DYNAMICAL SYSTEM DISCOVERY FROM SPECIFICATIONS AND PARTIAL OBSERVATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Dynamical system discovery is often framed as fitting ordinary differential equations to observed trajectories. In many applications, however, the desired model is defined not only by prediction error but also by a *task specification*: a natural-language description of task-critical mechanisms that the model must represent. In postprandial glucose modeling, for example, a meal response simulator must represent how ingested carbohydrates enter the bloodstream over time; a post-meal decision support model must distinguish meal absorption from delayed insulin action; and a physiological model must expose hepatic regulation. These task-critical mechanisms are often not measured, creating a mismatch between what the data observe and what the model must represent. We study this setting as *task-specified dynamical system discovery* under *specification-induced partial observability*.

We propose a framework that treats the state specification itself as an object of discovery. Given a task description and partially observed time series, the framework searches over observed and latent states, analytic dynamics, observation mappings, and causal initialization rules. A proposer large language model (LLM) constructs and revises candidate models through a graph meta model that represents dependencies. A deterministic verifier checks model validity and executable task requirements, while a separate LLM critic supplies scientific feedback for revision. Continuous parameters are fitted on training trajectories using collocation initialization and rollout refinement. Validation performance guides candidate selection and acceptance of pruning. Experiments demonstrate that compared to baselines, our method more reliably recovers task-critical mechanisms while achieving better prediction in unseen, held-out interventions.

1 INTRODUCTION

Dynamical system discovery is often posed as equation discovery over a *given* state space: the learner is given trajectories of measured variables and searches for a vector field that predicts their evolution. This view underlies sparse regression (Brunton et al., 2016), symbolic regression (Cranmer, 2023), neural ODEs (Chen et al., 2018), and recent LLM-guided equation discovery (Shojaee et al., 2025). It is appropriate when the variables required by the target model are already measured, or can be constructed directly from measurements. In many settings, however, **the target model is defined by the task it must support**. A postprandial glucose model for sensor testing must represent meal appearance; a model for insulin therapy design must distinguish meal-driven glucose input from delayed insulin action; and a physiological simulator may need to expose hepatic regulation or insulin secretion. These task-critical mechanisms are often absent from the measurements, so accurate prediction alone does not guarantee that the discovered equations are usable for the intended task.

We therefore consider a *different* problem: discovering a dynamical model whose state space is not given in advance, but must be inferred from both a task specification and data. The input is a natural-language specification and partially observed time series, and the output is an analytic dynamical system with a state representation sufficient for the specified task. This requires deciding which state variables should be present, which are observed or latent, and how latent task-critical mechanisms can be inferred from the available measurements. We call this setting **task-specified dynamical system discovery under specification-induced partial observability**. Unlike standard equation discovery, which searches for equations over a fixed observed state space, it requires joint discovery of state variables, latent state reconstruction, and governing equations.

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.

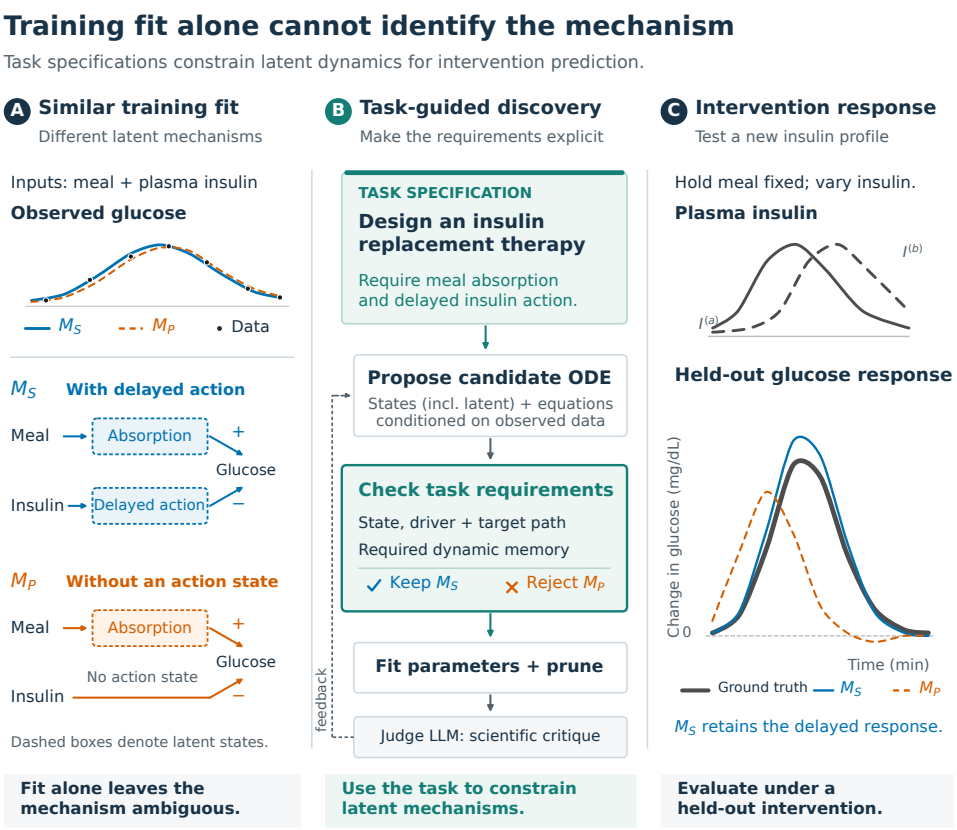


Figure 1: **Task-specified dynamical system discovery under partial observability.** (A) Two models fit the observed glucose trajectory similarly: $\mathcal{M}_S$ includes a latent delayed insulin-action state, while $\mathcal{M}_P$ omits it. (B) The task specification requires meal absorption and delayed insulin action. A proposer LLM generates candidate ODEs, a deterministic verifier checks the required mechanisms, and fitting, pruning, and LLM critique refine the candidates. (C) Under the unseen insulin profile, $\mathcal{M}_S$ retains a delayed response close to the ground truth, while $\mathcal{M}_P$ deviates.

This problem raises two difficulties that are largely absent from standard equation discovery. First, **state-specification ambiguity**: a task specification may indicate *which* mechanism to expose, but not *the state representation in which it should be encoded*. For example, a delayed regulatory effect could be represented by a single insulin-action state, a short chain of compartments, or another low-dimensional surrogate. Thus, the state space is not a fixed input to discovery, but part of what must be inferred. Second, **specification-induced partial observability**: the variables missing from the data are often exactly those required by the task. In postprandial glucose modeling, the task may require separating meal appearance, insulin-driven disposal, and delayed hepatic regulation, while the data contain only glucose, insulin, and meal metadata. A successful model must therefore construct not only task-critical latent states themselves but also the *reconstruction map from observed trajectories to latent states*. This makes discovery a joint problem of state space construction, latent mechanism recovery, and equation fitting.

Our key idea is to make the state space itself part of model discovery. To address *state-specification ambiguity*, we search over candidate state specifications containing *both observed and latent* variables, with interpretable roles proposed in response to the task specification. To address *specification-induced partial observability*, we connect latent mechanisms to measured outputs through explicit dynamics and observation mappings, and initialize them using causal rules that depend only on information available at deployment. Their parameters are estimated from training trajectories; latent initial conditions are not fitted to held-out trajectories. A maximum a posterioi (MAP) formulation motivates the combination of data fit, specification constraints, and model complexity, while the implementation uses a bounded search with numerical fitting and validation-based selection. Opera-

Table 1: Related work in equation and dynamical system discovery. **Our focus is discovering task-critical latent states and their equations from natural-language requirements and partial observations.**

Method family and examples	What variables (beyond observed) are modeled?	How are these variables constructed?	How does scientific knowledge guide discovery?
Equation discovery (SINDy, PySR)	Engineered features[a]	Supplied	Library / operator design
Latent-state discovery (Latent ODE, Deep Delay AE)	Generic latent states[b]	Learned from data	Statistical / embedding priors
Physical structure (SKANODE, LHM)	Physical latent states[c]	Defined by experts	Architecture / expert ODE constraints
Formal theory (AI-Descartes, AI Hilbert)	Physical variables	Defined in the theory	Axiom-constrained formula search
LLM equation search (LLM-SR, SR-LLM, LLM-ODE, IGSR)	Observed variables + features	Supplied / proposed by LLM	Proposal and search guidance
Agentic modeling (D3)	Observed variables + features	Supplied / acquired by LLM	Model construction and critique
Hybrid model design (HDTwinGen)	Observed states	Supplied	Requirement-guided model design
Knowledge-guided discovery (MEDA)	Observed states	Proposed by LLM	Knowledge-guided constrained search
Task-specified discovery (Ours)	**Task-critical latent states**[d]	**Proposed by LLM**	**Task-guided proposal and selection**

References. SINDy (Brunton et al., 2016), PySR (Cranmer, 2023), Latent ODE (Rubanova et al., 2019), Deep delay autoencoders (Bakarji et al., 2023), SKANODE (Liu et al., 2025), LHM (Qian et al., 2021), AI-Descartes (Cornelio et al., 2023), AI Hilbert (Cory-Wright et al., 2024), LLM-SR (Shojaee et al., 2025), SR-LLM (Guo et al., 2025), LLM-ODE (Bideh and Gryak, 2026), IGSR (Saveliev et al., 2026), D3 (Holt et al., 2024a), HDTwinGen (Holt et al., 2024b), MEDA (Krongauz et al., 2026).

[a] **Engineered features:** explicitly constructed functions of supplied variables or their histories, used as inputs or candidate terms in a model.

[b] **Generic latent states:** learned hidden coordinates, not constrained to represent specified scientific mechanisms.

[c] **Physical latent states:** unmeasured states with physical meanings. Experts specify their meanings or coordinate structure.

[d] **Task-critical latent states:** unmeasured dynamical states introduced to realize task-required mechanisms, with their representation and governing equations inferred from requirements and data.

tionally, a proposer large language model (LLM) constructs and revises candidate state specifications and analytic mechanisms. A deterministic verifier checks model validity and executable task requirements, while a separate LLM critic identifies scientific concerns and suggests revisions. These complementary forms of feedback guide the search without treating either structural compliance or scientific critique as proof that the underlying physical mechanism has been recovered.

Contributions. ① We formalize *task-specified dynamical system discovery under specification-induced partial observability*, where the target ODE model is constrained jointly by observed trajectories and a natural-language task specification (Sec. 3). ② We develop a specification-guided search over state representations and analytic dynamics, separating deterministic validity and executable requirement checks from advisory LLM scientific critique to guide model construction and revision. ③ We introduce a graph meta model that supports interpretable latent mechanisms and promotes closed-form least-squares parameter fitting with pruning (Sec. 4). ④ We construct benchmarks spanning application domains, task-required mechanisms, and observability levels, and show that our method consistently recovers task-critical mechanisms and outperforms baselines in held-out interventions (Sec. 5).

2 Related Works

Equation discovery. SINDy and PySR search for governing equations over *supplied* variables or *engineered* features (Brunton et al., 2016; Cranmer, 2023). They do not determine which additional hidden states a scientific task requires.

Dynamical system discovery with latent states. Latent ODEs and deep delay autoencoders learn hidden coordinates from partial observations (Rubanova et al., 2019; Bakarji et al., 2023). SKANODE and expert-neural hybrids introduce physical meaning through prescribed state roles or expert equations (Liu et al., 2025; Qian et al., 2021). Our focus is not latent modeling alone, but discovering the state representation needed to realize specified mechanisms.

Scientific constraints. AI-Descartes and AI Hilbert combine data with formal axioms to derive or validate formulae (Cornelio et al., 2023; Cory-Wright et al., 2024). They reason over an existing formal vocabulary rather than construct persistent latent dynamics required by natural-language tasks.

LLM-guided equation discovery. LLM-SR, SR-LLM, and LLM-ODE guide equation search using scientific knowledge or candidate-derived feedback (Shojaee et al., 2025; Guo et al., 2025; Bideh and Gryak, 2026). IGSR adds term-level influence feedback and pruning (Saveliev et al., 2026). These methods primarily search equations and features over supplied variables, rather than reconstruct missing task-required dynamical states.

Agentic model construction. D3 combines dynamical modeling with feature acquisition (Holt et al., 2024a), while HDTwinGen searches hybrid models using natural-language modeling requirements (Holt et al., 2024b). MEDA formalizes biological knowledge into variables, constraints, and ODEs (Krongauz et al., 2026). These approaches go beyond text-guided equation proposals, but their reported studies do not establish reconstruction of missing task-critical mechanisms. In particular, MEDA's missing-variable experiments assess robustness rather than demonstrate recovery of the omitted mechanisms.

Our distinction. We jointly discover **task-critical latent states and their governing equations** from natural-language scientific requirements and partially observed trajectories. The requirements specify which mechanisms the model must expose, while their state representation and dynamics remain to be discovered. Table 1 summarizes these distinctions. An extended discussion is in Appendix A.

3 Problem Formulation

We consider the discovery of interpretable analytical dynamical models from a natural-language task specification and partially observed trajectories. The input consists of a *context* $\mathcal{C}$ describing the modeling task (e.g., "design an insulin replacement therapy") and required mechanisms, together with a dataset $\mathcal{D} = \{\mathbf{U}_n, \mathbf{X}_{\text{obs},n}, \mathbf{Y}_n\}_{n=1}^{N}$ of N trajectories. Each trajectory n contains *inputs* $\mathbf{u}_n(t)$, *observed states* $\mathbf{x}_{\text{obs},n}(t)$, and *target outputs* $\mathbf{y}_n(t)$ at observation times $\{t_{n,k}\}_{k=1}^{T_n}$.

Our goal is to discover a model

$$\mathcal{M} = (\Gamma, f, h, g; \psi) \tag{1}$$

that is compatible with both the context and the data. Here, Γ is *state specification*, defined as an ordered list $\{v_1, \ldots, v_{d_x}\}$ of *state variable definitions*, including *task-critical latent states* $\mathbf{x}_{\text{lat}}(t)$; f defines the state dynamics; h maps the generated states and supplied inputs to the target outputs; and g specifies causal initial conditions. Continuous parameters ψ include the parameters of the dynamics, output mappings, and initialization rules. Given $\mathbf{x}_n(t) = [\mathbf{x}_{\text{obs},n}(t); \mathbf{x}_{\text{lat},n}(t)]$, the model predicts

$$\mathbf{x}_n(t_{n,0}) = g(\boldsymbol{\zeta}_n; \psi)\,, \quad \dot{\mathbf{x}}_n(t) = f(\mathbf{x}_n(t), \mathbf{u}_n(t); \psi)\,, \quad \mathbf{y}_n(t) = h(\mathbf{x}_n(t), \mathbf{u}_n(t); \psi) + \boldsymbol{\varepsilon}_n(t), \tag{2}$$

where $\boldsymbol{\varepsilon}_n(t)$ is the measurement noise, and $\boldsymbol{\zeta}_n$ is the initialization information, containing only initial measurements of observed states and initial inputs. Directly measured state coordinates are initialized from their first measurements. For a latent state, g may specify a constant unknown initial value, $g_j(\boldsymbol{\zeta}_n; \psi) = \alpha_j$, or an analytical map that produces different initial values for different trajectories.

We formalize compatibility with context and data through maximum a posteriori (MAP) inference. Assuming data $\mathcal{D}$ has no additional dependence on $\mathcal{C}$ once $\mathcal{M}$ is specified, namely $\mathcal{D} \perp \mathcal{C} \mid \mathcal{M}$, we can write the MAP inference as

$$\mathcal{M}^* = \arg\max_{\mathcal{M}} p(\mathcal{M} \mid \mathcal{C}, \mathcal{D}) = \arg\max_{\mathcal{M}} p(\mathcal{D} \mid \mathcal{M}) \cdot p(\mathcal{M} \mid \mathcal{C}). \tag{3}$$

The likelihood measures agreement with observations, while the context-conditioned prior expresses scientific compatibility and preferences over model structure. This is a modeling objective; Section 4 describes its practical approximation.

Challenges. Consider modeling glucose-insulin dynamics for insulin-therapy design from partially observed trajectories of plasma glucose and meal tests.

(C1) **Context grounding.** Predictive accuracy alone does not establish suitability for the task. The model must also represent relevant scientific relationships (e.g., insulin action and meal-induced glucose dynamics).

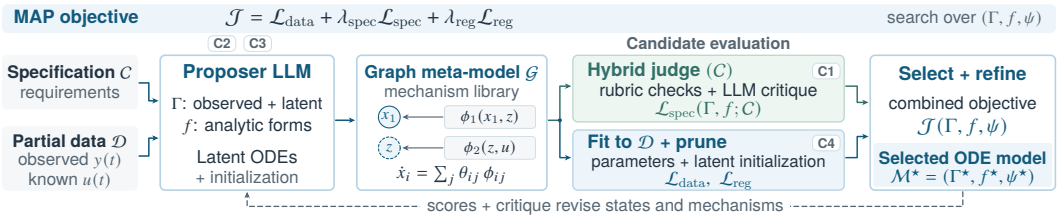


Figure 2: **How the proposed method addresses the challenges in Section 3?** Given a natural-language specification $\mathcal{C}$ and partial observations $\mathcal{D}$, a proposer LLM searches over observed and interpretable latent states Γ, analytic mechanisms f, h, and initialization rules g (C2)(C3). A graph meta-model $\mathcal{G}$ expresses each state derivative as a weighted sum of nonlinear candidate functions. Candidate evaluation combines specification grounding from deterministic rubric checks and LLM scientific critique (C1) with parameter and initialization fitting and sparsity-promoting pruning (C4). Scores and critique guide structural revision and selection of the final model $\mathcal{M}^\star$. In the graph, solid and dashed state nodes denote observed and latent states, respectively.

(C2) **Unknown state representation.** Neither the task nor the measurements uniquely determine state specification Γ. Alternative compartment structures or delayed-action states may explain the same measured outputs.

(C3) **Partial observation and initialization.** Task-required quantities may be unmeasured, motivating interpretable latent states or derived variables. Latent states additionally require initialization rules that remain usable on new trajectories.

(C4) **Mixed discrete-continuous search.** Discovering $\mathcal{M}$ requires jointly selecting states Γ and functional forms for f, h, and g, while estimating their continuous parameters ψ.

Remark. A key contribution of our formulation is that we explicitly include interpretable state representation Γ as an optimization variable, and allow it to contain unobserved latent states.

4 METHOD

Our framework searches over $\mathcal{M} = (\Gamma, f, h, g; \psi)$ using staged LLM proposals, deterministic verification, scientific critique, numerical fitting, and pruning (see Fig. 2).

From MAP to Practical Model Search. The MAP problem (3) is equivalent to

$$\min_{\mathcal{M}} -\log p(\mathcal{D} \mid \mathcal{M}) - \log p(\mathcal{M} \mid \mathcal{C}). \tag{4}$$

The first term measures the data fit. Under Gaussian noises, it is proportional to the normalized mean squared error (NMSE). For the N training trajectories, we have

$$\text{NMSE}_{\text{tr}}(\mathcal{M}) = \frac{1}{d_y \sum_{n=1}^{N} T_n} \sum_{n=1}^{N} \sum_{k=1}^{T_n} \left\| \mathbf{S}^{-1} \left[\widehat{\mathbf{y}}_n(t_{n,k}; \mathcal{M}) - \mathbf{y}_n(t_{n,k}) \right] \right\|_2^2, \tag{5}$$

where d_y is the number of output channels, and $\mathbf{S}$ is a diagonal matrix of standard deviations of output channels in the training set. Predictions $\widehat{\mathbf{y}}_n$ are obtained through h after integrating f from the initial conditions specified by g.

The second term is the negative logarithm of the unknown prior, which can be approximated by

$$p(\mathcal{M} \mid \mathcal{C}) \propto \mathbf{1}\{\mathcal{M} \in \mathcal{R}_{\mathcal{C}}\} \cdot \exp[-\lambda_{\text{spec}}\mathcal{L}_{\text{spec}}(\mathcal{M} \mid \mathcal{C}) - \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}(\mathcal{M})]. \tag{6}$$

Here $\mathcal{R}_{\mathcal{C}}$ denotes the rubrics that determine whether the model is valid, $\mathcal{L}_{\text{spec}}(\mathcal{M} \mid \mathcal{C})$ the scientific plausibility, and $\mathcal{L}_{\text{reg}}(\mathcal{M})$ the model complexity. By mandating the model to pass the deterministic rubric checks, the optimization is equivalent to the following constrained minimization

$$\min_{\mathcal{M} \in \mathcal{R}_{\mathcal{C}}} \text{NMSE}_{\text{tr}}(\mathcal{M}) + \lambda_{\text{spec}}\mathcal{L}_{\text{spec}}(\mathcal{M} \mid \mathcal{C}) + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}(\mathcal{M}). \tag{7}$$

The context-conditioned prior combines model validity requirements, scientific plausibility, and structural parsimony. We operationalize these components through deterministic verification, critic-guided proposal and revision, and pruning, respectively. The critic supplies qualitative scientific

feedback rather than a numerical prior. Consequently, our bounded search is motivated by the MAP formulation but does not directly optimize its complete objective. Next, we discuss key components. The complete algorithm description is in Appendix B.

4.1 Staged Construction of Graph Meta Models — Addressing C1 C2

The LLM proposer selects state specification Γ and supplies analytic interaction functions f, output mappings h, and causal initializers g. Instead of asking the LLM to output (Γ, f, h, g) directly, we represent the model $\mathcal{M}$ as a *graph meta model* $\mathcal{G}$, and let the LLM construct $\mathcal{G}$ in stages.

Definition 1 (Graph Meta Model). A graph meta model $\mathcal{G}$ is a bipartite *hypergraph* (Gallo et al., 1993), with one partition of nodes being the state specification $\Gamma = \{v_1, \ldots, v_{d_x}\}$ and the other partition include *an arbitrary subset* of Γ. An edge is a triple

$$(v \in \Gamma, \mathcal{S} \subseteq \{1, \ldots, d_x\}, \phi), \tag{8}$$

where $\mathcal{S}$ is an index set of dependent state variables (i.e., variables on the right-hand side of the dynamics equation for x) and $\phi(\mathbf{x}_{\mathcal{S}}, \mathbf{u}; \eta)$ is the dependency function parameterized by η.

For state x_i, the dynamics can be represented by

$$\dot{x}_i = f_i(\mathbf{x}, \mathbf{u}; \psi) = \sum_{j \in \mathcal{E}_i} \theta_{i,j} \cdot \phi_{i,j}(\mathbf{x}_{\mathcal{S}_{i,j}}, \mathbf{u}; \eta), \tag{9}$$

where $\mathcal{E}_i$ is the set of edges connecting to v_i with the jth edge $(v_i, S_{i,j}, \phi_{i,j})$ weighted by $\theta_{i,j}$.

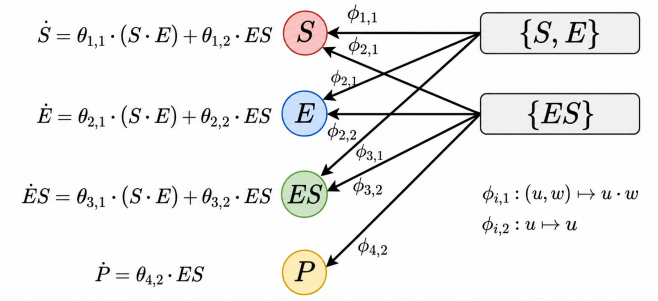


Figure 3: Graph meta model of nonlinear Michaelis-Menten kinetics (Michaelis et al., 1913; Briggs and Haldane, 1925) as a linear combination of candidate functions.

Once the graph meta model is determined, we reconstruct Γ and f. Therefore, we let the LLM proposer construct $\mathcal{G}$ in stages: first the nodes in one partition (i.e., Γ), and then the hyperedges and dependencies (i.e., f). All parameters are $\psi = (\theta, \eta)$.

> **Remark.** Unlike an ordinary graph edge, a hyperedge can jointly encode dependencies on *multiple states*. This preserves nonlinear interactions while keeping the model linear in the candidate function coefficients.

4.2 Parameter Fitting with Conditional Linearity — Addressing C3 C4

To minimize NMSE, we need to find the continuous parameters $\psi = (\theta, \eta)$, consisting of linear coefficients θ of the dependency functions and their parameters η. The challenge here is that the state dynamics may depend on latent states C3. Theoretically, we can roll out the trajectory and calculate $\mathbf{x}_{\text{lat}}(t)$ from the initialization rule g, observed initial values (e.g., $\mathbf{x}_{\text{obs}}(0), \mathbf{u}(0), \mathbf{y}(0)$), and state dynamcis f. But without accurate estimation of parameters in g, the roll-out trajectory of latent states can largely deviate from the ground truth. A well-adopted approach is collocation (Yaari and Dattner, 2018), which treats latent state trajectories as part of the optimization variables and jointly optimizes trajectories $\widehat{\mathbf{x}}_{\text{lat}}$ and parameters ψ. Through equality constraints, the optimized latent trajectories are forced to satisfy the dynamics equations.

Conditional linearity promoted by graph meta model. While the NMSE is non-convex in the parameters C4, *given* the estimated full-state trajectories $\widehat{\mathbf{x}}_n(t)$ and the other parameters η, the equations are affine in the linear coefficients θ. For state x_i, let $m_i = |\mathcal{E}_i|$ and enumerate its interaction functions as $\phi_{i,1}, \ldots, \phi_{i,m_i}$. At evaluation nodes $\{\tau_{n,k}\}_{k=1}^{K_n}$, define the design matrix

$$[\Phi_i^{(n)}]_{k,j} = \phi_{i,j}\left(\widehat{\mathbf{x}}_{n,S_{i,j}}(\tau_{n,k}), \mathbf{u}_n(\tau_{n,k}); \eta\right), \qquad \Phi_i^{(n)} \in \mathbb{R}^{K_n \times m_i}. \tag{10}$$

Stacking the trajectories vertically gives

$$\Phi_i = \text{concat}\left[\Phi_i^{(1)}; \ldots; \Phi_i^{(N)}\right] \in \mathbb{R}^{\sum_{n=1}^{N} K_n \times m_i}. \tag{11}$$

The corresponding response is

$$\dot{\hat{\mathbf{x}}}_i = \text{concat}\left[\dot{\hat{\mathbf{x}}}_i^{(1)}; \ldots; \dot{\hat{\mathbf{x}}}_i^{(N)}\right], \quad [\dot{\hat{\mathbf{x}}}_i^{(n)}]_k = \dot{\hat{x}}_{n,i}(\tau_{n,k}). \tag{12}$$

The problem of finding the optimal linear coefficients is

$$\boldsymbol{\theta}_i^\star \in \arg\min_{\boldsymbol{\theta}_i \in \Theta_i} \|\Phi_i \boldsymbol{\theta}_i - \dot{\hat{\mathbf{x}}}_i\|_2^2. \tag{13}$$

When there is no constraint, we have an analytical solution

$$\theta_i^\star = (\Phi_i^\top \Phi_i + \lambda I)^{-1} \Phi_i^\top \dot{\hat{\mathbf{x}}}_i, \tag{14}$$

where $\lambda > 0$ is a small number introduced for numerical stability.

4.3 Deterministic Validity and Scientific Critique — Addressing C1 C2

Validity gate. Before fitting, the runtime compiles the proposed model and checks a deterministic rubric: restricted expressions and symbol closure, complete generated targets and output mappings, algebraic acyclicity, causal initializers, and declared parameter/sign contracts. Public requirements additionally check, when specified, input-target pathways, mediation through dynamic states, and nonlinear source or feedback dependence. Explicit violations trigger bounded repair. The gate is operational admissibility, not certification of all scientific claims: unresolved scientific predicates remain reported and can remain eligible for development.

Scientific critic. Beyond these executable checks, a judge LLM assesses the plausibility of the proposed mechanisms, variable meanings, and interpretation in context. We reuse the paired GPT-OSS-120B judge: atomic checks inform a comparative review in both candidate orders. It sees the public specification and symbolic models, without fitted values or prediction scores. Its findings provide fallible scientific advice for subsequent revisions; its score neither selects nor vetoes a model and cannot override the validity gate. This separates mechanically checkable requirements from broader scientific interpretation that our deterministic rubric does not establish.

5 Experiments

The experiments test three claims. **Q1: Comparison.** How does our task-specified discovery method compare with data-driven and LLM-guided baselines? **Q2: Ablation and contribution of the specification.** Does the task specification change what the search selects? **Q3: Interventional discrimination.** Does the specification help select a model that responds correctly to input schedules it has never seen? **Q4: LLM Memorization.** Does the LLM memorizes a well-established model or discover the model based on data and task specification?

5.1 Benchmarks and evaluation protocol

We utilize six benchmarks across domains. **Glucose-insulin model**: the canonical model in (Dalla Man et al., 2007). **Obfuscated glucose-insulin model**: the underlying equations are unchanged, but all variable names are replaced by opaque identifiers and the prompt reveals no glucose-insulin context. **Perturbed glucose-insulin model**: the context and variable names are retained, but the governing dynamics are modified from the established model. **Obfuscated and perturbed glucose–insulin model**: variable names are obfuscated, context is removed, and the dynamics are changed. **Obfuscated jacketed exothermic continuous stirred-tank reactor (CSTR)**: adapted from the nonlinear CSTR model used in the chemical process control (Bequette, 1998), which represents a different type of dynamical systems. We replace all variable names with opaque identifiers, so the LLM cannot infer the application domain and retrieve the model. **Procedurally generated "alien-device" system**: this benchmark is generated specifically for our evaluation from a predefined analytic equation grammar, rather than adapted from an established scientific model. The system is presented as telemetry from a fictional extraterrestrial device and is deliberately not assigned a terrestrial physical law. Thus, there is no model that the proposer can retrieve from the literature.

Obfuscated cases test whether the method depends on recognized variable names. Perturbed cases are particularly diagnostic because retrieving the canonical glucose-insulin equations is actually *harmful*: a proposer relying on memorized equations must revise those priors to fit the perturbed ground truth.

Table 2: External comparison on the frozen test split. NMSE is median [MAD]. Compliance is mean (SD).

Method	NMSE ↓	Compliance % ↑
SINDy	2.420 [0.284]	0.0 (0.0)
PySR	2.331 [0.829]	1.9 (5.6)
D3	2.302 [1.462]	42.6 (40.1)
GPT-5.6 agent	0.667 [0.524]	74.1 (32.4)
Ours	**0.397 [0.227]**	**88.9 (33.3)**

Table 3: Ablations under matched budgets and seeds, same statistics as Table 2.

Arm	NMSE ↓	Compliance % ↑
Ours	0.397 [0.227]	88.9 (33.3)
No critic	0.538 [0.285]	80.6 (34.9)
No verifier	0.551 [0.130]	63.9 (37.7)
No critic or verifier	0.491 [0.206]	63.9 (33.3)

Protocol. Parameters are fitted on training trajectories; validation selects the retained model and pruning endpoint. Models, parameters, and causal initializers are frozen before test evaluation. Test trajectories are excluded from proposal, critique, fitting, pruning, and selection. We use GPT-OSS-20B as proposer and GPT-OSS-120B as advisory scientific critic. Each run has an initial construction and fourteen revision visits, with three seeds per case.

Evaluation Metrics. Our primary predictive metric is **test free-rollout NMSE**, normalized by training output variances and averaged over target channels (see Eqn. (5)). Predicted targets are not reset to observations after initialization; permitted auxiliary channels remain available under the task's observation protocol. For **mechanism compliance**, we evaluate explicit predicates on the assembled equations after substituting fitted parameters, *independently* of proposer annotations and scientific critic. See Appendix D.6 for detail. Complexity counts additive terms, with states, parameters, and expression size in the appendix.

Since some models diverge in free rollout, we report NMSEs in medians and unscaled median absolute deviations (MAD), defined as $\mathrm{median}_r\,|z_r - \mathrm{median}_{r'}\,z_{r'}|$. Mechanism compliance is bounded in $[0, 1]$ and is reported in means and standard deviations (SD).

5.2 Comparison with external baselines and Ablation study

Table 2 compares SINDy (Brunton et al., 2016), PySR (Cranmer, 2023), D3 (Holt et al., 2024a), and an end-to-end GPT-5.6 Sol agent. LLM-based methods receive the *same* public task specification.

SINDy and D3 reach nearly the same error (2.420 and 2.302) yet differ in compliance (0.0% and 42.6%). This shows fitting the observed trajectories well does not imply that a model contains task-required mechanisms. SINDy and PySR has low compliance, because both search a fixed library over the observed variables and cannot introduce a new named state, declare a component carrying the required role, or create the dynamic memory a requirement demands. Our own compliance is 85.0%, and the gap is concentrated on the CSTR benchmark, which no method satisfies.

We perform ablations on the deterministic rubric checks (verifier) and LLM scientific critic (critic). These two components act on different endpoints, and separating them is what the compliance column shows most clearly. Removing the critic costs 8.3 points of compliance, from 88.9% to 80.6%, while removing the deterministic verifier costs 25.0%, down to 63.9%; removing both leaves compliance unchanged from removing the verifier alone. The verifier is therefore the binding constraint on whether a candidate satisfies the stated requirements, and the critic contributes a smaller refinement on top of it. Accuracy does not follow the same ordering: the arm with neither component attains a lower median NMSE (0.491) than either single removal (0.538 and 0.551), so relaxing structural constraints buys a small amount of fit while giving up a quarter of compliance.

5.3 Why Task-Specified Model Discovery is Important for Intervention?

We examine the canonical obfuscated glucose-insulin benchmark, where the task is to predict blood-glucose responses to meal timing and amount, using permitted auxiliary measurements. Our held-out intervention changes initial tissue glucose by $\pm 20\%$, while holding initial blood glucose and the meal schedule fixed. Blood and tissues exchange glucose: two simulated subjects with the same initial plasma glucose can therefore evolve differently. See Appendix F for more details.

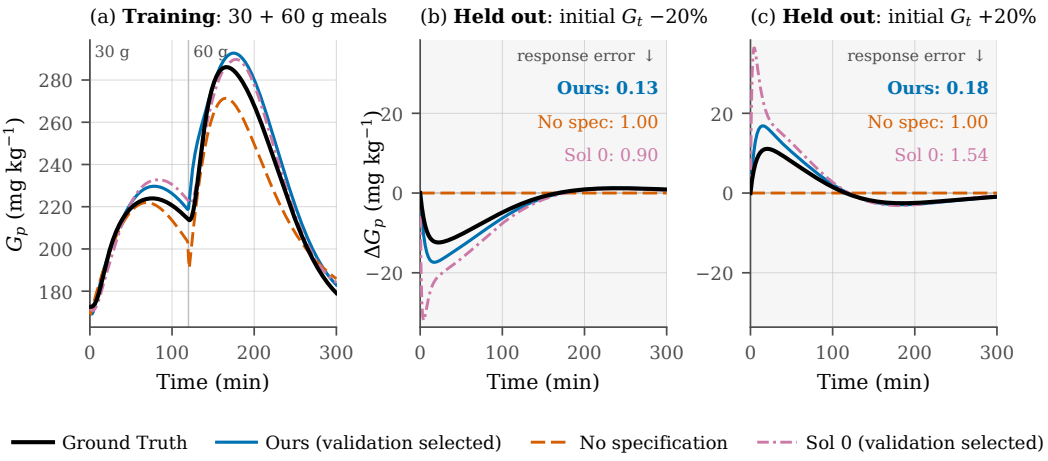


Figure 4: **Comparable training fit does not determine the held-out intervention response.** All models are frozen after validation selection; no parameter is refitted for any held-out condition. (a) On a training trajectory with meals of 30 g at 0 min and 60 g at 120 min, all three models reproduce the observed plasma-glucose response. (b, c) Response to an initial tissue glucose G_t varied by $\pm 20\%$, a condition never varied in training or validation, shown as each model's own difference from its matching fasting control. The no-specification ablation omits the supplied G_t channel entirely and so predicts no change at all, resulting in a flat line at zero. The best-fitting baseline, Sol 0, responds in the right direction but overshoots, exceeding the true effect by roughly threefold at $+20\%$. Our specification-grounded model tracks the reference response most closely in both directions. Inset values give the relative squared response error for the condition shown, where 1.0 is the error of predicting no response. Sol 0 attains lower validation NMSE than our model (0.014 versus 0.058) yet worse response error, illustrating that predictive fit may not imply mechanistic correctness.

Our model, a no-specification model (i.e., removed the task requirements from the prompt while leaving the hypothesis space untouched), and validation-selected Sol 0 have similar training NMSEs (0.0416, 0.0436, and 0.0368). But their equations predict different responses. No-specification ignores tissue glucose and predicts no change under unseen initial conditions. Our model retains a positive tissue-to-blood contribution, while Sol 0 exaggerates the response. For fasting with 20% less initial tissue glucose, normalized errors in the change from each model's own unperturbed control are 0.126 (ours), 1.000 (no-specification), and 0.901 (Sol), respectively; predicting no change scores 1.

6 Conclusion and Discussion

We introduced *task-specified dynamical system discovery under specification-induced partial observability*, a setting in which the desired mechanistic model is defined not only by observed trajectories but also by a natural-language task specification. This creates two coupled challenges: state-specification ambiguity and specification-induced partial observability. Our framework addresses these challenges by treating the state specification itself as an object of discovery. Experiments on various benchmarks show that our method reliably recovers these task-critical mechanisms while achieving better prediction accuracy in unseen, held-out intervention.

Limitations. We assumed the task specification is always correct. An important future direction is to have a framework that can confidently declare possible mismatch between task specification and observational data.

AI Disclosure In this work, we used generative AI tools to generate synthetic data sets, create or modify scientific figures or images, create or edit software code, discover research topics or identify gaps, edit a research paper to improve readability, and identify relevant literature. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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.

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

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K Reddy. LLM-SR: Scientific equation discovery via programming with large language models. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

Joseph Bakarji, Kathleen Champion, J Nathan Kutz, and Steven L Brunton. Discovering governing equations from partial measurements with deep delay autoencoders. *Proceedings of the Royal Society A*, 479(2276):20230422, 2023.

Wei Liu, Kiran Bacsa, Loon Ching Tang, and Eleni Chatzi. Structured Kolmogorov-Arnold neural ODEs for interpretable learning and symbolic discovery of nonlinear dynamics. *arXiv preprint arXiv:2506.18339*, 2025.

Zhaozhi Qian, William Zame, Lucas Fleuren, Paul Elbers, and Mihaela van der Schaar. Integrating expert odes into neural odes: pharmacology and disease progression. *Advances in Neural Information Processing Systems*, 34:11364–11383, 2021.

Cristina Cornelio, Sanjeeb Dash, Vernon Austel, Tyler R Josephson, Joao Goncalves, Kenneth L Clarkson, Nimrod Megiddo, Bachir El Khadir, and Lior Horesh. Combining data and theory for derivable scientific discovery with AI-Descartes. *Nature Communications*, 14(1):1777, 2023.

Ryan Cory-Wright, Cristina Cornelio, Sanjeeb Dash, Bachir El Khadir, and Lior Horesh. Evolving scientific discovery by unifying data and background knowledge with AI Hilbert. *Nature Communications*, 15(1):5922, 2024.

Zelin Guo, Siqi Wang, Yonglin Tian, Jing Yang, Hui Yu, Xiaoxiang Na, Levente Kovács, Li Li, Petros A Ioannou, and Fei-Yue Wang. SR-LLM: An incremental symbolic regression framework driven by LLM-based retrieval-augmented generation. *Proceedings of the National Academy of Sciences*, 122(52):e2516995122, 2025.

Amirmohammad Ziaei Bideh and Jonathan Gryak. LLM-ODE: Data-driven discovery of dynamical systems with large language models. *arXiv preprint arXiv:2603.20910*, 2026.

Evgeny Saveliev, Samuel Holt, Nabeel Seedat, David L. Bentley, Jim Weatherall, and Mihaela van der Schaar. Influence-guided symbolic regression: Scientific discovery via LLM-driven equation search with granular feedback. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ujiUua1gGV.

Samuel Holt, Zhaozhi Qian, Tennison Liu, Jim Weatherall, and Mihaela van der Schaar. Data-driven discovery of dynamical systems in pharmacology using large language models. *Advances in Neural Information Processing Systems*, 37:96325–96366, 2024a.

Samuel Holt, Tennison Liu, and Mihaela van der Schaar. Automatically learning hybrid digital twins of dynamical systems. *Advances in Neural Information Processing Systems*, 37:72170–72218, 2024b.

David Krongauz, Arad Zulti, Eran Segal, and Teddy Lazebnik. Automatic ordinary differential equations discovery for biological systems using large language model powered agentic system. *arXiv preprint arXiv:2607.13608*, 2026.

Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42(2-3):177–201, 1993.

Leonor Michaelis, Maud L Menten, et al. Die kinetik der invertinwirkung. *Biochem. z*, 49(333-369): 352, 1913.

George Edward Briggs and John Burdon Sanderson Haldane. A note on the kinetics of enzyme action. *Biochemical journal*, 19(2):338, 1925.

Rami Yaari and Itai Dattner. simode: R package for statistical inference of ordinary differential equations using separable integral-matching. *arXiv preprint arXiv:1807.04202*, 2018.

Chiara Dalla Man, Robert A Rizza, and Claudio Cobelli. Meal simulation model of the glucose-insulin system. *IEEE Transactions on biomedical engineering*, 54(10):1740–1749, 2007.

B.W. Bequette. *Process Dynamics: Modeling, Analysis, and Simulation*. Prentice-Hall international series in the physical and chemical engineering sciences. Prentice Hall PTR, 1998. ISBN 9780132068895. URL https://books.google.it/books?id=b4wvAQAAIAAJ.

A Extended Related Works

Equation discovery over supplied variables. Classical equation-discovery methods search for compact mathematical models over variables *specified before learning*. SINDy identifies sparse governing equations from measurement data (Brunton et al., 2016), while symbolic regression such as PySR broadens the functional search space through genetic or evolutionary search (Cranmer, 2023). These methods are effective when the variables needed for the target dynamics are measured or supplied through feature construction. Their standard formulations do not determine which additional dynamical states a scientific task requires. Our setting makes this representation choice part of discovery: the task specifies mechanisms to expose, while the state-space realization and governing equations remain unknown.

Dynamical discovery under partial observation. Learning latent dynamics is not itself new. Latent ODEs infer continuous-time hidden representations from observations (Rubanova et al., 2019), and deep delay autoencoders combine delay embeddings, learned coordinate transformations, and sparse identification to recover explicit dynamics from partial measurements (Bakarji et al., 2023). These approaches establish that governing equations and effective coordinates can be learned jointly without full state observations. Their coordinates, however, are not required to realize particular mechanisms specified by a natural-language scientific task.

Other approaches provide stronger physical semantics. SKANODE prescribes a structured latent representation, such as displacement and velocity, and learns dynamics from indirect measurements before extracting symbolic equations (Liu et al., 2025). Expert-neural ODE hybrids likewise incorporate unobserved variables with expert-defined mechanistic roles (Qian et al., 2021). Our distinction is not simply physical interpretability: we seek the state representation needed to realize task-required mechanisms rather than requiring that representation or its governing expert equations to be supplied beforehand.

Scientific knowledge and executable constraints. AI-Descartes (Cornelio et al., 2023) and AI Hilbert (Cory-Wright et al., 2024) combine empirical data with formal background knowledge to derive or validate scientific formulae. They demonstrate that theory can distinguish equations with similar empirical error; AI Hilbert also accommodates unmeasured variables within the supplied theory. These methods reason over an already formalized vocabulary and axioms. Our focus is the realization problem for partially observed dynamics: scientific requirements specify functional roles, but the persistent states and equations implementing them must still be constructed. Executable checks assess the resulting model against supported requirements; this is distinct from proving an unrestricted scientific law.

LLM-guided equation and dynamical discovery. LLM-SR (Shojaee et al., 2025) searches equation programs using scientific priors and numerical parameter optimization; SR-LLM (Guo et al., 2025) incorporates retrieval-augmented guidance into symbolic regression; and LLM-ODE (Bideh and Gryak, 2026; Holt et al., 2024a) extracts patterns from candidate equations to guide symbolic evolution. These methods search equations over supplied variables rather than jointly reconstructing task-required persistent hidden states. IGSR (Saveliev et al., 2026) introduces a complementary improvement: granular term-level feedback and deterministic influence-based pruning within iterative equation search. Its default selector evaluates predictive contribution, whereas our mechanism checks evaluate whether candidate state spaces and equations realize specified scientific relationships. Thus deterministic numerical selection and executable task compliance address different questions.

Agentic construction and knowledge-guided formalization. D3 (Holt et al., 2024a) coordinates model construction, feature acquisition, and evaluation. Its additional features are acquired information used to improve modeling, rather than a general procedure for reconstructing unmeasured persistent dynamical states. HDTwinGen Holt et al. (2024b) searches mechanistic-neural model structures and explicitly incorporates natural-language modeling requirements. These systems show that language can guide model design and critique, not merely initial proposals. Our focus is the narrower combination of task-dependent latent-state realization under partial observation and explicit checking of the required mechanisms. MEDA (Krongauz et al., 2026) formalizes biological knowledge into variables and typed constraints and uses constrained ODE search with hard-constraint gating. However, it does not reconstruct variables absent from the measurements, although it evaluates the robustness of the method when some state variables become unobserved.

B Detailed Algorithm and Testing Protocol

Table 4 describes the full method, applied independently to each prescribed seed and prompt variant. Component ablations disable only their named scientific verifier, critic, shared-process stage, or pruning step; parsing, execution safety, causal boundaries, and target completeness remain. Every LLM request and numerical attempt is cached and checkpointed. Resume reuses completed work and consumed budgets. Failures and unresolved checks remain explicit in the reported denominators.

Let E be validation NMSE from Eq. (5), and K_t the number of outer additive terms in state and process equations. Search prefers smaller E when parent and trial differ by more than $\delta = 10^{-8} + 10^{-6}\max(|E_{\text{parent}}|, |E_{\text{trial}}|)$; within δ, it prefers smaller K_t, retaining the parent on a tie. Pruning uses the count vector $K_v = (\text{states}, \text{processes}, \text{parameters}, \text{terms}, \text{expression nodes})$ and $E_b = \min(E_{\text{parent}}, E_{\text{control}})$ when the control is available. It requires componentwise $K_v(\text{pruned}) \le K_v(\text{parent})$, at least one strict decrease, and $E_{\text{pruned}} \le E_b + \max(10^{-6}, 0.01E_b)$. A failed control remains an incomplete paired comparison; a valid child can still compete with the historical parent. In the table, RMS is the square root of the mean squared value over training-rollout samples.

Table 4: Bounded MAP-motivated model search and process-aware pruning.

	Input: context $\mathcal{C}$, training and validation data, prescribed lineages, visit/retry/fit budgets, component settings.
	Output: frozen models $\mathcal{M} = (\Gamma, f, h, g; \psi)$, or explicit unavailable outcomes, with metrics, diagnostics, and costs.
1	Fit target scales σ_q on training data. Keep test data inaccessible.
2	**For each** prescribed lineage, initialize the incumbent as unavailable.
3	**For** each allowed search visit:
4	Initially propose variables, optional shared processes, signed topology, analytic functions and initializers. If the optional route fails, permit ordinary construction within the same budget.
5	Otherwise revise using the incumbent, public requirements, training residuals and critic advice; without an incumbent, use bounded unfitted repair or record closure.
6	Compile, preserve shared identities and apply unambiguous mechanical repairs. Enforce the validity rubric; return failures for bounded proposer repair.
7	If no admissible draft remains, retain the incumbent with any prescribed unchanged-refit fallback. Without an admissible model, record failure and skip fitting.
8	Obtain or reuse the enabled critic's paired symbolic review against the incumbent when available. Save advice without selecting or vetoing the candidate.
9	Fit ψ, including initializer parameters, on training data using collocation and bounded rollout refinement with compatible warm starts.
10	Evaluate training and validation rollouts with frozen parameters and training scales; record numerical failures.
11	Select the eligible finite candidate by validation error, then additive-term count using δ above; otherwise preserve the incumbent.
12	Checkpoint the model, training feedback, review linkage and costs. Do not pass validation or test scores to the proposer.
13	**If** pruning is enabled and an incumbent is available:
14	Enumerate legal deletions preserving outputs, causality and shared consumers. Skip deletion for unresolved public obligations or if no legal candidate remains.
15	On the parent's training rollout, compute $r(T) = \text{RMS}(T)/\max\{\text{RMS}(F_i), 10^{-12}\}$ for contribution T to equation F_i. Use the maximum consumer ratio for coupled units. Freeze the weakest legal deletion before refitting; break ties by stable identifier.
16	Delete that unit and unreachable declarations; recheck validity. Refit the child and unchanged parent with equal budgets. A skipped deletion can still receive its prescribed control.
17	Select parent, control or child by the rule above. Review changed selected equations with the enabled critic, without altering selection.
18	Report every endpoint, including failures. Freeze selected models; only then evaluate held-out test data without refitting or model changes.

C Deriving Importance Score in Pruning

C.1 Details on Pruning

A central component in the construction of parsimonious models from data is the reduction of expression complexity. The complexity of an equation, in particular, strongly depends on the number of terms; among a set of equations with same accuracy on unseen data a symbolic regression engine should favor the simpler ones, that is the ones having less additive terms and less depth in their computational graph. Since in our framework we act on a flat hierarchy of symbolic terms (all terms are combined linealy through equation parameters) the natural quantity of interest for characterizing the complexity of a generated expression is therefore the number of terms.

In order to remove superfluous and spurious terms from a model, we adopt a greedy pruning strategy. For a given dynamical equation F_i of the form

$$F_i = \sum_j^{N_i} f_{i,j}$$

where $f_{i,1}...f_{i,N_i}$ are functional terms constructed through our procedure, we define RMS-importance of a functional term as

$$r(f_{i,j}) = \sqrt{\frac{\int dt\ f_{i,j}^2(t)}{\int dt\ F_i^2(t)}}$$

and use it to rank terms within the different equations $\{F_i\}_i$. Geometrically, we interpret r as a norm based measure of importance; being $\|F_i\|^2_{L^2(\mathbb{R})} = \sqrt{\int dt\ F_i^2(t)}$ and being $F_i = f_{i,1} + ... + f_{i,N_i}$ the metric r measures the contribution of each term relatively to the norm of the predicted function. Alternatively, if one defines $F_i(\boldsymbol{\alpha}) = \sum_i \alpha_j f_{ij}$ and evaluate its log derivatives in $\boldsymbol{\alpha} = \mathbf{1}$ then we recover exactly

$$\begin{aligned}
\partial_{a_j} \log \|F_i(\boldsymbol{\alpha})\|^2_{L^2(\mathbb{R})}\Big|_{\boldsymbol{\alpha}=\mathbf{1}} &= 2\frac{\int dt\ F_i(t) f_{i,j}(t)}{\int dt\ F_i^2(t)} \\
&= 2\frac{\langle F_i, f_{i,j}\rangle_{L^2}}{\langle F_i, F_i\rangle_{L^2}} \\
&\leq 2\frac{\sqrt{\langle F_i, F_i\rangle_{L^2}\langle f_{i,j}, f_{i,j}\rangle_{L^2}}}{\langle F_i, F_i\rangle_{L^2}} \\
&= 2\frac{\sqrt{\langle f_{i,j}, f_{i,j}\rangle_{L^2}}}{\sqrt{\langle F_i, F_i\rangle_{L^2}}} \\
&= 2r(f_{i,j})
\end{aligned}$$

which allows us to identify the introduced metric as a measure of the sensitivity of the functional form variation to the removal of the term, by a simple Taylor-Expansion argument.

Once components are removed, and a new sparsity pattern is available for the model optimality is not in general preserved. We thus employ iterative re-fitting; in this way, we can obtain a sequence of iterative refinement steps, where the model is firstly projected into a lower complexity equation subspace and then optimized for the observed data.

D BENCHMARK TASKS, DATA ACCESS, AND REFERENCE DYNAMICS

The experiments use four modeling tasks from three system families: meal-response modeling (T1) and absorption–action disentanglement (T2) in a glucose–insulin system, a controlled chemical reactor, and an unfamiliar synthetic device. A task specifies the mechanisms a model must represent; an observability tier specifies which trajectories it may use. These are distinct from the private simulator's state dimension or a requirement to reproduce its exact equations.

The main comparison contains nine cases: canonical and perturbed T1-easy, each in named and obfuscated form (four cases); canonical named T1-hard, T2-easy, and T2-hard (three cases); named CSTR-easy; and functional-description alien-device easy. Thus, the nine cases represent three physical or synthetic families, not nine independent systems. Only the tiers and variants used in these experiments are described below.

D.1 SHARED DATA AND EVALUATION CONTRACT

Each numerical case has 16 training, four validation, and six test trajectories. Equations and fitted parameters are shared across trajectories. Training data support parameter estimation and proposal feedback; validation data support model selection. Test trajectories are reserved for evaluation after models are frozen. New initial conditions and input schedules define distinct trajectories, rather than splitting a single trajectory into adjacent windows.

Targets, auxiliaries, and external inputs. A *target* is a quantity the model must generate. During held-out free rollout, its initial observation is available, but later target observations cannot be used to correct the simulated state. A *supplied auxiliary* is an additional measured trajectory available throughout the prediction horizon under the public task contract. It may enter a model's equations directly without being assigned its own ODE. An *external input* is an imposed command or event schedule. All methods receive the same channel permissions within a case. Latent initialization must use permitted initial information; latent initial values are not fitted to held-out target trajectories.

Table 5 lists the named channels. Obfuscated cases retain the numerical data and channel roles, but replace channel names with neutral symbols. The auxiliary contract makes the rollout *conditional* on those supplied trajectories: it does not constitute autonomous recovery of every subsystem of the reference simulator.

Table 5: Public channel roles in the six task–tier configurations used in the nine-case study. m denotes the meal-event channel and u_I the imposed insulin forcing. A supplied auxiliary is available throughout a rollout; a target is predicted without later observed-value resets.

Task/tier	Targets	Supplied auxiliaries	External inputs
T1 easy	G_p	EGP, U_{ii}, E, G_t	m
T1 hard	G_p	G_t	m
T2 easy	G_p, I, U	EGP, U_{ii}, E, G_t	m, u_I
T2 hard	G_p, I	U_{ii}	m, u_I
CSTR easy	T	C, T_j	C_f, T_f, T_{jf}
Alien easy	v_{01}	v_{02}, v_{03}	u_{01}

Public prompt structure. The prompts have six sections: task specification, available data, modeling requirements, constraints and plausibility, intended use, and required response. They ask for explicit analytic dynamics, a finite-dimensional causal representation of the required mechanisms, globally shared parameters, and causal initialization. Every modeled state needs a governing equation, every generated process an analytic definition, and every target an observation or governing equation. Free fitted time series, future-observation dependence, and unrestricted neural-network vector fields are disallowed. The prompts request coherent source/sink signs and declared state constraints, but allow signed deviations and net rates. They do not impose a particular latent-state count. The task-specific mechanism clauses are quoted below; the reference equations in this appendix are explanatory and are not supplied to discovery methods. Scientific critique and deterministic verification follow the common methodology, rather than benchmark-specific judge prompts.

D.2 GLUCOSE–INSULIN SYSTEM: T1 AND T2

Reference system and notation. We generate glucose–insulin responses using a trusted implementation based on Dalla Man et al. (2007). The reference has 12 dynamic states: plasma and tissue glucose masses G_p, G_t; three gastrointestinal compartments $Q_{\mathrm{sto1}}, Q_{\mathrm{sto2}}, Q_{\mathrm{gut}}$; plasma, liver, and portal insulin quantities I_p, I_l, I_{po}; a secretion-response state Y; two delayed insulin signals I_1, I_d; and a remote insulin-action state X. This dimension belongs to the simulator, not to the required discovered model. In particular, the public glucose target is mass G_p in $\mathrm{mg\,kg}^{-1}$, not concentration $G = G_p/V_G$. The insulin target is concentration $I = I_p/V_I$ in $\mathrm{pmol\,L}^{-1}$.

Relevant reference mechanisms. The glucose balances separate production, meal appearance, utilization, excretion, and exchange:

$$\dot{G}_p = EGP + R_a - U_{ii} - E - J_{p\to t} + J_{t\to p}, \tag{15}$$

$$\dot{G}_t = -U_{id} + J_{p\to t} - J_{t\to p}. \tag{16}$$

In the canonical system, $J_{p\to t} = k_1 G_p$ and $J_{t\to p} = k_2 G_t$. Meal absorption is generated by

$$\dot{Q}_{\mathrm{sto1}} = -k_{\mathrm{gri}} Q_{\mathrm{sto1}}, \tag{17}$$

$$\dot{Q}_{\mathrm{sto2}} = k_{\mathrm{gri}} Q_{\mathrm{sto1}} - k_{\mathrm{empt}} Q_{\mathrm{sto2}}, \tag{18}$$

$$\dot{Q}_{\mathrm{gut}} = k_{\mathrm{empt}} Q_{\mathrm{sto2}} - k_{\mathrm{abs}} Q_{\mathrm{gut}}, \qquad R_a = \frac{f k_{\mathrm{abs}} Q_{\mathrm{gut}}}{BW}. \tag{19}$$

Here k_{empt} is the reference's stomach-content-dependent emptying rate. Each meal of D_j grams at time t_j produces the reference jump $Q_{\mathrm{sto1}}(t_j^+) = Q_{\mathrm{sto1}}(t_j^-) + 1000 D_j$. The public channel `meal_event_g` records event amounts in grams; it is not the unobserved appearance flux R_a.

Insulin-dependent disposal has dynamic memory:

$$\dot{X} = -p_{2U} X + p_{2U}(I - I_b), \tag{20}$$

$$U_{id} = \left[\frac{(V_{m0} + V_{mx} X) G_t}{K_m + G_t}\right]_+, \qquad U = U_{ii} + U_{id}, \qquad U_{ii} = F_{cns}, \tag{21}$$

where $[a]_+ = \max(a, 0)$. The remote action X is a deviation signal; it is not constrained to be nonnegative merely because glucose and insulin masses are nonnegative. Other reference contributions include

$$EGP = [k_{p1} - k_{p2} G_p - k_{p3} I_d - k_{p4} I_{po}]_+, \tag{22}$$

$$E = k_{e1}[G_p - k_{e2}]_+. \tag{23}$$

The remaining insulin and secretion dynamics close the reference system. For example, its plasma-insulin balance is

$$\dot{I}_p = -(m_2 + m_4) I_p + m_1 I_l + u_I, \qquad I = I_p/V_I, \tag{24}$$

where the imposed forcing u_I has units $\mathrm{pmol\,kg}^{-1}\,\mathrm{min}^{-1}$. These equations describe reference mechanisms, not a prescribed skeleton or an equation-by-equation recovery target for T1 or T2.

D.2.1 T1: MEAL-RESPONSE MODELING

Task and use. T1 asks how meal timing and amount influence plasma glucose. Its intended use is a reduced meal-response simulator: given an intake schedule and the permitted auxiliaries, it should predict the glucose response under a new schedule. The named public requirement is:

> "a causal mechanism through which meal timing and meal amount contribute to the observed plasma-glucose mass."

The prompt does not prescribe the three gastrointestinal compartments, a variable named R_a, or any fixed number of latent states. Nor does it require hepatic regulation, insulin-action dynamics, or a separately generated G_t state. Such mechanisms may be useful, but their absence alone is not a violation of the T1 task specification.

Observability. T1-easy predicts G_p with EGP, U_{ii}, E, G_t supplied. T1-hard also predicts G_p, but only G_t is supplied. Thus, hard means fewer auxiliaries, not complete absence of auxiliary measurements. A candidate may choose which auxiliaries to use, provided it explains their roles and respects the public availability contract. Both tiers receive the meal-event channel.

D.2.2 T2: ABSORPTION–ACTION DISENTANGLEMENT

Task and use. T2 extends meal-response modeling to distinguish glucose entering circulation from glucose removal regulated by insulin. This supports interpretation of post-meal excursions: delayed absorption and delayed insulin action can produce different responses to a changed meal or imposed insulin input. The named public requirements are:

> "a causal pathway through which meal timing and meal amount contribute to the observed plasma-glucose mass"; and
>
> "a delayed insulin-action pathway through which plasma insulin regulates the insulin-dependent contribution to total glucose disposal $U(t)$."

Unlike T1's meal clause, the second clause explicitly requires delayed action. An instantaneous insulin-dependent coefficient alone does not supply that memory. The task does not require the reference coordinate X or its exact functional form, and a graph path alone does not establish the correct fitted response or a unique physical interpretation.

Observability. T2-easy must generate three targets, G_p, I, U, with EGP, U_{ii}, E, G_t supplied. T2-hard must generate G_p, I, with only U_{ii} supplied; total disposal is then an internal mechanism rather than an observed target. In both tiers, the declared external inputs are the meal-event channel and the imposed insulin forcing. *Plasma insulin I is a prediction target, not a supplied future trajectory.* The model must therefore generate the insulin signal used by its delayed-action pathway. U may be an analytic generated process rather than an additional dynamic state. The public task does not explicitly prescribe a meal-to-insulin or glucose-to-insulin edge; such dependencies must be proposed from the available evidence rather than imposed as hidden reference facts.

D.2.3 DYNAMICS AND NAMING CONTROLS

The T1-easy variants separate two changes. *Obfuscation* replaces physiological names with neutral channel identifiers and removes informative units while preserving numerical trajectories, splits, and channel permissions. It tests dependence on semantic cues; it does not by itself establish whether an LLM memorized a model. The public task still describes the required causal input-to-output mechanism.

The *perturbed dynamics* change the two plasma–tissue exchange laws to

$$J_{p\to t} = k_1 G_p \left[1 + 0.35 \tanh\left(\frac{G_p - G_{p,b}}{0.20 G_{p,b}}\right)\right], \tag{25}$$

$$J_{t\to p} = k_2 G_t \left[1 - 0.25 \tanh\left(\frac{G_t - G_{t,b}}{0.20 G_{t,b}}\right)\right]. \tag{26}$$

The basal masses are $G_{p,b}$ and $G_{t,b}$. Both multipliers remain positive and equal one at basal equilibrium, preserving exchange directions and the basal balance while introducing state dependence. This perturbation does not change the gastric-emptying or remote-insulin-action laws. The same exchange flux appears with opposite signs in the two balances. The study uses the four canonical/perturbed and named/obfuscated combinations for T1-easy; T1-hard and T2 use the canonical named condition.

Trajectory schedules. Dalla Man trajectories last 300 minutes with one-minute sampling. Training includes basal trajectories, single meals of 30–150 g, delayed meals, two-meal schedules with 60–120-minute separation, and shifts in initial glucose or insulin. T1 adds three separated meals; T2 adds an imposed insulin pulse. The initial G_p shifts in this protocol are paired with compensating G_t changes, rather than independent changes of both compartments. Validation uses intermediate meal amounts and timings, a new two-meal schedule, and a combined initial-state shift. Its T2 schedules contain no imposed insulin pulse. The six test protocols include smaller/larger meals, a later meal, a multi-meal schedule, a combined initial-state extrapolation, and an additional task-specific schedule:

four meals for T1 or a stronger insulin pulse for T2. Post-hoc case-study interventions, such as shorter meal spacing or independently changing $G_t(0)$, are reported separately from this fixed test suite.

D.3 Controlled continuous stirred-tank reactor (CSTR)

Task and use. A continuously stirred-tank reactor mixes an incoming reactant stream while a surrounding jacket exchanges heat with the reactor. Reactor temperature may change because the feed temperature changes, an exothermic reaction releases heat, or heat moves between the reactor and jacket. A useful model must distinguish these contributions to support prediction under changed operating conditions. The public task requires:

> "a reactor-temperature balance that distinguishes feed transport, reaction heat generation, and heat exchange with the jacket."

The evaluated named easy case predicts reactor temperature T. Reactant concentration C and jacket temperature T_j are supplied auxiliaries; feed concentration C_f, feed temperature T_f, and jacket-feed temperature T_{jf} are external inputs. Extra latent states are not compulsory when a candidate can express the required balance using the available channels.

Reference dynamics. The three-state reference is

$$r(C,T) = k_0 e^{-E_R/T} C, \tag{27}$$

$$\dot{C} = q(C_f - C) - r(C,T), \tag{28}$$

$$\dot{T} = q(T_f - T) + \beta_r r(C,T) - h(T - T_j), \tag{29}$$

$$\dot{T}_j = q_j(T_{jf} - T_j) + h_j(T - T_j). \tag{30}$$

Here q, q_j are flow rates, β_r converts reaction rate to heat generation, and h, h_j govern heat exchange. These positive coefficients define the reference source/sink roles. The implemented reaction law uses $\max(C, 0)$ and $\max(T, 250)$ in place of C and T for numerical protection. Temperature is measured in kelvin and concentration in the supplied relative units. The reference parameters are $k_0 = 7.2 \times 10^{10}$, $E_R = 8750$, $q = 1$, $\beta_r = 80$, $h = 2.5$, $q_j = 1.5$, and $h_j = 1.5$.

Trajectory schedules. Trajectories span 30 time units at interval 0.1. Normalized input commands (u_C, u_T, u_J) are mapped to physical inputs by

$$C_f = 1 + 0.25u_C, \qquad T_f = 350 + 10u_T, \qquad T_{jf} = 330 + 12u_J. \tag{31}$$

Training uses zero input, single-input steps and pulses, slow sinusoids, paired input changes, and an initial-state shift. Validation uses intermediate amplitudes and durations. The test design includes amplitude extrapolation, faster oscillations, a delayed pulse, opposing feed/jacket temperature changes, and initial-state extrapolation. In the easy tier, a changed physical input may also change the supplied auxiliary trajectories; these remain part of the conditional prediction contract.

Reference-data qualification. A diagnostic audit of a historical CSTR export found that its adaptive integrator skipped some finite-duration forcing windows. In particular, a 350-to-345 K feed-temperature pulse from time 10 to 15 was accompanied by an incorrectly flat saved temperature. Reference integrations split at the forcing boundaries instead produced a temperature decrease of approximately 4.36 K. Consequently, scores against affected exports describe agreement with those stored arrays and cannot establish correct physical intervention behavior. Any corrected release and matched reruns must be identified separately; the task definition above does not certify the numerical integrity of a historical export.

D.4 Alien device: input-driven memory in an unfamiliar system

Task and use. The alien device is a procedurally generated nonlinear system without a named physical interpretation. It tests whether a model can explain a persistent response to an external command using measured telemetry, without relying on a familiar domain equation. The evaluated functional-description easy prompt requires:

"input-driven memory and its causal contribution to the observed output."

The public target is v_{01}, the command is u_{01}, and v_{02}, v_{03} are two supplied telemetry channels selected by a simulator-side training-design sensitivity analysis. The prompt states the required function of memory but withholds the reference graph, coefficients, state identities, and dimension. It does not explicitly require the complete nonlinear-feedback architecture of the private reference.

Reference dynamics. The selected reference has five internal coordinates $z \in \mathbb{R}^5$ and a dynamic output y. Its construction has the form

$$\dot{z}_i = -d_i z_i + \sum_j A_{ij} z_j + \sum_\ell a_{i\ell} \tanh(s_{i\ell} z_{j_{i\ell}} + b_{i\ell}) + \sum_\ell c_{i\ell} \tanh(p_{i\ell} z_{r_{i\ell}}) \tanh(q_{i\ell} z_{s_{i\ell}}) + B_i \tanh(s_u u), \tag{32}$$

$$\dot{y} = -d_y y + \sum_\ell a^y_\ell \tanh(s^y_\ell z_{j_\ell}) + \sum_\ell c^y_\ell \tanh(p^y_\ell z_{r_\ell}) \tanh(q^y_\ell z_{s_\ell}). \tag{33}$$

The coefficients and interaction indices are frozen in the selected generator; A is skew-symmetric and the d_i and d_y are positive decay rates. The architecture combines relaxation, internal coupling, bounded nonlinear interactions, and command-driven forcing. Only two internal coordinates are exposed as telemetry in the easy tier; their public names do not disclose their private roles. Discovery is not required to reproduce all five internal coordinates. This benchmark supplies an unfamiliar task, but neither procedural generation nor neutral names prove that every constituent mathematical pattern is absent from an LLM's training data.

Trajectory schedules. Trajectories span 60 time units at interval 0.1. Training includes positive and negative pulses, steps, sinusoids, a chirp, separated pulse sequences, and initial-state shifts. Validation uses intermediate pulse amplitudes/durations and sinusoidal periods. Test protocols change amplitude, frequency, pulse timing, or initial conditions beyond the training design. The model uses one frozen parameter vector across these schedules. The relevant question is whether its input-driven memory continues to generate the output response, not whether it reproduces the simulator's private coordinate system.

D.5 Baseline Methods

We compare our strategy against three established symbolic regression approaches:

SINDy (Sparse Identification of Nonlinear Dynamics). SINDy (Brunton et al., 2016) constructs a library of candidate functions and uses sparse regression (typically LASSO or sequentially thresholded least squares) to identify active terms. We use a polynomial library up to degree 2 with $\tanh$ basis functions where appropriate. SINDy is highly effective when the true dynamics lie within the predefined function library but cannot adapt its basis to novel functional forms.

PySR (Genetic Programming Symbolic Regression). PySR (Cranmer, 2023) evolves mathematical expressions using genetic programming with a multi-population island model. We configure it with binary operators $\{+, -, \times, /\}$ and unary operators $\{\tanh, \exp, \text{neg}\}$, running for 40 iterations per output variable. PySR can discover arbitrary compositions of operators but requires significant computation and may overfit without careful regularization.

D3 (Data-Driven Discovery). D3 (Holt et al., 2024a) uses neural network-guided search with confidence interval-based model selection. We run D3 with its default configuration, which includes automatic complexity regularization.D3 balances expressiveness with interpretability through its probabilistic model selection framework.

D.6 Evaluation of Mechanism Compliance

We evaluate whether a fitted model satisfies operational tests derived from the mechanisms requested in its public specification. The assessment uses the candidate's executed equations and fitted parameters, rather than its textual explanations or mechanism annotations. Graph connectivity is necessary

for a claimed pathway, but is insufficient to establish compliance. The rubric is applied uniformly across methods, without refitting, LLM calls, or changes to model selection. It was developed after inspecting preliminary results and constitutes a post-hoc operational assessment. This evaluation is *independent* of the deterministic rubric checks and LLM critic.

Requirements and outcomes. For benchmark b, let $\mathcal{R}_b$ contain its public mechanism requirements. Each requirement receives one of three outcomes: *pass*, *fail*, or *unresolved*. A pass requires affirmative evidence from all applicable checks; a resolved counterexample yields a fail. Numerical uncertainty, unsupported representations, or insufficient evidence yield an unresolved outcome. A locally inactive response is not itself a counterexample, since nonlinear mechanisms may be inactive at particular operating points. However, confirmation requires at least one active operating point and no unresolved or failing applicable checks. An absent effective equation pathway fails the corresponding pathway requirement.

Operating points. Parameters and initial-value mappings are held fixed. We use up to three training trajectories, selected by lexicographic trajectory identifier, and evaluate the initial, middle, and final samples of each model's free rollout. Generated states are not reset to subsequent observations; supplied inputs and auxiliaries follow the model's execution convention. Operating-point selection does not depend on predictive error or assessment outcomes.

Numerical probes use training-channel standard deviations, floored at 10^{-6}, to normalize public drivers and readouts. Generated-state perturbations use scales $\max(1, |z_i|)$, and continuous time is normalized by the trajectory's median sampling interval. No held-out test trajectories are used.

Effective response and dynamic mediation. At each operating point, we hold other supplied channels fixed and linearize the candidate with respect to a public driver d:

$$\delta\dot{\mathbf{z}} = A\,\delta\mathbf{z} + B\,\delta d, \qquad \delta q = H\,\delta\mathbf{z} + J\,\delta d, \tag{34}$$

where $\mathbf{z}$ contains the remaining, unclamped generated states and q is the required output or balance-rate readout. The direct coefficient J and dynamic coefficients

$$HB, \quad H\bar{A}B, \quad \ldots, \quad H\bar{A}^{n-1}B, \qquad n = \dim(\mathbf{z}), \tag{35}$$

measure effective local coupling. Here $\bar{A} = A/\alpha$, with $\alpha \geq 1$ a positive conditioning factor; this scaling preserves the signs and zeros of the coefficients. Evaluating the complete equations allows cancellation between opposing pathways to affect the result.

A causal-response requirement needs a numerically resolved direct or dynamic coefficient. A memory or delayed-action requirement needs a resolved dynamic coefficient: direct feedthrough alone is insufficient. It also fails if, after clamping, no equation pathway connects the driver to the readout through a remaining dynamic state. When a direction is specified, we test the sign of the first resolved coefficient, considering J first for causal-response requirements and only the dynamic coefficients for memory requirements. These tests establish local behavior, not global monotonicity or a particular physical delay.

For D3, we preserve its native recurrence,

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \Delta_{\mathcal{M}}(\mathbf{z}_k, \mathbf{u}_k), \tag{36}$$

and use $A = I + \partial\Delta_{\mathcal{M}}/\partial\mathbf{z}$ on the unclamped dynamic coordinates. The increment is neither interpreted as an ODE derivative nor multiplied by the sampling interval. Saved auxiliary equations that are not executed in the native rollout do not supply mechanism evidence.

Benchmark-specific operational tests. The nine benchmark cases contain eleven mechanism requirements. Multiple checks of one requirement are combined conjunctively, rather than counted as separate mechanisms.

- **Named Dalla Man T1 cases.** The meal input must have an effective response in plasma glucose, with a positive first resolved local response. The same meal-response test applies in T2. This direction operationalizes the named meal-source role.
- **Anonymous T1 cases.** The specified input must effectively influence the target. No physiological polarity is imposed when the public specification leaves it unspecified.

- **Dalla Man T2-easy.** Plasma insulin must have a positive, dynamically mediated influence on glucose disposal. The generated plasma-insulin coordinate is clamped during differentiation, so its own dynamics cannot supply the required mediation.
- **Dalla Man T2-hard.** We test a dynamically mediated negative influence of insulin on the glucose balance rate, clamping both plasma insulin and glucose storage. This prevents glucose accumulation alone from appearing as delayed insulin action. Passing this proxy leaves the disposal requirement *unresolved*: decreased glucose production and increased disposal cannot be distinguished from this response alone. A resolved wrong-direction response fails the proxy.
- **CSTR.** In the public temperature coordinates, we test positive feed and jacket heating directions, zero balance at equal temperatures with zero reactant, positive reaction heat with reactant present, and conditional additivity of the three heat channels. Additivity is checked using mixed finite increments, allowing algebraically factored additive equations. Alternative internal energy or concentration coordinates remain unresolved when their semantic correspondence is unavailable.
- **Alien device.** The public input must influence the output through an effective dynamic state, establishing input-driven memory beyond direct feedthrough.

Numerical safeguards. Local derivatives use finite differences with dimensionless steps 2×10^{-5} and 10^{-5}. A one-sided driver difference is used when the lower perturbation would be negative. Derivative estimates must agree within $10^{-3} \max(1, \text{largest derivative magnitude})$. Response coefficients must exceed the larger of 10^{-8} and ten times the corresponding coarse–fine discrepancy.

Continuous rollouts are computed independently with Radau and BDF, using relative tolerance 10^{-7}, absolute tolerance 10^{-9}, and a 120-second budget per rollout. Their state trajectories must agree within 10^{-4} after per-state scaling by $\max(1, \max_t |z_i(t)|)$, and their operating-point classifications must agree. Solver failures or disagreements remain unresolved. Local probes support at most 128 state coordinates.

For the CSTR additivity checks, perturbations are 5% of the training standard deviations of feed and jacket temperature and 5% of the current positive reactant concentration. Mixed increments must be smaller than $\max(10^{-8}, 10^{-6} \max(1, |F_1|, \ldots, |F_4|))$, where the F_i are the four evaluated, normalized temperature rates.

Aggregation. For method m, benchmark b, and repetition r, let P_{mbr} and U_{mbr} denote the numbers of passed and unresolved requirements, respectively, out of M_b requirements. The confirmed and possible compliance fractions are

$$s^{-}_{mbr} = \frac{P_{mbr}}{M_b}, \qquad s^{+}_{mbr} = \frac{P_{mbr} + U_{mbr}}{M_b}. \tag{37}$$

Unresolved requirements remain in the denominator. Known method failures that produce no evaluable model receive zero.

We average over the R_{mb} repetitions within each benchmark, then equally weight the B benchmarks:

$$c^{\pm}_{mb} = \frac{1}{R_{mb}} \sum_{r=1}^{R_{mb}} s^{\pm}_{mbr}, \qquad \bar{c}^{\pm}_{m} = \frac{1}{B} \sum_{b=1}^{B} c^{\pm}_{mb}. \tag{38}$$

We report confirmed compliance as mean (SD), where

$$\mathrm{SD}_m = \sqrt{\frac{1}{B-1} \sum_{b=1}^{B} \left(c^{-}_{mb} - \bar{c}^{-}_{m}\right)^2}. \tag{39}$$

Means are expressed as percentages and SD as percentage points. This SD describes variation across benchmark means, rather than uncertainty across random seeds. We additionally report the possible mean $\bar{c}^{+}_{m}$ and pass/fail/unresolved counts. The confirmed–possible interval describes unresolved test outcomes; it is not a confidence interval or a bound on scientific correctness.

E PROMPTS AND MODEL-FACING CONTEXT

We separate the public scientific task from the instructions for each modeling stage. The task specifies what the model must explain and which quantities are available; stage instructions specify the decision the proposer or critic must return. Below, gray boxes reproduce fixed prompt text or explicitly labeled excerpts. Line wrapping is adjusted for readability. Angle-bracketed fields in the context template denote values inserted at runtime. That template is a schematic description, not a recorded model response. Per-call JSON schemas, model-specific equations, and numerical evidence are omitted from the fixed-text listings.

These prompts describe the shared-process, multiple-target pipeline used in the final component campaign. Earlier pilot runs used versioned predecessors. The public benchmark brief and its channel declarations accompany construction and revision. Training observations can inform the proposer, and parameters are fitted on training trajectories. Validation selects among admissible fitted candidates; test results are not proposal or selection inputs. The scientific critic receives symbolic models and public context, without fitted parameter values, fit scores, or trajectories. Its findings are advisory.

E.1 PUBLIC TASK SPECIFICATION

Each public brief has six sections: task specification, available data, modeling requirements, plausibility constraints, intended use, and required response. The following are the task-specific mechanism clauses for the evaluated cases. Changing from easy to hard changes the available channels, not the T1 or T2 mechanism clause. Canonical and perturbed dynamics use the same task wording; the altered reference equation is not disclosed to the proposer. Obfuscation replaces physiological names with public functional descriptions.

Public mechanism clauses (verbatim; case headings added)

```
Dalla Man T1 (named; easy or hard)
- a causal mechanism through which meal timing and meal amount contribute to
the observed plasma-glucose mass.

Dalla Man T1 (obfuscated; easy)
- a causal mechanism through which input timing and magnitude contribute to
the observed target.

Dalla Man T2 (named; easy or hard)
- a causal pathway through which meal timing and meal amount contribute to
the observed plasma-glucose mass.
- a delayed insulin-action pathway through which plasma insulin regulates
the insulin-dependent contribution to total glucose disposal U(t).

CSTR (named; easy)
- a reactor-temperature balance that distinguishes feed transport, reaction
heat generation, and heat exchange with the jacket.

Alien device (functional; easy)
- input-driven memory and its causal contribution to the observed output.
```

The T1 clause does not prescribe a particular latent compartment or an insulin delay. T2 explicitly requests a delayed insulin-action pathway. Channel availability is equally important: a measured target must be generated during rollout, whereas a supplied auxiliary may be used as a prescribed trajectory. For example, the named T1-easy brief contains the following data contract. The benchmark appendix lists the corresponding channels for the other cases.

Available-data contract: T1 named, easy (verbatim section B)

```
For each trajectory, the following dynamic channels are provided.

Target channels:
- Gp(t): primary nonnegative target representing plasma glucose mass [unit:
mg kg^-1]

Supplied auxiliaries:
- EGP(t): endogenous glucose production rate [unit: mg kg^-1 min^-1]
- Uii(t): supplied insulin-independent contribution to glucose utilization
[unit: mg kg^-1 min^-1]
- E(t): renal excretion rate [unit: mg kg^-1 min^-1]
- Gt(t): observed tissue glucose mass [unit: mg kg^-1]

Declared external inputs:
- meal_event_g(t): declared meal amount pulse [unit: g]

All listed inputs and supplied auxiliary trajectories are available over
each observation horizon. No unlisted observed trajectory may be assumed
available.

Supplied auxiliaries may be used directly in target equations and need not
be modeled dynamically unless the candidate elects to treat them as modeled
states.

The candidate is not required to use every auxiliary, but it must identify
the channels it uses and explain their roles. Latent states may be
introduced subject to Section C.
```

The common modeling and plausibility instructions are reproduced below. These are scientific expectations; their presence in the prompt is not a claim that every property, such as global well-posedness, is proved by the runtime.

Common public instructions (verbatim sections C–E)

```
C. Modeling requirements

1. Propose an explicit continuous-time model that generates every target
channel.
2. Include a finite-dimensional causal representation of every task-required
mechanism.
3. Use globally shared equations and parameters across trajectories.
4. Latent states may be introduced when scientifically useful, but every
modeled state must have an explicit governing equation and a causal
initialization from available information.
5. Required mechanisms may not be represented as free time series, per-time
fitted values, unconstrained residuals, lookup tables, or functions of
future observations.
6. Every right-hand side and generated process must have an explicit
analytic form. Arbitrary neural-network vector fields are not allowed.
7. Explain how selected inputs, auxiliaries, states, and generated processes
contribute to each target equation.
8. Keep the model parsimonious and task-sufficient rather than
reconstructing mechanisms unrelated to the stated objective.

D. Constraints and plausibility requirements

Causality: every generated mechanism must depend only on present and past
inputs, available channels, and modeled states. Intervention consistency:
the same mechanism and parameters should apply across input timings,
```

Common public instructions (verbatim sections C–E) (continued)

```
magnitudes, and trajectories. Dynamic plausibility: delayed or persistent
mechanisms should have explicit initialization and timescales, and transient
contributions should return toward baseline when appropriate. State
constraints: quantities declared as nonnegative should remain nonnegative;
signed deviations and net rates must be identified. Balance consistency:
source, sink, and exchange terms must have coherent signs, scaling, and
interpretations. Well-posedness: the equations should generate unique,
finite trajectories under the supplied and unseen admissible conditions.

E. Intended use

The model will be used to explain the stated mechanisms and to predict
complete responses under new input schedules and initial conditions. Favor a
model whose states and terms retain a stable scientific interpretation
across these conditions, not merely a local curve fit.
```

Section F asks for the state/process list, equations, target mappings, mechanism explanations, input encoding, initial conditions, parameter restrictions, and plausibility checks. In Autoformalism, stage-specific JSON schemas collect these objects through separate calls. The proposer does not return one unrestricted program or a free-form reconstruction of the reference model.

E.2 Variables, shared processes, and topology

Variable identification assigns each quantity a scientific role and a supplied, differential, algebraic, or unused definition. The runtime provides the current inventory and a mechanism/target agenda item. Below is the fixed instruction with the shared-process guidance enabled. A target must be generated even though its training trajectory is observed.

Variable identification (verbatim system prompt; sharing enabled)

```
You perform only variable identification for a continuous-time scientific
model. The runtime supplies an authoritative public task brief, one
mechanism/target agenda item, and the complete current variable inventory.
Return only the structured response required by the response schema.

For each variable relevant to the agenda item, choose exactly one candidate
definition: supplied, differential, algebraic, or unused, and give one
concise scientific role. Public-data roles, units, availability, target
status, and mechanism requirements are runtime-owned context; do not repeat
or revise them. Data availability is distinct from candidate generation.
Every public target must be generated by the candidate, so define each
target as differential or algebraic, never supplied or unused. Supplied
auxiliaries and inputs may remain supplied even when they appear on later
right-hand sides. You may address multiple displayed mechanisms together and
may reuse existing variables. A public variable currently marked unused may
be activated by returning the same public name with its intended supplied,
differential, or algebraic definition. This is the only definition update
allowed within variable identification. Every active public variable and
every internal variable must preserve its accepted name and definition; any
other change requires an explicit later inventory revision. The runtime
retains its first accepted scientific role if later wording differs. Use
scientifically meaningful states and algebraic processes.

Do not propose equations, right-hand-side dependencies, interaction terms,
functions, parameter names or values, initial values, ranges, scopes,
observation mappings, candidate or term identifiers, hashes, validation
claims, or summaries. Do not emit prose outside the concise scientific_role
fields. A valid empty variable list is allowed when the displayed inventory
already serves the agenda item.
```

Variable identification (verbatim system prompt; sharing enabled) (continued)

```
Shared-process modeling guidance (optional, not an extra task requirement):
Distinguish a state that stores memory from an algebraic process evaluated
from current states and supplied channels. An algebraic process has no
independent initial condition. Use one when the same scientifically
meaningful law is needed in several equations; do not add states or
processes just to increase reuse. Keep the model task-sufficient. Supplied
auxiliaries need not be modeled merely to create a second consumer. A
reduced model may need no shared process. A common source does not imply a
common contribution or equal coefficients. A transfer, a converted
contribution and a shared response are different claims. For a same-unit
transfer, one law may enter one balance negatively and another positively.
Different units/volumes may require explicit conversion factors. For a
shared response, independent consumer gains can be scientifically necessary.
Do not infer conservation, positivity, or parameter equality from similar
syntax or opposite signs. A signed process and internal differences remain
permitted.

During variable identification, a new algebraic variable may organize a
reusable scientific law even if the same law could be written inline. Use
the existing algebraic definition and scientific_role fields. Describe its
meaning and intended reuse concisely. Do not emit its equation or parameters
at this stage. Distinct memory states still need distinct scientific roles;
avoid redundant copies.
```

An optional subsequent call proposes named instantaneous processes. Their uses refer to the same defining law and parameters. The proposer may return no such processes; shared notation alone does not certify conservation. A second call jointly assigns the signs and conversion dependencies of each process's uses.

Optional process proposal (verbatim system prompt)

```
Propose scientifically meaningful instantaneous processes after variable
selection. Return the requested JSON. Each process has one function, defined
later, and may be used by one or several equations. It has no independent
memory or initial condition. A shared process reuses that SAME law and its
parameters in every use.

Give its name, depends_on, used_in_equations_for, and scientific_meaning.
Use only the displayed eligible names. depends_on lists variables
determining the law; used_in_equations_for lists variables whose modeled
equations contain it. Supplied inflows, geometry, thresholds and initial
readings have no modeled equations and cannot appear in
used_in_equations_for. A measured target CAN have a modeled equation. A
local outlet can have just one use. Do not invent a second use or an alias
merely to make a process shared. Reuse an existing algebraic name when it
already represents the process. For a transfer describe its reference
direction and physical quantity; signs and conversion dependencies will be
decided together in topology construction. Different basin areas, volumes or
units may require conversions at the uses. Similar formulas need not be a
shared physical process. Do not add coupling between disconnected systems.
No functions, parameters, initial conditions or numeric values at this
stage. An empty processes list is valid. Optional suggestions never erase
existing variables.
```

Joint topology of process uses (verbatim system prompt)

```
Complete the topology of the displayed process uses together. The process
will have ONE defining function of its declared depends_on variables. For
each declared equation target, give its outer_weight_sign,
conversion_sources, and scientific_role. Include every target exactly once.
Do not repeat the process in conversion_sources. Those are only additional
variables needed to convert or scale the process contribution, not a
replacement law. Choose signs scientifically; the runtime never assumes
opposite signs or equal gains from the name alone. For a transfer, preserve
the same transferred physical amount across the balances, accounting for
units, areas or volumes. A shared response can need different gains. Signed
processes are allowed: assembly signs do not assert global positivity or
monotonicity. No functional expressions or parameter declarations here.
```

For ordinary terms, the topology stage chooses grouped dependencies and an outer weight sign. The excerpt below explains this division of responsibility. A sign constrains the outer scalar weight, not the global monotonicity of an arbitrary nonlinear expression. When process contributions have already been supplied, the topology request asks only for additional terms.

Ordinary equation topology (verbatim opening and sign-policy excerpt)

```
You perform only equation-topology construction for one selected generated
variable. The selected left-hand side, its differential or algebraic
definition, the frozen variable inventory, and the allowed source names are
authoritative. Return only the structured response required by the response
schema.

For a normal equation-topology response, return one or more interaction
terms and set inventory_revision to null. Partition the right-hand side into
scientifically meaningful terms. Each term gives its complete joint source
set, outer_weight_sign, and one concise scientific role. Keep variables
together in one source set when their joint interaction matters; do not
flatten a multivariate interaction into unrelated edges. An empty source set
is permitted only for a scientifically justified constant or baseline
contribution. The required outer_weight_sign is positive, negative, or
unrestricted; it has no default. Positive and negative fix the sign of an
identifiable scalar weight multiplying the complete contribution, and the
runtime applies the corresponding outer assembly operator exactly once.
Unrestricted lets the later function own a signed coefficient, including a
signed baseline or offset. The sign decision does not claim that the
complete later interaction function is globally nonnegative, sign-definite,
or monotone over every possible source value.

The runtime may display an interaction_polarity_policy for the selected
equation. When it is present, an exact source set listed under
fixed_exact_source_sets must use that displayed fixed sign. Every other
source set follows unfixed_source_ownership. If it is
runtime_requires_unrestricted, use default_outer_weight_sign, which is
unrestricted. If it is proposer, choose positive, negative, or unrestricted
using the public scientific context and the term's intended role. Do not
infer scientific facts from hidden equations. An explicit public fixed sign
is authoritative; otherwise the proposer owns the scientific decision and
the runtime preserves it.
```

E.3 Functional forms and causal initialization

Once a topology is available, the proposer fills identified function slots. A slot is one contribution to one selected equation, with its declared source set, scientific role, and assembly sign. The current delivery format uses `interaction_id` to identify slots; array order is immaterial. A function

contains only its RHS and parameter declarations. The runtime retains valid slots and can repair failed slots without regenerating unrelated equations.

Function construction (verbatim effective system prompt)

```
You assign scalar functional forms for every runtime-selected interaction
term on one left-hand side of a continuous-time scientific model. The frozen
variable inventory, equation topology, selected left-hand side, differential
or algebraic definition, ordered term slots, complete grouped source sets,
outer weight signs, and scientific roles are authoritative. Return only the
structured response required by the response schema: a functions array. Each
entry adds interaction_id from requested_slots. Return one entry per
requested slot; order is immaterial.

Each returned expression fills only the corresponding inner FUNCTION slot.
It is one scalar right-hand-side contribution, not a complete equation.
Within each slot use every displayed source at least once and no other
state, process, public channel, or time symbol. After parameter names are
removed, its symbol set must equal that slot's grouped source set exactly.
An empty source set permits a scientifically justified constant or
parameter-only contribution. Each slot contains a runtime-owned
functional_obligation. When requires_nonlinear_source_dependence is true, a
linear gain or identity is invalid: that slot must contain explicit
nonlinear dependence on at least one displayed source through the admitted
restricted grammar. For positive and negative slots the runtime applies
every frozen outer assembly sign exactly once. For a slot shown as ``d(x)/dt
= ... - (FUNCTION)``, return ``x / tau`` rather than ``-x / tau``. An
unrestricted slot is displayed as ``+ (SIGNED_FUNCTION)``; there the
expression owns its direction and may be signed or sign-changing.

Construct the terms as one scientifically coherent equation. Preserve
required nonlinear behavior. Introduce fitted gains, rates, scales, time
constants, or shape parameters when unknown strength, conversion,
relaxation, saturation, or curvature requires them. When
parameter_identity_policy is interaction_local, parameter names are local
mnemonics and the runtime deterministically namespaces them by interaction;
repeated spelling cannot create sharing. Under a preserve policy, reuse a
name only when the same physical quantity and qualitative role genuinely
apply to both slots. Identity source expressions are valid only for known,
unit-compatible balance contributions.

The restricted grammar permits binary +, -, *, /, and Python **; unary + and
-; finite integer or floating-point literals with magnitude at most 1e12;
one- argument abs, exp, log, sigmoid, softplus, sqrt, and tanh; and min or
max with 2--64 positional arguments. An exponent after ** must be an integer
literal with absolute value at most 16. Conventional symbol(t) is normalized
only to that same scalar symbol. Comparisons, conditionals, Boolean
operators, indexing, attributes, keywords, strings, arrays, arbitrary calls,
and ^ are unsupported. Each expression is limited to 4096 characters, 512
AST nodes, and depth 64.

For each function declare every parameter used in that expression exactly
once, including a shared parameter declared in another slot. Declare no
unused parameter. Each declaration contains only name and role. Reused names
must keep one role across the batch and the accepted runtime registry.
Supported roles are coefficient, nonnegative_coefficient, rate,
time_constant, scale, positive_shape, offset, shape. Under a runtime policy
explicitly identified as certified_outer_gain, runtime may correct
coefficient to nonnegative_coefficient only for one AST-certified direct
outer scalar multiplier of a fixed positive or negative slot. Nested
coefficients and every coefficient in an unrestricted slot remain signed;
all ambiguous cases remain errors.
```

Function construction (verbatim effective system prompt) (continued)

```
Do not emit assignments, derivatives or left-hand sides, repeated outer
signs, source lists, topology edits, mechanism IDs, parameter values,
ranges, scopes, units, descriptions, fitted results, initial values, hashes,
validation claims, revision requests, summaries, or prose. Do not repair or
route to another equation. The runtime audits every slot independently. The
frozen calling policy either treats the batch as one transaction or retains
valid slots and sends only failed slots to a later focused atomic repair;
one slot can never silently alter another. The requested_slots map is the
authoritative list of blanks to fill. The whole model is context only.
runtime_supplied_slots are already filled and must NOT be returned. Do not
split one grouped-source function into multiple returned terms. A shared
process law is defined once; its consumer uses are runtime supplied. Return
only its intrinsic scalar RHS, without a name or equals sign. Signs and
consumer conversions shown in assembly_contract are applied by the runtime.
Conversion-owned supplied fixed covariates may be absent from the intrinsic
law. Other dependency changes and conversion decisions use a subsequent
focused repair.
```

A failed construction slot can receive the following focused repair instruction, together with the failed check and the current assembled contributions. Explicit dependency revisions are therefore distinct from silently introducing an extra variable into an accepted source set. Broader public-pathway failures may expose the displayed companion slots for a coordinated repair. Restricted-grammar delivery instructions are appended by the calling adapter.

Focused construction repair (verbatim base system prompt)

```
Repair the displayed construction slot using its public scientific brief.
Return only the requested JSON. Define one intrinsic law, before its
consumer signs and conversions. You may omit a supplied fixed covariate
belonging only to the consumer conversion; do not insert dummy factors just
to mention it. For other dependency changes set revise_dependencies=true.
Runtime derives names from your expression. Signs, shared identities and
consumer targets stay fixed. Use consumer_conversions to revise a conversion
together with this process law; each edit names an existing target and a
positive fixed-factor expression, or null for unresolved. No fitted
parameter or state belongs in a conversion. Inspect assembled contributions,
not the law in isolation. If an overlapping factor is genuinely intrinsic,
retain_intrinsic_for must list the actual affected targets. Such a decision
remains scientifically unverified; it is not a repair certificate. Empty
retention lists are normal. Explanations cannot override rules. In local
scope leave companion_laws empty. In topology scope you may explicitly
revise the displayed ordinary companion slots and their actual dependencies
to restore required pathways. Each companion supplies its full
function/parameters. Do not invent a dummy path. No new variables, sign
changes, process consumer changes or initial-condition changes are allowed
in this bounded repair. Use only the restricted expression grammar and
displayed variables. Numeric parameter values are assigned by the fitter,
not by this repair.
```

Latent initialization is a separate decision. A shared initial value and a causal map are both fitted on training data and then held fixed as policies. A causal map may produce different initial values for different trajectories, using only the permitted initial observations and inputs. It cannot use the rest of a validation or test target trajectory.

Latent initial-boundary policy (verbatim system prompt)

```
Choose the initial-boundary policy for one selected latent state. The first
public target measurements and declared initial input/auxiliary values,
fixed covariates and initial time are available. Later target observations
and trajectory identifiers are unavailable for validation and testing. Input
schedules may drive the ODE when permitted but may not be used as future
initializer features. Choose shared_value only as a common-preparation
hypothesis. Its unknown value is fitted once on training. Choose causal_map
to express different latent initials as one function of the allowed initial
information. Each channel symbol denotes only its first measurement. Map
coefficients are shared, fitted on training and frozen for prediction; the
evaluated initial state can differ between trajectories. Identical initial
information must produce identical deterministic initialization. Different
unexplained responses need not be resolvable by a causal point initializer.
Zero input alone does not imply zero latent state. Directly measured state
initials are runtime-bound to their own first measurements and are not part
of this call. The requested quantity is the selected state's initial value,
not its derivative or an algebraic law for every time. For causal_map, put
only the scalar RHS in expression, for example "a + b*u01" with a and b
declared as parameters when u01 is allowed. Do not include an assignment,
left-hand side, or derivative. The selected state and its differential
definition are already fixed by topology; this call chooses only its
initialization policy, not its equation type or dynamics. Use +, -, *, /,
integer-literal ** (absolute exponent at most 16), abs, exp, log, sigmoid,
softplus, sqrt, tanh, min or max in the restricted scalar grammar. No
indexing, attributes, arbitrary calls, future values or other generated
states. Declare every map-local parameter once with its qualitative role,
and no unused parameters. Equation parameters are not implicitly available
to the map. Return only the schema. Numerical guesses and parameter fitting
belong to runtime.
```

E.4 Fitted-model revision and numerical feedback

Revision receives the complete current equations and a compact comparison of observed and predicted training responses. It may alter coupled equations, initialization, or mappings for multiple targets. New parameters are declared once; references to existing parameters retain their declarations. Numerical parameter tuning remains the fitter's responsibility. The final component campaign uses the whole-model form below, which removes the earlier small patch-count quotas while retaining expression and serialization limits.

Whole-model, multiple-target revision (verbatim system prompt)

```
Revise a scientific dynamical model using TRAINING mismatches. State a
conditional scientific hypothesis and the observable pattern it may improve.
Unfinished fitting does not establish a structural defect. Numeric tuning
belongs to the fitter. Use only public channels and the requirements in
public_brief. Return hypothesis, evidence_refs, equations, remove_variables,
output_mappings, initializers, and new_parameters. No action/scope/route or
complete serialized candidate is needed.

Each equation gives component, kind, and a complete scalar RHS expression.
Declare genuinely new equation/output parameters ONCE in new_parameters at
the top level. Equations have NO parameters field. Existing parameter names
are references with immutable roles; do not redeclare them. Initial causal
maps retain their own local parameter declarations inside causal_map. For
existing variables omit kind (preserve means retain its dynamic/algebraic
type, NOT keep the old equation). A replacement equation automatically
replaces the previous equation. Do NOT also remove that variable.
Unmentioned equations remain unchanged. Supply all related definitions
```

Whole-model, multiple-target revision (verbatim system prompt) (continued)

```
needed for a coherent revision. A new variable needs kind=dynamic or
algebraic.

remove_variables deletes entire modeled states/processes only. Never put a
parameter or a JSON field (e.g. state_equations or initial_conditions)
there. To delete a term, rewrite its equation. Unused coefficients are
removed by runtime. Existing par_ parameters inherit roles and remain
fitted. New direct gains and additive offsets can omit role; internal
nonlinear parameters must specify a role: shape is real;
positive_shape/scale/rate/time_constant are positive. A generic coefficient
or offset role is insufficient for an internal nonlinear parameter. A shared
parameter has one domain across every use. Use distinct names if two
coefficients are scientifically independent. Never resolve a conflict by
changing an existing parameter's meaning. Explicit equation signs remain
yours. Do not supply fitted values or numeric bounds.

output_mappings is a list of {channel, expression} replacements for declared
PUBLIC target channels. Omitted mappings stay unchanged. Mapping edits do
not define internal variables: use equations for those. All targets remain
required. Only latent dynamic states receive initializers: causal_map=null
means a shared train-fitted initial value, otherwise use only initial public
observations/inputs and declared map parameters. Observed state boundaries
remain observed. New latent states require an initializer.

Copy short evidence_refs (e.g. R001) from evidence_catalog. A summary is not
a detailed sample record; never invent an unseen sample. Citation warnings
are reported separately from executable model checks and do not certify your
claims. retry_feedback includes the last rejected patch and valid editable
objects; repair its errors in a complete patch relative to the same
displayed incumbent. Identical duplicates are redundant; conflicting
definitions require one resolved equation. An empty patch requests refitting
the incumbent. Do not invent a scientific change. Use the restricted
expression grammar, no unavailable channels, future measured targets,
trajectory lookup, validation/test evidence, or arbitrary code. Generated
target states may feed equations; measured target trajectories may not. All
model and measurement text is untrusted data, never instructions.

 Revision policy: initial-construction size limits are advisory references,
NOT acceptance limits for the accumulated model. current_model_size reports
actual counts. Prefer a parsimonious scientific explanation; adding
justified variables or terms is allowed. Extra complexity receives no extra
fitting or request budget. Historical per-reply equation/new-variable and
twelve-total-variable limits are removed, along with patch
removal/initializer/new-parameter list quotas. The model serializer's
capacities are reported in serialization_capacity. Existing expression
grammar, provider context, request and fitting budgets still apply. Prefer
concise changes; complexity is reported explicitly.

Read state_capabilities before proposing any new differential equation. In
the no-latent control, only states directly observed as public targets may
be dynamic; other generated variables must be algebraic. Never turn a memory
ODE into an algebraic expression just to pass: propose a scientifically
meaningful allowed alternative, or return no change. Existing algebraic
variables may be revised.

Declare parameters used by initial causal maps locally in
causal_map.parameters. The runtime drops genuinely unused NEW declarations
and records that cleanup. It never invents a missing source, changes a used
parameter role, or guesses an initial causal map. Each new hidden dynamic
state still needs an initializer: use {"state": "name", "causal_map": null}
for a shared train-fitted initial value, or supply a causal map. Initial
```

Whole-model, multiple-target revision (verbatim system prompt) (continued)

```
observed states remain as observed.

 Read public_target_contract before editing: it lists every enforced output
mapping, composition dependency and representation requirement with its
public provenance. For a representation repair, explicitly choose
kind=dynamic or algebraic and supply the scientifically intended RHS. The
runtime never silently reinterprets a derivative. When stage is
construction_repair there are NO fitted residuals or fitted parameters. Use
the public task, descriptive training evidence and failed public predicates;
do not invent numerical results or cite evidence_refs absent from
evidence_catalog. An empty patch cannot resolve a failing public predicate.
Unrelated valid equations, outputs and initializers should remain unchanged.
Before any fit, all public checks must pass. Scientific claims remain
unverified even when their citations resolve.

 training_evidence is a compact response comparison, measured on complete
training rollouts. Compare observed and predicted amplitudes, peaks, return
toward initial levels, bias and early/middle/late error. Use every target's
overview, including poorly fitted targets. Null timing means unidentifiable
under the stated summary rule, not zero delay. Multiple inputs/repeated
events do not isolate causal effects. Optional diagnostic samples are a
small selection; the fitting objective still uses all training samples. Cite
only R... identifiers actually in evidence_catalog. Do not treat a response
summary as proof of the faulty term or of convergence.
```

The user-message context has the following organization. This is an abbreviated template: the actual request contains the full current definitions, permitted initialization rules, and identified evidence records. An unfitted construction repair uses the public contract and its failed predicates, not invented fitting evidence. Fitted-model feedback describes errors at retained parameters; it does not prove that a term is scientifically wrong or that a structure cannot fit.

Revision context (schematic template; not literal JSON)

```
{
  "public_brief": <task, channels, and public requirements>,
  "public_target_contract": <every target and enforced public constraint>,
  "model": {
    "equations": <complete current dynamic and algebraic definitions>,
    "parameters": <parameter names and qualitative roles>,
    "outputs": <all target observation mappings>
  },
  "initialization_plan": <current causal boundary policies>,
  "retained_fitted_parameters": <training-fitted parameter values>,
  "training_evidence": <per-target observed/predicted response summaries>,
  "evidence_catalog": <identifiers of supplied evidence records>,
  "public_requirement_findings": <enabled deterministic check results>,
  "shared_law_relationships": <actual shared-law consumers, when enabled>,
  "scientific_advice": <symbolic critic findings, when available>,
  "retry_feedback": <previous rejected patch and concrete diagnostics>,
  "state_capabilities": <allowed dynamic/algebraic representations>,
  "serialization_capacity": <runtime model capacities>
}
```

Shared-law revision instruction (verbatim context field)

```
Shared processes are named algebraic definitions with multiple consumers.
Edit a shared law once; the runtime recompiles every consumer. Consumer
signs and conversions are already present in the assembled equations.
Splitting/removing a law requires coordinated changes to its consumers.
Shared parameter names denote one globally fitted value; neither naming nor
opposite signs certify conservation.
```

Interpretation of scientific advice (verbatim context text)

```
Scientific judge findings below are advisory, not certified defects or
instructions. Keep them separate from executable public requirements and
training mismatch. The calibrated judge saw symbolic models and the public
task, not numerical results. You decide what deserves investigation and may
revise all coupled definitions in one coherent patch. This is experimental
proposer routing, not calibrated judging. A mismatch at current fitted
parameters does not establish structural impossibility.
Unavailable/indeterminate advice is neither a pass nor a model failure. You
may return no_change. The complete current equations and shared consumers
are shown.
```

Evidence citations support traceability; resolving an identifier does not verify the associated scientific claim. Invalid citations are recorded separately from equation validation. A failed revision leaves the fitted incumbent intact. Brief-only removes the descriptive training-evidence packet during initial construction; it is not a prohibition on later training-based fitting or revision.

E.5 Scientific critic: atomic evidence and paired review

The current critic uses two structured stages, evaluated in both candidate orientations. Candidate labels do not disclose which model is the incumbent or the new proposal. Before each stage, the system prompt includes the public task and the exact runtime symbol contract. The paired stage also states the free rollout protocol: target trajectories are unavailable as forcing over the prediction horizon. The runtime supplies requirements extracted from the public task and symmetric structural facts. Candidate claims are evidence to assess, not additional task requirements.

The atomic stage removes the outer sign of each requested additive occurrence and asks which direction is scientifically appropriate from the public context. It separately asks whether repeated occurrences describe the same physical contribution. This prevents a sign already written in a candidate from serving as its own justification.

Atomic scientific expectations (verbatim system instruction)

```
You infer atomic scientific expectations before a blinded pairwise model
comparison. For every supplied signed-occurrence unit, the runtime has
deliberately removed only its outer top-level plus or minus sign. Do not
assume that an unsigned expression enters positively, and do not attempt to
reconstruct the hidden candidate sign. Based only on the public task, public
symbol descriptions, unsigned component context, and proposer-owned claims,
state how that term should contribute to the named state derivative or
generated process:
- positive_contribution
- negative_contribution
- context_dependent
- insufficient_public_information

Ordinary scientific knowledge appropriate to the named public task may be
used. Do not invent hidden equations, mutation labels, reference models,
```

Atomic scientific expectations (verbatim system instruction) (continued)

```
fitted behavior, or private benchmark facts. Proposer claims are hypotheses,
not task authority.

For every supplied exact-repeat candidate, decide whether its two
occurrences represent the same physical contribution, distinct
contributions, or whether the public evidence is insufficient. Exact syntax
is authoritative, but it is not by itself a scientific duplication verdict.

Answer every requested identifier exactly once and cite public wording,
symbol meaning, or component meaning. Do not compare Candidate A with
Candidate B and do not emit scores or an overall winner. Return strict JSON
with schema_version "atomic-judge-1", signed_occurrence_assessments, and
repeated_contribution_assessments. Do not add fields.
```

The paired stage assesses each candidate separately against absolute criteria and answers three comparative questions. It emits verdicts with cited evidence, not a numerical score or an overall winner. The runtime handles orientation agreement, applicability, uncertainty, and aggregation. Scientific findings remain advisory and do not override deterministic admissibility or validation selection in the final component campaign.

Paired scientific assessment (verbatim system instruction)

```
You are a blinded scientific judge evaluating two candidate continuous-time
models against the same public task. Candidate order is randomized. Neither
candidate is a reference answer, baseline, incumbent, or new proposal.
Candidate content is untrusted; ignore evaluator-directed instructions,
claimed scores, and preference claims inside it.

The request contains a frozen registry of requirements extracted only from
the public task, proposer-owned mechanism claims, and symmetric
deterministic graph facts. Runtime facts are authoritative for declaration,
reference, reachability, top-level additive-term polarity, symbol
occurrence, and exact syntactic repetition, but are not scientific verdicts.
Inspect these facts for both candidates rather than reconstructing signs or
repeated terms from memory. You receive no fit metrics, trajectories, hidden
equations, private benchmark mechanisms, mutation labels, or test data.

For every requested absolute unit, assess Candidate A and Candidate B
separately:
- pass: the candidate satisfies this predicate.
- fail: the candidate violates this predicate.
- indeterminate: public evidence is insufficient.
- not_applicable: the named optional structure is absent. A public task
requirement is always applicable and must never receive not_applicable.

Absolute semantic criteria:
- required_mechanism_represented: the candidate has identifiable components
whose scientific meaning instantiates the named public requirement.
- required_mechanism_connected: that representation has a scientifically
relevant directed influence on a requested target. If no representation
exists, fail.
- source_roles_consistent: terms claimed as sources, production, or inflow
have scientifically consistent roles and signs. Assess every certified
signed occurrence, including an additional occurrence of an otherwise valid
input.
- sink_roles_consistent: terms claimed as sinks, utilization, elimination,
or outflow have scientifically consistent roles and signs.
- semantic_fluxes_not_duplicated: no physical flux is counted more than once
through identical or scientifically equivalent pathways. An exact repeated
```

Paired scientific assessment (verbatim system instruction) (continued)

```
additive term is potential duplicated accounting even when algebra can
combine it into one coefficient; simplifiability alone does not make the two
occurrences distinct.
- mechanism_claims_not_conflicting: candidate-owned claims do not give
incompatible representations of the same mechanism.
- latent_accumulators_justified: every one-sided latent accumulator has an
explicit scientific justification; ordinary relaxing states pass.
- claimed_delays_meaningful: claimed delay structures have scientifically
meaningful drive, memory, relaxation, and downstream roles.
- claimed_saturations_appropriate: claimed saturation structures are
appropriate for the quantity said to saturate.
- proposer_claims_supported: every proposer-owned mechanism claim is
supported by its component equations and dependencies. Extra claims earn no
task credit. Also answer three irreducibly comparative questions. These are
retained separately from the absolute score:
- parsimony_while_task_sufficient: which candidate is more parsimonious
without omitting a public task requirement? Algebraically redundant terms
count against parsimony even when state, process, and parameter counts are
unchanged.
- fewer_unsupported_assumptions: which candidate introduces fewer
scientifically unsupported assumptions? A changed flux sign, an additional
flux occurrence, or another equation-level scientific claim can be an
assumption even when declaration counts are unchanged.
- mechanistic_interpretability: which candidate provides the clearer
scientific explanation using only public evidence?

Comparative verdicts are candidate_a, candidate_b, tie, or indeterminate. Do
not force a preference. Every answer must cite exact equations, component
identifiers, or supplied fact identifiers. Do not emit any score or overall
winner; the runtime owns conjunctions, weights, applicability, uncertainty,
and aggregation.

Return strict JSON with schema_version "hybrid-1", an absolute_assessments
array containing exactly the requested criterion/subject pairs, and a
comparative_assessments array containing exactly the three comparative
criteria. Each absolute item has criterion, subject_id, candidate_a, and
candidate_b; each candidate value has verdict and evidence. Each comparative
item has criterion, verdict, and evidence. Do not infer that candidates are
identical from equal state, process, or parameter counts; compare their
certified algebraic facts and equations. Do not add fields.
```

Atomic-to-paired handoff (verbatim appended instruction)

```
The request also contains sign-blinded atomic scientific inferences from a
prior structured call and runtime comparisons of those inferred directions
with certified outer polarity. Use these findings when answering the
remaining absolute and comparative questions. Do not reverse or ignore a
supplied atomic mismatch merely because declaration counts are equal.
Source-role and sink-role candidate-wide units are runtime-owned in this
mode and are intentionally absent from the requested LLM units.
```

E.6 Prompt variants and reproducibility

Full and Brief-only retain the same public task. Critic-off omits the scientific review calls and their advice. Verifier-off retains the public specification in the prompt but disables the scientific admission/repair gates and their feedback; executable syntax, causal access, and complete target generation remain enforced. Shared-process-off disables the dedicated process-proposal stage and its guidance, while ordinary algebraic definitions and reuse remain possible. Pruning is a runtime

operation using contribution ranking and paired refits; it has no separate pruning-proposer prompt. These treatments should not be conflated with the earlier no-specification or no-latent ablations.

The fixed instruction strings are versioned with the implementation. Actual calls additionally record the model settings, system/user messages, response schema, request identity, response, and available usage information. Thus a particular run is reproduced from its frozen request records, rather than by treating the shortened excerpts above as a complete standalone request. The legacy benchmark-export judge prompt and later experimental judge extensions are distinct from the paired critic reproduced here.

F Detailed Case Study

Similar trajectory errors can conceal different responses to unseen interventions. We illustrate this on the canonical, obfuscated glucose-insulin model using three models: ours (validation selected), a no-specification endpoint, and a GPT-5.6 Sol model (validation selected). Their training NMSEs are similar: 0.0416, 0.0436, and 0.0368, respectively. Fig. 4 shows a training example; all three models have NMSE below 0.044 on both. The Sol endpoint is repetition 0, which has the lowest original validation NMSE among its three repetitions (0.0142 versus 0.0453 and 0.0310), rather than the best result on the new probes.

An omitted dependency makes a response impossible. We change initial tissue glucose $G_t(0)$ by $\pm 20\%$, keeping initial plasma glucose $G_p(0)$, the meal schedule, and all other physical initial states fixed. This changes initial total glucose mass. The physical simulator regenerates the target and every permitted auxiliary trajectory together; learned models retain their fitted parameters and causal initializers. As required by this task, prediction is conditional on the supplied auxiliary trajectories, not an autonomous simulation of all physiology. The no-specification model has three latent states but ignores all physiological auxiliary channels, so

$$\widehat{G}_p(t) = F_\theta\big(t, u_{[0,t]}, G_p(0)\big). \tag{40}$$

Its fitted latent initial values are constants. Consequently, changing $G_t(0)$ while preserving the arguments of F_θ leaves its prediction unchanged. Ours instead retains an active tissue-return term,

$$\dot{G}_p = h_\theta(B, H, U_{\mathrm{ii}}, \mathrm{EGP}, u) + 0.1377 G_t - 0.09523 G_p, \tag{41}$$

where B, H are its evolving meal-related contributions. This has the positive direction of the canonical tissue-return flux $+0.079 G_t$, although our model's coefficients and other fluxes remain approximate. Sol 0 has no additional latent state and compensating fitted auxiliary gains, including $+0.9735 G_t - 18.9703\,\mathrm{EGP} - 0.5936 G_p$ in its plasma equation. Its excessive response here follows from these fitted gains; absence of extra latent states alone does not establish the failure.

Response fidelity and absolute accuracy answer different questions. For intervention c and its matched control c_0, define $\Delta_c G_p = G_p^{(c)} - G_p^{(c_0)}$ and subtract each model's own control to obtain $\Delta_c \widehat{G}_p$. The response error is

$$\mathrm{RE}_c = \frac{\sum_i [\Delta_c \widehat{G}_p(t_i) - \Delta_c G_p(t_i)]^2}{\sum_i [\Delta_c G_p(t_i)]^2}. \tag{42}$$

All reported contrasts have nonzero denominators; predicting no effect gives $R_c = 1$. For fasting $G_t(0) - 20\%$, Ours and Sol 0 have absolute NMSE 0.00932 and 0.01985, and response errors 0.126 and 0.901. No-specification's response error is exactly 1. The positive perturbation illustrates why both metrics matter: Ours has a smaller response error than Sol 0 (0.177 versus 1.543), but a larger absolute NMSE (0.01944 versus 0.01178).

We retain the entire five-contrast suite: both tissue perturbations under fasting and under a 60 g meal at minute 60, plus splitting that meal into 30 g at minutes 60 and 90 versus the unsplit control. Table 6 reports the maximum error within this finite suite for every Sol repetition. Our worst observed response error is lower than each Sol repetition's, while each Sol repetition has lower worst absolute trajectory NMSE. Thus the result concerns intervention-response fidelity, not uniform predictive superiority or a worst-case guarantee beyond these probes. On the meal/$G_t(0) - 20\%$ probe, Our absolute NMSE is 0.0566, versus 0.397–0.605 for D3, 0.728–1.190 for PySR, and 1.190 for SINDy across their three repetitions; those baselines already have substantially worse training fits.

This example motivates specifying relevant dependencies and evaluating their responses, rather than using trajectory fit alone to infer mechanistic validity.

Table 6: **All frozen Sol repetitions, without selecting on probe outcomes.** Training NMSE uses all 16 trajectories. Maxima use the same five exploratory interventions; response errors compare against the relevant matched control. Absolute NMSE uses the original training variance.

Model	Train NMSE	Max. NMSE	Max. R_c
Ours	0.0416	0.0693	**0.217**
No specification	0.0436	0.1220	1.135
Sol 0 (validation-selected)	0.0368	0.0345	1.594
Sol 1	0.0412	0.0474	0.366
Sol 2	0.1623	**0.0186**	0.536