0.995**	18.5	0.797	14.5	0.964	16.7	0.832	15.7	0.772	36.9
SRBench	Feynman	**0.788**	16.4	-	-	0.616	18.1	0.735	22.0	0.538	26.8
	Strogatz	**0.970**	15.8	-	-	0.854	21.3	0.754	31.5	0.523	32.6

*: DiffuSR is not open-sourced, all of its test data is directly taken from the original paper (also top-20 sampling). They only test on 4 benchmarks, so there are blanks. †: We don't include Symbolic Diffusion (Tymkow et al., 2025) since it's reported to have equivalent performance to SymbolicGPT (Mojtaba Valipour, 2021), which is inferior in our experiments.

SYMBOLIC LATENT SPACE ANALYSIS (T-SNE)

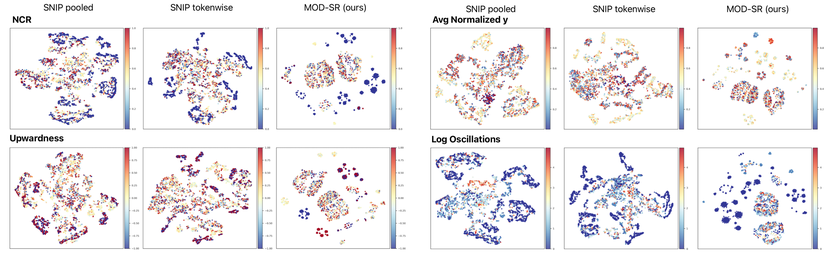

Figure 4: t-SNE visualization comparing three representation spaces across four function-level attributes (NCR, Upwardness, Avg Normalized y, Log Oscillations).

Three Representation Spaces Compared:

SNIP pooled: Mean-pooled global vector (0.65 reconstruction accuracy). Loses token-level information, making it unsuitable for latent diffusion.

SNIP token-wise: No pooling, preserves all token information (0.9997 accuracy).

MOD-SR embedder: Jointly trained with the denoiser.

Key Findings from t-SNE Visualization:

Structured clustering: MOD-SR forms compact, clearly separated clusters compared to broad clouds in SNIP spaces.

Attribute consistency: Higher within-cluster homogeneity in MOD-SR; nearby points have similar functional behavior across NCR, Upwardness, and Oscillation attributes.

Extreme value separation: Clearer isolation of extreme values (red/blue regions) in MOD-SR compared to intermixed distributions in SNIP baselines.

Implications for Latent Diffusion:

SNIP pooled discards token identity, rendering it unusable for latent diffusion. While SNIP token-wise space works, it requires a separate decoder. MOD-SR's denoising objective actively reshapes latent geometry into semantically meaningful regions, providing reliable geometry for both generation and optimization in the embedding space.