SYMMETRY- AND SEMANTIC-AWARE SYMBOLIC DISCOVERY OF ORDINARY DIFFERENTIAL EQUATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recovering symbolic ordinary differential equations (ODEs) from trajectories requires solutions that are both expressive and consistent with known physics. Library-based sparse regression methods provide structural guarantees only within predefined invariant/equivariant bases, while free-form symbolic regression explores many symmetry-violating expressions and obtains behavioral feedback mainly from completed candidates. From this perspective, we develop Symmetry- and Semantic-Aware Graph Symbolic Regression (S^2GSR) by searching over canonical, symmetry-typed expression graphs equipped with sub-expression-level semantics. The local typing rules of expression graphs guarantee the prescribed symmetry of the discovered ODEs by construction, and path-conditioned semantic features compare closed sub-expressions with position-specific behavioral targets. This framework makes invariant-basis methods a special case of S^2GSR while supporting *free-form, non-polynomial* dynamics, and further integrates path-conditioned semantic guidance into a neural-guided Monte Carlo Tree Search pipeline. Experiments on polynomial and non-polynomial symmetry-constrained systems and on ODEBench demonstrate improved symbolic recovery and trajectory reconstruction of S^2GSR while maintaining physical consistency.

1 INTRODUCTION

Discovering governing ordinary differential equations (ODEs) from observed trajectories is a central problem in scientific machine learning, with applications spanning physics, biology, chemistry, materials science, and control. Given measurements of a dynamical state $\mathbf{x}(t)$, the goal is to recover an interpretable vector field $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ that explains the observed evolution while enabling extrapolation, simulation, and mechanistic insight beyond the training data.

Existing approaches to symbolic ODE discovery fall into two broad families. Library-based sparse regression methods, such as SINDy and its variants (Brunton et al., 2016; Messenger & Bortz, 2021; Both et al., 2021), assume a predefined dictionary of candidate functions and identify a sparse combination that best fits the dynamics; these methods are computationally efficient when the correct basis is known. In contrast, free-form symbolic regression searches over expressions built from variables, constants, and operators, offering greater expressivity without being limited to a fixed library (Schmidt & Lipson, 2009; Kronberger et al., 2019; Udrescu & Tegmark, 2020; Makke & Chawla, 2024). Recent neural-guided and search-based techniques further accelerate this process using transformers, Monte Carlo Tree Search (MCTS), and learned priors (d'Ascoli et al., 2024; Shojaee et al., 2023; Xiang et al., 2025).

Despite these advances, both families encounter fundamental limitations in expressive, physics-constrained settings. Consider a planar system with state (x_1, x_2) and rotational symmetry whose vector field has non-polynomial radial dependence,

$$\dot{x}_1 = -x_2\, g(x_1^2 + x_2^2), \quad \dot{x}_2 = x_1\, g(x_1^2 + x_2^2), \quad g(r) = \sin(r)\, e^{-r}. \tag{1}$$

Here, $r = x_1^2 + x_2^2$ denotes the squared radial coordinate. Library-based methods cannot exactly represent the radial factor $\sin(r)e^{-r}$; recovering it would require manually including the relevant non-polynomial and composed invariant features. Symmetry-aware sparse regression (Yang et al., 2024; 2026) can enforce equivariance, but its expressivity is still tied to the chosen invariant/equivariant library or basis. Free-form symbolic regression is sufficiently expressive, but its search space includes

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.

rotation-violating candidates, such as $(-x_2,\, x_1^2)$, that can fit short trajectory segments well. Existing syntactic search cannot exclude them because physical validity is imposed only as a soft reward term. Together, these limitations reveal that the central challenge is not merely the size of the search space, but how candidates are represented, constrained, and guided during construction. Algebraically equivalent expressions inflate search paths, syntax-only guidance delays behavioral feedback until complete candidates, and physical constraints are imposed softly or tied to finite invariant bases.

Motivated by these challenges, we recast symbolic ODE discovery as a neural-guided search over canonical *symmetry-typed expression graphs* equipped with *sub-expression-level semantics*. This formulation makes physical validity a hard constraint on the search space rather than a soft penalty and allows behavioral evidence to guide each local expansion rather than only complete candidates. We instantiate this perspective as **Symmetry- and Semantic-Aware Graph Symbolic Regression** (**S^{2}GSR**), a neural-MCTS framework for ODE discovery. Our contributions are three-fold:

- **(i) Symmetry by construction with a global guarantee.** Local typing rules over signatures (ρ, κ, η) encode group, scaling, and time-reversal behavior, provably enforcing the prescribed symmetry of the discovered vector field (Theorem 3.2). This construction makes invariant-basis methods a special case of S^{2}GSR while supporting free-form non-polynomial dynamics.
- **(ii) Path-conditioned sub-expression semantics.** Each closed sub-expression is compared against the locally inverted behavioral target at its position in the graph, providing slot-aware functional guidance during top-down search rather than only at termination.
- **(iii) A unified neural-MCTS pipeline.** A graph-transformer integrates joint reasoning over symbolic structure, sub-expression semantics, and path-conditioned alignment into neural-MCTS with a clean separation between hard validity constraints and soft behavioral priors.

Experiments on polynomial and non-polynomial symmetry-constrained systems demonstrate improved symbolic recovery, verify the complementary contributions of each component, and support validation-based selection among candidate symmetries. On ODEBench with trivial symmetry, S^{2}GSR achieves robust trajectory reconstruction with competitive complexity and inference time.

2 Related Work

Library-based and free-form equation discovery. Equation-discovery methods either select terms from fixed libraries or construct free-form expressions. Library-based approaches include SINDy and WSINDy (Brunton et al., 2016; Messenger & Bortz, 2021), with neural variants such as DeepMoD and SINDy-Autoencoder (Both et al., 2021; Champion et al., 2019). Free-form approaches employ genetic programming (Koza, 1994; Mundhenk et al., 2021; Cranmer, 2023), differentiable learners (Martius & Lampert, 2016; Sahoo et al., 2018), reinforcement learning (Petersen et al., 2021; Landajuela et al., 2022), and pretrained transformers (Biggio et al., 2021; Kamienny et al., 2022). ODE-specific methods include ODEFormer and D-CODE (d'Ascoli et al., 2024; Qian et al., 2022), while SPL, TPSR, and GSR incorporate MCTS, transformer priors, and graph search (Sun et al., 2023; Shojaee et al., 2023; Xiang et al., 2025). These two families expose a central tension: fixed libraries limit expressivity, whereas free-form search admits symmetry-incompatible candidates.

Physics-constrained equation discovery. Physics priors have been imposed through unit-consistent generation, constrained architectures, and symmetry-adapted libraries. PhySO masks unit-inconsistent tokens (Tenachi et al., 2023), dimensionless machine learning enforces scaling equivariance (Villar et al., 2023), and Hamiltonian and Lagrangian networks encode conservation but return non-symbolic models (Greydanus et al., 2019; Cranmer et al., 2020a). For symbolic dynamical systems, symmetry-informed methods construct invariant/equivariant bases (Yang et al., 2024; 2026), restricting discovery to combinations of prespecified generators and libraries. Existing work therefore does not enforce prescribed linear symmetries by construction within a free-form symbolic search space.

Semantic and behavioral guidance. Semantic search methods use functional behavior rather than syntax alone to guide symbolic discovery. Geometric semantic genetic programming (Moraglio et al., 2012; Vanneschi et al., 2014) and competent geometric operators (Pawlak & Krawiec, 2018) compare complete expressions through semantic distances to guide population-level variation. However, these methods do not assess whether a partial sub-expression matches the behavioral target of its specific operand slot, leaving top-down expansion without position-aware semantic guidance.

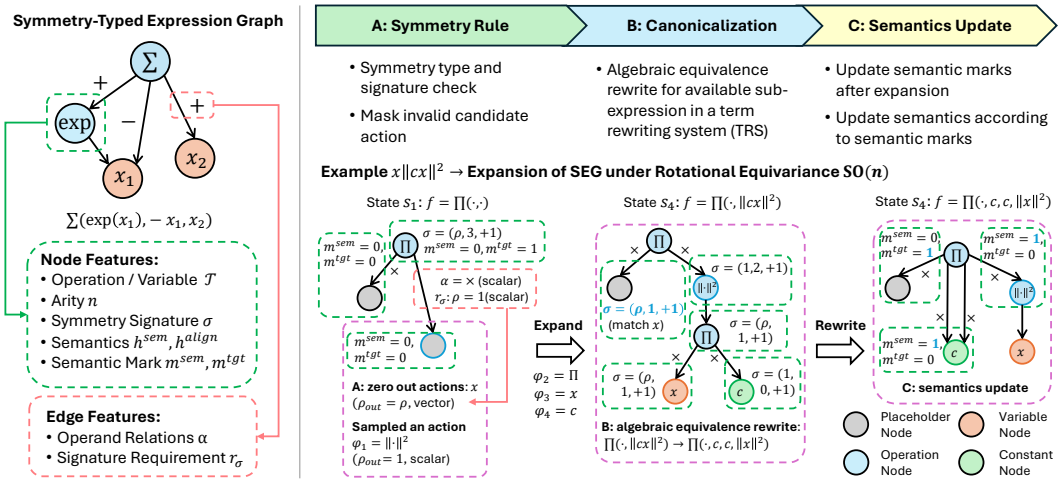


Figure 1: SEG representation and local expansion workflow. (a) A Symmetry-Typed Expression Graph augments an expression graph with node features for syntax, symmetry signature, masked semantics, and edge features for operand roles. (b) Each expansion masks symmetry-invalid actions, canonicalizes the accepted transition, and updates semantics for newly closed sub-expressions, thereby preserving symmetry validity, algebraic compactness, and semantic guidance during the search.

3 METHOD

In this paper, we introduce S^{2}GSR, an ODE discovery framework that unifies search over canonical symmetry-typed expression graphs (SEGs) with path-conditioned semantic guidance in a neural-MCTS pipeline. This integration enables free-form, symmetry-aware symbolic discovery with algebraic canonicalization across polynomial and non-polynomial dynamics, and provides dense position-aware behavioral guidance on sub-expressions during top-down symbolic search.

3.1 SYMBOLIC ODE DISCOVERY AND EQUIVARIANT VECTOR FIELDS

We consider the problem of discovering governing ODEs from observed data $\mathcal{D} = \{\mathbf{x}(t_i)\}_{i=1}^N$, consisting of discrete-time measurements of the system state $\mathbf{x}(t) \in \mathbb{R}^d$, following the dynamics:

$$\dot{\mathbf{x}}(t) \;=\; \frac{d\mathbf{x}(t)}{dt} \;=\; \mathbf{f}(\mathbf{x}(t)), \tag{2}$$

where $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$ is an unknown vector field admitting a closed-form symbolic representation. We seek $\hat{\mathbf{f}} = (\hat{f}^{(1)}, \ldots, \hat{f}^{(d)})$ by solving

$$\hat{\mathbf{f}} \;=\; \arg\min_{\mathbf{f} \in \mathcal{H}^d} \; \mathcal{L}(\mathbf{f}; \mathcal{D}) + \lambda \mathcal{C}(\mathbf{f}), \tag{3}$$

where each component $\hat{f}^{(k)}$ belongs to a hypothesis space $\mathcal{H}$ generated by a grammar $\mathcal{G} = (\mathcal{T}, \Phi)$ with terminal symbols $\mathcal{T}$ (state variables and constants) and mathematical operators Φ (e.g., $+, -, \times, /$ and elementary nonlinear functions). The loss $\mathcal{L}(\mathbf{f}; \mathcal{D})$ measures data-fitting error ((24) in Appendix D.2), and $\mathcal{C}(\mathbf{f})$ penalizes expression complexity to encourage parsimony.

Equivariant ODEs. Physical systems often obey rotational, reflection, scaling, or time-reversal symmetries, which impose corresponding transformation laws on the vector field $\mathbf{f}$. For a symmetry group G, our goal is to recover $\hat{\mathbf{f}}$ that solves (3) while satisfying the equivariance condition in (4).

Definition 3.1 (Group action and equivariant vector fields)**.** Let $X \subseteq \mathbb{R}^d$ denote the state space and G a Lie group acting linearly on X through $F : G \times X \to X,\ (g, x) \mapsto g \cdot x = \rho_X(g)x$, with tangent map $DF_g(x) = \rho_X(g)$ (Appendix B.5). A vector field $\mathbf{f} : X \to \mathbb{R}^d$ is *G-equivariant* if

$$\mathbf{f}(g \cdot x) \;=\; DF_g(x)\, \mathbf{f}(x), \qquad \forall\, g \in G,\ x \in X. \tag{4}$$

Table 1: Components of the SEG symmetry signature $\sigma(v) = (\rho_v, \kappa_v, \eta_v)$ carried by each node v. f_v denotes a sub-expression induced by the subgraph of v and its descendants. $\mathrm{GL}(\mathcal{V}_v)$ denotes the general linear group of $\mathcal{V}_v$, i.e., the group of invertible linear maps from $\mathcal{V}_v$ to itself. F_t denotes the time-reversal action on state variables and $F_{t,\mathcal{V}_v}$ denotes the corresponding action on the value space.

Component	Domain	Meaning	Transformation law
ρ_v	$G \to \mathrm{GL}(\mathcal{V}_v)$	Group representation	$f_v(g \cdot x) = \rho_v(g) f_v(x)$
κ_v	$\mathbb{R}$	Scaling degree	$f_v(\lambda x) = \lambda^{\kappa_v} f_v(x)$
η_v	$\{+1, -1\}$	Time-reversal parity	$f_v(F_t x) = \eta_v F_{t,\mathcal{V}_v} f_v(x)$

Table 2: Exemplar of SEG typing rules. u and v denote different input child nodes, $\mathbf{1}$ denotes the trivial scalar representation, and ρ denotes a nontrivial representation such as a vector representation.

Operator	Input signatures	Output signature	Validity condition
$\times$ scalar-scalar	$(\mathbf{1}, \kappa_u, \eta_u), (\mathbf{1}, \kappa_v, \eta_v)$	$(\mathbf{1}, \kappa_u + \kappa_v, \eta_u \eta_v)$	both inputs scalar
$\times$ scalar-vector	$(\mathbf{1}, \kappa_u, \eta_u), (\rho, \kappa_v, \eta_v)$	$(\rho, \kappa_u + \kappa_v, \eta_u \eta_v)$	first input scalar
$\lVert \cdot \rVert^2$ norm squared	(ρ, κ, η)	$(\mathbf{1}, 2\kappa, +1)$	ρ orthogonal

3.2 Symmetry-typed Hypothesis Space

The goal of S^{2}GSR is to generate symbolic vector fields that satisfy (4) by construction via *free-form* and symmetry-typed SEG hypothesis space, rather than enforcing this condition through *fixed-form* polynomial construction (Yang et al., 2024) or *post hoc* filtering/regularization. (Appendix B.2)

Symmetry-typed expression graph (SEG). Specifically, an SEG is a directed acyclic expression graph (Xiang et al., 2025) whose nodes v represent terminals or operators. The rooted subgraph induced by v and its descendants forms a sub-expression $f_v : X \to \mathcal{V}_v$, mapping from the state space X to the value space $\mathcal{V}_v$ of the sub-expression. Node v is assigned a symmetry signature

$$\sigma(v) = (\rho_v, \kappa_v, \eta_v).$$

Table 1 defines the detailed symmetry signature. The representation component ρ_v generalizes the scalar/vector distinction. A scalar invariant has the trivial representation $\rho_v(g) = 1$, while a vector-valued quantity transforms under a nontrivial representation, such as the standard vector representation for rotations. Figure 1(a) illustrates the structure of an SEG.

Typed free-form closure grammar. Let $\Sigma = (G, F, \mathcal{T}_\Sigma, \mathcal{G}_\Sigma, \sigma(v_{\mathrm{root}}))$ denote the symmetry specification, where $\mathcal{T}_\Sigma$ is a set of typed terminals (including state variables, constants and motifs), and $\mathcal{G}_\Sigma$ is a grammar of symbolic operators equipped with typing rules. The SEG hypothesis space is defined recursively as the *free-form* closure $\mathcal{H}_{\mathrm{SEG}}(\Sigma) = \mathrm{Cl}_{\mathcal{G}_\Sigma}(\mathcal{T}_\Sigma)$, where $\mathrm{Cl}_{\mathcal{G}_\Sigma}$ denotes all SEGs obtainable from typed terminals by repeatedly applying type-valid grammar operators. Concretely, an m-ary operator $\phi : \mathcal{V}_1 \times \cdots \times \mathcal{V}_m \to \mathcal{V}_{\mathrm{out}}$ is admissible for input signatures $\sigma(u_1), \ldots, \sigma(u_m)$ and output signature $\sigma(v)$ only if it is an equivariant intertwiner:

$$\phi(\rho_{u_1}(g) z_1, \ldots, \rho_{u_m}(g) z_m) = \rho_v(g) \phi(z_1, \ldots, z_m), \qquad \forall g \in G. \tag{5}$$

Here, z_i denotes an arbitrary value in the value space of the child node u_i. Table 2 is an example SEG typing rules for the operator-specific propagation (full table in Appendix B.1). For example, the norm-squared operator maps a vector input under an orthogonal ρ representation to a scalar invariant $\mathbf{1}$. Figure 1(b) illustrates such an SEG expansion under the closure grammar in Table 2.

Local admissibility yields global symmetry. The most significant and direct benefit of the symmetry-pruned SEG hypothesis space $\mathcal{H}_{\mathrm{SEG}}(\Sigma)$ lies in the fact that it preserves the *free-form* nature of SR while establishing the symmetry guarantee by construction (permitting arbitrary symbolic compositions whenever the local composition is type-sound under the closure grammar). Theorem 3.2 formalizes this guarantee, implying local type-valid SEG expansion yields global symmetry validity.

Theorem 3.2 (Local-to-global symmetry preservation)**.** *Let G act on X, and let every terminal in $\mathcal{T}_\Sigma$ satisfy its assigned transformation law. Assume every operator admitted by $\mathcal{G}_\Sigma$ satisfies the intertwining condition in (5) and obeys the declared scaling and time-reversal propagation rules. Then every complete well-typed SEG generated by the typed expansion rules (Table 6) denotes an expression whose root node satisfies its assigned signature. In particular, if the root node is r with*

signature $\sigma(r) = (\rho_X, \kappa_r, \eta_r)$ corresponding to the tangent representation of the ODE state, then the generated root expression is a G-equivariant vector field satisfying (4).

Proof sketch. Proceed by structural induction on SEGs. Terminals satisfy their signatures by definition. For an internal node, assume each child satisfies its signature. Since ϕ is admitted only when its typing rule holds, the intertwining condition implies f_v satisfies $\sigma(v)$. The same holds for scaling and parity. Applying this to the root yields the claimed symmetry property; choosing the root signature as the tangent action gives ODE equivariance. Appendix B.4 provides the full proof.

3.3 Path-conditioned and Sub-expression Semantics

Conventional symbolic regression relies on syntax-only representations or semantics of complete expressions: the former poorly reflect functional behavior, while the latter cannot guide position-specific expansion of partial sub-expressions. We therefore equip each sub-expression with semantic and path-conditioned alignment descriptors for dense, slot-aware guidance during top-down search.

Sub-expression semantic embedding. Let $\mathcal{P} = \{x(t_{i_1}), \ldots, x(t_{i_M})\}$ be a fixed probe set sampled only from the training trajectories, where $M < N$. For a closed node v corresponding to a fully specified sub-expression $f_v(x)$, we define its raw semantic vector as

$$S_v = \left[f_v(x(t_{i_1})), \ldots, f_v(x(t_{i_M}))\right]^\top \in \mathbb{R}^M. \tag{6}$$

We normalize the semantic vector as $\tilde{S}_v$ and record numerical quality q_v in Appendix C.1. Here, a node is *closed* if the sub-expression f_v contains no unresolved placeholder, otherwise it is *open* with semantic values undefined. We therefore introduce a binary semantic availability mask $m_v^{\text{sem}} = \mathbb{I}[v \text{ is closed}]$. The *sub-expression semantic embedding* is then

$$h_v^{\text{sem}} = \begin{cases} E_{\text{sem}}\left(\tilde{S}_v \,\|\, q_v\right), & m_v^{\text{sem}} = 1, \\ u_{\sigma(v)}^{\text{sem}}, & m_v^{\text{sem}} = 0, \end{cases} \tag{7}$$

where E_{sem} is a multilayer perceptron (MLP) encoder and $u_{\sigma(v)}^{\text{sem}}$ is a learnable signature-indexed embedding used when semantic evaluation is undefined. This mask-aware formulation distinguishes, e.g., unresolved scalar-invariant nodes from resolved vector-equivariant nodes.

Path-conditioned alignment embedding. The semantic vector $\tilde{S}_v$ describes what f_v is, but not whether its behavior fits the slot v would occupy in the expression. Effective sub-expression-level guidance requires a slot-specific target. For the j-th ODE component, we associate each node u with a *path-conditioned derivative target* $d_u^{(j)} \in \mathbb{R}^M$, defined by inverting the operators above u:

$$d_u^{(j)} = \begin{cases} \widehat{x}_j, & u = \text{root}, \\ \phi_v^{-1}\big(d_v^{(j)}; \{f_k\}_{k \in \text{sib}(u)}\big), & u \neq \text{root} \ \wedge \ m_u^{\text{tgt}} = 1, \\ \text{undefined}, & u \neq \text{root} \ \wedge \ m_u^{\text{tgt}} = 0, \end{cases} \tag{8}$$

where v denotes the parent of u, $\text{sib}(u)$ the siblings of u under ϕ_v, and ϕ_v^{-1} the local inverse of ϕ_v in u's input slot given the sibling values (Appendix C.2). The *target-availability mask* $m_u^{\text{tgt}} = \mathbb{I}[m_k^{\text{sem}} = 1 \ \forall k \in \text{sib}(u) \ \wedge \ m_v^{\text{tgt}} = 1]$ flags whether the recursion is well-posed at u. For a closed scalar node v with $m_v^{\text{sem}} = m_v^{\text{tgt}} = 1$, the path-conditioned alignment feature is defined in (9), with $\tilde{S}_v$ and $\tilde{d}_v^{(j)}$ the normalized semantic vector and normalized derivative target on $\mathcal{P}$.

$$A_v^{(j)} = \frac{\langle \tilde{S}_v, \tilde{d}_v^{(j)} \rangle}{\|\tilde{S}_v\|_2 \, \|\tilde{d}_v^{(j)}\|_2 + \epsilon}. \tag{9}$$

$$h_v^{\text{align}} = \begin{cases} E_{\text{align}}\big(A_v^{(j)}\big), & m_v^{\text{sem}} = m_v^{\text{tgt}} = 1, \\ u_{\sigma(v)}^{\text{align}}, & \text{otherwise}. \end{cases} \tag{10}$$

For vector-valued sub-expressions, alignment is computed componentwise and concatenated. The *path-conditioned alignment embedding* is defined in (10), where E_{align} is an MLP encoder and $u_{\sigma(v)}^{\text{align}}$ is a signature-indexed learnable embedding for the unavailable case. Rather than naively comparing with the global residual, this embedding provides a dense, position-aware search signal indicating whether a closed sub-expression, in its current graph position, is likely to explain the remaining dynamics, before a complete equation is generated.

3.4 SYMMETRY-PRUNED AND SEMANTIC-GUIDED MCTS

S^{2}GSR performs symbolic ODE discovery by running hybrid neural-guided Monte Carlo Tree Search (hnMCTS; Xiang et al., 2025) over SEGs. The typing rules define a hard admissible action space, while a graph transformer encoder (Ying et al., 2021) provides soft policy and value guidance over this symmetry-pruned space. The two mechanisms are complementary: typing enforces validity by construction, and the encoder learns to prioritize among the remaining valid actions using SEG structure, sub-expression semantics, and path-conditioned alignment.

Symmetry-pruned admissible action set. Each MCTS state s is a canonical SEG. Let $\mathcal{F}(s)$ denote the set of open frontier nodes. An action $a = (v, \varphi)$ expands an open node $v \in \mathcal{F}(s)$ using an operator or terminal $\varphi \in \mathcal{G}$. Theorem 3.2 for SEG typing directly induces the admissible action set

$$\mathcal{A}_{\mathrm{SEG}}(s) = \{(v, \varphi) : v \in \mathcal{F}(s),\ \varphi \in \mathcal{G},\ \mathrm{TypeCheck}(s, v, \varphi) = 1\}. \tag{11}$$

Here, $\mathrm{TypeCheck}(s, v, \varphi)$ verifies that the operator input signatures, output signature, and parent-required signature are compatible under the SEG typing rules (Table 6). The transition function only permits actions in $\mathcal{A}_{\mathrm{SEG}}(s)$, so symmetry-incompatible branches are removed before tree traversal rather than penalized after completion. Proposition D.1 establishes the validity of the action mask.

Graph-transformer joint guidance. For each state s, the encoder consumes initial node features that combine SEG structure, sub-expression semantics, and path-conditioned alignment into three additive streams:

$$h_v^{(0)} = h_v^{\mathrm{struct}} + h_v^{\mathrm{sem}} + h_v^{\mathrm{align}}, \qquad h_v^{\mathrm{struct}} = h_{\mathrm{syn}}(v) + h_{\mathrm{sig}}(\sigma(v)), \tag{12}$$

where $h_{\mathrm{syn}}(v)$ encodes operator identity, arity, and operand role; $h_{\mathrm{sig}}(\sigma(v))$ encodes the SEG signature; h_v^{sem} is the sub-expression semantic embedding (7); and h_v^{align} is the path-conditioned alignment embedding (10). Each stream gates itself through its own availability mask, so the encoder receives well-formed input regardless of which mix of m_v^{sem} and m_v^{tgt} holds at each node.

The graph transformer maps these initial features to contextual node embeddings $\{h_v\}_{v \in V}$ and a pooled graph embedding h_s. The policy and value heads are

$$P_\theta(a = (v, \varphi) \mid s) = \mathrm{softmax}_{a \in \mathcal{A}_{\mathrm{SEG}}(s)} \big(\mathrm{MLP}_\pi(h_v, h_s, \mathrm{emb}(\varphi))\big), \tag{13}$$

$$V_\theta(s) = \mathrm{MLP}_V(h_s), \tag{14}$$

where the policy softmax is restricted to the admissible action set $\mathcal{A}_{\mathrm{SEG}}(s)$. Semantic and alignment information thus influence search only through learned policy and value estimates, while algebraic canonicalization and SEG typing remain responsible for hard validity constraints. This separation of concerns ensures that the encoder's soft predictions never violate symmetry or algebraic canonicality.

S^{2}GSR pipeline. Following GSR (Xiang et al., 2025), S^{2}GSR uses an ϵ-UCB policy that combines unbiased UCT exploration with predictor-based PUCT exploitation. At each visited state s, a random variable $\epsilon \sim U(0, 1)$ selects between the two:

$$a = \begin{cases} \arg\max_{a \in \mathcal{A}_{\mathrm{SEG}}(s)} \mathrm{UCT}(s, a), & \epsilon < c_\epsilon, \\ \arg\max_{a \in \mathcal{A}_{\mathrm{SEG}}(s)} \mathrm{PUCT}(s, a; P_\theta), & \epsilon \geq c_\epsilon, \end{cases} \tag{15}$$

where c_ϵ controls the probability of using the unbiased UCT branch. The UCT branch maintains exploration when the neural prior is under-trained, while PUCT exploits the encoder's prior over symmetry-valid expansions. Full forms of both rules are given in (22) of Appendix D.2. Each simulation combines SEG-valid canonical expansion, semantic updates for newly closed sub-expressions, leaf evaluation, and return backup. Algorithm 1 in Appendix D.3 summarizes the complete procedure.

4 EXPERIMENTS

To evaluate the complementary strengths of symmetry typing and semantic guidance in S^{2}GSR, we conduct experiments to address four questions in the order of each subsection: (i) Can S^{2}GSR recover ground-truth ODEs beyond the tested polynomial libraries while satisfying prescribed symmetry constraints? (ii) How does each component contribute to its performance? (iii) Can validation data select a useful symmetry specification when the symmetry is not known in advance? (iv) Does S^{2}GSR improve reconstruction on general ODEs under a trivial symmetry specification?

Table 3: Symbolic recovery rate (RR) and trajectory reconstruction on polynomial (P-) and non-polynomial (NP-) systems over 100 seeds. Reconstruction values are mean $R^2 \pm$ the half-width of a 95% confidence interval (CI); RR is the proportion of seeds recovering the complete ground-truth vector field. Stars denote symmetry-aware methods. Recovery intervals are provided in Appendix E.2.

System	**P-SO(2)** (16)		**P-SE** (17)		**NP-SO(2)** (1)		**NP-TR** (18)	
Method	**RR**↑	$\mathbf{R^2}$ ↑	**RR**↑	$\mathbf{R^2}$ ↑	**RR**↑	$\mathbf{R^2}$ ↑	**RR**↑	$\mathbf{R^2}$ ↑
SINDy	0.13	0.83 ± 0.05	0.05	0.85 ± 0.04	0.00	0.69 ± 0.06	0.00	0.84 ± 0.06
WSINDy	0.00	0.76 ± 0.06	0.00	0.81 ± 0.06	0.00	0.66 ± 0.09	0.00	0.85 ± 0.07
PySR	0.02	0.82 ± 0.04	0.03	0.85 ± 0.05	0.00	0.78 ± 0.06	0.03	0.88 ± 0.05
D-CODE	0.00	0.74 ± 0.08	0.00	0.78 ± 0.07	0.00	0.68 ± 0.12	0.02	0.80 ± 0.07
uDSR	0.08	0.79 ± 0.06	0.04	0.84 ± 0.05	0.00	0.75 ± 0.09	0.05	0.84 ± 0.06
ODEFormer	0.02	0.81 ± 0.03	0.02	0.86 ± 0.03	0.00	0.79 ± 0.05	0.03	$\underline{0.91 \pm 0.03}$
GSR	0.15	0.82 ± 0.04	0.10	0.86 ± 0.05	$\underline{0.05}$	$\underline{0.81 \pm 0.06}$	$\underline{0.12}$	0.90 ± 0.04
EquivSINDy-c*	$\underline{0.89}$	$\underline{0.86 \pm 0.03}$	**1.00**	$\mathbf{0.91 \pm 0.03}$	0.00	0.71 ± 0.10	0.00	0.85 ± 0.05
S²GSR*	**0.96**	$\mathbf{0.87 \pm 0.02}$	**1.00**	$\mathbf{0.91 \pm 0.03}$	**0.36**	$\mathbf{0.85 \pm 0.08}$	**0.58**	$\mathbf{0.93 \pm 0.03}$

4.1 Polynomial and Non-polynomial Symmetry-constrained Discovery

Systems and protocol. We study four symmetry-constrained systems: the polynomial (P-) systems from Yang et al. (2024) with damped oscillator (P-SO(2))(16) of SO(2) fields equivariant under planar rotations, and the growth system (P-SE)(17) obeying the weighted scaling equivariance $(x_1, x_2) \mapsto (\lambda^2 x_1, \lambda x_2)$; the non-polynomial (NP-) systems with the rotational field (NP-SO(2))(1) of the same SO(2) equivariance, and a time reversible system (NP-TR)(18) under $(t, x_1, x_2) \mapsto (-t, x_1, -x_2)$.

$$\begin{aligned} \dot{x}_1 &= -0.1x_1 - x_2, \\ \dot{x}_2 &= x_1 - 0.1x_2. \end{aligned} \tag{16}$$

$$\begin{aligned} \dot{x}_1 &= -0.3x_1 + 0.1x_2^2, \\ \dot{x}_2 &= x_2. \end{aligned} \tag{17}$$

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\sin(x_1)/(1 + x_1^2). \end{aligned} \tag{18}$$

We compare against PySR (Cranmer, 2023), D-CODE (Qian et al., 2022), SINDy (Brunton et al., 2016), WSINDy (Messenger & Bortz, 2021), uDSR (Landajuela et al., 2022), ODEFormer (d'Ascoli et al., 2024), GSR (Xiang et al., 2025), and EquivSINDy-c (Yang et al., 2024) which enforces symmetry within a specified library. We report the symbolic recovery rate (RR), defined as the fraction of seeds recovering the complete ground-truth vector field, separately from mean trajectory reconstruction R^2 with 95% confidence intervals. Each experiment uses 100 seeds and Gaussian-process smoothing, with 20% additive noise for P-SO(2) and 5% multiplicative Gaussian noise for the other systems. Appendix E.1 gives the per-system settings.

Polynomial and non-polynomial recovery. Table 3 reveals a clear contrast between the polynomial and non-polynomial systems. On the polynomial systems, S²GSR and EquivSINDy-c achieve substantially higher RR than the other methods: 0.96 and 0.89, respectively, on P-SO(2), and 1.00 for both on P-SE. These results highlight the benefit of explicitly enforcing the prescribed symmetries. On the more complex non-polynomial systems, S²GSR retains the highest recovery rates, reaching 0.36 on NP-SO(2) and 0.58 on NP-TR, compared with 0.05 and 0.12 for GSR and zero for EquivSINDy-c. S²GSR also achieves mean reconstruction R^2 values of 0.85 and 0.93, respectively. Although EquivSINDy-c attains reconstruction scores of 0.71 and 0.85, its tested polynomial libraries exclude the nonlinear terms and compositions required for exact recovery. The contrast illustrates how the free-form typed grammar of S²GSR combines symmetry constraints with the expressivity needed to discover these non-polynomial dynamics.

4.2 Contributions of Symmetry Typing and Semantic Guidance

Ablation settings. To disentangle the effects of hard symmetry constraints and semantic search guidance, we perform controlled ablations of the two mechanisms in S²GSR. Symmetry typing is disabled by replacing the prescribed group with the trivial group, while the graph representation, algebraic canonicalization, and search procedure remain unchanged. Semantic information is then introduced incrementally, from structural guidance alone to sub-expression semantics and path-conditioned alignment. All variants use the same other settings, as detailed in Appendix E.2.

Complementary contributions. Table 4 highlights complementary roles: symmetry typing enforces admissibility, while semantic guidance prioritizes expressions by behavior. Excluding symmetry-incompatible expansions raises structural-only RR from 0.15 to 0.89 on P-SO(2) and from

Table 4: Symbolic recovery rate (RR) of discovering ground-truth ODEs over 100 seeds on P-SO(2) and NP-SO(2). Rows toggle symmetry typing; columns compare structural guidance (None), sub-expression semantics (Sem-only), and semantics with path-conditioned alignment (Sem+Align).

System	**P-SO(2)** (16)			**NP-SO(2)** (1)		
Symmetry typing	None	Sem-only	Sem+Align	None	Sem-only	Sem+Align
Active	0.89	0.92	**0.96**	0.22	0.28	**0.36**
Relaxed	0.15	0.17	0.21	0.06	0.09	0.18

Table 5: Symmetry selection by validation reconstruction. Entries are mean validation $R^2 \pm$ the half-width of a 95% CI over 100 seeds. Bold entries correspond to the ground-truth symmetry.

Candidate Symmetry	**System**			
	P-SO(2) (16)	**P-SE** (17)	**NP-SO(2)** (1)	**NP-TR** (18)
SO(2)	$\mathbf{0.87 \pm 0.02}$	0.64 ± 0.06	$\mathbf{0.85 \pm 0.06}$	0.58 ± 0.07
TR	0.62 ± 0.05	0.59 ± 0.05	$\mathbf{0.82 \pm 0.06}$	$\mathbf{0.94 \pm 0.02}$
SE	0.56 ± 0.08	$\mathbf{0.90 \pm 0.03}$	0.52 ± 0.10	0.57 ± 0.05

0.06 to 0.22 on NP-SO(2). Sub-expression semantics describes closed partial expressions, while path-conditioned alignment, when available, assesses their fit to position-specific targets. Adding these signals within the typed space increases RR from 0.22 to 0.28 and then 0.36 on NP-SO(2), versus 0.89, 0.92, and 0.96 on P-SO(2). Full semantic guidance also raises NP-SO(2) RR under relaxed typing from 0.06 to 0.18, but does not enforce symmetry. Thus, hard admissibility constraints and behavioral guidance complement each other, with larger semantic gains on the complicated task.

4.3 Selecting a Symmetry Specification

Model selection. When symmetry itself is an object of inquiry rather than prior knowledge, choosing a symmetry specification can be formulated as model selection based on validation reconstruction. Table 5 illustrates this approach using the same four systems and experimental settings as Table 3. We compare three candidate specifications defined in Section 4.1: planar rotational symmetry (SO(2)), time-reversal symmetry (TR), and weighted scaling equivariance (SE). Discovery runs across candidates use equal budget, and the fitted model with the highest validation reconstruction is selected.

Symmetry specification. Table 5 supports validation reconstruction as a criterion for selecting useful symmetry-constrained models: every highest-scoring candidate corresponds to a symmetry of the ground-truth system. On P-SO(2), P-SE, and NP-TR, the corresponding SO(2), SE, and TR specifications achieve mean validation R^2 values of 0.87, 0.90, and 0.94, respectively, exceeding the incompatible candidates. NP-SO(2) illustrates that valid specifications need not be unique: both SO(2) and TR are exact symmetries and yield high scores (0.85 and 0.82), whereas SE yields 0.52. Thus, validation favors symmetry constraints that support accurate reconstruction without necessarily identifying a unique symmetry. Appendix E.3 provides the validation protocol and further analysis.

4.4 General ODE Reconstruction

Dataset, configuration, and metric. To assess the robustness, complexity, and efficiency of S^2GSR under different noise, sparse sampling, and dimensionality settings, and evaluate whether S^2GSR can still improve general ODE reconstruction without prescribed physical symmetries, we follow the ODEBench from (d'Ascoli et al., 2024). The benchmark contains 63 one- to four-dimensional systems with polynomial, rational, trigonometric, exponential, and coupled nonlinear dynamics. S^2GSR uses trivial symmetry throughout (no selected system-specific symmetry). Each method infers a vector field from a single observed trajectory under multiplicative Gaussian noise $\sigma \in \{0, 0.02, 0.05\}$ and random removal of a fraction $\rho \in \{0, 0.5\}$ of time points. We compare against GSR (Xiang et al., 2025), ODEFormer (d'Ascoli et al., 2024), SINDy (Brunton et al., 2016), WSINDy (Messenger & Bortz, 2021), PySR (Cranmer, 2023), and uDSR (Landajuela et al., 2022). Reconstruction accuracy is evaluated by the percentage of systems whose integrated prediction achieves $R^2 > 0.9$ against the clean reference trajectory. Appendix E.4 provides further details.

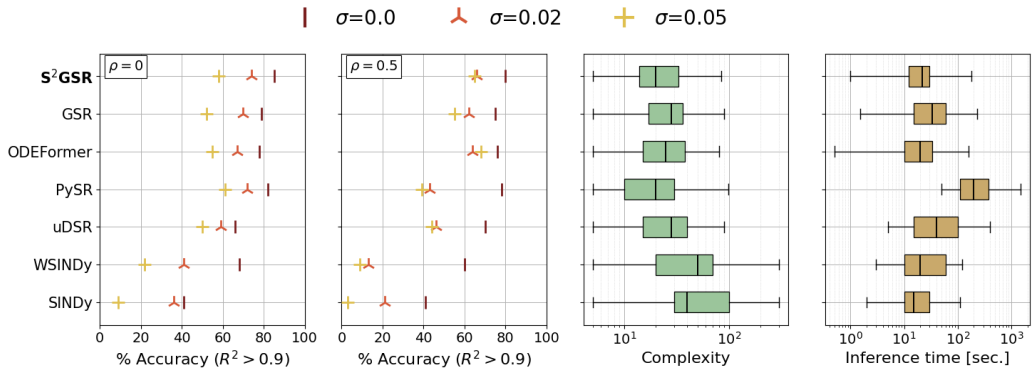


Figure 2: General ODEBench reconstruction with trivial symmetry for S^{2}GSR. The first two panels report the percentage of systems with trajectory $R^2 > 0.9$ under noise $\sigma \in \{0, 0.02, 0.05\}$ and subsampling $\rho \in \{0, 0.5\}$. The remaining panels show expression complexity and wall-clock inference time in seconds, on logarithmic axes. The accuracy markers distinguish noise levels.

Reconstruction, complexity, and time. Figure 2 demonstrates that S^{2}GSR improves general ODE reconstruction across the tested noise and sampling conditions while maintaining competitive expression complexity and inference time. On this collection of 63 ODE systems, S^{2}GSR outperforms GSR in reconstruction accuracy in all six conditions, highlighting the benefit of sub-expression semantics and path-conditioned guidance even without prescribed physical symmetries. Its accuracy is competitive with the strongest free-form baselines under both dense and sparse sampling, and significantly outperforms library-based SINDy methods. These reconstruction gains over GSR come with expression complexity comparable to other free-form methods, including PySR. S^{2}GSR also achieves a lower median inference time than GSR and a substantially lower median than PySR, with a runtime distribution overlapping that of ODEFormer. Together, these results establish a favorable balance of reconstruction accuracy, symbolic compactness, and inference cost, supporting semantic guidance as an effective approach to ODE discovery from noisy and incomplete observations.

5 Conclusion and Limitation

We introduced **S^{2}GSR**, an ODE discovery framework that unifies algebraically canonical, symmetry-typed expression graph representations with path-conditioned sub-expression semantics in a neural-MCTS pipeline. Local typing rules guarantee the prescribed symmetry of complete expressions, enforcing physical validity by construction while retaining free-form expressivity for non-polynomial dynamics beyond the fixed invariant-basis constructions. Path-conditioned semantic alignment further provides dense, slot-aware guidance for sub-expression expansion during top-down search.

Experiments validate S^{2}GSR across complementary settings: (i) symmetry-constrained benchmarks demonstrate substantial symbolic recovery gains, particularly on non-polynomial dynamics; (ii) ablations confirm the complementary benefits of symmetry typing and sub-expression semantics; (iii) when symmetry is unknown, validation reconstruction favors a valid candidate symmetry specification; and (iv) on ODEBench with trivial symmetry, S^{2}GSR remains competitive with the strongest free-form baselines while maintaining comparable expression complexity and favorable inference time. These results show that hard symmetry constraints and dense behavioral guidance improve symbolic ODE discovery without sacrificing free-form expressivity.

Limitations and future work. S^{2}GSR requires a symmetry specification derived from domain knowledge, validation-based selection, or data-driven tools (Kreider et al., 2025) for symmetry-constrained search. A misspecified symmetry may exclude the true vector field. Moreover, the current guarantee applies to linear group actions; affine and nonlinear actions require extended typing rules (Appendix B.5). The framework targets equation discovery rather than the identification of a unique or maximal symmetry group. Path-conditioned alignment reverts to signature-indexed guidance when a path contains non-invertible operators. Future work includes jointly inferring symmetries and equations, extending SEG typing to affine and nonlinear actions, improving alignment for non-invertible compositions, and incorporating learned coordinates for latent-space discovery (Champion et al., 2019; Cranmer et al., 2020b).

AI USE STATEMENT

In this work, we used generative AI tools for generating the synthetic datasets for the two non-polynomial benchmark systems, NP-SO(2) (1) and NP-TR (18). We have not used generative AI tools for interpreting results, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for or writing proofs, proposing or refining hypotheses, or designing research methodology or experiments, and the rest of the required disclosure tasks, including translation, dataset cleaning or reformatting, and qualitative or thematic data analysis, are not applicable to this work. Additionally, we used generative AI tools for drafting and editing portions of the paper, improving readability, suggesting manuscript structure, identifying potentially relevant literature, and editing portions of the research code. We have reviewed all AI-assisted work. The synthetic datasets and associated code were checked against the defining ODEs and numerical protocols; AI-assisted code was reviewed and tested by the authors; suggested literature was independently retrieved, read, and verified; and all edited passages were manually reviewed for technical accuracy, consistency, and appropriate attribution. We take responsibility for the final content of this work, including text, claims or 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 *International Conference on Machine Learning*, pp. 936–945. PMLR, 2021.

Gert-Jan Both, Subham Choudhury, Pierre Sens, and Remy Kusters. DeepMoD: Deep learning for model discovery in noisy data. *Journal of Computational Physics*, 428:109985, 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, 2016.

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.

Miles Cranmer. Interpretable machine learning for science with PySR and SymbolicRegression.jl. *arXiv preprint arXiv:2305.01582*, 2023.

Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020a.

Miles Cranmer, Alvaro Sanchez-Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17429–17442, 2020b.

Stéphane d'Ascoli, Sören Becker, Philippe Schwaller, Alexander Mathis, and Niki Kilbertus. ODE-Former: Symbolic regression of dynamical systems with transformers. In *The Twelfth International Conference on Learning Representations*, 2024.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

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.

John R Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, 4:87–112, 1994.

Max Kreider, John Harlim, and Daning Huang. A model-free method for discovering symmetry in differential equations. *arXiv preprint arXiv:2511.09779*, 2025.

Gabriel Kronberger, Lukas Kammerer, and Michael Kommenda. Identification of dynamical systems using symbolic regression. In *International Conference on Computer Aided Systems Theory*, pp. 370–377. Springer, 2019.

Mikel Landajuela, Chak Shing 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 *Advances in Neural Information Processing Systems*, volume 35, pp. 33985–33998, 2022.

Nour Makke and Sanjay Chawla. Interpretable scientific discovery with symbolic regression: a review. *Artificial Intelligence Review*, 57(1):2, 2024.

Georg Martius and Christoph H Lampert. Extrapolation and learning equations. *arXiv preprint arXiv:1610.02995*, 2016.

Daniel A Messenger and David M Bortz. Weak SINDy: Galerkin-based data-driven model selection. *Multiscale Modeling & Simulation*, 19(3):1474–1497, 2021.

Alberto Moraglio, Krzysztof Krawiec, and Colin G Johnson. Geometric semantic genetic programming. In *International Conference on Parallel Problem Solving from Nature*, pp. 21–31. Springer, 2012.

T Nathan Mundhenk, Mikel Landajuela, Ruben Glatt, Claudio P Santiago, Daniel M Faissol, and Brenden K Petersen. Symbolic regression via neural-guided genetic programming population seeding. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24912–24923, 2021.

Tomasz P Pawlak and Krzysztof Krawiec. Competent geometric semantic genetic programming for symbolic regression and boolean function synthesis. *Evolutionary Computation*, 26(2):177–212, 2018.

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. In *International Conference on Learning Representations*, 2021.

Zhaozhi Qian, Krzysztof Kacprzyk, and Mihaela van der Schaar. D-CODE: Discovering closed-form ODEs from observed trajectories. In *International Conference on Learning Representations*, 2022.

Subham Sahoo, Christoph Lampert, and Georg Martius. Learning equations for extrapolation and control. In *International Conference on Machine Learning*, pp. 4442–4450. PMLR, 2018.

Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9323–9332. PMLR, 18–24 Jul 2021.

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

Parshin Shojaee, Kazem Meidani, Amir Barati Farimani, and Chandan K. Reddy. Transformer-based planning for symbolic regression. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Fangzheng Sun, Yang Liu, Jian-Xun Wang, and Hao Sun. Symbolic physics learner: Discovering governing equations via Monte Carlo tree search. In *International Conference on Learning Representations*, 2023.

Wassim Tenachi, Rodrigo Ibata, and Foivos I Diakogiannis. Deep symbolic regression for physics guided by units constraints: Toward the automated discovery of physical laws. *The Astrophysical Journal*, 959(2):99, 2023.

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

Leonardo Vanneschi, Mauro Castelli, and Sara Silva. A survey of semantic methods in genetic programming. *Genetic Programming and Evolvable Machines*, 15(2):195–214, 2014.

Soledad Villar, David W Hogg, Kate Storey-Fisher, Weichi Yao, and Ben Blum-Smith. Scalars are universal: Equivariant machine learning, structured like classical physics. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 28848–28863. Curran Associates, Inc., 2021.

Soledad Villar, Weichi Yao, David W Hogg, Ben Blum-Smith, and Bianca Dumitrascu. Dimensionless machine learning: Imposing exact units equivariance. *Journal of Machine Learning Research*, 24 (109):1–32, 2023.

Ziyu Xiang, Kenna Ashen, Xiaofeng Qian, and Xiaoning Qian. Graph-based symbolic regression with invariance and constraint encoding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jianke Yang, Wang Rao, Nima Dehmamy, Robin Walters, and Rose Yu. Symmetry-informed governing equation discovery. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Jianke Yang, Manu Bhat, Bryan Hu, Yadi Cao, Nima Dehmamy, Robin Walters, and Rose Yu. Discovering symbolic differential equations with symmetry invariants. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. Featured Certification, J2C Certification.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.

APPENDIX

A EXPRESSION GENERATION AND GRAPH REPRESENTATION

MDP formulation of expression generation. We frame symbolic expression generation as a Markov decision process $(\mathcal{S}, \mathcal{A}, T, \mathcal{R})$, following prior neural-guided search frameworks (Sun et al., 2023; Xiang et al., 2025). Each state $s \in \mathcal{S}$ corresponds to a partially constructed expression, and an action $a \in \mathcal{A}(s)$ applies a grammar rule from $\mathcal{G}$, with the deterministic transition $T(s, a)$ producing the resulting partial expression. A terminal state is reached when the expression is complete and valid. The reward $\mathcal{R}$ derives from the negative SR objective in (3). We solve this MDP via Monte Carlo Tree Search (MCTS), which incrementally expands candidate expressions guided by visit-count statistics and learned priors.

Expression graphs and algebraic canonicalization. We represent each MDP state s as an *expression graph* (EG; Xiang et al., 2025), a directed acyclic graph whose nodes denote operators or terminals and whose edges encode functional composition. Unlike conventional tree-based representations (Makke & Chawla, 2024), EGs reuse shared terminals for compact storage and admit richer structural and relational information through node and edge features — a property we will exploit in Section 3.3 to encode symmetry signatures and sub-expression semantics. Algebraic redundancy is reduced by a term-rewriting system (TRS) that maps algebraically equivalent expressions to a shared canonical form, ensuring that syntactically distinct but algebraically identical expressions collapse to the same graph during search.

B SEG REPRESENTATION DETAILS

B.1 SEG TYPING RULES

Table 6 details the signature-propagation rules used to determine whether an SEG expansion is admissible. Each rule maps the group representation, scaling degree, and time-reversal parity of the child nodes to the parent signature, allowing symmetry-invalid actions to be masked before expansion.

Table 6: Representative SEG local admissibility typing rules to mask invalid graph expansions during search. Here u and v denote different input child nodes, $\mathbf{1}$ denotes the trivial scalar representation, and ρ denotes a nontrivial representation such as a vector representation.

Operator	Input signatures	Output signature	Validity condition
$+, -$	$(\rho, \kappa, \eta), (\rho, \kappa, \eta)$	(ρ, κ, η)	same signatures
$\times$ scalar-scalar	$(\mathbf{1}, \kappa_u, \eta_u), (\mathbf{1}, \kappa_v, \eta_v)$	$(\mathbf{1}, \kappa_u + \kappa_v, \eta_u\eta_v)$	both inputs scalar
$\times$ scalar-vector	$(\mathbf{1}, \kappa_u, \eta_u), (\rho, \kappa_v, \eta_v)$	$(\rho, \kappa_u + \kappa_v, \eta_u\eta_v)$	first input scalar
dot product	$(\rho, \kappa_u, \eta_u), (\rho, \kappa_v, \eta_v)$	$(\mathbf{1}, \kappa_u + \kappa_v, \eta_u\eta_v)$	ρ orthogonal
$\div$ scalar/scalar	$(\mathbf{1}, \kappa_u, \eta_u), (\mathbf{1}, \kappa_v, \eta_v)$	$(\mathbf{1}, \kappa_u - \kappa_v, \eta_u\eta_v)$	both inputs scalar
$\div$ vector/scalar	$(\rho, \kappa_u, \eta_u), (\mathbf{1}, \kappa_v, \eta_v)$	$(\rho, \kappa_u - \kappa_v, \eta_u\eta_v)$	denominator scalar
$\|\cdot\|^2$ norm squared	(ρ, κ, η)	$(\mathbf{1}, 2\kappa, +1)$	ρ orthogonal
exp	$(\mathbf{1}, 0, +1)$	$(\mathbf{1}, 0, +1)$	dimensionless scalar
log	$(\mathbf{1}, 0, +1)$	$(\mathbf{1}, 0, +1)$	dimensionless scalar
sin	$(\mathbf{1}, 0, \eta)$	$(\mathbf{1}, 0, \eta)$	dimensionless scalar
cos	$(\mathbf{1}, 0, \eta)$	$(\mathbf{1}, 0, +1)$	dimensionless scalar
power z^n	$(\mathbf{1}, \kappa, \eta)$	$(\mathbf{1}, n\kappa, \eta^n)$	scalar input, $n \in \mathbb{Z}$

B.2 RELATION TO INVARIANT-BASIS METHODS.

The typed closure contains the invariant/equivariant basis construction commonly used in symmetry-aware discovery, under the condition stated in Proposition B.1. In the special case of polynomial equivariant vector fields under compact or reductive group actions, classical invariant theory implies that polynomial invariants admit finite generators, and equivariant polynomial maps can often be represented as a module over the invariant ring. In such settings, an equivariant vector field can be

written in the form

$$f(x) = \sum_{k=1}^{K} p_k(I_1(x), \ldots, I_r(x)) B_k(x), \tag{19}$$

where $I_1, \ldots, I_r$ are invariant scalar generators, $B_1, \ldots, B_K$ are equivariant basis fields, and p_k are scalar polynomial functions.

In S^{2}GSR, (19) is not the definition of the search space. Its relation to the SEG hypothesis space is stated precisely as follows.

Proposition B.1 (Containment)**.** *If every generator I_j and B_k in (19) lies in $\mathrm{Cl}_{\mathcal{G}_\Sigma}(\mathcal{T}_\Sigma)$, then every field of the form (19) lies in $\mathcal{H}_{\mathrm{SEG}}(\Sigma)$. The inclusion is strict whenever $\mathcal{G}_\Sigma$ admits a non-polynomial scalar operator.*

Proof. Each p_k is built from the scalar-typed I_j by the $+$, $\times$, and z^n rules of Table 6; each B_k is in the closure by assumption; the scalar–vector $\times$ and $+$ rules assemble the sum. The first field below lies in $\mathcal{H}_{\mathrm{SEG}}(\Sigma)$ but is not of the form (19) with polynomial p_k.

The condition is met by generic rules for contraction invariants: $\|x\|^2$ and dot products are derived from vector-typed terminals. For $O(n)$ acting on vectors, the first fundamental theorem of invariant theory states that polynomial invariants are generated by inner products and equivariant polynomial maps by the vectors themselves over this ring (Villar et al., 2021), so the condition holds with Table 6 alone. For $SO(n)$, generators involving the Levi-Civita tensor require one group-specific typed operator, e.g., the planar J (R_{90}) for SO(2) (Appendix E.1), which is supplied once per group rather than per system. Beyond this containment, SEG constructs non-polynomial and nested equivariant expressions such as:

$$f(x) = \sin(\|x\|^2) \exp(-\|x\|^2) x$$

or

$$f(x) = \frac{x}{1 + \|x\|^2},$$

both can be naturally generated by composing invariant scalar sub-expressions with equivariant vector-valued sub-expressions through typed operators. Therefore, SEG preserves the free-form construction ability of EG while adding representation-theoretic guarantees.

B.3 Algebraic Canonicalization versus Symmetry Typing.

SEG uses two complementary mechanisms. Algebraic redundancy is handled by a term rewriting system (TRS) based canonicalization, as similarly adopted in EG (Xiang et al., 2025), which maps algebraically equivalent expressions to a shared graph representation. Symmetry validity, in contrast, is enforced by typed expansion and action masking. Thus, S^{2}GSR does not require global canonicalization over continuous group orbits. When closed-form invariantization or orbit representatives are available, they can be added as optional typed operators or canonicalization rules; however, the core soundness guarantee of SEG follows from local typed closure rather than global orbit canonicalization.

B.4 Proof of Theorem 3.2

Proof. We prove the result by structural induction over the SEG. For a terminal node v, the claim holds by the terminal soundness assumption. For an internal node v with children $u_1, \ldots, u_m$ and operator

$$f_v(x) = \phi(f_{u_1}(x), \ldots, f_{u_m}(x)),$$

assume by induction that each child satisfies

$$f_{u_i}(g \cdot x) = \rho_{u_i}(g) f_{u_i}(x).$$

Then

$$\begin{aligned} f_v(g \cdot x) &= \phi(f_{u_1}(g \cdot x), \ldots, f_{u_m}(g \cdot x)) \\ &= \phi(\rho_{u_1}(g) f_{u_1}(x), \ldots, \rho_{u_m}(g) f_{u_m}(x)) \\ &= \rho_v(g) \phi(f_{u_1}(x), \ldots, f_{u_m}(x)) \\ &= \rho_v(g) f_v(x), \end{aligned}$$

where the third equality follows from the operator intertwining condition. The scaling and time-reversal claims follow analogously by induction using their operator-specific propagation rules. Applying the induction result to the root node r gives $f_r(g \cdot x) = \rho_r(g) f_r(x)$. When the root represents an ODE vector field and $\rho_r = \rho_X$, the tangent representation of the linear action, this becomes exactly the G-equivariance condition, (4), for the vector field.

B.5 Scope of the Symmetry Guarantee

Linear actions. Theorem 3.2 is stated for groups acting linearly on the state, $g \cdot x = \rho_X(g)x$, so that $DF_g(x) = \rho_X(g)$ is independent of x and the root signature $\rho_r = \rho_X$ is well defined. This covers rotations, reflections, permutations, weighted scalings, and time-reversal parities, including every symmetry in Section 4. Linearity restricts the action, not the dynamics: NP-SO(2) is non-polynomial under a linear SO(2) action.

State-dependent representations. The induction in Appendix B.4 is pointwise in x and holds verbatim when each ρ_v is replaced by a 1-cocycle $\rho_v : G \times X \to \mathrm{GL}(\mathcal{V}_v)$, $\rho_v(gh, x) = \rho_v(g, h \cdot x)\rho_v(h, x)$, with (5) required pointwise in x. The tangent map of a smooth action is such a cocycle by the chain rule, $DF_{gh}(x) = DF_g(h \cdot x)DF_h(x)$, so (4) for smooth nonlinear actions is a special case of the extended statement. The extension, however, moves the burden to the premises: under a nonlinear action the state terminal carries no canonical linear type and contractions such as $\| \cdot \|^2$ are no longer invariant, so constructing typed terminals and operators remains open. Our implementation instantiates constant cocycles. Actions that are linear in other coordinates, e.g., locally near a fixed point of a compact group by Bochner's linearization theorem, reduce to the implemented case by running S^{2}GSR in those coordinates.

Affine actions. Translation $x \mapsto x+b$ admits no law $f(g \cdot x) = \rho(g, x) f(x)$ with $\rho(g, x) \in \mathrm{GL}(\mathbb{R}^d)$ for the state terminal $f(x) = x$: at $x = 0$ the left side is $b \neq 0$ while the right side is 0. Affine symmetries therefore require a point/displacement type pair and lie outside the current typing. Translating the full state of an autonomous ODE only forces $\mathbf{f}$ to be constant along the translated directions; the physically relevant case acts on positions alone (e.g., N-body systems), where relative coordinates $x_i - x_j$ are linearly typed terminals, as in E(n)-equivariant networks (Satorras et al., 2021). We leave this extension to future work.

C Sub-expression Semantics

C.1 Probe Semantics Normalization and Numerical Quality

The probe set defined in Sec. 3.3 is fixed throughout search, which makes semantic evaluation deterministic, prevents validation or test leakage, and enables caching across repeated canonical subgraphs. Raw semantic vectors can be sensitive to scale and numerical pathologies. Therefore, for each closed node, we compute a normalized semantic vector

$$\tilde{S}_v = \frac{S_v - \mu(S_v)}{\mathrm{std}(S_v) + \epsilon}, \tag{20}$$

where $\mu(\cdot)$ and $\mathrm{std}(\cdot)$ are computed over the probe set and $\epsilon > 0$ is a small stabilizing constant. We also record numerical quality statistics

$$q_v = \big[r_{\mathrm{nan}}(S_v),\ r_{\mathrm{inf}}(S_v),\ \log(\mathrm{var}(S_v) + \epsilon),\ \log(\|S_v\|_\infty + \epsilon)\big], \tag{21}$$

where r_{nan} and r_{inf} denote the fraction of undefined and infinite values, respectively. These features allow the neural guidance model to downweight unstable sub-expressions involving operators such as division, logarithms, square roots, or exponentials in numerically unsafe regions.

C.2 Handling Operators Inversion in Path-Conditioned Alignment

The path-conditioned derivative target $d_u^{(j)}$ defined in (8) relies on the local inverse ϕ_v^{-1} of the parent operator ϕ_v at u's input slot, given the sibling values $\{f_k\}_{k \in \mathrm{sib}(u)}$. In practice, only a subset of operators in the SEG grammar admit such a slot-wise local inverse on the probe set. We therefore adopt a two-tier dispatch over each non-root node u depending on whether its parent operator ϕ_v is invertible at u's slot.

Slot-wise invertible operators. For operators whose action at slot u, with siblings held fixed, is a bijection on the relevant value space, we instantiate ϕ_v^{-1} in closed form. Representative cases include:

- *Additive slots:* for $\phi_v(u,k) = u + k$, $\phi_v^{-1}(d;k) = d - f_k$.
- *Multiplicative slots away from zero:* for $\phi_v(u,k) = u \cdot k$ with $f_k \neq 0$ on $\mathcal{P}$, $\phi_v^{-1}(d;k) = d/f_k$.
- *Division slots:* for $\phi_v(u,k) = u/k$ with $f_k \neq 0$, $\phi_v^{-1}(d;k) = d \cdot f_k$; for $\phi_v(u,k) = k/u$ with $d \neq 0$, $\phi_v^{-1}(d;k) = f_k/d$.
- *Strictly monotone unaries:* $\exp$, $\log$, odd integer powers z^{2n+1}, and other strictly monotone ϕ_v on the probe range admit unique pointwise inverses.

Whenever ϕ_v falls in this class and the sibling-availability condition $m_k^{\mathrm{sem}} = 1$ for all $k \in \mathrm{sib}(u)$ together with $m_v^{\mathrm{tgt}} = 1$ holds, we set $m_u^{\mathrm{tgt}} = 1$ and compute $d_u^{(j)}$ by (8). The alignment feature $A_v^{(j)}$ in (9) and the embedding h_v^{align} in (10) are then used as stated in the main text.

Non-invertible operators. For operators whose slot-wise action lacks a single-valued inverse on the probe set, we set $m_u^{\mathrm{tgt}} = 0$ and skip target propagation entirely. This class includes:

- *Even powers and absolute values:* z^2, z^{2n}, $|\cdot|$, $\|\cdot\|^2$, where pre-images come in sign- or rotation-equivalent pairs.
- *Periodic unaries:* $\sin$, $\cos$, where pre-images form a discrete family modulo 2π.
- *Saturating or many-to-one maps:* operators with non-injective behavior on the probe range, including contractions whose inverse is ill-conditioned.
- *Multiplicative or division slots near zero:* when f_k or d is too close to zero for stable pointwise inversion, as flagged by the numerical quality features q_v of Appendix C.1.

For any node u whose ancestor path contains at least one such operator, the recursion in (8) terminates and h_u^{align} falls back to the signature-indexed learnable embedding $u_{\sigma(u)}^{\mathrm{align}}$ defined in (10). The mask-aware additive composition in (12) ensures that the encoder still receives well-formed input at u: structural and semantic streams remain active, and only the path-conditioned alignment stream is replaced by its signature prior.

Soundness and graceful degradation. This dispatch preserves the semantics of (8) wherever it is well-posed and never injects spurious targets where it is not. Because m_u^{tgt} is propagated downward, any descendant of a non-invertible ancestor inherits $m_u^{\mathrm{tgt}} = 0$, so the alignment signal degrades gracefully along entire subtrees rather than producing inconsistent partial targets. Symmetry typing and algebraic canonicalization are unaffected by this dispatch, so SEG validity (Theorem 3.2) and the action-mask soundness (Proposition D.1) hold regardless of which tier applies at each node.

D MCTS Pipeline Details

D.1 Validity of the SEG action mask

Terminology. A frontier node $v \in \mathcal{F}(s)$ is an unfilled *placeholder*; filling it with a terminal or an operator makes it a *filled* node. A filled node is *closed* in the sense of Section 3.3 only once all its descendants are filled, so filling an operator node with unfilled children does not close it. Each placeholder carries a signature constraint c_v inherited from its parent's typing rule: a fixed signature, or, for operators whose output signature does not determine their input signatures ($\times$, $\div$, dot product), a constraint linking siblings, e.g., $\kappa_u + \kappa_{u'} = \kappa_v$ and $\eta_u \eta_{u'} = \eta_v$, which is resolved when the first sibling is filled.

Proposition D.1 (Validity of the SEG action mask)**.** *Call a partial SEG* well-typed *if every filled operator node satisfies its typing rule with respect to its children's (possibly constrained) signatures, and every placeholder's constraint is consistent with its parent's rule. If s is well-typed and $a \in \mathcal{A}_{\mathrm{SEG}}(s)$, then the successor state $s' = T(s,a)$ is well-typed. Consequently, every complete SEG reachable by repeated SEG-valid expansions satisfies its root signature and thus denotes a symmetry-admissible expression.*

Proof. An admissible action fills one placeholder v. For a terminal, $\mathrm{TypeCheck}$ ensures that its signature satisfies c_v; for an operator, it ensures that the output signature satisfies c_v and creates placeholder children whose constraints follow from the operator's typing rule. Sibling constraints are updated by propagating the chosen signature, and all other nodes are unchanged, so s' is well-typed. Induction over the search trajectory proves the first claim. A complete SEG has no placeholders, so every node is filled and satisfies its typing rule, and Theorem 3.2 applies to the root.

Remark (completability). Soundness does not imply that every placeholder remains completable within the search limits. States that cannot be completed yield no terminal candidate and are pruned through the backup in Appendix D.2.

D.2 Detailed S^{2}GSR MCTS Procedure

This section provides the detailed tree traversal, reward, backup, and policy-value training procedure used by S^{2}GSR. The procedure follows the hybrid neural-guided MCTS framework of GSR, with two modifications: all actions are restricted to the SEG-valid action set $\mathcal{A}_{\mathrm{SEG}}(s)$, and the neural prior/value estimates are produced by the semantic graph transformer.

Hybrid ϵ-UCB tree traversal. At each visited state s, a random variable $\epsilon \sim \mathcal{U}(0,1)$ selects between vanilla Upper Confidence Bounds for Trees (UCT) and predictor-based UCT (PUCT):

$$a = \begin{cases} \underset{a \in \mathcal{A}_{\mathrm{SEG}}(s)}{\arg\max} \left[\mathrm{UCT}(s,a) = Q(s,a) + C_U \sqrt{\dfrac{\log \sum N(s)}{1+N(s,a)}} \right], & \epsilon < c_\epsilon, \\ \underset{a \in \mathcal{A}_{\mathrm{SEG}}(s)}{\arg\max} \left[\mathrm{PUCT}(s,a) = Q(s,a) + C_P P_\theta(a \mid s) \dfrac{\sqrt{\sum N(s)}}{1+N(s,a)} \right], & \epsilon \geq c_\epsilon. \end{cases} \tag{22}$$

Here $Q(s,a)$ denotes the empirical mean action value, $N(s,a)$ denotes the visit count, C_U and C_P are exploration constants, and c_ϵ controls the probability of using the unbiased UCT branch. The UCT branch maintains exploration when the neural prior is under-trained, while the predictor-based branch exploits the graph-transformer prior over symmetry-valid expansions.

Expansion and canonicalization. After selecting $a = (v, \varphi)$, the transition replaces the open placeholder v by the typed operator or terminal φ and produces

$$s' = \mathrm{Canon}(T(s,a)). \tag{23}$$

The canonicalization operator applies algebraic rewriting, node sharing, and SEG signature propagation. If this expansion closes new sub-expressions, their semantic descriptors are computed on the training-only probe set.

Terminal reward. A terminal state s_T corresponds to a complete candidate vector field $\hat{f}$. After optimizing free constants on the training trajectories, we compute

$$\mathcal{L}_{\mathrm{ODE}}(\hat{f}) = \lambda_{\mathrm{der}} \frac{1}{N} \sum_{i=1}^{N} \left\| \hat{f}(x(t_i)) - \widehat{\dot{x}}(t_i) \right\|_2^2 + \lambda_{\mathrm{traj}} \frac{1}{N} \sum_{i=1}^{N} \left\| \hat{x}(t_i) - x(t_i) \right\|_2^2, \tag{24}$$

where $\widehat{\dot{x}}(t_i)$ is the estimated derivative target and $\hat{x}(t_i)$ is obtained by integrating $\hat{f}$ from the corresponding initial condition. In experiments that use only derivative matching, we set $\lambda_{\mathrm{traj}} = 0$. The terminal reward is

$$\mathcal{R}(s_T) = \exp\left(-\mathcal{L}_{\mathrm{ODE}}(\hat{f}) - \lambda_c C(\hat{f}) - \lambda_n \Omega_{\mathrm{num}}(\hat{f}) \right), \tag{25}$$

where $C(\hat{f})$ is expression complexity and Ω_{num} is a numerical-instability penalty for undefined or infinite values. Symmetry validity is not included as a soft reward term because it is enforced by the SEG action mask.

Backup. After evaluating a leaf or terminal state, the reward is backed up along the selected path. For each traversed edge (s_t, a_t), we update

$$N(s_t, a_t) \leftarrow N(s_t, a_t) + 1, \tag{26}$$

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \frac{\mathcal{R} - Q(s_t, a_t)}{N(s_t, a_t)} \tag{27}$$

For terminal leaves, $\mathcal{R}$ is calculated by $\mathcal{R}(s_T)$. For non-terminal leaves evaluated by the value head, R is replaced by $V_\theta(s)$.

Policy-value training. After each search episode, the MCTS-improved policy at state s_t is computed from visit counts:

$$\pi_{\mathrm{MCTS}}(a \mid s_t) = \frac{N(s_t, a)^{1/\tau}}{\sum_{b \in \mathcal{A}_{\mathrm{SEG}}(s_t)} N(s_t, b)^{1/\tau}}, \tag{28}$$

where τ is a temperature parameter. The graph transformer is trained using

$$\mathcal{L}_{\mathrm{S^2GSR}} = \sum_{s_t} \left[\big(\mathcal{R}_t - V_\theta(s_t)\big)^2 - \lambda_\pi \sum_{a \in \mathcal{A}_{\mathrm{SEG}}(s_t)} \pi_{\mathrm{MCTS}}(a \mid s_t) \log P_\theta(a \mid s_t) \right], \tag{29}$$

where $\mathcal{R}_t$ is the return assigned to state s_t. This objective transfers search-improved decisions back into the graph transformer, allowing future MCTS iterations to prioritize expansions that are symmetry-valid, residual-aligned, compact, and numerically stable.

D.3 S^{2}GSR Algorithm

Algorithm 1 summarizes the complete S^{2}GSR pipeline. Starting from the typed initial state, each simulation expands only SEG-valid actions, canonicalizes every successor state, evaluates terminal and non-terminal leaves with either the ODE reward or the value network, and backs up the resulting return to guide subsequent expansions. Newly closed sub-expressions trigger updates of h_v^{sem} and, when $m_v^{\mathrm{tgt}} = 1$, of h_v^{align}.

Algorithm 1 S^{2}GSR: Symmetry-Pruned and Semantic-Guided Search

Require: Data $\mathcal{D}$; symmetry specification Σ; typed grammar $\mathcal{G}_{\mathrm{typed}}$; policy/value network (P_θ, V_θ)

Build the initial canonical SEG state s_0 with root signature required by Σ
Construct training-only probe set $\mathcal{P}$ for sub-expression semantic evaluation
for each search episode **do**
  for each hnMCTS simulation **do**
    $s \leftarrow s_0, \quad \mathcal{P}_{\mathrm{tree}} \leftarrow \emptyset$
    while s is expanded and non-terminal **do**
      $\mathcal{A} \leftarrow \mathcal{A}_{\mathrm{SEG}}(s)$ by SEG type checking
      Select $a \in \mathcal{A}$ by UCT with prob. c_ϵ, otherwise by PUCT using $P_\theta(a \mid s)$
      $\mathcal{P}_{\mathrm{tree}} \leftarrow \mathcal{P}_{\mathrm{tree}} \cup \{(s, a)\}, \quad s \leftarrow \mathrm{Canon}(T(s, a))$
    end while
    Expand s if non-terminal; update h_v^{sem} and h_v^{align} for newly closed sub-expressions
    $\mathcal{R} \leftarrow \mathcal{R}(s)$ if s is terminal; otherwise $\mathcal{R} \leftarrow V_\theta(s)$
    Backup $\mathcal{R}$ along $\mathcal{P}_{\mathrm{tree}}$ to update $N(s, a) \leftarrow N(s, a) + 1, \quad Q(s, a) \leftarrow Q(s, a) + \frac{\mathcal{R} - Q(s, a)}{N(s, a)}$
  end for
  Obtain $\pi_{\mathrm{MCTS}}(\cdot \mid s)$ from visit counts over $\mathcal{A}_{\mathrm{SEG}}(s)$
  Sample the next SEG expansion using π_{MCTS} and update $s_0 \leftarrow \mathrm{Canon}(T(s_0, a))$
  Train (P_θ, V_θ) using the MCTS policy target and the evaluated return
end for
return Best vector field $\hat{\mathbf{f}}$

D.4 Constant Optimization: Global BFGS with Local Alignment Fitting

The SEG grammar treats numerical constants as typed placeholder terminals $c_1, c_2, \ldots$ that are resolved by numerical optimization rather than sampled symbolically. We optimize constants at two stages with different scopes and objectives.

Global optimization at terminal states. At a terminal state s_T with complete vector field $\hat{f}$, all placeholder constants $\mathbf{c} = (c_1, \ldots, c_K)$ are jointly fit by BFGS to minimize the ODE objective

$$\mathbf{c}^\star \;=\; \arg\min_{\mathbf{c}} \; \mathcal{L}_{\text{ODE}}\big(\hat{f}(\,\cdot\,;\mathbf{c});\, \mathcal{D}\big), \tag{30}$$

where $\mathcal{L}_{\text{ODE}}$ is given by (24). We use multi-start BFGS with random initializations augmented by warm-starts from the locally-fit values described below, and the resulting $\mathbf{c}^\star$ enters the terminal reward $R(s_T)$ in (25).

Constants in normalized alignment. Because the alignment feature $A_v^{(j)}$ in (9) compares the standardized semantic vector $\tilde{S}_v$ against the standardized target $\tilde{d}_v^{(j)}$, a placeholder constant c leaves $A_v^{(j)}$ unchanged whenever its effect on f_v reduces to an affine reparameterization of the raw semantic vector S_v. We classify each constant as *alignment-invariant* or *alignment-relevant* according to its syntactic position in f_v:

- **Alignment-invariant (no local fit required).** $\tilde{S}_v$ is unchanged, up to sign, when c appears as
  - a top-level multiplicative prefactor: $f_v = c \cdot g(x)$;
  - a top-level additive offset: $f_v = c + g(x)$;
  - the outermost affine wrapper of a closed sub-expression: $f_v = a\,g(x) + b$ with a, b both placeholders;
  - a multiplicative prefactor inside any operator that normalizes its input (e.g., the immediate child of a downstream standardization step).

  These constants drop out of zero-mean unit-variance normalization and are deferred entirely to the global stage in (30). The sign of multiplicative prefactors, which a signed cosine does retain, is fit by a one-dimensional grid $\{+1, -1\}$ rather than BFGS.
- **Alignment-relevant (local fit required).** Constants whose effect on f_v is not affine in the raw semantics change the *shape* of $\tilde{S}_v$ and must be fit before $A_v^{(j)}$ is meaningful. Representative cases include constants inside any nonlinearity, e.g., $\sin(cx)$, $\exp(cx)$, x^c, $1/(c+x)$, $\tanh(cx)$; constants in non-outermost positions of a composed expression; and constants appearing inside parent operators along the path-conditioned inverse chain when these operators are themselves nonlinear in c.

Local fitting for alignment-relevant constants. When a closed node v contains K_v^{loc} alignment-relevant constants $\mathbf{c}_v^{\text{loc}}$, we solve a low-dimensional inner optimization

$$\mathbf{c}_v^{\text{loc},\star} \;=\; \arg\max_{\mathbf{c}_v^{\text{loc}}} \; \frac{\big\langle \tilde{S}_v(\mathbf{c}_v^{\text{loc}}),\, \tilde{d}_v^{(j)} \big\rangle}{\big\|\tilde{S}_v(\mathbf{c}_v^{\text{loc}})\big\|_2 \,\big\|\tilde{d}_v^{(j)}\big\|_2 + \epsilon}, \tag{31}$$

holding all other constants and the SEG structure fixed. The fit is run with BFGS over the probe set $\mathcal{P}$, which makes it cheap because K_v^{loc} is typically one to three. Alignment-invariant constants are held at neutral values (1 for multiplicative, 0 for additive) during this fit so that they do not interfere with the optimization landscape.

Caching and warm-starting. Local fits are cached jointly by the canonical sub-expression key and a quantized fingerprint of the normalized target $\tilde{d}_v^{(j)}$, so that the same canonical sub-expression encountered at slots with similar behavioral targets reuses prior fits. The terminal global BFGS in (30) is warm-started from $\mathbf{c}_v^{\text{loc},\star}$ values aggregated across the SEG, which we observe to reduce the number of restarts required for convergence. This two-tier scheme isolates expensive global optimization to terminal evaluations while keeping per-expansion alignment computations bounded by a small low-dimensional fit, with many sub-expressions incurring no local optimization at all by virtue of the normalization-induced invariance.

E EXPERIMENTAL DETAILS

E.1 SYMMETRY-CONSTRAINED BENCHMARK PROTOCOL

Systems and symmetry actions. The polynomial systems are P-SO(2), the damped oscillator in (16), and P-SE, the growth system in (17). The non-polynomial systems are NP-SO(2), (1), and NP-TR, (18). For a planar rotation $Q \in SO(2)$, both SO(2) systems obey $f(Qx) = Qf(x)$. In NP-SO(2), the squared radius $r = \|x\|^2$ is invariant, and the field is $g(r)Jx$, where

$$J = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \qquad g(r) = \sin(r)e^{-r}.$$

The growth system obeys $f(S_\lambda x) = S_\lambda f(x)$ with $S_\lambda = \mathrm{diag}(\lambda^2, \lambda)$ and $\lambda > 0$. Its coordinate weights are $(2, 1)$: x_1 and x_2^2 have the same weight, whereas x_2 has weight one. This is a weighted state scaling with unchanged time. The growth field also has the ordinary reflection equivariance $f(Tx) = Tf(x)$ for $T = \mathrm{diag}(1, -1)$. Time reversal uses a different vector-field law: NP-TR satisfies $f(Tx) = -Tf(x)$, corresponding to reversing time while changing the sign of x_2.

Trajectory generation and preprocessing. All four systems are evaluated over 100 random seeds, with separate training, validation, and test trajectories. P-SO(2) uses $50/10/10$ trajectories, each containing 100 time steps at $\Delta t = 0.2$. Initial conditions are sampled as $x(0) = a(\cos\theta, \sin\theta)$, with radius $a \sim \mathcal{U}[0.5, 2]$ and angle $\theta \sim \mathcal{U}[0, 2\pi]$; the radius a is distinct from the squared radial coordinate r in NP-SO(2). Observations receive 20% additive noise per dimension. P-SE uses $100/20/20$ trajectories with 100 time steps at $\Delta t = 0.02$, initial conditions in $[0.2, 1] \times [0.2, 1]$, and 5% multiplicative Gaussian noise. Each NP system uses $100/20/20$ trajectories with 100 time steps at $\Delta t = 0.2$ and 5% multiplicative Gaussian noise. Gaussian-process smoothing is applied before equation discovery. These are distinct per-system protocols; the oscillator's additive corruption is not replaced by the multiplicative corruption used for the other three systems.

Typed primitives and admissible compositions. Typed terminals are $\{x, c\}$ for the SO(2) systems, with x vector-typed, and scalar $\{x_1, x_2, c\}$ for P-SE and NP-TR. For SO(2), the state vector x carries the standard vector representation and $\|x\|^2$ carries the trivial scalar representation. The norm-squared rule constructs the latter from x. A typed J operator, corresponding to R_{90}, maps vectors to vectors because $JQ = QJ$ for planar rotations. Consequently, the grammar can compose $\sin(r)$ and e^{-r} into an invariant scalar coefficient and multiply it by Jx to represent NP-SO(2). The added J map is specific to the planar SO(2) configuration; it is not an intertwiner for arbitrary reflection actions. For P-SE, scalar nodes carry graded weights $(2, 1)$, so x_1 and x_2^2 can contribute to the first component. For NP-TR, the coordinate parities under T are $(+1, -1)$, while the required vector-field component parities are $(-1, +1)$. Thus x_2 is admissible for the first component, and $-\sin(x_1)/(1 + x_1^2)$ is admissible for the second.

The free-form operator vocabulary used for the NP comparison comprises arithmetic operations and elementary functions $\{+, -, \times, \div, \sin, \cos, \exp, \log, z^n\}$. S^{2}GSR admits their compositions according to the SEG typing rules, with the additional planar J map where admissible. This describes how the target can be constructed; the comparison does not require an unconditional claim that every method receives identical typed objects.

Baseline configurations and expressivity. The eight baselines in Table 3 cover evolutionary symbolic regression (PySR (Cranmer, 2023)), D-CODE, sparse regression (SINDy and WSINDy), neural symbolic methods (uDSR and ODEFormer), the GSR graph-search predecessor, and symmetry-constrained sparse regression (EquivSINDy-c). For P-SO(2) and P-SE, the sparse-regression methods follow the official task-specific configurations of EquivSINDy: they use the complete quadratic library

$$\Theta(x) = [1, x_1, x_2, x_1^2, x_1x_2, x_2^2]^\top,$$

with Gaussian-process-smoothed state–derivative pairs. EquivSINDy-c imposes the known rotation or weighted-scaling action by restricting the coefficient matrix to the corresponding equivariant nullspace; no autoencoder or latent coordinate transformation is used for these two systems. For the NP systems, SINDy and WSINDy use degree-five polynomial libraries. EquivSINDy-c uses the equivariant basis $\{x, Jx\}$ with coefficients polynomial in $r = \|x\|^2$ through degree five for NP-SO(2),

and a parity-constrained polynomial library for NP-TR. The factors $\sin(r)e^{-r}$ and $-\sin(x_1)/(1+x_1^2)$ lie outside these finite polynomial libraries. Zero structural recovery in these configurations is therefore compatible with substantial reconstruction R^2. Adding suitable non-polynomial features would change the library and the comparison; the reported results do not evaluate all such extensions.

Recovery and reconstruction. Recovery is a structural, componentwise criterion. After algebraic simplification and canonical collection of numerical multipliers, write the ground-truth and inferred components as

$$f_j(x) = \sum_{\tau \in S_j} c_{j,\tau}\, \tau(x), \qquad \widehat{f}_j(x) = \sum_{\tau \in \widehat{S}_j} \widehat{c}_{j,\tau}\, \tau(x),$$

where S_j and $\widehat{S}_j$ are the sets of retained symbolic terms in component j. A seed is counted as recovered if and only if $\widehat{S}_j = S_j$ for every vector-field component. The numerical coefficient values $c_{j,\tau}$ and $\widehat{c}_{j,\tau}$ do not enter this indicator; missing, spurious, or component-misassigned terms cause failure, and recovering only a subset of the components is insufficient. This definition follows the official EquivSINDy (Yang et al., 2024) evaluation, which compares the inferred and ground-truth support masks and reports coefficient error separately. RR is the fraction of the 100 seeds meeting this criterion. Reconstruction instead evaluates trajectories obtained by integrating the fitted field against reference trajectories. Table 3 reports mean reconstruction R^2 across seeds, with $\pm$ denoting the half-width of a 95% confidence interval for that mean. Table 5 uses the same summary convention for validation reconstruction. A high reconstruction score does not certify correct term support, since structurally different fields can agree closely over the observed trajectories.

E.2 Ablation Configuration and Recovery Uncertainty

Controlled comparisons. Table 4 evaluates P-SO(2) and NP-SO(2) under the corresponding data protocols above. All variants retain the graph representation, algebraic canonicalization, search budget, probe-set size, MCTS exploration constants, constant-optimization routine, and stopping criteria. Active typing enforces the prescribed SO(2) signatures and admissibility mask. Relaxed typing uses a trivial symmetry specification while retaining ordinary operator validity and the graph-search machinery. The comparison therefore removes physical symmetry constraints without removing canonicalization.

None uses structural and signature features without semantic embeddings. Sem-only adds the masked semantics of closed sub-expressions, but removes the alignment stream entirely. Sem+Align additionally uses the path-conditioned targets and their availability masks. Probe points are sampled only from training trajectories and kept fixed during a search, as described in Appendix C.1. When a path does not admit a valid local inverse, the target mask suppresses alignment and the signature-indexed fallback is used (Appendix C.2). Sem-only evaluates the complete removal of the alignment stream; it does not measure the frequency with which alignment is available in the full model.

Magnitude of the contributions. On P-SO(2), activating typing increases structural-only RR by 0.74 (0.15 $\rightarrow$ 0.89); semantics and alignment together then add 0.07 (0.89 $\rightarrow$ 0.96). With full guidance, relaxing typing reduces RR from 0.96 to 0.21. On NP-SO(2), typing increases structural-only RR by 0.16 (0.06 $\rightarrow$ 0.22), while adding full semantic guidance within the typed space contributes 0.14 (0.22 $\rightarrow$ 0.36). With full guidance, typing contributes 0.18 (0.18 $\rightarrow$ 0.36), exceeding the semantic gain but by a much smaller margin than on P-SO(2). Under active typing, Sem-only contributes 0.06 on NP-SO(2), and alignment adds a further 0.08, compared with 0.03 and 0.04 on P-SO(2). Under relaxed typing, NP-SO(2) RR rises from 0.06 to 0.09 and 0.18 as semantics and alignment are added. These observed differences support complementary contributions, with larger gains from semantic guidance on the non-polynomial task.

Wilson intervals for symbolic recovery rate. Table 7 supplements the point estimates in Table 3. Each reported proportion $\hat{p}$ corresponds to an integer count $k = 100\hat{p}$ among $n = 100$ seeds. We compute a two-sided Wilson 95% interval $\mathrm{CI}_{0.95}(\hat{p})$as

$$\left[\frac{\hat{p} + z^2/(2n) - z\sqrt{\hat{p}(1-\hat{p})/n + z^2/(4n^2)}}{1 + z^2/n}, \frac{\hat{p} + z^2/(2n) + z\sqrt{\hat{p}(1-\hat{p})/n + z^2/(4n^2)}}{1 + z^2/n} \right], \quad (32)$$

Table 7: Wilson 95% confidence intervals for the symbolic recovery rates of S^{2}GSR and the eight baselines in Table 3, computed from 100 seeds per method and system and rounded to three decimal places.

Method	P-SO(2)	P-SE	NP-SO(2)	NP-TR
SINDy	[0.078, 0.210]	[0.022, 0.112]	[0.000, 0.037]	[0.000, 0.037]
WSINDy	[0.000, 0.037]	[0.000, 0.037]	[0.000, 0.037]	[0.000, 0.037]
PySR	[0.006, 0.070]	[0.010, 0.085]	[0.000, 0.037]	[0.010, 0.085]
D-CODE	[0.000, 0.037]	[0.000, 0.037]	[0.000, 0.037]	[0.006, 0.070]
uDSR	[0.041, 0.150]	[0.016, 0.098]	[0.000, 0.037]	[0.022, 0.112]
ODEFormer	[0.006, 0.070]	[0.006, 0.070]	[0.000, 0.037]	[0.010, 0.085]
GSR	[0.093, 0.233]	[0.055, 0.174]	[0.022, 0.112]	[0.070, 0.198]
EquivSINDy-c*	[0.814, 0.937]	[0.963, 1.000]	[0.000, 0.037]	[0.000, 0.037]
S^{2}GSR*	[0.902, 0.984]	[0.963, 1.000]	[0.273, 0.458]	[0.482, 0.672]

where $z = \Phi^{-1}(0.975)$, and Φ is the standard normal cumulative distribution function. These intervals quantify binomial uncertainty across seeds for the symbolic recovery rate; they are distinct from the supplied confidence intervals for mean reconstruction R^2. They also do not constitute a paired significance test between methods. In particular, observing zero recoveries does not imply a zero population recovery probability.

E.3 Symmetry-selection Protocol

Candidates and data separation. We use the three candidate specifications $\Sigma_{SO(2)}$, Σ_{TR}, and Σ_{SE}. Their state transformations and required vector-field laws are

$$\begin{aligned} SO(2): &\quad x \mapsto Qx, && f(Qx) = Qf(x), && Q \in SO(2), \\ \mathrm{TR}: &\quad x \mapsto Tx, && f(Tx) = -Tf(x), && T = \mathrm{diag}(1,-1), \\ \mathrm{SE}: &\quad x \mapsto S_\lambda x, && f(S_\lambda x) = S_\lambda f(x), && S_\lambda = \mathrm{diag}(\lambda^2, \lambda),\ \lambda > 0. \end{aligned}$$

Within each seed, discovery fits a candidate field under each specification using the training trajectories, with an equal search budget per candidate. The resulting fields are evaluated by integrating them on validation trajectories. If $\hat{f}_{\Sigma,s}$ denotes the field fitted under specification Σ in seed s, the selection rule is

$$\hat{\Sigma}_s \in \operatorname*{arg\,max}_{\Sigma \in \{\Sigma_{SO(2)}, \Sigma_{\mathrm{TR}}, \Sigma_{\mathrm{SE}}\}} R^2_{\mathrm{val}}(\hat{f}_{\Sigma,s}), \qquad \hat{f}_s = \hat{f}_{\hat{\Sigma}_s, s}. \tag{33}$$

There is no extra complexity penalty in this selection score. Any complexity regularization used within discovery remains part of the terminal reward in (25). Validation trajectories are used for selection, not for constructing training probes or fitting candidate constants. Table 5 reports each candidate's mean validation score and its 95% confidence interval across the 100 seeds. Bold entries mark candidate specifications that are symmetries of the ground-truth field, including both SO(2) and TR for NP-SO(2). Selection follows the highest validation score within each seed, independently of this highlighting; the reported means are not per-seed selection frequencies. A separate evaluation of the selected models on test trajectories would be needed to report their independent test performance.

Interpretation and computational cost. The best mean-scoring candidates are SO(2), SE, SO(2), and TR for P-SO(2), P-SE, NP-SO(2), and NP-TR, respectively. This is selection among three specified search spaces with a finite budget. It assumes that useful candidate actions can be enumerated; the present experiment neither includes a trivial-group fallback nor evaluates an unrestricted search over all symmetries. It requires three discovery runs plus validation evaluations per seed. Equal per-candidate budgets do not imply identical elapsed time, since admissible expansions and terminal evaluations differ between specifications.

The candidate symmetries need not be mutually exclusive. For NP-SO(2), $r(Tx) = r(x)$ and $JT = -TJ$, giving

$$f(Tx) = g(r)JTx = -Tg(r)Jx = -Tf(x).$$

Thus both SO(2) and the tested TR action are valid for this field, despite their different scores in Table 5. Different constraints organize the admissible search space differently, and finite-budget

discovery may perform differently even when both spaces contain the target. Accordingly, a lower validation score alone does not prove that a candidate symmetry is absent, and the highest score does not identify a unique or maximal symmetry group. A misspecified constraint can also exclude the true field while retaining an adequate trajectory approximation; reconstruction is evidence for predictive suitability on the validation distribution, not a proof of global symmetry.

E.4 ODEBench Protocol and Reporting

Dataset and corruption. ODEBench (d'Ascoli et al., 2024) contains 63 curated systems: 23 one-dimensional, 28 two-dimensional, 10 three-dimensional, and 2 four-dimensional ODEs. They include polynomial, rational, trigonometric, exponential, oscillatory, and chaotic dynamics. Each system has a symbolic reference field and an initial condition. Following the reconstruction protocol, each method receives a single observed trajectory. We apply multiplicative Gaussian noise at levels $\sigma \in \{0, 0.02, 0.05\}$ and uniformly remove a fraction $\rho \in \{0, 0.5\}$ of time points, yielding six conditions. The same corrupted observations are used across methods within a condition. Reconstruction is evaluated against clean reference trajectories.

Trivial symmetry and baselines. S^2GSR uses a trivial symmetry specification for every ODEBench system. No per-system symmetry annotation, ground-truth symmetry, or candidate-selection stage is used. Symmetry constraints therefore do not prune candidates on this benchmark, while ordinary operator validity, algebraic canonicalization, and semantic guidance remain. The GSR comparison evaluates the addition of sub-expression semantics and path-conditioned guidance to the graph-search backbone in this setting; it does not measure gains from prescribed physical symmetry.

The six baselines are GSR (Xiang et al., 2025), ODEFormer (d'Ascoli et al., 2024), SINDy (Brunton et al., 2016), WSINDy (Messenger & Bortz, 2021), PySR (Cranmer, 2023), and uDSR (Landajuela et al., 2022). ODEFormer performs pretrained trajectory-to-equation prediction, SINDy and WSINDy use sparse regression over libraries, PySR uses evolutionary symbolic search, and uDSR uses neural-guided symbolic regression. Baselines requiring state–derivative pairs receive numerical derivative estimates from the observed trajectory, with smoothing and finite-difference settings selected by validation. The NP-system library choices above should not be assumed to define the separate ODEBench configuration.

Reconstruction metric. For each inferred field, we integrate the predicted ODE from the reference initial condition over the reference time interval and evaluate trajectory-level R^2 against the clean trajectory. Figure 2 reports the percentage of systems satisfying the strict threshold $R^2 > 0.9$ in each corruption condition. This threshold describes reconstruction, whereas RR in Table 3 requires componentwise recovery of the ground-truth term support. A thresholded trajectory score cannot determine whether an inferred expression has the correct symbolic terms.

Complexity and inference time. Expression complexity counts operators, variables, and constants in the returned symbolic equations. Inference time is wall-clock time required to produce the final symbolic prediction, in seconds. Figure 2 displays their distributions on logarithmic axes. The complexity comparison concerns returned expression size, not the number of search states explored. Likewise, wall-clock time includes costs beyond MCTS traversal and is distinct from the number of MCTS simulations. The displayed runtime distributions alone do not compare total training or pretraining expenditure.