SELF-REGENERATION AS A SEARCH SIGNAL FOR DIFFUSION SYMBOLIC REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Symbolic regression seeks explicit mathematical expressions that describe relationships in numerical observations, but the vast discrete expression space makes it difficult to find high-quality solutions. Pretrained diffusion generators learn expression priors and directly produce candidates, while their continuous reverse process provides an interface for inference-time refinement before discrete decoding. Yet black-box fitness evaluates only completed expressions and does not directly indicate how to steer this process. We observe that time-aligned deviations between an original generation trajectory and its self-regeneration trajectories contain reusable directional information that can guide subsequent denoising toward useful symbolic transitions. Based on this insight, we introduce Self-Regeneration Controlled Diffusion Evolution (SR-CDE), which constructs such deviations when the target trajectory is unknown and assigns suitable donors to individual recipient contexts. SR-CDE injects the selected deviations during reverse diffusion and uses black-box fitness on the original regression task to select and evolve improved trajectory states. It thereby turns temporal information from past generations into trajectory states that can be reused and iteratively refined during inference-time search, while keeping the pretrained generator frozen. Across a 73-task symbolic regression benchmark, SR-CDE achieves competitive overall performance against leading methods. Controlled experiments on Hard-18 further show that full trajectory deviations consistently outperform endpoint-only transfer and random directions matched in summary statistics, confirming the joint importance of temporal application and self-regeneration-specific directional information.

1 INTRODUCTION

Symbolic regression seeks explicit mathematical expressions that explain numerical observations, requiring the discovery of both a functional structure and its coefficients (**?**). Pretrained neural generators reuse knowledge from synthetic equation–data pairs to propose expressions for a new regression task (**??**). These proposals provide a starting point for search, but improving their numerical fit can require coordinated structural changes, such as revising an operator together with its arguments. Inference-time search must therefore connect the generator's structural knowledge with the numerical errors of its current candidates.

Continuous diffusion provides an interface for such revisions: the entire expression remains represented by jointly updated token embeddings until decoding (**?**). In standard autoregressive generation, changing an earlier decision requires revisiting the prefix on which later tokens depend. In embedding diffusion, multiple positions can instead be modified within the same reverse process, with subsequent denoising responding to those modifications. DiffuSR and MOD-SR instantiate this form of generation for symbolic regression (**??**). Their intermediate states expose both where an intervention can be applied and how it propagates toward a complete expression. Exploiting this interface requires identifying interventions that improve the decoded expression's numerical fit.

The obstacle is that numerical fitness is evaluated after discrete decoding. Deterministic hard decoding partitions endpoint embeddings into regions that produce the same expression and therefore the same fitness under a fixed evaluator. Moving within such a region supplies no new fitness feedback; crossing its boundary changes the expression but need not improve its fit. Euclidean displacement

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

alone does not specify whether a structural revision improves the objective. Intervening earlier in the reverse process adds another dependency: the effect of a perturbation is mediated by all subsequent denoising steps. Search needs information about transitions that survive this process and reach better expressions.

Representation learning and objective relaxation offer two established routes to search guidance. GenSR trains a generative latent space with local numerical smoothness and searches it with a modified CMA-ES (**??**). MOD-SR introduces a differentiable expression relaxation to route fitting-objective gradients into reverse diffusion (**?**). Here, we investigate a different source of information: *can a frozen diffusion generator's own reconstruction behavior provide useful trajectory controls, selected solely through the fitness of decoded expressions?*

We probe the generator through *self-regeneration*: using a current candidate to define a reconstruction task for the same model. Evaluating the candidate expression on the observed inputs produces synthetic observations whose reference behavior is known. Conditioning the frozen generator on these observations yields a regenerated expression and its denoising trajectory. Comparing this trajectory with the candidate's original generation trajectory at corresponding timesteps gives a *self-regeneration trajectory deviation* across token positions and embedding channels. The candidate thus supplies a reference against which the generator's reconstruction behavior can be examined, even though the expression underlying the original observations is unknown. We use these trajectory deviations to construct candidate controls.

The rationale for reusing these trajectory deviations is that they record variations produced by the generator, with their temporal and cross-token structure intact. An endpoint-only deviation discards how those variations evolved through denoising; an isotropic perturbation does not use this recorded structure. Our central hypothesis is that self-regeneration trajectory deviations from related numerical conditions contain transferable information for proposing useful expression transitions. Applying deviation-derived controls during denoising lets the generator respond to these proposals before hard decoding. A reconstruction discrepancy does not uniquely specify the correction needed on the original task, so the choice of control remains a search problem.

This principle leads to a population-based search over diffusion trajectory states. Each expansion extracts controls from a parent's self-regenerations, assigns a donor independently to each recipient, samples a guidance-start index, and tests signed scales across fixed initial noises. Each controlled rollout is conditioned on the original observations, and the completed expression is evaluated on that regression task. Improving trajectories become new search states, with further expansions allocated to the current best state and a challenger drawn from structurally distinct alternatives. This closes the loop between model-generated proposals and numerical fitness: regeneration supplies variations to explore, while black-box selection determines their relevance to the target. The search leaves the pretrained model fixed and requires no differentiation through a surrogate expression objective.

Our contributions are:

- **Trajectory deviations as search objects.** We formulate inference-time search over temporally aligned trajectory deviations rather than final endpoint embeddings, allowing interventions at multiple denoising stages and subsequent nonlinear adaptation by the frozen generator. Controlled comparisons establish full trajectory deviation as substantially more effective than direct endpoint search.

- **Self-regeneration controlled evolution.** SR-CDE constructs deviations from a target-validated anchor and its self-regenerations, archives and assigns these proposals to compatible recipients, and tests them through signed, multi-scale DDIM rollouts. This yields iterative black-box trajectory search without generator retraining or a differentiable symbolic objective.

- **Strong recovery and mechanism evidence.** On 73 SRBench tasks, SR-CDE attains the best mean R^2, P10, average rank, and strict recovery count (64/73 at $R^2 > 0.999$) among the compared methods. On Hard-18, it improves all 18 tasks and outperforms endpoint-only transfer and norm- or covariance-matched random controls.

2 Related Work

Symbolic regression with learned proposals. Genetic programming searches directly over expression structures, commonly combining structural evolution with numerical parameter estimation (**??**). Neural symbolic regression introduces a learned prior over expressions: NeSymReS amortizes skeleton prediction, and E2E predicts structures and constants together (**??**). Search can subsequently exploit these priors rather than restart from an uninformed expression distribution. DGSR combines a generative model with Monte Carlo tree search, while TPSR uses a pretrained Transformer to guide planning over expression construction (**???**). These methods establish that target-task evaluation can improve learned proposals. Our focus is the information used to construct proposals: time-dependent responses extracted from a diffusion generator supplement the final fitness signal.

Latent-space search for expressions. GenSR trains a conditional variational autoencoder and searches a latent space designed for symbolic continuity and local numerical smoothness using modified CMA-ES (**??**). SR-CDE instead extracts directions from paired trajectories of an existing diffusion generator and evaluates them through the completed reverse process, requiring no separately trained search representation.

Diffusion-based symbolic regression. DiffuSR conditions continuous token-embedding diffusion on numerical observations and uses the generated logit matrix to guide genetic-programming subtree construction (**???**). MOD-SR integrates contrastive numerical conditioning and representation alignment with gradient-guided diffusion (**?**). Its differentiable expression relaxation and separately trained head route fitting-objective gradients into denoising. SR-CDE instead derives controls from self-regeneration trajectory deviations and selects them using the target fitness of decoded expressions.

Continuous generation and discrete decoding. Diffusion-LM established continuous intermediate representations as an interface for controllable text generation (**?**). Subsequent work studies their interaction with discrete outputs: RDLM incorporates categorical geometry into continuous diffusion, while CoDAR identifies token rounding as a bottleneck and learns a contextual decoder (**??**). These results motivate separating continuous generation dynamics from decoded-sample geometry. SR-CDE addresses the corresponding search question in symbolic regression: how to turn an existing reverse process into informative interventions under a discrete, numerically evaluated objective.

3 Problem Formulation

Symbolic regression as black-box expression search. Let $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$ denote the observed data, and let $\boldsymbol{\tau}$ be a token sequence representing an expression $f_{\boldsymbol{\tau}}$. We seek a valid expression that maximizes the black-box target fitness $J(f_{\boldsymbol{\tau}}; \mathcal{D})$ after constant optimization. In our experiments, this fitness is R^2; more generally, it may combine numerical fit and expression complexity without requiring derivatives with respect to tokens or intermediate embeddings. Constant fitting, validity checks, and complexity treatment belong to the expression evaluator and are held fixed when comparing search procedures.

Continuous states and expression regions. A pretrained conditional diffusion generator produces a trajectory of token states $\mathbf{z}_t \in \mathbb{R}^{L\times d}$, with L token positions and embedding dimension d. Using the forward loop index employed by our sampler, we write a reverse-process update as $\mathbf{z}_{t+1} = F_{\theta,t}(\mathbf{z}_t; \mathcal{D}, \boldsymbol{\xi}_t)$, where $\boldsymbol{\xi}_t$ collects any sampling randomness. Let Z_{term} be the terminal predicted-clean representation passed to the decoder, giving $\boldsymbol{\tau} = \mathrm{Dec}(Z_{\mathrm{term}})$. For a deterministic decoder, define the expression region

$$\mathcal{C}_{\boldsymbol{\tau}} = \{Z : \mathrm{Dec}(Z) = \boldsymbol{\tau}\}, \qquad J_{\mathrm{dec}}(Z; \mathcal{D}) = J(f_{\mathrm{Dec}(Z)}; \mathcal{D}). \tag{1}$$

If the evaluator depends only on the decoded expression and fixed data, J_{dec} has the same value everywhere in $\mathcal{C}_{\boldsymbol{\tau}}$. With continuous decoder logits and a strict argmax margin at every token position, an open neighborhood around Z retains the same decoded sequence. The fitness is consequently

locally constant in that neighborhood, even though Z is continuous. This is a property of hard decoding, independent of the optimizer.

Reachability depends on the reverse process. An endpoint region and the states that reach it need not have the same geometry. Let $\Phi_{\theta,t}(Z; \mathcal{D}, \boldsymbol{\xi})$ be the remaining reverse process from loop index t through the terminal readout Z_{term}, with future randomness $\boldsymbol{\xi}$ fixed. The preimage

$$\Omega_{\boldsymbol{\tau},t}(\mathcal{D}, \boldsymbol{\xi}) = \{Z : \text{Dec}(\Phi_{\theta,t}(Z; \mathcal{D}, \boldsymbol{\xi})) = \boldsymbol{\tau}\} \tag{2}$$

collects states that produce token sequence $\boldsymbol{\tau}$ under this continuation. We refer to this set as an expression basin at loop index t; it need not be connected. The index matters because subsequent denoising can attenuate, redirect, or amplify an intervention. A useful search proposal must therefore change the decoded outcome after continuation, rather than merely move far in embedding distance.

The search question. For an anchor expression $f_{\boldsymbol{\tau}_a}$, the relevant event is reaching a valid expression with higher target fitness under a specified proposal distribution and inference budget. The frequency of this event measures useful transitions; the fraction of proposals that change the decoded expression measures transitions alone. These quantities distinguish productive exploration from duplicate generation and from unhelpful structural changes. Our objective is to construct proposals from the frozen generator's own trajectories so that black-box search can exploit this distinction. The prevalence of high-quality basins is an empirical property of the model, task, and proposal distribution, separate from the decoder partition in Equation **??**.

4 METHOD

4.1 PROBLEM SETUP AND METHOD OVERVIEW

Given observations $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$, we seek an expression f maximizing $J(f) = R^2(f(\mathbf{x}), \mathbf{y})$ after numerical constant optimization. A frozen condition encoder maps $\mathcal{D}$ to $\mathbf{c}_\mathcal{D}$, and a frozen diffusion generator and decoder produce an expression through deterministic DDIM sampling (**??**). At reverse step t, the generator predicts $\widehat{\mathbf{x}}_{0,t} = G_\theta(\mathbf{z}_t, t, \mathbf{c})$; we denote the complete sequence of predicted-clean representations by $\mathcal{X} = (\widehat{\mathbf{x}}_{0,0}, \ldots, \widehat{\mathbf{x}}_{0,T-1})$.

Hard token decoding partitions continuous endpoints into symbolic basins: within-basin motion leaves fitness unchanged, whereas boundary crossings can abruptly change the expression. Endpoint fitness therefore ranks completed candidates but provides little information about when or how to alter the reverse process. A trajectory instead records successive clean-representation predictions and exposes an intervention point at each DDIM transition. If a desirable trajectory $\mathcal{X}^*$ were available, the stepwise error

$$\mathcal{E} = \mathcal{X}^* - \mathcal{X} \tag{3}$$

would provide a temporally aligned correction signal that endpoint fitness alone cannot supply. Injecting its components modifies the current transitions; the frozen denoiser then recomputes subsequent predictions from the altered latent states. Trajectory error thus connects final-solution quality to executable corrections of the generation process.

Because symbolic regression provides no target trajectory, *Self-Regeneration Controlled Diffusion Evolution* (SR-CDE) uses a target-validated trajectory as a realizable reference. Self-regeneration produces alternative paths under the behavioral condition induced by its expression; their deviations from the reference become reusable proposals. SR-CDE assigns these proposals to recipient contexts, tests them through complete controlled rollouts, and evolves the resulting trajectory states using target fitness (Figure **??**).

SR-CDE maintains dynamic *trajectory states*, which store target-validated expressions and the trajectories that generated them, and fixed *recipient contexts*, which provide stable generative contexts in which to test proposals. Expanding a state constructs its proposal archive, assigns a donor to each recipient, evaluates signed multi-scale rollouts, and retains parent-improving candidates through phenotype-aware selection.

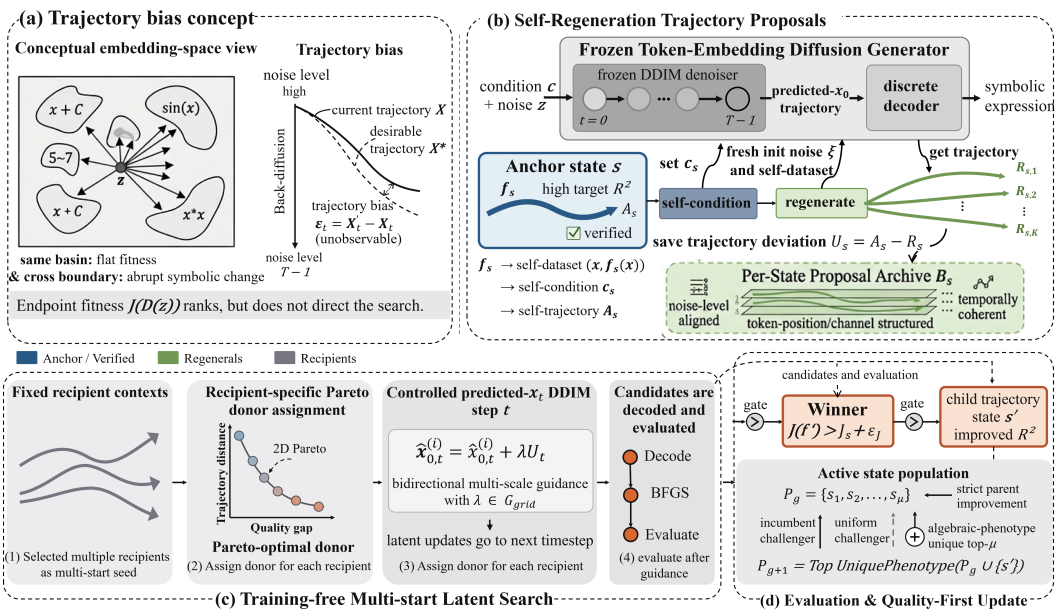


Figure 1: SR-CDE constructs trajectory-deviation proposals by self-regenerating a target-validated anchor, assigns them to fixed recipient contexts, and tests signed scales during DDIM denoising. Target evaluation of completed expressions updates the trajectory-state population.

4.2 Trajectory Proposals via Self-Regeneration

Trajectory states and self-conditioning. The search unit is a trajectory state $s = (f_s, J_s, \mathcal{A}_s, \mathbf{z}_s, \boldsymbol{\tau}_s)$, containing an optimized expression, its target fitness, the initial noise and token sequence, and its anchor trajectory $\mathcal{A}_s = \{\mathbf{A}_{s,t}\}_{t=0}^{T-1}$. To self-regenerate it, SR-CDE evaluates $\widetilde{y}_{s,n} = f_s(\mathbf{x}_n)$, encodes $\mathcal{D}_s = \{(\mathbf{x}_n, \widetilde{y}_{s,n})\}_{n=1}^{N}$, and runs unguided DDIM from independent noises under this self-condition. This yields alternative trajectories $\mathcal{R}_{s,j} = \{\mathbf{R}_{s,j,t}\}_{t=0}^{T-1}$ that share the anchor's behavioral condition but follow different denoising paths.

Temporal trajectory deviations. For the jth self-regeneration, SR-CDE defines the stepwise trajectory deviation

$$\mathbf{U}_{s,j,t} = \mathbf{A}_{s,t} - \mathbf{R}_{s,j,t}, \qquad t = 0, \ldots, T-1, \tag{4}$$

The proposal $\mathcal{U}_{s,j} = \{\mathbf{U}_{s,j,t}\}_{t=0}^{T-1}$ retains the generator's timestep-aligned, positional, channel-wise, and cross-token structure. It is therefore a temporally resolved proposal axis rather than an endpoint difference. Its reusability has two sources. Within a state, the anchor and self-condition remain fixed, so archived deviations remain comparable across later expansions. Across recipients, all trajectories share the predicted- $\mathbf{x}_0$ coordinates, timestep schedule, and model parameters, allowing $\mathbf{U}_{s,j,t}$ to modify the corresponding denoising stage. This shared coordinate system makes a proposal applicable, not universally effective: recipient assignment and complete rollouts determine its useful context, orientation, and scale.

Per-state proposal archives. Each state owns an archive $\mathcal{B}_s$ initialized on its first expansion and augmented on later visits. Archives remain tied to their anchors, and a child begins with an empty archive. Every entry stores the deviation, regeneration trajectory and expression, and its target fitness $J_{s,j}^{\mathrm{tar}}$ after constants fitted to the self-target are frozen. Capacities and update rules appear in Appendix **??**.

4.3 Recipient-Conditioned Donor Selection and Guidance

Fixed recipient contexts. SR-CDE fixes N_{rec} initial noises and their unguided target-condition trajectories $\mathcal{X}_i = \{\mathbf{X}_{i,t}\}_{t=0}^{T-1}$. This panel covers multiple basins and compares proposals in stable contexts. Recipients do not survive as states; an improving controlled rollout creates a new state while the panel remains fixed. Because basins respond differently to the same deviation, SR-CDE selects a donor independently for each recipient.

Pareto donor assignment. After filtering invalid and low-fitness regenerations into $\mathcal{B}_s^{\text{valid}}$, SR-CDE measures compatibility for guidance start t_0 by

$$d_{\text{traj}}(i, j \mid s, t_0) = \sum_{t=t_0}^{T-1} w_t \left\| \text{unit}(\mathbf{X}_{i,t}) - \text{unit}(\mathbf{R}_{s,j,t}) \right\|_2^2, \tag{5}$$

where $\text{unit}(\mathbf{v}) = \mathbf{v}/\max(\|\mathbf{v}\|_2, \epsilon)$ and $w_t \propto \sqrt{\bar{\alpha}_t}$ is normalized over $t \geq t_0$. This distance emphasizes trajectory orientation and shape, identifying regenerations whose paths are compatible with the recipient. A second objective, $d_{\text{gap}}(s, j) = |J_{s,j}^{\text{tar}} - J_s|$, favors regenerations whose decoded behavior retains the anchor's target quality. Following Pareto selection (**?**), the recipient-specific front is

$$\mathcal{F}_i(s) = \text{ParetoMin}_{j \in \mathcal{B}_s^{\text{valid}}} \left(d_{\text{traj}}(i, j \mid s, t_0), d_{\text{gap}}(s, j) \right) \tag{6}$$

SR-CDE samples j_i from this front, prioritizing donors not yet attempted in the current context. Appendix **??** gives fallback and bookkeeping rules.

Bidirectional multi-scale guidance. Each expansion samples a shared start t_0. Before t_0, recipients follow unguided DDIM; thereafter, donor j_i modifies each clean prediction by

$$\widetilde{\mathbf{x}}_{0,t}^{(i)} = \widehat{\mathbf{x}}_{0,t}^{(i)} + \lambda \mathbf{U}_{s,j_i,t}, \qquad \lambda \in \Lambda. \tag{7}$$

The DDIM noise term remains computed from the unmodified prediction:

$$\mathbf{z}_{t+1}^{(i)} = \sqrt{\bar{\alpha}_{t+1}} \widetilde{\mathbf{x}}_{0,t}^{(i)} + \sqrt{1 - \bar{\alpha}_{t+1}} \widehat{\boldsymbol{\epsilon}}_t^{(i)}. \tag{8}$$

Feeding the resulting latent back to the frozen denoiser lets subsequent steps respond nonlinearly. SR-CDE tests multiple positive and negative λ, treating each deviation as a bidirectional axis whose useful orientation is not fixed in advance, then completes decoding, constant optimization, and target evaluation. Scale and start-index ranges are specified in Appendix **??**.

4.4 Quality-First Trajectory-State Search

Initialization and parent allocation. Unguided target-condition generations provide the recipient panel and initial states. After removing duplicate algebraic phenotypes, SR-CDE retains the best μ states. Each round expands the best state and, when available, one uniformly sampled non-best challenger, balancing exploitation with alternative symbolic lineages. Phenotypes ignore numerical constants and canonicalize associative and commutative addition and multiplication, preventing equivalent forms from occupying multiple population slots.

Offspring generation and acceptance. For every selected parent, SR-CDE updates its archive and evaluates all assigned recipient–scale trials. The best trial becomes a child only when $J^\star > J_s + \varepsilon_J$. Its complete controlled trajectory is retained: the winning recipient supplies the generative context and the parent archive supplies the temporal proposal.

The child is then merged with the active population. SR-CDE first removes duplicate algebraic phenotypes and then retains the μ states with the highest target fitness:

$$\mathcal{P}_{g+1} = \text{Top}_\mu \left(\text{UniquePhenotype}(\mathcal{P}_g \cup \{s'\}), J \right). \tag{9}$$

This rule preserves phenotype diversity while remaining quality-first. A complete archive retains accepted states for final selection; only the active population receives further expansions. The archive also preserves each accepted state's complete controlled trajectory and parent–child provenance: the winning recipient becomes the child's generative base, while its temporal proposal remains attributable to the parent archive. Search returns the best archived expression upon reaching the fitness target or computational budget. Full pseudocode and hyperparameters are in Appendix **??**.

5 EXPERIMENTS

We evaluate whether self-regeneration trajectory deviations provide useful search directions for a frozen diffusion generator. The experiments establish three parts of this claim on a 73-task benchmark. We first measure final recovery quality, then isolate search behavior on tasks not already solved at initialization by comparing endpoint search with trajectory control. Finally, we replace model-generated deviations with randomized controls to determine which information in the direction produces the gain. Unless stated otherwise, all expressions undergo constant optimization and symbolic simplification.

5.1 EXPERIMENTAL SETUP

Benchmarks. The benchmark contains 73 SRBench tasks (**?**): Nguyen (12), Keijzer (12), Koza (2), Constant (8), Livermore (22), Vladislavleva (8), R (3), and Jin (6), each with 200 noiseless observations following MOD-SR. Initialization exceeds $R^2 = 0.99$ on 55 tasks; the remaining predefined *Hard-18* expose search differences. All Hard-18 methods share the same frozen noises, endpoints, and evaluations.

Baselines. The main benchmark compares SR-CDE with E2E, DiffuSR, MOD-SR, and GenSR under a common task-level protocol (**????**). Each method uses its published decoding and post-processing procedure. GenSR uses its official latent-space optimization, MOD-SR uses local perturbation with gradient guidance, and SR-CDE freezes the condition encoder, denoiser, decoder, and DDIM schedule throughout inference-time search.

Metrics and reporting protocol. We report mean R^2, the 10th-percentile R^2 (P10), expression complexity C, average rank, and strict $\mathrm{Success}@0.999$. SR-CDE runs for at most 50 rounds with a 1,200-second wall-clock cap. Appendix **??** provides evaluation-budget accounting, additional thresholds and baselines, and per-task results; the present experiments support recovery-quality rather than faster-inference claims.

5.2 OVERALL PERFORMANCE

SR-CDE recovers 64/73 tasks above $R^2 = 0.999$, six more than DiffuSR, 10 more than E2E, and 11 more than GenSR (Table **??**). It also leads in mean R^2, P10, and average rank; GenSR produces the least complex expressions.

Table 1: Results on the 73-task benchmark. Success uses the strict criterion $R^2 > 0.999$. Bold denotes the best result.

Method	Mean R^2	P10 R^2	Mean C	Avg. rank	S@.999
E2E	0.983922	0.980246	36.14	2.2397	54/73
DiffuSR	0.991754	0.987134	23.21	2.4385	58/73
MOD-SR	0.982267	0.956758	16.89	2.7397	43/73
GenSR	0.995848	0.990460	**16.37**	3.1233	53/73
SR-CDE	**0.996984**	**0.993003**	20.58	**1.8973**	**64/73**

5.3 ENDPOINT SEARCH VERSUS TRAJECTORY-LEVEL CONTROL

On Hard-18, GWO, Gaussian interpolation, diagonal CMA-ES, and DE/rand/1/bin search final predicted-$\mathbf{x}_0$ endpoints (**???**). Endpoint-only transfer applies the final regeneration–anchor deviation before decoding; SR-CDE applies its time-aligned sequence during reverse diffusion.

Table 2: Search-direction comparison on Hard-18 from a shared initialization with mean initial $R^2 = 0.911862$. "Best wins" counts per-task winners among the six methods, with ties split evenly.

Method	Mean final R^2	Mean ΔR^2	Improved	Best wins
GWO	0.912290	+0.000428	9/18	0
Gaussian interpolation	0.912290	+0.000428	9/18	0
Diagonal CMA-ES	0.912290	+0.000428	9/18	0
DE/rand/1/bin	0.954471	+0.042608	16/18	2
Endpoint-only deviation	0.958702	+0.046840	17/18	2
SR-CDE	**0.988849**	**+0.076987**	**18/18**	**14**

Local endpoint movements gain only 0.000428 mean R^2 (Table **??**). Population differences allow DE to cross more decoding boundaries, increasing the gain to 0.042608. Applying the model-generated deviation only at the endpoint performs similarly (0.046840), whereas retaining its temporal alignment raises the gain to 0.076987, improves all tasks, and wins 14/18. Effective search therefore requires both a proposal that induces useful symbolic transitions and an intervention early enough for the frozen denoiser to adapt the remaining generation path.

5.4 Information Carried by the Control Direction

To separate directional information from perturbation scale, timestep-RMS Gaussian matches each deviation's per-step norm, while covariance-matched random controls additionally preserve per-step empirical covariance. Both use the same recipients and multi-scale rollout protocol as SR-CDE.

Table 3: Control-direction ablation on Hard-18. Recipient-level statistics compare the best of eight signed scales with the corresponding unguided recipient; full definitions and sample counts are in Appendix **??**.

Direction	Mean final R^2	Mean ΔR^2	P10	Improve rate	Pos. gain
Timestep-RMS Gaussian	0.953313	+0.041451	0.905165	45.32%	0.144036
Covariance-matched	0.978121	+0.066259	0.961880	82.25%	0.380607
Trajectory deviation	**0.988849**	**+0.076987**	**0.966938**	**87.76%**	**0.426915**

Norm matching reaches a mean gain of 0.041451 and covariance matching reaches 0.066259, versus 0.076987 for real deviations (Table **??**). Preserving per-timestep covariance narrows the gap, showing that the proposal family carries useful temporal structure beyond isotropic noise. The realized trajectory deviation remains best in P10, recipient improvement rate, and positive gain, improving 87.76% of recipient applications versus 45.32% for norm-matched controls. The result cannot be explained by control energy alone: it depends both on temporal statistics and on the specific direction produced by self-regeneration.

6 Discussion

Generation dynamics as search information. Self-regeneration makes the frozen generator's response to candidate behavior observable. The resulting trajectory pairs construct structured interventions when the target expression and its embedding are unknown. This shifts the design question from choosing an arbitrary continuous neighborhood to reusing model-generated variations on which the reverse process can act. Diffusion provides the temporally resolved intervention interface, while expression evaluation supplies task-specific selection. The endpoint and random-control comparisons show that both roles are necessary.

Implications and scope. SR-CDE places inference-time adaptation in the search over controls rather than model parameters and therefore accommodates objectives defined only on executable expressions. Its practical scope is limited by the backbone's vocabulary and expression length, access to intermediate states, and the cost of regeneration and reverse-process evaluation. A natural

next question is how to allocate computation between collecting informative trajectory pairs and testing new controls. Resolving this trade-off would connect proposal quality to the budget required to obtain a useful symbolic model.

7 CONCLUSION

SR-CDE makes trajectory deviation, rather than endpoint displacement, the search object for diffusion symbolic regression. Self-regeneration constructs these directions and black-box controlled rollouts evolve them with a frozen generator. Across 73 tasks, SR-CDE attains the best mean and lower-tail R^2 and recovers 64/73 tasks at $R^2 > 0.999$; controlled studies further isolate the benefits of temporal application and self-regeneration-specific structure.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the design and refinement of the experimental plan, the selection of experimental parameters, and the implementation and execution of experimental workflows based on our existing codebase. Generative AI tools also assisted in constructing experiment scripts and scripts for statistical aggregation and analysis, interpreting experimental results, and improving the clarity and readability of the manuscript.

All AI-assisted work was reviewed and verified by the authors. In particular, the authors manually inspected the generated or modified code, validated the experimental configurations and execution procedures, and checked the statistical computations and reported values against the underlying raw experimental outputs. AI-assisted interpretations were independently assessed by the authors for consistency with the experimental evidence, and all AI-assisted manuscript revisions were reviewed for technical accuracy.

Generative AI tools did not fabricate, selectively alter, or directly manipulate any raw experimental measurements or results. All experimental decisions, analyses, interpretations, and conclusions were ultimately reviewed and approved by the authors. We take full responsibility for the final content of this work, including all text, claims, code, analyses, and other artifacts produced with the aid of generative AI.

REFERENCES

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

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. doi: 10.1109/4235.996017.

Xiaoxu Han, Chengzhen Ning, Jinghui Zhong, Fubiao Yang, Yu Wang, and Xin Mu. Discovering mathematical equations with diffusion language model. *arXiv preprint arXiv:2509.13136*, 2025. URL https://arxiv.org/abs/2509.13136.

Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001. doi: 10.1162/106365601750190398.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://arxiv.org/abs/2006.11239.

Jaehyeong Jo and Sung Ju Hwang. Continuous diffusion model for language modeling. In *Advances in Neural Information Processing Systems*, 2025.

Pierre-Alexandre Kamienny, Stéphane d'Ascoli, Guillaume Lample, and François Charton. End-to-end symbolic regression with transformers. In *Advances in Neural Information Processing Systems*, volume 35, pp. 10269–10281, 2022. URL https://arxiv.org/abs/2204.10532.

Pierre-Alexandre Kamienny, Guillaume Lample, Sylvain Lamprier, and Marco Virgolin. Deep generative symbolic regression with Monte-Carlo-Tree-Search. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 15655–15668. PMLR, 2023. URL https://proceedings.mlr.press/v202/kamienny23a.html.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*, volume 4212, pp. 282–293. Springer, 2006. doi: 10.1007/11871842_29.

John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.

William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabrício Olivetti de França, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason H. Moore. Contemporary symbolic regression methods and their relative performance. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://arxiv.org/abs/2107.14351.

Qian Li, Yuxiao Hu, Juncheng Liu, and Yuntian Chen. GenSR: Symbolic regression based on equation generative space. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.20557.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2205.14217.

Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61, 2014. doi: 10.1016/j.advengsoft.2013.12.007.

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

Junzhe Shen, Jieru Zhao, Ziwei He, and Zhouhan Lin. CoDAR: Continuous diffusion language models are more powerful than you think. *arXiv preprint arXiv:2603.02547*, 2026. URL https://arxiv.org/abs/2603.02547.

Parshin Shojaee, Kazem Meidani, Amir Barati Farimani, and Chandan Reddy. Transformer-based planning for symbolic regression. In *Advances in Neural Information Processing Systems*, volume 36, pp. 45907–45919, 2023. doi: 10.52202/075280-1990.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://arxiv.org/abs/2010.02502.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://arxiv.org/abs/2011.13456.

Rainer Storn and Kenneth Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997. doi: 10.1023/A:1008202821328.

Chuyang Xiang, Yichen Wei, and Junchi Yan. MOD-SR: Unifying multimodal learning and direct optimization with gradient-guided diffusion model for symbolic regression. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306. PMLR, 2026.

A Search Protocol and Implementation Details

A.1 Configuration

Table **??** specifies the reference search configuration. We write N_{rec} for the number of fixed recipient contexts and r_s for the number of fresh self-regenerations generated when state s is expanded. A sampler step denotes one predicted clean embedding. Thus, T clean predictions produce $T-1$ state transitions followed by terminal decoding. The guidance-start index t_0 is inclusive in this forward loop, rather than an index in the underlying diffusion-time convention. Model weights, the tokenizer, the decoder, and the diffusion schedule remain fixed throughout search.

Table 4: Reference search configuration. Fresh regenerations are appended to the per-state proposal archive associated with the expanded state.

Quantity	Symbol	Value
Initial target trajectories	N_0	50
Fixed recipient contexts	N_{rec}	10
Fresh regenerations	r_s	50 on first expansion; 10 thereafter
Clean predictions per rollout	T	50
Maximum search rounds	B	50
Active-state capacity	μ	5
Guidance-start index	t_0	Uniform integer in $[15, 45]$
Signed scales	Λ	$\{-1, -0.5, -0.25, -0.1, 0.1, 0.25, 0.5, 1\}$
Donor target-fitness floor	γ	0.8
Strict-improvement tolerance	ε_J	10^{-12}
Constant refinement		BFGS with two-point finite differences
Endpoint decoding		Greedy; no additional token samples

Initialization and fixed recipients. SR-CDE first generates and evaluates N_0 target-conditioned trajectories. The N_{rec} recipient contexts are fixed initial noises together with their unguided target-conditioned trajectories; they remain immutable throughout search and do not participate in survivor selection. Separately, valid initial trajectories are ranked by target fitness, deduplicated by algebraic phenotype, and used to initialize up to μ active trajectory states. Consequently, the recipient panel and active population have different algorithmic roles even when both originate from the same initialization batch. A controlled endpoint that improves its parent can become a new trajectory state, but it does not replace or update the recipient on which it was generated.

Randomness. Seeded random streams determine non-best challenger selection, donor sampling, and sampling of the guidance-start index t_0. The incumbent is always expanded; when at least one other active state exists, the challenger is sampled uniformly from the non-best active states. Every state owns its self-regeneration samples, and fresh samples produced on a revisit are appended to that state's existing per-state proposal archive.

Expression evaluation. Target rollouts fit constants to $\mathcal{D}_{\mathrm{fit}}$ and receive target fitness values on $\mathcal{D}$; self-regenerations instead fit constants to the parent's outputs at the same fitting inputs. Donor target fitness is computed by evaluating these frozen self-fitted constants against the target, without a second refitting step. The reference evaluator executes expressions with NumExpr and optimizes their constants by BFGS using SciPy's two-point finite-difference gradient. Expressions without constants bypass optimization, and unsuccessful refinement retains the decoded expression. Test observations do not participate in conditioning, constant fitting, donor selection, state acceptance, or stopping. Replay reuses the selected fitted expression and target fitness rather than repeating constant fitting.

Per-state proposal archives. Each trajectory state s owns a per-state proposal archive $\mathcal{B}_s$. Its first expansion generates 50 regenerations, and each revisit appends 10 fresh regenerations. Donor selection uses all accumulated valid entries in $\mathcal{B}_s$. A newly accepted child starts with an empty per-state proposal archive and therefore generates 50 regenerations when first expanded.

Algebraic phenotypes. Algebraic phenotypes are represented by constant-agnostic preorder token sequences. Addition and multiplication subtrees are flattened and their operands canonicalized so that associative and commutative reorderings share a phenotype key. These keys define structural deduplication in initialization and survivor selection. Normalized token Levenshtein distance is used when an auxiliary structure distance is reported, but it is not one of the two Pareto objectives used for donor assignment.

A.2 Recipient-Specific Donor Assignment

For each expanded state, donor assignment is performed independently for every recipient. Invalid regeneration entries are removed first. SR-CDE then retains valid donors with target fitness at least γ; if none meets this floor, it falls back to all valid entries in $\mathcal{B}_s$. Within the resulting pool, recipient i obtains the two-objective Pareto front defined in Section **??** by trajectory distance and the donor–anchor target-fitness gap.

To avoid immediate repetition, SR-CDE maintains an attempted-donor set $\mathcal{H}_{s,i}$ for each state–recipient context. It samples uniformly, with a seeded draw, from Pareto donors not currently in $\mathcal{H}_{s,i}$. After the selected donor has been evaluated at every scale in Λ, its identifier is recorded in $\mathcal{H}_{s,i}$. When the best target fitness observed for that state–recipient context improves, the set is cleared so that donors can be reconsidered relative to the new best target fitness. If the attempted set exhausts the current Pareto front before such an improvement, the exclusion cycle is reset for that recipient and sampling resumes from the full current front. This history is a temporary no-repeat mechanism for attempted applications, not an intrinsic success or failure label for a donor.

Tied target-fitness values retain the earlier scale in Table **??**, followed by the earlier recipient. Survivor ties within an algebraic phenotype retain the earlier state.

A.3 Search Procedure

The following procedure gives the execution order; each state expansion can produce at most one accepted child.

1. Generate and evaluate N_0 target-conditioned trajectories. Fix the N_{rec} recipient contexts. Initialize the trajectory archive and active population $\mathcal{P}$ with up to μ highest-fitness, phenotype-distinct valid trajectory states, and set a_{best} to the highest-fitness active state.
2. At the start of each round, select a_{best} and, when available, one uniformly sampled state from $\mathcal{P} \setminus \{a_{\mathrm{best}}\}$ as the fixed parent list.
3. For each parent s, generate $r_s = 50$ self-regenerations on its first expansion or $r_s = 10$ on a revisit, and append all valid proposals to $\mathcal{B}_s$. Sample a shared guidance-start index t_0, independently assign a Pareto donor j_i to each recipient, and evaluate every pair (i, λ) for $i \in \{1, \ldots, N_{\mathrm{rec}}\}$ and $\lambda \in \Lambda$.
4. Update each state–recipient attempted-donor history after the multi-scale evaluation. Let $J^\star$ be the highest target fitness among the endpoints produced by the $N_{\mathrm{rec}}|\Lambda|$ controlled rollouts. If $J^\star \leq J_s + \varepsilon_J$, continue to the next parent.
5. Otherwise, replay the winning scale batch to recover the complete winning controlled trajectory and form child state s'. If its trajectory is absent from the trajectory archive, record it, update a_{best} when appropriate, merge s' into $\mathcal{P}$, remove duplicate phenotypes, and retain the top-μ states by target fitness.
6. After all selected parents have been expanded, stop if the round budget is exhausted or an enabled target-fitness threshold is reached; otherwise begin the next round. Return $f_{a_{\mathrm{best}}}$.

A.4 Computation Accounting

An expansion generates r_s self-regenerations and evaluates $N_{\mathrm{rec}}|\Lambda| = 10 \times 8 = 80$ controlled rollouts, each producing one controlled endpoint. With exact reuse of the unguided prefix, its denoiser work before replay, measured in predicted-clean- embedding evaluations rather than batched calls, is

$$r_s T + N_{\mathrm{rec}}\big[t_0 + |\Lambda|(T - t_0)\big]. \tag{10}$$

The term $T - t_0$ counts controlled clean predictions at loop indices $t_0, \ldots, T-1$. Only $T - 1 - t_0$ of these predictions alter a subsequent state, because the prediction at index $T - 1$ is followed directly by terminal decoding. This distinction resolves the apparent off-by-one difference between prediction-count and transition-count accounting.

The one-time initialization cost is $N_0 T$. A strict-improvement trial incurs a winning-scale batch replay costing a further $N_{\mathrm{rec}}(T - t_0)$ clean-prediction evaluations when the prefix is cached. Thus, the 80 controlled rollouts per expansion exclude both self-regeneration and replay costs. A round normally contains two expansions—the incumbent and one non-best challenger—and therefore evaluates 160 controlled rollouts unless no challenger is available or the run stops between expansions.

Scales are evaluated sequentially at the fixed recipient batch size, while their decoded expressions are pooled for parallel constant fitting. Exact-token and data-context caches can reduce optimizer invocations, but do not remove self-regeneration, condition encoding, or constant fitting from end-to-end cost. For a run accepting M trajectory states, storing all clean-prediction sequences requires $O(MTLd)$ memory; the active population bounds challenger selection, not the size of the trajectory archive.

B Additional Experimental Details

B.1 Evaluation and Reporting Protocol

For every task, $\mathrm{Success@}\tau$ uses the strict criterion $R^2 > \tau$. Mean R^2 is the task-weighted arithmetic mean without clipping negative values, and P10 is the empirical 10th percentile over task-level target-fitness values. The 73-task benchmark (**?**) evaluates all predictions on the same fixed 200 points for every task and method. Average rank is computed only among methods with predictions for every task in the relevant comparison. Complexity is obtained from the available simplified expression tree; because external systems use different tokenizations, it should be interpreted as a broad structural comparison rather than an exact cross-system operation count.

Hard-18 is determined solely by the frozen initialization shared by all search methods. It contains the 18 tasks for which the best initialized endpoint has $R^2 < 0.99$, and its mean initial target fitness is 0.911862. No outcome from an endpoint optimizer, control ablation, or SR-CDE is used to construct the subset.

B.2 Full-Benchmark Endpoint Search

Table **??** reports the endpoint-search comparison on all 73 tasks. These runs use the same initialization and task protocol as the main benchmark; the Hard-18 analysis in the main text is a controlled subset selected before search to expose differences that are masked when most tasks are already solved at initialization.

Table 5: Endpoint-search comparison on the complete 73-task benchmark.

Method	S@.99	S@.999	Mean R^2	P10	Runtime (s)
GWO	55/73	37/73	0.977378	0.963687	39.51
Gaussian interpolation	55/73	38/73	0.977413	0.963687	40.13
DE/rand/1/bin	61/73	44/73	0.988081	0.979029	60.79
SR-CDE	**66/73**	**64/73**	**0.996984**	**0.993003**	203.01

B.3 Search Budget and Runtime

The endpoint baselines and trajectory search have different computational units, so runtime is reported for transparency rather than treated as a controlled speed comparison. On the 73-task benchmark, GWO, Gaussian interpolation, and diagonal CMA-ES use 4,000 endpoint evaluations per task; DE uses an average of 3,525. SR-CDE evaluates an average of 3,489 controlled rollouts on Hard-18; this count excludes self-regenerations and any winning-scale replay. Its mean runtime on the full 73-task benchmark is 203.01 seconds, compared with 39.51–60.79 seconds for the endpoint

baselines. This difference reflects the cost of complete controlled DDIM rollouts and regeneration rather than a claim of faster inference. All runs stop after at most 50 rounds or a 1,200-second wall-clock limit.

B.4 Recipient-Level Analysis

For the recipient-level statistics in Table **??**, one application unit fixes a task, round, parent state, and recipient. We retain the best result over the eight signed scales and compare it with the corresponding unguided recipient. The improvement rate is the fraction of units with a strictly better result; positive gain averages the improvement over successful units only. Methods produce different numbers of units because of early stopping, so rates and conditional gains are comparable, whereas raw counts do not represent identical run lengths.

The empirical-covariance control is constructed locally within each expansion. At each timestep, we estimate the empirical covariance across the real recipient controls produced in that expansion, draw a new direction from the resulting covariance model, and match its row-wise RMS to the corresponding real control. It therefore preserves the task- and expansion-specific direction-family statistics while randomizing the realized direction. The construction uses no target fitness, improvement indicator, or other outcome label.

Table 6: Recipient-level control statistics on Hard-18.

Direction	Units	Improved	Improve rate	Positive gain
Timestep-RMS Gaussian	16,820	7,622	45.32%	0.144036
Covariance-matched	7,820	6,432	82.25%	0.380607
Endpoint-only deviation	16,540	6,349	38.39%	0.262990
Trajectory deviation	7,850	6,889	**87.76%**	**0.426915**

B.5 Per-Task Results on Hard-18

Table **??** gives the final R^2 for every task used in the search-direction and control ablations. Rand denotes timestep-RMS Gaussian, Cov. denotes covariance-matched control, End denotes endpoint-only deviation, and Traj. denotes the complete SR-CDE trajectory deviation.

Table 7: Final task-level R^2 on Hard-18.

Task	Init.	GWO	Gauss	DE	CMA	End	Rand	Cov.	Traj.
Constant-2	.984872	.984872	.984872	.997573	.984872	.997573	.986231	.999999	1.000000
Constant-7	.980835	.980835	.980835	.982662	.980835	1.000000	.982662	1.000000	1.000000
Jin-5	.936376	.936376	.936376	1.000000	.936376	.999857	.998411	.999858	.999529
Keijzer-11	.955654	.955654	.955654	.955654	.955654	.961549	.955654	.980713	.961549
Keijzer-13	.980757	.980757	.980757	.980757	.980757	.980757	.980757	1.000000	1.000000
Keijzer-14	.950638	.950638	.950638	.968519	.950638	.950638	.950638	.971266	.972921
Keijzer-15	.974494	.974494	.974494	.974494	.974494	.982361	.974494	.999238	.999239
Livermore-11	.113995	.115241	.115241	.634891	.115241	.654650	.654650	.763919	.971524
Livermore-17	.973723	.973723	.973723	1.000000	.973723	.977593	.997808	1.000000	.999968
Livermore-18	.978597	.978597	.978597	.978597	.978597	.993793	.993793	.999837	.999837
Livermore-3	.963214	.963214	.963214	.982459	.963214	.982459	.982459	.997027	.983929
Livermore-5	.983745	.983761	.983761	.983763	.983761	.999571	.999246	.999571	.999246
Nguyen-10	.965579	.965579	.965579	1.000000	.965579	.993341	.996593	1.000000	1.000000
Nguyen-12	.989788	.989788	.989788	.990571	.989788	.990147	.990000	.999585	.999585
Nguyen-5	.985356	.985357	.985357	.997244	.985357	.997478	.997244	.999970	.999970
Vladislavleva-1	.911905	.911906	.911906	.925974	.911906	.918092	.925925	.949744	.966472
Vladislavleva-6	.870740	.870740	.870740	.874954	.870740	.918421	.874954	.978371	.978371
Vladislavleva-8	.913252	.919691	.919691	.952359	.919691	.958361	.918113	.967082	.967138