NEURAL SELECTIVITY AS A COMPATIBILITY PROBLEM: FROM LATENT COORDINATES TO EXPLICIT FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Whether a neuron is called purely selective, linearly mixed, or nonlinearly mixed is usually decided under a feature set fixed in advance by the experimenter. This label belongs to the pair of neuron and description, not to the neuron alone. Information-preserving re-descriptions of the same inputs can turn nonlinear dependence into linear dependence. We therefore treat the feature description itself as the object of discovery, organized by neural compatibility, the degree to which a feature description supports prediction of a target neural population through a specified simple-readout class. We introduce compatibility-guided feature discovery. A shared population encoder learns a response-predictive representation, and a sparse autoencoder (SAE) provides candidate activation patterns. Language-model-assisted feature proposals and shallow decision trees turn these patterns into binary features defined by executable sensory and behavioral rules. These rule-defined features are selected for their contribution to linear neural prediction. Across 575 neurons from 22 mouse primary somatosensory cortex sessions, full-data discovery produced a descriptive catalog of 164 rule-defined features. Under four-fold nested cross-validation, with encoder training and feature discovery restricted to outer-training trials, adding fold-specific prediction-selected features to inputs already augmented with input-informed LLM features increased mean linear prediction R^2 from 0.1079 to 0.1254. On these recordings, previously analyzed for nonlinear sensory integration, the discovered features made part of the nonlinear predictive structure linearly accessible, reducing the nonlinear-over-linear R^2 gap by 54.2%. Even with trial time in the baseline, these features retained 92% of their original R^2 gain, which was 2.4 times the gain from random features matched in vocabulary, tree structure, and prevalence. Features discovered and fixed in 15 sessions also improved held-out linear prediction in each of seven excluded sessions after within-session readout calibration. Treating descriptions as discoverable turns a known caveat of selectivity analysis into a procedure for discovering and testing interpretable features.

1 INTRODUCTION

Neurons are routinely classified as purely selective, linearly mixed, or nonlinearly mixed, and these labels carry theoretical weight. For example, mixed selectivity can support flexible computation by providing population representations from which diverse quantities can be read out (Rigotti et al., 2013; Fusi et al., 2016). Yet each label is assigned under a feature description fixed in advance, and the same response can appear purely selective, linearly mixed, or nonlinear under different feature descriptions (Kaufman et al., 2022; Tye et al., 2024). Re-describing measured inputs leaves fixed neuronal responses and their geometry unchanged, but can change how simply those responses are predicted. This dependence has been noted as a caveat (Kaufman et al., 2022). We turn it into a method by making the feature description itself the object of discovery for a target neural population.

Building on relational accounts of representation (Pohl et al., 2026), we organize this search around *neural compatibility*, the predictive adequacy of a feature description for a target neuron or population under a specified simple-readout class. The neural target, readout class, and evaluation conditions are held fixed while the feature description is varied. Following the encoding-model use of

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.

linearizing feature spaces (Naselaris et al., 2011), we search for shared, interpretable input features and evaluate their linear compatibility through held-out prediction.

We train a shared population encoder on the original sensory and task measurements, with session-specific linear readouts predicting neuronal responses. Its hidden representation supports linear prediction, but its coordinates lack recognizable sensory or behavioral meanings. A sparse autoencoder (SAE) provides candidate activation patterns. Proposals from a large language model (LLM) expand the input vocabulary, and shallow decision trees use it to approximate component activation with executable binary rules, each defining a feature. We select these rule-defined features for their conditional contribution to linear neural prediction. We call this procedure compatibility-guided feature discovery, because candidates proposed from a response-predictive representation are retained by their contribution to linear compatibility. We interpret representative rules using their exact conditions and sensory and behavioral profiles.

We reanalyze recordings from the dataset described by Rodgers (2022) and studied by Nogueira et al. (2023), using 575 neurons from 22 mouse primary somatosensory cortex sessions. Full-data discovery yields a descriptive catalog of 164 rule-defined features (Supplementary Data 1). Illustrated features (Figure 3) capture temporal position changes, cross-whisker position contrasts, and contact-angle use. Separately, four-fold nested cross-validation repeats representation learning and feature discovery within outer-training data. Adding fold-specific prediction-selected features to inputs already augmented with input-informed LLM features (Section 3.2) raises mean linear prediction R^2 from 0.1079 to 0.1254. On the recordings in which Nogueira et al. (2023) reported nonlinear sensory integration, the discovered features convert part of the nonlinear predictive structure into linear dependence on explicit input conditions, reducing the nonlinear-over-linear gap by 54.2%. Features discovered and fixed in 15 sessions also improve held-out linear prediction in each of seven excluded sessions after within-session readout calibration.

Our contributions are threefold. (i) We formalize selectivity as a relation between a neural target and a feature description, show that information-equivalent descriptions can differ maximally in linear compatibility while predictable response variance sets its supremum (Proposition 1), and relate the attainable compatibility at fixed dimension to conditional-mean geometry. (ii) We introduce compatibility-guided feature discovery, which turns a response-predictive latent representation into explicit, rule-defined features, selected by their conditional contribution to neural prediction. (iii) In mouse S1, these features improve linear prediction on unseen trials beyond matched random features, make part of the nonlinear predictive structure linearly accessible, and improve prediction in sessions excluded from discovery.

2 Neural selectivity as a compatibility problem

We treat selectivity as a relation between neuronal responses and the features used to describe their inputs. Neural compatibility makes this relation the object of comparison by asking how well a feature description supports prediction of a specified neural target through a specified class of simple readouts. Fixing the target, readout class, and evaluation distribution turns this question into a search over feature descriptions.

2.1 Observed inputs, latent inputs, and neuronal responses

Let $x \in \mathcal{X}$ denote a vector of researcher-defined features, including the relevant history and task context. A schematic latent-input model for neuron n writes its conditional mean response as

$$\mu_n(x) = \phi_n\big(w_n^\top g_n(x) + b_n\big), \qquad Y_n = \mu_n(x) + \varepsilon_n, \qquad \mathbb{E}[\varepsilon_n \mid x] = 0. \tag{1}$$

Here $g_n : \mathcal{X} \to \mathbb{R}^{k_n}$ is an unobserved transformation summarizing upstream computations, $z_n = g_n(x)$ is a distributed feature vector, and w_n and b_n are its readout weights and bias. The response function ϕ_n is monotone non-decreasing. Y_n is the measured response and ε_n its residual variation.

The latent vector z_n is a *compatible representation* for neuron n under the affine-plus-monotone readout in this model. We call such a stage of feature description a *compatible layer*. The aligned scalar feature $f_n^*(x) = w_n^\top z_n$ is also compatible. Figure S3 (Appendix E) illustrates this general model.

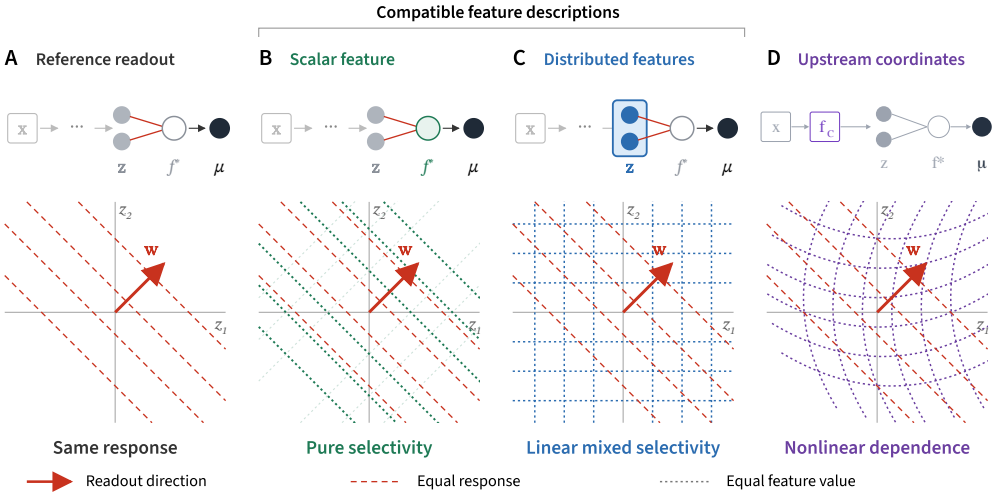

Figure 1: **The same response is pure, mixed, or nonlinear depending only on the description.** All panels show the same response $\mu = (z_1 + z_2)/\sqrt{2}$, with identity response function and zero bias. **(A)** Reference readout, $w = (1, 1)/\sqrt{2}$. **(B)** The aligned feature $f_{A1} = (z_1 + z_2)/\sqrt{2} = \mu$ gives pure selectivity. Bold and faint contours show equal f_{A1} and $f_{A2} = (z_1 - z_2)/\sqrt{2}$, respectively. **(C)** Coordinates $f_{B1} = z_1$ and $f_{B2} = z_2$ both contribute linearly. **(D)** Upstream coordinates related by $z_1 = f_{C1} + \alpha f_{C2}^2$, $z_2 = f_{C2} + \alpha f_{C1}^2$ ($\alpha = 0.1$), on a locally one-to-one domain, give nonlinear additive dependence without loss of information (Proposition 1).

2.2 Selectivity depends on the feature description

A feature description is a vector $F(x) = (f_1(x), \ldots, f_p(x))$ of specified functions on $\mathcal{X}$. With the affine-plus-monotone readout, pure selectivity uses one coordinate, linear mixed selectivity requires multiple coordinates, and another description can require a nonlinear combination. Figure 1 illustrates a fixed response under aligned scalar (B: pure selectivity), distributed (C: linear mixed selectivity), and upstream (D: additive quadratic) descriptions. B and C both permit exact affine prediction. D re-describes the same inputs by a recoding that is one-to-one on the displayed domain, and the dependence becomes nonlinear. Proposition 1 shows how far such recodings can move linear compatibility.

2.3 Neural compatibility under linear readouts

For empirical evaluation, we focus on *linear compatibility*, the special case in which the permitted readouts are affine functions of the features. Let $\mathcal{N}$ be a target population of M neurons. For each neuron n, P_n is the joint distribution of the observed input vector X and measured response Y_n under the evaluation conditions. Input distributions may differ across recording sessions. We hold $\mathcal{P} = (P_1, \ldots, P_M)$ fixed when comparing feature descriptions. With finite second moments and nonzero response variances, define

$$\mathcal{C}_{\mathrm{lin}}(\mathcal{N}, F; \mathcal{P}) = \frac{1}{M} \sum_{n=1}^{M} \left[1 - \frac{\inf_{\beta_n \in \mathbb{R}^p,\, \alpha_n \in \mathbb{R}} \mathbb{E}_n \left[(Y_n - \alpha_n - \beta_n^\top F(X))^2 \right]}{\mathrm{Var}_n(Y_n)} \right]. \tag{2}$$

$\mathbb{E}_n$ and Var_n denote expectation and variance under P_n. Each neuron has its own weights β_n and intercept α_n, while the feature functions are shared. This is the ideal explained-variance criterion familiar from linear encoding models (Naselaris et al., 2011).

Three observations connect this criterion to feature discovery (Appendix E). *First, the affine prediction space matters, rather than its coordinates.* Replacing F by $AF + a$ for invertible A preserves equation 2, even when a pure/mixed selectivity description changes. *Second, information-preserving nonlinear recodings can change that space and its compatibility* (Proposition 1 below), a distinction also formalized by predictive $\mathcal{V}$-information (Xu et al., 2020). *Third, compatibility measures approximation of the conditional mean response* $\mu_n(X)$ by the span of $1, f_1(X), \ldots, f_p(X)$.

Proposition 1 (information-equivalent descriptions can have different linear compatibility)**.** *Let F be square-integrable under each P_n, and let $\mathcal{G}$ contain the measurable injections G, of any finite output dimension, that have a measurable inverse on the range of F and keep $G \circ F$ square-integrable under each P_n. Then (i) $I(G(F(X)); Y_n) = I(F(X); Y_n)$ for every $G \in \mathcal{G}$ and neuron n. (ii) The supremum of linear compatibility over these recodings is the fraction of response variance predictable from $F(X)$,*

$$\sup_{G \in \mathcal{G}} \mathcal{C}_{\mathrm{lin}}(\mathcal{N}, G \circ F; \mathcal{P}) = \frac{1}{M} \sum_{n=1}^{M} \frac{\mathrm{Var}_n\big(\mathbb{E}_n[Y_n \mid F(X)]\big)}{\mathrm{Var}_n(Y_n)}.$$

(iii) There exist a fixed neural target and evaluation distribution for which two information-equivalent descriptions of the same dimension have linear compatibility zero and one.

Proof. Part (i) holds because $G(F(X))$ and $F(X)$ determine each other. For (ii), every affine readout of $G(F(X))$ is a function of $F(X)$, so the conditional mean bounds its accuracy. Appending conditional means truncated at $\pm T$ to F stays in $\mathcal{G}$ and approaches the bound as $T \to \infty$ (Appendix E, Property 5). For (iii), take one neuron with independent A, B uniform on $\{-1, 1\}$ and $Y = AB$. Under $F = (A, B)$, $\mathbb{E}[Y] = 0$ and $\mathrm{Cov}(Y, A) = \mathbb{E}[A^2 B] = \mathbb{E}[B] = 0$. Likewise $\mathrm{Cov}(Y, B) = 0$, so $\mathcal{C}_{\mathrm{lin}} = 0$. The invertible recoding $G(F) = (A, AB)$ makes Y a coordinate, so $\mathcal{C}_{\mathrm{lin}} = 1$.

Whether a neuron is nonlinearly mixed is therefore not an invariant of the neuron but of the pair (neuron, description). Part (ii) sets the ceiling without constraints on dimension or form. With $F = X$, it is the fraction of response variance predictable from the observed inputs, and appending response-specific conditional means such as $f_n(x) = \mu_n(x)$ approaches it. Part (iii) shows that description matters even at fixed dimension. Under a common input distribution, conditional-mean geometry can bound p-dimensional descriptions more tightly when $p < M$ (Appendix E). The scientific problem is to approach these bounds with shared, interpretable features computable from inputs, which Section 3 addresses with compatibility-guided feature discovery.

An added feature helps when its component outside the existing affine span correlates with residual responses (Appendix E). We assess feature descriptions through held-out linear prediction, which also reflects regularization and finite-sample estimation (Section 3). Comparing linear and nonlinear prediction measures, as a continuous quantity, how much predictive structure is accessible to affine readouts (Section 4.2).

3 Compatibility-guided feature discovery

Compatibility-guided feature discovery extracts short sensory and behavioral *rules* from a response-predictive representation and selects the features they define by their conditional contribution to neural prediction (Figure 2; Appendix B). Each rule defines a binary feature that equals one when its conditions hold and zero otherwise, which we call a *rule-defined feature*.

We apply this procedure to sensory and behavioral measurements paired with recordings from 575 neurons across 22 mouse primary somatosensory cortex sessions in seven animals (Rodgers, 2022; Nogueira et al., 2023). Appendices A and C give data processing and evaluation protocols.

3.1 Population representation learning

In Figure 2A, a shared population encoder maps sensory and task inputs x_i for sample i in session s to a hidden representation h_i. A session-specific linear readout predicts the recorded responses:

$$h_i = E_\theta(x_i), \qquad \hat{y}_i = R_s h_i + b_s. \tag{3}$$

Here θ denotes encoder parameters, R_s the readout weight matrix, and b_s its bias vector. Encoder and readouts are trained jointly to minimize mean squared prediction error.

The hidden layer is a candidate compatible representation, assessed by held-out neural prediction (Section 4.2). A sparse autoencoder (SAE) is then trained to reconstruct normalized hidden vectors. Its activation patterns serve as rule-fitting targets and as contrasts that guide feature proposals.

The SAE decomposes these hidden representations into learned directions, called *atoms*, with non-negative activation coefficients c_{ij}. We use 256 atoms and at most eight positive coefficients per

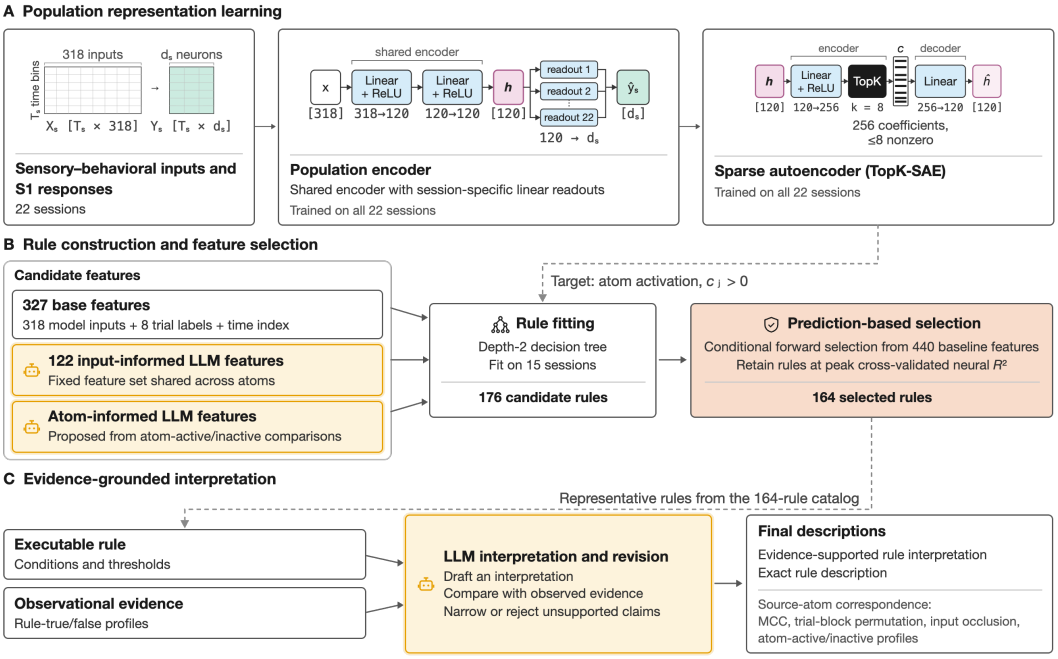

Figure 2: **Compatibility-guided feature discovery for the full-data catalog. (A)** A shared population encoder predicts S1 responses, and a TopK SAE decomposes its hidden representation. Both use all 22 sessions. T_s denotes sample rows (time bins across trials), and d_s denotes the number of neurons in session s. **(B)** Depth-two trees fitted on 15 sessions yield 176 candidate rules. Conditional neural prediction selects 164 to augment 440 baseline predictors. **(C)** Rule-true/false profiles ground representative interpretations (Figure 3). Matthews correlation coefficient (MCC), permutation, and occlusion characterize source-atom correspondence. Orange outlines denote LLM-assisted steps.

sample. The rule-fitting target $a_j(x_i) = \mathbf{1}[c_{ij} > 0]$ equals one when atom j is active and zero otherwise. Atoms active in at least 1% of fitting samples are termed *live atoms* and enter rule construction.

The fitting data depend on the evaluation. For the full-data catalog, the encoder and SAE use all 22 sessions and rules are fitted on 15 sessions. In nested prediction, each outer-training partition is split into 70% discovery trials and 30% rule-selection trials. Normalization, encoder/SAE training, atom-informed proposal evidence, and tree fitting use only discovery trials. Outer-test trials are excluded from both discovery and selection (Table 1; Appendix C.3).

3.2 Rule Construction

Shallow decision trees predict atom activation from input features. LLM-proposed executable features let these trees express combinations of measurements and temporal changes without increasing tree depth.

Rule construction uses three groups of candidate input features (Figure 2B). Base features comprise the 318 model inputs, eight trial labels, and the time-bin index. *Input-informed LLM features* are defined from the input structure and shared across atoms. *Atom-informed LLM features* are proposed from evidence specific to each atom.

Baseline trees use base and input-informed features. The LLM receives out-of-fold examples of correct and incorrect predictions, active/inactive input contrasts, and a baseline rule fitted to all rule-fitting samples. It proposes features from predefined executable operations. All evidence is confined to rule-fitting data.

A class-balanced depth-two tree is fitted for each live atom using all three feature groups. Its rule $p_j(o)$ combines positive root-to-leaf paths by OR, with each path requiring all its input conditions

Table 1: Evaluation questions and fitting scope.

Analysis	Question	Fitting and evaluation scope
Full-data catalog	What interpretable input features were discovered?	All 22 sessions for representation learning.
Nested prediction	Does the discovery procedure find useful features for unseen trials?	Discovery and selection within each outer-training fold, evaluated on outer-test trials.
Cross-session transfer	Are fixed rule-defined features useful in sessions excluded from discovery?	Encoder, SAE, and rules learned from 15 sessions. Ordering and readouts fitted in seven other sessions, evaluated on held-out trials.

to hold for observation o. The LLM proposes candidate features, and the fitted tree determines the final rule. Candidate rule definitions are frozen before prediction-based set selection.

3.3 Feature selection and characterization

We select rule-defined features for their conditional contribution to linear neural prediction. Starting from 318 base inputs and 122 input-informed features, forward selection adds the remaining feature that yields the highest cross-validated mean neural-response R^2 with session-specific Ridge readouts, conditional on previously selected features. The entire addition path is evaluated across four size-evaluation folds, each scoring prefixes on trials excluded from ordering. The number of features K is chosen at the maximum of the cross-validated prediction curve, and the final set comprises the first K features of the refitted ordering (Appendix B.3).

Because full-data representation learning and selection use all 22 sessions, nested prediction repeats both within outer-training data for held-out evaluation (Appendix C.3). Source-atom diagnostics (MCC, trial-block permutation, and model-input occlusion; Appendix B.3) characterize correspondence for each catalog feature.

Evidence-grounded interpretation. For representative prediction-selected rules, the LLM drafts and revises interpretations from exact rule conditions and rule-positive versus rule-negative observations (Figure 2C; Appendix B.4), distinguishing required conditions from additional associations.

4 Empirical evaluation

4.1 Explicit rules describe sensory and behavioral activation patterns

In full-data discovery, forward selection retained 164 rule-defined features at the cross-validated prediction peak. Clustering their binary feature values yielded four exploratory groups: position patterns, choice, and licking; elevated or rising position signals; low positions and restricted ranges; and contact signals, choice, and licking (Figure 3A; Appendix B.7).

In Figure 3B (source atom 111), C1–C3 positions rise from below to above their session means over the recent history. The rule requires an increase in their mean position and an upper bound on summed contact-start signals, and separately detected contact is less frequent than in session- and time-matched feature-negative windows. Figure 3C (137) captures a relative configuration across whiskers: C0 is above its usual position while C3 is below its own. The rule compares the five-bin mean standardized C0 and C3 positions and bounds the four-whisker mean from below one bin before the current bin. In Figure 3D (146), C3 contact angle in feature-positive windows stays 0.27–0.37 SD above its session mean across all five bins, whereas feature-negative windows remain near zero. The rule requires at least one positive C3 contact-angle bin together with current C3 position near or above its session mean.

The three examples illustrate temporal change in position (B), cross-whisker contrast (C), and contact-angle use (D). To test whether these attributes simply reflect the available input vocabulary, we compared the inputs and operations used by the 164 catalog features with those of matched random features sharing each feature's vocabulary, reduced tree structure, and positive prevalence (Table S3; Appendix B.8). Catalog features used all three more often (11.0% versus 2.2%, 26.2%

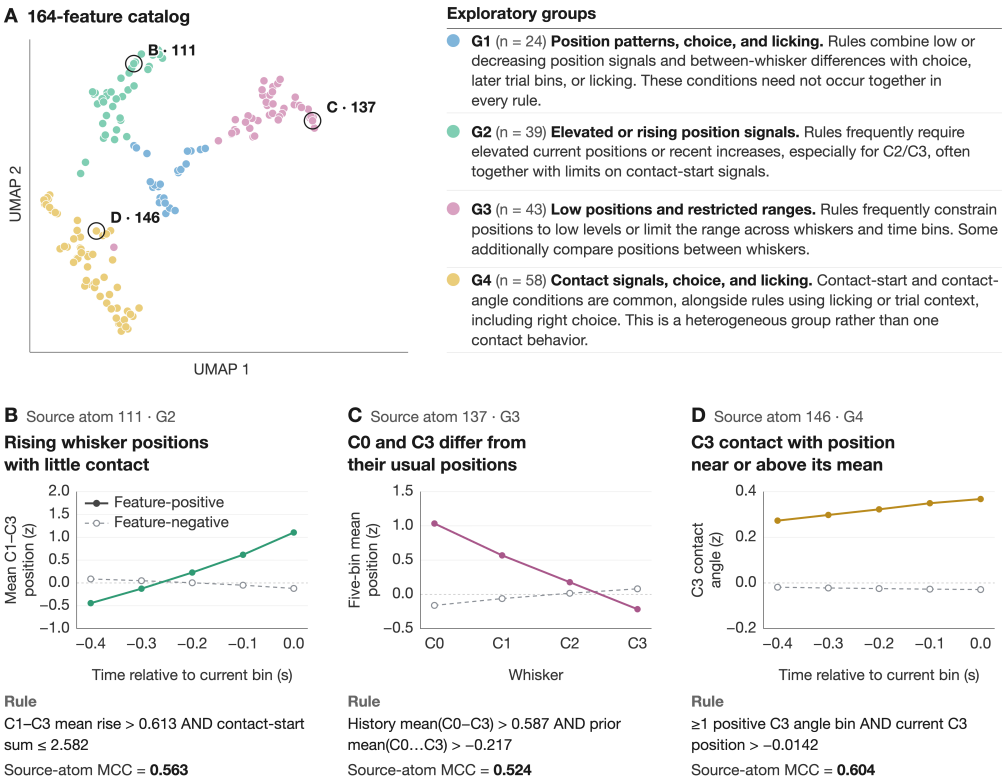

Figure 3: **Organization and illustrative interpretations of 164 rule-defined features. (A)** UMAP of similarity between binary feature-value vectors, colored by group (G1–G4; Appendix B.7). Rings mark the examples in B–D. **(B–D)** Mean session-standardized (z) profiles in feature-positive (rule true; solid) and feature-negative (rule false; dashed) windows of five 0.1-s bins. Headers give the source atom and group. Source-atom MCC measures agreement between the feature and its source atom's activation. **(B)** Mean C1–C3 position over time. **(C)** Five-bin mean position of each whisker. **(D)** C3 contact angle over time, a measurement distinct from the contact-start rate used in B.

versus 4.5% versus 8.1%), as well as explicit time-bin conditions (11.6% versus 1.4%), and each difference recurred in every fold of nested discovery.

Across 20 additional SAE initializations, 63 of 164 rules reached best-match MCC ≥ 0.40 in all 20 runs, versus 6 (range 4–8) matched random counterparts and none under trial-block permutation (Appendix C.2, Figure S1).

4.2 Rule-defined features improve held-out linear prediction and make nonlinear structure linearly accessible

Four-fold trial-grouped nested cross-validation evaluated prediction on unseen trials of the same recorded neurons. Linear Ridge predictors and nonlinear predictors with the population-encoder architecture (Section 3.1), fitted separately for each input set, shared outer splits (Appendix C.3).

We compared four input representations (Figure 4A): base inputs (318 features), base plus input-informed LLM features (440), the same inputs plus fold-specific prediction-selected features (583–606 inputs, with 143–166 features selected per fold), and all 256 continuous SAE activation coefficients learned within outer-training data.

Adding prediction-selected features to the 440 base plus input-informed predictors increased mean linear R^2 from 0.1079 to 0.1254 (gain 0.0175; session-bootstrap 95% confidence interval (CI), 0.0157–0.0195; Figure 4A).

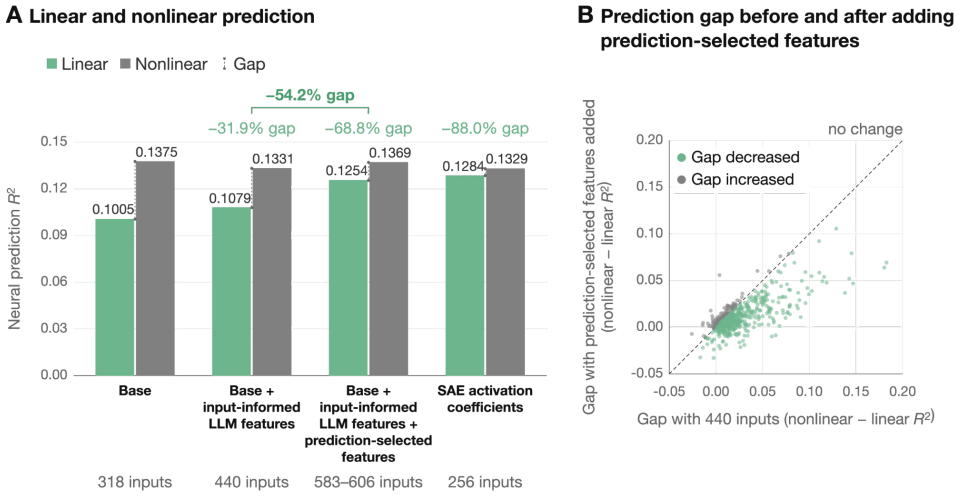

Figure 4: **Neural prediction with nested cross-validation.** **(A)** Linear and nonlinear prediction R^2, averaged over folds per neuron, then across neurons. Prediction-selected features augment base + input-informed LLM features, whereas SAE coefficients replace the inputs. Green percentages show gap reductions relative to Base or, where bracketed, to the 440-feature baseline. **(B)** Each point compares one neuron's gap before and after adding the prediction-selected features to base + input-informed LLM features. The diagonal denotes no change.

These recordings were previously analyzed with different methods for nonlinear sensory integration (Nogueira et al., 2023). Nonlinear R^2 changed little (0.1331 to 0.1369), whereas the linear gain reduced the nonlinear-over-linear gap from 0.0252 to 0.0116, a 54.2% reduction (absolute reduction 0.0137; session-bootstrap 95% CI, 0.0117–0.0157). The gap decreased for 455 of 575 neurons (79.1%; median change -0.0091; Figure 4B). Continuous SAE coefficients, whose encoder and SAE used all outer-training trials rather than the discovery subset, reached linear R^2 0.1284, about 0.003 above the rule-defined features, with gap 0.0044 (Appendix C.3).

Because rule search, unlike the 440-feature baseline, can condition directly on the time-bin index, three controls add a time basis of 31 time-bin indicators to that baseline (Appendices C.4–C.6). With feature sets fixed, prediction-selected features raised linear R^2 from 0.1100 to 0.1261 (gain 0.0161; session-bootstrap 95% CI, 0.0145–0.0177; 92.0% of the original gain), and reselection under this baseline (145–165 features per fold) gave the same R^2. Features defined by matched random rules, matched for candidate count, tree structure, vocabulary, and discovery prevalence, reached 0.1166, so prediction-selected features gave 2.4 times their gain (paired difference 0.0094, CI 0.0085–0.0103).

The third control fitted rules directly to session-specific principal components (PCs) of the neural responses, using the same base and input-informed search features but no atom-informed proposals; they reached 0.1234, 0.0027 below the SAE-guided route (atom targets with atom-informed proposals; CI 0.0021–0.0032). Given the same atom-informed proposals, PC-target rules reached 0.1256, within 0.0004 of the SAE-guided route (CI -0.0001 to 0.0010; Appendix C.6), whereas an equal number of random features from the same 14-operation grammar reached 0.1239 (paired difference 0.00173, CI 0.00141–0.00205; Appendix C.6).

4.3 Transfer of rule-defined features across sessions

Representation learning and feature discovery used 15 sessions, where selection retained all 156 candidate features; this set was then held fixed. In each of seven excluded sessions, feature ordering and Ridge readouts were fitted on training trials and evaluated on separate trials, starting from the 440-feature baseline.

Adding all 156 features improved held-out linear prediction in all seven sessions, with session-level R^2 gains of 0.0137–0.0190. Across 190 neurons, mean R^2 increased from 0.1024 to 0.1190. Prefixes of 20 and 80 features captured approximately 70% and 96% of this gain, respectively. Appendix D reports the procedure, session-level scores (Table S9), and full prediction curves.

5 Related work

Neural selectivity and feature descriptions. Mixed selectivity links response diversity to linear decoding (Rigotti et al., 2013; Fusi et al., 2016; Tye et al., 2024), and population geometry characterizes separability and generalization (Bernardi et al., 2020; Posani et al., 2026). Kaufman et al. (2022) separate parameterization-dependent single-neuron selectivity from fixed population geometry. Encoding models compare hand-designed *linearizing feature spaces* (Naselaris et al., 2011; Naselaris & Kay, 2015), e.g., whisker-afferent force versus angle and barrel-cortex touch, whisking, and task variables (Campagner et al., 2016; Peron et al., 2015; Rodgers et al., 2021, the last on our dataset), or enrich the readout (e.g., GLMs or additive models; Pillow et al., 2008; Lou et al., 2013). We instead fix an affine readout and search over descriptions of fixed measurements.

Sparse components, rules, and stability. Sparse coding and SAEs decompose sensory and learned representations (Olshausen & Field, 1996; Cunningham et al., 2023), but components vary with seed and dictionary size (Paulo & Belrose, 2025; Leask et al., 2025; Fel et al., 2025; Gerasimov et al., 2026). Rule extraction approximates networks with trees (Craven & Shavlik, 1996), and RuleFit, the closest classical method, lasso-selects rules from trees grown on the response (Friedman & Popescu, 2008). We instead use SAE activations both as rule-fitting targets and as evidence for feature proposals, then select features for neural prediction.

Language-model-assisted feature discovery and interpretation. LLMs generate executable features (Hollmann et al., 2023), evolve programs against evaluators (Romera-Paredes et al., 2024; Shojaee et al., 2025; Castro et al., 2025), and describe cortical selectivity (Luo et al., 2024; Matsuyama et al., 2026). HypotheSAEs describes predictive SAE features (Movva et al., 2025), and QA-Emb selects LLM-answered questions for linear fMRI encoding (Benara et al., 2024). Here, the LLM proposes features in one round per atom, the resulting rules are tested for linear compatibility on held-out trials and sessions, and, being executable, they are scored against their atoms (MCC) without the LLM simulator used in explanation scoring (Bills et al., 2023; Singh et al., 2023, Appendix F).

6 Discussion

Because selectivity labels describe a neuron together with a feature description (Proposition 1), we made the description the object of discovery. In mouse S1, the discovered rule-defined features improved held-out linear prediction and reduced the nonlinear-over-linear gap by about half. The gain persisted after adding a time basis and exceeded that of matched random features and of features from rules fitted directly to neural PCs without atom-informed proposals; given those proposals, PC-target rules nearly matched the full method but random features did not, so the proposals carry most of the predictive advantage, whereas atom targets link each rule to a latent component (Section 4.1). Features fixed in 15 sessions also improved held-out prediction in seven other sessions after refitting readout weights. This perspective complements population geometry (Posani et al., 2026): under a common input distribution, conditional-mean geometry bounds ideal compatibility at fixed feature dimension (Appendix E).

The study covers one task, area, and cohort, with retrospective evaluations after method development on these data. Cross-session tests largely reuse animals. Our comparisons cover alternative discovery routes within shallow-tree rule families. Because we assess compatibility through affine readouts, nonlinear response functions can contribute to the residual gap. Candidate proposals depend on stochastic outputs from a proprietary LLM. Causal relevance and usefulness to experimenters require targeted interventions and user evaluation.

The central shift is from assigning selectivity under a fixed feature description to discovering and testing feature descriptions suited to a target neural population. A long-noted caveat thereby becomes a principle for discovery.

AI USE STATEMENT

Beyond their role in the method (Section 3; Appendix B), large language models (ChatGPT, Claude, and Codex) assisted the authors in implementing analysis code, cleaning and reformatting the public dataset, refining the theoretical framework and drafting its proofs, giving feedback on the experimental design, interpreting results, and translation. The authors directed this work and checked all proofs, analysis outputs, and reported numbers. The authors take full responsibility for the content of this paper.

REFERENCES

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E. Hinton. Neural additive models: Interpretable machine learning with neural nets. In *Advances in Neural Information Processing Systems*, volume 34, pp. 4699–4711, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/251bd0442dfcc53b5a761e050f8022b8-Abstract.html.

Robert Andrews, Joachim Diederich, and Alan B. Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 8(6):373–389, 1995. doi: 10.1016/0950-7051(96)81920-4.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6541–6549, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Bau_Network_Dissection_Quantifying_CVPR_2017_paper.html.

Vinamra Benara, Chandan Singh, John X. Morris, Richard J. Antonello, Ion Stoica, Alexander G. Huth, and Jianfeng Gao. Crafting interpretable embeddings for language neuroscience by asking LLMs questions. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3944. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e06f8704aca42f1ed8619165892e1195-Abstract-Conference.html.

Silvia Bernardi, Marcus K. Benna, Mattia Rigotti, Jérôme Munuera, Stefano Fusi, and C. Daniel Salzman. The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell*, 183(4): 954–967.e21, 2020. doi: 10.1016/j.cell.2020.09.031.

Jacob Bien, Jonathan Taylor, and Robert Tibshirani. A lasso for hierarchical interactions. *The Annals of Statistics*, 41(3):1111–1141, 2013. doi: 10.1214/13-AOS1096.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. OpenAI, 2023. URL https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

Dario Campagner, Mathew Hywel Evans, Michael Ross Bale, Andrew Erskine, and Rasmus Strange Petersen. Prediction of primary somatosensory neuron activity during active tactile exploration. *eLife*, 5:e10696, 2016. doi: 10.7554/eLife.10696.

Pablo Samuel Castro, Nenad Tomasev, Ankit Anand, Navodita Sharma, Rishika Mohanta, Aparna Dev, Kuba Perlin, Siddhant Jain, Kyle Levin, Noémi Éltető, Will Dabney, Alexander Novikov, Glenn C. Turner, Maria K. Eckstein, Nathaniel D. Daw, Kevin J. Miller, and Kimberly L. Stachenfeld. Discovering symbolic cognitive models from human and animal behavior. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 6849–6890, 2025. URL https://proceedings.mlr.press/v267/castro25a.html.

Mark W. Craven and Jude W. Shavlik. Extracting tree-structured representations of trained networks. In *Advances in Neural Information Processing Systems*, volume 8, pp. 24–30. MIT Press, 1996. URL https://proceedings.neurips.cc/paper/1995/hash/45f31d16b1058d586fc3be7207b58053-Abstract.html.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. arXiv:2309.08600, 2023. URL https://arxiv.org/abs/2309.08600.

Thomas Fel, Ekdeep Singh Lubana, Jacob S. Prince, Matthew Kowal, Victor Boutin, Isabel Papadimitriou, Binxu Wang, Martin Wattenberg, Demba E. Ba, and Talia Konkle. Archetypal SAE: Adaptive and stable dictionary learning for concept extraction in large vision models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 16543–16572, 2025. URL https://proceedings.mlr.press/v267/fel25a.html.

Jerome H. Friedman and Bogdan E. Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008. doi: 10.1214/07-AOAS148.

Stefano Fusi, Earl K. Miller, and Mattia Rigotti. Why neurons mix: high dimensionality for higher cognition. *Current Opinion in Neurobiology*, 37:66–74, 2016. doi: 10.1016/j.conb.2016.01.010.

Gleb Gerasimov, Timofei Rusalev, Nikita Balagansky, Daniil Laptev, Vadim Kurochkin, and Daniil Gavrilov. Unstable features, reproducible subspaces: Understanding seed dependence in sparse autoencoders. arXiv:2606.12138, 2026. URL https://arxiv.org/abs/2606.12138.

Trevor Hastie and Robert Tibshirani. Generalized additive models. *Statistical Science*, 1(3):297–310, 1986. doi: 10.1214/ss/1177013604.

Noah Hollmann, Samuel Müller, and Frank Hutter. Large language models for automated data science: Introducing CAAFE for context-aware automated feature engineering. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c2df4c35cdbee764ebb9e9d0acd5197-Abstract-Conference.html.

Jing Huang, Atticus Geiger, Karel D'Oosterlinck, Zhengxuan Wu, and Christopher Potts. Rigorously assessing natural language explanations of neurons. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 317–331, 2023. doi: 10.18653/v1/2023.blackboxnlp-1.24. URL https://aclanthology.org/2023.blackboxnlp-1.24/.

Matthew T. Kaufman, Marcus K. Benna, Mattia Rigotti, Fabio Stefanini, Stefano Fusi, and Anne K. Churchland. The implications of categorical and category-free mixed selectivity on representational geometries. *Current Opinion in Neurobiology*, 77:102644, 2022. doi: 10.1016/j.conb.2022.102644.

Patrick Leask, Bart Bussmann, Michael Pearce, Joseph Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse autoencoders do not find canonical units of analysis. arXiv:2502.04878, 2025. URL https://arxiv.org/abs/2502.04878.

Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 623–631, 2013. doi: 10.1145/2487575.2487579.

Andrew F. Luo, Margaret M. Henderson, Michael J. Tarr, and Leila Wehbe. BrainSCUBA: Fine-grained natural language captions of visual cortex selectivity. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.04420.

Takuya Matsuyama, Shinji Nishimoto, and Yu Takagi. LaVCa: LLM-assisted visual cortex captioning. In *International Conference on Learning Representations*, 2026.

Rajiv Movva, Kenny Peng, Nikhil Garg, Jon Kleinberg, and Emma Pierson. Sparse autoencoders for hypothesis generation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 44997–45023. PMLR, 2025. URL https://proceedings.mlr.press/v267/movva25a.html.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17153–17163, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c74956ffb38ba48ed6ce977af6727275-Abstract.html.

Thomas Naselaris and Kendrick N. Kay. Resolving ambiguities of MVPA using explicit models of representation. *Trends in Cognitive Sciences*, 19(10):551–554, 2015. doi: 10.1016/j.tics.2015.07.005.

Thomas Naselaris, Kendrick N. Kay, Shinji Nishimoto, and Jack L. Gallant. Encoding and decoding in fMRI. *NeuroImage*, 56(2):400–410, 2011. doi: 10.1016/j.neuroimage.2010.07.073.

Ramon Nogueira, Chris C. Rodgers, Randy M. Bruno, and Stefano Fusi. The geometry of cortical representations of touch in rodents. *Nature Neuroscience*, 26:239–250, 2023. doi: 10.1038/s41593-022-01237-9. URL https://www.nature.com/articles/s41593-022-01237-9.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. InterpretML: A unified framework for machine learning interpretability. arXiv:1909.09223, 2019. URL https://arxiv.org/abs/1909.09223.

Tuomas Oikarinen, Ge Yan, and Tsui-Wei Weng. Evaluating neuron explanations: A unified framework with sanity checks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 47090–47131, 2025. URL https://proceedings.mlr.press/v267/oikarinen25a.html.

Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996. doi: 10.1038/381607a0.

Gonçalo Paulo and Nora Belrose. Sparse autoencoders trained on the same data learn different features. arXiv:2501.16615, 2025. URL https://arxiv.org/abs/2501.16615.

Gonçalo Santos Paulo, Alex Troy Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48393–48421, 2025. URL https://proceedings.mlr.press/v267/paulo25a.html.

Simon P. Peron, Jeremy Freeman, Vijay Iyer, Caiying Guo, and Karel Svoboda. A cellular resolution map of barrel cortex activity during tactile behavior. *Neuron*, 86(3):783–799, 2015. doi: 10.1016/j.neuron.2015.03.027.

Jonathan W. Pillow, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M. Litke, E. J. Chichilnisky, and Eero P. Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, 2008. doi: 10.1038/nature07140.

Gabriel Poesia and Georgia Gabriela Sampaio. Programmatic representation learning with language models. arXiv:2510.14825, 2025. URL https://arxiv.org/abs/2510.14825.

Stephan Pohl, Edgar Y. Walker, David L. Barack, Jennifer Lee, Rachel N. Denison, Ned Block, Florent Meyniel, and Wei Ji Ma. Clarifying the conceptual dimensions of representation in neuroscience. *Nature Reviews Neuroscience*, 27:357–372, 2026. doi: 10.1038/s41583-026-01030-8.

Lorenzo Posani, Shuqi Wang, Samuel P. Muscinelli, Liam Paninski, and Stefano Fusi. Rarely categorical, highly separable representations along the cortical hierarchy. *Nature*, 2026. doi: 10.1038/s41586-026-10668-4.

Mattia Rigotti, Omri Barak, Melissa R. Warden, Xiao-Jing Wang, Nathaniel D. Daw, Earl K. Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497:585–590, 2013. doi: 10.1038/nature12160. URL https://www.nature.com/articles/nature12160.

Chris C. Rodgers. A detailed behavioral, videographic, and neural dataset on object recognition in mice. *Scientific Data*, 9:620, 2022. doi: 10.1038/s41597-022-01728-1. URL https://www.nature.com/articles/s41597-022-01728-1.

Chris C. Rodgers, Ramon Nogueira, B. Christina Pil, Esther A. Greeman, Jung M. Park, Y. Kate Hong, Stefano Fusi, and Randy M. Bruno. Sensorimotor strategies and neuronal representations for shape discrimination. *Neuron*, 109(14):2308–2325.e10, 2021. doi: 10.1016/j.neuron.2021.05.019.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024. doi: 10.1038/s41586-023-06924-6.

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 *International Conference on Learning Representations*, pp. 16054–16085, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/28df8e730c054c5331855fd4d5403ba9-Abstract-Conference.html.

Chandan Singh, Aliyah R. Hsu, Richard Antonello, Shailee Jain, Alexander G. Huth, Bin Yu, and Jianfeng Gao. Explaining black box text modules in natural language with language models. arXiv:2305.09863, 2023. URL https://arxiv.org/abs/2305.09863.

Kay M. Tye, Earl K. Miller, Felix H. Taschbach, Marcus K. Benna, Mattia Rigotti, and Stefano Fusi. Mixed selectivity: Cellular computations for complexity. *Neuron*, 112(14):2289–2303, 2024. doi: 10.1016/j.neuron.2024.04.017.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1eBeyHFDH.

Jan Ruben Zilke, Eneldo Loza Mencía, and Frederik Janssen. DeepRED – rule extraction from deep neural networks. In *Discovery Science (DS 2016)*, volume 9956 of *Lecture Notes in Computer Science*, pp. 457–473. Springer, 2016. doi: 10.1007/978-3-319-46307-0_29.

A Data preprocessing and representation learning

This appendix gives the data and representation details. Appendix B specifies rule construction, prediction-based selection, and interpretation, Appendix C gives evaluation protocols and statistics, and Appendix D reports the cross-session transfer analysis.

A.1 Dataset and trial selection

We analyzed publicly available mouse primary somatosensory cortex recordings distributed as DANDI 000231 (Rodgers, 2022) and previously studied by Nogueira et al. (2023). The retained cohort comprised 575 neurons from 22 sessions in seven animals, with 4,512 trials and 139,872 sample windows. Each trial contributed 31 prediction bins of 0.1 s. The 318 base inputs comprised 14 sensory channels across five lags and eight current/previous-trial variables expanded across 31 time bins ($70 + 248$ columns). Because the inputs include the current trial's outcome and choice, the models predict responses conditional on these trial events.

Trial selection and temporal alignment. Sessions were restricted to discrimination recordings with behavioral accuracy at least 0.70. Accuracy was computed on non-ignored, random-stimulus trials excluding no-go, spoiled, and optogenetic trials. The input-construction mask used random-stimulus, non-no-go, non-spoiled, non-optogenetic trials and excluded the last trial of each original session. It did not independently apply the ignore-trial flag. The first eligible trial was subsequently omitted to construct previous-trial labels. Signals were aligned to response-window opening and binned from -2.5 to $+1.0$ s in 0.1-s intervals. Five consecutive bins form each input window, yielding 31 windows from the original 35 bins. The response target is the firing rate in the final bin of each window.

A.2 Construction of the 318 input columns

The 14 time-varying channels are signed licking rate, its absolute value, and contact-start rate, contact angle, and angular position for each of whiskers C0–C3. Contact-start counts and spike counts use lower-inclusive, upper-exclusive bins and are divided by 0.1 s. Contact angle is averaged over contacts starting in a bin, with missing values when none start. Angular position is averaged over video samples strictly inside the bin. Signed licking rate counts left licks positively and right licks negatively, using strict bin boundaries. For standardized pre-expansion channel values $\tilde{v}_{t,c}$ and window start b, the sensory block is

$$x_{b,5c+\ell} = \tilde{v}_{b+\ell,c}, \qquad c = 0, \ldots, 13, \quad \ell = 0, \ldots, 4. \tag{4}$$

Thus lag 0 is oldest and lag 4 is current, spanning offsets -0.4 to 0 s relative to the target bin. Each of eight standardized trial variables (previous/current reward, stimulus, position, and choice) occupies a separate 31-column block. Only the column corresponding to the current time index contains that variable's value. The other 30 entries are zero. This gives $14 \times 5 + 8 \times 31 = 318$ inputs. Rule search additionally uses the unexpanded labels in their original coding.

A.3 Normalization and missing values

Normalization precedes lag construction and time expansion. In the full-data catalog, each of the 22 pre-expansion input channels was standardized within session using its mean and population standard deviation over retained trials and all 35 bins, ignoring missing values. Missing standardized input entries were set to zero. Responses were standardized separately for each neuron over the same trial–bin axes.

For nested discovery, channel and response statistics were instead fitted separately within each session on discovery trials and frozen for rule-selection validation and outer-test trials. Prediction-model preprocessing was fitted within its applicable training partition, as specified in Appendix C.3. For session-separated discovery, a common input mean and standard deviation were estimated by pooling only the 15 training sessions and applied unchanged to all 22 sessions. Encoder-response scaling used only each training session's responses. These training-only standardizers use observed training values, set zero-variance scales to one, impute missing inputs to zero after standardization, and set channels with no observed training values to zero. Missing response targets and infinite raw

values are rejected. Consequently, standardized positivity is defined relative to the reference fitted for each analysis, such as the pooled training-session mean in session-separated discovery.

A.4 Population representation and sparse decomposition

For a sample x_i in session s, the population model predicts $\hat{y}_i = R_s E_\theta(x_i) + b_s$, where $E_\theta(x_i) = h_i \in \mathbb{R}^{120}$ and the session-specific output has one entry per recorded neuron. Encoder and readout parameters minimize mean squared prediction error jointly. The encoder has two ReLU hidden layers with dropout. The final affine readout has no output activation.

The SAE is then fitted to normalized hidden vectors:

$$\tilde{h}_i = \frac{h_i}{\|h_i\|_2 + \epsilon}, \qquad c_i = \mathrm{TopK}_k\Big(\mathrm{ReLU}(W\tilde{h}_i)\Big), \qquad \hat{h}_i = Dc_i. \tag{5}$$

Here $W \in \mathbb{R}^{K\times H}$ and $D \in \mathbb{R}^{H\times K}$ are bias-free encoder and decoder matrices learned independently (untied), $\epsilon = 10^{-8}$, and TopK_k retains the k largest entries and sets the remainder to zero. The vector $c_i \in \mathbb{R}^K$ contains the activation coefficients of K components, and $\hat{h}_i$ reconstructs the normalized vector $\tilde{h}_i$. Each column of D defines an atom direction, and c_{ij} is the coefficient of atom j in sample i. We use component and atom for these learned directions. Training minimizes mean squared reconstruction error between $\hat{h}_i$ and $\tilde{h}_i$. Our hidden dimension is $H = 120$, dictionary size $K = 256$, and number of retained coefficients $k = 8$. At most eight coefficients are positive per sample. We define the binary activation indicator for component j as $a_j(x_i) = \mathbf{1}[c_{ij} > 0]$, which equals 1 when the component is active and 0 otherwise. This indicator serves as the target for rule fitting. Components with positive coefficients on at least 1% of the representation-fitting samples are eligible for rule construction (live atoms).

Optimization. Population encoders used two width-120 hidden layers, dropout 0.3, and Adam with learning rate 10^{-3}, weight decay 10^{-4}, batch size 128, and 80 epochs. Sessions were sampled uniformly and rows within a session with replacement. Each epoch contained $\lfloor N_{\mathrm{train}}/128 \rfloor$ parameter updates, where N_{train} is the total number of fitting rows across sessions. SAEs used 256 components, TopK sparsity eight, and 10,000 Adam updates with learning rate 10^{-3} and batch size 4,096.

A.5 Auxiliary dictionary-size diagnostic

A dictionary-size design sweep used unnormalized width-120 encoder states, dictionary sizes 128, 144, 160, 192, 224, and 256, TopK sparsity eight, and seeds 0–4. Unit-normalized decoder directions were matched to seed 0 by Hungarian assignment maximizing total cosine similarity. At dictionary size 256, the mean across seeds 1–4 of the median matched cosine similarity was 0.63. This diagnostic informed dictionary-size selection; decoder cosine similarity measures atom-direction correspondence, whereas Appendix C.2 evaluates rule–activation correspondence across 20 initializations.

B Feature discovery details: rule construction, selection, and interpretation

The procedures below implement the pipeline in Section 3. Fitting and evaluation partitions depend on the analysis, as summarized in Table 1.

B.1 Candidate feature construction and rule fitting

Rules operate on an observation $o_i = (x_i, u_i, t_i)$. Here x_i contains the 318 model inputs, and u_i contains eight trial-level labels: reward, stimulus identity, stimulus position, and choice for the current and previous trials. Each label is repeated across the sample rows of a trial, independently of time. The index $t_i \in \{0, \ldots, 30\}$ identifies the current time bin within that trial. Together, these give 327 base search features. The separate trial labels allow shallow rules to express trial-wide conditions without combining time-specific input columns. Session identity is not a candidate

predictor. We distinguish two sources of LLM-assisted features. *Input-informed LLM features* form a fixed set of 122 features shared across atoms and developed from input structure. *Atom-informed LLM features* are proposed separately for each live atom, based on evidence about its activation.

Input-informed LLM features. Let $v_{mw\ell}$ denote the standardized value for modality m, whisker $w \in \{\mathrm{C0}, \ldots, \mathrm{C3}\}$, and lag $\ell \in \{0, \ldots, 4\}$ in one sample window. The modalities are contact-start rate, contact angle, and angular position. Define $n_{m\ell} = \sum_w \mathbf{1}[v_{mw\ell} > 0]$. Table S1 specifies all 122 columns. Licking and trial variables remain available in the base inputs. The 122 columns are computed from the base inputs and are partly redundant. For example, the per-lag mean is the sum divided by four. Positive entries mark standardized values above the normalization reference, so counts of positive entries or lags give the number of above-reference values rather than detected contacts or contact durations.

Table S1: Definitions of the 122 input-informed LLM features. All operations use standardized inputs. "Range" denotes maximum minus minimum.

Scope	Features	Count
Each modality and lag	Across four whiskers: sum, maximum, minimum, mean, range, and $n_{m\ell}$	90
Contact-start rate, each lag	$\mathbf{1}[n_{m\ell} \geq q]$, for $q = 1, 2, 3$	15
Each modality, entire window	Across 20 whisker–lag entries: maximum, sum, number of positive entries, and range	12
Contact-start rate, entire window	Number of lags with any positive whisker value: $\sum_\ell \mathbf{1}[n_{m\ell} \geq 1]$	1
Contact-start rate, each whisker	Number of positive lags: $\sum_\ell \mathbf{1}[v_{mw\ell} > 0]$	4
Total		122

Atom-informed LLM features. The baseline decision tree has maximum depth two and uses the base features and input-informed LLM features, before adding any atom-informed features. Combining all paths from the root to leaves that predict the active class gives the baseline rule. We call these positive root-to-leaf paths. To propose atom-informed features, the LLM is shown where baseline predictions succeed and where they fail. We construct these examples using only the rule-fitting partition. Trials are divided into three folds, keeping all sample windows from a trial together. For each fold, a baseline decision tree is trained on the other two folds and predicts activation in the remaining fold.

Comparing these predictions with the activation indicator a_j gives four groups: correctly predicted active samples (true positives), missed active samples (false negatives), inactive samples incorrectly predicted as active (false positives), and correctly predicted inactive samples (true negatives). For each live atom, the LLM receives up to three examples from each group, together with the baseline rule extracted from a decision tree fitted to all rule-fitting samples and summaries of input differences between active and inactive samples. It also receives input-channel relevance hints computed from products of model weight matrices. These are approximate because they omit sample-dependent nonlinear gating. We request up to six additional features that could help explain the atom's activation and use one successful proposal response per atom.

Proposals select input columns and one of 14 predefined operations, such as a mean, a difference between whiskers, or a run of consecutive lags above a threshold (Appendix B.2). An optional trial-label condition sets the feature to zero outside that condition. Validated specifications are evaluated with these predefined functions.

Rule fitting. A scikit-learn decision tree (maximum depth two, class balancing, random state 0, and other settings at their defaults) is refitted for every live atom using the 327 base search features, the 122 input-informed features, and valid atom-informed proposals. Class balancing gives active and inactive samples equal total fitting weight. If $\mathcal{B}_j$ is its set of positive root-to-leaf paths and $\mathcal{C}_{jr}$

the conditions on path r, its executable rule is

$$p_j(o) = \mathbf{1}\left[\bigvee_{r\in\mathcal{B}_j}\bigwedge_{v\in\mathcal{C}_{jr}}\{q_v(o)=1\}\right], \qquad q_v(o) = \begin{cases}\mathbf{1}[f_v(o)\le\tau_v], & \text{left-branch condition,}\\ \mathbf{1}[f_v(o)>\tau_v], & \text{right-branch condition.}\end{cases} \tag{6}$$

Here $q_v(o)$ indicates whether observation o satisfies branch condition v, where f_v is the feature tested by that condition and τ_v its fitted threshold. The operators $\wedge$ and $\vee$ mean AND and OR, respectively. The rule predicts 1 if every condition on at least one positive root-to-leaf path is satisfied. All positive leaves contribute to the rule. Predictions use full-precision thresholds; displayed thresholds are rounded.

B.2 Allowed operations for atom-informed proposals

Each proposal specifies an ordered set of distinct columns from the 70 sensory-history inputs (14 channels across five lags), an operation, and an optional trial-label condition. Let $v_1, \ldots, v_m$ be the selected standardized values for one sample, and $\mathbf{1}[\cdot]$ is an indicator. Table S2 defines the 14 operations.

Table S2: Allowed operations for atom-informed feature proposals. Operands are standardized sensory-history inputs. Operand and condition constraints follow the table.

Operation	Definition
Sum	$\sum_{r=1}^{m} v_r$
Mean	$m^{-1}\sum_{r=1}^{m} v_r$
Maximum	$\max_r v_r$
Minimum	$\min_r v_r$
Range	$\max_r v_r - \min_r v_r$
Positive count	$\sum_{r=1}^{m} \mathbf{1}[v_r > 0]$
Product	$\prod_{r=1}^{m} v_r$; at most four operands
Ratio	$v_1/(\lvert v_2\rvert + 10^{-6})$
Signed difference	$v_1 - v_2$
Absolute difference	$\lvert v_1 - v_2\rvert$
All-positive indicator	$\mathbf{1}[v_r > 0$ for all $r]$
Ordered positivity pattern	$\mathbf{1}[v_1 > 0$ and $v_2 \le 0]$
Group contrast	$S_L(v_L) - S_R(v_R)$ for two selected groups
Longest above-threshold run	Largest number of consecutive selected lags with value greater than τ, where $\tau \in \{-1, 0, 1\}$

The ratio, two differences, and ordered positivity pattern require exactly two operands. Operand order defines the sign or direction of these comparisons. For group contrast, each summary S_L, S_R is a sum, mean, minimum, maximum, or range. The two groups must measure the same quantity. A longest-run proposal uses consecutive chronological lags of one sensory channel. The ordered positivity pattern denotes an input-value comparison. Its temporal meaning depends on the selected operands.

For an optional condition on trial label u_q taking value a, the operation output $g(v)$ becomes $g(v)\mathbf{1}[u_q = a]$. The value a must occur in the fitting data. Proposals are checked for permitted operations, operand counts, column indices, and condition values before their features are computed.

B.3 Feature selection and source-atom characterization

Prediction-based rule-set selection. The full-data catalog starts from 176 fixed candidate rules and 440 base plus input-informed predictors. Four trial-grouped size-evaluation folds within each session evaluate the complete path from zero to all candidate rules. On each training portion, four inner folds determine a common forward ordering. For every remaining candidate addition, a session-specific Ridge penalty is selected by summed validation squared error, and candidates are ranked by R^2 averaged over inner folds per neuron and then across neurons. Each step adds the highest-scoring candidate conditional on previously selected rules. Exact candidate ties follow the fixed candidate

order. Scaling of readout inputs and responses is fitted within the applicable training split, while rule definitions and their discovery preprocessing remain frozen.

Each prefix is scored on the size-evaluation fold excluded from ordering. Averaging defined scores over folds per neuron and then across neurons gives the size curve. Near-constant validation responses with undefined R^2 are omitted from this average. We select its maximum, taking the smallest K in an exact tie, then refit the forward ordering by four-fold validation using all catalog-selection trials and retain the first K rules. The reported catalog uses the peak at $K = 164$. The nested benchmark repeats ordering and size selection using only rule-selection trials within each outer-training partition, as specified in Appendix C.3.

Activation agreement. For the full-data catalog, source-atom diagnostics use the original seven rule-evaluation sessions. MCC is the Pearson correlation between binary rule predictions $p_j(o_i)$ and source-atom activation indicators $a_j(x_i)$, assigning zero when either vector is constant. It measures rule–atom correspondence; catalog selection uses neural prediction alone.

Trial-block permutation. Rule predictions are held fixed while complete trial blocks of source-atom activation are permuted within each evaluation session, preserving within-trial bin order. We report observed MCC standardized relative to the mean and standard deviation of 200 permutation MCC values, with a 10^{-12} denominator stabilizer. Unlike the cross-initialization analysis in Appendix C.2, this diagnostic evaluates the original source atom without a best-match search.

Model-input occlusion. We test whether perturbing the inputs used by a rule reduces its atom's activation more than perturbing matched random inputs. Before identifying these inputs, we simplify the tree without changing its predictions by removing splits with identical reduced child subtrees. Rule features are then traced to their underlying encoder-input columns, including behavioral-condition blocks. The selected columns are jointly permuted across trials within each session and time bin, and the fixed encoder and SAE are rerun. For perturbed input x'_i, activation loss is

$$\ell_j = 1 - \frac{\sum_{i \in \mathcal{E}} a_j(x_i) a_j(x'_i)}{\sum_{i \in \mathcal{E}} a_j(x_i)}, \tag{7}$$

where $\mathcal{E}$ is the set of samples used for this evaluation. Thus, ℓ_j is the fraction of originally active samples that become inactive after perturbation. As a control, we perturb randomly selected input sets matched to the rule's input set in the numbers of sensory and time-expanded trial-variable columns. Each of 100 repetitions draws one matched random input set and uses the same trial permutation for the targeted and control perturbations. There are therefore 100 targeted losses and 100 control losses per rule. We report the median targeted loss, the 95th percentile of the control losses (NumPy linear percentile interpolation), and their difference as source-atom sensitivity diagnostics.

B.4 Evidence-grounded interpretation

The 164-rule full-data catalog specifies exact executable conditions. We report sensory and behavioral interpretations for representative rules, using evidence packets that separately contrast rule-positive versus rule-negative samples ($p_j = 1$ versus $p_j = 0$) and source-atom-active versus inactive samples ($a_j = 1$ versus $a_j = 0$). The rule-centered interpretation uses exact conditions, input profiles, trial-label composition, time-bin frequencies, and contact measurements. Atom-conditioned observations support a separate source-atom correspondence account.

Evidence construction and scope. Input profiles, trial-label composition, and time-bin frequencies use all 139,872 windows from the 22 sessions, including rule-fitting and rule-evaluation sessions. Contact-detector summaries use samples with detector records. Context-matched contact differences compare states within strata sharing session, time bin, and the trial labels used by the rule, then average differences with weights proportional to positive-state sample counts. These comparisons are computed separately for rule states and atom states. Atom-informed feature proposals instead use fitting-only evidence.

Generation and evidence review. The LLM produces one draft and one revision using the exact rule, variable definitions, and evidence packet. Automated checks resolve cited evidence identifiers and compare reported measurements with stored values. Semantic review checks condition

scope, label direction, and whether claims extend beyond the measured variables or observations. In particular, OR branches are interpreted separately. Standardized positions do not by themselves establish anatomical direction, and an endpoint increase need not imply a monotonic trajectory. Unsupported claims are narrowed or replaced by supported input-pattern descriptions. These checks test consistency with stored evidence; prose accuracy is not scored independently.

Defining conditions and observational associations remain distinct. A contrast in a variable already constrained by the rule is partly implied by its definition. Contrasts in other variables are reported as associations; for example, contact observed in rule-positive samples can inform the interpretation of a rule that tests licking alone.

Source-atom correspondence and reporting. Source-atom characterization uses the diagnostics defined in Appendix B.3. MCC measures rule–atom agreement, the standardized permutation score compares that agreement with shuffled-trial controls, and occlusion measures sensitivity of modeled atom activation. The reported interpretations retain their exact executable rules and supporting evidence references.

B.5 LLM ROLES AND REQUEST SETTINGS

Input-informed feature construction. LLM assistance was used to construct a shared vocabulary of features from the sensory-input structure, without conditioning on individual atoms. The final set comprises the 122 executable features defined in Table S1. These definitions were fixed and applied to every atom. Computing their values requires no LLM call.

Atom-informed proposals and interpretation. Atom-informed proposals use fitting-partition evidence to suggest additional executable features for each atom (Appendix B.1). Interpretation uses prediction-selected rules and observational evidence to draft and revise descriptions without changing the rules (Appendix B.4). These two stages used Anthropic Claude Opus 4.8 (API identifier `claude-opus-4-8`) with adaptive thinking and structured JSON outputs. Proposal calls used a maximum output budget of 8,192 tokens, increased to 16,384 for a truncation-recovery attempt. Interpretation calls used 32,768 tokens for each draft and revision. Temperature was left at the API default.

Recovery was triggered only by a truncated proposal response and allowed one attempt with the larger token budget. Interpretation used one planned draft and one revision.

Prompts, output schemas, request settings, and corresponding evidence records are provided with the supplementary code.

B.6 AUXILIARY COMPARISON OF FEATURE PROPOSALS

Using the 176 live atoms from the SAE trained for 10,000 updates, the same 15/7 session split, and depth-two tree settings, we compared the 449-column baseline with its augmentation by atom-informed proposals. The baseline included the 122 input-informed LLM features. Mean evaluation MCC increased from 0.3336 to 0.3470 with the adopted contrastive-evidence, 14-operation proposal procedure (mean paired increase, 0.0134). The number of rules reaching MCC ≥ 0.40 increased from 46 to 54, with nine crossing above the threshold and one falling below it. The MCC threshold summarizes rule–atom agreement in this proposal comparison.

B.7 EXPLORATORY ORGANIZATION OF PREDICTION-SELECTED RULES

Figure 3 summarizes 164 rules selected at the cross-validated neural-prediction peak of forward selection. We grouped their fixed binary feature values on all 139,872 sample rows from 22 sessions, using pairwise dissimilarity $d_{jk} = 1 - \mathrm{MCC}(p_j, p_k)$. Average-linkage hierarchical clustering used this matrix directly. Among cuts with 2–10 groups, the highest silhouette score was 0.2197 for four groups, with 24, 39, 43, and 58 rules. Extending the search to 2–20 groups retained the same optimum. Group descriptions summarize recurring rule conditions. UMAP displays the same distance matrix with 10 neighbors, minimum distance 0.1, and seed 0.

Rules from source atoms 111, 137, and 146 illustrate temporal position change, cross-whisker position contrast, and contact-angle use. We chose them as cases with reviewed interpretations, not as cluster medoids. Panel C of Figure 3 averages each whisker's standardized position over the five-bin history, so its horizontal axis is whisker identity; B and D show time relative to the current bin. Supplementary Data 1 lists all 164 features with their group, a plain-language summary, the executable rule, and source-atom diagnostics (MCC, trial-block permutation score, model-input occlusion, and cross-initialization best-match MCC).

B.8 Executable-feature composition and matched random reference

Figure 3A groups features by the similarity of their binary values across observations. Here we instead describe the inputs and computations in their executable definitions. These attributes overlap and are distinct from the four groups. We first analyze the fixed 164-feature full-data catalog selected over the 440-predictor baseline, then assess the consistency of its composition differences in the fold-specific selected features of the nested experiment.

Inventory and classification. We expanded every rule-true path of the reduced trees and resolved every used split-input expression to its saved channel indices, operation, optional trial-label gate, and full-precision threshold. The inventory comprised 164 binary features, 364 feature-specific split-input definitions, 222 rule-true paths, and 425 path-condition occurrences; shared root conditions can occur on more than one path. Across base inputs, fixed input-informed summaries, and atom-informed proposals, the catalog used 15 of 19 supported top-level operator labels. We grouped all 19 into 13 families by their executable meaning (Table S4), retaining families absent from the catalog. Internal group-contrast summaries were preserved separately: 22 features used means and three used sums. These analyst-defined families account exhaustively for the supported operations. We performed this inventory audit after an initial inspection of composition differences.

Domain attributes. We annotated input types, operand relationships, temporal comparisons, and contextual co-use (Table S3). Cross-whisker contrasts compare the same measured quantity across different whiskers or whisker groups; aggregation summarizes values across multiple whiskers. Temporal change requires an arithmetic comparison across lags of the same measured channel or channel group. This criterion is stricter than multiple-lag use: 132 features used multiple lags within an expression, whereas 21 used an arithmetic temporal contrast, including 18 involving position.

Sensory–context co-use requires sensory information and the indicated task label or licking signal on at least one rule-true path; sensory co-use with an explicit time bin follows the same criterion and is listed under temporal structure. The context can be absent in some feature-positive observations: a label-gated expression equals zero outside its gate, and some thresholds accept that zero. Likewise, of 18 features using a longest-run operator, 14 only distinguish absence from presence because the tree threshold is 0.5; four retain distinctions between run lengths. The all-positive indicator, products, and ratios are recorded separately.

Matched reference and scope. For each of the 176 original candidates, one random input rule was generated for each of three seeds (0–2), using the candidate-specific vocabulary, including all valid atom-informed proposals. Reduced tree topology, leaf labels, branch proportions, and overall positive prevalence were matched on the original 15 rule-fitting sessions. The same 164 catalog IDs then defined the paired reference sets. We replayed the saved definitions and applied the same attribute predicates to the catalog and every random set. Random features were not reselected for neural prediction. The comparison therefore characterizes catalog composition relative to this vocabulary and generator, without significance testing. In Tables S3 and S4, rows count catalog features with each attribute (a feature can appear in several rows), random values average seeds 0–2, and differences are catalog minus random-mean percentages in percentage points, computed before rounding. Task-label information includes current- and previous-trial labels, label-gated expressions, and time-gated label columns, which are nonzero in a single trial bin; in Table S4, input-channel values include these columns, trial labels and the time-bin index are counted as separately available metadata, and the three logic rows partition the 164 features, with equal counts across arms because tree topology was matched.

Consistency in nested discovery. We applied the same classifier to the fold-specific features of the nested experiment: 615 feature occurrences selected over the 440-predictor baseline (143/166/152/154 in folds 0–3), and their corresponding matched random features from each of seeds 0–2. Relative to the three-seed mean reference, nested features more often used cross-whisker contrasts (26.0% versus 3.3%), temporal changes in position (10.9% versus 1.9%), and contact angle (15.8% versus 7.8%), and less often used cross-whisker aggregation (53.3% versus 77.8%) and contact-start signals (40.2% versus 61.6%). Each difference had the same sign in all four folds and persisted when both learned and random features were selected under the time-augmented 471-predictor baseline, each using its own saved forward order and selected size. Consistency varied for other attributes. Licking was similar in the full-data catalog (14.0% versus 13.8%) but less frequent in nested discovery (10.6% versus 14.9%). Explicit time-bin conditions were more frequent in both analyses, but their union with time-gated inputs was not consistently enriched (12.2% versus 15.9%; 12.5% versus 11.8%). This difference reflects the greater prevalence of time-gated task-label columns in the matched random features (learned versus random: 1.8% versus 14.4%; 1.1% versus 10.0%). Sensory–task co-use within a rule-true branch changed sign across folds. Sensory co-use with an explicit time bin was also more frequent (7.3% versus 1.4%; 7.2% versus 1.5%), alongside the enrichment of explicit time-bin conditions. Fold-level counts are dependent because outer-training sets overlap.

Connection to the illustrated features. The three illustrated features (Figure 3B–D) exemplify temporal position change, cross-whisker contrast, and contact-angle use, the attributes enriched in the catalog and in every nested fold.

Table S3: Domain composition of the 164-feature catalog versus matched random features. These overlapping attributes place the illustrative cases in Figure 3 in the context of the full catalog.

Comparison dimension	Feature property	Catalog n (%)	Matched random mean (%)	Difference (pp)
Cross-whisker relations	Cross-whisker contrast	43 (26.2)	4.5	+21.7
	Cross-whisker aggregation	88 (53.7)	77.2	−23.6
	Contrast or aggregation	112 (68.3)	78.9	−10.6
Sensory-context co-use	Sensory + task-label information	19 (11.6)	14.2	−2.6
	Sensory + licking	12 (7.3)	12.6	−5.3
	Sensory + licking/task information	29 (17.7)	25.6	−7.9
Temporal structure	Temporal change in position	18 (11.0)	2.2	+8.7
	Temporal change across input types	21 (12.8)	2.2	+10.6
	Explicit time-bin condition	19 (11.6)	1.4	+10.2
	Time-gated task-label columns	3 (1.8)	14.4	−12.6
	Explicit time-bin or time-gated inputs	20 (12.2)	15.9	−3.7
	Sensory + explicit time-bin	12 (7.3)	1.4	+5.9
Input types used (reference)	Whisker position	124 (75.6)	79.3	−3.7
	Contact-start signal	64 (39.0)	58.9	−19.9
	Contact angle	30 (18.3)	8.1	+10.2
	Licking	23 (14.0)	13.8	+0.2
	Task-label information	29 (17.7)	15.0	+2.6

C Evaluation protocols and statistical procedures

C.1 Score aggregation and uncertainty

Rule fidelity was measured by MCC against binary component activation. Neural prediction used test-set R^2, averaged first over four folds per neuron and then over neurons. The nonlinear advantage was $G(F) = \bar{R}^2_{\mathrm{NL}}(F) - \bar{R}^2_{\mathrm{L}}(F)$ for input representation F. Relative gap reduction from F to F'

Table S4: Complete operation-family and logical-structure inventory. Across base inputs, fixed summaries, and atom-informed proposals, all 19 supported top-level operator labels map to these families; the catalog uses 15.

Construction level	Operation / logic	Catalog n (%)	Matched random mean (%)	Difference (pp)
Expression operations	Input-channel value	92 (56.1)	72.4	−16.3
	Trial labels / time-bin input	36 (22.0)	2.2	+19.7
	Direct sum / mean operation	50 (30.5)	36.2	−5.7
	Minimum or maximum	20 (12.2)	31.9	−19.7
	Range (maximum minus minimum)	20 (12.2)	22.8	−10.6
	Signed / absolute / group contrast	62 (37.8)	6.7	+31.1
	Positive-entry / positive-lag count	38 (23.2)	8.9	+14.2
	Positive-count threshold indicator	0 (0.0)	1.6	−1.6
	All-positive indicator	1 (0.6)	0.0	+0.6
	Longest above-threshold run	18 (11.0)	0.4	+10.6
	Product	2 (1.2)	0.4	+0.8
	Ratio	1 (0.6)	0.0	+0.6
	Ordered sign pattern	0 (0.0)	0.0	+0.0
Reduced-tree logic	One threshold	8 (4.9)	4.9	+0.0
	One path with two thresholds	98 (59.8)	59.8	+0.0
	Alternative rule-true paths (OR)	58 (35.4)	35.4	+0.0

was $100[G(F) - G(F')]/G(F)$. Differences, gaps, and relative reductions were computed before rounding. Paired neural-prediction differences used 10,000 session-bootstrap draws, retaining all neurons of each sampled session. Animal-bootstrap intervals were also computed. Thus, neurons contributed equally to the reported means, whereas sessions or animals were the resampling units.

C.2 Correspondence across 20 SAE initializations

This analysis uses the 164 prediction-selected rules in the full-data catalog. The population encoder and all rule definitions remain fixed, while 20 additional SAE dictionaries are trained on the same 22-session dataset using seeds 1–20 and 10,000 updates per dictionary. Correspondence is evaluated on the original seven rule-evaluation sessions under the full-data encoder.

Best-match correspondence. For each rule and seed, we compute the signed MCC between the rule's binary predictions and every destination atom active in at least 1% of all rows, pooling the original seven rule-evaluation sessions. The maximum score defines best-match MCC. A zero denominator gives MCC zero. Matching and scoring use the same samples, different rules can match the same atom, and the matching atom can change across seeds. Best-match MCC thus measures how closely some atom in each new dictionary tracks the rule's binary predictions under the fixed encoder.

Trial-block permutation reference. We generate 200 permutations of the joint rule-prediction matrix by shuffling whole 31-bin trial blocks within each evaluation session (random seed 20260924). The same permutation is applied to all rules and all SAE seeds, preserving within-trial timing, session-specific prevalence, and inter-rule dependence. Destination activations and live-atom sets remain fixed, and the maximum signed MCC over live atoms is recomputed for every rule, seed, and permutation. This reference assumes trial exchangeability within sessions.

Matched random-rule reference. We applied the same best-match search to the three matched random counterparts of each catalog rule described in Appendix B.8 (seeds 0–2). Each counterpart shares its source candidate's vocabulary, including valid atom-informed proposals, reduced tree topology, leaf labels, node split proportions, and positive prevalence on the 15 rule-fitting sessions, and was not reselected for prediction. Because best-match MCC is a maximum over live atoms,

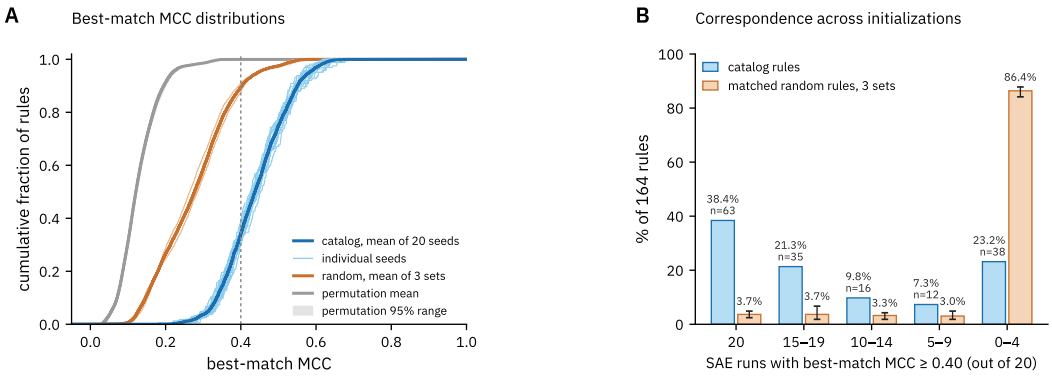

Figure S1: **Rule correspondence across SAE initializations.** The population encoder and 164 prediction-selected rules are fixed across 20 additional SAE initializations. Best-match MCC is the maximum signed MCC between a rule and any live atom. **(A)** Thin blue lines show per-seed empirical cumulative distributions; the thick blue line shows their mean. Orange lines show three sets of matched random rules (thin, each set averaged over seeds; thick, their mean). Grey indicates the mean and pointwise 95% range across 200 within-session trial-block permutations, with best matches reselected. **(B)** Percentage and count of rules by the number of initializations reaching MCC ≥ 0.40, for catalog rules (blue) and matched random rules (orange; mean across the three sets, with bars showing their range). See Appendix C.2 for methods and interpretation.

any input rule of matched form and prevalence can obtain some correspondence. This reference measures that level, whereas the permutation reference removes rule–activation alignment.

For each seed, we compute the empirical cumulative distribution function (ECDF) over the 164 best-match scores. The observed curve averages these 20 ECDFs. For each permutation, we likewise average the 20 seed ECDFs, then summarize these 200 curves by their mean and pointwise 2.5th–97.5th percentiles. Each random set is summarized by the mean of its 20 seed ECDFs.

Results. The pooled median best-match MCC is 0.437 for the observed scores and 0.120 under permutation. Across the 20×164 observed seed–rule scores, 65.8% reach MCC ≥ 0.40, compared with 0.0% under permutation. Of the 164 rules, 63 (38.4%) reach this threshold in all 20 initializations, and 35, 16, 12, and 38 reach it in 15–19, 10–14, 5–9, and 0–4 initializations, respectively (Figure S1). Matched random rules gave a pooled median of 0.276 (0.264–0.285 across the three sets), and 10.5% of their scores reached MCC ≥ 0.40 (9.0–12.3%). On average, 6 of their 164 rules (range 4–8) reached this threshold in all 20 initializations, and 142 (138–144) reached it in 0–4. For 160 of the 164 rules, the median best-match MCC across seeds exceeded the mean of the corresponding random-set medians (median paired difference 0.157). The MCC threshold of 0.40 summarizes correspondence strength; these summaries of dependent seed–rule scores are reported without significance tests.

C.3 Nested trial-held-out prediction

We evaluated unseen trials of the same recorded neurons with four-fold nested cross-validation. Trials were shuffled within each session with seed 0, keeping all 31 rows together. Each outer-training partition was divided into 70% discovery trials for learning representations and candidate rules, and 30% rule-selection validation trials. Normalization, encoder/SAE training, proposal evidence, and tree fitting used discovery data only. For Figure 4, all stored fold-local candidates were eligible for prediction-based forward selection, with rule-set selection based on conditional neural predictive contribution. Selection used only the rule-selection validation trials. Four size-evaluation folds each used four inner folds to determine the forward ordering and per-session/candidate Ridge penalty. The number of rules maximized mean size-evaluation R^2, choosing the smallest size in an exact tie. The ordering was then refitted by four-fold validation on all rule-selection trials, and the selected prefix was frozen. Atom-informed proposals were generated anew for every live atom in each fold. Outer-test trials were reserved for the final prediction evaluation.

After selection, rules and the input standardization estimated from discovery trials were frozen. Linear Ridge and nonlinear prediction models were fitted on outer-training trials and evaluated on outer-test trials. For each session and input condition, Ridge penalties in $\{0.1, 1, 10, 100, 1000, 10000\}$ were selected by four-fold inner trial-grouped validation, minimizing aggregate squared error across responses. Raw-input and response normalization was refitted within each applicable training partition. Nonlinear models used the shared architecture in Appendix A.4, with independent fits for each condition and outer fold. All conditions shared the outer trial partitions.

We compared base inputs (318 columns), base plus input-informed features (440), and that augmented input plus the fold-selected rules. The folds selected 143, 166, 152, and 154 rules, yielding 583, 606, 592, and 594 predictor columns. A separate comparison used all 256 continuous SAE coefficients *instead of* the original inputs. Its encoder and SAE were fitted on all outer-training trials, a superset of the discovery trials used for rule construction, and no live-component filter was applied.

C.4 Detailed prediction scores and time-basis sensitivity

Table S5 collects the original scores underlying Figure 4 and a time-basis sensitivity analysis. Scores are averaged over outer folds per neuron and then across neurons.

Table S5: Neural prediction with and without an independent time basis. L and N are the original linear and nonlinear scores; L_T adds 31 time-bin indicators to the linear inputs. $G = N - L$ and $G_T^{\mathrm{ref}} = N - L_T$. The nonlinear models keep their original inputs, so G_T^{ref} compares linear and nonlinear predictors whose inputs differ by the 31 time indicators. Parenthesized dimensions exclude the 31 added columns.

Input representation	L	L_T	N	G	G_T^{ref}
Base inputs (318)	0.1005	0.1033	0.1375	0.0371	0.0343
Base + input-informed features (440)	0.1079	0.1100	0.1331	0.0252	0.0231
Base + input-informed features + selected rules (583–606)	0.1254	0.1261	0.1369	0.0116	0.0109
SAE coefficients (256; replace inputs)	0.1284	0.1285	0.1329	0.0044	0.0044

Independent trial-time basis. The 318 base inputs contain standardized trial labels gated by time bin, but no independent time-bin indicators. Rule search additionally accesses the trial-bin index. We therefore appended all 31 one-hot time indicators to every linear condition, including SAE coefficients. Indicators remain unscaled and receive the same Ridge penalty as other features, and the intercept remains unpenalized. Positive Ridge penalties resolve the indicator/intercept redundancy. We retained the original outer partitions, discovery models, rule definitions, and selected sets. Each session's Ridge penalty was retuned using the original four training-only inner folds and penalty grid. Input/response normalization followed the original training-partition procedure. The SAE comparator replayed its frozen outer-training representation. Only the linear readouts were refitted. This is a fixed-rule sensitivity analysis.

With time indicators, adding the fixed rules increased mean linear R^2 from 0.1100 to 0.1261 (gain 0.0161, session-bootstrap 95% CI 0.0145–0.0177, animal-bootstrap 95% CI 0.0153–0.0168). The gain retained 92.0% of its original value. Its paired change was -0.0014 (session-bootstrap 95% CI -0.0019 to -0.0010). With the original nonlinear scores fixed, the reference gap decreased from 0.0231 to 0.0109 (53.0%, absolute reduction 0.0123).

After removing splits with identical child subtrees, the time-bin index appears explicitly in 19/164 full-data catalog rules (11.6%), in 17/143, 20/166, 21/152, and 13/154 nested selected rules (11.9%, 12.0%, 13.8%, and 8.4%), and in 26/156 cross-session transfer rules (16.7%).

C.5 Matched random-rule controls

To test learned rule construction against random input-threshold rules, we constructed three prespecified control sets (seeds 0, 1, and 2) for each outer fold. Candidate counts were 157, 166, 157, and 154. Each control rule retained its corresponding learned candidate's reduced tree topology, leaf labels, and available vocabulary, namely the 449 base/input-informed search features plus that

candidate's discovery-derived LLM proposals. Features and thresholds were randomized using discovery inputs only. At each node, a feature was sampled in random order and a threshold chosen uniformly from midpoints yielding a conditional left-branch fraction within 0.025 of the learned node. Each branch required at least five discovery rows. Rules were resampled until their discovery prevalence differed from the learned rule by at most $\max(0.002, 0.1\min(p, 1-p))$, where p is the learned rule's prevalence. All candidate sets were fixed before outer evaluation.

Every arm used 440 baseline features plus 31 time-bin indicators. We reran the same complete forward-ordering and peak-K selection procedure using only the original rule-selection partition, then retuned Ridge on outer-training trials and evaluated outer-test trials. Table S5 instead keeps the original rule sets (143, 166, 152, and 154 rules) and adds only the time indicators, whereas Tables S6 and S7 reselect rules under the time-augmented baseline (145, 165, 151, and 154 rules). The two analyses give mean R^2 of 0.126066 and 0.126062, which agree after rounding. The primary comparison uses each arm's own training-selected K. Secondary comparisons use the prediction-selected arm's training-selected K for every arm, or all candidates. All three random seeds are reported. The paired bootstrap resamples sessions or animals after averaging outer-fold scores per neuron and, for the control mean, averaging the three seeds.

The prediction-selected arm exceeded the seed-mean random control at its own selected size, with mean R^2 0.1261 versus 0.1166 and a paired difference of 0.0094 (session-bootstrap 95% CI 0.0085–0.0103, animal-bootstrap 95% CI 0.0089–0.0100). At the same fold-specific K, the paired difference remained 0.0094 (session-bootstrap 95% CI 0.0085–0.0103). Using all learned candidates yielded $R^2 = 0.1260$, compared with 0.1261 after selection, so the learned-candidate result was also robust to omitting subset selection. This comparison holds the discovery-derived LLM vocabulary available to the random controls. It assesses learned rule construction within that vocabulary.

Table S6: Matched random-rule controls with a common time-augmented baseline. K lists the four outer folds. Own-K uses each arm's selected count, and common-K uses the prediction-selected arm's count in each fold. All uses every candidate. Entries are held-out mean linear R^2.

Arm	K (folds 0–3)	Own-K	Common-K	All
Prediction-selected rules	145/165/151/154	0.1261	0.1261	0.1260
Random seed 0	153/159/128/153	0.1165	0.1165	0.1165
Random seed 1	155/152/157/147	0.1167	0.1167	0.1167
Random seed 2	151/160/128/139	0.1168	0.1168	0.1167
Baseline (440 + time)	—		0.1100	

C.6 Direct neural-PC rules and a PC-target variant

We next tested a direct neural-target discovery route using the same fold-specific candidate budgets (157, 166, 157, and 154). Because different neurons were recorded in each session, we performed principal component analysis (PCA) separately on each session's standardized discovery responses, recentered over the 31 aligned trial bins. Budgets were allocated to sessions in proportion to their neuron counts by largest remainder, breaking ties by session order. We used that many leading principal components (PCs) per session, orienting each PC so its largest absolute loading was positive. For each PC, a depth-2 squared-error regression tree with at least five discovery rows per leaf predicted its score. Leaves with predicted scores above zero defined the binary rule. Identical-label subtrees were collapsed. These input rules were then applied to every session. PCA and tree fitting used discovery trials only.

Direct trees used only the 449 base and input-informed search features, namely 318 model inputs, eight behavioral labels, the trial-bin index, and the 122 input-informed features of Table S1. They used no atom-informed LLM proposals, atom activations, or encoder/SAE checkpoints. Every constructed candidate was retained. All arms used the time-augmented baseline and the selection and outer-evaluation procedure of Appendix C.5. Candidate counts and maximum depth were matched, but exact topology and prevalence were not constrained. The SAE-guided route retained its atom-informed proposal vocabulary and pooled-session representation, whereas direct trees used session-local neural targets.

At each arm's training-selected size, direct PC rules yielded mean $R^2 = 0.1234$ versus 0.1261 for prediction-selected rules from the SAE-guided route. The paired advantage of prediction-selected rules was 0.0027 (session-bootstrap 95% CI 0.0021–0.0032, animal-bootstrap 95% CI 0.0019–0.0033). It was 0.0025 at the same fold-specific K and 0.0026 with all candidates (session-bootstrap 95% CI 0.0020–0.0031 for both).

PC-target variant with SAE-derived proposals. A variant of our method kept the SAE-derived proposals but replaced atom targets with the neural-response PCs above. Each PC candidate slot received the discovery-derived atom-informed LLM proposals of the corresponding learned candidate slot, in addition to the 449 search features. PCA targets, budgets, tree settings, selection, and evaluation matched the direct route above. PC-target rules reached $R^2 = 0.1256$ at their own selected size (131/143/151/151 rules), versus 0.1261 with atom targets. The difference was 0.0004 (session-bootstrap 95% CI -0.0001 to 0.0010, animal-bootstrap 95% CI -0.0004 to 0.0014), and also 0.0004 at the same fold-specific K and with all candidates, with session-bootstrap intervals that included zero. PC-target rules thus obtained 97% of the atom-target gain over the baseline (0.0157 versus 0.0161). In this comparison, target type varies together with session pooling: atom-target fitting pools sessions, whereas PC-target fitting is session-local.

Effect of the proposals. The proposal-free direct route and the PC-target variant share outer folds, neurons, baseline, PC targets, and candidate budgets, and differ only in whether atom-informed proposals were available. We therefore compared them from the saved per-neuron scores, with the session and animal bootstrap of Appendix C.1 (10,000 draws, seed 0). Adding the proposals raised mean R^2 by 0.0022 at each arm's own selected size (session-bootstrap 95% CI 0.0019–0.0025, animal-bootstrap 95% CI 0.0017–0.0025). The gain was 0.0021 at the same fold-specific K and with all candidates (session-bootstrap 95% CI 0.0018–0.0024 for both). Session-average gains on held-out trials were positive in all 22 sessions, and 445 of 575 neurons improved. The SAE-derived proposals thus also benefited the PC-target route, closing most of its gap to the default pipeline.

Table S7: Prediction-selected rules (atom targets) versus rules fitted to neural-response PCs with or without the SAE-derived proposals, using the common time-augmented baseline. Conventions follow Table S6. All scores are held-out mean linear R^2.

Arm	K (folds 0–3)	Own-K	Common-K	All
Prediction-selected rules (atom targets)	145/165/151/154	0.1261	0.1261	0.1260
PC-target rules with SAE proposals	131/143/151/151	0.1256	0.1257	0.1255
Direct neural-PC rules, no SAE proposals	129/146/139/151	0.1234	0.1235	0.1234
Baseline (440 + time)	—		0.1100	

Equal-count random feature proposals. To distinguish the benefit of the existing proposals from merely expanding the candidate vocabulary, we replaced them with non-LLM random features while keeping the session-local PC targets fixed. This differs from the random-rule control in Appendix C.5: here feature definitions are randomized and trees are fitted to the same PC targets, rather than randomizing rule thresholds over a fixed vocabulary. The 449 base and input-informed search features remain available in all conditions; "no proposals" therefore means no additional atom-informed proposals, not removal of the fixed input-informed features.

For each PC candidate slot, the number of additional features exactly matched the saved LLM-proposal condition. We used generator seeds 0, 1, and 2 with NumPy SeedSequence([20260926, seed, outer fold, candidate slot]). Operators were sampled uniformly from the 14-operation grammar in Appendix B.2. Quantity-compatible groups were sampled uniformly where required by the grammar, then operands without replacement: two for binary operators, two to four for products, and two to five for other groups. Run features used two to five consecutive lags from one channel and thresholds sampled uniformly from $\{-1, 0, 1\}$. Group-contrast summaries were sampled uniformly from sum, mean, minimum, maximum, and range. With probability 0.25, a feature was gated by a uniformly sampled trial-label variable and one of its discovery-observed levels. Operator frequencies and prevalence were not matched to the LLM proposals.

Using source-session discovery inputs only, we rejected features with nonfinite or constant values, or exact float32-value duplicates of a base or already generated feature. The maximum was 10,000 draws per accepted feature, with no fallback based on predictive performance. Proposal generation

used no neural target scores, atom evidence, selection trials, or outer-test data. The generator distribution was fixed before its outer evaluation, as a follow-up to the earlier ablations. All 12 candidate sets (four folds by three seeds) were constructed before evaluating their outer-test performance.

The PCA targets were numerically verified against both saved PC-target comparators. Candidate budgets (157/166/157/154), depth-two regression trees, minimum leaf size five, binarization, rule selection, time-augmented baseline, and outer evaluation followed the preceding PC-target analysis without modification. Comparisons used each arm's internally selected K; secondary comparisons used the same fold-specific K as the original time-augmented SAE-guided arm, or all candidates. We averaged the four fold scores for each neuron and compared conditions using 10,000 paired session- and animal-cluster bootstrap draws, as in Appendix C.1. Random seeds were averaged within neuron, not treated as independent biological replicates; all seed-specific results are reported.

Table S8: PC-target rules with no additional proposals, random grammar proposals, or SAE-informed LLM proposals. All conditions share PC targets and downstream evaluation. Random and LLM conditions have equal additional-feature counts per candidate slot. Entries are mean held-out linear R^2.

Proposal source	K (folds 0–3)	Own-K	Common-K	All
None	129/146/139/151	0.12341	0.12353	0.12339
Random seed 0	129/142/149/154	0.12391	0.12400	0.12386
Random seed 1	132/138/137/151	0.12378	0.12390	0.12378
Random seed 2	132/141/150/151	0.12399	0.12406	0.12392
Random seed mean	—	0.12390	0.12399	0.12385
SAE-informed LLM	131/143/151/151	0.12562	0.12566	0.12552

At each arm's selected size, random proposals increased R^2 over no additional proposals by 0.00049 (session-bootstrap 95% CI 0.00039–0.00059; animal-bootstrap CI 0.00034–0.00060). The SAE-informed LLM proposals exceeded the random-seed mean by 0.00173 (session-bootstrap CI 0.00141–0.00205; animal-bootstrap CI 0.00125–0.00204). The advantage over each individual seed was 0.00163–0.00184, with both cluster-bootstrap intervals excluding zero for every seed. The advantage persisted at common K (0.00167; session-bootstrap CI 0.00136–0.00198) and with all candidates (0.00167; CI 0.00136–0.00197). These results support the utility of the current proposals relative to this prespecified random generator, not the necessity of an LLM or superiority to every non-LLM search procedure. They do not isolate the benefit of SAE evidence from that of the language model.

D Prediction in sessions excluded from feature discovery

D.1 Analysis design

Cross-validated size selection within the 15 discovery sessions retained all 156 candidate rules at the peak of its prediction curve. Representation learning and feature discovery used 97,681 sample rows from 15 sessions. Input standardization was fitted on these sessions and frozen when applying rules to 42,191 rows from the seven excluded sessions. The SAE used 10,000 updates. Atom-informed proposals used training-only evidence. The fixed rules were subsequently evaluated for neural-response prediction within each excluded session.

We held fixed the 156-rule set selected within the 15 discovery sessions and evaluated its incremental predictive value separately in each of the seven excluded sessions (190 neurons). The baseline was a linear Ridge model with 440 base plus input-informed features. Within each session, four outer trial-grouped folds (seed 0) separated readout fitting from evaluation. Four inner trial-grouped folds within each outer-training partition (seed $100 + f$ for outer fold f) selected the next rule by its conditional contribution to mean neural-response R^2. Ridge penalties were selected by inner summed squared error. Rule order therefore varied across sessions and outer folds.

D.2 Session-level prediction results

Every prefix from zero to 156 rules was evaluated on the outer-held-out trials. Scores were averaged over folds per neuron, then over neurons within a session. The combined curve weights sessions

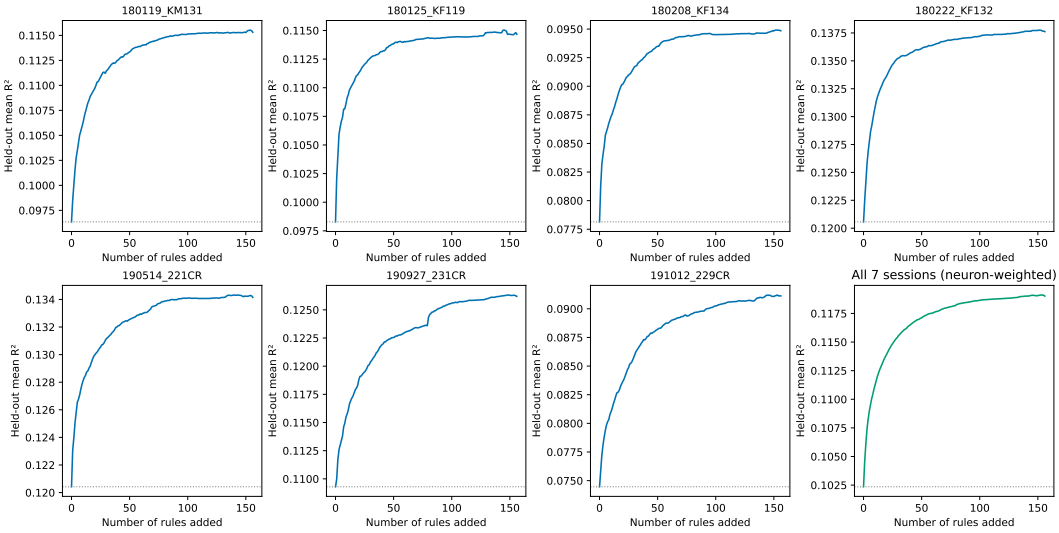

Figure S2: **Incremental prediction in seven excluded sessions.** Each curve shows held-out mean linear R^2 as the fixed rules from the 15 discovery sessions are added to 440 base plus input-informed features. Ordering uses inner training folds separately for each session and outer fold. Dotted lines mark the zero-rule baseline. The final panel averages over 190 neurons. Vertical scales differ across panels.

by their neuron counts. Adding all 156 rules improved mean R^2 in all seven sessions, with gains from 0.0137 to 0.0190. Across the 190 neurons, mean R^2 increased from 0.102353 to 0.119014 ($\Delta R^2 = 0.016661$). Table S9 reports selected points and Figure S2 shows the complete curves.

Table S9: Session-specific held-out linear R^2 after adding the fixed rules from the 15 discovery sessions. Counts are fixed summary points. The final column compares 156 rules with the 440-feature baseline. The combined row averages over all 190 neurons.

Session	0 rules	20 rules	80 rules	156 rules	ΔR^2
180119_KM131	0.096347	0.109703	0.114797	0.115310	0.018963
180125_KF119	0.098281	0.111293	0.114373	0.114676	0.016395
180208_KF134	0.078143	0.090180	0.094416	0.094853	0.016710
180222_KF132	0.120565	0.133670	0.136924	0.137619	0.017054
190514_221CR	0.120419	0.129975	0.133874	0.134149	0.013731
190927_231CR	0.109295	0.118768	0.124332	0.126202	0.016907
191012_229CR	0.074464	0.083649	0.089563	0.091124	0.016660
All neurons	0.102353	0.113953	0.118269	0.119014	0.016661

E ANALYTICAL PROPERTIES OF LINEAR COMPATIBILITY

These standard least-squares identities and spectral characterization formalize the response–feature relation in Section 2.3. They clarify which changes of description preserve linear compatibility, when additional features improve it, and how conditional-mean geometry constrains its attainable value at a fixed feature dimension. The results concern optimal affine prediction for fixed neural targets and evaluation distributions. The final paragraph relates this criterion to the empirical Ridge procedure. Figure S3 illustrates the broader affine-plus-monotone model introduced in Section 2.1.

Setting. Let P_n be neuron n's evaluation distribution of (X, Y_n), with $0 < V_n = \mathrm{Var}_n(Y_n) < \infty$. The shared functions $F = (f_1, \ldots, f_p)$ are square-integrable under each P_n. When F is estimated from data, we condition on the fitted functions and evaluate expectations on an independent draw from P_n. Write $\mu_n(X) = \mathbb{E}_n[Y_n \mid X]$ and define

$$\mathcal{V}_{n,F} = \mathrm{span}_{L^2(P_n)}\{1, f_1(X), \ldots, f_p(X)\}, \qquad \Pi_{n,F} : L^2(P_n) \to \mathcal{V}_{n,F}, \tag{8}$$

where $\Pi_{n,F}$ is the orthogonal projection onto the finite-dimensional space, allowing linearly dependent features. The neuron-specific compatibility and population average are

$$C_{n,\mathrm{lin}}(F; P_n) = 1 - \frac{\mathbb{E}_n[(Y_n - \Pi_{n,F} Y_n)^2]}{V_n}, \qquad \mathcal{C}_{\mathrm{lin}}(\mathcal{N}, F; \mathcal{P}) = \frac{1}{M} \sum_{n=1}^{M} C_{n,\mathrm{lin}}(F; P_n). \tag{9}$$

Property 1: invariance to a change of affine coordinates. If $\widetilde{F}(x) = AF(x) + a$ with invertible A, then $\mathcal{V}_{n,\widetilde{F}} = \mathcal{V}_{n,F}$ for every neuron, and hence

$$\mathcal{C}_{\mathrm{lin}}(\mathcal{N}, \widetilde{F}; \mathcal{P}) = \mathcal{C}_{\mathrm{lin}}(\mathcal{N}, F; \mathcal{P}). \tag{10}$$

Derivation. The forward transformation and $F = A^{-1}(\widetilde{F} - a)$ establish equality of the prediction spaces, so their optimal risks coincide. This holds for any descriptions with the same affine span, even when their coordinates give different pure/mixed selectivity descriptions. Different spans can also achieve the same score for a particular target.

Property 2: approximation of the conditional mean. Define $d_n^2(F) = \mathbb{E}_n[(\mu_n(X) - \Pi_{n,F}\mu_n(X))^2]$. Then

$$C_{n,\mathrm{lin}}(F; P_n) = \frac{\mathrm{Var}_n(\mu_n(X)) - d_n^2(F)}{V_n}. \tag{11}$$

Derivation. For every $v \in \mathcal{V}_{n,F}$, conditional-mean orthogonality gives

$$\mathbb{E}_n[(Y_n - v)^2] = \mathbb{E}_n[\mathrm{Var}_n(Y_n \mid X)] + \mathbb{E}_n[(\mu_n(X) - v)^2]. \tag{12}$$

Minimizing over v and applying total variance yields equation 11. At fixed P_n, higher compatibility means smaller L^2 approximation error of the conditional mean. The ideal score lies between zero and $\mathrm{Var}_n(\mu_n(X))/V_n \le 1$. This upper limit is the prediction ceiling given the observed inputs X, attained when the affine span contains μ_n. The aligned scalar feature alone and the distributed pair in Figure 1B,C both attain this ceiling for the illustrated response, although their spans differ.

Property 3: conditional contribution of an added rule. For a fixed binary rule $p_j(X)$, let

$$r_{nj} = p_j(X) - \Pi_{n,F} p_j(X), \qquad e_{n,F} = Y_n - \Pi_{n,F} Y_n. \tag{13}$$

If $\mathbb{E}_n[r_{nj}^2] > 0$, its incremental compatibility is

$$C_{n,\mathrm{lin}}((F, p_j); P_n) - C_{n,\mathrm{lin}}(F; P_n) = \frac{\{\mathbb{E}_n[e_{n,F} r_{nj}]\}^2}{V_n\, \mathbb{E}_n[r_{nj}^2]}. \tag{14}$$

If $r_{nj} = 0$ almost surely, the gain is zero. The population gain averages these neuron-specific quantities.

Derivation. The augmented space is $\mathcal{V}_{n,F} \oplus \mathrm{span}\{r_{nj}\}$. Projecting $e_{n,F}$ onto r_{nj} reduces squared error by $\{\mathbb{E}_n[e_{n,F} r_{nj}]\}^2 / \mathbb{E}_n[r_{nj}^2]$, giving equation 14. Because the projection is linear, a nonlinear rule of already observed inputs can add predictive value without adding a measurement. This conditional contribution differs from fidelity to a source atom's activation pattern.

Remaining-error bound. Combining Properties 2 and 3 gives

$$C_{n,\mathrm{lin}}((F,p_j);P_n) - C_{n,\mathrm{lin}}(F;P_n) = \frac{d_n^2(F) - d_n^2((F,p_j))}{V_n} \le \frac{d_n^2(F)}{V_n}. \tag{15}$$

Equality holds exactly when the augmented affine span contains μ_n. For nonzero residual rule variance and remaining approximation error, this requires r_{nj} to be proportional to $\mu_n - \Pi_{n,F}\mu_n$ almost surely. Thus a rule's gain is limited by the conditional-mean structure that the current description leaves unexplained.

Property 4: sensitivity to information-preserving nonlinear re-description. Let $\widetilde{F} = G \circ F$, where G is measurable with a measurable inverse on its range, and both descriptions are square-integrable. Mutual information is preserved:

$$I_n(\widetilde{F}(X); Y_n) = I_n(F(X); Y_n). \tag{16}$$

An invertible nonlinear recoding can nevertheless change the affine prediction space and hence linear compatibility. Property 1 guarantees invariance for affine-span-preserving recodings. This distinguishes information content from accessibility under a restricted readout class (Pohl et al., 2026; Xu et al., 2020).

For example, let $X = (A, B)$ consist of independent, equiprobable $\{-1, 1\}$ variables and let $Y = AB$. The descriptions $F = (A, B)$ and $\widetilde{F} = (A, AB)$ are invertibly related because $B = A(AB)$. Both determine Y, yet their linear compatibility scores are zero and one, respectively, because Y is orthogonal to $1, A, B$ but is a coordinate of $\widetilde{F}$. The response and its information content are fixed while linear accessibility changes. This example is part (iii) of Proposition 1 in Section 2.3.

Property 5: the supremum over information-preserving recodings. This proves Proposition 1(ii). Let $\mathcal{G}$ be as defined there, and write $m_n(f)$ for a measurable version of $\mathbb{E}_n[Y_n \mid F(X) = f]$. For $G \in \mathcal{G}$, the variables $G(F(X))$ and $F(X)$ generate the same σ-algebra under each P_n. Every affine function v of $G(F(X))$ is therefore a function of $F(X)$, and the argument of equation 12 with $F(X)$ in place of X gives

$$\mathbb{E}_n[(Y_n - v)^2] = \mathbb{E}_n[\mathrm{Var}_n(Y_n \mid F(X))] + \mathbb{E}_n[(m_n(F(X)) - v)^2] \ge \mathbb{E}_n[\mathrm{Var}_n(Y_n \mid F(X))]. \tag{17}$$

Hence $\mathcal{C}_{n,\mathrm{lin}}(G \circ F; P_n) \le \mathrm{Var}_n(m_n(F(X)))/V_n$ for every n, which bounds the population average.

For the reverse inequality, let $c_T(u) = \max\{-T, \min\{T, u\}\}$ and

$$G_T(f) = \big(f,\, c_T(m_1(f)), \ldots, c_T(m_M(f))\big). \tag{18}$$

The projection onto the first block inverts G_T, and the appended coordinates are bounded, so $G_T \circ F$ is square-integrable under each P_n and $G_T \in \mathcal{G}$. Reading out the coordinate $c_T(m_n(F(X)))$ gives error $\mathbb{E}_n[\mathrm{Var}_n(Y_n \mid F(X))] + \mathbb{E}_n[(m_n - c_T(m_n))^2]$. Since $|m_n - c_T(m_n)| \le |m_n|$ and $m_n(F(X))$ is square-integrable under P_n, dominated convergence sends the second term to zero as $T \to \infty$. Each neuron's compatibility thus approaches $\mathrm{Var}_n(m_n(F(X)))/V_n$, and so does the average.

Truncation may be needed when the P_n differ. A conditional mean that is square-integrable for one neuron's distribution need not be square-integrable for another's. For example, with $F(X) = T_0 \in (0, 1)$, density $3t^2$ under P_1 and uniform under P_2, the response $Y_1 = 1/T_0$ has $\mathbb{E}_1[Y_1^2] = 3$ but $\mathbb{E}_2[m_1(T_0)^2] = \infty$. If all P_n share the distribution of $F(X)$, every $m_n(F(X))$ is square-integrable under it, and $G(f) = (f, m_1(f), \ldots, m_M(f))$ attains the supremum. The construction adds M coordinates. At fixed dimension the attainable value can be lower, as the following spectral bound shows, while part (iii) shows that information-equivalent descriptions of equal dimension can still differ maximally.

A reference bound from conditional-mean geometry. Suppose all P_n share the same input marginal P_X. Let $\mathcal{F}_p$ contain all shared square-integrable feature descriptions with at most p non-constant coordinates, without interpretability constraints. Define

$$u_n(X) = \frac{\mu_n(X) - \mathbb{E}_n[Y_n]}{\sqrt{V_n}}, \qquad K_{nm} = \mathbb{E}_{P_X}[u_n(X)u_m(X)]. \tag{19}$$

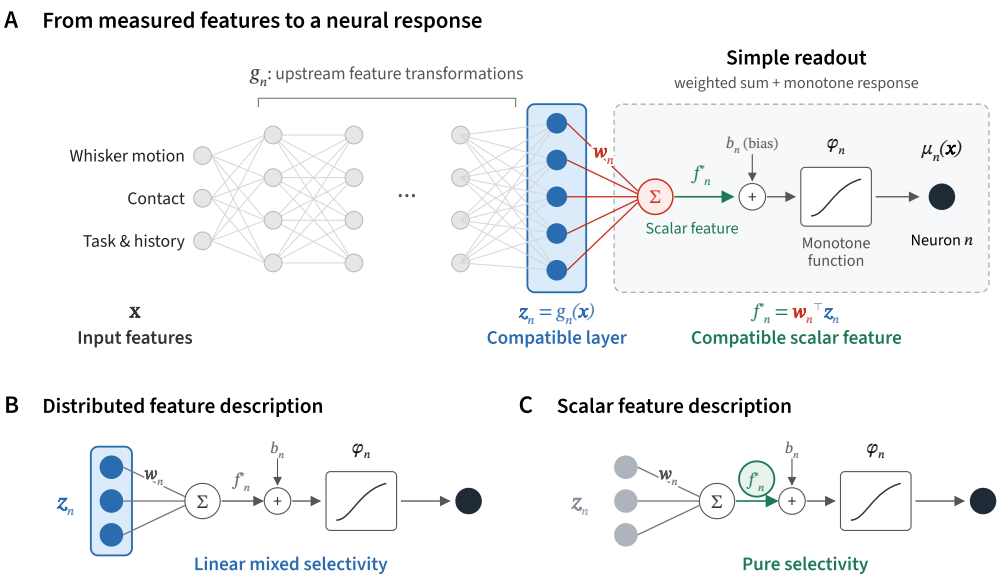

Figure S3: **A latent-input model of neuronal responses.** Schematic of equation 1. **(A)** An unobserved transformation maps researcher-defined input features x to latent inputs $z_n = g_n(x)$. Red connections denote readout weights w_n, defining the scalar feature $f_n^* = w_n^\top z_n$. Adding bias b_n and applying a monotone response function gives the mean response $\mu_n(x)$. **(B)** The coordinates of the compatible layer z_n jointly contribute to the readout, yielding a linearly mixed description. **(C)** The scalar feature yields a purely selective description of the same response. Both are compatible with the affine-plus-monotone readout.

For eigenvalues $\lambda_1(K) \geq \cdots \geq \lambda_M(K) \geq 0$,

$$\sup_{F \in \mathcal{F}_p} \mathcal{C}_{\text{lin}}(\mathcal{N}, F; \mathcal{P}) = \frac{1}{M} \sum_{k=1}^{\min(p,M)} \lambda_k(K). \tag{20}$$

Derivation. With centered feature span S, the score is $M^{-1} \sum_n \|\Pi_S u_n\|^2_{L^2(P_X)}$. The finite-rank operator $T = \sum_n u_n \otimes u_n$ has the same nonzero eigenvalues as its Gram matrix K. Maximizing the projected sum over subspaces of dimension at most p retains up to p leading nonzero eigenfunctions. They are linear combinations of the normalized conditional means u_n and thus admissible features.

The spectrum of variance-normalized conditional-mean responses therefore sets the best achievable compatibility at a fixed feature dimension. Discovering an interpretable description asks how well a constrained feature span approximates these responses. This population reference bound is not estimated here.

Relation to the empirical procedure. Property 3 motivates the conditional-contribution criterion used by forward selection. Each candidate is evaluated relative to the features already included. Our empirical analogue ranks candidates by their contribution to cross-validated Ridge prediction (Appendices C.3 and D.1). These scores also reflect regularization and finite-sample estimation. General affine coordinate changes need not preserve a fixed Ridge penalty, and adding features can reduce held-out performance. We therefore select features within training partitions and assess their usefulness on held-out trials.

F Extended Related Work

This appendix expands Section 5 and points to the closest analyses in this paper.

Rule ensembles. RuleFit, the closest classical counterpart of our procedure, grows an ensemble of decision trees on the response, converts every non-root node into a conjunctive binary rule, and

fits a lasso-penalized linear model on these rules together with linear input terms (Friedman & Popescu, 2008). In both methods, conjunctions read from shallow trees define binary features of a regularized linear predictor that retains the original inputs, and both rely on the principle expressed by equation 14: a nonlinear function of observed inputs adds linear predictive value through its component outside the existing affine span. The procedures differ in four respects. (i) Our trees target the live SAE atoms of a response-predictive population encoder rather than the response, with one class-balanced depth-two tree per atom over a vocabulary expanded by LLM-proposed features. (ii) Each tree yields one rule, the OR of its positive root-to-leaf paths (equation 6), rather than one rule per node. (iii) Rule-defined features are added one at a time by their conditional contribution under session-specific Ridge readouts, rather than weighted jointly along a lasso path. (iv) The predicted responses are 575 neurons recorded across 22 sessions, with different neurons in each, rather than a single response. With one tree per live atom, the candidate set is compact by construction, and forward selection retained 91–100% of candidate features in the catalog, nested, and transfer analyses. In the nested benchmark, all candidate features together gave $R^2 = 0.1260$, versus 0.1261 after selection (Table S6).

The analysis closest to RuleFit is the direct neural-PC control (Appendix C.6), which, like RuleFit, fits trees directly to response targets: one depth-two regression tree per session-specific principal component of the neural responses, using the 449 base and input-informed search features and the SAE-guided route's candidate budgets and selection procedure. It reached a held-out linear R^2 of 0.1234, versus 0.1261 for the SAE-guided route, 0.1166 for matched random rules, and 0.1100 for the baseline (Tables S6 and S7). With the SAE-guided route's atom-informed proposals, these session-local PC-target trees reached 0.1256, close to the 0.1261 of atom targets pooled across sessions, with gains over the direct route in all 22 sessions; most of the SAE-guided route's advantage over the direct route thus came with the proposals.

Rule extraction and distillation. Rule extraction approximates a trained network with rules or trees, assessed by accuracy, fidelity, consistency across training runs, and comprehensibility (Andrews et al., 1995). Pedagogical methods such as TREPAN query the network as an oracle to grow a decision tree (Craven & Shavlik, 1996), whereas decompositional methods such as DeepRED fit trees to hidden units layer by layer and merge them into input-level rules (Zilke et al., 2016). Our construction fits trees to internal components (SAE atoms), as decompositional methods do, expresses each rule directly over inputs and LLM-proposed features, as pedagogical methods do, and selects the resulting features by neural prediction, with rule–atom MCC as a correspondence diagnostic (Appendix B.3). Our counterpart to the consistency criterion keeps the rules fixed and matches them to atoms of 20 independently trained SAEs: 63 of 164 catalog rules reached best-match MCC ≥ 0.40 in all 20, versus a mean of 6 (range 4–8) for matched random rules (Appendix C.2).

Encoding models with richer readouts. Generalized linear models of spiking pass the summed outputs of stimulus filters, often represented in a smooth temporal basis, and of spike-history and coupling filters through a monotone nonlinearity (Pillow et al., 2008), matching the affine-plus-monotone readout of equation 1; our Ridge predictors are identity-link models of this kind, with five 0.1-s lags per sensory channel forming a delta temporal basis that spans any smooth basis over the same lags (Appendix A). Additive models use smooth univariate (Hastie & Tibshirani, 1986) or neural (Agarwal et al., 2021) shape functions, GA2M and explainable boosting machines add a few pairwise shape functions (Lou et al., 2013; Nori et al., 2019), and the hierarchical interaction lasso selects sparse pairwise products (Bien et al., 2013). With a fixed spline or bin basis, such a model is an affine readout of a basis-expanded description, to which the compatibility criterion applies directly. Each depth-two rule, a Boolean combination of thresholded inputs learned from atom activation, is a sum of at most two pairwise terms sharing its root split feature, the interaction form of GA2M; like fixed-basis expansions, the resulting features are shared across neurons and sessions, and after readout recalibration, 156 fixed features raised mean R^2 in seven excluded sessions from 0.1024 to 0.1190 (Section 4.3).

Encoding models in the whisker system. On recordings that include the sessions we reanalyze, whisk-cycle GLMs showed nearly all neurons modulated by contact, whisking, or task variables, with contact responses depending mainly on the identity of the contacting whisker (Rodgers et al., 2021). In superficial barrel cortex, volumetric calcium imaging during single-whisker object localization, analyzed with an encoding model of behavioral variables, revealed sparse, spatially in-

termingled representations of touch and whisker movement (Peron et al., 2015), and for primary whisker afferents, GLMs predicted spiking poorly from whisker angle but well from rotational force (Campagner et al., 2016). With the neurons and readout class fixed, the choice of sensory variable thus changed predictive adequacy, the dependence that neural compatibility formalizes; our search instead constructs and selects recodings of one fixed set of 14 lagged licking and whisker channels.

LLM-guided program search. FunSearch evolves programs with a pretrained LLM and a systematic evaluator, feeding high-scoring programs back into its prompts (Romera-Paredes et al., 2024); LLM-SR applies this approach to equation discovery (Shojaee et al., 2025), and CogFunSearch evolves symbolic cognitive models that outperformed human-designed ones in humans, rats, and fruit flies (Castro et al., 2025). LeaPR combines LLM-written feature functions with decision trees, including a variant that requests new features while splitting tree nodes (Poesia & Sampaio, 2025), much as our proposals draw on a baseline tree's out-of-fold errors (Appendix B.1). As in these methods, an LLM proposes executable features and data-driven fitting decides what is kept: proposals use 14 predefined operations on sensory-history inputs (Appendix B.2), trees fix thresholds and logic, and rule-defined features are selected by their conditional contribution to neural prediction. In the full-data catalog, the proposals raised mean rule–atom MCC on the seven rule-evaluation sessions from 0.3336 to 0.3470 (Appendix B.6); in nested discovery, fold-specific proposals raised the R^2 of PC-target trees from 0.1234 to 0.1256, versus 0.1239 with an equal number of random features from the same grammar (Appendix C.6 and Table S8).

Scoring explanations of model units. Explanations of model units are scored by LLM simulation of activations (Bills et al., 2023), by classification of activating versus non-activating contexts for SAE latents (Paulo et al., 2025), by the module's response to generated text related versus unrelated to the explanation, as in SASC (Singh et al., 2023), or by the overlap of concepts or logical formulas with binarized activation (Bau et al., 2017; Mu & Andreas, 2020), the form closest to our OR-of-AND rules. Executable rules are their own simulators: rule–atom MCC is the Pearson correlation between binary rule outputs and binarized atom activation over all samples of the seven sessions excluded from rule fitting, the binary case of correlation scoring; Oikarinen et al. (2025) show that correlation computed on uniformly sampled inputs passes their missing- and extra-label sanity checks. Because explanations with high simulation scores can still fail observational and interventional tests (Huang et al., 2023), we score the executable rule and check the LLM's prose against it and stored evidence (Appendix B.4). Model-input occlusion, an intervention on the encoder's inputs that parallels SASC's generated probes, and trial-block permutation further characterize each rule's source atom (Appendix B.3).