BUCKETING THE GOOD APPLES: A METHOD FOR DIAGNOSING AND IMPROVING CAUSAL ABSTRACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a method for diagnosing interpretation in neural networks by identifying an input subspace where a proposed interpretation is highly faithful. Our method is particularly useful for causal-abstraction-style interpretability, where a high-level causal hypothesis is evaluated by interchange interventions. Rather than treating interchange intervention accuracy as a single global summary, we refine this framework by partitioning the input space into well-interpreted and under-interpreted regions according to pairwise interchange-intervention behavior. This turns causal abstraction from a purely global evaluation into a more diagnostic tool: it not only measures whether an interpretation works, but also reveals where it works, where it fails, and what distinguishes the two cases. This diagnostic view also provides practical heuristics for improving interpretations. By analyzing the structure of the well-interpreted and under-interpreted regions, we can identify missing distinctions in a high-level hypothesis, and discover previously unmodeled intermediate variables. We instantiate this idea as a simple four-step recipe and show that it yields informative error analyses across multiple causal abstraction settings. In a logic inference task, recursively applying the recipe recovers the intermediate variables of the high-level hypothesis, starting from the output variable alone. More broadly, our results suggest that partitioning the input space is a useful step toward more precise, constructive, and scalable mechanistic interpretability.

1 INTRODUCTION

As Language Models (LMs) evolve, the mechanistic interpretability community has developed various tools for studying the representations and algorithms underlying model behavior, including gradient-based attribution (Sundararajan et al., 2017), activation-based localization (Meng et al., 2022a;b), and feature decomposition with Sparse Autoencoders (SAEs) (Bricken et al., 2023; Cunningham et al., 2023). Among these paradigms, causal abstraction provides a particularly rigorous framework: given a task, a high-level causal model and an alignment between high-level variables and internal neural representations, it asks whether the model and the hypothesis agree under counterfactual interventions (Geiger et al., 2021; 2025), and it has been applied to a wide variety of tasks (Wu et al., 2024a; Arora et al., 2024; Huang et al., 2025; Boguraev et al., 2025).

In practice, causal abstraction is usually evaluated through *interchange interventions*, summarized by a single scalar metric, *interchange intervention accuracy* (IIA) (Geiger et al., 2021; 2025). The IIA score provides a simple metric to measure how well hypotheses align with neural models, but it says little about *where* a hypothesis is faithful, and intermediate scores are especially hard to interpret (Makelov et al., 2023; Wu et al., 2024b; Méloux et al., 2025). Moreover, although methods such as DAS make it easier to find candidate alignments, the overall workflow remains largely evaluative rather than constructive: it can tell us that a hypothesis is imperfect, but not how to improve it.

We address this gap by shifting the unit of analysis from a global score to the structure of the input space. Given a low-level model $\mathcal{L}$, a high-level hypothesis $\mathcal{H}$, and an alignment Π, we partition the input space into well-interpreted *target buckets* with high IIA and a complementary bucket that captures the remaining failure modes. The key question is no longer just whether an abstraction works on average, but *for which inputs* it is actually faithful. This yields a more informative diagnosis

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

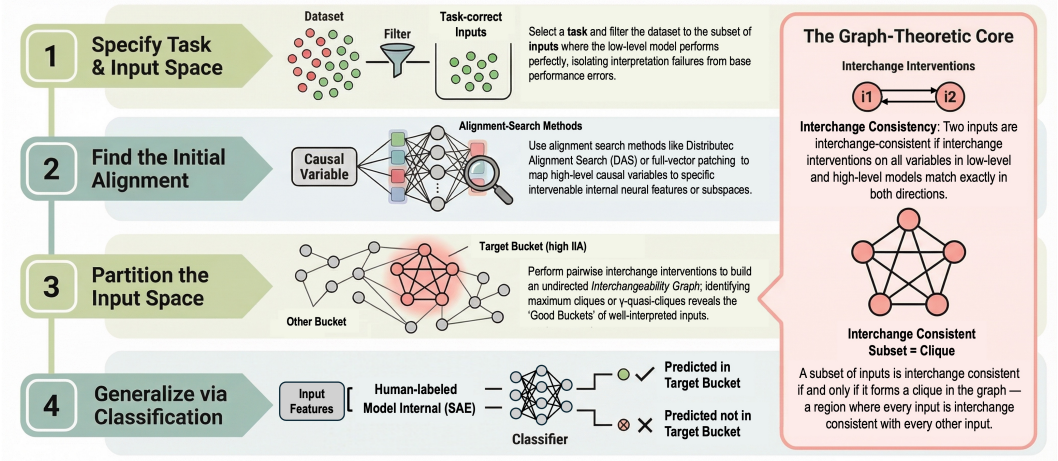

Figure 1: **Four-step interpretation diagnosis pipeline.** Given a causal abstraction for a task-performing model, we focus on task-correct inputs, identify an alignment, partition the input space by pairwise interchangeability, and train a classifier to generalize the diagnosis.

and turns abstraction failures into evidence for refinement. That an abstraction may hold only on part of the input space has been noted before—Geiger et al. (2021) define local abstractions through cliques of an intervention-success graph, Pîslar et al. (2025) combine partial causal models over input partitions, and causal feature learning exploits partitions of observational and interventional data (Chalupka et al., 2014; 2017)—but it has not been developed into a procedure for diagnosing and refining a hypothesis. We do so by generalizing bucket membership to unseen inputs and reading the bucket boundary as evidence for what the hypothesis is missing.

Our main contribution is a practical four-step pipeline for diagnosing and improving causal abstractions: specify a reliable task and input space, obtain a candidate alignment, partition the input space by pairwise interchangeability to identify nearly interchange-consistent subsets, and train a classifier that extends this diagnosis to the rest of the input space. without further interventions. to generalize this diagnosis beyond the analyzed sample. Across fine-tuned and pretrained models of different sizes, and across alignment methods including full-vector patching, DAS, and MDAS, we show that this bucketing procedure is broadly useful for diagnosing causal abstractions. Empirically, we validate it on logic, entity binding (Gur-Arieh et al., 2026), and factual recall (Geva et al., 2021; Hernandez et al., 2023; Geva et al., 2023; Huang et al., 2024). In the logic inference task, recursively applying the recipe supports iterative refinement of the high-level hypothesis itself, showing how causal abstraction can move from post hoc evaluation toward constructive hypothesis discovery.

The rest of the paper is organized as follows. Section 2 reviews the causal abstraction framework and the alignment and feature-learning tools used in this work. Section 3 formalizes interchange-consistent subsets and introduces our diagnosis pipeline. Section 4 evaluates the method on three settings of increasing complexity, and Section 5 concludes with limitations and future directions.

2 Preliminaries

Causal abstraction for LMs involves hypothesizing and verifying that a higher-level causal model is an abstraction of the lower-level LM. Causal models (Pearl 2009, Bareinboim et al. 2022) comprise variables, on which interventions predict counterfactual behavior. After hypothesizing alignments of the causal variables within LM representations and verifying the predicted counterfactual behavior for all such representations of an LM with appropriate interventions, the higher-level model can be taken to causally abstract the LM, if the corresponding interventions on the corresponding causal variables predict the same counterfactual output. We adopt notation from Geiger et al. (2023).

Causal Abstraction. Specifically, we use $\mathcal{H}, \mathcal{L}$ to represent a high-level and a low-level causal model, respectively. For each variable X, we notate its domain as $\mathbb{V}_X$. For X in $\mathcal{H}$, (π_X, τ_X) is an alignment that pairs a high-level variable X with a set of low-level variables π_X and maps values of

π_X to values of X, i.e., $\tau_X : \prod_{Z \in \pi_X} \mathbb{V}_Z \to \mathbb{V}_X$. Moreover, we define an alignment Π of two models $\mathcal{H}, \mathcal{L}$ as a collection of variable alignments, i.e., $\Pi = (\{\pi_X\}, \{\tau_X\})$. Given a causal model $\mathcal{M}$, we define $\mathcal{M}_{V \leftarrow v}(x)$ to be the output of $\mathcal{M}$ with input x after setting $V = v$. For $\mathcal{M}$, a set of source inputs $\{s_k\}$, and a set of intermediate variables $\{V_k\}$, the interchange intervention (*II*) is defined as:

$$II(\mathcal{M}, \{s_k\}, \{V_k\})(b) = \mathcal{M}_{V_k \leftarrow \text{GetVals}_{V_k}(\mathcal{M}(s_k))}(b),$$

where $\text{GetVals}_{V_k}(\mathcal{M}(s_k))$ is the value of variables V_k in model $\mathcal{M}(s_k)$. Similarly, one can define *II* with multiple sources as in Geiger et al. (2023; 2024). Ideally, we want the alignment Π to satisfy

$$\tau(II(\mathcal{L}, s, \pi_X)(b)) = II(\mathcal{H}, \tau(s), X)(\tau(b)).$$

Distributed Alignment Search (DAS). Standard-basis interventions implicitly assume that a high-level variable is localized in a particular set of neurons, but neural representations are often distributed across overlapping directions (Smolensky, 1986; Olah et al., 2020; Scherlis et al., 2022; Geiger et al., 2024). Distributed Alignment Search (DAS) addresses this by replacing "localist" alignment search with gradient-based search over distributed linear subspaces. Concretely, for a high-level variable X and a candidate low-level representation $\mathbf{h}_{\pi_X} \in \mathbb{R}^n$, DAS learns an orthogonal matrix R_θ and aligns X with a k-dimensional subspace of the rotated representation $R_\theta(\mathbf{h}_{\pi_X})$, rather than with coordinates in the standard basis (Geiger et al., 2024). A distributed interchange intervention rotates the base and source representations, swaps the aligned subspace from source into base, and rotates back before continuing the forward pass. The low-level and high-level models are kept fixed; only the alignment parameters are trained so that the intervened low-level output matches the high-level intervention on X.

Sparse Autoencoders (SAEs). SAEs (Bricken et al., 2023; Cunningham et al., 2023) decompose dense, polysemantic activations (Elhage et al., 2022) into interpretable components by mapping an activation vector $x \in \mathbb{R}^d$ to a high-dimensional, sparse feature space $f(x) \in \mathbb{R}^m$ (where $m \gg d$) via an encoder $f(x) = \text{ReLU}(W_{enc}x + b_{enc})$, such that the original activation can be reconstructed as $\hat{x} = W_{dec}f(x) + b_{dec}$. By training with an L_1 penalty to enforce sparsity, SAEs identify features that often correspond to discrete semantic or syntactic concepts (Huang et al., 2024). In our diagnostic pipeline, we use SAEs to extract model-internal features for our classifiers.

3 Finding Interchange-Consistent Subsets

Given a causal abstraction candidate $(\mathcal{L}, \mathcal{H}, \Pi)$ and an input space $\mathcal{I}$, we aim to identify regions of the input space on which the abstraction is faithful. Rather than summarizing abstraction quality with a single IIA score, we seek a more structured view: which inputs are well interpreted by the current hypothesis, which are not, and how this distinction can be used to improve the hypothesis itself. This section proceeds in two parts. Section 3.1 formalizes interchange consistency at the level of input pairs and input subsets, yielding a graph-theoretic view of abstraction success and failure (Geiger et al., 2020; Pîslar et al., 2025). Section 3.2 introduces a practical four-step diagnosis pipeline based on this structure to partition the input space and generalize the result beyond the analyzed sample.

3.1 Interchange-Consistent Input Subset and the Interchangeability Graph

Causal abstractions are typically evaluated using interchange intervention accuracy (IIA). Given a causal abstraction $(\mathcal{L}, \mathcal{H}, \Pi)$ and a finite set of input pairs $\mathcal{P}$, IIA is the proportion of pairs $(i_1, i_2) \in \mathcal{P}$ for which the low-level intervention matches the corresponding high-level counterfactual for every variable in $\mathcal{H}$ (i.e., $X \in \mathsf{Var}(\mathcal{H})$):

$$\tau(\Pi(\mathcal{L}, i_1, \pi_X)(i_2)) = \Pi(\mathcal{H}, \tau(i_1), X)(\tau(i_2))\,.$$

While IIA is a useful global summary,[1] it does not reveal how abstraction failures are distributed over the input space. To move beyond a single scalar score, we define a pairwise notion of exact success under interchange interventions and then lift it to subsets of inputs.

[1] IIA is one of several ways to evaluate an alignment (see, e.g., Zennaro et al., 2023); because our partition is defined on the outcome of each intervened pair, any pairwise success criterion could take its place.

Definition 1 (Interchange-Consistent Input Pair)**.** *Given a causal abstraction* $(\mathcal{L}, \mathcal{H}, \Pi)$ *and an input set* $\mathcal{I}$*, two inputs* $i_1, i_2 \in \mathcal{I}$ *are* interchange-consistent *under* $(\mathcal{L}, \mathcal{H}, \Pi)$*, denoted* $\mathbf{p}_{(\mathcal{H},\mathcal{L},\Pi)}\langle i_1, i_2\rangle$*, iff for all* $X \in \mathsf{Var}(\mathcal{H})$*,*

$$\tau(\mathit{II}(\mathcal{L}, i_1, \pi_X)(i_2)) = \mathit{II}(\mathcal{H}, \tau(i_1), X)(\tau(i_2)),$$
$$\tau(\mathit{II}(\mathcal{L}, i_2, \pi_X)(i_1)) = \mathit{II}(\mathcal{H}, \tau(i_2), X)(\tau(i_1)).$$

Definition 1 requires exact counterfactual agreement in both directions: patching from i_1 into i_2, and from i_2 into i_1. We then lift this pairwise notion to subsets.

Definition 2 (Interchange-Consistent Input Subset)**.** *An input subset* $I \subseteq \mathcal{I}$ *is* interchange-consistent *by* $(\mathcal{L}, \mathcal{H}, \Pi)$ *iff for all* $i_1, i_2 \in I$*,* $\mathbf{p}_{(\mathcal{H},\mathcal{L},\Pi)}\langle i_1, i_2\rangle$ *holds.*

Since we aim to find (quasi-)interchange-consistent input subsets[2] we refer to such subsets as *target buckets*. This setwise notion naturally admits a graph-theoretic reformulation.

Definition 3 (Interchangeability Graph)**.** *Given a causal abstraction* $(\mathcal{L}, \mathcal{H}, \Pi)$ *and an input set* $\mathcal{I}$*, the* interchangeability graph *is an undirected graph* $G = (V, E)$ *with* $V = \mathcal{I}$*. An edge connects two distinct vertices* $i_1, i_2 \in V$ *iff* $\mathbf{p}_{(\mathcal{H},\mathcal{L},\Pi)}\langle i_1, i_2\rangle$ *holds.*

The interchangeability graph provides a map of where a candidate abstraction succeeds. A subset $I \subseteq \mathcal{I}$ is interchange-consistent iff every pair of vertices in I is connected by an edge; in graph-theoretic terms, such a subset is a clique, and the search for an exactly faithful region of the input space becomes the search for a (maximum) clique in the interchangeability graph.

3.2 The Diagnosis Pipeline

We present a four-step diagnosis pipeline for turning a non-perfect causal abstraction into a more informative object of analysis. The pipeline is task-agnostic, but to make each step concrete we illustrate it with a running logic-inference example (Fig. 2), here for a *single* diagnosis pass on one target variable; Section 4.1 reapplies the pipeline recursively to recover a fuller hypothesis. The complete diagnosis algorithm is shown in Algorithm 1 in the appendix.

Step 1: Specify the task and the input space. The first step of the pipeline is to define a task and an input space $\mathcal{I}$ on which the low-level model $\mathcal{L}$ performs reliably. This isolates failures of *interpretation* from ordinary task failures. In practice, we therefore restrict attention to the subset of correct predictions: $\mathcal{I}_{\text{correct}} = \{\, i \in \mathcal{I} : \mathcal{L}(i) \text{ is task-correct} \,\}$.

Example: In the logic inference task, we fine-tuned the model on a generated dataset of size $20,000$, where o_1, o_2, o_3 are each true with probability 0.5. The model achieves 99.7% test accuracy. We filtered out all the failure cases, and the input space consists of all instances where the model correctly predicts the truth value of o_5.

Step 2: Find a candidate alignment. Next, given a high-level variable X, we identify a candidate alignment π_X between X and an intervenable low-level feature. This can be done using any standard alignment-search method, such as DAS or full-vector patching. At this stage, global IIA provides only a preliminary signal: if it is near-perfect, the hypothesis may already be adequate; if it is at chance, the candidate alignment is likely uninformative; but if it is intermediate, the result will need some further investigation.

Example: We deliberately begin from an underspecified high-level hypothesis containing only the final output variable o_5. DAS finds a near-perfect alignment at the final output position, but also an earlier candidate alignment at position 77, layer 7, with IIA of about 0.8 on the DAS evaluation set (0.68 over all ordered pairs of the 200 inputs that form the interchangeability graph; Table 1). This is exactly the kind of intermediate result that motivates diagnosis: above chance, yet not near-perfect.

Step 3: Bucket the input space by interchangeability. We then ask whether a non-perfect alignment is faithful on some *subset* of the input space. To answer this, we construct the interchangeability graph over $\mathcal{I}_{\text{correct}}$, where two inputs are connected when they are interchange-consistent under the

[2] In practice, we usually search for quasi-interchange-consistent or γ-interchange-consistent input subsets rather than strict interchange-consistent subsets. A subset $S \subseteq \mathcal{I}$ is γ-interchange-consistent if at least a γ proportion of input pairs in S are interchange-consistent. This relaxation reduces computational cost and avoids returning only trivially small subsets when a few failed interventions break an otherwise high-faithfulness region.

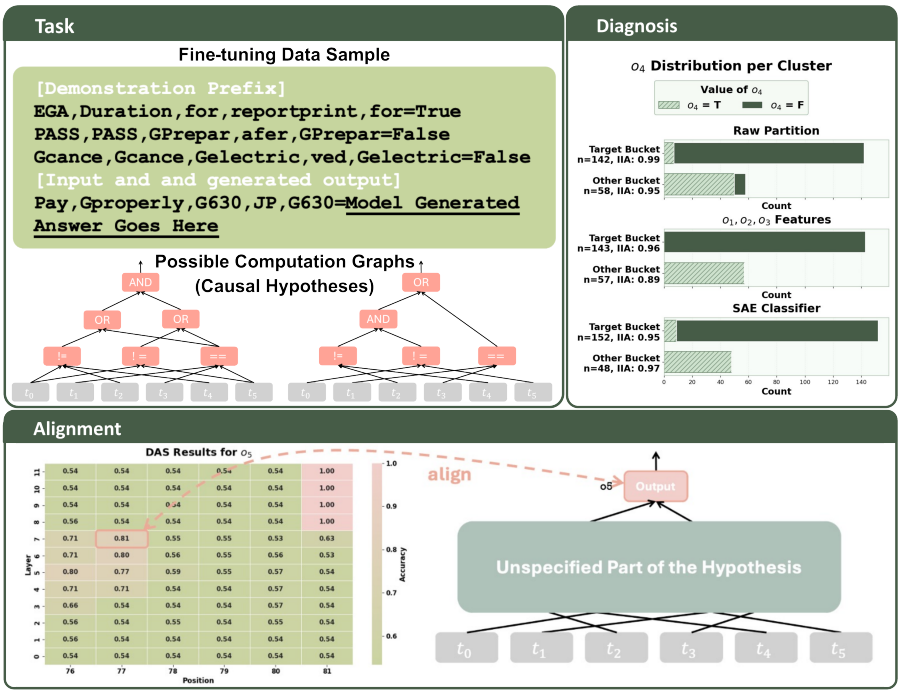

Figure 2: **Running example – the logic inference task.** We fine-tune a 12-layer GPT-2-small model to predict the truth value of $o_5 = ((t_2 \neq t_4) \wedge (t_0 \neq t_5)) \vee (t_1 = t_3)$. For exposition, we denote the primitive Boolean variables (the (non)equalities) by o_1, o_2, o_3, so that the target computation can be written as $o_5 = (o_1 \wedge o_2) \vee o_3$. The model is trained only on input-output pairs, so it may implement either $(o_1 \wedge o_2) \vee o_3$ or the equivalent $(o_1 \vee o_3) \wedge (o_2 \vee o_3)$ to get the same correct result; further tests are needed to tell which computation it actually uses when performing task.

candidate abstraction. In practice we relax exact cliques to γ-quasi-cliques, subgraphs of density at least γ ($\gamma = 0.98$ in all experiments), call the dense quasi-clique the "target bucket" and the remainder the "other bucket" (algorithm in Appendix A). The threshold is a tunable hyperparameter: sweeping $\gamma \in [0.80, 1.00]$ trades coverage for purity but leaves the diagnosis unchanged in every experiment (Appendix A.1).

Example: Applying this step to the candidate o_5 alignment reveals that the alignment is not merely noisy. The well-interpreted bucket is dominated by inputs for which $o_1 \wedge o_2 = \text{False}$, so that the target computation simplifies to $o_5 = o_3$. On this subspace, a feature that tracks only the o_3 branch can still appear faithful to the full output variable. By contrast, the under-interpreted bucket is dominated by inputs for which $o_1 \wedge o_2 = \text{True}$, where that simplification no longer holds. The diagnosis therefore localizes a specific failure mode of the coarse output-only hypothesis.

Step 4: Generalize and characterize the partition. Finally, we train a classifier $g : \mathcal{I} \to \{1, \ldots, K\}$ to predict bucket membership for unseen inputs. Its input is a feature representation of the input—hand-labeled features, or model-internal features such as SAE activations at the aligned site—and its output is a bucket label. This step serves two purposes: it tests whether the discovered buckets reflect a genuine structural boundary rather than an artifact of a finite intervention graph, and it helps characterize that boundary in terms of the features that best predict membership. This final step makes the recipe scale: once the interchangeability graph is built on a small sample of inputs (200–512 in our experiments), the classifier can assign bucket membership to the rest of the input space without any further interventions.

Example: In the logic inference task, after bucketing the input space, we train two classifiers using bucket membership as the label. One takes the truth values of o_1, o_2, o_3 as hand-labeled input features, and the other takes SAE features at the aligned position as model-internal input features. Both predict the membership of unseen inputs and largely agree, indicating a stable structural distinction: membership is governed by whether the conjunction branch $o_4 = o_1 \wedge o_2$ is active.

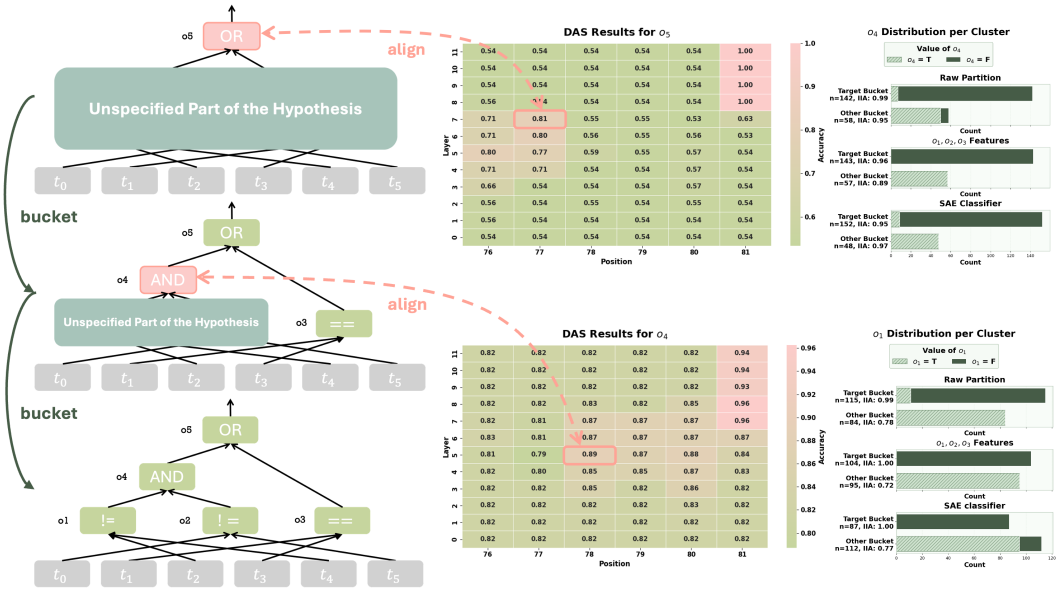

Figure 3: **Recursive hypothesis discovery in the logic inference task. Top:** Diagnosing the candidate o_5 alignment at position 77, layer 7 partitions the input space into two high-IIA buckets separated by the latent variable $o_4 = o_1 \wedge o_2$. **Middle:** After promoting o_4 to an explicit variable, DAS identifies a near-perfect signal at position 81, layer 7 and a non-trivial earlier signal at position 78, layer 5. **Bottom:** Diagnosing the earlier o_4 signal reveals o_1 as the next missing component, recovering the complete hierarchy.

Taken together, the four steps turn a single imperfect IIA score into a structured diagnosis (Fig. 2). By deliberately aligning the output variable to a location with an imperfect but above-baseline IIA, we identify buckets with high in-bucket and low cross-bucket IIA: the alignment is locally faithful within each bucket but collapses a distinction across buckets, indicating that the output variable can be broken down into two intermediate variables.

4 Experiments

We evaluate the diagnosis recipe in three settings of increasing complexity: a synthetic logic task, an entity-binding task adapted from prior work (Gur-Arieh et al., 2026), and an entangled factual recall setting based on the RAVEL benchmark (Huang et al., 2024). In each case, we begin with an existing causal abstraction obtained using a standard alignment method, observe that its global IIA is informative but non-perfect, and then apply our partitioning procedure to identify buckets with high within-bucket and low cross-bucket IIA. Throughout, we also compare the recovered buckets against a purely geometric baseline, PCA + K-means on the aligned-site activations (Appendix A.3), by which the recovered partitions are often noisier. We finally test whether these partitions generalize beyond the analyzed sample and whether they suggest useful refinements to the original abstraction.

4.1 Recursive Hypothesis Discovery in the Logic Inference Task

Section 3.2 established the first diagnosis pass for the logic inference task. Aligning o_5 to position 77, layer 7 partitions the input space into two buckets, both of which have high within-bucket IIA but low cross-bucket IIA. The partition has a clear semantic interpretation: in the target bucket, $o_4 = o_1 \wedge o_2$ is almost always `False`, whereas in the other bucket, o_4 is predominantly `True`. The candidate o_5 alignment thus separates two regimes of computation rather than representing the output uniformly, suggesting that o_5 be refined into o_4 and o_3.

We then apply the same diagnosis procedure to o_4. DAS identifies a near-perfect signal for o_4 at position 81, layer 7, indicating that the completed o_4 computation can be localized there, and a non-trivial earlier signal at position 78, layer 5. We align o_4 to the earlier site and repeat the bucketing step: the partition again yields two buckets with high within-bucket IIA, and in the target bucket o_1 is

predominantly `False`, so the earlier o_4 signal can itself be decomposed into the primitives o_1 and o_2. Classifiers on hand-labeled and SAE features generalize the bucket assignments and largely agree.

Taken together, these two passes recover the hierarchy $o_1, o_2, o_3 \to o_4 \to o_5$. Starting from the output variable alone, recursive diagnosis spells out the full high-level causal hypothesis from top to bottom. It demonstrates that the model solves the task via the intended factorization $y = (o_1 \wedge o_2) \vee o_3$, rather than the equivalent alternative $y = (o_1 \vee o_3) \wedge (o_2 \vee o_3)$. Consistent with this picture, alignment search over all variables localizes o_3 at position 77, layer 7, and o_1 at position 78, layer 5 (Appendix B).

4.2 Entity Binding Task

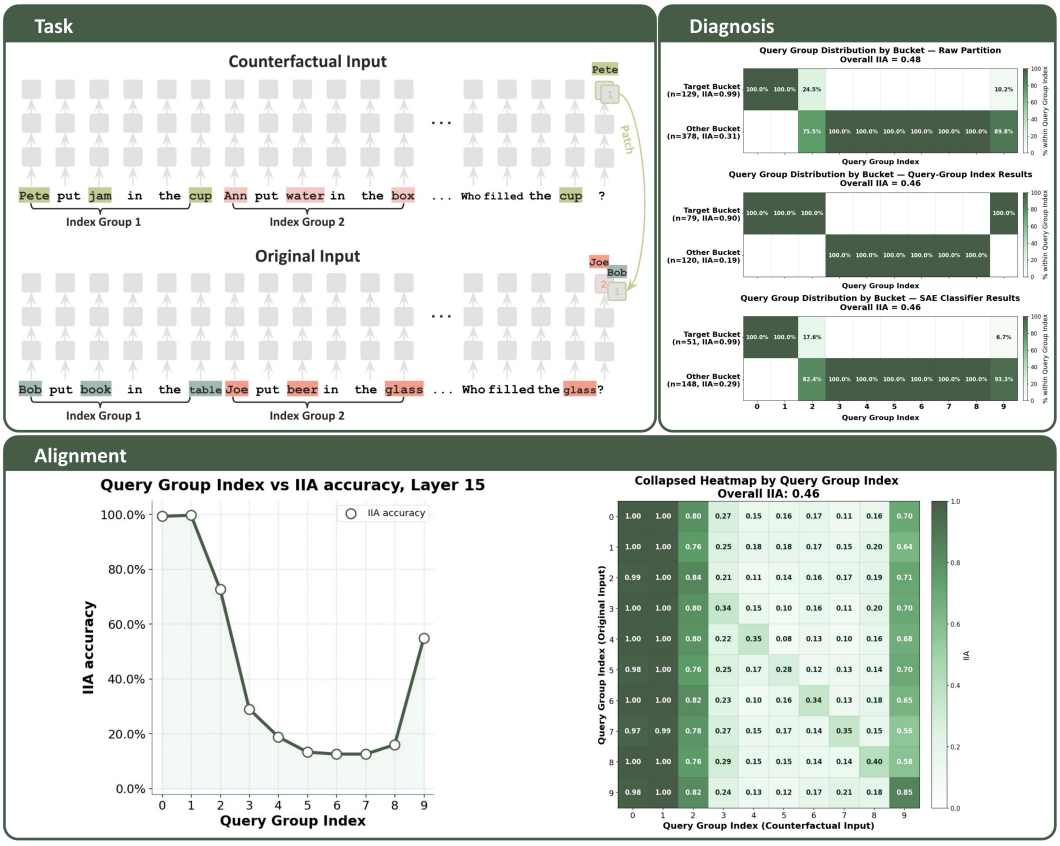

Figure 4: **Diagnosis of the Entity Binding Task in Gemma-2-2B-Instruct. Top Left:** The entity-binding task evaluates in-context retrieval from sequences of groups (e.g., "John fills a cup with beer..."). A query (e.g., "Who filled a cup?") then requests an entity from a specific group. We test a *positional hypothesis*: retrieval is mediated by a high-level variable q_{group} representing the queried group's context position. **Bottom:** Preliminary evaluation via full-vector patching at layer 15 reveals a U-shaped alignment curve as in Gur-Arieh et al. (2026). **Top Right:** Bucketing the interchangeability graph recovers the failure mode automatically: the target bucket is dominated by edge groups ($q_{group} \in \{0, 1, 2, 9\}$), while the other bucket contains medial groups ($q_{group} \in \{3, 4, 5, 6, 7, 8\}$).

Task and input space. We evaluate our framework using the entity-binding task (Geiger et al., 2020; Gur-Arieh et al., 2026), where a model must retrieve specific entities from a sequence of templatic *entity groups*. We test the *positional hypothesis*—the idea that retrieval is mediated by a high-level variable q_{group} representing the context position of the queried group—using the `filling_liquids` task with `Gemma-2-2B-Instruct`. We diagnose the model's positional mechanism by analysing an input space $\mathcal{I}$ consists of the correctly answered 10-group prompts (task accuracy exceeds 98%), annotated by the queried group position (Fig. 4).

Alignment and Preliminary Evaluation. We replicate the positional-hypothesis experiment from Gur-Arieh et al. (2026), defining a high-level model where a single variable, q_{group}, represents the

queried group's position. We localize the corresponding low-level site and evaluate the hypothesis via full-vector patching at the final query position. Our preliminary results at layer 15 reveal a sharply non-uniform alignment curve, shown in Fig. 4, successfully reproducing the U-shape reported by Gur-Arieh et al. (2026). The positional mechanism is highly faithful for entity groups at the sequence's edges but degrades significantly for those in the middle. The global IIA is neither near-perfect nor near-chance, so the scalar summary alone does not say where the interpretation holds.

Bucketing the Input Space. We apply our partitioning procedure to the positional abstraction $(\mathcal{L}, \mathcal{H}_{\text{pos}}, \Pi)$. The overall IIA of the interchangeability graph is 48%, signaling that the abstraction is not uniformly faithful across the entire input space. Upon partitioning this graph, a clear semantic pattern emerges (Fig. 4, top right). The well-interpreted bucket is dominated by inputs where the queried group is located at the start or end of the sequence, while the other bucket is dominated by inputs in the middle. This recovers, directly from the structure of the intervention graph and without inspecting per-index curves, the failure mode identified manually by Gur-Arieh et al. (2026).

Generalization through classifier. To test generalization beyond the graph, we train classifiers on the queried group index and on internal SAE features to predict bucket membership for unseen in-distribution inputs. The index-based classifier recovers an interpretable rule—assigning $q_{\text{group}} \in \{0, 1, 2, 9\}$ to the target bucket and $\{3, \ldots, 8\}$ to the other bucket—attaining 90% IIA with 81% density (Fig. 4, middle right). The SAE-based classifier further improves this to 98% IIA at 97% density (Fig. 4, bottom right).[3] That internal features outperform hand-labeled features suggests the distinction in different buckets is fundamentally encoded within the model's own representations.

Takeaway. In this experiment, a known mechanistic failure is turned into reusable diagnostic signals: the positional hypothesis is not merely "imperfect" on average but highly faithful on a structured subset and systematically unfaithful elsewhere, and this classifier generalizes to unseen inputs, best through internal SAE features. It demonstrates that our recipe can localize where an interpretation holds, characterize the inputs on which it breaks, and operationalize that distinction for future inputs.

4.3 Entangled Factual Recall

Task and input space. We evaluate our framework on *entangled factual recall* using the RAVEL benchmark (Huang et al., 2024). The task is to isolate one entity attribute from others co-encoded in the same representation (Fig. 5), which is hard because the attributes are entangled within a single internal state. Using a pretrained `Llama-3.1-8B` model, we focus on isolating the LANGUAGE attribute from others.

Alignment and Preliminary Evaluation. A successful alignment in RAVEL must be causally effective (a high **Cause** score: the intervention transfers the source's value of the target attribute $X \in \mathcal{A}$ into the base output) and isolated (a high **Iso** score: it leaves the other attributes $Y \in \mathcal{A} \setminus \{X\}$ unchanged); the losses are given in (1) and (2) in the appendix. Although Multi-task Distributed Alignment Search (MDAS) optimizes both criteria jointly, the resulting alignment reaches a held-out IIA of only 13.9%: the learned subspace is a fragile abstraction, and the aggregate metric cannot say why. Our method specifies why the alignment fails to generalize beyond the training distribution.

Bucketing the input space. We next apply our bucketing procedure to the learned MDAS subspace. The resulting graph is sparse overall, with edge density only about 10%. Once we partition this graph, however, a clear structure emerges (Fig 5 right). The dense buckets correspond almost exactly to single-language regions: one bucket is dominated by English-speaking cities and a second by Spanish-speaking cities, each with high IIA (around 99%). The remaining cities fall into a diffuse "other languages" bucket with very low IIA. Because the cities inside a bucket share a language, the high-level counterfactual within a bucket is the identity: the subspace swap leaves the answer intact within a language but does not carry the LANGUAGE value across languages. This shows that the learned MDAS subspace does not faithfully represent the LANGUAGE attribute in general. Instead, it collapses the input space into a small number of coarse language-specific clusters, preserving interchangeability mainly within the same language while failing on cross-language interventions.

Generalization through classifier. To test generalization, we train an SAE-based classifier to predict bucket membership for unseen inputs. The classifier reproduces the partition with 90%

[3] These numbers are measured on a fresh sample of inputs whose bucket assignments come from the classifier alone; the interventions on these inputs serve only to validate the predicted buckets, not to construct them.

accuracy, confirming that the language-specific structure is indeed encoded in the model's internal representations. Inspecting the highest-weight SAE features reveals only weak signals that are loosely consistent with the corresponding languages. The buckets themselves are far more direct: examining the grouped inputs assigned to each bucket makes the structure immediately clear, whereas reading the SAE feature descriptions directly give only weak and noisy evidence.

Takeaway. This experiment reveals a limitation of MDAS-style disentanglement in the entangled factual recall setting. Although MDAS aims to isolate the target concept from correlated non-target attributes, our analysis shows that it can do so only by *collapsing the representation*. The resulting subspace is partially isolated but only weakly faithful: it preserves coarse within-language structure while losing the finer-grained information needed for robust cross-language interchangeability. In this case, bucketing makes the failure mode explicit by showing that the learned representation separates a few major language clusters rather than capturing the LANGUAGE attribute as a whole.

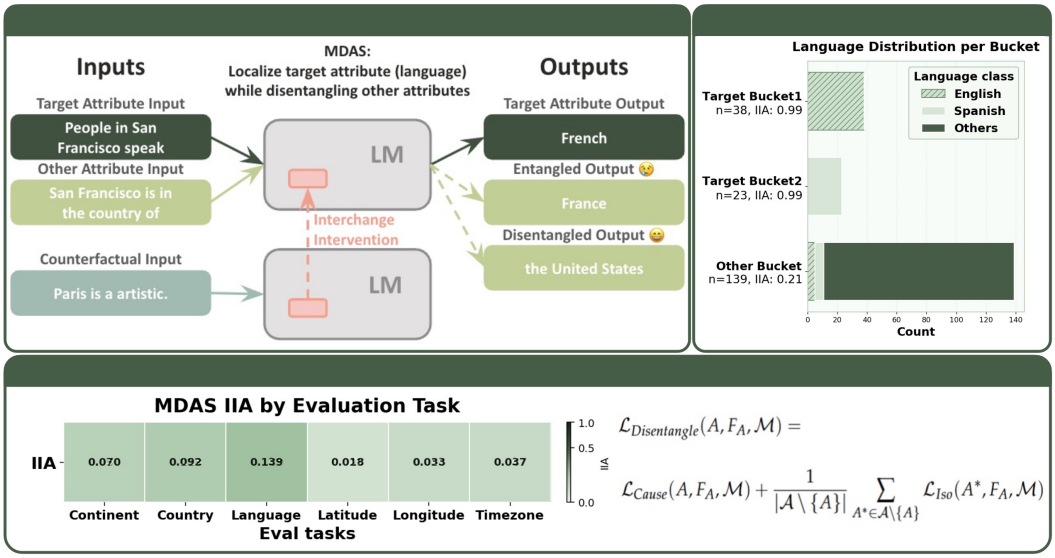

Figure 5: **Entangled Factual Recall Task and Diagnosis. Top Left:** This task utilizes the RAVEL benchmark to evaluate whether a specific entity attribute (e.g., LANGUAGE) can be isolated from other co-encoded attributes like COUNTRY, CONTINENT, and TIMEZONE. **Bottom:** MDAS Results. We train the alignment using all samples from all attributes to isolate the LANGUAGE attribute and evaluate it on all attributes. **Top Right:** After partition, the "Target Buckets" correspond to regions where the representation is highly faithful for specific languages—English and Spanish—while the "Other Bucket" captures the remaining modes.

5 CONCLUSION AND FUTURE WORK

We introduced a method for diagnosing and improving causal abstractions by bucketing the input space according to interchange-intervention behavior. Instead of treating IIA as a single global summary, our method identifies subsets of inputs on which a proposed abstraction is nearly perfect. This turns causal abstraction into a more diagnostic and constructive framework. Across fine-tuned and pretrained models, and across alignment methods including DAS, MDAS, and full-vector patching, our experiments show that bucketing is a broadly applicable diagnostic tool for causal abstraction; in the logic inference task, it also supports iterative discovery of the high-level hypothesis.

Our approach has two main limitations. First, it depends on a prior task construal: it operates within a specified task and candidate abstraction, rather than performing unconstrained automatic interpretation. In this sense, it is better understood as a structured program search than as a general solution to mechanistic interpretability. Second, our experiments focus on mostly single-variable hypotheses. In more complex settings, interacting variables and accumulated approximation error may make it difficult to discover large buckets with high IIA. These limitations suggest two directions for future work: scaling the method to richer tasks and genuinely multi-variable abstractions, and turning the structure of identified buckets into a systematic search over high-level hypotheses.

AI USE STATEMENT

We used generative AI tools to polish the writing of this paper and to help write the code for our experiments. We have reviewed all AI-assisted work: AI-edited text was checked against our intended meaning, and AI-assisted code was read and tested by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Aryaman Arora, Dan Jurafsky, and Christopher Potts. CausalGym: Benchmarking causal interpretability methods on linguistic tasks. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14638–14663, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.785.

Elias Bareinboim, Juan D Correa, Duligur Ibeling, and Thomas Icard. On pearl's hierarchy and the foundations of causal inference. In *Probabilistic and causal inference: the works of judea pearl*, pp. 507–556. 2022.

Sasha Boguraev, Christopher Potts, and Kyle Mahowald. Causal interventions reveal shared structure across English filler–gap constructions. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 25032–25053, Suzhou, China, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.emnlp-main.1271/.

Adly Bricken, Adni Templeton, Joshua Batson, Brian Chen, Adam Jerome, Scott Moore, Shahar Tamkin, Landon Jones, Dustin Conerly, Hoagy Cunningham, et al. Towards monosemanticity: Decomposing language models with sparse autoencoders. *Transformer Circuits Thread*, 2023.

Krzysztof Chalupka, Pietro Perona, and Frederick Eberhardt. Visual causal feature learning. *arXiv preprint arXiv:1412.2309*, 2014.

Krzysztof Chalupka, Frederick Eberhardt, and Pietro Perona. Causal feature learning: an overview. *Behaviormetrika*, 44(1):137–164, 2017.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Atticus Geiger, Kyle Richardson, and Christopher Potts. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 163–173, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.16. URL https://www.aclweb.org/anthology/2020.blackboxnlp-1.16.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34:9574–9586, 2021.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, et al. Causal abstraction: A theoretical foundation for mechanistic interpretability. *arXiv preprint arXiv:2301.04709*, 2023.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning*, pp. 160–187. PMLR, 2024.

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

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, 2021.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12216–12235, 2023.

Yoav Gur-Arieh, Mor Geva, and Atticus Geiger. Mixing mechanisms: How language models retrieve bound entities in-context. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, et al. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *arXiv preprint arXiv:2410.20526*, 2024.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. *arXiv preprint arXiv:2308.09124*, 2023.

Jing Huang, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger. Ravel: Evaluating interpretability methods on disentangling language model representations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8669–8687, 2024.

Jing Huang, Junyi Tao, Thomas Icard, Diyi Yang, and Christopher Potts. Internal causal mechanisms robustly predict language model out-of-distribution behaviors. In *Forty-second International Conference on Machine Learning*, 2025.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *arXiv preprint arXiv:2408.05147*, 2024.

Aleksandar Makelov, Georg Lange, and Neel Nanda. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching. *arXiv preprint arXiv:2311.17030*, 2023.

Maxime Méloux, Silviu Maniu, François Portet, and Maxime Peyrard. Everything, everywhere, all at once: is mechanistic interpretability identifiable? In *The Thirteenth International Conference on Learning Representations*, 2025.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36, 2022a.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022b.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Theodora-Mara Pîslar, Sara Magliacane, and Atticus Geiger. Combining causal models for more accurate abstractions of neural networks. In *Proceedings of the Fourth Conference on Causal Learning and Reasoning*, volume 275 of *Proceedings of Machine Learning Research*, pp. 114–138. PMLR, 2025.

Adam Scherlis, Kshitij Sachan, Adam S. Jermyn, Joe Benton, and Buck Shlegeris. Polysemanticity and capacity in neural networks. *CoRR*, abs/2210.01892, 2022. doi: 10.48550/ARXIV.2210.01892. URL https://doi.org/10.48550/arXiv.2210.01892.

Paul Smolensky. Neural and conceptual interpretation of pdp models. *Parallel distributed processing: Explorations in the microstructure of cognition*, 2:390–431, 1986.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Zhengxuan Wu, Atticus Geiger, Aryaman Arora, Jing Huang, Zheng Wang, Noah D Goodman, Christopher D Manning, and Christopher Potts. pyvene: A library for understanding and improving pytorch models via interventions. *arXiv preprint arXiv:2403.07809*, 2024a.

Zhengxuan Wu, Atticus Geiger, Jing Huang, Aryaman Arora, Thomas Icard, Christopher Potts, and Noah D Goodman. A reply to makelov et al.(2023)'s" interpretability illusion" arguments. *arXiv preprint arXiv:2401.12631*, 2024b.

Fabio Massimo Zennaro, Paolo Turrini, and Theodoros Damoulas. Quantifying consistency and information loss for causal abstraction learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)*, pp. 5750–5757, 2023.

APPENDIX

A METHODOLOGY

The formal diagnosis pipeline is shown in Algorithm 1 and Algorithm 2.

Greedy Multi-Seed Quasi-Clique Search As the identification of maximum cliques within the Interchangeability Graph is generally NP-hard, we implement a heuristic greedy search to approximate the interchange-consistent subspaces (Algorithm 2). This algorithm serves as the implementation of the FINDMAXIMALQUASICLIQUE function within our diagnosis pipeline.

To maximize the probability of discovering large dense regions, the algorithm employs a multi-seed strategy starting from the most promising candidates. We first compute the degrees of all nodes in $V \subseteq \mathcal{I}_{\text{correct}}$ based on the current subgraph and sort them in descending order. The expansion is then attempted independently for each of the top 10 highest-degree nodes as seeds. For a given seed, the algorithm iteratively expands the set C by selecting the candidate node w that maximizes the resulting edge density ρ, provided that ρ remains above the user-defined threshold γ.

The density threshold $\gamma \in (0, 1]$ allows the framework to be robust to minor representational noise; while $\gamma = 1.0$ identifies a perfect clique, a slightly lower value (e.g., $\gamma = 0.9$) allows the pipeline to isolate regions of high causal faithfulness that might otherwise be fragmented by trivial neural variations. We choose $\gamma = 0.98$ for all experiments in the paper. The algorithm tracks the results of each seed and returns the largest quasi-clique C_{best}, which is then extracted as a "Target Bucket" before the search space is updated for the next iteration.

Algorithm 1 The Diagnosis Pipeline

Require: Causal abstraction $(\mathcal{L}, \mathcal{H}, \Pi)$, input space $\mathcal{I}$, density threshold γ, minimum size s, max buckets K

$\mathcal{I}_{\text{correct}} \leftarrow \{i \in \mathcal{I} : \mathcal{L}(i) \text{ is correct}\}$
Construct Interchangeability Graph $G = (V, E)$ on $\mathcal{I}_{\text{correct}}$ (Definition 1)
for $j = 1$ **to** $K - 1$ **do**
  $C_j \leftarrow$ FINDMAXIMALQUASICLIQUE$(G[V], \gamma, s)$ # Identify a dense, faithful region
  if $C_j = \emptyset$ **then**
    break
  end if
  $V \leftarrow V \setminus C_j$ # Remove identified region from the search space
end for
$C_K \leftarrow V$ # Residual bucket
$\mathcal{D}_{\text{train}} \leftarrow \{(i, j) \mid i \in C_j, j \in \{1, \ldots, K\}\}$
$g \leftarrow$ TRAINCLASSIFIER$(\mathcal{D}_{\text{train}})$ # Learn to generalize the partition
return Buckets $\{C_j\}_{j=1}^{K}$ and diagnostic classifier g

A.1 SENSITIVITY TO THE DENSITY THRESHOLD γ AND TO THE SEARCH HEURISTIC

The bucketing step has two free hyperparameters: the density threshold γ and the number of seeds used by the greedy expansion in Algorithm 2. Since FINDMAXIMALQUASICLIQUE is a heuristic for an NP-hard problem, both deserve an ablation. We therefore sweep $\gamma \in \{0.80, 0.85, 0.90, 0.95, 1.00\}$ and the seed budget over $\{5, 10, 15, 20\}$ on all four diagnosis passes reported in the paper, with $K = 2$ buckets. All quantities reported below are pre-classification in Step 3, before any classifier is trained (Step 4).

We fix $K = 2$ to keep the four settings comparable. The one place it departs from the main text is RAVEL, where the reported diagnosis uses $K = 3$ and isolates English and Spanish as two separate faithful regions (Fig. 5); panel (d) reports the English bucket only, and the Spanish cluster remains inside the residual. Spanish is its most frequent language at every threshold, at 0.17–0.19, and its IIA of 0.21–0.22 averages a clean Spanish region together with the diffuse remainder, so it understates the former. The English bucket itself is unaffected, since it is what the search extracts first at every γ;

Algorithm 2 Multi-Seed Greedy Quasi-Clique Search

Require: Adjacency matrix A, available nodes $V \subseteq \mathcal{I}_{\text{correct}}$, density threshold γ, minimum size s

if $|V| < s$ **then**
  return $\emptyset$
end if
Compute degrees of nodes in V based on subgraph $G[V]$
$V_{\text{sorted}} \leftarrow$ nodes in V sorted by degree descending
$C_{\text{best}} \leftarrow \emptyset$
for v_{seed} **in** $V_{\text{sorted}}[1 \ldots \min(10, |V|)]$ **do**
  $C \leftarrow \{v_{\text{seed}}\}$; $S \leftarrow V \setminus \{v_{\text{seed}}\}$
  improved $\leftarrow$ True
  while improved **and** $S \neq \emptyset$ **do**
    improved $\leftarrow$ False
    $u^* \leftarrow$ None; $\rho^* \leftarrow 0$
    for w **in** S **do**
      $\rho \leftarrow \text{Density}(C \cup \{w\})$
      if $\rho \geq \gamma$ **and** $\rho > \rho^*$ **then**
        $u^* \leftarrow w$; $\rho^* \leftarrow \rho$; improved $\leftarrow$ True
      end if
    end for
    if improved **then**
      $C \leftarrow C \cup \{u^*\}$; $S \leftarrow S \setminus \{u^*\}$
    end if
  end while
  if $|C| \geq s$ **and** $|C| > |C_{\text{best}}|$ **then**
    $C_{\text{best}} \leftarrow C$
  end if
end for
return C_{best}

its residual English fraction rises from 0.01 to 0.12 across the sweep, which is simply the English cities shed as the extracted bucket tightens from 49 inputs to 31.

The greedy search is insensitive to random seeds on our graphs. Varying the seed budget leaves the recovered partition almost unchanged. It is bit-for-bit identical across all four budgets on the logic o_4 pass, and elsewhere the largest disagreement between any two budgets is 3 of 200 nodes (1.5%) on the logic o_5 pass, 1 of 507 (0.2%) on entity binding, and 2 of 200 (1.0%) on RAVEL. The governing primitive is unchanged in every case. The reason is structural rather than lucky: each target bucket is dense relative to the ambient graph (within-bucket IIA 0.89–1.00 against overall IIA 0.23–0.86), so greedy expansion climbs to the same plateau no matter which high-degree vertex it starts from. The NP-hardness of exact maximum-clique search is therefore not the operative difficulty in this regime. Table 1 accordingly reports the 10-seed setting used throughout the paper.

γ trades coverage for purity without changing the diagnosis. Across the entire range $\gamma \in [0.80, 1.00]$, each partition is governed by a single primitive, and it is the same primitive at every threshold: o_1 for the logic o_4 pass, o_4 for the logic o_5 pass, whether the queried group sits at the start or tail of the prompt for entity binding, and whether the city's language is English for RAVEL. What γ moves is only the operating point: lowering it admits more inputs into the target bucket at lower within-bucket IIA, raising it yields a smaller and purer bucket. This is what a thresholding parameter should do—let the user choose a point on a purity/coverage curve.

The two buckets keep their identities across the entire sweep: in panel (a) the target bucket's $\Pr[o_1]$ falls monotonically from 0.41 to 0.00 while the residual sits at $o_1 = \texttt{True}$ throughout, and panel (b) behaves the same way with o_4. What moves with γ is which of the two buckets is the interchange-consistent one. Because the greedy search maximizes the size of the extracted set subject to density $\geq \gamma$, that set always comes to rest essentially at the threshold, and at $\gamma \leq 0.90$ it is the small, homogeneous residual that happens to be an exact clique. At $\gamma = 1.00$ the extracted set must

itself be an exact clique, and the largest one available is the region on which the target computation degenerates: $o_1 = \texttt{False}$ forces $o_4 = o_1 \wedge o_2 = \texttt{False}$, and $o_4 = \texttt{False}$ forces $o_5 = o_3$, so every pair inside that region is trivially interchangeable. Both regimes isolate the same boundary and support the same conclusion: the o_4-aligned representation under-encodes o_2, and the o_5-aligned representation under-encodes the conjunction branch. Panels (c) and (d) show nothing of the kind: the target bucket is 100% start/tail and at least 94% English, respectively, at every threshold.

Table 1: **Sensitivity of the partition structure to the density threshold** γ (10 seeds). Every panel uses $K = 2$, so each reports one extracted bucket and one residual. This is the partition the main text reports in panels (a)–(c); the RAVEL diagnosis in the main text uses $K = 3$, so panel (d) shows its English bucket only and leaves the Spanish bucket inside the residual. "Target" is the bucket extracted by Algorithm 2 (identifiable in every row by its density coming to rest at γ); "other" is the residual. IIA is the directed within-bucket interchange intervention accuracy: the fraction of ordered input pairs (i, j) for which the interchange intervention is correct. $\Pr[\text{prim.}]$ is the frequency with which the governing primitive named in each panel header holds inside the bucket. Bold marks values of exactly 1.00; † marks the default $\gamma = 0.98$ used in the main text. All values are pre-classification. The governing primitive is the same at every γ in all four settings, and each bucket keeps its identity across the sweep; only sizes and purity shift. See Appendix A.1 for why the perfectly interchangeable bucket changes sides at $\gamma = 1.00$ in panels (a) and (b).

	Target bucket			**Other bucket**		
γ	size	IIA	$\Pr[\text{prim.}]$	size	IIA	$\Pr[\text{prim.}]$
(a) Logic, o_4 at position 78, layer 5 — overall IIA 0.856*; governing primitive* o_1						
0.80	176	0.89	0.41	23	**1.00**	**1.00**
0.85	158	0.92	0.34	41	**1.00**	**1.00**
0.90	144	0.95	0.28	55	**1.00**	**1.00**
0.95	130	0.98	0.20	69	0.93	**1.00**
$0.98^\dagger$	115	0.99	0.10	84	0.78	**1.00**
1.00	103	**1.00**	0.00	96	0.72	0.99
(b) Logic, o_5 at position 77, layer 7 — overall IIA 0.680*; governing primitive* o_4						
0.80	158	0.93	0.09	42	**1.00**	**1.00**
0.85	158	0.93	0.09	42	**1.00**	**1.00**
0.90	154	0.95	0.08	46	**1.00**	0.98
0.95	149	0.98	0.07	51	0.99	0.92
$0.98^\dagger$	142	0.99	0.05	58	0.95	0.86
1.00	125	**1.00**	0.02	75	0.84	0.73
(c) Entity binding at layer 15 — overall IIA 0.480*; governing primitive: start/tail group*						
0.80	207	0.90	**1.00**	300	0.21	0.02
0.85	191	0.92	**1.00**	316	0.23	0.07
0.90	171	0.95	**1.00**	336	0.25	0.12
0.95	148	0.98	**1.00**	359	0.28	0.18
$0.98^\dagger$	129	0.99	**1.00**	378	0.31	0.22
1.00	104	**1.00**	**1.00**	403	0.35	0.27
(d) RAVEL LANGUAGE *at layer 14, English bucket only — overall IIA* 0.234*; governing primitive: English*						
0.80	49	0.90	0.96	151	0.22	0.01
0.85	47	0.92	0.96	153	0.22	0.03
0.90	44	0.95	0.95	156	0.21	0.05
0.95	41	0.98	0.98	159	0.22	0.06
$0.98^\dagger$	38	0.99	0.97	162	0.21	0.07
1.00	31	**1.00**	0.94	169	0.21	0.12

A.2 COMPUTATIONAL COST

A diagnosis pass spends almost all of its compute in Step 3, which needs an interchange intervention for every ordered pair of analyzed inputs. Building the graph on N inputs therefore costs $O(N^2)$ interventions. This subsection describes how the pipeline is run.

The cost grows with the number of inputs placed in the graph rather than the complexity of the hypothesis. By Definition 1, an edge requires perfect interchangeability with respect to every variable in $\mathsf{Var}(\mathcal{H})$, so a candidate edge costs one intervention per variable and graph construction is linear in the number of high-level variables. Moving from a single-variable hypothesis to a multi-variable one multiplies the constant and leaves the N^2 alone.

We don't need to train on every possible pairs in Step 3. Instead, we subsample some pairs and train a classifier on the discovered buckets. The classifier assigns bucket membership to the remaining inputs by generalizing what it learns from the subsample. A predicted bucket can then be checked by drawing a small testing sample of size m from it and measuring within-bucket IIA at a cost of $O(m^2)$; if the predicted target bucket is a faithful region, its IIA on inputs the graph never saw should stay close to IIA on the training data. In our experiments N between 200 and 512 was enough—the sweep in Table 1 was computed at these sizes—and the Step-4 classifiers generalized bucket membership to held-out inputs with high accuracy in every setting. Moreover, the interventions themselves are independent. Each pair depends only on the frozen model and the fixed alignment, with no ordering constraints and no shared state, so the pairs can be parallelized.

A.3 Baseline: Clustering the Activations Directly

A natural question is whether the partitions recovered in Step 3 are specific to the interchangeability-graph framing, or whether a simpler diagnostic—clustering the residual-stream activations—would find the same regions. To test this, we replace Step 3 with a purely geometric baseline. For every input in the graph we extract its activation at the aligned site (GPT-2 at layer 5, position 78 and at layer 7, position 77 for the two logic passes; Gemma-2-2B at layer 15, final query token, for entity binding; Llama-3.1-8B at layer 14, final city token, for RAVEL), reduce the activations to 50 principal components, and run K-means (20 restarts, seed 42) with the same number of buckets K as our partition. The resulting buckets are then scored on the same interchange interventions, so both methods are judged by the same quantity: the within-bucket IIA of a bucket is the fraction of ordered (source, base) pairs of its inputs on which the interchange intervention succeeds, the directed IIA used in Table 1. (Our partition itself is still computed on the undirected graph of Definition 1, whose edges require success in both directions.) Agreement between the two partitions is measured by the adjusted Rand index (ARI) and normalized mutual information (NMI), both equal to 1 for identical partitions and close to 0 for unrelated ones. Table 2 reports the numbers, and Fig. 6 shows both partitions on the first two principal components of the activations.

Activation geometry recovers our partition only where the governing primitive happens to dominate the geometry. On the logic o_4 pass and on entity binding, where o_1 and the start/tail position are close to linearly separable in the representation, K-means lands near our buckets (ARI 0.77 and 0.85), and its densest bucket is a slightly smaller and slightly purer version of ours. On the logic o_5 pass it is unrelated to our partition (ARI 0.04): it isolates a small geometric cluster of 40 inputs that happens to be dense, and misses the 142-input bucket organized by the value of o_4. On RAVEL it fails outright: the densest K-means bucket reaches an IIA of only 0.41 (K=2) or 0.34 (K=3), against 0.99 for ours, and Fig. 6 shows why—the 38 English cities that form our target bucket are spread across the whole activation cloud, whereas K-means simply splits that cloud along its first principal component. Our partitions reach a within-bucket IIA of at least 0.99 in every setting because they are read off intervention outcomes rather than geometry; where geometric proximity and causal interchangeability come apart, only the former is visible to a clustering baseline.

B Logic Inference Task Details

B.1 Task Specification and Dataset

The logic inference task is designed as a controlled synthetic setting where the target computation is perfectly specified by a known Boolean expression. The model is presented with a sequence of six input tokens, $t_0, \ldots, t_5$, drawn uniformly from a predefined vocabulary. The target label is defined by the high-level causal model $\mathcal{H}$, which computes the expression $o_5 = ((t_2 \neq t_4) \wedge (t_0 \neq t_5)) \vee (t_1 = t_3)$.

Table 2: **PCA + K-means baseline vs. our interchangeability-graph partition.** For each diagnosis pass, the buckets of each method are ranked by within-bucket IIA (directed, as in Table 1: the fraction of ordered source/base pairs inside the bucket on which the intervention succeeds): "target" is the bucket with the highest IIA, "other" the lowest, and "second" the remaining bucket when K=3 (ties broken by extraction order). "Overall" is the directed IIA of the whole graph. ARI and NMI measure the agreement of the K-means partition with ours. Our partitions use $\gamma = 0.98$ and 10 seeds; the baseline uses 50 principal components and 20 K-means restarts (seed 42) on the activations of the same inputs. RAVEL uses the MDAS graph of Fig. 5: K=2 extracts the English bucket only, K=3 additionally splits off Spanish.

Experiment	Method	Target		Second		Other		Overall	Agreement	
		n	IIA	n	IIA	n	IIA	IIA	ARI	NMI
Logic o_4	Ours	115	0.99	–	–	84	0.78	0.86	–	–
	PCA+K-means	103	1.00	–	–	96	0.72	0.86	0.77	0.73
Logic o_5	Ours	142	0.99	–	–	58	0.95	0.68	–	–
	PCA+K-means	40	0.97	–	–	160	0.60	0.68	0.04	0.01
Entity binding	Ours	129	0.99	–	–	378	0.31	0.48	–	–
	PCA+K-means	111	1.00	–	–	396	0.34	0.48	0.85	0.77
RAVEL, K=2	Ours	38	0.99	–	–	162	0.21	0.23	–	–
	PCA+K-means	60	0.41	–	–	140	0.22	0.23	0.34	0.23
RAVEL, K=3	Ours	38	0.99	23	0.99	139	0.21	0.23	–	–
	PCA+K-means	42	0.34	90	0.28	68	0.25	0.23	0.09	0.10

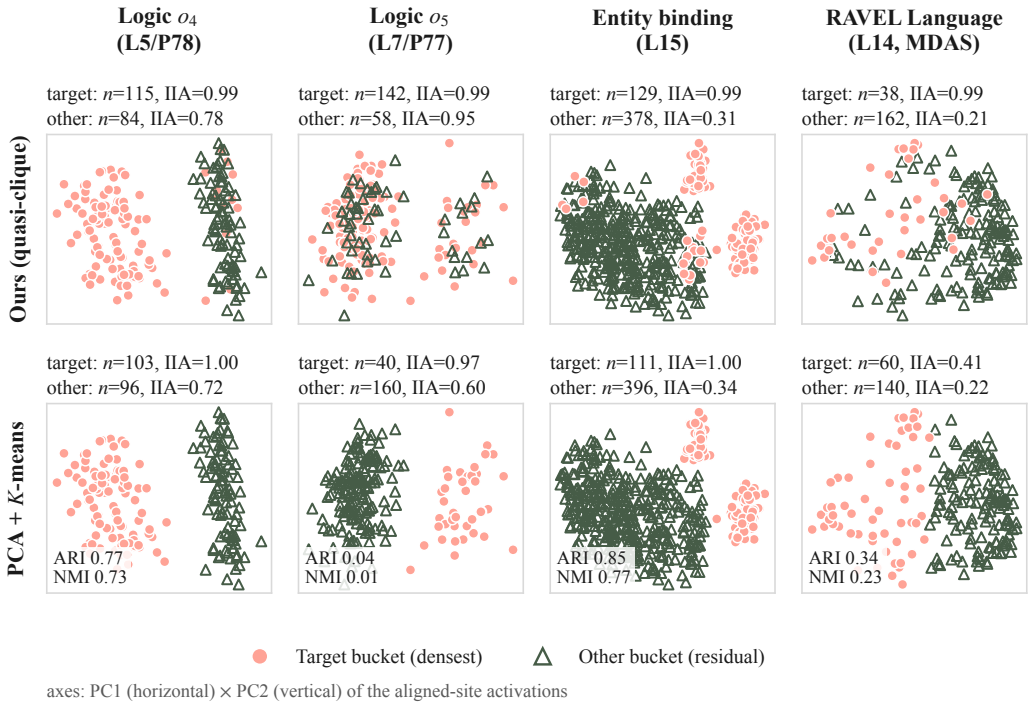

Figure 6: **Both partitions in activation space.** Each column is one diagnosis pass; each point is one input, placed at the first two principal components of its activation at the aligned site. Top row: points coloured by our bucket (quasi-clique on the interchangeability graph). Bottom row: the same points coloured by their K-means bucket. Panel headers give bucket sizes and within-bucket IIA; ARI and NMI compare the two partitions. Where the governing primitive is geometrically separable (logic o_4, entity binding) the two colourings nearly coincide; on logic o_5 and RAVEL the target bucket that interventions reveal does not correspond to any geometric cluster.

For our analysis, we explicitly define the intermediate causal variables as follows:

$$o_1 := (t_2 \neq t_4)$$
$$o_2 := (t_0 \neq t_5)$$
$$o_3 := (t_1 = t_3)$$
$$o_4 := o_1 \wedge o_2$$
$$o_5 := o_4 \vee o_3$$

Dataset Construction To format the inputs for the language model, we use an in-context learning template. Each prompt consists of 5 randomly sampled context examples (with their corresponding ground-truth Boolean labels) to establish the task format, followed by the target six-token sequence formatted as "`t0,t1,t2,t3,t4,t5=`". The dataset is constructed such that the intermediate variables (e.g., o_3) are balanced to be True with a probability of approximately 0.5. We train a 12-layer `GPT-2-small` ($\sim$117M parameters) on 20,000 such examples, achieving 99.7% task accuracy.

Before any alignment search or graph construction, we rigorously filter the dataset to the $\mathcal{I}_{\text{correct}}$ subset: we ensure that the model correctly predicts both the base input and the source input in isolation, verifying that any downstream failure under interchange intervention stems strictly from the alignment hypothesis, not from base model incompetence.

B.2 Experimental Setup: Alignment Search

We use DAS to map each high-level variable to the model's internal representations. We utilize the `pyvene` library to train DAS. Fig. 7 shows the heat maps for o_1–o_4; the o_5 heat map appears in Fig. 2.

Training Details Because the causal variables are binary, we train a 1-dimensional orthogonal rotation matrix (`subspace_dimension=1`) operating on the block output. For each candidate layer and token position, we train the DAS intervention for 5 epochs using the Adam optimizer (learning rate = 0.001) with a batch size of 32. The objective minimizes the Cross-Entropy loss between the model's counterfactual output and the target causal model's predicted counterfactual label.

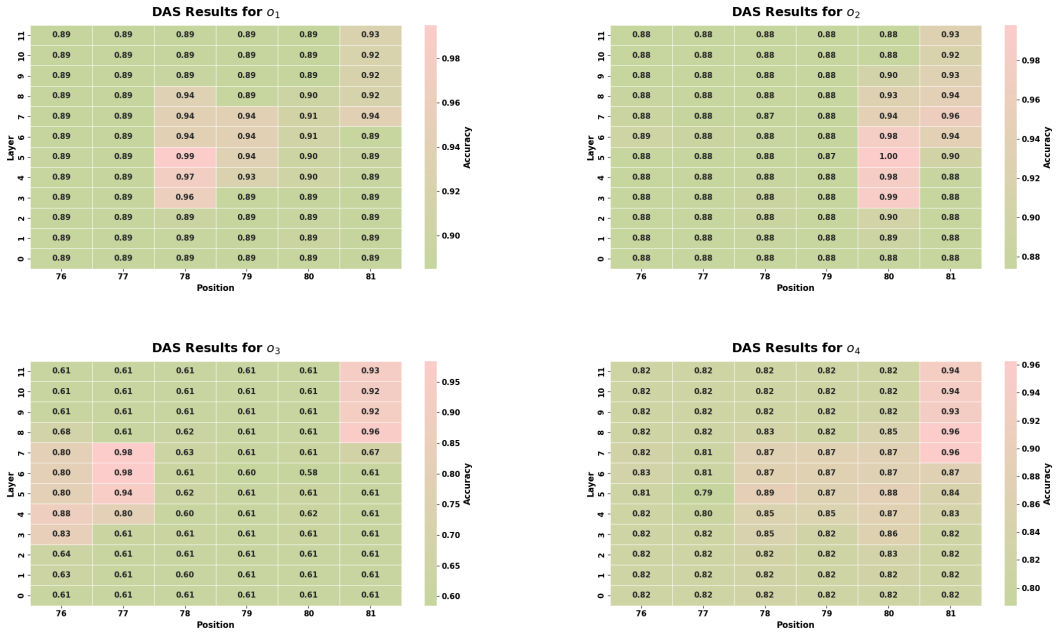

Figure 7: DAS Alignment Heat Maps for o_1, o_2, o_3, o_4.

B.3 Diagnostic Graph Construction

Once the optimal candidate layer and position for an intermediate variable are identified via DAS, we construct the Interchangeability Graph to diagnose the alignment's structural faithfulness. For N sampled inputs from the filtered dataset, we perform exhaustive directed interventions for all pairs (i, j) using the trained DAS weight. An undirected edge is drawn between node i and node j if and only if the interchange intervention succeeds bidirectionally—meaning the model accurately predicts the counterfactual output when patching from $i \to j$, and independently when patching from $j \to i$.

Partitioning We partition the resulting graph using a heuristic multi-seed greedy quasi-clique search (Algorithm 2). To accommodate slight representational noise inherent to neural activations, we apply a density threshold of $\gamma = 0.98$ and a minimum clique size of 2. The resulting clusters explicitly separate the input space into interchange-consistent regions and under-interpreted regions.

B.4 SAE-Based Generalization and Classification

To determine if the boundaries of the interchange-consistent subspaces are explicitly encoded in the model's feature space, we train classifiers using SAE features to predict bucket membership.

Feature Extraction and Classification For the nodes in the intervention graph, we extract the residual-stream activations at the targeted layer and token position. We encode these activations using pretrained SAEs for GPT-2 Small (specifically, the `gpt2-small-res-jb` release from `sae_lens`). Using the resulting sparse feature activations, we train an ℓ_1-regularized Logistic Regression classifier on an 80/20 train/test split. The model is trained to classify whether an input belongs to the target bucket or the other bucket. The high test accuracy of this classifier and the sparse set of non-zero ℓ_1 coefficients allow us to identify the specific features (such as o_4) that dictate whether the high-level causal hypothesis holds.

C Entity Binding Task Details

C.1 Task Specification and Dataset

We instantiate the entity binding task using the `filling_liquids` task family. In this setting, the model is provided with a prompt containing 10 distinct entity groups. Each group follows a fixed template: *"[Person] fills a [Container] with [Liquid]."* The prompt concludes with a query such as *"Who filled the [Container]?"*, requiring the model to retrieve the specific [Person] associated with that container from the preceding context.

Entity Pools To ensure high diversity and minimize accidental overlaps, we utilize the following entity pools:

- **Persons:** John, Mary, Bob, Sue, Tim, Kate, Dan, Lily, Max, Eva, Sam, Zoe, Leo, Mia, Noah, Ava, Ben, Liz, Tom, Joy.
- **Containers:** cup, glass, bottle, mug, jar, pitcher, bowl, flask, tumbler, chalice, vessel, container, tank, can, tube, vial, goblet, stein, carafe, decanter.
- **Liquids:** beer, wine, water, juice, milk, coffee, tea, soda, lemonade, smoothie, soup, broth, sauce, syrup, oil, honey, cider, nectar, punch, tonic.

Dataset Construction We generate a dataset of 1,024 samples. Following our diagnostic pipeline, we filter this dataset to include only instances where `Gemma-2-2B-Instruct` correctly predicts the target entity for both the base and counterfactual inputs. After filtering, the dataset is split into training (80%) and testing (20%) sets.

C.2 Experimental Setup

We use `pyvene` to conduct vanilla interchange interventions. We test the *positional hypothesis*, which posits that the model's retrieval mechanism is mediated by a high-level variable q_{group} representing the context position of the queried group.

Alignment Procedure Counterfactuals are generated by swapping the positions of entity groups while maintaining the templatic structure. We perform full-vector patching on the residual stream at the final query token position across all layers. A faithful abstraction requires that patching the internal representation from a source input into a base input causes the model to retrieve the entity corresponding to the source's queried group position.

C.3 Diagnostic Results

We sample 512 correctly answered prompts from the `filling_liquids` input space, perform pairwise interchange interventions, and construct an Interchangeability Graph whose vertices represent individual inputs. We analyze the resulting IIA to construct the interchangeability graph shown in Fig. 8.

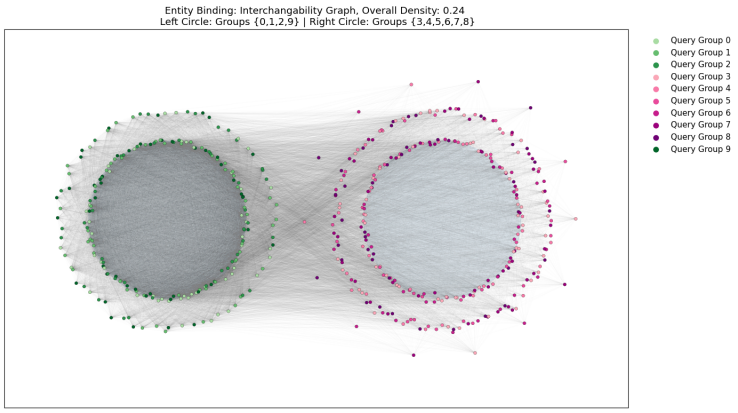

Figure 8: **Interchangeability Graph for Entity Binding.** The graph nodes represent individual inputs, and edges denote perfect pairwise interchangeability. The emergent community structure corresponds to the "Target Buckets" where the positional hypothesis is perfectly faithful, primarily at the start and end of the prompt sequence.

For the classification step of our diagnosis, we utilize internal features from `Gemma Scope` (Lieberum et al., 2024) to determine if the boundaries between well-interpreted and under-interpreted regions are explicitly encoded in the model's feature space.

D Entangled Factual Recall Task Details

D.1 Task Specification and Dataset

We evaluate the entangled factual recall setting using the RAVEL benchmark (Huang et al., 2024) on the `Llama-3.1-8B` model. We focus on the Language attribute of the *city* entity type. The input space consists of prompts where the base and source inputs share the exact same template but differ only in the city name (e.g., *"People in San Francisco speak..."* vs. *"People in Paris speak..."*).

To ensure our interventions target the specific entity representation, we extract the residual stream activations at the final token of the city name. The dataset is filtered to strictly include instances where the base model correctly predicts the factual target. The filtered dataset is then split into an 80% training set and a 20% testing set.

D.2 Experimental Setup: MDAS

We employ Multi-task Distributed Alignment Search (MDAS) to learn a disentangled subspace that isolates the Language attribute from other co-encoded attributes (Continent, Country, Latitude, Longitude, and Timezone).

MDAS Loss Following (Huang et al., 2024), given an entity E and an attribute A with a ground-truth value A_E (e.g., Paris and Continent), we seek to learn a feature F_A. A high **Cause** score

indicates that intervening on this representation successfully transfers a source attribute value $A_{E'}$ from a counterfactual input x' to the model's prediction for a base input x, optimized as:

$$\mathcal{L}_{Cause}(A, F_A, \mathcal{M}) = \text{CE}(\text{II}(\mathcal{M}, F_A, x, x'), A_{E'}) \tag{1}$$

Conversely, a high **Iso** (isolation) score ensures the intervention is surgical and does not alter other attributes $A^* \in \mathcal{A} \setminus \{A\}$. For a prompt x^* querying a different attribute A^* with ground-truth value A^*_E (e.g., LANGUAGE), the isolation loss ensures the model still predicts the original value A^*_E despite the intervention on F_A:

$$\mathcal{L}_{Iso}(A, F_A, \mathcal{M}) = \frac{1}{|\mathcal{A} \setminus \{A\}|} \sum_{A^* \in \mathcal{A} \setminus \{A\}} \text{CE}(\text{II}(\mathcal{M}, F_A, x^*, x'), A^*_E) \tag{2}$$

The MDAS training loss is the summation of $\mathcal{L}_{Cause}$ and $\mathcal{L}_{Iso}$.

Training Details Rather than exhaustively training across all layers, we first perform a coarse search using a large layer gap across subspace dimensions $k \in \{32, 128, 512, 2048\}$. Based on the preliminary alignment results from this initial sweep, we zoom in on the most promising regions to localize the optimal subspace and identify layer 14 as the ideal intervention site. To optimize for both causal transfer and isolation, we construct a weighted training distribution: pairs evaluating the target attribute (LANGUAGE) are sampled at a 5:1 ratio compared to pairs evaluating the other five off-target attributes. This forces the subspace to maximize the interchange intervention accuracy (IIA) for LANGUAGE while ensuring interventions do not alter the model's predictions for queries about a city's continent or timezone.

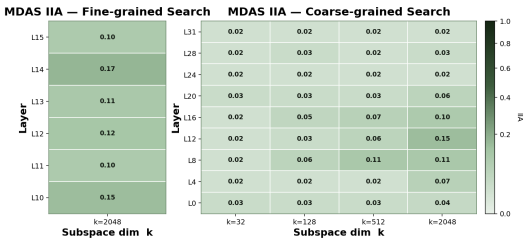

Figure 9: MDAS Alignment Results for Entangled Factual Recall Task

D.3 DIAGNOSTIC RESULTS AND CLASSIFICATION

Following the MDAS optimization, we apply our diagnostic pipeline to the test set.

Graph Construction and Partition We perform exhaustive pairwise interchange interventions. An undirected edge is formed between two inputs if and only if the bidirectional interchange intervention (source $\rightarrow$ base, and base $\rightarrow$ source) is perfectly consistent with the high-level causal model. We partition this graph using Algorithm 2 with a density threshold of $\gamma = 0.98$ and a minimum clique size of 2, which successfully isolates the language-specific clusters (English, Spanish) discussed in the main text. We find that with fewer buckets, the partition first separates English from non-English cities, and then Spanish splits off as a second clean group.

SAE-Based Generalization To test if these boundaries are encoded in the model's internal feature space, we train a classifier using sparse features from *Llama Scope* (He et al., 2024). We extract the residual stream activations at layer 14 for the graph nodes and encode them through the corresponding Llama SAE.

Using the resulting sparse feature activations, we train a multi-class ℓ_1-regularized Logistic Regression classifier to predict the quasi-clique bucket labels. To interpret the structural differences between buckets, we extract the highest-weight SAE features by analyzing the ℓ_1 coefficients.