A SPARSE PROTOTYPE READOUT FOR LLM IN-CONTEXT CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

What class-level information in a language model's contextual representations drives in-context classification? How is that information transformed into a prediction? Across four open-weight LLMs, we trace this computation from contextual representations to the final readout. Contextual processing progressively constructs a low-dimensional class geometry whose class-mean directions are causally linked to the model's decision: continuously exchanging these directions reverses the output. We then follow this decision variable into a sparse set of attention heads. Contextual class means are reflected in class-mean keys, yielding headwise query–prototype margins in the native attention computation. Under the same geometric intervention, these margins vary approximately linearly and cross near zero where the contextual class-mean directions meet. On held-out instances, a fixed sparse combination of the native prototype margins predicts the model's output with $R^2 = .896$–$.955$ and 93.4–95.2% prompt-level agreement. Thus, we identify a class-level geometric decision variable for in-context classification and a sparse prototype-based circuit that reads it out.

1 INTRODUCTION

Consider a prompt containing

$$\texttt{bedroom} \to a, \quad \texttt{hammer} \to b, \quad \texttt{lobby} \to a, \quad \texttt{chisel} \to b, \quad \ldots, \quad \texttt{attic} \to ?$$

The categories are unnamed and the labels are arbitrary. Exchanging every a and b preserves the semantic grouping but reverses the required answer. The model must relate *attic* to the demonstrated rooms and express that comparison through their temporary label. What internal representation of the classes supports this decision? More specifically, can we identify a class-level component of the model's layerwise representational state, manipulate it continuously, and follow its effect into the native computation that produces the output?

Prior work establishes important pieces of this picture. Sparse attention pathways support semantic matching and temporary-label transport (Yu & Ananiadou, 2024; Cho et al., 2025a); label-token representations accumulate demonstration information (Wang et al., 2023b); contextual representations develop task-aligned geometry (Park et al., 2025; Yang et al., 2025); and prototype-based quantities can describe native output differences (Xiong et al., 2026). These results motivate a more specific mechanistic question: *Can a controlled change in the represented class state produce a quantitative change in the native readout? And if so, how does the model implement this transformation?*

What is new? We answer this question by connecting four internal quantities through a sequence of intervention, functional, and predictive tests:

$$\boxed{\boldsymbol{\mu}_c^{\ell} \;\longrightarrow\; \bar{\mathbf{k}}_{h,c} \;\longrightarrow\; m_h \;\longrightarrow\; m}$$

Here μ_c^{ℓ} describes the class-level means of the representational word states at layer ℓ; $\bar{\mathbf{k}}_{h,c}$ is the mean support-label key for displayed label y in an implicated attention head; m_h is that head's query–prototype margin; and m is the native output margin. First, we manipulate the upstream word representations, measure how the native head margin responds. Next, we test the causal and functional role of the key prototypes. Finally we ask whether the resulting head margins predict ordinary decisions on unseen semantic groups. The result is a falsifiable account of how a controllable class-level representation is transformed into a native output.

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.

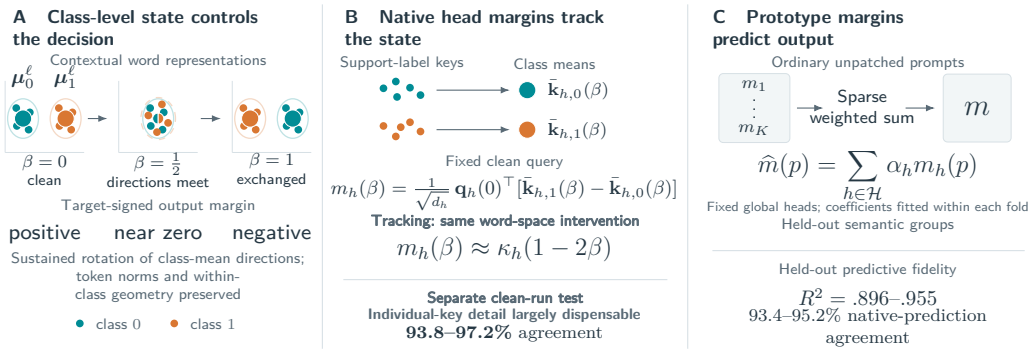


Figure 1: **From a manipulable class-level representation to a sparse prototype readout.** (A) A sustained intervention moves the contextual support-class mean directions $\boldsymbol{\mu}_0^\ell, \boldsymbol{\mu}_1^\ell$ from their clean configuration, through coincidence, to exchange; the model's target-signed preference crosses zero and reverses. (B) Under the same intervention, class-mean keys in causally implicated heads produce prototype margins that track the manipulated geometry. In a separate clean-run intervention, replacing individual support-label keys by their within-prompt class means preserves 93.8–97.2% of clean predictions. (C) On ordinary unpatched prompts, a fixed sparse combination of these native prototype margins predicts signed output margins on held-out semantic groups with $R^2 = .896$–$.955$ and 93.4–95.2% agreement with native predictions. The intervention and held-out prediction are distinct tests of the same native readout quantities.

Task Setup. We study balanced binary semantic classification in four open-weight LLMs. To suppress a simple input-centroid explanation, we select class pairs whose true-class centroid separation at the input approximately matches that expected under a random balanced partition. Despite this control, contextual processing develops increasingly useful within-prompt class structure: same-class alignment rises from 51–53% at the input to 67–70% in later layers.

Does the class-level state control the decision (Figure 1A)? For each prompt p and layer ℓ, let $\boldsymbol{\mu}_c^\ell(p)$ denote the mean contextual representation of the support words in semantic class c. In Section 4, we continuously rotate the two class clouds toward exchanged mean directions, using a sustained intervention across layers that preserves token norms and within-class inner products while changing the relative class-mean directions. Accuracy moves from 82–85% under the clean geometry to approximately chance when the mean directions coincide, and to 14–22% when they have exchanged; the target-signed output margin crosses zero and changes sign along the same intervention. Thus the class means are more than a decodable summary of the representation: their relative geometry is a controllable aspect of the layerwise representational state with a directional effect on the model's decision.

Which native computation reads it out (Figure 1B)? In Section 5, we identify a sparse attention route capable of transferring that decision. Bidirectional interchange of prediction-position head residual updates localizes the route to 16 heads in the three smaller models and 32 in Llama-70B; matched channel restorations associate query semantics with Q, support semantics with K, and temporary-label binding with V. We then ask what class-level quantity is available to these heads. Writing $\bar{\mathbf{k}}_{h,c}(p)$ for the within-prompt mean of head h's actual support-label keys of semantic class $c \in \{0, 1\}$, define its native prototype margin

$$m_h(p) = d^{-1/2} \mathbf{q}_h(p)^\top \left[\bar{\mathbf{k}}_{h,1}(p) - \bar{\mathbf{k}}_{h,0}(p)\right], \tag{1}$$

where d is the head dimension. These key-space prototypes are distinct from the contextual word-space means $\boldsymbol{\mu}_c^\ell$ manipulated upstream: they are operands of the native attention readout.

Does the manipulated representation reach this same readout (Figure 1B)? The intervention acts on contextual word-representation means, whereas Equation 1 reads label-position keys in selected heads. We therefore follow the intervention from the former into the latter. Holding each head's query fixed, let $\bar{\mathbf{k}}_{h,c}(p; \beta)$ denote its support-label key prototype under the word-space intervention and define

$$m_h(p; \beta) = d^{-1/2} \mathbf{q}_h(p; 0)^\top \left[\bar{\mathbf{k}}_{h,1}(p; \beta) - \bar{\mathbf{k}}_{h,0}(p; \beta)\right]. \tag{2}$$

If the selected route tracks the manipulated class representation, this margin should weaken as the contextual prototypes approach one another, vanish near their coincidence, and reverse as their di-

rections exchange. Empirically, we observe that this margin varies approximately linearly with the signed exchange coordinate $1-2\beta$, crossing near zero when the contextual class-mean directions coincide. Across the four models, affine fits over the intervention trajectory have median selected-head R^2 values of .933–.966, and their fitted values at the midpoint are near zero. Thus a one-parameter manipulation of the class-level representational state produces an approximately proportional, sign-reversing response in the native quantity used by the localized readout.

Does the prototype comparison capture the route's function (Figure 1B)? We test this in two complementary ways. First, directly exchanging the two key prototypes in the selected heads continuously reverses the model's prediction while preserving the within-class key residuals. The relative position of the two key prototypes is therefore causally consequential. Second, native attention has access to every individual support-label key, so a useful class-mean contrast need not retain its computation. Replacing every selected-head support-label key by its clean within-prompt class mean removes within-class key-score variation. Retaining the clean query, values, and non-label keys, we recompute and patch the head output into the clean forward pass; this simplification retains accuracy and 93.8–97.2% agreement with clean predictions.

Do the same margins explain ordinary decisions (Figure 1C)? On unpatched prompts, a fixed sparse combination of the native prototype margins,

$$\widehat{m}(p) = \sum_{h \in \mathcal{H}} \alpha_h m_h(p), \tag{3}$$

predicts the model's signed native margin $m(p) = \mathrm{logit}(\pi_p(1) \mid p) - \mathrm{logit}(\pi_p(0) \mid p)$ where $\pi_p : \{0, 1\} \to \{a, b\}$ is the arbitrary class-to-label mapping. The globally selected heads are fixed per model. On held-out semantic groups and prompts, Equation 3 achieves $R^2 = .896$–$.955$ and 93.4–95.2% agreement with native predictions.

2 Related Work

Mechanistic accounts of in-context classification. Prior work has identified attention-mediated mechanisms for matching demonstrations and routing their labels. Induction-head analyses establish attention-based matching and copying as a mechanism for in-context behavior (Olsson et al., 2022; Crosbie & Shutova, 2025). Label-token analyses show that demonstration information is accumulated at label positions (Wang et al., 2023b). Function vectors act as compact causal representations that trigger specific task procedures across contexts (Todd et al., 2024). Most directly, Yu & Ananiadou (2024) causally implicate sparse heads under arbitrary labels and associate query–key interactions with semantic matching and value–output pathways with label transport. Cho et al. (2025a) identify a multi-stage ICL circuit in which demonstration semantics are incorporated into label-token representations and subsequently retrieved, and their attention analysis includes query–class score contrasts. These works establish important precedents for sparse causal ICL heads and for the functional roles of queries, keys, and values in semantic matching and temporary-label routing. Our contribution is not this functional division itself; we use the localized native route to connect a manipulable within-prompt class state to the computation that reads it out.

Contextual geometry and prototype accounts. Classical prototypical networks established that class centroids in an embedding space serve as a simple yet effective decision rule for few-shot learning (Snell et al., 2017). Extending this geometric perspective to language models, Cho et al. (2025b) demonstrate that demonstrations induce linearly separable intra-class clusters in an LLM's final hidden states, allowing external nearest-centroid classifiers to significantly outperform standard token-probability decoding. Further, recent work shows that representational geometry is reorganized across layers and becomes increasingly aligned with task structure thanks to attention heads (Park et al., 2025; Yang et al., 2025). The closest precedent is Xiong et al. (2026), who compare prototype and exemplar accounts, derive a prototype-margin expression from normalized final-layer representations, and calibrate it to native logit differences. Related kernel-regression accounts retain explicitly exemplar-weighted retrieval (Han et al., 2025). Thus, both class-level geometric descriptions and exemplar-level retrieval mechanisms have substantial precedent. Our contribution is to connect these views: contextual class geometry is not only predictive of the decision, but is causally read out through class-mean keys in a sparse set of attention heads.

Appendix A more broadly discusses related work.

3 SETUP

We study balanced binary semantic classification in LLMs. Each prompt contains twenty semantic support demonstrations (w_i, y_i) drawn equally from two semantic classes, followed by an unlabeled test word w_{test}. We distinguish semantic class $c_i \in \{0, 1\}$ from the temporary displayed label $y_i \in \{a, b\}$; the latter is assigned arbitrarily within each classification problem. For prompt p, let

$$m(p) = \text{logit}(\pi_p(1) \mid p) - \text{logit}(\pi_p(0) \mid p)$$

denote the native signed output margin where $\pi_p : \{0, 1\} \to \{a, b\}$ is the arbitrary class-to-label mapping. The language model predicts $\hat{y}_{\text{test}} = \pi_p(1)$ if $m(p) > 0$ and $\pi_p(0)$ otherwise; Appendix B.1 shows that this matches natural next-token prediction on our data. At decoder layer ℓ, $\mathbf{h}_i^\ell$ denotes the representation of semantic word-token i, and $\mathbf{z}_i^\ell$ its RMS-normalized, position-centered form. RMS normalization matches the state seen by attention, while position centering removes the mean component shared across prompts at each position. Full notation appears in Appendix B.2. We use four open-weight LLMs: Llama-3.1-8B and 70B (Grattafiori et al., 2024), Qwen2.5-7B (Qwen Team et al., 2024), and Mistral-NeMo-12B (Mistral AI, 2024); model details are in Appendix B.3.

Controlled semantic tasks. We construct semantic classes from WordNet (Miller, 1995) and, for every pair of classes, select eleven words per class so that the true-class centroid separation in input-token embedding space is approximately the expected separation under a random balanced repartition of the same words. The median relative deviation from this null centroid-separation baseline is 0.0% for every model. Thus the construction suppresses this particular input-centroid cue without affecting pretrained semantic information. Each selected pair yields 22 balanced classification problems and twenty demonstrations per prompt, so that each element of the classes takes the role of/appear as the query. Dataset construction, optimization, dataset sizes, and baseline accuracies are given in Appendix B.4.

4 SEMANTIC PROTOTYPE GEOMETRY

An accurate classifier on hidden representations establishes that class information is available, but not that the model uses that information in its own decision. We first characterize the class-level representation that develops within a prompt, then test its behavioral role by continuously changing the relative directions of the two support-class means.

4.1 CLASS STRUCTURE DEVELOPS IN CONTEXTUAL WORD REPRESENTATIONS

The input construction in Section 3 matches the true-class centroid separation to a random-partition baseline. It controls that particular input statistic, rather than removing pretrained semantic information. Does contextual processing produce a more useful class-level representation? In contrast to prior work on emergent geometry in in-context classification (Yang et al., 2025; Cho et al., 2025a; Xiong et al., 2026), we study geometry that develops within individual prompts, closer in spirit to Park et al. (2025).

Let $c_i \in \{0, 1\}$ denote the semantic class of support word i. Using the RMS-normalized, position-centered word representations $\mathbf{z}_i^\ell$ defined in Section B.2, we form within-prompt prototypes

$$\boldsymbol{\mu}_c^\ell(p) = \frac{2}{n} \sum_{i : c_i = c} \mathbf{z}_i^\ell(p), \qquad m_{\text{proto}}^\ell(p) = (\mathbf{z}_{\text{test}}^\ell(p))^\top [\boldsymbol{\mu}_1^\ell(p) - \boldsymbol{\mu}_0^\ell(p)]. \tag{4}$$

The sign of m_{proto}^ℓ predicts the test word's semantic class; the prompt's binding determines the corresponding output label. These word-space means describe the contextual representation, not yet the model's native attention readout.

We measure semantic alignment as the probability that a word representation has a larger inner product with an earlier same-class word than with an earlier opposite-class word, averaged over prompts. Its chance reference is 0.5. At the final test position, alignment rises from 51–53% at the input to 67–70% in later layers; later word positions, which have seen more demonstrations, generally attain stronger alignment (Figure 2, top). Because a demonstration word precedes its own displayed label, its representation cannot obtain this alignment by directly reading that label.

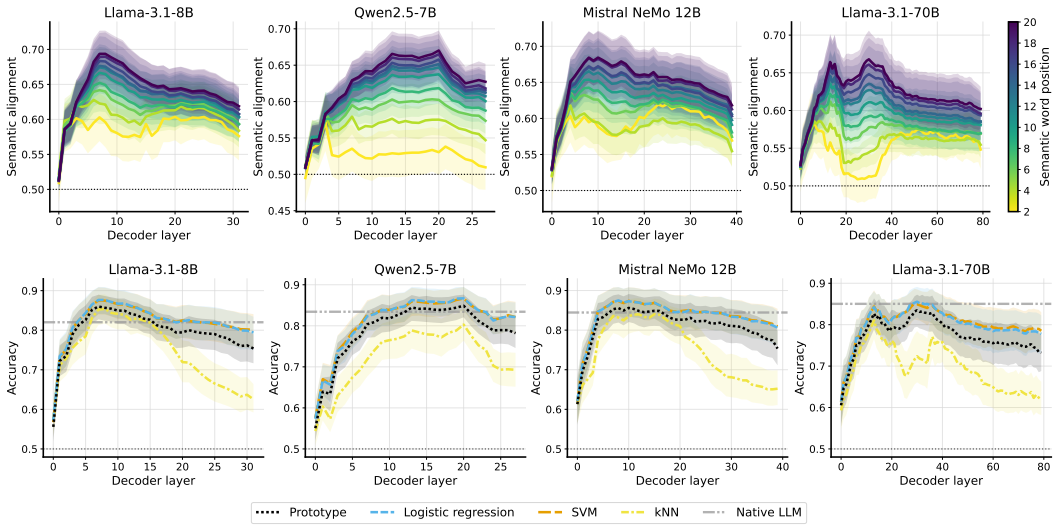


Figure 2: **Contextual processing makes the semantic classes more geometrically accessible. Top:** alignment with earlier same-class rather than opposite-class words across layers and word positions. Later positions have observed more demonstrations. **Bottom:** test-word accuracy of classifiers fitted separately to each prompt's support representations, including the prototype rule in Equation 4. The dashed baseline is the full model. Shading shows 95% confidence intervals. These probes measure representational information; they do not replace/intervene on native computation.

For predictive utility, we fit logistic regression, a linear SVM, and k-nearest neighbors separately to each prompt's twenty supports and evaluate its unlabeled test word; details in Appendix D.2. The prototype rule predicts class 1 when $m^{\ell}_{\text{proto}} > 0$ and class 0 otherwise. It is competitive with the other classifiers and the native LLM (Figure 2, bottom). Hyperparameters for the other classifiers are selected by pooled peak query accuracy. The effect of position-centering on these conclusions is discussed in Appendix B.5. These probes establish accessible class information in the prototype geometry. Next, we ask: Does the native decision actually depend on it?

4.2 Changing prototype directions changes the decision

We manipulate the relative directions of the contextual class means while retaining the internal structure of each class's support hidden states. We run a clean forward pass on every prompt, and record $\mathbf{u}^{\ell}_{c}(p) = \boldsymbol{\mu}^{\ell}_{c}(p)/\|\boldsymbol{\mu}^{\ell}_{c}(p)\|_2$ and the angle $\theta^{\ell}(p)$ between the two directions. Spherical interpolation then specifies the β-intervened target directions

$$\mathbf{u}^{\ell,\beta}_{c} = \frac{\sin((1-\beta)\theta^{\ell})}{\sin\theta^{\ell}}\mathbf{u}^{\ell}_{c} + \frac{\sin(\beta\theta^{\ell})}{\sin\theta^{\ell}}\mathbf{u}^{\ell}_{1-c}, \qquad 0 \le \beta \le 1. \tag{5}$$

At $\beta = 0$ the directions are clean, at $\beta = \frac{1}{2}$ they coincide, and at $\beta = 1$ they are exchanged. Coincidence refers to directions, not equality of the prototype magnitudes.

In the intervened forward pass, a common orthogonal rotation is applied to all current support representations in each class, $\widetilde{\mathbf{h}}^{\ell}_{i} = R^{\ell,\beta}_{c}\mathbf{h}^{\ell}_{i}$ for $c_i = c$, so that the resulting position-centered class mean $\widetilde{\boldsymbol{\mu}}^{\ell}_{c}(p)$ points along the target direction $\mathbf{u}^{\ell,\beta}_{c}$. Each rotation acts in a two-dimensional subspace and preserves hidden-state norms and within-class inner products. We apply this intervention at every layer, rotating the current activations toward the β-intervened clean-reference directions to sustain the target configuration throughout the forward pass. The position-centering baseline is accounted for when constructing the rotation; Appendix C.1 gives the construction.

If the relative class directions $\mathbf{u}^{\ell}_{0}$ and $\mathbf{u}^{\ell}_{1}$ influence the decision, moving them toward exchange should weaken and then reverse the clean-target preference. This is what the aggregate trajectories show (Figure 3). Accuracy decreases from 82–85% in the clean configuration to approximately chance near $\beta = \frac{1}{2}$ and to 14–22% at $\beta = 1$; the target-signed output margin $m_{\text{target}}(p) = t(p)m(p)$, with $t(p) = +1$ for target class 1 and -1 for target 0 changes from posi-

tive to negative. Thus a sustained, norm-preserving reorientation of the class representations has a directional causal effect on the native decision.

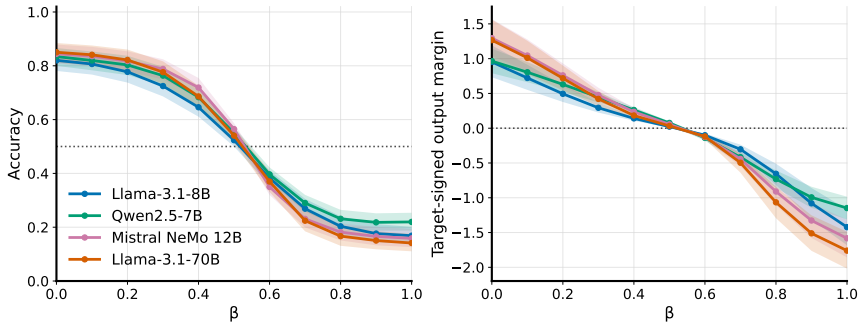


Figure 3: **Exchanging contextual prototype directions reverses the output preference.** Left: accuracy relative to the original target under the sustained word-space intervention. Right: the output margin oriented toward that same target. The two class-mean directions move from clean ($\beta = 0$), through coincident directions ($\beta = \frac{1}{2}$), to exchanged ($\beta = 1$). Curves show all four models; shading shows 95% confidence intervals. This intervention acts on support-word hidden states at every layer, not directly on attention keys.

Could these aggregate curves conceal abrupt or heterogeneous responses within individual prompts? We therefore also examine each prompt's output trajectory. For each prompt, we fit

$$m_{\text{target}}(p; \beta) \approx a_p + c_p \beta \tag{6}$$

across the 11 intervention strengths. Prompt-level affine fits confirm that the aggregate reversal is typical rather than driven by averaging: across the four models, median R^2 ranges from .882 to .927 and median slopes are negative from -2.9 to -2 (Appendix C.2). The remaining question is which native attention quantities respond to this manipulated representational state.

5 Prototype Prediction Circuit

Which native attention route reads out this class-level representation? We identify a sparse prediction circuit and test its response to the word-space intervention, its dependence on key prototypes, and its quantitative relationship to ordinary output.

5.1 Localizing the Route and its Inputs

We rank attention heads by a counterfactual-gradient score explained in Appendix B.6; essentially, it favors heads whose outputs change under a label swap in directions that affect the target margin.

We then test whether the top-ranked heads are sufficient to control the language model's prediction using bidirectional patching. Starting from a label-swapped forward pass, we restore clean writes for the top-k heads; conversely, starting from a clean forward pass, we insert label-swapped writes for the same heads. To test generalization across semantic groups, we select heads in a leave-two-groups-out loop, excluding both groups of each evaluated semantic pair from head selection. The rescue and corruption curves change mainly at early ranks (Figure 4, top), showing that few selected heads can move behavior between the two label-binding solutions. These interventions therefore localize a causal route to the decision. Subsequent analyses use one fixed head set per model selected across the full prompt set: the top-16 heads for Llama-3.1-8B, Qwen2.5-7B, and Mistral-NeMo-12B, and the top-32 for Llama-3.1-70B (detail in B.6). We denote this set by $\mathcal{H}$.

We next ask what information is carried by the query (Q), key (K), and value (V) projections in the selected heads. Matched counterfactual prompts independently alter the test word (T), support-class assignment (S), or class-to-label binding (L) in the prompt; each change reverses the correct output label, details in Appendix B.7. Starting from each counterfactual prompt, we restore exactly one clean projection—query, key, or value—in the selected heads $\mathcal{H}$, only for the attention update at the final prediction position, and measure how much this restores the original prediction. Recovery is selective: T is restored primarily through Q, S through K, and L through V (Figure 4, bottom).

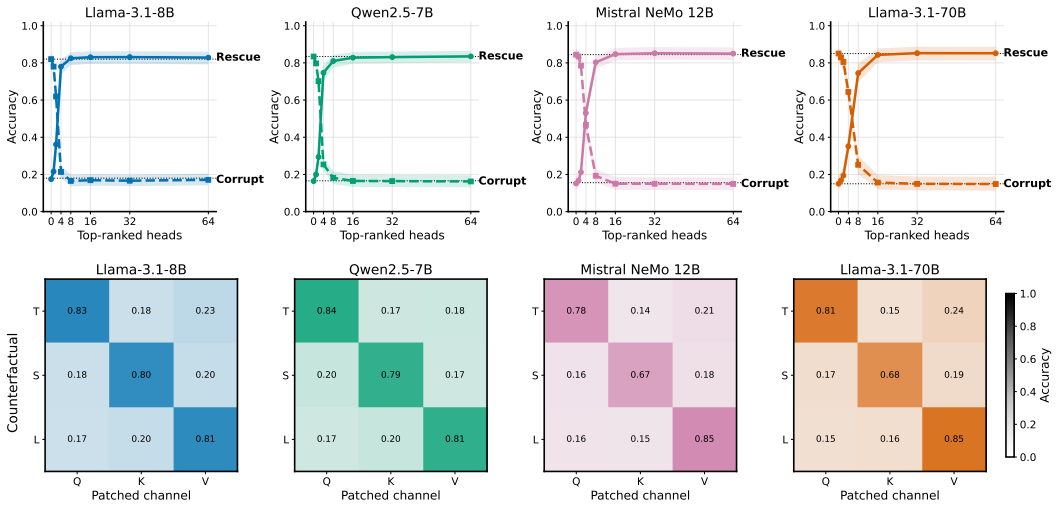


Figure 4: **A sparse route transfers the label decision and separates its inputs.** Top: selected-head writes rescue a label-swapped run and corrupt a clean run. Bottom: channel restoration associates test semantics with Q, support semantics with K, and temporary-label binding with V.

This characterizes $\mathcal{H}$ as causal heads that ingest test-word, support-class, and label-binding information separately across query, keys, and values, consistent with prior accounts of semantic matching through query/key and label transport through value (Yu & Ananiadou, 2024; Cho et al., 2025a).

5.2 The word-space intervention reaches native key prototypes

Does the manipulated word representation reach the selected route in a quantitatively predictable way? Let $y_i \in \{a, b\}$ denote support i's displayed label, distinct from its semantic class $c_i \in \{0, 1\}$. Although the n demonstrations contribute $4n$ support-token keys, we focus on the n label-position keys, motivated by prior work showing that demonstration information is consolidated and retrieved at label positions (Wang et al., 2023b; Cho et al., 2025a; Yu & Ananiadou, 2024). We treat these keys as candidate operands of the selected-head readout; Section 5.3 tests this choice causally,

$$\bar{\mathbf{k}}_{h,c}(p) = \frac{2}{n} \sum_{i:c_i=c} \mathbf{k}_{h,i}(p), \qquad m_h(p) = d^{-1/2} \mathbf{q}_h(p)^\top \left[\bar{\mathbf{k}}_{h,1}(p) - \bar{\mathbf{k}}_{h,0}(p)\right]. \tag{7}$$

Grouping by displayed label incorporates the prompt's class-to-label binding. These are actual attention queries and label-position keys, not the centered word representations used in Section 4.

We rerun the sustained *word-space* intervention of Equation 5. At each selected head we measure the intervened label-key means, but evaluate them against the fixed clean query,

$$m_h(p;\beta) = d^{-1/2} \mathbf{q}_h(p;0)^\top \left[\bar{\mathbf{k}}_{h,1}(p;\beta) - \bar{\mathbf{k}}_{h,0}(p;\beta)\right], \qquad m_h(p;0) = m_h(p). \tag{8}$$

Holding the query fixed isolates the key-side response: it diagnoses how the intervened keys change relative to the same query, rather than introducing an additional query intervention. Because these per-head margins are defined in semantic-class space, before the prompt-specific a/b binding is applied, no a/b-target alignment is needed.

A proportional response to the signed exchange coordinate $1 - 2\beta$ has two testable features: an approximately linear trajectory, an approximately zero midpoint. We fit for every prompt–head pair

$$m_h(p;\beta) \approx b_{h,p} + c_{h,p}\beta, \qquad \nu_{h,p} := b_{h,p} + \tfrac{1}{2} c_{h,p}, \tag{9}$$

over eleven intervention strengths. For each selected head we summarize R^2 and $\nu_{h,p}$ by their medians across prompts; the distributions across selected heads are shown in Figure 5.

Across models, the median selected-head R^2 is .933–.966 and the median fitted midpoint is close to zero ($-.009$ to .087), yielding

$$m_h(p;\beta) \approx \kappa_h(p)(1 - 2\beta). \tag{10}$$

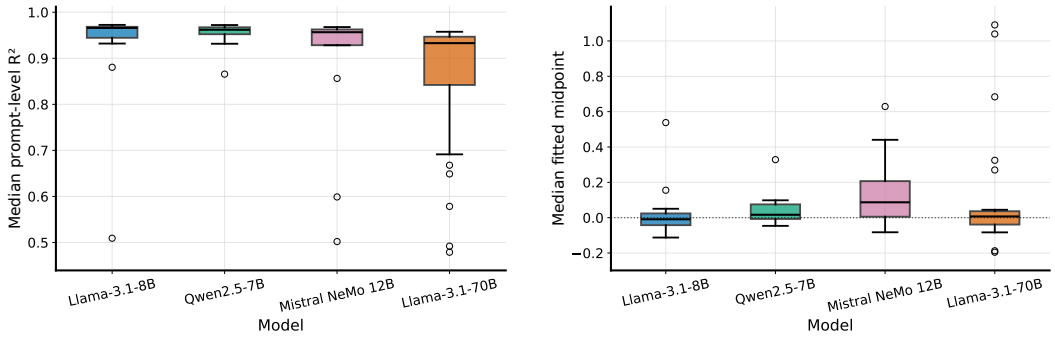


Figure 5: **Native key margins track the manipulated word-space geometry.** Left: median prompt-level R^2 for each selected head's affine response. Right: median fitted midpoint $\nu_{h,p}$. High linearity and midpoints near zero support an approximately proportional response.

The prompt-specific orientation and head scale are absorbed in $\kappa_h(p)$. This relation summarizes the affine-fit linearity and midpoint analyses. It connects the controlled movement of contextual class directions to the native key comparison of the selected attention route.

5.3 Testing between-class means and within-class key detail

The tracking result in Section 5.2 does not establish that the candidate label-position key means are functionally important. Moreover, prior mechanistic accounts retain retrieval of individual demonstrations (Yu & Ananiadou, 2024; Cho et al., 2025a), whereas prototype classifiers aggregate supports by class (Snell et al., 2017). We therefore intervene directly on these keys in two complementary ways: change their class means while retaining within-class residuals, then retain their class means while removing those residuals.

Changing the key prototypes. As before, we first run a clean forward pass and record the clean key means $\bar{\mathbf{k}}^{\mathrm{cl}}_{h,0}$ and $\bar{\mathbf{k}}^{\mathrm{cl}}_{h,1}$. For intervention strength $\gamma \in [0,1]$, define the target means

$$\begin{aligned}\bar{\mathbf{k}}^{\mathrm{tar}}_{h,0}(\gamma) &= (1-\gamma)\bar{\mathbf{k}}^{\mathrm{cl}}_{h,0} + \gamma\bar{\mathbf{k}}^{\mathrm{cl}}_{h,1},\\ \bar{\mathbf{k}}^{\mathrm{tar}}_{h,1}(\gamma) &= (1-\gamma)\bar{\mathbf{k}}^{\mathrm{cl}}_{h,1} + \gamma\bar{\mathbf{k}}^{\mathrm{cl}}_{h,0}.\end{aligned} \tag{11}$$

We then run an intervened forward pass. At each selected head, only for its attention update to the final prediction position, we translate the *current* label-position keys according to

$$\widetilde{\mathbf{k}}_{h,i} = \bar{\mathbf{k}}^{\mathrm{tar}}_{h,c_i}(\gamma) + \left(\mathbf{k}_{h,i} - \bar{\mathbf{k}}_{h,c_i}\right). \tag{12}$$

This sets the class means to the targets while preserving each current within-class residual. At $\gamma = 0$ the computation is clean, at $\gamma = \frac{1}{2}$ the key means meet at their midpoint, and at $\gamma = 1$ they exchange. Unlike the post-RMS normalization word-space intervention, which had to preserve representation norms and therefore used rotations, the key-space intervention has no such constraint, so we directly translate the prototypes. Figure 6 shows that exchanging these key means progressively reverses predictions in all four models while individual within-class key residuals remain available. The between-class arrangement of the key prototypes is therefore causally consequential.

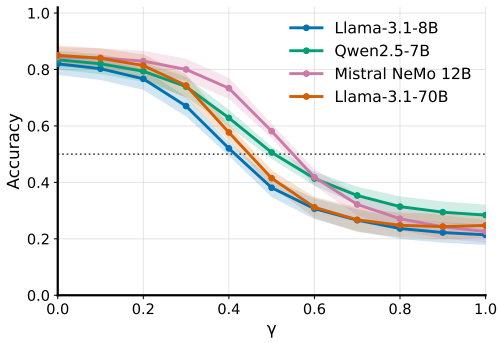


Figure 6: **Exchanging class-mean keys reverses predictions.** The intervention preserves within-class key residuals; $\gamma = 0, \frac{1}{2}, 1$ denote clean, coincident, and exchanged means.

Removing within-class key variation. In a separate simplification, we replace each support-label key by its clean within-prompt class mean, $\widetilde{\mathbf{k}}_{h,i} = \bar{\mathbf{k}}_{h,c_i}$. We apply this intervention to all heads in $\mathcal{H}$, only for their attention update to the final prediction position. The two class prototypes are preserved, but within-class key-score variation is eliminated.

Task accuracy remains unchanged under this intervention after replacement (differs by up to 0.4 percentage points; Table 1). More importantly, agreement with the individual clean predictions is

93.8–97.2% and R^2 to the clean signed margin is .943–.989. The simplification thus retains both most decisions and much of their margin variation, rather than merely matching aggregate accuracy.

Table 1: **Class-mean key replacement preserves native behavior; the same prototype margins predict ordinary output.** The middle group is a clean-run intervention; the right group is a predictor evaluated on unpatched prompts from held-out semantic groups. Accuracy is a fraction, agreement is a percentage of clean predictions, and R^2 is relative to the clean signed candidate-logit margin. Full 95% confidence intervals are in the Appendix Tables 6a–6c.

	Clean	Class-mean label keys			Held-out margin readout		
Model	Acc.	Acc.	Agree. (%)	R^2	Acc.	Agree. (%)	R^2
Llama-3.1-8B	.820	.816	93.9	.961	.830	93.4	.919
Qwen2.5-7B	.834	.832	93.8	.943	.838	93.5	.896
Mistral-NeMo-12B	.844	.840	96.2	.981	.850	95.2	.955
Llama-3.1-70B	.850	.851	97.2	.989	.861	94.6	.932

After class-mean replacement, balanced class sizes give the exact attention-level relation

$$\log \frac{A_{h,1}}{A_{h,0}} = \frac{\mathbf{q}_h^\top(\bar{\mathbf{k}}_{h,1} - \bar{\mathbf{k}}_{h,0})}{\sqrt{d_h}} = m_h(p) \tag{13}$$

across the n label-token keys k_i. Thus, the surviving query–prototype margin determines the relative attention mass assigned to the two support classes; the derivation is given in Appendix C.5.

5.4 The same prototype margins predict ordinary output

How much of the language model's output is captured by these native per-head margins $m_h(p)$ across selected heads in $\mathcal{H}$? On clean prompts, we fit a zero-intercept readout to the per-head margins

$$\widehat{m}(p) = \sum_{h \in \mathcal{H}} \alpha_h m_h(p), \qquad m(p) = \mathrm{logit}(\pi_p(1) \mid p) - \mathrm{logit}(\pi_p(0) \mid p). \tag{14}$$

For each evaluated semantic pair, coefficients are fit only on prompts containing neither semantic group (Appendix D.1). The held-out readout achieves $R^2 = .896$–$.955$ and 93.4–95.2% agreement with native predictions (Table 1, held-out margin readout). Thus, the model's output margin is well approximated by a weighted sum of the selected heads' prototype margins. Relatedly, Xiong et al. (2026) show that final-layer prototype margins track native outputs. Here, we instead compute the margins directly from query–key interactions in causally localized heads, tying the prototype comparison to the model's native readout.

6 Conclusion

We study pretrained LLMs performing binary semantic classification from in-context demonstrations, where the semantic classes are latent and class-centroid separation in the input embeddings is suppressed. In this setup, we identify a class-level decision variable for in-context semantic classification and trace how it is read out by the model's native computation. Contextual processing constructs within-prompt class means whose relative directions causally control the model's decision. A sparse set of attention heads reads this state through query–key comparisons between class-mean keys: changing those means reverses predictions, whereas removing much of the within-class key variation preserves them. The same native prototype margins also predict ordinary output margins across held-out semantic groups. These results provide, for the specific classification task we study, a causal and quantitative link from an internal class-level word representation to a sparse set of native readout heads. Together, this connects the geometry that emerges during contextual processing to the native computation that turns that geometry into a prediction.

Two limitations of our work define clear directions for future work. First, our experiments use a controlled binary classification setting; whether the same causal prototype geometry and readout mechanism extends to more realistic natural-language ICL tasks remains open. Second, we characterize how the emergent class geometry is read out, but not how that geometry is constructed during contextual processing. Explaining the upstream mechanism that builds these class-level representations is therefore an important next step.

AI USE STATEMENT

We used generative AI tools for literature search and synthesis; manuscript organization, editing, and readability improvements; figure refinement; software-code drafting, debugging, and review; consistency checks across experimental results and supporting materials; and feedback on scientific presentation. Where AI tools assisted with code, all generated or modified code was reviewed and tested by the authors. We did not use generative AI to generate datasets, formulate mathematical claims or proofs, determine experimental results, or independently interpret the results reported in this paper. The experiments and primary analyses were conducted by the authors, and all AI-assisted outputs were reviewed and verified by the authors. The authors take responsibility for the complete contents of the paper.

ETHICS STATEMENT

This work studies mechanistic properties of publicly available language models on synthetic semantic-classification prompts derived from WordNet. It does not involve human subjects, private or personally identifiable data, or deployment of models in high-stakes settings. We are not aware of additional ethical risks specific to the experiments beyond those associated generally with research on large language models.

REPRODUCIBILITY STATEMENT

Appendix B specifies the model checkpoints, prompt and dataset construction, input-centroid control, counterfactuals, and head-selection procedure. Appendix C gives the intervention protocols, and Appendix D gives the held-out evaluation and diagnostic procedures. The experiments use publicly available model checkpoints and WordNet-derived semantic groups.

REFERENCES

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Hritik Bansal, Karthik Gopalakrishnan, Saket Dingliwal, Sravan Bodapati, Katrin Kirchhoff, and Dan Roth. Rethinking the role of scale for in-context learning: An interpretability-based case study at 66 billion scale. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Hakaze Cho, Mariko Kato, Yoshihiro Sakai, and Naoya Inoue. Revisiting in-context learning inference circuit in large language models. In *International Conference on Learning Representations*, 2025a. URL https://arxiv.org/abs/2410.04468.

Hakaze Cho, Yoshihiro Sakai, Mariko Kato, Kenshiro Tanaka, Akira Ishii, and Naoya Inoue. Token-based decision criteria are suboptimal in in-context learning. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5378–5401, Albuquerque, New Mexico, April 2025b. Association for Computational

Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.278. URL https://aclanthology.org/2025.naacl-long.278/.

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

Joy Crosbie and Ekaterina Shutova. Induction heads as an essential mechanism for pattern matching in in-context learning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5049–5111, 2025.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. Transformer Circuits Thread, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? A case study of simple function classes. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30583–30598, 2022.

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

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *PMLR*, 2024.

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

Chi Han, Ziqi Wang, Han Zhao, and Heng Ji. Understanding Emergent In-Context Learning from a Kernel Regression Perspective. *Transactions on Machine Learning Research*, 2025. URL https://mlanthology.org/tmlr/2025/han2025tmlr-understanding/.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*, 2024.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019.

Artem Kirsanov, Chi-Ning Chou, Kyunghyun Cho, and SueYeon Chung. The geometry of prompting: Unveiling distinct mechanisms of task adaptation in language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1855–1888, 2025.

Jannik Kossen, Yarin Gal, and Tom Rainforth. In-context learning learns label relationships but is not conventional learning. In *International Conference on Learning Representations*, volume 2024, pp. 3209–3280, 2024.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. AtP*: An efficient and scalable method for localizing LLM behaviour to components. *arXiv preprint arXiv:2403.00745*, 2024.

Patrick Lutz, Aditya Gangrade, Hadi Daneshmand, and Venkatesh Saligrama. Linear transformers implicitly discover unified numerical algorithms. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2509.19702.

Patrick Lutz, Themistoklis Haris, Arjun Chandra, Aditya Gangrade, and Venkatesh Saligrama. Symmetry reveals layerwise dynamics: How transformers perform in-context classification. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2604.11613.

Aleksandar Makelov, Georg Lange, Atticus Geiger, and Neel Nanda. Is this the subspace you are looking for? An interpretability illusion for subspace activation patching. In *International Conference on Learning Representations*, 2024.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *Conference on Language Modeling*, 2024.

Douglas L. Medin and Marguerite M. Schaffer. Context theory of classification learning. *Psychological Review*, 85(3):207–238, 1978.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

Mistral AI. Mistral NeMo. https://mistral.ai/news/mistral-nemo, 2024. Blog post, July 18, 2024.

Neel Nanda. Attribution patching: Activation patching at industrial scale. https://www.neelnanda.io/mechanistic-interpretability/attribution-patching, 2023.

Robert M. Nosofsky. Attention, similarity, and the identification–categorization relationship. *Journal of Experimental Psychology: General*, 115(1):39–57, 1986.

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, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022. URL https://arxiv.org/abs/2209.11895.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. What in-context learning "learns" in-context: Disentangling task recognition and task learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8298–8319, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.527. URL https://aclanthology.org/2023.findings-acl.527/.

Core Francisco Park, Andrew Lee, Ekdeep Singh Lubana, Yongyi Yang, Maya Okawa, Kento Nishi, Martin Wattenberg, and Hidenori Tanaka. Iclr: In-context learning of representations. In *International Conference on Learning Representations*, volume 2025, pp. 53258–53284, 2025.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *PMLR*, 2024.

Michael I. Posner and Steven W. Keele. On the genesis of abstract ideas. *Journal of Experimental Psychology*, 77(3):353–363, 1968.

Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. In *International Conference on Learning Representations*, 2024.

Eleanor Rosch. Cognitive representations of semantic categories. *Journal of Experimental Psychology: General*, 104(3):192–233, 1975.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *PMLR*, pp. 3319–3328, 2017.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 407–416, 2024. doi: 10.18653/v1/2024.blackboxnlp-1.25.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron Wallace, and David Bau. Function vectors in large language models. In *International conference on learning representations*, volume 2024, pp. 17282–17333, 2024.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12388–12401, 2020.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 35151–35174. PMLR, 2023. URL https://proceedings.mlr.press/v202/von-oswald23a.html.

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

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. Label words are anchors: An information flow perspective for understanding in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9840–9855, 2023b.

Jerry Wei, Le Hou, Andrew Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, and Quoc Le. Symbol tuning improves in-context learning in language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 968–979, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.61. URL https://aclanthology.org/2023.emnlp-main.61/.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023b.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. Interpretability at scale: Identifying causal mechanisms in Alpaca. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations*, 2022.

Hua-Dong Xiong, Li Ji-An, Robert C. Wilson, Kwonjoon Lee, and Xue-Xin Wei. Large language models reorganize representational geometry during in-context learning. In *Conference on Language Modeling*, 2026. URL https://arxiv.org/abs/2605.28854.

Haolin Yang, Hakaze Cho, Yiqiao Zhong, and Naoya Inoue. Unifying attention heads and task vectors via hidden state geometry in in-context learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 21701–21759, 2025.

Haolin Yang, Hakaze Cho, and Naoya Inoue. Localizing task recognition and task learning in in-context learning via attention head analysis. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 93780–93839, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/974726d56a6ddfac48486e57d5e798e4-Paper-Conference.pdf.

Kayo Yin and Jacob Steinhardt. Which attention heads matter for in-context learning? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *PMLR*, pp. 72428–72461, 2025.

Zeping Yu and Sophia Ananiadou. How do large language models learn in-context? query and key matrices of in-context heads are two towers for metric learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3281–3292, 2024. URL https://aclanthology.org/2024.emnlp-main.192/.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *PMLR*, 2021.

A ADDITIONAL RELATED WORK

Arbitrary labels and class-to-symbol binding. Empirical studies involving arbitrary symbol tuning and randomized labels demonstrate that language models can perform in-context learning by capturing novel input-label mappings—often termed Task Learning—rather than merely relying on pre-trained semantic priors for Task Recognition (Wei et al., 2023a; Pan et al., 2023). While models may struggle to fully override strong pre-training preferences, they actively process and depend on these in-context label relationships (Kossen et al., 2024). Recent mechanistic accounts formalize this functional division, identifying distinct sets of attention heads that either align representations to a task subspace for task recognition or rotate them toward specific target labels for task learning (Yang et al., 2026). Our framework renders this separation explicit at the level of representation geometry: latent semantic class inference operates independently of prompt-specific label binding. Through counterfactual interventions, we establish a causal dissociation where support-class geometry is constructed via query-key (Q/K) matching, while the binding of arbitrary symbols to those classes is executed primarily through value-output (V/O) pathways.

Behavioral studies of in-context learning. In-context learning was popularized by Brown et al. (2020). Min et al. (2022) found that randomizing demonstration labels often changes classification accuracy slightly, whereas Wei et al. (2023b) showed that larger models can override semantic priors when labels are flipped or replaced by semantically unrelated symbols. Our setting is in the latter regime by construction: the displayed labels are arbitrary, clean accuracy is 82–85%, and exchanging the class-to-label binding reverses the output (Section 5.1). Because our decision variable is a difference of two label-token logits, it is related to the label-token calibration problem studied by Zhao et al. (2021); Appendix B.4 shows that in our prompts one of the two labels is always the global top-1 token, so the margin sign coincides with the model's greedy prediction.

Causal localization methodology. Our localization score (Appendix B.7) belongs to the family of gradient-based approximations to activation patching. Activation patching is based on causal mediation analysis (Vig et al., 2020) and causal tracing (Meng et al., 2022). The residual-stream view of attention heads (Elhage et al., 2021), underlies circuit analyses such as Wang et al. (2023a) and automated circuit discovery (Conmy et al., 2023); Zhang & Nanda (2024) and Heimersheim & Nanda (2024) discuss metric choice and the distinction between denoising and noising patches. Attribution patching (Nanda, 2023; Syed et al., 2024) replaces each patch by a first-order term $(o_h^{\mathrm{cl}} - o_h^L)^\top g$ with the gradient taken at a single endpoint, and Kramár et al. (2024) document failure modes of this linearization, including attention saturation and cancellation between effects. Our score averages the linearizations at the clean and label-swapped endpoints, which is the two-point trapezoidal estimate of the integrated-gradients path integral (Sundararajan et al., 2017) along the interpolation between the two runs. We use the score only to rank candidates. Sufficiency of the selected heads is established by exact bidirectional patching: restoring clean writes into the label-swapped run is a denoising test, and inserting label-swapped writes into the clean run is a noising test (Figure 4, top).

Causal abstraction and distributed interventions. The sustained word-space rotation (Section 4.2, Appendix C.1) intervenes on a distributed representation rather than on individual neurons or heads. Interchange interventions (Geiger et al., 2021) and distributed alignment search (Geiger et al., 2024; Wu et al., 2023) also act in rotated bases of the residual stream, but they learn the rotation by optimizing the counterfactual behavior it produces. Makelov et al. (2024) show that optimized subspace patches can change model output by activating a dormant pathway rather than by manipulating the feature the subspace is taken to represent. Our intervention differs in two respects that bear on this concern. First, the rotation plane is not learned: it is fixed by each prompt's clean class-mean directions and the prescribed interpolation target, and no parameter is fitted to the output. Second, the rotation is orthogonal and preserves hidden-state norms and within-class inner products, so it changes only the between-class arrangement named in the hypothesis. This does not by itself identify which downstream components mediate the effect; Sections 5.2–5.3 address that by tracking the intervention into the key prototypes of $\mathcal{H}$ and by intervening on those prototypes directly.

Decodability versus use. A probe that recovers a variable from hidden states establishes that the information is present, not that the model uses it (Hewitt & Liang, 2019; Belinkov, 2022). Amnesic probing (Elazar et al., 2021), built on iterative nullspace projection (Ravfogel et al., 2020), tests use

by removing a probed property and measuring the behavioral change. The classifiers in Figure 2 are probes in this sense. Section 4.2 supplies the interventional test with a stronger prediction than removal: if the model uses the class-mean directions, exchanging them should not merely degrade accuracy but reverse the sign of the target-oriented margin, which is what Figure 3 and Table 5 show.

In-context learning mechanisms and representation geometry. Beyond induction heads, Hendel et al. (2023) show that ICL compresses demonstrations into a task vector that can be patched into zero-shot runs, and Yin & Steinhardt (2025) find that few-shot ICL in larger models depends mainly on function-vector heads rather than induction heads. Bansal et al. (2023) find that a large fraction of attention heads in a 66B-parameter model can be removed with little loss of ICL accuracy, consistent with the sparse routes we localize. On the geometric side, Kirsanov et al. (2025) analyze how prompting reshapes the geometry of category representations and find that demonstrations and instructions adapt models through distinct mechanisms. The linear representation hypothesis (Park et al., 2024) and evidence of linear structure for truth values (Marks & Tegmark, 2024) motivate treating class-mean directions as meaningful coordinates of the representation. In small transformers trained on in-context classification, Chan et al. (2022) identify data properties that favor in-context over in-weights learning, and Reddy (2024) link the abrupt emergence of in-context classification to induction-head formation. Our analysis concerns pretrained LLMs and a geometry that forms within individual prompts, and it connects that geometry to the key-space operands of the localized heads.

Algorithmic accounts and the one-step gradient connection. A theoretical line studies which learning algorithms transformers can implement in context (Garg et al., 2022; von Oswald et al., 2023; Ahn et al., 2023), alongside Bayesian accounts of ICL (Xie et al., 2022). Beyond existence constructions, recent work extracts the algorithm a trained model actually runs. Lutz et al. (2025) show that linear transformers trained on masked-block matrix completion implicitly discover a unified iterative solver. Closest to our setting, Lutz et al. (2026) study in-context multi-class linear classification in softmax transformers trained from scratch; by enforcing feature- and label-permutation equivariance at every layer, they extract an explicit depth-indexed update rule in which attention formed from mixed feature–label Gram structure jointly updates the training points, labels, and test probe. Those models are trained on synthetic tasks where the full algorithm can be identified; here we study pretrained LLMs on natural-language classification and identify the readout stage of the computation. The prototype rule has a direct link to this line. For supports (z_i, r_i) with $r_i \in \{-1, +1\}$ and squared loss, one gradient step from $w = 0$ with step size η gives $w_1 = \frac{\eta}{n} \sum_i r_i z_i$. For balanced classes and the class means of Equation 4, this equals $\frac{\eta}{2}(\mu_1 - \mu_0)$, so $w_1^\top z_{\text{test}}$ is proportional to m_{proto}. The one-step linear-attention constructions of von Oswald et al. (2023) therefore compute a prototype margin in their feature space when applied to balanced binary classification. The circuit we identify is not of this form, since it uses softmax attention over label-position keys. Equation 13 shows, however, that after class-mean replacement the query–prototype margin determines the log ratio of attention mass on the two classes, which is the softmax counterpart of the same class-mean comparison.

Prototype and exemplar models of categorization. The prototype–exemplar distinction originates in cognitive psychology. Prototype models classify by similarity to an abstracted central tendency of each class (Posner & Keele, 1968; Rosch, 1975); exemplar models classify by summed similarity to stored instances (Medin & Schaffer, 1978; Nosofsky, 1986). Softmax attention over individual support keys has the form of an exemplar computation, which makes it informative that replacing each support-label key by its class mean in the selected heads retains 93.8–97.2% agreement with clean predictions (Section 5.3). For these tasks, this indicates that the exemplar-level key detail available to the selected heads contributes little beyond the class means. It does not show that exemplar information is unused elsewhere in the model.

B Experimental Setup and Reproducibility

B.1 Margin Sign Recovers Top-1 Predictions

We extract the model prediction from the sign of the output margin, recording $\hat{y} = \pi_p(1)$ if $m(p) > 0$ and $\hat{y} = \pi_p(0)$ otherwise. This treats prediction as a binary choice between the two task labels. In practice, however, this simplification does not change the model's prediction: with $n = 20$, either a or b is the global top-1 next token for every prompt (Table 2). Consequently, the margin-sign

Model	Binary acc.	Top-1 acc.	a/b is top-1
Llama-3.1-8B	0.820 [0.781, 0.855]	0.820 [0.781, 0.855]	100.0% [100.0%, 100.0%]
Qwen2.5-7B	0.834 [0.800, 0.866]	0.834 [0.800, 0.866]	100.0% [100.0%, 100.0%]
Mistral NeMo 12B	0.844 [0.808, 0.879]	0.844 [0.808, 0.879]	100.0% [100.0%, 100.0%]
Llama-3.1-70B	0.850 [0.812, 0.885]	0.850 [0.812, 0.885]	100.0% [100.0%, 100.0%]

Table 2: Validation of margin-based predictions. Binary accuracy uses the sign of the a/b margin; top-1 accuracy uses the model's global most likely next token. Values in brackets are 95% confidence intervals.

prediction exactly matches the model's actual top-1 next-token prediction, justifying our use of the margin as the central prediction quantity throughout the paper.

B.2 Notation and prompt format

Each prompt contains $n = 20$ support words w_i, a semantic class $c_i \in \{0, 1\}$ for each support, a displayed temporary label $y_i \in \{a, b\}$, and an unlabeled test word w_{test}. The prompt-specific binding between semantic class and displayed label is arbitrary. We write

$$m(p) = \text{logit}(\pi_p(1) \mid p) - \text{logit}(\pi_p(0) \mid p)$$

for the native signed output margin. Here, $\pi_p : \{0, 1\} \to \{a, b\}$ is the prompt-dependent, arbitrary class-to-label mapping. Let $t(p) = +1$ when the correct label is 1 and -1 when it is 0; the target-oriented margin is $m_{\text{target}}(p) = t(p)m(p)$.

At decoder layer ℓ, let $\mathbf{h}_i^\ell$ denote the residual-stream representation of semantic word i entering the layer, and define

$$\mathbf{x}_i^\ell = \sqrt{d}\,\mathbf{h}_i^\ell / \|\mathbf{h}_i^\ell\|_2, \qquad \mathbf{z}_i^\ell = \mathbf{x}_i^\ell - \mathbf{b}_i^\ell, \qquad \mathbf{b}_i^\ell = \mathbb{E}_p[\mathbf{x}_i^\ell],$$

where the expectation in $\mathbf{b}_i^\ell$ is over all prompts at the same word position using the same LLM. For semantic class c, let

$$\mathcal{I}_c = \{i : c_i = c\}, \qquad \boldsymbol{\mu}_c^\ell = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} \mathbf{z}_i^\ell, \qquad \mathbf{u}_c^\ell = \frac{\boldsymbol{\mu}_c^\ell}{\|\boldsymbol{\mu}_c^\ell\|_2}.$$

At the final prediction position, head h has query $\mathbf{q}_h$ and residual-stream write $\mathbf{o}_h$. At support i's *label-token* position, its key and value are $\mathbf{k}_{h,i}$ and $\mathbf{v}_{h,i}$. For displayed label y, define

$$\mathcal{J}_y = \{i : y_i = y\}, \qquad \bar{\mathbf{k}}_{h,y} = \frac{1}{|\mathcal{J}_y|} \sum_{i \in \mathcal{J}_y} \mathbf{k}_{h,i}, \qquad m_h(p) = d^{-1/2} \mathbf{q}_h(p)^\top \left[\bar{\mathbf{k}}_{h,b}(p) - \bar{\mathbf{k}}_{h,a}(p)\right].$$

Thus μ_c^ℓ is a prototype in contextual word-representation space, whereas $\bar{\mathbf{k}}_{h,c}$ is a prototype in a selected head's support-label key space.

The fixed prompt format is

$$w_1 \to y_1 \backslash \texttt{n} \;\; \cdots \;\; w_n \to y_n \backslash \texttt{n} \;\; w_{\text{test}} \to.$$

All support words and displayed labels are single tokens. With $n = 20$, the fixed format contains exactly $4n + 2$ tokens. For illustration, Table 7 (c) shows two example prompts, with alternating colors marking individual tokenizer tokens.

B.3 Models, datasets, and uncertainty

Table 3 gives the exact checkpoints. Semantic groups are constructed separately for each model because admissible single-token words depend on the tokenizer. Table 4 summarizes the resulting datasets and clean in-context classification accuracy. Unless stated otherwise, intervals in tables and shaded regions are 95% confidence intervals, with uncertainty estimated over the underlying 25–26 semantic groups rather than treating individual prompts as independent units.

Model	Hugging Face checkpoint
Llama-3.1-8B	`https://huggingface.co/meta-llama/Llama-3.1-8B`
Llama-3.1-70B	`https://huggingface.co/meta-llama/Meta-Llama-3.1-70B`
Qwen2.5-7B	`https://huggingface.co/Qwen/Qwen2.5-7B`
Mistral-NeMo-12B	`https://huggingface.co/mistralai/Mistral-Nemo-Base-2407`

Table 3: Model checkpoints used in the experiments.

Model	# semantic groups	# problems	Clean accuracy
Llama-3.1-8B	25	6,600	.820 [.781, .855]
Qwen2.5-7B	25	6,600	.834 [.800, .866]
Mistral-NeMo-12B	26	7,150	.844 [.808, .879]
Llama-3.1-70B	25	6,600	.850 [.812, .885]

Table 4: Semantic class-pair datasets and baseline in-context classification accuracy. Each semantic pair uses eleven single-token words per class before forming the balanced twenty-demonstration prompts. Brackets denote 95% confidence intervals.

B.4 Semantic-group construction and input-centroid control

We construct semantic groups from WordNet (Miller, 1995) noun synsets. Candidate synsets lie at least five levels below a noun-hierarchy root; each group contains lemmas from the synset and its immediate hyponyms. Lemmas are lowercased and retained only when they contain letters only, have no spaces, have at least three characters, and form a single tokenizer token. We exclude the word naming the semantic group itself, require at least 25 admissible words, and remove nested groups by retaining the more specific category.

For each unordered pair of retained semantic groups, words occurring in both groups are removed. We then choose $r = 11$ words per class using the model's raw input-token embeddings. For normalized embeddings x_i^0 and any selected set S, let $\mu(S) = |S|^{-1} \sum_{x \in S} x$. The selected subsets A, B minimize

$$\Delta_{\text{sep}} = \left| \|\mu(A) - \mu(B)\|_2^2 - \mathbb{E}_{\tilde{A},\tilde{B}}\left[\|\mu(\tilde{A}) - \mu(\tilde{B})\|_2^2\right] \right|,$$

where $(\tilde{A}, \tilde{B})$ is a uniformly sampled balanced partition of the same $2r$ selected embeddings. The random-partition expectation is evaluated analytically:

$$\mathbb{E}_{\tilde{A},\tilde{B}}\left[\|\mu(\tilde{A}) - \mu(\tilde{B})\|_2^2\right] = \frac{2}{r(2r-1)} \sum_{x_i^0 \in A \cup B} \|x_i^0 - \mu(A \cup B)\|_2^2.$$

The subset optimization uses greedy within-class swaps with 64 random restarts. At each step we evaluate all exchanges of one selected and one unselected embedding from the same semantic group, apply the exchange that most reduces Δ_{sep}, and stop when no improving exchange remains or after 25 passes. We keep the restart with the smallest final objective.

For each semantic pair we report

$$\delta_{\text{rel}} = 100 \, \frac{\Delta_{\text{sep}}}{\mathbb{E}_{\tilde{A},\tilde{B}}[\|\mu(\tilde{A}) - \mu(\tilde{B})\|_2^2]}.$$

The median δ_{rel} is 0.0% for every model. The 90th percentiles are 4.77% for Qwen2.5-7B, 5.79% for Llama-3.1-8B, 8.26% for Llama-3.1-70B, and 11.50% for Mistral-NeMo-12B; modelwise means range from 1.35% to 4.30%. Figure 7 gives the full distributions.

From each selected semantic pair we construct $2r = 22$ classification problems. In each problem, one word is the test word and one word from the opposite class is removed, leaving ten support words from each class. Across the 22 problems, every word serves once as the test word and once as the removed word. Table 4 gives a overview of the resulting data set for each model. For illustration, Table 7 shows five semantic groups used for Llama-3.1-8B. Panel (a) lists the eligible single-token words in each group. Panel (b) shows the optimized 11-word subsets for five pairings of these groups, together with the resulting deviation from the random-partition baseline.

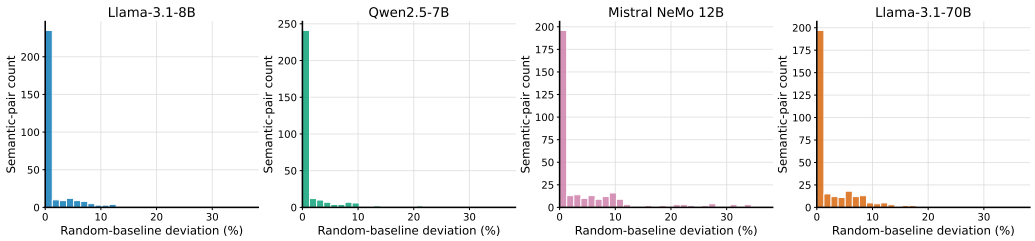


Figure 7: Distribution of the relative deviation δ_{rel} between the optimized semantic centroid separation and the expected separation under balanced random partitions of the same $2r$ input embeddings. A value of 0% indicates exact matching to the random-partition baseline.

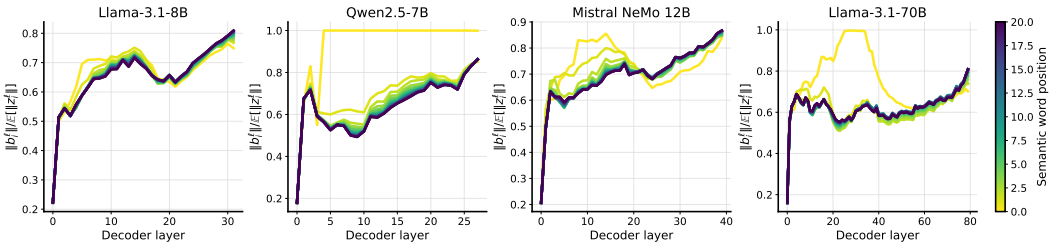


Figure 8: Position-mean magnitude. Norm of the global position-specific mean relative to the mean norm of the corresponding semantic-word representation, $b_i^\ell / \mathbb{E}[z_i^\ell]$, across layers and positions. The position mean constitutes a substantial component of the representation, particularly in later layers.

B.5 Position centering

Semantic-word representations contain systematic information about prompt position (Cho et al., 2025a). We remove the first-order positional component before the geometric analyses. For RMS-normalized representation x_i^ℓ, define

$$z_i^\ell(p) = x_i^\ell(p) - \mathbb{E}_p[x_i^\ell(p)],$$

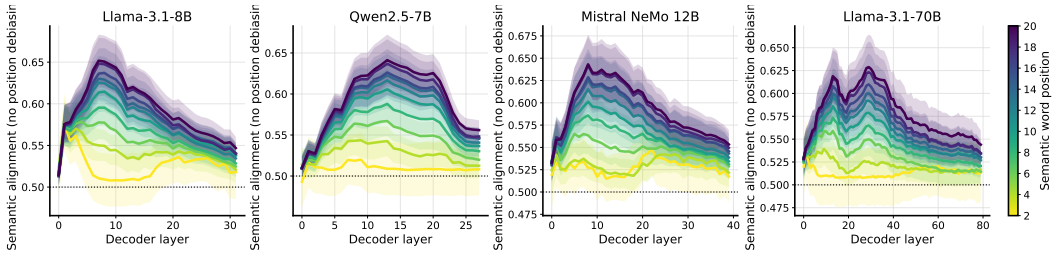
where the expectation is over prompts at the same semantic-word position for the same model. This subtracts the mean representation associated with the position while retaining prompt-specific variation. Higher-order or context-dependent positional information may remain.

The position mean is a substantial component of the raw representation. As shown in Figure 8, its norm is typically more than 50% of the mean representation norm and rises to 80%–86% in the final layer. Thus, position centering removes a large, structured component, but one that is shared across prompts at the same position rather than specific to the semantic content of an individual prompt.

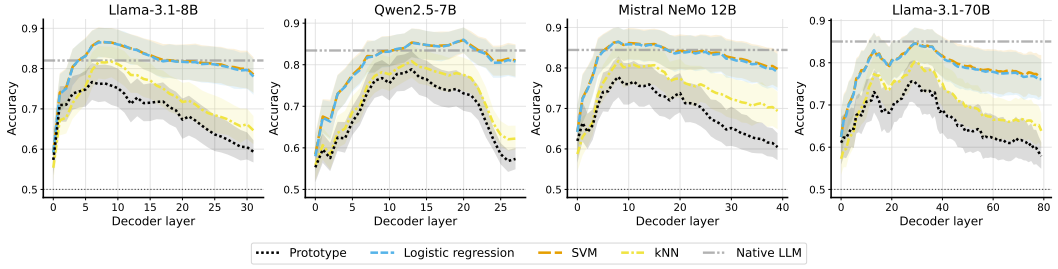
To test the role of position centering, we repeat the three geometric analyses from Section 4 (geometric alignment, layerwise readout, and prototype steering) using the raw normalized representations x_i^ℓ in place of the centered representations $z_i^\ell = x_i^\ell - b_i^\ell$. Equivalently, we omit the subtraction of the position-specific mean b_i^ℓ and define class prototypes directly from the raw states,

$$\mu_c^\ell = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} x_i^\ell.$$

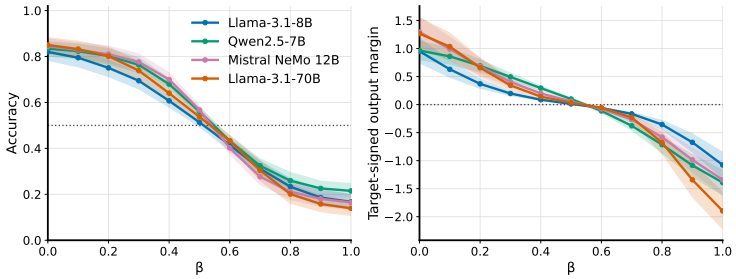
All metrics, classifiers, and interventions are otherwise defined exactly as in the main analysis. Without centering, semantic class remains strongly decodable from word-representations, although the simple prototype classifier is substantially less competitive with learned classifiers (Figure 9b). In contrast, geometric alignment and the prototype intervention is largely unchanged: same-class geometric alignment still emerges across layers (Figure 9a), and progressively reversing the raw-state prototypes produces the same qualitative behavioral reversal as in the main analysis (Figure 9c).



(a) Semantic geometric alignment computed directly from raw normalized word representations, without subtracting the position mean. For each layer and semantic-word position, the plotted quantity is the within-prompt probability that a same-class dot product exceeds an opposite-class dot product (chance $= 0.5$); bands show 95% semantic-group bootstrap intervals.



(b) Held-out query classification using raw normalized semantic-word representations. Semantic class remains strongly decodable, although the prototype classifier is less competitive with learned classifiers.



(c) Prototype intervention using uncentered representations. The intervention produces the same qualitative reversal as in the main analysis, showing that the behavioral effect does not depend on position centering.

Figure 9: Ablation without position centering. All three panels repeat the main geometric analyses using raw normalized representations x_i^ℓ instead of centered representations $z_i^\ell = x_i^\ell - b_i^\ell$. Centering makes the geometric readout cleaner, especially for the prototype classifier, but the qualitative semantic geometry and the prototype-steering effect remain visible without it.

B.6 Counterfactual-gradient head localization

We rank candidate readout heads by combining their response to a label-binding counterfactual with the local sensitivity of the final margin. Let o_h^{cl} and o_h^L be head h's residual-stream writes at the final prediction position in the clean and label-swapped runs (see B.7). Let g_ℓ^{cl} and g_ℓ^L be the gradients of the clean-target-oriented margin with respect to the full attention write at layer ℓ, evaluated at the two endpoints. The localization score is

$$s_h^{\mathrm{loc}} = \mathbb{E}_p\left[(o_h^{\mathrm{cl}} - o_h^L)^\top \frac{g_\ell^{\mathrm{cl}} + g_\ell^L}{2}\right], \qquad h \in \mathcal{H}_\ell.$$

A high score identifies a head whose write changes under the reversed label binding and whose displacement points along a direction that increases the clean-target margin. To test whether the head-selection procedure generalizes beyond the semantic groups used for ranking, Figure 4 (top) uses out-of-fold head selection: heads are ranked using only prompts whose semantic groups are

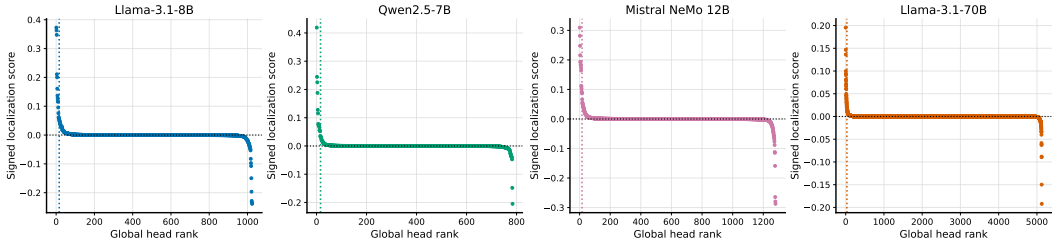


Figure 10: Candidate readout-head rankings. Heads are ordered by the signed counterfactual-gradient score. The dotted marker in the diagnostic is at rank 16; the canonical intervention uses 16 heads for the three smaller models and 32 for Llama-3.1-70B.

excluded from the corresponding evaluation fold. For all subsequent analyses, we instead use a single fixed set of heads selected across the full prompt set.

Figure 10 shows the rank curves. These scores select candidates; the bidirectional interchange experiment in the main text provides the behavioral test of the selected route.

For completeness, Tables 8 list the fixed globally selected head sets used in the downstream analyses. Heads are ordered by the causal selection score computed across the full prompt set; the top 16 heads are retained for the three smaller models and the top 32 for Llama-3.1-70B.

This approach scores the heads by taking the difference between their residual-stream writes in a clean run and a label-swapped run, and then multiplying that displacement by the gradients of the clean-target margin. This isolates heads whose output changes under reversed label binding in a way that actively increases the target margin.

In contrast, Yu & Ananiadou (2024) identify in-context heads without gradients by combining two scores: a causal-tracing ablation, which sets each deep-layer head's parameters to zero and measures the resulting accuracy drop, and a saliency-style direct-effect score, the increase in the predicted label's log probability when the head's output is added to its layer's input and projected through the unembedding. Heads ranked in the top ten under both scores are retained. Our counterfactual-gradient score instead weights each head's clean-versus-label-swapped write displacement by the local sensitivity of the target margin, so it selects heads whose contribution depends on the label binding rather than heads that are important within a single run.

Another commonly used method for head attribution, that departs from both gradient-based counterfactuals and raw probability additions and uses a geometric framework, is called Task Subspace Logit Attribution (Yang et al., 2026). Instead of analyzing clean versus swapped runs, it evaluates heads based on how much of their output concentrates within a specific mathematical subspace spanned by the unembedding vectors of demonstration labels. It further identifies specific heads by measuring the alignment between the head's output and the contrastive direction separating correct from incorrect labels within that projected subspace. This method fundamentally evaluates heads based on their structural alignment and projection magnitude within predefined task subspaces.

B.7 Matched counterfactual prompts

Starting from a clean prompt p, we construct three matched counterfactuals that change one task variable at a time. In p^T, the test word is replaced by the held-out opposite-class word while the supports and label binding remain fixed. In p^C, the same support words are permuted across positions so that each moves to a position previously occupied by a word from the opposite semantic class, while the displayed labels and test word remain fixed. In p^L, every displayed support label is exchanged ($a \leftrightarrow b$), reversing the class-to-label binding while preserving all words and positions. These matched runs provide natural activation donors for the interchange and channel-restoration experiments.

Model	Slope c_p	R^2
Llama-3.1-8B	-2.000 $[-3.489, -0.682]$	.900 [.772, .954]
Qwen2.5-7B	-2.119 $[-3.591, -0.739]$	.882 [.662, .952]
Mistral-NeMo-12B	-2.750 $[-4.591, -1.057]$	.927 [.791, .969]
Llama-3.1-70B	-2.898 $[-4.727, -1.148]$	.913 [.796, .960]

Table 5: The word-space intervention has a graded effect within individual prompts. Each row summarizes the affine fits in Equation 6. Entries are median [first quartile, third quartile] across prompts, *not* confidence intervals. Figure 3 shows the aggregate intervention for all four models.

C Intervention Protocols

The main text uses three distinct experimental regimes. The sustained word-space intervention manipulates contextual class representations (Appendix C.1); the selected-head interventions manipulate or simplify native attention variables (Appendices C.3, C.4); and the held-out readout is fitted and evaluated only on ordinary, unpatched prompts (Appendix D.1). These protocols use different recipients and evaluation targets unless stated otherwise.

C.1 Sustained contextual prototype-direction intervention

For each clean prompt and layer ℓ, the main text defines the two position-centered semantic-class prototypes μ_c^ℓ and their directions $u_c^\ell = \mu_c^\ell / \|\mu_c^\ell\|_2$. Let θ^ℓ be the angle between the clean directions. The desired direction at intervention strength $\beta \in [0, 1]$ is the spherical interpolation

$$u_c^{\ell,\beta} = \frac{\sin((1-\beta)\theta^\ell)}{\sin\theta^\ell} u_c^\ell + \frac{\sin(\beta\theta^\ell)}{\sin\theta^\ell} u_{1-c}^\ell.$$

Thus $\beta = 0$ is the clean direction, $\beta = 1/2$ makes the two directions coincident, and $\beta = 1$ exchanges them.

We realize these target directions with a common orthogonal rotation for all current support-word hidden states in a class. During the intervened forward pass define the current RMS-normalized class centroid and mean positional baseline

$$r_c^\ell = \frac{1}{n_c} \sum_{i:c_i=c} x_i^\ell, \qquad \bar{b}_c^\ell = \frac{1}{n_c} \sum_{i:c_i=c} b_i^\ell.$$

If $R_c^{\ell,\beta}$ is applied to every current hidden state in class c, the resulting position-centered class mean is

$$\tilde{\mu}_c^\ell = R_c^{\ell,\beta} r_c^\ell - \bar{b}_c^\ell.$$

We require this mean to point along $u_c^{\ell,\beta}$. Equivalently, for some $\lambda_c^{\ell,\beta} > 0$,

$$R_c^{\ell,\beta} r_c^\ell = \bar{b}_c^\ell + \lambda_c^{\ell,\beta} u_c^{\ell,\beta}.$$

Orthogonality requires the target vector to have norm $\|r_c^\ell\|_2$, so $\lambda_c^{\ell,\beta}$ is chosen to satisfy

$$\|\bar{b}_c^\ell + \lambda_c^{\ell,\beta} u_c^{\ell,\beta}\|_2 = \|r_c^\ell\|_2.$$

A positive solution exists for every prompt, layer, class, and intervention strength in the experiments. We choose the solution requiring the smaller rotation, then take the minimal orthogonal rotation from r_c^ℓ to the resulting target. This rotation acts only in the two-dimensional plane spanned by the current and target centroids and as the identity on its orthogonal complement. We apply

$$\tilde{h}_i^\ell = R_c^{\ell,\beta} h_i^\ell, \qquad c_i = c,$$

independently for the two classes at every layer. Hence hidden-state norms and all within-class inner products are preserved while the prescribed class-mean directions are sustained throughout the forward pass.

C.2 PROMPT-LEVEL RESPONSE TO THE WORD-SPACE INTERVENTION

To verify that the aggregate reversal in Figure 3 is not driven by averaging across heterogeneous prompts, we analyze each prompt separately. For every prompt p, we record the target-signed output margin $m_{\text{target}}(p;\beta)$ at

$$\beta \in \{0, .1, .2, \ldots, .9, 1\},$$

and fit the affine model

$$m_{\text{target}}(p;\beta) \approx a_p + c_p\beta.$$

A negative c_p indicates the expected within-prompt response: as the class directions are moved toward exchange, the model's preference for the clean target decreases. The corresponding R^2 measures how closely this response follows a graded trajectory rather than changing abruptly.

Table 5 summarizes the prompt-level slopes and R^2 values. Across all four models, the slope quartiles are negative and median R^2 ranges from .882 to .927, showing that the aggregate causal reversal is reflected broadly at the level of individual prompts. The same prompt-wise fitting procedure is used later for the per-head response analysis in Section 5.2.

C.3 DIRECT EXCHANGE OF SELECTED-HEAD KEY PROTOTYPES

This intervention acts directly in the selected heads' support-label key spaces and is distinct from the sustained word-space rotation. Let $\bar{k}^{\text{cl}}_{h,0}$ and $\bar{k}^{\text{cl}}_{h,1}$ be the clean-reference support-label key means. At exchange strength γ, define target means

$$\bar{k}^{\text{tar}}_{h,0}(\gamma) = (1-\gamma)\bar{k}^{\text{cl}}_{h,0} + \gamma\bar{k}^{\text{cl}}_{h,1},$$
$$\bar{k}^{\text{tar}}_{h,1}(\gamma) = (1-\gamma)\bar{k}^{\text{cl}}_{h,1} + \gamma\bar{k}^{\text{cl}}_{h,0}.$$

During the forward pass, let $k^{\text{cur}}_{h,i}$ be the current key at a support-label position and $\bar{k}^{\text{cur}}_{h,c_i}$ the current mean for its displayed-label class. Replace

$$k'_{h,i} = \bar{k}^{\text{tar}}_{h,c_i}(\gamma) + \left(k^{\text{cur}}_{h,i} - \bar{k}^{\text{cur}}_{h,c_i}\right).$$

This sets the two class means to the clean-reference targets while preserving each current within-class residual. Current queries, values, and non-label keys are retained at that intervention site; the modified selected-head attention output is recomputed and patched into the forward pass. At $\gamma = 0$ the means are clean, at $\gamma = 1/2$ they meet at their midpoint, and at $\gamma = 1$ they exchange.

C.4 CLASS-MEAN KEY REPLACEMENT

In the separate clean-run simplification, every selected-head support-label key is replaced by its clean within-prompt displayed-label mean,

$$k_{h,i} \leftarrow \bar{k}^{\text{cl}}_{h,c_i}.$$

The clean query, values, and non-label keys are retained, the attention output is recomputed, and the resulting selected-head update is patched into the clean forward pass. This preserves the two class means while eliminating within-class key-score variation. It is not the label-swapped rescue protocol used in the localization experiment. Table 6 gives 95% confidence intervals for the numbers reported in the main section's Table 1.

C.5 ATTENTION-MASS IDENTITY UNDER CLASS-MEAN REPLACEMENT

Let $A_{h,c}$ be the total attention mass that selected head h assigns to support-label positions displaying label $\pi_p(c)$, and let $n_c = |\mathcal{J}_c|$. After class-mean replacement, every key in displayed-label class y has score $q_h^\top \bar{k}_{h,c}/\sqrt{d}$. Therefore

$$A_{h,c} = \frac{n_y \exp(q_h^\top \bar{k}_{h,c}/\sqrt{d})}{\sum_j \exp(q_h^\top k_{h,j}/\sqrt{d})},$$

where the denominator includes all attended positions and cancels in the class-mass ratio. Hence

$$\log\frac{A_{h,1}}{A_{h,0}} = \log\frac{n_1}{n_0} + \frac{q_h^\top(\bar{k}_{h,1} - \bar{k}_{h,0})}{\sqrt{d}} = \log\frac{n_1}{n_0} + m_h(p).$$

(a) Target accuracy.

	Clean	Class-mean label keys	Selected-head margin readout
Llama-3.1-8B	.820 [.781,.855]	.816 [.778,.850]	.830 [.793,.862]
Qwen2.5-7B	.834 [.800,.866]	.832 [.796,.863]	.838 [.804,.868]
Mistral-NeMo-12B	.844 [.808,.879]	.840 [.804,.874]	.850 [.814,.882]
Llama-3.1-70B	.850 [.812,.885]	.851 [.813,.886]	.861 [.826,.893]

(b) Prediction agreement with the clean model (%).

	Clean	Class-mean label keys	Selected-head margin readout
Llama-3.1-8B	100.0 [100.0,100.0]	93.9 [92.3,95.4]	93.4 [91.6,94.9]
Qwen2.5-7B	100.0 [100.0,100.0]	93.8 [92.5,95.0]	93.5 [92.3,94.7]
Mistral-NeMo-12B	100.0 [100.0,100.0]	96.2 [95.1,97.2]	95.2 [93.7,96.4]
Llama-3.1-70B	100.0 [100.0,100.0]	97.2 [96.2,98.1]	94.6 [92.9,96.2]

(c) R^2 with respect to the clean signed candidate-logit margin.

	Clean	Class-mean label keys	Selected-head margin readout
Llama-3.1-8B	1.000 [1.000,1.000]	.961 [.947,.969]	.919 [.902,.931]
Qwen2.5-7B	1.000 [1.000,1.000]	.943 [.927,.954]	.896 [.882,.906]
Mistral-NeMo-12B	1.000 [1.000,1.000]	.981 [.976,.984]	.955 [.947,.960]
Llama-3.1-70B	1.000 [1.000,1.000]	.989 [.986,.991]	.932 [.918,.943]

Table 6: Prototype compression and selected-head readout across models. **(a)** Target accuracy. **(b)** Prediction agreement with the clean model. **(c)** R^2 with respect to the clean signed candidate-logit margin. Brackets denote 95% confidence intervals.

For the balanced prompts $n_1 = n_0 = 10$, the count term vanishes. This identity concerns the simplified attention comparison; values, non-label positions, and downstream computation still participate in the final output.

D Quantitative Evaluation and Diagnostics

D.1 Held-out semantic-group readout

The ordinary-output readout is evaluated on unpatched prompts using the fixed global head sets in Appendix B.6. For an evaluated semantic pair (G_1, G_2), the zero-intercept coefficients in

$$\widehat{m}(p) = \sum_{h \in \mathcal{H}} \alpha_h m_h(p)$$

are fitted only on prompts whose semantic pair contains neither G_1 nor G_2. The coefficients are then fixed while evaluating all prompts from the held-out pair. This guarantees that the learned regression generalized across semantic groups.

We report three complementary metrics. *Accuracy* evaluates the target semantic classification. *Prediction agreement* evaluates whether the intervention or readout reproduces the clean model's individual label choices, including its errors. R^2 evaluates fidelity to the clean signed candidate-logit margin.

Tables 6 report the full confidence intervals for the clean model, class-mean key replacement, and the held-out selected-head margin readout.

D.2 CONTEXTUAL-GEOMETRY CLASSIFIER BASELINES

The geometric classifiers operate on the same RMS-normalized, position-centered contextual word representations used in the main geometry analysis. Each classifier is fitted independently for every prompt and layer: the training set is the twenty support words (ten per semantic class), and the unlabeled test word is the single held-out example. Thus these probes do not pool labeled examples across prompts and are distinct from the shared semantic-group-held-out readout above.

The prototype classifier uses the class means μ_c^ℓ and predicts the larger dot product with z_{test}^ℓ. The k-nearest-neighbor classifier uses Euclidean distance and inverse-distance-weighted voting, with

$$k \in \{1, 3, 5, 7, 9, 11, 15, 20\}.$$

Logistic regression and the linear SVM use the score $f_{w,b}(z) = w^\top z + b$ with L_2 regularization and

$$C \in \{10^{-2}, 10^{-1}, 1, 10, 10^2\}.$$

For labels $r_i \in \{-1, +1\}$, logistic regression minimizes

$$\mathcal{L}_{\text{LR}}(w, b) = \frac{1}{n} \sum_i \log\Big(1 + e^{-r_i f_{w,b}(z_i)}\Big) + \frac{\|w\|_2^2}{2nC},$$

with an unregularized intercept, while the squared-hinge SVM minimizes

$$\mathcal{L}_{\text{SVM}}(w, b) = \frac{1}{n} \sum_i \max(0, 1 - r_i f_{w,b}(z_i))^2 + \frac{\|w\|_2^2 + b^2}{2nC}.$$

These objectives match the implementations used in the classifier analysis. The linear models are optimized from zero for 1,000 full-batch gradient steps without momentum, using the curvature-based step sizes

$$\eta_{\text{LR}} = \left(\frac{\rho}{4n} + \frac{1}{nC}\right)^{-1}, \qquad \eta_{\text{SVM}} = \left(\frac{2\rho}{n} + \frac{1}{nC}\right)^{-1},$$

where $\rho = \lambda_{\max}([X, \mathbf{1}][X, \mathbf{1}]^\top)$ for the support design matrix X.

For the main-text layerwise curves, one hyperparameter is selected separately for each model and classifier by pooled peak query accuracy and then held fixed across layers. The resulting $(C_{\text{LR}}, C_{\text{SVM}}, k)$ are $(10^{-2}, 10, 20)$ for Llama-3.1-8B, $(10^2, 10^{-2}, 5)$ for Qwen2.5-7B, $(10^{-1}, 10^{-2}, 20)$ for Mistral-NeMo-12B, and $(10^{-2}, 10^{-2}, 20)$ for Llama-3.1-70B.

Figures 11–13 report optimization and hyperparameter diagnostics. Across all models and regularization values, the pooled training objective decreases by 66.6%–99.9% for logistic regression and by 93.6% to more than 99.99% for the squared-hinge SVM. Convergence is slowest for the larger-C SVM settings on Qwen2.5-7B and Llama-3.1-70B, but this has little effect on query accuracy. Across the tested C values, the range in peak accuracy is at most .003 for logistic regression and .011 for the SVM; at the selected layer, the spread is at most .003 and .016, respectively. For k-NN, $k = 1$ is distinctly worse, with peak accuracies of .627–.792; the best k is 20 for Llama-3.1-8B, Mistral-NeMo-12B, and Llama-3.1-70B, and 5 for Qwen2.5-7B.

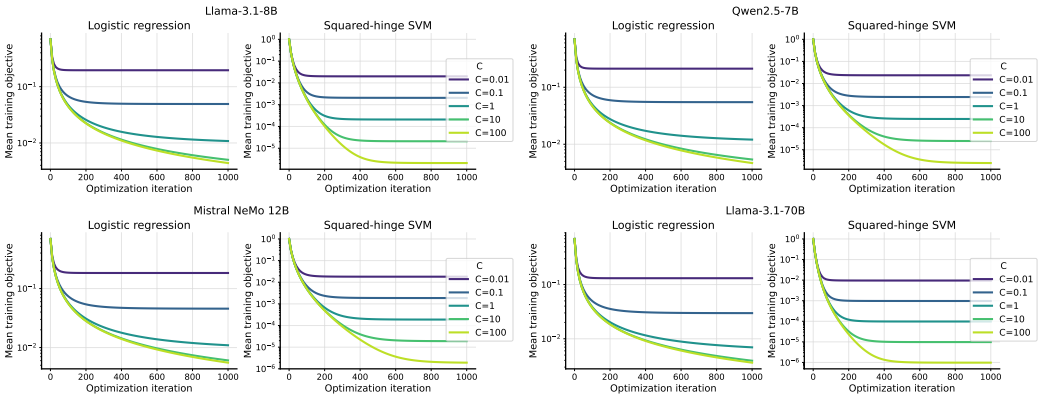


Figure 11: Optimization trajectories for logistic regression and the squared-hinge SVM across regularization values. Objectives decrease substantially within the 1,000-step optimization budget; convergence is slower for larger-C SVMs on Qwen2.5-7B and Llama-3.1-70B.

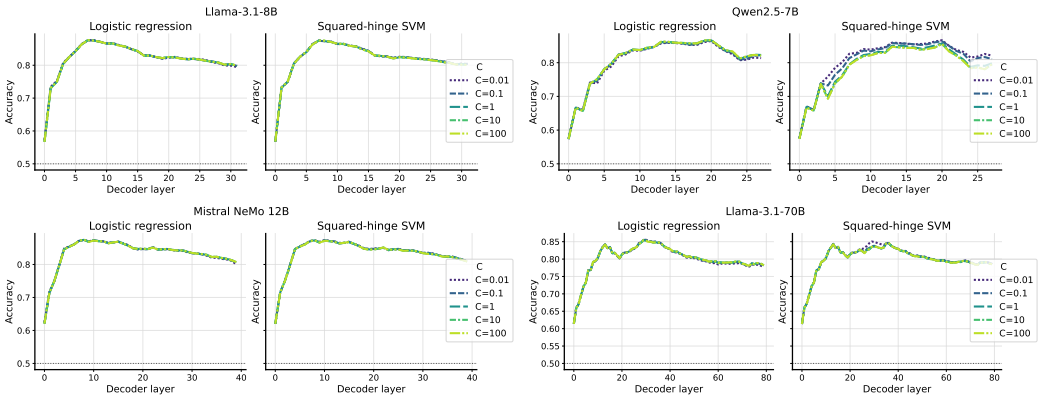


Figure 12: Sensitivity of layerwise held-out query accuracy to the linear-model regularization parameter C. The curves are similar across the tested values.

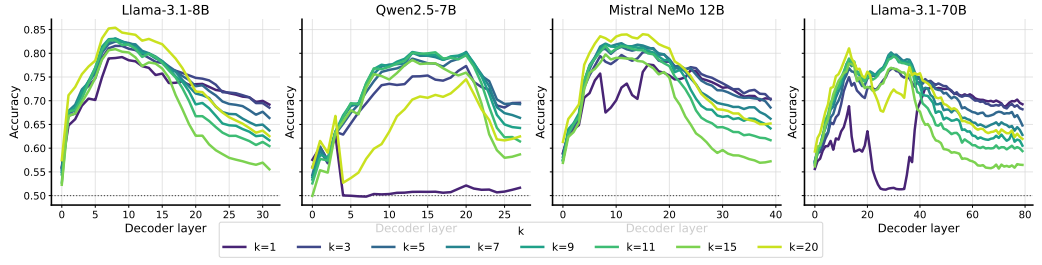


Figure 13: Sensitivity of layerwise k-nearest-neighbor accuracy to neighborhood size. Very small neighborhoods, particularly $k = 1$, perform worse than intermediate or large neighborhoods.

Table 7: Example semantic groups, optimized group pairs, and prompts for Llama-3.1-8B. (a) Five sampled semantic groups and their eligible single-token words. (b) Optimized 11-word subsets for five pairings of these groups; the final column gives the resulting deviation from the random-partition baseline. (c) One example prompt for each pair. Alternating backgrounds indicate individual tokenizer tokens. An underscore denotes a leading space that is part of the following token; for example, `_nun` is a single token.

(a) Sampled semantic groups

Semantic group	WordNet synset	N	Eligible single-token words
beginning	beginning.n.05	29	activation, attack, commencement, constitution, creation, debut, entry, establishment, formation, foundation, founding, groundbreaking, inauguration, initiation, initiative, innovation, installation, installing, installment, institution, introduction, kickoff, launching, opening, organisation, organization, start, startup, unveiling
letter	letter.n.02	25	alpha, beta, chi, delta, epsilon, eta, gamma, initial, iota, kappa, khi, lambda, mem, nun, omega, phi, psi, resh, rho, shin, sigma, sin, tau, theta, vowel
motion	motion.n.06	42	advance, advancement, approach, approaching, ascending, ascent, chase, coast, coming, crawl, descent, displacement, flow, following, glide, haste, hurry, maneuver, migration, move, movement, play, procession, progress, progression, pursuit, return, rise, rush, rushing, shift, shifting, slide, speed, speeding, stream, swing, swinging, translation, travel, traveling, travelling
occupation	occupation.n.01	28	accounting, appointment, berth, biz, business, calling, career, catering, craft, employment, farming, game, job, land, line, medium, office, photography, place, position, post, profession, situation, sport, spot, trade, treadmill, work
tract	tract.n.01	30	battlefield, battleground, center, clearing, common, commons, desert, diamond, field, green, grounds, infield, left, lot, midway, oasis, outfield, parcel, park, patch, plot, range, right, sector, short, site, square, subdivision, terrain, yard

(b) Optimized 11-word subsets

Group A	Optimized subset	Group B	Optimized subset	Deviation (%)
beginning	activation, attack, commencement, constitution, debut, entry, groundbreaking, installing, installment, start, unveiling	letter	alpha, delta, initial, iota, khi, mem, nun, resh, shin, sin, vowel	19.33
beginning	activation, attack, commencement, debut, entry, founding, groundbreaking, innovation, installment, institution, kickoff	tract	battleground, center, clearing, common, infield, oasis, parcel, range, sector, site, subdivision	0.00
letter	delta, gamma, initial, iota, khi, mem, nun, resh, shin, sin, vowel	tract	battleground, common, diamond, green, infield, left, midway, oasis, plot, range, subdivision	4.92
motion	advancement, approach, chase, coast, play, procession, return, shift, slide, speeding, travelling	occupation	appointment, berth, calling, employment, line, place, post, situation, sport, trade, treadmill	0.00
occupation	berth, catering, farming, land, line, medium, office, profession, situation, spot, treadmill	tract	battleground, clearing, common, left, midway, parcel, plot, sector, short, square, terrain	0.00

(c) Example tokenized prompts

Group pair	Query	Target	Prompt
beginning / tract	institution	a	`<|begin_of_text|>` `_kickoff` `_->` `_a` `\n` `_battleground` `_->` `_b` `\n` `_site` `_->` `_b` `\n` `_debut` `_->` `_a` `\n` `_groundbreaking` `_->` `_a` `\n` `_oasis` `_->` `_b` `\n` `_parcel` `_->` `_b` `\n` `_activation` `_->` `_a` `\n` `_range` `_->` `_b` `\n` `_subdivision` `_->` `_b` `\n` `_infield` `_->` `_b` `\n` `_commencement` `_->` `_a` `\n` `_installment` `_->` `_a` `\n` `_center` `_->` `_b` `\n` `_attack` `_->` `_a` `\n` `_clearing` `_->` `_b` `\n` `_common` `_->` `_b` `\n` `_innovation` `_->` `_a` `\n` `_entry` `_->` `_a` `\n` `_founding` `_->` `_a` `\n` `_institution` `_->`
motion / occupation	slide	b	`<|begin_of_text|>` `_return` `_->` `_b` `\n` `_calling` `_->` `_a` `\n` `_employment` `_->` `_a` `\n` `_advancement` `_->` `_b` `\n` `_sport` `_->` `_a` `\n` `_trade` `_->` `_a` `\n` `_place` `_->` `_a` `\n` `_appointment` `_->` `_a` `\n` `_coast` `_->` `_b` `\n` `_travelling` `_->` `_b` `\n` `_berth` `_->` `_a` `\n` `_treadmill` `_->` `_a` `\n` `_shift` `_->` `_b` `\n` `_approach` `_->` `_b` `\n` `_procession` `_->` `_b` `\n` `_situation` `_->` `_a` `\n` `_chase` `_->` `_b` `\n` `_speeding` `_->` `_b` `\n` `_play` `_->` `_b` `\n` `_line` `_->` `_a` `\n` `_slide` `_->`

Table 8: Canonical selected-head circuits used in our analyses. Heads are ordered by the causal selection score computed across the full prompt set. We retain the top 16 heads for Llama-3.1-8B, Qwen2.5-7B, and Mistral NeMo 12B, and the top 32 heads for Llama-3.1-70B.

(a) Llama-3.1-8B

Rank	Layer	Head	Score
1	17	24	.3727
2	16	1	.3621
3	24	27	.3477
4	18	28	.2103
5	18	20	.1998
6	30	24	.1602
7	16	20	.1365
8	31	1	.1260
9	15	30	.1163
10	26	3	.1156
11	18	16	.0957
12	17	25	.0759
13	20	13	.0609
14	30	11	.0606
15	18	23	.0562
16	16	3	.0530

(b) Qwen2.5-7B

Rank	Layer	Head	Score
1	23	11	.4190
2	22	4	.2446
3	22	3	.2252
4	24	27	.1880
5	22	7	.1279
6	22	1	.1153
7	22	13	.0783
8	22	2	.0738
9	24	21	.0737
10	19	15	.0715
11	26	25	.0665
12	19	22	.0556
13	26	14	.0549
14	19	19	.0520
15	24	26	.0352
16	23	13	.0337

(c) Mistral NeMo 12B

Rank	Layer	Head	Score
1	19	28	.3096
2	31	13	.2816
3	26	18	.2482
4	29	4	.2157
5	20	9	.1935
6	19	31	.1916
7	20	10	.1839
8	37	28	.1827
9	38	9	.1738
10	31	21	.1641
11	18	16	.1120
12	28	30	.1101
13	39	7	.1024
14	33	20	.0922
15	34	30	.0879
16	20	15	.0869

(d) Llama-3.1-70B

Rank	Layer	Head	Score
1	39	40	.2710
2	74	9	.2075
3	53	1	.2038
4	76	57	.1413
5	77	50	.1326
6	73	62	.1320
7	77	42	.1251
8	35	43	.1080
9	54	15	.1031
10	56	42	.1010
11	52	19	.0965
12	39	43	.0945
13	77	47	.0879
14	35	18	.0658
15	35	17	.0649
16	54	9	.0633
17	64	27	.0627
18	38	49	.0599
19	47	17	.0543
20	77	44	.0532
21	39	42	.0476
22	35	19	.0475
23	70	5	.0445
24	42	31	.0423
25	35	46	.0408
26	79	32	.0403
27	52	48	.0386
28	55	20	.0348
29	78	2	.0330
30	35	23	.0310
31	38	55	.0308
32	76	10	.0305