# From Circuit Evidence to Mechanistic Theory: An Inductive Logic Approach

**Anonymous ACL submission**

## Abstract

Mechanistic interpretability produces circuit-level causal analyses of neural network behaviour, but discovered circuits often remain isolated experimental artefacts: there is no shared formal representation for what circuits compute, how they relate, or when two findings provide evidence for the same mechanism. This work provides a formal infrastructure for cumulative mechanistic science by treating circuit interpretation as *inductive theory construction*. Each circuit is characterised at two levels: a *Causal Functional Signature* (CFS), which grounds component behaviour in causal attribution evidence and token role profiles, and an *architectural signature* $\tau_{\text{arch}}$, learned by inductive logic programming (ILP) from scale-invariant structural predicates. Together, these constitute a formal coherence layer that makes mechanistic claims explicit, comparable via $\theta$-subsumption, and portable across model scales. CFS reveals qualitatively distinct computational strategies across task types, including attention-mediated copying versus MLP-mediated binding. ILP signatures achieve substantially better structural separation than graph kernel and feature-vector baselines, and support principled transfer across model scales and architecture families.[1]

## 1 Introduction

Mechanistic interpretability (MI) aims to explain large language models (LLMs) by reverse-engineering their internal computation into human-understandable mechanisms (Ferrando et al., 2024). As MI scales across models, training regimes, and tasks a fundamental scientific challenge emerges: *How can we transform mechanistic discoveries into cumulative knowledge that coheres across experiments, models and tasks?* (Naveed et al., 2025; Zamfirescu-Pereira et al., 2025; Ferrando et al., 2024). A core focus of MI is circuit analysis, which aims to identify *circuits*: sparse, causally-relevant subgraphs whose interventions reliably alter model behaviour (Nanda et al., 2023; Meng et al., 2022; Aljaafari et al., 2025b). However, as more circuits are discovered, the field faces a growing problem of *experimental fragmentation*: results arise from different datasets, model variants, and evaluation protocols, and are often summarised with informal labels (e.g., *induction head*). Merullo et al. (2024) demonstrate that circuit component reuse occurs across tasks, but without a formal representation language, such reuse can only be observed post hoc and cannot be predicted or transferred systematically.

**Problem: coherence, going beyond discovery.** Existing discovery pipelines, activation patching (Meng et al., 2022), and related attribution-based approaches (Hanna et al., 2024), can identify causally relevant subgraphs, but they do not produce representations of *what was found* that support systematic comparison and accumulation. Across experimental settings, we lack:

- **A semantic language** to state what a circuit computes beyond model-specific descriptions;
- **Equivalence and refinement criteria** for deciding when new evidence confirms, specialises, or contradicts an existing mechanistic claim;
- **Transfer principles** that reuse prior mechanistic hypotheses to guide new interventions across model scales.

As a result, the central bottleneck is formal coherence: representing mechanistic claims so they can be compared, refined, and reused across settings.

**Approach.** We reframe circuit interpretation as **inductive theory construction** and introduce a

[1] Code and supplementary materials are available at [anonymisedforreview].

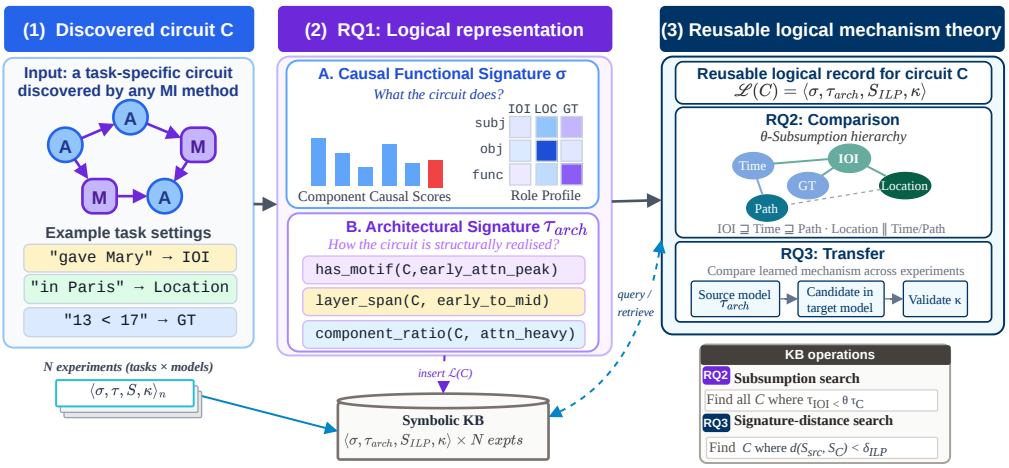


Figure 1: **Overview for inductive circuit theory construction.** A discovered circuit $C$ receives two complementary descriptions: a *Causal Functional Signature* $\sigma$ capturing what the circuit computes via causal attribution evidence, and an *architectural signature* $\tau_{\text{arch}}$ capturing how it is structurally realised as a scale-invariant Horn clause learned by $\text{ILP}_{\text{arch}}$. They form a reusable logical record $\mathcal{L}(C) = \langle \sigma, \tau_{\text{arch}}, \mathcal{S}_{\text{ILP}}, \kappa \rangle$ enabling formal comparison via $\theta$-subsumption (RQ2) and transfer to new models (RQ3).

formal coherence layer that makes MI results explicit, comparable, and reusable. From experimental traces and circuit structure, we learn *logical mechanism theories* that are testable, comparable, and refinable under new evidence. Each circuit receives two coupled descriptions:

- **Causal Functional Signature** (*what* the circuit computes), derived from causal attribution evidence over token positions labelled at two levels: shared linguistic roles enabling cross-task comparison, and task-specific roles capturing within-task functional structure;
- **Architectural signature** $\tau_{\text{arch}}$ (*how* the mechanism is structurally realised), learned by ILP from circuit graph patterns and model-size-invariant structural features.

The what/how distinction is load-bearing: CFS grounds claims in causal behaviour, while $\tau_{\text{arch}}$ provides portability across model scales within an architectural family. Logical representations support this by separating intended computation from incidental details, enabling structural comparison via relations such as $\theta$-subsumption, and allowing learned predicates to serve as building blocks in new hypotheses. ILP serves as the inductive engine, integrating structural motifs, attribution scores, and behavioural traces into coherent theories (Muggleton, 1991). These capabilities motivate three research questions:

- **RQ1 (Representation):** To what extent can logical representations jointly capture causal behavioural evidence and structural circuit patterns in a form portable across experiments?
- **RQ2 (Comparison):** How well do learned logical mechanism theories support formal comparison and refinement across MI experiments?
- **RQ3 (Transfer):** To what extent does prior mechanistic knowledge, captured as logical architectural signatures grounded in causal behaviour, support transfer to new models, and what is lost in the abstraction?

This work makes four contributions: **(i) Coherence-first formulation:** we frame circuit analysis as an inductive science of mechanisms, where the bottleneck is cross-experiment comparability and knowledge accumulation; **(ii) Causal Functional Signature (CFS):** a two-level causal characterisation of circuit behaviour using linguistic roles for cross-task comparison and task-specific roles for within-task structure; **(iii) ILP architectural signatures:** explicit logical mechanism descriptions that support testing, $\theta$-subsumption-based comparison, and cross-scale transfer; and **(iv) Evaluation:** 15 circuits (5 tasks $\times$3 models), scaled to 30 circuits per model across 10 tasks and three prompt splits, with ILP signatures achieving a 3$-$4$\times$ structural separation advantage over Weisfeiler-Lehman (WL) kernel (Shervashidze et al., 2011) and random-forest (Breiman, 2001) baselines.

## 2 Related work

**Circuit discovery and circuit-level mechanisms.** MI studies model behaviour, including identifying circuits: subgraphs or feature pathways that causally support a task. Foundational work on transformer circuits and induction heads established circuit-level analysis as a way to relate internal components to algorithmic behaviour (Elhage et al., 2021; Olsson et al., 2022). Subsequent studies have identified mechanisms for indirect object identification (Wang et al., 2023), arithmetic reasoning (Stolfo et al., 2023), semantic roles (Aljaafari et al., 2025b), and other task-specific behaviours. Activation patching (Meng et al., 2022), path patching (Goldowsky-Dill et al., 2023), automated circuit discovery (Conmy et al., 2023), and EAP–IG (Hanna et al., 2024) provide complementary methods for identifying causally relevant components or edges. Work on circuit component reuse further suggests that mechanisms may generalise across tasks (Merullo et al., 2024). Our work is orthogonal to discovery: given a discovered circuit, we provide a logical specification layer for representing its causal-functional evidence, learning its architectural signature, and comparing it with other mechanism hypotheses.

**Feature-level circuits and reusable structure.** Sparse autoencoders decompose activations into more interpretable features (Huben et al., 2024); sparse feature circuits and circuit tracing extend this view by identifying causal feature-level subgraphs and attribution graphs (Marks et al., 2025; Ameisen et al., 2025). These approaches refine the granularity of circuit nodes. Our representation is agnostic to this granularity: the same logical layer can represent circuits over attention heads, MLP blocks, or finer-grained feature nodes, provided that graph structure and causal-functional evidence are available.

**Abstraction, formality, and global perspectives.** Causal abstraction formalises interpretability as the search for high-level causal models that preserve low-level intervention behaviour (Geiger et al., 2025), while causal scrubbing provides an operational framework for testing mechanistic hypotheses through intervention-based resampling (Chan et al., 2022). Recent formal MI work applies neural-network verification to circuit discovery, seeking guarantees for robustness, patching, and minimality (Hadad et al., 2026). At a broader level, He et al. (2025) propose global-level MI through reusable modular circuits, and cognitive studies suggest that LLMs may develop language-independent abstractions beyond surface cues (Chen et al., 2026). These works share our motivation to move beyond isolated circuit findings. We differ by formalising the representation layer itself: mechanism hypotheses are stored as typed logical objects that support comparison, refinement, and transfer.

**Logic and neuro-symbolic interpretability.** Several mechanistic studies analyse how language models implement logical or formal reasoning, including propositional logic circuits (Hong et al., 2026) and content-independent syllogistic reasoning circuits (Kim et al., 2025). Related work also compares circuits for formal and functional linguistic abilities (Hanna et al., 2026). Palumbo et al. (2025) formalise mechanistic interpretations axiomatically, characterising when mechanistic descriptions approximately preserve network semantics. Our use of logic is different: rather than studying circuits that compute logical operations, we use inductive logic programming (Muggleton, 1991; Cropper and Dumančić, 2022) to construct architectural signatures over discovered circuits, making mechanism claims explicit, comparable, refinable, and transferable.

## 3 Methodology

We formalise circuit interpretation as inductive theory construction (Fig. 1). We introduce the core representation and learning components below, with additional ILP and architectural-signature details in Appendices A and B.

### 3.1 Problem Formulation

Let $M$ denote a transformer with computational graph $G = (V, E)$, and let $C = (V_C, E_C)$ be a sparse subgraph identified by any circuit discovery method. Given $C$, a task-specific evaluation suite $\mathcal{D}_T$, and background knowledge $\mathcal{B}$, we associate each circuit with a logical identity

$$\mathcal{L}(C) = \langle \sigma,\ \tau_{\text{arch}},\ \mathcal{S}_{\text{ILP}},\ \kappa \rangle,$$

where $\sigma$ is the CFS, $\tau_{\text{arch}}$ is the learned architectural signature, $\mathcal{S}_{\text{ILP}}$ summarises the learned theory, and $\kappa$ records validation statistics. All components are derived automatically from the stored circuit representation.

### 3.2 Formal Circuit Representation

Each circuit $C{=}(V_C, E_C)$ is represented as a set of typed ground facts organised into four layers. Layer 0 records provenance, such as task, model, and discovery method. Layer 1 records graph structure through node, type, layer, edge, component-ratio, relative-size, layer-span, and named-motif predicates, e.g., `has_motif`$(C, m)$ and normalised faithfulness $\mathrm{Faith}(C){=}(\mathrm{Acc}_T(C){-}\mathrm{Acc}_T^{\mathrm{abl}})/(\mathrm{Acc}_T(M){-}\mathrm{Acc}_T^{\mathrm{abl}})$ (Mueller et al., 2025). Layer 2 stores the CFS $\sigma$, and Layer 3 stores the learned architectural signature $\tau_{\mathrm{arch}}$. This representation exposes circuit evidence to $\mathrm{ILP}_{\mathrm{arch}}$ as Prolog-style facts while keeping storage and extraction details separate (detailed extraction procedure is in App. C).

### 3.3 Causal Functional Signature (CFS)

The CFS grounds each circuit component in causal behavioural evidence through two quantities. First, we compute the Direct Logit Attribution (DLA) score (nostalgebraist, 2020) $\delta_v{\in}\mathbb{R}$ for each node $v{\in}V_C$, measuring its additive contribution to the correct-answer logit. Components with $|\delta_v| < \epsilon_{\mathrm{DLA}}$ are flagged as causally marginal. Second, for each causally relevant component, we compute an *attribution-weighted attention profile* $\pi_v$: the attention distribution over input positions, weighted by $|\delta_v|$, with positions labelled by linguistic roles (subject, object, verb, function, entity, other) and task-specific roles (e.g., scaffold, repeated_name; see App. D). The CFS is the collection $\sigma{=}\{\langle v, \delta_v, \pi_v^{\mathrm{ling}}, \pi_v^{\mathrm{task}}\rangle : v{\in}V_C\}$, stored as Layer 2 facts that enrich the background knowledge for $\mathrm{ILP}_{\mathrm{arch}}$.

### 3.4 Learning Architectural Signatures

In ILP, learning is formulated as rule induction from examples and background knowledge. In our setting, $\mathrm{ILP}_{\mathrm{arch}}$ learns an *architectural signature* $\tau_{\mathrm{arch}}$ for a circuit family. Positive examples $E^+$ are circuits that implement the target mechanism (e.g., IOI or Location binding) and negative examples $E^-$ are circuits implementing other mechanisms. The background knowledge $\mathcal{B}_{\mathrm{arch}}$ contains logical facts describing each circuit's structure. ILP then searches for a rule that covers the positive circuits while excluding the negatives. The learned rule is a Horn clause, e.g., `arch_task(C) :- pred_1(C), pred_2(C,X), ...`, which reads as: circuit $C$ has architectural type `task` if the predicates in the body hold, making the learned signature an explicit structural hypothesis.

The predicates in $\mathcal{B}_{\mathrm{arch}}$ are chosen to be scale-invariant such that signatures can be compared across models of different scales. They include normalised depth $\rho(\ell){=}\ell/L$, component composition (`component_ratio`, `rel_size`), named structural motifs (`has_motif`), and CFS facts. For example, a semantic-role circuit may be represented by facts stating that it contains an early attention peak, has a high MLP ratio, spans a particular layer range, and contains components whose CFS profiles attend to role-bearing tokens. $\mathrm{ILP}_{\mathrm{arch}}$ uses these facts to induce a compact clause that distinguishes this circuit family from alternatives. Motifs are ranked before clause search by an attribution-weighted information gain score:

$$\mathrm{ATTR_IG}(m){=}\mathrm{IG}(m) \times (1 + \bar{s}_m),$$

where $\bar{s}_m$ is the mean edge attribution score of motif $m$ in the positive circuits. This favours motifs that are both discriminative and causally salient. We then search two complementary hypothesis spaces. Pool A contains clauses built from named motif predicates, such as `has_motif(C, early_attn_peak)`, and producing directly more interpretable signatures. Pool B contains clauses built from blind structural predicates, such as edge-type patterns, layer-span predicates, and component-composition predicates. Pool B acts as a check that the learned signature is not limited to the hand-defined motif names. Clause selection is based on one-vs-rest F1 over the positive and negative circuit sets: among candidate clauses from both pools, $\mathrm{ILP}_{\mathrm{arch}}$ selects the highest-scoring clause, with shorter clauses preferred under ties. The output is the learned Horn clause $\tau_{\mathrm{arch}}$ with an ILP signature

$$\mathcal{S}_{\mathrm{ILP}}(\tau_{\mathrm{arch}}) = \langle n_{\mathrm{clauses}}, \bar{d}, \mathcal{P}_H, \bar{\ell}, \texttt{complexity}\rangle,$$

which summarises the number of clauses, derivation depth, predicate vocabulary, clause length, and composite complexity score. The clause $\tau_{\mathrm{arch}}$ serves as the interpretable mechanistic hypothesis: a logical claim that can be read, challenged, and refined. The summary $\mathcal{S}_{\mathrm{ILP}}$ serves as its compact computational handle, enabling distance-based comparison between signatures and candidate matching during transfer. Full definitions of motif extraction, motif scoring, dual-pool search, and ILP signature construction are provided in App. B.

### 3.5 Validation Criteria

A circuit $C$ is retained as a valid mechanism candidate only if it satisfies behavioural consistency and causal relevance. Behavioural consistency requires the circuit to support the target behaviour on the evaluation suite, $\mathrm{Acc}_T(C) \geq \theta_{\mathrm{behav}}$. Causal relevance measures the task-performance degradation caused by ablating the circuit:

$$\Delta_T(C) = \mathrm{Acc}_T(M) - \mathrm{Acc}_T(M^{\mathrm{ablate},C}).$$

We require $\Delta_T(C) \geq \theta_{\mathrm{causal}}$ and check approximate $\epsilon$-minimality by greedy edge ablation. Threshold values, random-subgraph significance tests, and sensitivity analyses are found in App. G.

### 3.6 Comparison via $\theta$-Subsumption

Learned clauses support cross-experiment comparison via $\theta$-subsumption (Gottlob, 1987). A clause $\gamma_1$ subsumes a clause $\gamma_2$, written $\gamma_1 \preceq_\theta \gamma_2$, when some substitution $\theta$ makes every literal in $\gamma_1\theta$ appear in $\gamma_2$. This induces an ordering over learned signatures: one signature may generalise or specialise another, two signatures may be equivalent under mutual subsumption, or they may be incomparable. In the relaxed comparisons, numeric thresholds are abstracted away, so the ordering captures predicate-level structure rather than exact threshold agreement. We use this relation to identify when one mechanism theory makes stronger structural commitments than another; strict and relaxed definitions are given in App. B.7.

### 3.7 Transfer of Mechanistic Knowledge

Given a source mechanism record $\mathcal{L}(C_\alpha)$ in model $M_\alpha$ and a target model $M_\beta$, transfer treats the source architectural signature as a retrieval hypothesis. The procedure has four stages: (i) retrieve $\tau_{\mathrm{arch},\alpha}$ and $\sigma_\alpha$ from the registry; (ii) identify candidate circuits in $M_\beta$ whose scale-normalised structural predicates and ILP signature distance match the source signature within tolerance; (iii) screen candidates using behavioural and causal validation on the target evaluation suite; and (iv) select the best validated candidate. If no candidate passes validation, we fall back to full circuit discovery in the target model and compare the re-learned signature against the source signature. Thus, transfer is not treated as proof of mechanism identity from structure alone; it is a proposal-and-validation procedure in which logical abstraction retrieves candidates and empirical tests determine whether the transfer is supported. The full transfer algorithm, including tolerance values and fallback criteria, is given in App. I.

### 3.8 Experimental Setup

We evaluate on three task families covering distinct operation types: semantic role binding ($\kappa$=`binding`), indirect object identification ($\kappa$=`selection`), and numerical comparison ($\kappa$=`comparison`). Circuits are extracted using EAP-IG (Hanna et al., 2024) with $k$=200 edges retained per circuit; the framework accepts any discovery method producing attributed subgraphs (Section 3.2). The primary evaluation covers 15 circuits (5 tasks $\times$ 3 models): semantic roles Location, Time, and Path (Aljaafari et al., 2025b), IOI (Wang et al., 2023), and Greater-Than, across Pythia (14M and 1B) (Biderman et al., 2023), and LLaMA-3.2-1B (Grattafiori et al., 2024). The scaled evaluation expands to 10 task types (8 semantic roles, IOI, Greater-Than) with 3 circuits per task discovered from disjoint prompt subsets, yielding 30 circuits per model. Task selection rationale, model cards, and implementation details are in Appendix E and G. Full circuit inventory (node counts and split) is in Appendix E.5.

## 4 Results

We report the main findings for each RQ below; supporting tables and extended results are provided in App. H.

### 4.1 CFS Reveals Distinct Computational Strategies

**Role-consistent attention across scales.** CFS profiles (Tab. 1) show that dominant attention roles are determined more by task type than scale. Location and Path circuits are `object`-dominant at every scale (Location: 76%, 74%, 71%; Path: 71%, 55%, 68% across Pythia-14M, Pythia-1B, LLaMA), consistent with circuits attending to the grammatical object position hosting the role filler. Time circuits are `function`-dominant throughout (81%, 87%, 100%[§]), reflecting attention to temporal prepositions and determiners. GT circuits are `entity`-dominant at 14M (77%) but shift toward `function`-dominant at larger scales (48%$^\star$, 58%). IOI shifts from `object`-dominant at 14M (50%) to `function`-dominant at 1B and LLaMA (58%, 67%), suggesting that the copy mechanism relocates attention from name positions to

| Model | Task | Caus. | Marg. | Dom. Role | Faith. |
|---|---|---|---|---|---|
| Py-14M | IOI | 26 | 0 | object (50%) | 0.79 |
| | Location | 21 | 1 | object (76%) | 0.57 |
| | Time | 21 | 0 | function (81%) | 1.22 |
| | Path | 12 | 12 | object (71%) | 1.21 |
| | GT | 28 | 0 | entity (77%) | 0.88 |
| Py-1B | IOI | 34 | 2 | function (58%) | 0.14 |
| | Location | 29 | 3 | object (74%) | 0.76 |
| | Time | 28 | 2 | function (87%) | $-7.23^{\dagger}$ |
| | Path | 31 | 5 | object (55%) | 1.08 |
| | GT | 40 | 0 | function$^{\star}$ (48%) | 0.85 |
| LL-1B | IOI | 0 | 29 | function (67%)$^{\ddagger}$ | 0.01 |
| | Location | 8 | 23 | object (71%) | 0.32 |
| | Time | 5 | 21 | function (100%)$^{\S}$ | 0.88 |
| | Path | 10 | 18 | object (68%) | 0.18 |
| | GT | 11 | 18 | function (58%) | 0.05 |

Table 1: **CFS summary across models.** Dom. Role = most common $\arg\max \pi_v$ over linguistic roles. † Near-marginal circuit ($\Delta_T = 0.003$). ‡ All LLaMA-IOI components are causally marginal despite $\Delta_T = 5.24$. ⋆ Tie between `function` and `entity`. § All active components attend to function words.

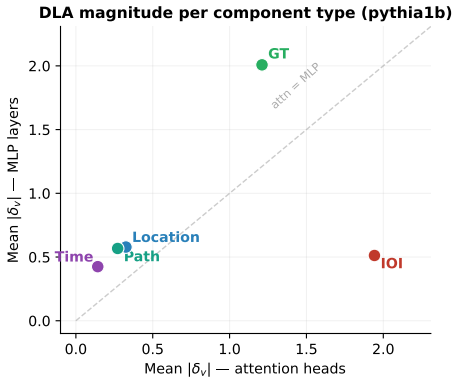


Figure 2: **DLA magnitude per component type (Pythia-1B).** Mean $|\delta_v|$ for attention heads (x-axis) vs. MLP layers (y-axis). IOI falls *below* the diagonal (attn-dominant, $|\delta_v|_{\text{attn}} \approx 2.0$); Location, Time, and Path fall *above* (MLP-dominant). GT is MLP-dominant with high absolute magnitude ($|\delta_v|_{\text{MLP}} \approx 2.0$), consistent with MLP-mediated numerical comparison.

structural tokens as capacity increases. CFS feature vectors cluster significantly by task family on Pythia-1B ($p$=0.008, permutation test). LLaMA supports task-type consistency across the evaluated architectural families. The LLaMA IOI circuit has all 29 components causally marginal ($|\delta_v|$<0.05) despite high circuit-level causal relevance ($\Delta_T$=5.24); its full-component profile is nonetheless `function`-dominant (67%), matching the Pythia-1B pattern and suggesting a distributed implementation of the same functional strategy.

**Distinct wiring: attention copying vs. MLP binding.** The CFS differences reflect qualitatively different circuit wiring (App. H.3). In Pythia-1B, IOI routes information through 85 attn→attn edges, forming three-hop attention chains (`attn_chain_3`) characteristic of the copy mechanism. Semantic role circuits rely instead on 64–96 mlp→mlp edges, with motifs such as `mlp_heavy` reflecting MLP-dominated information flow; LLaMA preserves this pattern (44 vs. 0–2 attn→attn edges). Fig. 2 provides a complementary attribution-level view: IOI falls *below* the diagonal (attn-dominant), while Location, Time, and Path fall *above* (MLP-dominant). GT also falls above the diagonal but with high absolute MLP magnitude, consistent with MLP-mediated numerical comparison rather than distributed binding. Full linguistic role profiles are in App. D.

### 4.2 ILP Captures Structural Boundaries

**Learned clauses encode the binding/selection distinction.** $\text{ILP}_{\text{arch}}$ learns one Horn clause per task via one-vs-rest classification (Tab. 2; full

| Task | Predicates in $\tau_{\text{arch}}$ | $\mathcal{C}$ |
|---|---|---|
| IOI | `comp_ratio(C, attn, R)` `R > 0.63` | 0.2 |
| LOC | `has_motif(C, mlp_heavy)` | 0.9 |
| | `has_motif(C, mid_layer_attn_peak)` | |
| | `layer_span(C, L, L')` `comp_ratio(C, attn, R)` | |
| | `R > 0.38` `size(C, N)` `N<38` | |

Table 2: **Example architectural signatures (Pythia-1B, 5-task pool).** IOI is structurally simple, while Location requires a conjunctive signature. Colour identifies predicate type: `named structural motif`; `layer position`; `circuit composition`.

results in App. B.8). In the 5-task pool, the IOI clause requires only a single predicate, high attention ratio, directly reflecting the attention-dominated wiring identified by CFS; in the expanded 10-task pool, IOI acquires additional predicates (`attn_chain_3`, `layer_span`, `size`) to maintain discrimination against structurally similar roles (full clauses in App. B.8). Role and GT clauses require conjunctions of motifs, component ratios, and size constraints in both pools, capturing the MLP-heavy composition that no single feature can express. Thus, the clause structure turns the mechanistic distinction into a testable structural hypothesis: circuits satisfying the IOI clause are retrieved as candidates for attention-mediated copying, and CFS with $\Delta_T$ determines whether that interpretation is empirically supported.

**ILP distance reveals task family structure.** Fig. 3 shows the pairwise ILP signature distance on Pythia-1B. IOI is distant from all other circuits ($d$=0.32–0.45), while semantic roles and GT form a tight cluster ($d$=0.05-0.13). GT clusters with

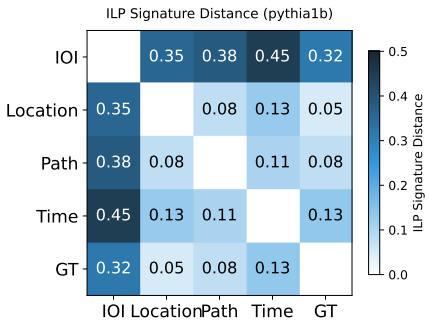


Figure 3: **ILP signature distance (Pythia-1B).** IOI is distant from all circuits ($d \geq 0.32$). Roles and GT cluster tightly ($d \leq 0.13$).

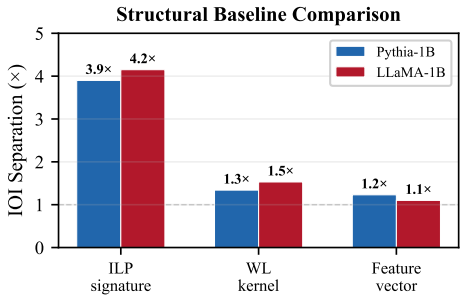


Figure 4: **IOI separation: ILP vs. structural baselines.** Grouped by model. ILP achieves 3.9–4.2×; WL kernel 1.3–1.5×.

roles because its clause uses the same predicate vocabulary despite implementing a different computation, suggesting that GT circuits are realised through structural patterns closer to binding than to attention-copying. The threshold $\delta_{\text{ILP}}$=0.30 cleanly separates IOI from all other circuits; the same pattern holds on LLaMA ($d_{\text{IOI}}$=0.34–0.47; see App. H.1 for all models). We compare ILP distance against three structural alternatives (Fig. 4). These are a Weisfeiler-Lehman (WL) graph kernel (Shervashidze et al., 2011) capturing local neighbourhood structure, a random forest (Breiman, 2001) trained on the same structural features used as ILP background knowledge, and Euclidean distance over the same feature vectors. ILP achieves 3.9× IOI separation on Pythia-1B (4.2× on LLaMA), compared to 1.3× for the WL kernel. The random forest has access to the same predicates, but does not return compact inspectable clauses over those predicates; its 60% leave-one-out accuracy (misclassifying GT and IOI) indicates that explicit logical structure provides useful discrimination beyond the feature inventory alone (details in App. H.4).

**Subsumption reveals a non-trivial refinement hierarchy.** Using the 5-task-pool clauses under relaxed $\theta$-subsumption, where numeric thresholds are abstracted away and predicate structure determines the ordering, a non-trivial hierarchy emerges (Tab. 3).

| | IOI | LOC | PATH | TIME | GT |
|---|---|---|---|---|---|
| **IOI** | $=$ | $\sqsupseteq$ | $\sqsupseteq$ | $\sqsupseteq$ | $\sqsupseteq$ |
| **LOC** | $\sqsubseteq$ | $=$ | $\parallel$ | $\parallel$ | $\parallel$ |
| **PATH** | $\sqsubseteq$ | $\parallel$ | $=$ | $\sqsubseteq$ | $\parallel$ |
| **TIME** | $\sqsubseteq$ | $\parallel$ | $\sqsupseteq$ | $=$ | $\parallel$ |
| **GT** | $\sqsubseteq$ | $\parallel$ | $\parallel$ | $\parallel$ | $=$ |

Table 3: **Relaxed $\theta$-subsumption (Pythia-1B).** $\sqsupseteq$=generalises; $\sqsubseteq$=specialises; $\parallel$=incomparable. IOI generalises all; Time generalises Path; Location is incomparable to Time/Path.

In this relaxed sense, the single-predicate IOI clause generalises the other clause, any circuit satisfying a role or GT clause also satisfies the attention-ratio constraint. Within the role family, Time generalises Path, which adds `attn_chain_3`, while Location is incomparable to both because it uses different motifs (`mlp_heavy`, `mid_layer_attn_peak`). The hierarchy also changes with scale: in Pythia-14M, Time's size-only clause generalises all others and Location $\equiv$ Path, reflecting greater structural homogeneity; in Pythia-1B, the richer hierarchy emerges because the larger model produces circuits diverse enough to support meaningful differentiation. This partial order formalises relationships that informal labels and distance metrics do not express: Location and Time are not simply close or far, but distinct structural specialisations of a broader binding template (App. B.7).

**Scaled evaluation and within-task consistency.** The expanded 10-task evaluation shows that $\text{ILP}_{\text{arch}}$ remains informative beyond the 5-task pool, but also exposes genuine within-task variability. Structurally stable tasks retain high confidence (IOI: 0.93, GT: 0.89), whereas prompt-sensitive semantic roles drop substantially (Location: 0.31, Beneficiary: 0.31). Extended distance matrices, within-task variance, and sensitivity analyses are reported in Appendices H.1–H.2.

### 4.3 ILP Signatures Support Transfer

**Scale-invariant predicates enable cross-scale and cross-family transfer.** Raw component overlap is near zero across all model pairs (node Jaccard 0.09–0.24 within Pythia; lower cross-family), thus transfer via node identity is not possible. Scale-invariant predicates

(`component_ratio`, `layer_span`, `has_motif`) abstract over these differences. On Pythia-1B each semantic role circuit finds 2–3 candidates within $\delta_{\text{ILP}} = 0.30$, while IOI finds none ($d = 0.32$–$0.45$). LLaMA shows the same separation pattern ($d_{\text{IOI}}$=0.34–0.47), confirming that the boundary between attention-copying and MLP-binding mechanisms is architecture-independent.

**Live transfer across scales and families.** End-to-end transfer accepts candidates for all 5 tasks in both directions: Pythia-14M→Pythia-1B (cross-scale, 70× gap) and LLaMA→Pythia-1B (cross-family). For IOI and GT, the best accepted candidate matches the same task in the target model, with high selectivity (2/37–41 for GT, 8/41 for IOI). For semantic roles, the best candidate is Goal, a role not in the source pool, because it has the highest $\Delta_T$ among binding circuits in Pythia-1B ($\Delta_T$=1.00). This is a semantically valid match: both implement binding via the same motif vocabulary. The accept/reject pattern is identical across both transfer directions despite very different source $\Delta_T$ values (e.g. IOI: $-0.91$ from Pythia-14M vs. $5.24$ from LLaMA), confirming that the structural signature, not the source behaviour, drives candidate selection (App. H.6).

**Binding mechanisms preserve causal contribution; selection and comparison do not.** Tab. 4 reports $\Delta_T$ across all three models. In this evaluation, causal preservation aligns more closely with mechanism type than with architectural similarity. Within the Pythia family, IOI undergoes a sign reversal from 14M to 1B ($-0.91$→$4.37$), indicating a qualitative change in the functional role of the circuit with scale. GT shows a large gap across pairs (3.97 within Pythia, 2.74 cross-family). Among binding mechanisms, Location and Path show strong cross-family preservation (|diff|=0.08 and 0.26 respectively), while Time does not (|diff|=0.85), likely because its near-zero $\Delta_T$ on Pythia-1B (0.003) makes this comparison unreliable. Where binding circuits have robust causal contributions, MLP-mediated binding appears more stable across architectures than attention-mediated copying or numerical comparison. To confirm that these causal contributions reflect structured architectural rather than chance, we evaluate 50 size-matched random circuits per task across both Pythia models (34 circuits on 14M, 35 on 1B, including multi-split variants). Random subgraphs consistently yield $\Delta_T \approx 0$, and real circuits are far better in 29/34 cases on Pythia-14M and 28/35 on Pythia-1B ($p$<0.05). See App. H.5 for per-circuit details.

| Task | Py-14M | Py-1B | LL-1B | LL↔Py-1B |
|---|---|---|---|---|
| IOI | −0.91 | 4.37 | 5.24 | 0.87 |
| Location | 1.93 | 1.28 | 1.20 | **0.08** |
| Time | 0.75 | 0.003 | −0.85 | 0.85 |
| Path | 0.53 | 0.21 | 0.48 | 0.26 |
| GT | 7.95 | 3.97 | 1.23 | 2.74 |

Table 4: $\Delta_T$ **across models and families.** Binding mechanisms (Location, Path) preserve causal contribution across architectures (|diff|=0.08–0.26); selection (IOI) undergoes a sign reversal across Pythia scales; comparison (GT) shows large gaps across all pairs.

## 5 Conclusion

Mechanistic interpretability produces circuit discoveries that are locally valid but difficult to accumulate: without a shared representation, it remains unclear how findings from different experiments, models, or tasks should be compared. We addressed this coherence problem by treating circuit interpretation as inductive theory construction. Causal Functional Signatures ground circuit components in causal attribution evidence, while ILP-learned architectural signatures capture structural regularities as inspectable clauses. Together, they form a representation layer in which mechanistic claims can be stated explicitly, compared via $\theta$-subsumption, and reused as transfer hypotheses. The results support three findings. First, CFS distinguishes computational strategies such as attention-mediated copying and MLP-mediated binding that are preserved across architectural families. Second, ILP signatures capture structural boundaries that raw component overlap and feature-vector baselines do not expose as clearly. Third, transfer succeeds most reliably when treated as candidate retrieval followed by behavioural and causal validation: binding mechanisms with robust causal contributions show stronger cross-architecture preservation than the selection and comparison mechanisms evaluated here. This work provides a formal infrastructure between circuit discovery and knowledge accumulation. Proxy tasks establish local validity; our method formalises generalisability, cross-experiment relationships, and the level of abstraction at which a claim holds. Extending the approach to more distributed mechanisms and broader architectural families remains the primary direction for future work.

## Limitations

Each $\tau_{\text{arch}}$ clause is falsifiable in principle, but we test only whether it predicts causal contribution across models, not task behaviour on novel prompts, the level at which our framework and proxy-task approaches are complementary rather than competing. The evaluation covers 30 circuits across 10 task types, two Pythia scales, and one cross-family model (LLaMA-3.2-1B); the binding/selection/comparison schema is not a universal theory of internal mechanisms, and extending to additional architectural families remains important future work. All thresholds are empirically chosen; sensitivity analysis confirms stability across the primary range (App. H.2). CFS role profiles depend on a lightweight dependency parser, and the greedy $\epsilon$-minimality procedure may miss non-contiguous minimal subcircuits.

## Ethical Considerations

This work advances mechanistic interpretability by formalising circuit findings as reusable logical identities, which supports transparency and cumulative safety research. However, the symbolic registry and transfer engine directly lower the cost of locating circuits in new models: a practitioner can transfer a validated circuit description to a target model without full rediscovery, which can be more actionable for targeted model editing than opaque attribution scores, introducing a dual-use risk. The tasks studied are analytically benign, but the framework is task-agnostic and its outputs could be applied to circuits governing more sensitive behaviours.

## References


Nura Aljaafari, Danilo Carvalho, and Andre Freitas. 2025a. TRACE: Training and inference-time interpretability analysis for language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 806–820, Suzhou, China. Association for Computational Linguistics.

Nura Aljaafari, Danilo S. Carvalho, and André Freitas. 2025b. Emergence and localisation of semantic role circuits in llms. *Preprint*, arXiv:2511.20910.

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, and 8 others. 2025. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and 1 others. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Leo Breiman. 2001. Random Forests. *Machine Learning*, 45(1):5–32.

Lawrence Chan, Adria Garriga-Alonso, Nicholas Goldowsky-Dill, Ryan Greenblatt, Jenny Nitishinskaya, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. 2022. Causal scrubbing, a method for rigorously testing interpretability hypotheses. *AI Alignment Forum*.

Yuxin Chen, Yiran Zhao, Yang Zhang, An Zhang, Kenji Kawaguchi, Shafiq Joty, Junnan Li, Tat-Seng Chua, Michael Qizhe Shieh, and Wenxuan Zhang. 2026. The emergence of abstract thought in large language models beyond any language. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Noam Chomsky. 1993. *Lectures on Government and Binding*. De Gruyter Mouton, Berlin, New York.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, volume 36, pages 16318–16352. Curran Associates, Inc.

Andrew Cropper and Sebastijan Dumančić. 2022. Inductive logic programming at 30: A new introduction. *Journal of Artificial Intelligence Research*, 74:765–850.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. 2024. A primer on the inner workings of transformer-based language models. *Preprint*, arXiv:2405.00208.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. 2025. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. 2023. Localizing model behavior with path patching. *arXiv preprint arXiv:2304.05969*.

Georg Gottlob. 1987. Subsumption and implication. *Information Processing Letters*, 24(2):109–111.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Itamar Hadad, Guy Katz, and Shahaf Bassan. 2026. Formal mechanistic interpretability: Automated circuit discovery with provable guarantees. In *The Fourteenth International Conference on Learning Representations (ICLR)*.

Michael Hanna, Yonatan Belinkov, and Sandro Pezzelle. 2026. Are formal and functional linguistic mechanisms dissociated in language models? *Computational Linguistics*, 52(1):43–83.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. 2024. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *ICML 2024 Workshop on Mechanistic Interpretability*.

Yinhan He, Wendy Zheng, Yushun Dong, Yaochen Zhu, Chen Chen, and Jundong Li. 2025. Towards global-level mechanistic interpretability: A perspective of modular circuits of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 22865–22880. PMLR.

Guan Zhe Hong, Nishanth Dikkala, Enming Luo, Cyrus Rashtchian, Xin Wang, and Rina Panigrahy. 2026. A implies b: Circuit analysis in llms for propositional logical reasoning. *Advances in Neural Information Processing Systems*, 38:30749–30803.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations (ICLR)*, volume 2024, pages 7827–7845.

Geonhee Kim, Marco Valentino, and Andre Freitas. 2025. Reasoning circuits in language models: A mechanistic interpretation of syllogistic inference. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10074–10095, Vienna, Austria. Association for Computational Linguistics.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2025. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2024. Circuit component reuse across tasks in transformer language models. In *The Twelfth International Conference on Learning Representations*.

Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fried Fiotto-Kaufman, Tal Haklay, Michael Hanna, Jing Huang, Rohan Gupta, Yaniv Nikankin, Hadas Orgad, Nikhil Prakash, Anja Reusch, Aruna Sankaranarayanan, Shun Shao, Alessandro Stolfo, and 4 others. 2025. MIB: A mechanistic interpretability benchmark. In *Forty-second International Conference on Machine Learning*.

Stephen Muggleton. 1991. Inductive logic programming. *New generation computing*, 8(4):295–318.

Neel Nanda and Joseph Bloom. 2022. Transformerlens. `https://github.com/TransformerLensOrg/TransformerLens`.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2025. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72.

nostalgebraist. 2020. Interpreting gpt: The logit lens. LessWrong.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. In-context learning and induction heads. *Transformer Circuits Thread*.

Nils Palumbo, Ravi Mangal, Zifan Wang, Saranya Vijayakumar, Corina S. Pasareanu, and Somesh Jha. 2025. Validating mechanistic interpretations: An axiomatic approach. In *Forty-second International Conference on Machine Learning*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Gordon D Plotkin. 1970. A note on inductive generalization. *Machine intelligence*, 5(1):153–163.

Stuart Russell and Peter Norvig. 2020. *Artificial Intelligence: A Modern Approach*, 4 edition. Pearson.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. 2011. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(77):2539–2561.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. 2023. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7035–7052, Singapore. Association for Computational Linguistics.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

J.D. Zamfirescu-Pereira, Eunice Jun, Michael Terry, Qian Yang, and Bjoern Hartmann. 2025. Beyond code generation: Llm-supported exploration of the program design space. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA. Association for Computing Machinery.

# A Brief Primer on Inductive Logic Programming

Inductive logic programming (ILP) (Muggleton, 1991) is a subfield of machine learning that combines inductive inference with symbolic knowledge representation (Cropper and Dumančić, 2022). Unlike many machine learning algorithms that produce numeric parameters (weights, probabilities), such as probes (linear classifiers over representations), graph similarity kernels (pairwise distance scores), ILP produces hypotheses expressed in first-order logic (FOL): explicit rules that can be read, inspected, compared, and transferred independently of the model that produced them.

## A.1 Positive Examples, Negative Examples, and Background Knowledge

An ILP problem consists of positive examples $E^+$, negative examples $E^-$, and background knowledge $\mathcal{B}$. The positive examples are cases that the target concept should cover; the negative examples are cases it should exclude; and the background knowledge contains known facts and relations about the domain. ILP searches for a hypothesis $\tau$ such that:

$$\mathcal{B} \cup \tau \models E^+ \quad \text{and} \quad \mathcal{B} \cup \tau \not\models E^-. \qquad (1)$$

In practice, this means that $\tau$ should explain the positive examples while avoiding the negative examples, subject to constraints such as clause length, allowed predicates, and search depth.

## A.2 Horn Clauses

The hypothesis $\tau$ takes the form of a *Horn clause*: a rule with a single conclusion (the *head*) and one or more conditions (the *body*): `head(X) :- condition_1(X), condition_2(X, Y), ...`. read as: "`head` holds of `X` if all conditions hold."

## A.3 Why ILP is Suitable for Circuit Signatures

One could argue that simpler methods are easier to implement and may give a similar level of empirical separation. We argue that we need *theory formation*: descriptions of circuit mechanisms that are explicit, formally comparable, and transferable across models. ILP satisfies these requirements by construction, providing a level of formality and interpretability that purely statistical or feature-based methods do not. In particular, ILP is useful for three reasons (Russell and Norvig, 2020):

- **Relational representation.** ILP can learn concepts that are difficult to express using fixed attribute vectors, because it represents relationships between entities rather than properties of individual ones;

- **Inspectable hypotheses.** ILP produces rules in a form that is directly readable: a Horn clause states explicitly *why* a concept holds, not *that* it holds. A human can examine the rule, challenge it, and refine it against new evidence;
- **Formal comparison and transfer.** As the hypotheses share the same logical language, they can be compared via $\theta$-subsumption to determine when one mechanism generalises another (Appendix B.7), and transferred to new models without retraining.

Consider learning the concept *grandparent* from family relations. The background knowledge $\mathcal{B}$ contains facts of the form `parent(X, Y)`. The examples are:

| **Positive** $E^+$ | **Negative** $E^-$ |
|---|---|
| `grandparent(sara, carol)` | `grandparent(sara, bob)` |
| `grandparent(bob, dave)` | `grandparent(carol, sara)` |

Given these examples and the background facts, ILP can induce the clause `grandparent(X, Z) :- parent(X, Y), parent(Y, Z)`. It illustrates all three properties above. First, the concept cannot be captured by isolated attributes (age, gender) because it depends on a *chain of relations* between entities, expressed here via the shared variable `Y`. Second, the rule is immediately inspectable: one can read it, verify it against new cases, and refine it by adding or removing body literals. Third, the rule can be compared against a more specific hypothesis: `grandparent(X, Z) :- parent(X, Y), parent(Y, Z), female(X)`. This variant covers only grandmothers. The original rule *generalises* it: every case covered by the second is also covered by the first, but not vice versa. This relation is formalised by $\theta$-subsumption (Appendix B.7).

### A.4 Mapping ILP to Mechanistic Interpretability

The previous example connects naturally to mechanistic interpretability. The entities are neural network components rather than people, the background knowledge $\mathcal{B}_{\text{arch}}$ contains relational facts describing how those components are connected, and the target concept is a mechanistic type such as "this subgraph implements location-role binding". Positive examples are subgraphs that implement the target mechanism; negative examples are subgraphs that implement something else, for instance indirect object identification or numerical comparison. ILP induces a rule, the architectural signature $\tau_{\text{arch}}$, that captures exactly what structural properties distinguish the positive examples from the negative ones, just as the grandparent rule captures what relational property distinguishes grandparents from non-grandparents.

In both ILP and circuit analysis, complex behaviour is explained by identifying structured, interpretable components and the relations between them. In ILP, a higher-level concept such as *grandparent* is built from simpler relational components, e.g., `parent(X, Y)` and `parent(Y, Z)`, linked through a shared variable. In circuit analysis, a mechanism is characterised by how its components are connected and what structural patterns they form. $\text{ILP}_{\text{arch}}$ utilises this parallel directly: rather than representing a circuit as a flat feature vector, it expresses the circuit's structural identity as a Horn clause over relational predicates, capturing dependencies that isolated attributes cannot. For example, $\text{ILP}_{\text{arch}}$ induces the following clauses for two semantic role circuits in Pythia-1B:

```
arch_time(C) :- has_motif(C, early_attn_peak),
                layer_span(C, L_min, L_max),
             component_ratio(C, attn, R), R > 0.35,
                  size(C, N), N < 36.

arch_path(C) :- has_motif(C, early_attn_peak),
                has_motif(C, attn_chain_3),
                layer_span(C, L_min, L_max),
             component_ratio(C, attn, R), R > 0.44,
                  size(C, N), N < 42.
```

The Path clause extends the Time clause by adding one predicate, `has_motif(C, attn_chain_3)`, making it structurally more specific. Under relaxed $\theta$-subsumption, Time therefore generalises Path: every circuit satisfying the Path clause also satisfies the Time clause, but not vice versa. The full refinement hierarchy is reported in Section 4.2. A full sample of learned clauses is given in Table 5.

## B Architectural Signatures

This appendix specifies how architectural signatures are constructed from circuit graphs.

### B.1 Scale-Invariant Encoding

Absolute layer indices are normalised as:

$$\rho(\ell) = \ell/L \in [0, 1],$$

| Task | Learned clause $\tau_{\text{arch}}$ (Pythia-1B, 10-task pool) | Conf. | Cmplx. |
|---|---|---|---|
| *High confidence — single discriminative motif* | | | |
| IOI | `arch_ioi(C) :-` `has_motif(C, attn_chain_3)`, `layer_span(C, $L$, $L'$)`, `comp_ratio(C, attn, R)`, R > 0.58, `size(C, N)`, N < 44. | 0.93 | 0.8 |
| GT | `arch_gt(C) :-` `has_motif(C, mid_layer_attn_peak)`, `layer_span(C, $L$, $L'$)`, `comp_ratio(C, attn, R)`, R > 0.50, `size(C, N)`, N < 49. | 0.89 | 0.8 |
| *Intermediate confidence — two-motif conjunctions* | | | |
| Source | `arch_source(C) :-` `has_motif(C, attn_chain_3)`, `has_motif(C, early_attn_peak)`, `layer_span(C, $L$, $L'$)`, `comp_ratio(C, attn, R)`, R > 0.40, `size(C, N)`, N < 42. | 0.81 | 0.9 |
| Time | `arch_time(C) :-` `has_motif(C, mlp_heavy)`, `has_motif(C, early_attn_peak)`, `layer_span(C, $L$, $L'$)`, `comp_ratio(C, attn, R)`, R > 0.37, `size(C, N)`, N < 37. | 0.81 | 0.9 |
| Instrument | `arch_instrument(C) :-` `has_motif(C, attn_to_attn)`, `has_motif(C, mlp_heavy)`, `layer_span(C, $L$, $L'$)`, `comp_ratio(C, attn, R)`, R > 0.38, `size(C, N)`, N < 39. | 0.52 | 0.9 |
| Path | `arch_path(C) :-` `has_motif(C, early_attn_peak)`, `has_motif(C, mlp_heavy)`, `layer_span(C, $L$, $L'$)`, `comp_ratio(C, attn, R)`, R > 0.38, `size(C, N)`, N < 39. | 0.52 | 0.9 |
| Topic | `arch_topic(C) :-` `has_motif(C, early_attn_peak)`, `has_motif(C, attn_to_attn)`, `layer_span(C, $L$, $L'$)`, `comp_ratio(C, attn, R)`, R > 0.44, `size(C, N)`, N < 44. | 0.44 | 0.9 |
| *Low confidence — structurally variable binding roles* | | | |
| Goal | `arch_goal(C) :-` `has_motif(C, early_attn_peak)`, `layer_span(C, $L$, $L'$)`, `comp_ratio(C, attn, R)`, R > 0.44, `size(C, N)`, N < 42. | 0.37 | 0.8 |
| Beneficiary | `arch_beneficiary(C) :-` `has_motif(C, late_attn_peak)`, `layer_span(C, $L$, $L'$)`, `comp_ratio(C, attn, R)`, R > 0.42, `size(C, N)`, N < 42. | 0.31 | 0.8 |
| Location | `arch_location(C) :-` `has_motif(C, late_attn_peak)`, `layer_span(C, $L$, $L'$)`, `comp_ratio(C, attn, R)`, R > 0.44, `size(C, N)`, N < 45. | 0.31 | 0.8 |

Table 5: **Learned architectural signatures (Pythia-1B, 10-task pool).** Each row is a complete Horn clause; colour identifies predicate type within the conjunction. `named structural motif`; `normalised layer position`; `circuit composition`. Numeric thresholds (plain text) are learned per task and not colour-coded. Confidence = F1 (one-vs-rest; 3 positive, 27 negative circuits). Complexity = $(n_\ell \times \bar{d})/10$. The `motif` column is the only one that varies across tasks; `layer_span` and `composition` predicates are shared structural scaffolding present in every clause.

where $\ell$ is the layer index and $L$ is the number of layers in the model. Component counts are represented as ratios rather than raw counts, for example through predicates such as `component_ratio` and `rel_size`. These encodings allow circuits from models with different depths and widths to be compared in a shared representational space.

The main structural predicates are:

- `component_ratio(C, Type, R)`: the proportion $R$ of components in circuit $C$ whose type is `Type`, e.g. attention heads or MLP blocks;
- `rel_size(C, S)`: the size of circuit $C$ expressed relative to the model or to the circuit pool;
- `layer_span(C, $L_{min}$, $L_{max}$)`: the normalised layer interval covered by circuit $C$;
- `has_motif(C, M)`: whether circuit $C$ contains named motif $M$;
- CFS-derived predicates from Layer 2, which provide functional and causal facts about circuit components.

These predicates are designed to abstract away from model-specific identifiers while preserving structural properties relevant to mechanistic comparison. They facilitate cross-scale comparison, but are not sufficient to guarantee full mechanistic equivalence: qualitatively different computations may occur at similar relative depths in models of different scales. Transfer claims should therefore be interpreted as structural candidate retrieval followed by behavioural and causal validation, rather than as proof of mechanism identity.

| Task | Learned clause $\tau_{\text{arch}}$ (Pythia-14M, 10-task pool) | Conf. | Cmplx. |
|---|---|---|---|
| IOI | `arch_ioi(C) :-`<br>`has_motif(C, mid_layer_attn_peak)`, `layer_span(C, $L$, $L'$)`,<br>`component_ratio(C, attn, R)`, R > 0.64, `size(C, N)`, N < 33. | 0.76 | 0.8 |
| GT | `arch_gt(C) :-`<br>`has_motif(C, early_attn_peak)`, `layer_span(C, $L$, $L'$)`,<br>`component_ratio(C, attn, R)`, R > 0.65, `size(C, N)`, N < 33. | 0.31 | 0.8 |
| Beneficiary | `arch_beneficiary(C) :-`<br>`has_motif(C, mid_layer_attn_peak)`, `layer_span(C, $L$, $L'$)`,<br>`component_ratio(C, attn, R)`, R > 0.61, `size(C, N)`, N < 32. | 0.50 | 0.8 |
| Goal | `arch_goal(C) :-`<br>`size(C, N)`, N < 31. | 0.00 | 0.2 |
| Instrument | `arch_instrument(C) :-`<br>`has_motif(C, mid_layer_attn_peak)`, `layer_span(C, $L$, $L'$)`,<br>`component_ratio(C, attn, R)`, R > 0.60, `size(C, N)`, N < 29. | 0.50 | 0.8 |
| Location | `arch_location(C) :-`<br>`has_motif(C, early_attn_peak)`, `has_motif(C, balanced_mix)`,<br>`layer_span(C, $L$, $L'$)`, `component_ratio(C, attn, R)`, R > 0.60, `size(C, N)`, N < 30. | 0.33 | 0.9 |
| Path | `arch_path(C) :-`<br>`has_motif(C, early_attn_peak)`, `layer_span(C, $L$, $L'$)`,<br>`component_ratio(C, attn, R)`, R > 0.64, `size(C, N)`, N < 31. | 0.31 | 0.8 |
| Source | `arch_source(C) :-`<br>`has_motif(C, early_attn_peak)`, `layer_span(C, $L$, $L'$)`,<br>`component_ratio(C, attn, R)`, R > 0.62, `size(C, N)`, N < 30. | 0.31 | 0.8 |
| Time | `arch_time(C) :-`<br>`has_motif(C, balanced_mix)`, `layer_span(C, $L$, $L'$)`,<br>`component_ratio(C, attn, R)`, R > 0.59, `size(C, N)`, N < 30. | 0.31 | 0.8 |
| Topic | `arch_topic(C) :-`<br>`has_motif(C, early_attn_peak)`, `layer_span(C, $L$, $L'$)`,<br>`component_ratio(C, attn, R)`, R > 0.60, `size(C, N)`, N < 29. | 0.31 | 0.8 |

Table 6: **Learned architectural signatures (Pythia-14M, 10-task pool).** Each row is a complete Horn clause; colour identifies predicate type within the conjunction. `named structural motif`; `normalised layer position`; `circuit composition`. Numeric thresholds are learned per task and not colour-coded. Confidence = F1 on one-vs-rest. Complexity = $(n_\ell \times \bar{d})/10$. IOI is the only well-discriminated circuit; Goal degenerates to a size-only predicate.

### B.2 Named Motif Extraction

Named motifs are detected from Layer 1 graph facts by deterministic topology rules over predicates such as `edge`, `type`, and `layer`. The current motif vocabulary includes:

- `attn_to_attn`: a directed attention-to-attention edge;
- `attn_feeds_mlp`: an attention-to-MLP pair;
- `mlp_gates_attn`: an MLP-to-attention pair;
- `attn_chain_3`: a three-hop attention path;
- `attn_mlp_attn_sandwich`: an attention-to-MLP-to-attention pattern;
- `scaffold_event_entity`: an attention-to-attention-to-MLP pattern associated with scaffold-mediated entity binding.

The motif vocabulary provides interpretable structural predicates for $\text{ILP}_{\text{arch}}$, while the blind structural pool described in Appendix B.3 provides a check against over-reliance on manually named motifs.

### B.3 Dual-Pool ILP Search

$\text{ILP}_{\text{arch}}$ searches two hypothesis pools:

**Pool A: named motifs.** Pool A uses the named motif vocabulary through predicates such as `has_motif(C, M)`. These clauses are interpretable because the motif names correspond to predefined graph-topological patterns.

**Pool B: blind structural predicates.** Pool B enumerates raw edge-type sequences of length two and three, independently of the named motif vocabulary. A blind clause is labelled confirmatory if its edge-type pattern matches a named motif topology. Otherwise, it is flagged as a potential novel structural finding. When a novel blind clause achieves higher F1 than all named-motif clauses for a task, this indicates that the current named motif vocabulary may be incomplete for that task.

| Task | Learned clause $\tau_{\text{arch}}$ (LLaMA-3.2-1B, 10-task pool) | Conf. | Cmplx. |
|---|---|---|---|
| IOI | `arch_ioi(C) :-`<br>`component_ratio(C, attention, R)`, R > 0.59. | 1.00 | 0.2 |
| GT | `arch_gt(C) :-`<br>`has_motif(C, attn_chain_3)`, `layer_span(C, $L$, $L'$)`, `component_ratio(C, attention, R)`, R > 0.37, `size(C, N)`, N < 35. | 0.97 | 0.8 |
| Beneficiary | `arch_beneficiary(C) :-`<br>`has_motif(C, early_attn_peak)`, `layer_span(C, $L$, $L'$)`, `component_ratio(C, attention, R)`, R > 0.28, `size(C, N)`, N < 35. | 0.33 | 0.8 |
| Goal | `arch_goal(C) :-`<br>`has_motif(C, scaffold_event_entity)`, `has_motif(C, late_attn_peak)`, `layer_span(C, $L$, $L'$)`, `component_ratio(C, attention, R)`, R > 0.39, `size(C, N)`, N < 39. | 0.94 | 0.9 |
| Instrument | `arch_instrument(C) :-`<br>`has_motif(C, attn_to_attn)`, `layer_span(C, $L$, $L'$)`, `component_ratio(C, attention, R)`, R > 0.35, `size(C, N)`, N < 38. | 0.94 | 0.8 |
| Location | `arch_location(C) :-`<br>`has_motif(C, late_attn_peak)`, `layer_span(C, $L$, $L'$)`, `component_ratio(C, attention, R)`, R > 0.32, `size(C, N)`, N < 35. | 0.43 | 0.8 |
| Path | `arch_path(C) :-`<br>`has_motif(C, mid_layer_attn_peak)`, `has_motif(C, mlp_heavy)`, `layer_span(C, $L$, $L'$)`, `component_ratio(C, attention, R)`, R > 0.30, `size(C, N)`, N < 35. | 0.71 | 0.9 |
| Source | `arch_source(C) :-`<br>`has_motif(C, mid_layer_attn_peak)`, `layer_span(C, $L$, $L'$)`, `component_ratio(C, attention, R)`, R > 0.39, `size(C, N)`, N < 40. | 0.45 | 0.8 |
| Time | `arch_time(C) :-`<br>`has_motif(C, late_attn_peak)`, `layer_span(C, $L$, $L'$)`, `component_ratio(C, attention, R)`, R > 0.28, `size(C, N)`, N < 34. | 0.43 | 0.8 |
| Topic | `arch_topic(C) :-`<br>`has_motif(C, late_attn_peak)`, `layer_span(C, $L$, $L'$)`, `component_ratio(C, attention, R)`, R > 0.41, `size(C, N)`, N < 41. | 0.45 | 0.8 |

Table 7: **Learned architectural signatures (LLaMA-3.2-1B, 10-task pool).** Each row is a complete Horn clause; colour identifies predicate type within the conjunction. `named structural motif`; `normalised layer position`; `circuit composition`. Numeric thresholds are learned per task and not colour-coded. Confidence = F1 on one-vs-rest. Complexity = $(n_\ell \times \bar{d})/10$.

**Clause selection.** For each task, $\text{ILP}_{\text{arch}}$ selects the clause with the highest F1 score on the one-vs-rest classification problem. When multiple clauses obtain the same F1 score, shorter clauses are preferred. This preference implements a parsimony bias: among equally predictive hypotheses, the selected architectural signature should make the fewest structural commitments.

### B.4 Motif Scoring: Full Specification

Motifs are ranked before clause search using an attribution-weighted information gain score. Let $n_+$ and $n_-$ denote the number of positive and negative circuits. For motif $m$, let tp, fp, tn, and fn denote the standard contingency counts.

**Information gain.** The information gain of motif $m$ is

$$\text{IG}(m) = H(Y) - H(Y \mid m \text{ present}),$$

where $H$ is binary Shannon entropy using log base 2 (Shannon, 1948). This measures how much observing motif $m$ reduces uncertainty about the task label. $\text{IG}(m)$ ties at $H(Y)$ for all perfectly discriminative motifs ($\text{fp} = 0$, $\text{tp} = n_+$); it does not tie in general.

**Gain ratio.** The gain ratio is

$$\text{GR}(m) = \frac{\text{IG}(m)}{H(p_{\text{with}})},$$

where

$$p_{\text{with}} = \frac{\text{tp} + \text{fp}}{n_+ + n_-}.$$

This corrects for rarity bias and becomes more relevant as the circuit pool grows.

**Gini drop.** $\text{GD}(m) = \text{Gini}(n_+, n_+ + n_-) -$ weighted Gini after split, where $\text{Gini}(k, n) = 1 - (k/n)^2 - (1 - k/n)^2$.

This provides an alternative impurity-based measure that is less sensitive to skewed class ratios than entropy.

The final motif ranking score is

$$\text{ATTR_IG}(m) = \text{IG}(m) \times (1 + \bar{s}_m),$$

where $\bar{s}_m$ is the mean attribution score of motif $m$ in the positive circuits. For two-node motifs, $\bar{s}_m$ is computed from `pair_mean_score`, the mean $|\text{grad} \times \text{act}|$ score over all edges of that type in positive circuits. For three-node path motifs, $\bar{s}_m$ is computed from `path3_mean_score`, using the mean $\sqrt{s_1 s_2}$ over matching paths. Motifs absent from the edge-type attribution map receive $\bar{s}_m = 0$ and therefore fall back to plain information gain.

This scoring criterion favours motifs that are both discriminative and causally salient. It does not by itself establish that a motif implements the mechanism; it only prioritises motifs for subsequent ILP clause search and validation.

### B.5 ILP Signature and Complexity Score

The ILP signature $\mathcal{S}_{\text{ILP}}$ summarises the structural properties of the learned architectural signature $\tau_{\text{arch}}$:

$$\mathcal{S}_{\text{ILP}}(\tau_{\text{arch}}) = \langle n_{\text{clauses}},\, \bar{d},\, \mathcal{P}_H,\, \bar{\ell},\, \texttt{complexity} \rangle.$$

**$n_{\text{clauses}}$.** The number of Horn clauses in $\tau_{\text{arch}}$. In the current implementation, $\text{ILP}_{\text{arch}}$ returns a single best clause per task ($n_{\text{clauses}} = 1$); the field is retained for generality when learning clause sets.

**$\bar{d}$: mean derivation depth.** The maximum predicate nesting depth in the clause body, measured by counting nested parentheses:

$$\bar{d} = \max_{\text{clause}} \max_{i} \texttt{depth}(b_i),$$

where $\texttt{depth}(b_i)$ is the parenthesis nesting depth of literal $b_i$. For flat conjunctions of ground predicates, $\bar{d} = 1$. Deeper values arise when predicates take compound terms as arguments.

**$\mathcal{P}_H$: predicate set.** The set of predicate names appearing in the clause body. Used as the symbolic vocabulary for the Jaccard similarity term in the signature distance metric (Appendix B.6). Motifs (`has_motif` values) are stored separately as a topology discriminator, since two clauses may share predicate names while referring to structurally distinct circuit types.

**$\bar{\ell}$: mean clause length.** The number of literals in the clause body ($\bar{\ell} = n_\ell$ for single-clause signatures), counted by parenthesis-aware splitting on top-level commas.

**`complexity`: composite score.**

$$\texttt{complexity} = \frac{n_\ell \times \bar{d}}{10},$$

where $n_\ell = \bar{\ell}$ and $\bar{d}$ is the nesting depth above. The factor of 10 normalises to a convenient scale and has no theoretical significance. This composite penalises clauses that are simultaneously long and deeply nested; a flat 3-literal clause yields $\texttt{complexity} = 0.3$, while a 5-literal depth-2 clause yields 1.0. `complexity` is used as a descriptive summary statistic and as a tiebreaker in the transfer fallback procedure; it is not used during clause search.

### B.6 ILP Distance Metric

To compare two architectural signatures, we compute a distance over their ILP signatures:

$$d(\mathcal{S}_1, \mathcal{S}_2) = \sqrt{\left(\frac{n^{(1)}_{\text{clauses}} - n^{(2)}_{\text{clauses}}}{n_{\max}}\right)^2 + \lambda \cdot \big(1 - J(\mathcal{P}_1, \mathcal{P}_2)\big)}, \tag{2}$$

where $J(\mathcal{P}_1, \mathcal{P}_2) = |\mathcal{P}_1 \cap \mathcal{P}_2| / |\mathcal{P}_1 \cup \mathcal{P}_2|$ is predicate-set Jaccard similarity and $\lambda = 0.5$. This metric is intentionally coarse. It captures differences in clause count and predicate vocabulary, but it does not directly incorporate derivation depth $\bar{d}$, clause length $\bar{\ell}$, or the exact numeric thresholds inside predicates. As a results, two theories with identical predicate vocabularies but different proof structure may appear similar. We use this distance as an approximate structural comparison and as an acceptance criterion for fallback transfer, not as a complete measure of mechanistic equivalence.

### B.7 $\theta$-Subsumption: Operational Definition and Tractability

Clause $\gamma_1$ *$\theta$-subsumes* clause $\gamma_2$, written $\gamma_1 \preceq_\theta \gamma_2$, if there exists a substitution $\theta$ mapping variables in $\gamma_1$ to terms in $\gamma_2$ such that

$$\gamma_1 \theta \subseteq \gamma_2,$$

treating clauses as sets of literals. Equivalently, $\gamma_1$ is at least as general as $\gamma_2$: every model of $\gamma_1$ is a model of $\gamma_2$. The relation $\preceq_\theta$ is reflexive and

transitive. When clauses are considered up to variable renaming, it supports the following mechanistic interpretations:

- $\gamma_1 \prec_\theta \gamma_2$ (strict): $\gamma_1$ *refines* $\gamma_2$ — it makes strictly stronger structural commitments, matching a proper subset of circuits;
- $\gamma_1 \equiv_\theta \gamma_2$ (mutual subsumption): $\gamma_1$ and $\gamma_2$ are *equivalent* up to variable renaming, they characterise the same set of circuits;
- $\gamma_1 \not\preceq_\theta \gamma_2$ and $\gamma_2 \not\preceq_\theta \gamma_1$: the two signatures are *incomparable*, neither is a generalisation of the other.

Subsumption checking is NP-complete in general (Plotkin, 1970; Gottlob, 1987). Two restrictions make it tractable in our setting.

**Bounded clause length.** Mode declarations in $\text{ILP}_{\text{arch}}$ bound the number of literals in any learned clause body ($\bar{\ell} \leq 5$ in the current configuration). Under a fixed literal bound $k$, subsumption reduces to a matching problem of size $O(|\mathcal{P}_H|^k)$, which is polynomial for small $k$.

**Datalog-range predicates.** All predicates in $\mathcal{B}_{\text{arch}}$ are range-restricted: every variable in a clause body appears in a positive literal. This eliminates existential variables that would otherwise exponentially expand the substitution search space.

**Strict vs. Relaxed Subsumption** Under strict $\theta$-subsumption, numeric constants must match exactly. This makes many learned clauses incomparable, even when they share the same predicate structure but differ only in thresholds. For example, clauses containing $R > 0.63$ and $R > 0.35$ do not match strictly because the constants differ.

For interpretability analysis, we also report *relaxed* $\theta$-subsumption, where numeric constants are treated as wildcards (any threshold matches any threshold). This comparison abstracts away from exact thresholds and compares predicate structure. It should therefore be interpreted as structural generalisation, not as strict logical entailment under the original numeric constraints. Consider two learned clauses:

$$\gamma_{\text{ioi}} : \texttt{arch_ioi}(C) \leftarrow \texttt{component_ratio}(C, \texttt{attn}, R), \; R > 0.63.$$

$$\begin{aligned}\gamma_{\text{time}} : \texttt{arch_time}(C) \\ \leftarrow \texttt{has_motif}(C, \texttt{early_attn_peak}), \\ \texttt{layer_span}(C, L_{\min}, L_{\max}), \\ \texttt{component_ratio}(C, \texttt{attn}, R), \\ R > 0.35, \\ \texttt{size}(C, N), \; N < 36.\end{aligned}$$

$\gamma_{\text{ioi}} \preceq_\theta \gamma_{\text{time}}$ (relaxed): under numeric wildcarding, IOI's single literal $\texttt{component_ratio}(C, \texttt{attn}, R)$ with $R > _$ matches the corresponding literal in $\gamma_{\text{time}}$. Since all of $\gamma_{\text{ioi}}$'s body literals appear in $\gamma_{\text{time}}$'s body (under wildcard matching), IOI *subsumes* Time. IOI is the more general signature.

$\gamma_{\text{time}} \not\preceq_\theta \gamma_{\text{ioi}}$ (relaxed): $\gamma_{\text{time}}$ contains `has_motif`, `layer_span`, and `size` literals that have no counterpart in $\gamma_{\text{ioi}}$. Time makes strictly stronger structural commitments.

Similarly, Time $\preceq_\theta$ Path (relaxed), because Path adds $\texttt{has_motif}(C, \texttt{attn_chain_3})$ to Time's predicates. Location is *incomparable* to both Time and Path because it uses different motifs (`mlp_heavy`, `mid_layer_attn_peak`).

**Mechanistic interpretation.** The relaxed subsumption hierarchy reveals that attention component ratio is the minimal structural invariant shared by all circuits in Pythia-1B (captured by IOI's clause). Semantic role circuits *specialise* this structural invariant by adding motif and size constraints. Within the role family, Location and Time represent *distinct specialisation paths*: both refine a shared attention-ratio predicate, but through different motifs, indicating different structural realisations of binding. This is a testable claim: a circuit satisfying $\gamma_{\text{loc}}$ need not satisfy $\gamma_{\text{time}}$ (and vice versa), predicting that the location and time circuits are structurally non-interchangeable despite implementing the same operation type.

### B.8 Learned Architectural Clauses

Table 5–7 list the full set of learned clauses from the 10-task pool for all three models. Each clause is learned via one-vs-rest ILP with 3 positive circuits per task and 27 negatives. Confidence is the F1 score on this classification; complexity is $(n_\ell \times \bar{d})/10$.

## C Formal Circuit Representation: Logical Foundation

The formal circuit representation (FCR) stores circuit instances as sets of ground logical facts organised across our layers. The storage substrate

uses OWL 2 ontologies (providing SROIQ semantics and SPARQL-based extraction), but the logical form exposed to $\text{ILP}_{\text{arch}}$ is Prolog-style Horn clauses.

### C.1 Layers Structure

**Layer 0: Provenance.** Task family, evaluation corpus, annotation schema, and experimental protocol. Known from experimental design; not inferred.

**Layer 1: Structural.** Per-node facts: `node(`$C$`,`$v$`)`, `type(`$v$`,`$\tau$`)`, `layer(`$v$`,`$\ell$`)`. Per-edge facts: `edge(`$C$`,`$u$`,`$v$`)`. Aggregate facts: `component_ratio(`$C$`,attn,`$r$`)`, `rel_size(`$C$`,`$r$`)`, `layer_span(`$C$`,`$\ell_{\min}$`,`$\ell_{\max}$`)`, `size(`$C$`,`$n$`)`, `density(`$C$`,`$d$`)`, `skip_connections(`$C$`,`$k$`)`, `hub_count(`$C$`,`$k$`)`, `faithfulness(`$C$`,`$f$`)` where $f = (\text{Acc}_T(C) - \text{Acc}_T^{\text{abl}})/(\text{Acc}_T(M) - \text{Acc}_T^{\text{abl}})$ (Mueller et al., 2025). Named motifs: `has_motif(`$C$`,`$m$`)`. Edge attribution scores are stored per edge: `edge_score(`$u$`,`$v$`,`$s$`)`.

**Layer 2: Causal Functional Signature.** `dla_score(`$v$`,`$\delta$`)` and `attends_to(`$v$`,`$r$`,`$p$`)`, where $r$ is drawn from either the linguistic role vocabulary ($r \in$ {subject, object, verb, function, entity, other}) or a task-specific vocabulary (e.g., $r \in$ {role_filler, scaffold, other} for semantic roles). $p \in [0, 1]$ is the attribution-weighted attention frequency over $\mathcal{D}_T$. Both profiles are stored; the linguistic profile is used for cross-task comparison and the task-specific profile for within-task characterisation. See Appendix D for labeller details.

**Layer 3: Architectural Signatures.** $\text{ILP}_{\text{arch}}$ clauses of the form `arch_`$\tau$`(`$C$`) :- `$b_1$`, . . . , `$b_k$, where each $b_i$ is a scale-invariant predicate from $\mathcal{B}_{\text{arch}}$.

### C.2 OWA/CWA Reconciliation

We use OWL that operates under the Open World Assumption (OWA): absence of a fact means *unknown*, not *false*. $\text{ILP}_{\text{arch}}$ requires the Closed World Assumption (CWA). We reconcile this by applying a *closed-world completion* at the extraction boundary: for a fixed predicate set $\mathcal{P}_{\text{target}}$ and admissible domain $\mathcal{U}_{p,i}$ per argument position, any ground atom not entailed by the OWL representation is treated as false in the ILP training database. Negation-as-failure in learned clauses is therefore interpreted relative to this completed extraction, not relative to the OWL ontology directly.

### C.3 SPARQL Extraction

Background knowledge predicates are populated via SPARQL queries. For example, normalised depth for attention heads is computed as:

```
SELECT ?c ?depth WHERE {
  ?c rdf:type :AttentionHead .
  ?c :inLayer ?layer .
  ?c :modelDepth ?totalLayers .
  BIND(?layer/?totalLayers AS ?depth)
}
```

## D Token Role Labelling

CFS attention profiles require assigning a role label to each input token position. We use two labelling levels: a shared linguistic labeller for cross-task comparison, and task-specific labellers for within-task characterisation.

### D.1 Linguistic Roles (Shared)

A spaCy dependency parser (`en_core_web_sm`) assigns labels from the vocabulary {`subject`, `object`, `verb`, `function`, `entity`, `other`}. Tokens with dependency relations `nsubj`/`nsubjpass` are labelled `subject`; `dobj`/`pobj`/`iobj` are labelled `object`; verbs by POS tag; named entities by NER; determiners, prepositions, conjunctions, and auxiliaries are labelled `function`; all others are `other`. Subword tokens inherit the label of their aligned word via character-offset mapping. A capitalisation-based heuristic is used as fallback when spaCy is unavailable.

The `function` role separates structurally important closed-class words (determiners, prepositions) from open-class content, preventing them from being absorbed into the uninformative `other` category. A fine-grained variant (12 roles) further distinguishes `subject_agent` from `subject_experiencer`, `object_patient` from `object_theme`, `verb_action` from `verb_stative`, and function subtypes (`function_prep`, `function_det`, `function_aux`, `function_conj`). The coarse 6-role vocabulary is used for the cross-task results reported in the main text; the fine-grained vocabulary is available for detailed linguistic analysis.

### D.2 Task-Specific Roles

Each task family provides a labeller that uses metadata from the contrastive pair to assign task-relevant roles:

**Semantic Role Binding.** Roles: `role_filler` (the target word the model should predict), `scaffold_prep` (the preposition: "in", "from", "through"), `scaffold_det` (the determiner within the scaffold: "the", "a"), `predicate_verb` (the main verb), `agent` (subject noun), `theme` (direct object noun), `modifier` (adjectives and adverbs), `other`. Scaffold prepositions and determiners are distinguished to reveal whether components attend to the structural cue (preposition) or its accompanying determiner. The agent and theme are identified via dependency parsing when available.

**IOI.** Roles: `repeated_name` (the name appearing twice, target for copying), `first_occurrence` (its first mention), `distractor_name` (the foil name), `io_position` (the "to" before the indirect object slot), `context_verb` (the main verb: "gave", "sent"), `context_setting` (the setting clause), `other`. Names are identified from the `target_word` and `incorrect_word` columns; the verb is identified via dependency parsing.

**Greater-Than.** Roles: `first_number` (the four-digit year, e.g., 1352), `second_number` (the two-digit continuation, e.g., 13), `comparison_from` (the "from" keyword), `comparison_to` (the "to" keyword), `year_marker` ("the year" structural frame), `subject_noun` ("The war"), `predicate_verb` ("lasted"), `other`. The "from" and "to" keywords are distinguished as separate comparison markers rather than grouped into a single "comparison_context" category, enabling the CFS to reveal whether components track the source, the target, or the relational structure of the comparison.

### D.3 Design Rationale

The two-level design serves distinct purposes. The linguistic profile produces a *task-agnostic* characterisation: two circuits attending primarily to `subject` tokens share a functional property regardless of whether they implement location binding or time binding. This enables the cross-task CFS clustering test (Section 4.1). The task-specific profile produces a *task-informed* characterisation: a component attending to `scaffold` vs. `role_filler` reveals whether it processes the prepositional cue or the target entity, providing finer-grained mechanistic insight within a task family.

New task families can be added by registering a labeller function mapping `(text, token_strs, metadata)` $\rightarrow$ `List[str]` in the labeller registry, with no changes to the CFS computation or ILP pipeline.

## E Task and Model Selection

### E.1 Task Selection Rationale

The three task families are chosen to cover distinct operation types while ensuring that within-family comparison is meaningful.

**Semantic Role Binding.** Semantic roles provide a natural family of related tasks: each role (Location, Time, Path, Instrument, etc.) implements the same abstract operation, binding an entity to a thematic position within a predicate frame, but with different entities and structural cues. This enables within-family comparison: circuits implementing the same $\kappa$ should share structural properties that differ from circuits implementing a different $\kappa$. The primary evaluation uses 3 roles (Location, Time, Path) for which validated contrastive datasets exist across all three models. The scaled evaluation extends to 8 roles (adding Beneficiary, Goal, Instrument, Source, Topic) to test generalisation under increased within-family diversity.

**IOI.** Indirect Object Identification is the most extensively studied circuit in the MI literature (Wang et al., 2023; Conmy et al., 2023), providing a well-characterised reference mechanism with known sub-components (name movers, duplicate-token heads, S-inhibition heads). Its selection operation ($\kappa =$ `selection`) is qualitatively different from binding: the circuit identifies and copies a previously mentioned entity rather than constructing a new role-filler association. This ensures the framework is tested on a genuine cross-family contrast, not merely variations of the same operation.

**Greater-Than.** Numerical comparison introduces a non-linguistic computational domain. The comparison operation ($\kappa =$ `comparison`) requires extracting numerical values and evaluating a relational predicate, distinct from both binding and selection. This tests whether the formal framework generalises beyond linguistic tasks.

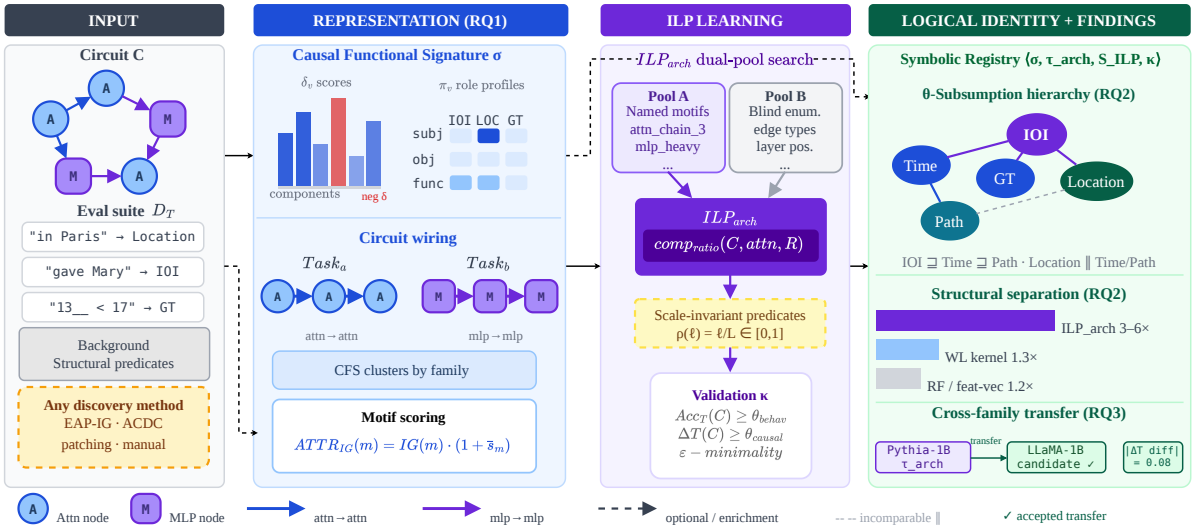


Figure 5: **Pipeline for inductive circuit theory construction.** A discovered circuit $C$ is encoded in a Formal Circuit Representation (FCR) across four layers (provenance, structure, causal behaviour, and learned signatures). A Causal Functional Signature (CFS, $\sigma$) is derived from causal attribution evidence (RQ1). $\text{ILP}_{\text{arch}}$ learns an architectural signature $\tau_{\text{arch}}$ over scale-invariant predicates. Validated triples $\langle \sigma, \tau_{\text{arch}}, \mathcal{S}_{\text{ILP}}, \kappa \rangle$ are stored in a Symbolic Registry supporting $\theta$-subsumption-based comparison (RQ2). Transfer Engine generates and validates candidates in a target model, with fallback to full rediscovery (RQ3). Dashed arrows indicate optional paths.

### E.2 Model Cards and Selection Rationale

Table 8 shows the cards for the models used in the paper. Pythia models (Biderman et al., 2023) were released under (Apache license 2.0) and mainly trained for English, while LLaMA (Grattafiori et al., 2024) was released under (Meta Llama 3 Community) and has additional multilingual capabilities, including, English, German, French, Italian, Portuguese, Hindi, Spanish, and Thai. We explain our selection rational next.

**Pythia-14M and Pythia-1B.** These two models share the same architecture (GPT-NeoX), tokeniser, and training corpus (The Pile), differing only in capacity (6 vs. 16 layers, 4 vs. 8 heads). This controls for confounds: any structural differences between circuits discovered in the two models reflect differences in how the mechanism is implemented at different scales, not differences in architecture or data distribution. The 14M model also serves as a negative control for CFS: its limited capacity produces structurally homogeneous circuits (all include all MLP blocks), testing whether the framework correctly reports this uniformity rather than producing artefactual differentiation.

**LLaMA-3.2-1B.** This model uses a different architecture (grouped-query attention, RoPE positional encoding, SiLU activation) and tokeniser (SentencePiece vs. GPT-NeoX BPE). It tests whether the structural patterns identified by CFS and $\text{ILP}_{\text{arch}}$, attention-dominated IOI vs. MLP-dominated binding, are architecture-specific properties of Pythia or general properties of how transformers implement these operations. The 1B scale matches Pythia-1B, isolating the effect of architecture from scale.

| | Pythia-14M | Pythia-1B | LLaMA-3.2-1B |
|---|---|---|---|
| Architecture | GPT-NeoX | GPT-NeoX | LLaMA |
| Layers | 6 | 16 | 16 |
| Heads | 4 | 8 | 32 (GQA) |
| $d_{\text{model}}$ | 128 | 2048 | 2048 |
| Parameters | 14M | 1B | 1.2B |
| Tokeniser | NeoX BPE | NeoX BPE | SentencePiece |
| Training data | The Pile | The Pile | Undisclosed |
| Positional enc. | Rotary | Rotary | RoPE |
| Activation | GELU | GELU | SiLU |

Table 8: Model specifications

### E.3 Circuit Discovery Method

All circuits in this work are extracted using EAP-IG (Hanna et al., 2024) with $k = 200$ edges and 3 integrated gradient steps. The framework is agnostic to the discovery algorithm: the formal circuit representation (Section 3.2) operates over stored nodes and attributed edges, not over the procedure that produced them. Any method that outputs a sparse subgraph with edge attribution scores, including ACDC (Conmy et al., 2023),

path patching (Goldowsky-Dill et al., 2023), activation patching (Meng et al., 2022), or manual circuit identification, produces valid input. We use EAP-IG because pre-validated circuits are available for our tasks across all three models; the choice does not limit the generality of the results.

### E.4 Implementation Details

All experiments were conducted on an NVIDIA RTX A6000 GPU and using Python 3.11.13 with the following core dependencies: NumPy 1.26.4, scikit-learn 1.7.0 (Pedregosa et al., 2011), SciPy 1.15.3, PyTorch 2.7.1 (Paszke et al., 2019), TransformerLens 2.16.1 (Nanda and Bloom, 2022), Transformers 4.52.4, spaCy 3.8.14 (Honnibal et al., 2020) (`en_core_web_sm`), and `trace` 0.2.0 package (Aljaafari et al., 2025a).

### E.5 Circuit Inventory

Table 9 shows the cicuit sizes for all the tasks used in the paper.

| Model | Task | Nodes | Attn | MLP | Attn% |
|---|---|---|---|---|---|
| Py-14M | IOI | 26 | 20 | 6 | 77 |
| | Location | 22 | 16 | 6 | 73 |
| | Time | 22 | 16 | 6 | 73 |
| | Path | 24 | 19 | 6 | 77 |
| | GT | 27 | 21 | 6 | 78 |
| | Beneficiary | 23 | 17 | 6 | 74 |
| | Goal | 24 | 18 | 6 | 75 |
| | Instrument | 22 | 16 | 6 | 73 |
| | Source | 24 | 18 | 6 | 75 |
| | Topic | 22 | 16 | 6 | 73 |
| Py-1B | IOI | 38 | 26 | 11 | 70 |
| | Location | 35 | 19 | 15 | 56 |
| | Time | 31 | 15 | 16 | 48 |
| | Path | 31 | 15 | 16 | 49 |
| | GT | 41 | 25 | 16 | 62 |
| | Beneficiary | 34 | 18 | 16 | 53 |
| | Goal | 35 | 20 | 16 | 56 |
| | Instrument | 32 | 16 | 16 | 49 |
| | Source | 33 | 17 | 16 | 52 |
| | Topic | 36 | 20 | 16 | 55 |
| LL-1B | IOI | 42 | 30 | 12 | 71 |
| | Location | 28 | 12 | 16 | 44 |
| | Time | 27 | 11 | 16 | 40 |
| | Path | 27 | 11 | 16 | 41 |
| | GT | 29 | 14 | 15 | 48 |
| | Beneficiary | 27 | 11 | 16 | 40 |
| | Goal | 32 | 16 | 16 | 51 |
| | Instrument | 30 | 14 | 16 | 47 |
| | Source | 31 | 16 | 15 | 51 |
| | Topic | 34 | 18 | 16 | 52 |

Table 9: **Circuit inventory (30 circuits, 10 tasks $\times$ 3 models).** Values are means across 3 prompt splits. Py = Pythia, LL = LLaMA-3.2. IOI is consistently attention-dominated (70–77%) across all models. Semantic role circuits are MLP-heavy at 1B scale (40–56% attention); Pythia-14M circuits are uniformly attention-heavy (73–78%) due to the small model's limited MLP capacity.

### E.6 Dataset Statistics

Table 10 reports the number of examples retained after the `both_correct` filtering step (both the clean and corrupted forward passes must yield the correct answer) for each task and model. All the data is in English and IOI and GreaterThan use hand-crafted templates; semantic role datasets are the same used in (Aljaafari et al., 2025b). Variation across models reflects differences in base accuracy: smaller models solve fewer prompts correctly, reducing the usable pool.

| Task | Py-14M | Py-1B | LLaMA-1B |
|---|---|---|---|
| IOI | 318 | 388 | 599 |
| GreaterThan | 733 | 862 | 339 |
| Beneficiary | 773 | 439 | 1,181 |
| Goal | 845 | 786 | 374 |
| Instrument | 459 | 909 | 825 |
| Location | 904 | 736 | 639 |
| Path | 802 | 537 | 734 |
| Source | 667 | 396 | 452 |
| Time | 681 | 548 | 587 |
| Topic | 1,120 | 389 | 916 |

Table 10: **Dataset sizes after `both_correct` filtering.** Each cell is the number of (clean, corrupted) pairs on which both forward passes produce the correct answer. Semantic role counts are for the `both_correct` split used in all experiments.

## F Linguistic Grounding

The operation types that emerge from CFS and $\text{ILP}_{\text{arch}}$ ($\kappa$ = `binding` vs. $\kappa$ = `selection`) correspond to well-studied distinctions in formal linguistics.

**Binding and $\theta$-role assignment.** In formal semantics, $\theta$-role assignment is the process by which a predicate assigns thematic roles (Agent, Patient, Location, Time, etc.) to its arguments (Chomsky, 1993). The semantic role circuits we study implement a computational analogue: the circuit *binds* an entity (the role filler) to a structural position within the predicate frame, mediated by syntactic markers (prepositions and determiners). CFS confirms this interpretation: at the 1B scale, binding circuits attend primarily to `subject` positions (the entity being bound) and to `scaffold` positions (the structural cue that signals which role is being assigned). The MLP-dominated edge composition (64–96 mlp→mlp edges) suggests that binding is implemented through feature transformation rather than token copying, consistent with the view

that $\theta$-role assignment is a compositional operation that constructs new representations rather than merely moving existing ones.

**Selection and referential processing.** IOI circuits implement a qualitatively different operation: identifying a previously mentioned entity (the indirect object) and copying it to the prediction position. This is closer to anaphora resolution or coreference than to argument structure. CFS confirms the distinction: IOI components attend to positional rather than thematic features (71–76% `other`-dominant), and the circuit is dominated by attention→attention edges (85 in Pythia-1B), reflecting a copy-via-attention mechanism. The task-specific profile reveals that 10 of 27 causal components attend to `repeated_name` positions, directly implementing the "find and copy the repeated token" strategy identified in prior work (Wang et al., 2023).

**Greater-Than and relational comparison.** Greater-Than circuits occupy an intermediate structural position: they share the predicate vocabulary of semantic role circuits (`has_motif`, `layer_span`, `size`) but use a more balanced attention/MLP composition (59% attention in Pythia-1B). This is consistent with numerical comparison requiring both relational processing (comparing two quantities, analogous to predicate-argument structure) and feature extraction (identifying the numerical tokens, analogous to entity recognition). The task-specific profile shows attention to `comparison_context` tokens ("from", "to"), suggesting that the circuit tracks the relational structure of the comparison rather than simply extracting digit values.

**Implications.** The alignment between our computationally derived categories (binding, selection, comparison) and linguistically motivated distinctions ($\theta$-role assignment, referential processing, relational semantics) is not a design choice; it emerges from the data. CFS and $\text{ILP}_{\text{arch}}$ were not given linguistic priors; the operation types were assigned post hoc based on the task definitions. That the resulting structural patterns (MLP-mediated vs. attention-mediated vs. hybrid) mirror the functional distinctions posited by linguistic theory provides convergent evidence. This evidence indicates that models appear to have developed computational strategies that reflect genuine properties of language processing, not merely surface statistical regularities.

## G Hyperparameters and Thresholds

Table 11 lists the threshold parameters used in the validation criteria (Section 3.5) and transfer procedure (Section 3.7).

| Parameter | Default | Rationale |
|---|---|---|
| $\theta_{\text{behav}}$ | 0.50 | Minimum fraction of $\mathcal{D}_T$ on which the circuit supports the target behaviour. Set conservatively to accommodate circuits in small models where faithfulness may be limited. |
| $\theta_{\text{causal}}$ | 0.10 | Minimum task-performance drop upon circuit ablation. Set low to retain circuits whose causal contribution is positive but small. |
| $\epsilon$ | 0.20 | Approximate minimality tolerance for greedy edge ablation. A circuit is treated as approximately minimal if no greedy removal path can reduce the circuit by more than an $\epsilon$ fraction while preserving both behavioural and causal validation criteria. |
| $\epsilon_{\text{DLA}}$ | 0.05 | DLA score below which a component is flagged as structurally present but causally marginal in the CFS. |
| $\delta_\rho$ | $1/L_\beta$ | Depth tolerance for candidate generation during transfer: one layer in the target model. This tolerance scales automatically with target-model depth. |
| $\delta_{\text{ILP}}$ | 0.30 | Maximum ILP signature distance (Appendix B.6) for the transfer fallback procedure. Sensitivity analysis on the 10-task pool (Appendix H.2) finds optimal F1 at $\delta_{\text{ILP}} = 0.10$; the default 0.30 is a conservative upper bound. |
| $\lambda$ | 0.50 | Weight on the predicate-set Jaccard term in the ILP signature distance metric (Appendix B.6). |

Table 11: Hyperparameters and threshold definitions. All values are defaults used in our experiments; they may require adjustment for tasks with very small evaluation suites, different ablation protocols, or circuits of atypical size.

**Threshold selection.** The validation thresholds $\theta_{\text{behav}}$ and $\theta_{\text{causal}}$ are not optimised on held-out data. They are set conservatively to retain circuits with positive but small causal contributions. We perform a limited sensitivity analysis for the transfer distance threshold $\delta_{\text{ILP}}$ in Appendix H.2; broader sensitivity analysis over all validation thresholds is left as a limitation.

**Random-subgraph significance tests.** For each validated circuit, we compare the observed causal effect $\Delta_T(C)$ against size-matched random subgraphs sampled from the same model. Statistical significance is assessed using permutation tests at $\alpha = 0.05$. This check ensures that the observed causal degradation is not explained merely by ablating a subgraph of comparable size. In the evaluated settings, random circuits yield $\Delta_T \approx 0$, supporting the interpretation that validated circuits capture task-relevant causal structure rather than generic ablation sensitivity.

| | IOI | LOC | PATH | TIME | GT |
|---|---|---|---|---|---|
| **IOI** | , | 0.35 | 0.38 | 0.45 | 0.32 |
| **LOC** | 0.35 | , | 0.08 | 0.13 | 0.05 |
| **PATH** | 0.38 | 0.08 | , | 0.11 | 0.08 |
| **TIME** | 0.45 | 0.13 | 0.11 | , | 0.13 |
| **GT** | 0.32 | 0.05 | 0.08 | 0.13 | , |

Table 12: **ILP signature distance (Pythia-1B).** Two clusters emerge: IOI is distant from all other circuits ($d = 0.32$–$0.45$), while semantic roles and GreaterThan form a tight cluster ($d = 0.05$–$0.13$). GT's structural proximity to role circuits reflects its learned clause using the same predicate set (`has_motif`, `layer_span`, `size`, `component_ratio`) despite implementing a different computation. The transfer threshold $\delta_{\text{ILP}} = 0.30$ cleanly separates IOI from all other circuits.

# H Additional Results

## H.1 Full ILP Distance Matrix (10 Tasks)

Figures 6, 7, and 8 show the full 10×10 ILP signature distance matrices for all three models. The distance metric uses clause count, predicate-set Jaccard, and motif-set Jaccard (Appendix B.6).

### H.1.1 Within-Task Structural Variance

To assess whether circuits discovered from different prompt subsets for the *same task* are structurally consistent, we compute pairwise WL kernel distances among the 3 splits per task (within-task) and compare to the mean distance to other tasks' circuits (between-task). The ratio (between/within) measures how much more similar same-task circuits are to each other than to different-task circuits.

On LLaMA-3.2-1B, the IOI within/between ratio is 2.3×, the highest across all models and tasks, confirming that the IOI attention-chain topology is a robust structural invariant of the architecture, not an artefact of the specific prompts used for discovery. For Pythia-14M, all tasks have ratios $\approx 1.0$, confirming the model's structural homogeneity.

## H.2 $\delta_{\text{ILP}}$ Sensitivity Analysis

Figure 10 shows the transfer-pair classification performance as the acceptance threshold $\delta_{\text{ILP}}$ is varied from 0.05 to 0.50. A pair is "same-family" if both circuits share a task family label and "cross-family" otherwise. Accepting a pair (predicting same-family) when $d < \delta_{\text{ILP}}$ yields the precision/recall/F1 curve.

## H.3 Edge Type Distribution

Table 13 breaks down the 200 retained edges per circuit by type (attn→attn, attn→mlp, mlp→attn, mlp→mlp) for both 1B-scale models. The distribution quantifies the wiring distinction identified by CFS in Section 4.1.

| Task | a→a | a→m | m→a | m→m | Attn% |
|---|---|---|---|---|---|
| *Pythia-1B* | | | | | |
| IOI | 85 | 19 | 44 | 11 | 73% |
| Location | 4 | 37 | 39 | 67 | 49% |
| Time | 1 | 19 | 27 | 96 | 45% |
| Path | 7 | 24 | 53 | 64 | 54% |
| GT | 25 | 34 | 47 | 44 | 59% |
| *LLaMA-3.2-1B* | | | | | |
| IOI | 44 | 15 | 33 | 10 | 69% |
| Location | 2 | 18 | 23 | 85 | 47% |
| Time | 0 | 9 | 16 | 105 | 37% |
| Path | 0 | 11 | 19 | 80 | 41% |
| GT | 9 | 14 | 27 | 59 | 47% |

Table 13: **Edge type distribution (1B-scale models).** IOI has 85/44 attn→attn edges in Pythia/LLaMA; semantic role circuits have 64–105 mlp→mlp edges. LLaMA Time has zero attn→attn edges.

## H.4 Baseline Comparison Details

ILP signature distance achieves ∼3× better IOI separation than the WL kernel on Pythia-1B and 2.7× better on LLaMA-1B (4.15× vs. 1.53×). The WL kernel correlates with ILP ($r = 0.81$–$0.95$) but captures only local graph topology, not conjunctive predicate structure. The random forest has access to the same features as ILP, but does not produce compact inspectable clauses over them; its 60% LOO accuracy (misclassifying GT and IOI) indicates that explicit logical structure contributes useful discrimination beyond the feature inventory alone. On Pythia-14M, no method achieves meaningful separation (all $\leq 1.1\times$), confirming structural homogeneity as a genuine property.

## H.5 Random Circuit Baseline

For each circuit in the corpus, 50 size-matched random subgraphs are sampled: each random subgraph contains the same number of nodes $|V|$ and edges $|E|$ as the real circuit, selected uniformly at random from the model graph. Each random subgraph is evaluated using the same CNP metric as the real circuit, producing a causal contribution score $\Delta_T^{\text{rand}} = \text{AccT}(M^{\text{rand}}) - \text{AccT}(M^{\text{ablate}})$, where $M^{\text{ablate}}$ denotes the model with all circuit edges zeroed. A one-sided permutation test then

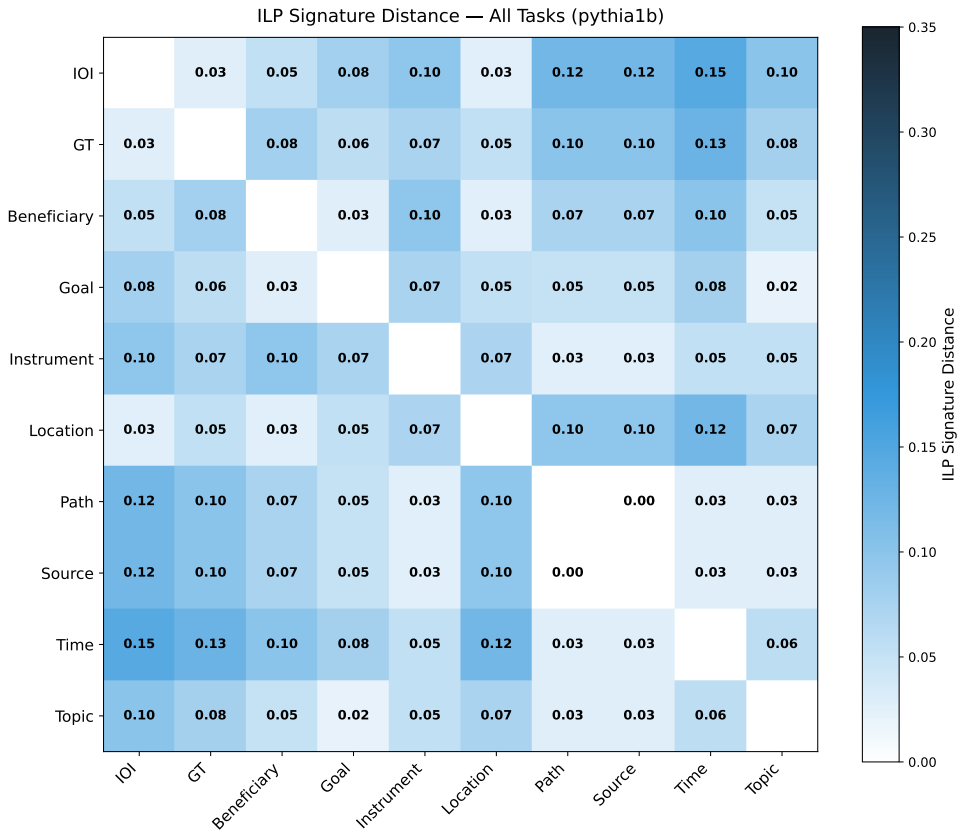


Figure 6: **Full ILP signature distance (Pythia-1B, 10 tasks).** IOI and GT retain high separation from the role cluster. Within roles, Source and Time form a tight sub-cluster, while Goal is the most distant role. The expanded matrix confirms that the 5-task results (Table 12) generalise: IOI remains structurally distinct across all task comparisons.

checks whether the real circuit's $\Delta_T$ exceeds the empirical distribution of $\Delta_T^{\text{rand}}$ at $p < 0.05$. Table 14 reports results across 34 circuits for Pythia-14M, 35 for Pythia-1B, and 33 for LLaMA-3.2-1B. Random subgraphs consistently yield $\Delta_T^{\text{rand}} \approx 0$, confirming that causal contribution is a property of the discovered circuit structure rather than of circuit size.

| Model | Circuits | Sig. | Non-sig. | Rate |
|---|---|---|---|---|
| Pythia-14M | 34 | 29 | 5 | 85% |
| Pythia-1B | 35 | 28 | 7 | 80% |
| LLaMA-3.2-1B | 33 | 26 | 7 | 79% |

Table 14: **Random circuit baseline summary.** For each model, the corpus of discovered circuits is tested against 50 size-matched random subgraphs per circuit. Sig. = real circuit $\Delta_T$ significantly exceeds the random distribution ($p < 0.05$, one-sided permutation test).

**Non-significant cases.** All non-significant circuits have negative or marginal $\Delta_T$ values, meaning the circuit's pathway performs at or below the ablated baseline; this reflects genuine properties of those circuits rather than a failure of the methodology. On Pythia-14M (5 non-significant): the three IOI circuits (primary and splits s0, s1) have $\Delta_T \in \{-0.72, -0.81, -0.60\}$, consistent with the known sign reversal of the IOI causal contribution at the 14M scale. Two Beneficiary splits (s0: $-0.21$; s1: $+0.13$, marginally non-significant) show near-zero contributions. On Pythia-1B (7 non-significant): three Time circuits (primary: $-0.02$; s0: $-0.31$; s2: $-0.13$) and one Path split (s1: $-0.02$) have negative $\Delta_T$, and three Beneficiary splits (s0: $-0.11$; s1: $-0.04$; s2: $-0.21$) contribute below the ablated baseline. On LLaMA-3.2-1B (7 non-significant): all four Time circuits (primary: $-0.75$; s0: $-0.51$; s1: $-0.34$; s2: $-0.75$) and three Beneficiary splits (s0: $-0.14$; s1: $-0.03$; s2: $-0.01$) are non-significant, following the same Time sign-reversal and Beneficiary marginal-contribution patterns seen across all three models.

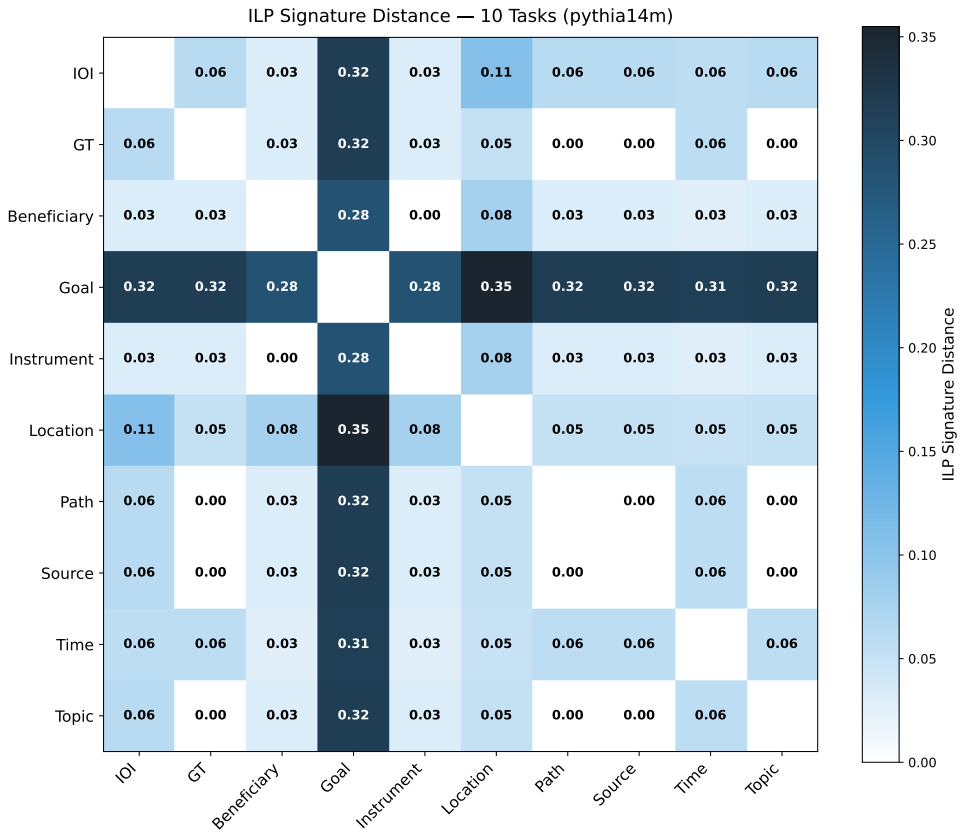


Figure 7: **ILP signature distance (Pythia-14M, 10 tasks).** Structural homogeneity dominates: most pairs cluster at $d \leq 0.06$, with many identical ($d = 0.00$). Goal is the sole outlier ($d = 0.28$–$0.36$), driven by its degenerate size-only clause (Table 6). No pair outside Goal clears $\delta_{\text{ILP}} = 0.30$.

### H.6 Live Transfer Results

We evaluate end-to-end transfer for two source→target pairs, both targeting Pythia-1B as the recipient: (1) Pythia-14M→Pythia-1B (cross-scale, same family, 70× parameter gap) and (2) LLaMA-3.2-1B→Pythia-1B (cross-family, different architecture and tokeniser). The transfer engine searches the Pythia-1B registry (10 tasks × 3 splits = 30 circuits) for candidates matching the source circuit's $\tau_{\text{arch}}$, then applies behavioural ($\theta_{\text{behav}} = 0.50$) and causal ($\theta_{\text{causal}} = 0.10$) validation. Table 15 shows that all 5 tasks are accepted for both transfer directions. Three patterns emerge:

**Task-correct transfer for IOI and GT.** Both IOI and GT find their exact task match in the target registry, despite having very different source $\Delta_T$ values ($-0.91$ vs. $5.24$ for IOI across the two sources). The selectivity is high: only 2/37–41 candidates pass for GT (both are GT circuits from different prompt splits), and 8/41 for IOI. GT candidates that fail are rejected at the behavioural threshold ($\theta_{\text{behav}} = 0.50$), confirming that the structural signature is discriminative.

**Family-correct transfer for binding tasks.** All three semantic role tasks (Location, Path, Time) select Goal as their best candidate, a role not present in the 5-task source pool. This is semantically valid: Goal implements binding via the same motif vocabulary (`mlp_heavy`, `early_attn_peak`), and its $\Delta_T = 1.00$ is the highest among binding circuits in Pythia-1B. The acceptance rate varies by task: Location accepts 22–25 candidates (structurally similar to many roles), while Time accepts only 3 (its clause is more restrictive). Rejected candidates fail primarily at the causal threshold, not the behavioural one.

**Identical outcomes across transfer directions.** The best accepted candidate and its $\Delta_T$ are identical for both source models, because the target registry is the same (Pythia-1B). What differs is the source $\Delta_T$: IOI's sign reversal ($-0.91$ in 14M vs. $5.24$ in LLaMA) and Time's negative source value in LLaMA ($-0.85$) confirm that the transfer engine correctly identifies structurally matched can-

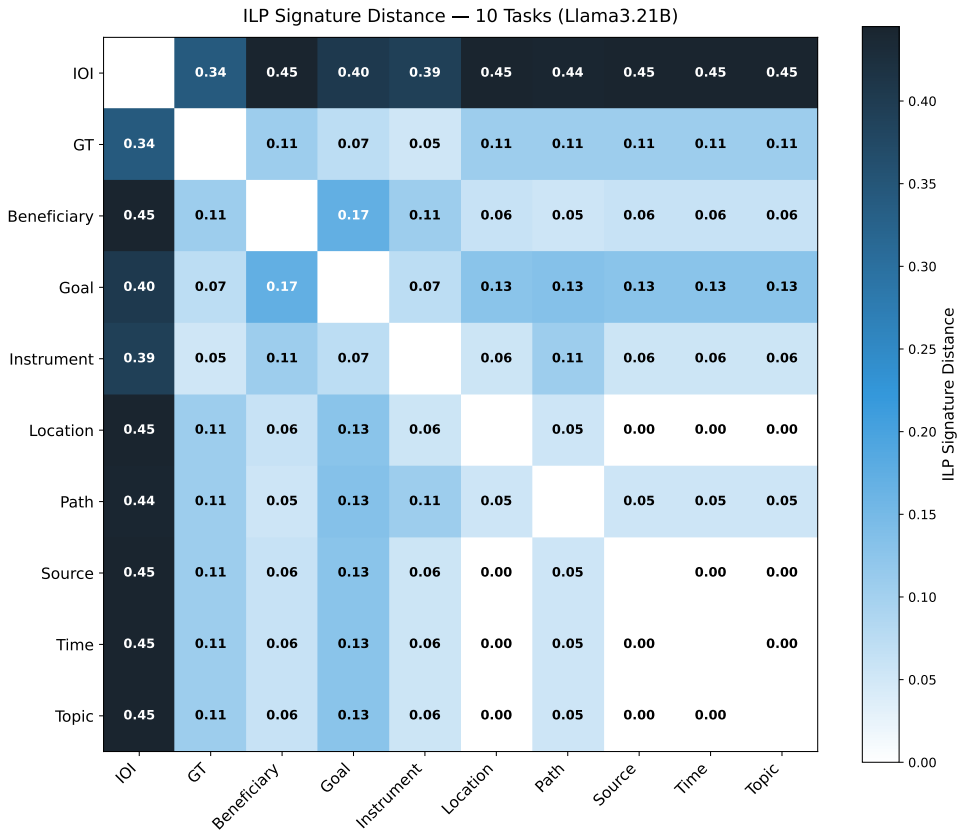


Figure 8: **ILP signature distance (LLaMA-3.2-1B, 10 tasks).** IOI is strongly separated from all others ($d = 0.34$–$0.45$). GT is intermediate ($d = 0.05$–$0.11$ from roles). Goal is distinctive ($d = 0.07$–$0.17$) owing to its unique `scaffold_event_entity` motif. Location, Time, Source, and Topic share identical signatures ($d = 0.00$), collapsing into one structural class.

didates regardless of source behaviour. The accept/reject pattern is architecture-independent.

# I Formal Algorithms

Algorithm 1 formalises the circuit characterisation pipeline described in Section 3: given a discovered circuit $C$ and a populated registry $\mathcal{R}$, it derives the logical identity $\mathcal{L}(C) = \langle \sigma, \tau_{\text{arch}}, \mathcal{S}_{\text{ILP}}, \kappa \rangle$ across four layers (provenance, structural predicates, causal functional signature, and ILP-learned architectural signature). Algorithm 2 formalises the transfer procedure described in Section 3: given a source logical identity and a target model $M_\beta$, it retrieves structurally compatible candidates from the registry, applies behavioural and causal screening, and falls back to full rediscovery if no candidate passes.

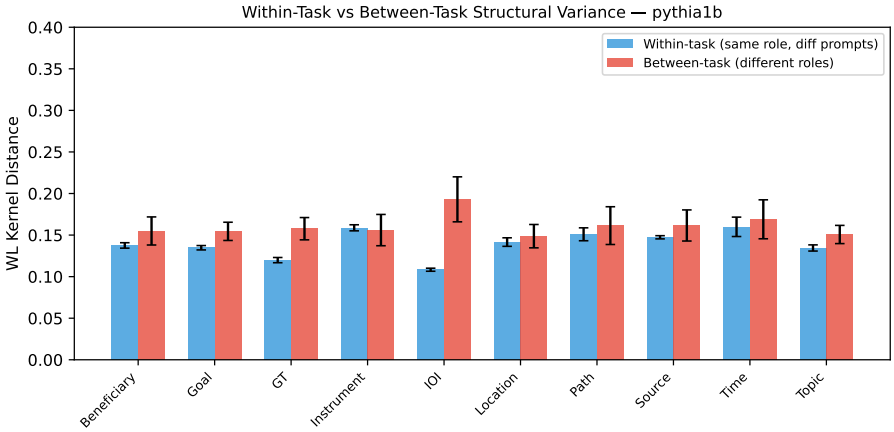


Figure 9: **Within-task vs. between-task structural variance (Pythia-1B).** Each bar pair shows the mean WL kernel distance among splits of the same task (blue, within) vs. to other tasks (red, between). IOI has the largest gap (1.8×), confirming its prompt-invariant circuit topology. Most semantic roles show ratios near 1.0, indicating that prompt subsets produce circuits as structurally different from each other as from other tasks, consistent with the lower ILP confidence for these roles.

| | | Pythia-14M → Pythia-1B | | | | | LLaMA-3.2-1B → Pythia-1B | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Task** | $\kappa$ | **Acc.** | **N** | **Best Match** | $\Delta_T^{\text{src}}$ | $\Delta_T^{\text{tgt}}$ | **Acc.** | **N** | **Best Match** | $\Delta_T^{\text{src}}$ | $\Delta_T^{\text{tgt}}$ |
| IOI | sel. | 8/41 | ✓ | IOI | −0.91 | 0.65 | 8/41 | ✓ | IOI | 5.24 | 0.65 |
| GT | comp. | 2/37 | ✓ | GT | 7.95 | 3.39 | 2/41 | ✓ | GT | 1.23 | 3.39 |
| Location | bind. | 22/29 | ✓ | Goal | 1.93 | 1.00 | 25/41 | ✓ | Goal | 1.20 | 1.00 |
| Path | bind. | 19/36 | ✓ | Goal | 0.53 | 0.81 | 19/41 | ✓ | Goal | 0.48 | 0.81 |
| Time | bind. | 3/32 | ✓ | Goal | 0.75 | 0.12 | 3/41 | ✓ | Goal | −0.85 | 0.12 |

Table 15: **Live transfer results.** Acc. = accepted/total candidates; N = task-correct match (✓); $\Delta_T^{\text{src}}$ = source circuit's causal contribution; $\Delta_T^{\text{tgt}}$ = best accepted candidate's causal contribution on target data. $\kappa$: sel. = selection, comp. = comparison, bind. = binding. § The target registry holds three prompt-split circuits per task; the best accepted IOI circuit captures a portion of the full Py-1B IOI computation ($\Delta_T = 4.37$, Table 4).

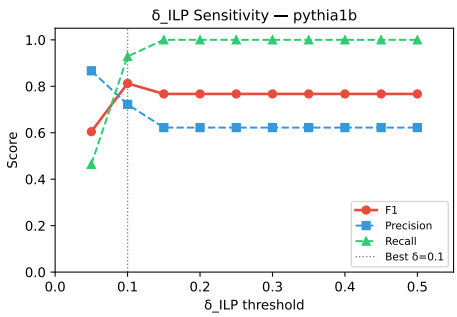


Figure 10: **$\delta_{\text{ILP}}$ sensitivity (Pythia-1B, 10 tasks).** F1 peaks at $\delta_{\text{ILP}} = 0.10$ (F1 = 0.81) with precision 0.72 and recall 0.93. The curve is stable across 0.10–0.15; beyond 0.15, all same-family pairs are accepted (recall = 1.0) but precision drops as cross-family pairs also fall below threshold.

**Algorithm 1** Circuit Characterisation Pipeline

**Input:** Circuit $C = (V_C, E_C)$, evaluation suite $\mathcal{D}_T$, circuit registry $\mathcal{R}$, background knowledge $\mathcal{B}$
**Output:** Logical identity $\mathcal{L}(C) = \langle \sigma, \tau_{\text{arch}}, \mathcal{S}_{\text{ILP}}, \kappa \rangle$

**Layer 0 — Provenance**
Record $\langle \texttt{task}, \texttt{model}, \texttt{discovery_method} \rangle$ as ground facts prov$(C)$
**Layer 1 — Structure and named motifs**
**for** each component $v \in V_C$ at layer $\ell$ **do**
  Assert $\texttt{node}(C, v)$, $\texttt{type}(v, \tau)$, $\texttt{layer}(v, \ell)$
**end for**
Compute and assert $\texttt{layer_span}(C, \ell_{\min}, \ell_{\max})$, $\texttt{component_ratio}(C, \tau, r)$, $\texttt{rel_size}(C, s)$
Extract named motifs $\mathcal{M}(C) \subseteq \{\texttt{attn_chain_3}, \texttt{mlp_heavy}, \ldots\}$; assert $\texttt{has_motif}(C, m)$ for each $m \in \mathcal{M}(C)$
Store Layer 1 facts in $\mathcal{B}$
**Layer 2 — Causal Functional Signature (CFS)**
**for** each node $v \in V_C$ **do**
  $\delta_v \leftarrow \text{DLA}(v, \mathcal{D}_T)$
  **if** $|\delta_v| < \epsilon_{\text{DLA}}$ **then**
    Flag $v$ as causally marginal
  **else**
    Compute attribution-weighted attention profile $\pi_v$
    Label positions with linguistic roles $\pi_v^{\text{ling}}$ and task roles $\pi_v^{\text{task}}$
  **end if**
**end for**
$\sigma \leftarrow \{\langle v, \delta_v, \pi_v^{\text{ling}}, \pi_v^{\text{task}} \rangle : v \in V_C,\ |\delta_v| \geq \epsilon_{\text{DLA}}\}$
Store Layer 2 CFS facts in $\mathcal{B}$
**Layer 3 — ILP$_{\text{arch}}$: learning $\tau_{\text{arch}}$**
$E^+ \leftarrow \{C' \in \mathcal{R} : \text{prov}(C').\texttt{task} = \text{prov}(C).\texttt{task}\}$
$E^- \leftarrow \{C' \in \mathcal{R} : \text{prov}(C').\texttt{task} \neq \text{prov}(C).\texttt{task}\}$
Rank named motifs: $\textsc{attr_ig}(m) \leftarrow \text{IG}(m) \times (1 + \bar{s}_m)$
$\text{Pool}_A \leftarrow$ hypothesis space over $\texttt{has_motif}$ predicates *named motifs*
$\text{Pool}_B \leftarrow$ hypothesis space over $\texttt{edge}$, $\texttt{type}$, $\texttt{layer}$, $\texttt{component_ratio}$, $\texttt{layer_span}$, $\texttt{skip_connections}$, $\texttt{hub_count}$ *blind structural predicates; no* $\texttt{has_motif}$
$\mathcal{H}_A \leftarrow \text{ILP}(E^+, E^-, \mathcal{B}, \text{Pool}_A)$
$\mathcal{H}_B \leftarrow \text{ILP}(E^+, E^-, \mathcal{B}, \text{Pool}_B)$
$\tau_{\text{arch}} \leftarrow \arg\max_{\gamma \in \mathcal{H}_A \cup \mathcal{H}_B} \text{F1}(\gamma)$, breaking ties by $|\gamma|$
$\mathcal{S}_{\text{ILP}} \leftarrow \langle n_{\text{clauses}}, \bar{d}, \mathcal{P}_H, \bar{\ell}, \texttt{complexity} \rangle$
**Validation**
Compute $\text{Acc}_T(C)$ and $\Delta_T(C)$ on $\mathcal{D}_T$
$\kappa \leftarrow \langle \text{Acc}_T(C), \Delta_T(C), \text{is_minimal}(C) \rangle$
Register $\mathcal{L}(C) = \langle \sigma, \tau_{\text{arch}}, \mathcal{S}_{\text{ILP}}, \kappa \rangle$ in $\mathcal{R}$
**return** $\mathcal{L}(C)$

**Algorithm 2** Mechanistic Knowledge Transfer

**Input:** Source identity $\mathcal{L}(C_\alpha) = \langle \sigma_\alpha, \tau_{\text{arch},\alpha}, \mathcal{S}_\alpha, \kappa_\alpha \rangle$; target model $M_\beta$ with depth $L_\beta$; registry $\mathcal{R}$; evaluation suite $\mathcal{D}_T^\beta$; thresholds $\theta_{\text{behav}}, \theta_{\text{causal}}, \delta_{\text{ILP}}$
**Output:** Accepted candidate $C^*$ in $M_\beta$, or $\varnothing$

**// Step 1: Retrieve source signature**
Retrieve $\tau_{\text{arch},\alpha}$ and $\mathcal{S}_\alpha$ from $\mathcal{R}$
**// Step 2: Identify structural candidates**
$\delta_\rho \leftarrow 1/L_\beta$ *one-layer depth tolerance, scales with model*
$\mathcal{C}_{\text{cand}} \leftarrow \varnothing$
**for** each stored circuit $C_\beta \in \mathcal{R}[M_\beta]$ **do**
  **if** $\text{arch_type}(\tau_{\text{arch},\beta}) = \text{arch_type}(\tau_{\text{arch},\alpha})$ **and** $|\text{depth_mean}(C_\beta) - \text{depth_mean}(C_\alpha)| \leq \delta_\rho$ **and** $|\text{rel_size}(C_\beta) - \text{rel_size}(C_\alpha)| \leq \delta_{\text{size}}$ **then**
    $\mathcal{C}_{\text{cand}} \leftarrow \mathcal{C}_{\text{cand}} \cup \{C_\beta\}$
  **end if**
**end for**
**// Step 3: Behavioural and causal screening**
$\mathcal{C}_{\text{pass}} \leftarrow \varnothing$
**for** each candidate $C_\beta \in \mathcal{C}_{\text{cand}}$ **do**
  **if** $\text{Acc}_T(C_\beta) \geq \theta_{\text{behav}}$ **and** $\Delta_T(C_\beta) \geq \theta_{\text{causal}}$ **then**
    $\mathcal{C}_{\text{pass}} \leftarrow \mathcal{C}_{\text{pass}} \cup \{C_\beta\}$
  **end if**
**end for**
**// Step 4: Select best candidate**
**if** $\mathcal{C}_{\text{pass}} \neq \varnothing$ **then**
  $C^* \leftarrow \arg\max_{C_\beta \in \mathcal{C}_{\text{pass}}} \Delta_T(C_\beta)$ **return** $C^*$
**end if**
**// Fallback: full rediscovery with compatibility check**
$C' \leftarrow$ circuit discovery on $M_\beta$ using $\mathcal{D}_T^\beta$
Compute $\mathcal{L}(C')$ via Algorithm 1
**if** $d(\mathcal{S}_{\text{ILP}}(\tau'), \mathcal{S}_{\text{ILP}}(\tau_{\text{arch},\alpha})) < \delta_{\text{ILP}}$ **then return** $C'$ *compatible mechanism found via rediscovery*
**elsereturn** $\varnothing$ *mechanisms are structurally incompatible*
**end if**