Investigating Anthropomorphism in Large Language Models through the Lens of Sparse Autoencoders

Anonymous ACL submission

Abstract

In this work, we identify several intriguing internal mechanisms shared across diverse Large Language Models (LLMs) during prompt processing. These behaviors are not explicitly trained, yet they arise reliably across model families and scales and exert influence on model behavior. By adopting a cognitive-inspired perspective, we demonstrate that these patterns resemble established heuristics in human information processing, such as implicit structural segmentation, forming unconscious expectations, and the dynamic adaptation of internal resources under constraint.

Using sparse autoencoders (SAEs) and decoder logit lens as analytical tools, we uncover multiple such phenomena, including (1) internal semantic parsing features that track document structure; (2) cross-exemplar interactions, where current representations are modulated by expectations induced by prior context; (3) role-adaptive features that exhibit functional plasticity by dynamically shifting their semantic profile based on contextual constraints; and (4) implicit expectations regarding the number of few-shot exemplars. We statistically validate these behaviors across multiple model architectures, suggesting that LLMs develop internal heuristics that, while not explicitly human, exhibit striking structural similarities to patterns observed in human cognition.

1 Introduction

> *"Language models just being programmed to try to predict the next word is true, but it's not the dunk some people think it is. Animals, including us, are just programmed to try to survive and reproduce, and yet amazingly complex and beautiful stuff comes from it."* — Sam Altman

Modern large language models (LLMs) are trained with a relatively simple objective: autoregressive next-token prediction (Brown et al., 2020). Despite the apparent simplicity of this training signal, a growing body of work has shown that such models exhibit a wide range of complex behaviors. In particular, prior work has shown that large language models exhibit *emergent abilities*, namely capabilities that are absent in smaller models but appear as model scale increases (Wei et al., 2022).

Motivated by this perspective, we ask how rich these capabilities (i.e., not explicitly trained, yet clearly emergent) are, how they influence model behavior, and whether they pose potential risks in real-world applications. Addressing these questions is not only intrinsically intriguing, but also important because they may reveal meaningful caveats and practical considerations for the deployment and use of LLMs. In this work, we build on recent advances in mechanistic interpretability, specifically, powerful analysis tools such as sparse autoencoders (SAEs) and decoder logit lens (Belrose et al., 2023). Standing on the shoulders of these giants, we investigate a collection of universal internal behaviors in LLMs that consistently arise across different model families and scales.

Our analysis identifies four distinct, consistent internal processing behaviors that emerge as LLMs interpret and process prompts, several of which closely resemble patterns observed in human cognition: (1) models maintain an internal state tracking mechanism that segments the input prompt into semantically coherent units, reminiscent of the hierarchical parsing humans employ when reading and structuring documents; (2) cross exemplar interactions, in which the representation of the current exemplar is modulated by expectations induced by prior context, resembling "predictive coding" like behaviors in the human brain; (3) adaptive functional modulation of internal features in response to input context, analogous to the "functional plasticity" observed in human cognition; and (4) an implicit expectation over the number of few shot

exemplars, reminiscent of how humans may unconsciously form expectations about the quantity of examples to be presented.

It is important to emphasize that our goal is not to anthropomorphize LLMs, but rather to draw on well established findings from human cognition as a principled and rich source of inspiration for experimental design. By bridging insights from biological cognition with computational analysis, we design experiments that expose intriguing internal patterns that have not been previously documented. Such patterns would be very difficult to uncover using purely bottom up approaches without guidance from human behavioral insights. Our results indicate that treating human cognition as a conceptual mirror provides a powerful and systematic methodology for revealing hidden structure in artificial intelligence systems.

2 Related Work

2.1 Studies on the Emergent Capabilities in LLMs

Emergent capabilities are behaviors or abilities that are not explicitly trained for but arise as language models scale to large sizes (Berti et al., 2025). Prior work has shown a wide range of such emergent capabilities in LLMs, including in-context learning (Wei et al., 2022), collaborative behaviors (Chen et al., 2024b), and arithmetic ability (Chen et al., 2024a). However, while these studies characterize what capabilities emerge at the level of task performance, the question of *which internal mechanisms or processing heuristics emerge inside the model* remains underexplored. In this work, we shift the focus from emergent *outcomes* to emergent *mechanisms*, investigating how large language models internally parse, organize, and integrate prompt information in ways that systematically shape their downstream behavior.

2.2 Anthropomorphisms in LLMs

Anthropomorphism refers to the tendency to attribute human-like understanding or intentions to non-human systems, a phenomenon commonly illustrated by the ELIZA effect (Weizenbaum, 1966). Rather than merely anthropomorphizing AI, we use it as a methodological lens for generating hypotheses about internal processing mechanisms. For example, insights from human cognition can offer useful clues about potential computational strategies for processing structured inputs, such as **Predictive Coding** (Rao and Ballard, 1999).

Predictive coding is a prominent theory in cognitive neuroscience that suggests that cognition is shaped by how strongly the brain responds to information that aligns with or violates prior expectations. Information that matches expectations tends to evoke weaker responses, while unexpected input produces stronger signals that prompt updates to internal representations (Friston and Kiebel, 2009; Friston, 2010; Bastos et al., 2012). Despite its prominence in neuroscience, whether predictive coding–like behavior arises in large language models remains an open question.

In biological systems, progress in understanding cognition has often come from identifying the functional roles of specific neurons or neuronal populations. A canonical example is the discovery of place cells in the rodent hippocampus, which revealed how spatial information is internally represented during navigation (O'Keefe and Dostrovsky, 1971). In contrast, despite the fact that all internal activations in artificial neural networks are fully accessible, our mechanistic understanding of large language models remains limited.

Prior work has identified neurons or features that respond selectively to particular concepts, such as the Golden Gate Bridge or multilingualism (Templeton et al., 2024), and has led to resources like Neuronpedia (Lin, 2023) that provide annotations describing common patterns in next-token predictions generated using LLMs as judges. However, because such annotations are limited to commonalities in next-token prediction, they offer little insight into why a neuron becomes active, and our work addresses this limitation by drawing inspiration from human subjective experience to generate mechanistic hypotheses about neuron activation during task execution.

2.3 Sparse Autoencoders

One of the most effective approaches for making the neural activations of large language models (LLMs) interpretable to humans is the Sparse Autoencoder (SAE) (Sharkey and Beren, 2022). A major obstacle to understanding LLMs is that their internal representations are expressed as vectors that are not directly human-interpretable. Features disentangled by SAEs, however, are often regarded as analogous to a *microscope* that allows us to probe the internal mechanisms of large language models (Team, 2024; Lindsey et al., 2025).

Specifically, while individual neurons in LLMs

tend to activate densely and participate in many unrelated computations, SAEs address this issue by learning representations in which only a small number of units activate for any given input. Thus, the intermediate neurons of an SAE, which activate sparsely, are referred to as *features*. Formally, let $\mathbf{x}_i^{(\ell)} \in \mathbb{R}^d$ denote the activation vector at layer ℓ of an LLM for the i-th data sample. A sparse autoencoder reconstructs the same activation:

$$\hat{\mathbf{x}}_i^{(\ell)} = W_{\text{dec}}\, \sigma\left(W_{\text{enc}}\, \mathbf{x}_i^{(\ell)}\right), \tag{1}$$

where W_{enc} and W_{dec} denote the encoder and decoder weight matrices, respectively. $\sigma(\cdot)$ is a non-linear function, and in this study, we use JumpReLU (Lieberum et al., 2024), as implemented in GemmaScope (Lieberum et al., 2024) and LlamaScope (He et al., 2024). In the loss function, the sparsity regularizer L_{sparsity} encourages only a small number of features to be active for each input:

$$L_{\text{SAE}} = \frac{1}{N}\sum_{i=1}^{N} \left\|\mathbf{x}_i^{(\ell)} - \hat{\mathbf{x}}_i^{(\ell)}\right\|_2^2 + \lambda\, L_{\text{sparsity}}. \tag{2}$$

3 Emergent Cognitive-Like Processing Heuristics in Large Language Models

Overview. In this section, we characterize four emergent internal mechanisms shared across diverse model families that resemble fundamental human cognitive behaviors. Specifically, we investigate: (1) structural parsing heuristics, (2) predictive coding patterns, (3) functional plasticity behaviors, and (4) implicit expectations regarding the number of few-shot exemplars provided in a prompt.

Throughout the study, we conduct our experiments primarily using three widely-adopted LLMs: `Gemma-2-2B-IT`, `Gemma-2-9B-IT` (Lieberum et al., 2024), and `Llama-3.1-8B-Instruct` (Dubey et al., 2024). These models were selected both for their strong performance and the availability of high-fidelity, pre-trained SAEs provided by the community. Throughout our analysis, we focus on SAEs trained on the middle layers of each model—unless otherwise specified. For our evaluation, we utilize the AGNews dataset (Zhang et al., 2015) as our primary experimental bed due to its widespread use and the generality of its classification tasks. Additionally, for specific analyses requiring more rigorous validation, we incorporate a specialized sub-task derived from the Berkeley Function Calling Leaderboard (BFCL) (Patil et al., 2025) (more details in Appendix A). This allows us to measure the generalizability of our findings.

3.1 Human-Like Parsing Behaviors in LLMs

Human readers do not process text as a flat sequence of tokens. Instead, they implicitly segment content into higher level semantic units, such as discourse segments or sections, to support efficient comprehension and reasoning (Grosz and Sidner, 1986; Kintsch and Van Dijk, 1978). A central mechanism underlying this behavior is "chunking" (Rosenbaum et al., 1983), in which complex inputs are decomposed into repetitive and predictable structures.

We hypothesize that if LLMs employ a structural strategy analogous to human chunking, their internal representations should exhibit periodic fluctuations when processing recurring patterns. Driven by this hypothesis, we systematically searched for SAE features exhibiting cyclic activation patterns that align with recurring structural motifs—specifically the boundaries between few-shot exemplars.

Experimental Setup For each SAE feature, we analyze its activation values over the token sequence of a prompt. Let $\mathbf{a} = (a_1, a_2, \ldots, a_T)$ denote the activation of a feature across a prompt of length T. Since few-shot exemplars vary in length, we do not assume a fixed period. Instead, we treat high-activation events as recurrent markers and examine their relative positions within each exemplar.

We first identify activation peaks $\mathcal{P} = \{t \mid a_t > \tau\}$ using a 95th-percentile threshold. For each peak t falling within an exemplar interval $[s_i, e_i]$, we compute the relative phase $r_t = (t - s_i)/(e_i - s_i)$. To quantify cyclic structure, we test the null hypothesis that these phases are uniformly distributed, $H_0 : r_t \sim \text{Uniform}(0, 1)$, using a Kolmogorov–Smirnov test. The Cyclic Parsing (CP) score for a feature f is defined as $\text{CP}(f) = 1 - p_{\text{KS}}$, where p_{KS} denotes the corresponding p-value. High scores indicate strong alignment between feature activations and exemplar boundaries. Final scores are averaged across 100 prompts to ensure stability.

Experimental Results Applying the aforementioned identification criteria, we successfully isolate a robust population of features exhibiting rhythmic activation, which we term Cyclic Parsing (CP) features. Through our automated scoring pipeline,

we identified approximately 10 candidate features per model achieving a CP score greater than 0.9.

However, a high CP score alone can be numerically inflated by "pseudo-cyclic" features—such as those that fire exclusively on terminal punctuation or redundant structural delimiters. To ensure functional relevance, we conducted a rigorous manual inspection of these candidates' activation trajectories across diverse prompt types. This qualitative validation allowed us to identify a **single, high-fidelity CP feature** that demonstrates consistent, cross-model behavior across all three architectures tested (`Gemma-2-2B`, `Gemma-2-9B`, and `Llama-3.1-8B`). This "universal" feature serves as our primary unit of analysis for investigating the model's structural parsing of tool-calling sequences.

The candidate CP feature is shown in Figure 1. As illustrated by the representative feature `Gemma2-9B-L9-4919`, CP features exhibit a recurring activation pattern: activation peaks when the model internally identifies the beginning of a new semantic block and then gradually decreases as processing proceeds toward the end of that block. Note that these boundaries are not supplied by any explicit supervision or annotation (although the model does appear to partially rely on the newline token "`\n`", which is natural); instead, they arise from the model's own internal dynamics. This consistent cyclic behavior (across different task types and examples) suggests that the LLM implicitly forms and maintains an internal notion of semantic blocks during processing, reflecting how it parses and organizes the structure of the input. In other words, the activation magnitude of a CP-feature functions as a predictive signal: a high-magnitude negative activation signals the imminent conclusion of the current semantic block. This suggests the model internally anticipates structural transitions, allowing it to prepare for the subsequent segment in the input stream.

For instance, in the AGNews prompt (bottom panel of Figure 1), which consists of (1) a task instruction followed by (2) a set of few-shot exemplars, we observe that the model identifies and treats each exemplar as a distinct semantic block, in a manner that closely aligns with how a human reader would naturally segment the prompt.

Similarly, in tool-calling prompts, which contain (1) a task instruction followed by (2) a set of function-description and argument-description pairs, (3) a user query, and (4) the model's generated output, we observe in the top panel of Figure 1 that the model acknowledges multiple semantic blocks. In particular, the model treats the task instruction as comprising three distinct blocks, regards each function–argument description pair as a single combined semantic block, identifies the user query as another block, and then recognizes the generation phase as a new semantic block. This parsing behavior closely mirrors how humans are likely to read and structure such prompts, demonstrating a high degree of alignment between the model's internal parsing dynamics and human reading behavior.

Furthermore, we observe the existence of similar features in other models (see Figure 5). This high degree of structural conservation across diverse model families suggests that cyclic parsing is a convergent mechanistic solution for managing and navigating structured input prompts.

Notably, these specialized, universal features might have remained overlooked had we not drawn inspiration from human cognitive behaviors. This serves as a compelling case study for **human-centric interpretability:** by using human behavioral patterns as a heuristic, we can uncover 'hidden' latent mechanisms in LLMs that purely automated discovery might miss.

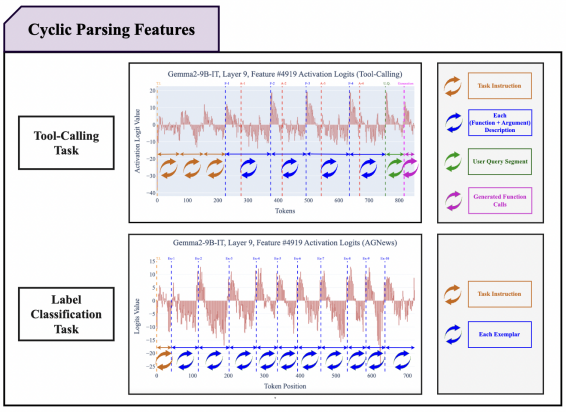


Figure 1: **Cyclic Parsing (CP) Feature Example (`Gemma2-9B-L9-4919`).** As shown, CP features exhibit a recurring activation pattern: their activation peaks at what the model internally identifies as the onset of a new semantic block and gradually decreases as processing advances toward the end of that block. Furthermore, this parsing pattern is consistent with how human readers would naturally segment the input. This consistent cyclic behavior across tasks and examples suggests that the LLM autonomously forms and tracks an internal notion of semantic blocks during processing, reflecting how the model appears to parse and organize the structure of the input.

3.2 Predictive Coding-Like Behaviors in LLMs

We identify an internal behavior, which we refer to as **Predictive Coding-Like Behavior (PCB)**. When an LLM processes a sequence of exemplars, this behavior manifests as internal feature activations that vary systematically with the immediately preceding context. Specifically, we observe a reciprocal activation pattern across a substantial population of SAE features: if a feature is strongly activated by the previous exemplar, its activation is typically suppressed during the processing of the current exemplar, and vice versa. This feature-level inversion reflects a dynamic cross-exemplar interaction, where the representation of the current input is modulated by expectations induced by the prior context.

From an anthropomorphic perspective, this mechanism closely resembles the principle of "Predictive Coding" in neuroscience. Predictive Coding posits that the brain is not a passive recorder of sensory input but an active inference engine. To optimize representational efficiency, the brain generates "top-down" predictions about incoming stimuli; if the input matches the prediction (redundancy), the neural response is minimized. Only the "prediction error"—the mismatch between expectation and reality—is propagated forward as a high-magnitude signal.

We propose that LLMs may have internalized a computationally analogous heuristic. We demonstrate that a high feature activation in the previous exemplar establishes a strong "top-down" expectation. When the subsequent exemplar contains the same feature, the model recognizes it as redundant and inhibits the activation, effectively filtering out "static" to focus on the "novel" components of the input. Conversely, the appearance of a feature that was absent in the prior context triggers a significant prediction error, resulting in the sensitization and high-magnitude activations characteristic of these features (See Figure 2a). If a significant number of SAE features exhibit such behavior, it would provide compelling evidence that the model's latent space may be optimized for information gain, mirroring the efficient coding strategies observed in various signal-processing systems.

Experimental Setup To empirically evaluate the PCB hypothesis, we curate a set of $M = 50$ target exemplars and $N = 50$ distinct previous exemplars, generating a total of $2,500$ unique prompts of the form $(\text{Exemplar}_{prev}, \text{Exemplar}_{tgt})$ sampled from the AGNews dataset. The design holds the target exemplar fixed while systematically varying the preceding exemplar across all N possibilities, repeated for each of the M target exemplars. This setup allows us to isolate the specific influence of prior context on current feature activations.

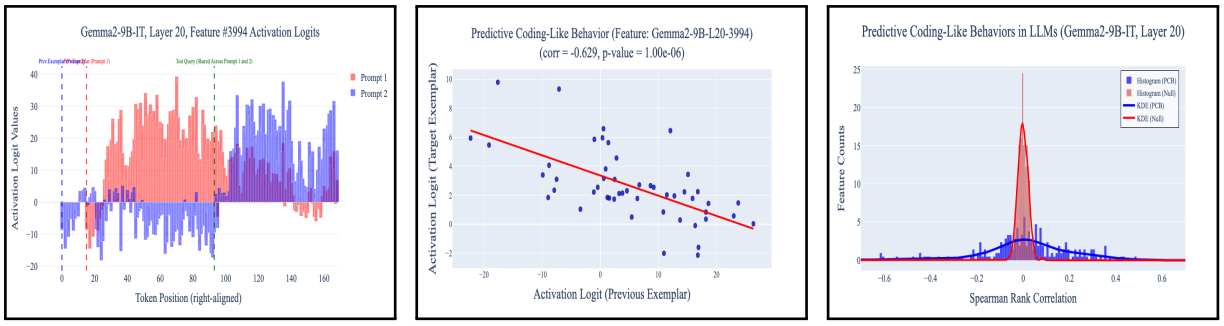
Intuitively, if a feature were strictly context-independent, its activation a_{tgt} would remain invariant across all N exemplars. However, the PCB hypothesis suggests that for a specific class of features, a_{tgt} fluctuates as an inverse function of the activation a_{prev}. To quantify this, we first filter out features that activate too infrequently, using an average activation frequency threshold of 5%. For the remaining features (usually a few hundreds), we compute the Spearman rank correlation coefficient (ρ_i) for each SAE feature i across the 2,500 prompts:

$$\rho_i = \frac{\text{cov}(R(\mathbf{A}i, \text{prev}), R(\mathbf{A}i, \text{tgt}))}{\sigma_{R(\mathbf{A}i,\text{prev})}\sigma_{R(\mathbf{A}i,\text{tgt})}} \quad (3)$$

where $R(\cdot)$ denotes the rank, and $\mathbf{A}_{i,\text{prev}}$ and $\mathbf{A}_{i,\text{tgt}}$ are vectors representing the previous and target exemplar activations. We aggregate activations into a single scalar by averaging SAE feature activations across all tokens in the exemplar. A strong negative ρ_i serves as a statistical signature of the hypothesized anti-relationship: *strong prior activation is associated with subsequent suppression, while prior absence is associated with relative sensitization.*

Prevalence of Features Exhibiting Negative Correlations The resulting distribution of Spearman rank correlations across features is presented in Figure 2c. To verify that these negative correlations represent a genuine property of the model rather than stochastic noise, we compare the empirical distribution against a null distribution generated via permutation. This null distribution is created by randomly shuffling the association between a_{prev} and a_{tgt}, as done in standard permutation tests, thereby destroying the temporal context while preserving marginal activation statistics.

The result is striking in that the observed distribution is remarkably symmetric, centered near zero but with significantly heavy tails in both directions compared to null. This observation challenges the natural expectation that the distribution would be heavily right-shifted. Under a traditional understanding of transformer architectures—which are



(a) PCB Example (Activation Flip) (b) PCB Example (Strong Negative Corr.) (c) Distribution of Spearman Corr.

Figure 2: **Illustration of the Predictive Coding–Like Behavior (PCB) in LLMs. (a)** shows the activation of a PCB feature, `Gemma2-9B-L20-3994`, across tokens in a representative prompt. Prompts 1 and 2 share the same target exemplar (onset marked in green). In Prompt 1 (red), the feature activates much more strongly in the exemplar immediately preceding the target exemplar, whereas in Prompt 2 (blue) the same feature exhibits a strong negative activation. Despite the identical target exemplar, a complete reversal in activation occurs. **(b)** shows the activations of the same feature across paired (previous, target) exemplars, showing strong negative correlation. **(c)** shows the overall distribution of features' correlation values across all active features.

designed to maintain and aggregate context through residual connections—one would expect "carry-on semantics" to be the dominant motif.

However, the presence of a substantial population of features in the negative tail ($\rho < -0.3$) suggests that inhibitory dynamics are as fundamental to the model's internal logic as "carry-on" ones (results on different models provided in Appendix C). A Kolmogorov-Smirnov test confirms that the observed distribution is significantly distinct from the null ($p < 0.01$), providing evidence that this PCB mechanism is a structural, non-random feature of the SAE representation space. Notably, this PCB behavior is even more pronounced in `Llama-3.1-8B-Instruct` and `Gemma2-2B-IT`. This suggests that PCB is likely a fundamental mechanism shared across architecturally diverse models and varying scales.

Practical Implications of PCB for Model Steering Beyond characterizing this tendency in LLMs, we believe the Predictive Coding-like Behavior (PCB) has significant implications for their practical application. Specifically, PCB can be viewed as a form of inter-exemplar interference, where the internal representation of a preceding exemplar modulates the processing of the current one, potentially introducing unintended bias.

For example, standard activation steering approaches often apply feature amplification or suppression uniformly across an entire prompt. However, our findings suggest that such strategies may yield counterintuitive results: amplifying a specific feature in a preceding exemplar may inadvertently trigger a compensatory suppression of that same feature when the model processes the target query. This interaction could effectively diminish or nullify the intended steering effect. Consequently, we suggest that accounting for PCB dynamics is essential for designing more robust steering interventions. We believe developing steering strategies that accommodate these internal cross-exemplar interactions represents an interesting direction for future research.

3.3 Functional Plasticity-Like Behaviors in LLMs

Functional plasticity is a well-established concept in neuroscience, referring to the brain's ability to dynamically reorganize or reassign functional roles in response to context, experience, or task demands (Merzenich et al., 1984; Sadato et al., 1996). Empirical evidence suggests that the same neural substrates can support diverse cognitive functions depending on immediate input or environmental constraints, enabling robust adaptation within a fixed underlying architecture (Sadato et al., 1996).

Drawing inspiration from this biological adaptability, we designed experiments to investigate whether LLMs exhibit analogous capabilities—specifically, whether their internal representations modulate their functional roles during inference. That is, rather than treating internal features as static semantic detectors, we examine whether these roles shift dynamically in response to contextual changes. Surprisingly, we identified a substantial number of features that demonstrate such plas-

ticity, actively adopting new functional roles when the input distribution is significantly constrained. This finding suggests that SAE features in LLMs are not rigid semantic labels, but dynamic resources that the model re-deploys to maximize utility in inference-time.

Experimental Setup To evaluate whether SAE features exhibit behaviors analogous to functional plasticity, we designed a controlled "label-exclusion" intervention. The experiment proceeds in two distinct phases: identification and perturbation.

1. Identifying Label-Specific Features We define label j-specific features through a two-step process: (1) To ensure statistical robustness, we exclude extremely low-frequency features, defined as those with an average firing frequency of less than 5% across all tokens in the dataset. (2) Among the remaining features, we define feature i as being label-specific to label j if its average activation frequency on label j is the highest among all candidate labels. We intentionally adopt this generous definition of specificity to capture a broad and diverse representative population of features for subsequent analysis.

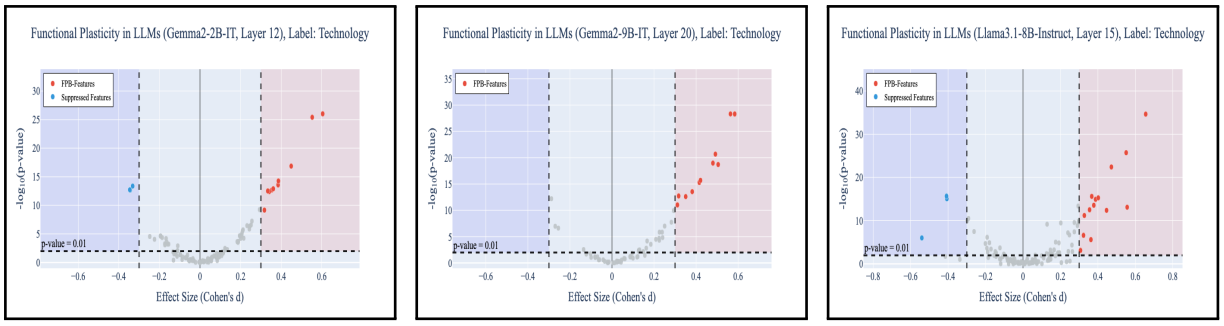
2. Label-Exclusion Perturbation For each label j, we construct evaluation prompts where all exemplars originally belonging to label j are systematically replaced by exemplars of other labels randomly sampled from the context. This allows us to isolate the "displaced" label j-specific features and observe their activation dynamics within a novel environment where their target label is explicitly absent. This setup allows us to test two plausible hypotheses regarding the nature of SAE representations: **(1) The Static Representation Hypothesis:** If features act as rigid semantic detectors, a label j-specific feature should go dormant when its primary trigger (label j) is absent. **(2) The Functional Plasticity Hypothesis:** Drawing inspiration from biological neural systems, we investigate whether these features are recruited to fire more on other labels when their primary target is unavailable.

Prevalence of Functional Plasticity Behavior (FPB) Features To quantify behavioral shifts within the label-specific features, we measure the change in firing frequency for each label j-specific feature i when its primary target (label j) is excluded from the context (i.e., activity on the remaining label exemplars). We visualize these results using volcano plots, mapping the effect size—calculated via Cohen's d—on the x-axis against the statistical significance—represented as $-\log_{10}(\text{p-value})$—on the y-axis. The p-values are derived from a two-sided Wilcoxon test, providing a non-parametric assessment of whether a feature's change in activation is statistically significant.

The results for the `Gemma2-9B-IT` model are presented in Figure 3, using "Technology" as the representative target label. We observe a pronounced right and up-shift in the distribution of feature activations; this signifies a substantial population of features that exhibit functional plasticity, increasing their firing frequency on alternative labels when their primary semantic trigger is removed. Comprehensive results for additional labels and model architectures are provided in Appendix D, which reveal highly analogous patterns across all tested conditions. These findings suggest that functional plasticity is likely a universal property across various model families and parameter scales.

Intuitive and Directed Functional Migration A natural follow-up question is whether this recruitment occurs randomly across the remaining labels or follows a structured logic. We find that the process operates in a highly intuitive manner: features primarily migrate toward labels that are semantically or contextually familiar to them in the baseline setting. To quantify this, we rank the alternative labels for each feature i based on their relative activation frequency in the baseline distribution. Following the label-exclusion intervention, we measure the change in frequency across all label-specific features that exhibited significant plasticity (defined as Cohen's $d \geq 0.3$). As shown in Figure 10, the magnitude of recruitment is strongly correlated with the baseline hierarchy; features are far more likely to be "repurposed" for labels they already partially recognized than for entirely unfriendly categories.

Implications of FPB for Mechanistic Interpretability The discovery of FPB-features reveals the complex, multifaceted nature of SAE representations and challenges the conventional view of features as static, binary semantic detectors. Under the prevailing paradigm, a feature is expected to activate only in the presence of a fixed, pre-defined concept within a token or sequence. While this interpretation remains valid for a subset of the feature population, our findings demonstrate



(a) `Gemma2-2B-IT` (b) `Gemma2-9B-IT` (c) `Llama3.1-8B-Instruct`

Figure 3: **Illustration of the Functional Plasticity-like Behavior (FPB) in LLMs.** Each subfigure shows ...Volcano plots of feature activation shifts exhibit a pronounced positive skew, identifying a significant population of "recruited" features. ... Aggregate recruitment magnitude as a function of baseline frequency; the results demonstrate that functional migration is not stochastic, but is systematically directed toward semantically familiar categories.

it is incomplete. Specifically, FPB-features function as context-aware functional units that dynamically modulate their semantic profiles based on the global constraints of the prompt. This suggests that "interpreting" a feature solely based on its top-activating examples may overlook its broader role as a flexible resource capable of representational migration.

3.4 Implicit Expectations over the Number of Few-Shot Exemplars

When humans are presented with a sequence of few-shot exemplars without an explicitly stated total count, they often form an implicit expectation about how many exemplars the sequence will contain. Such expectations may be shaped by task structure (e.g., anticipating a multiple of the number of labels) or by more general cognitive priors, such as the prevalence of base-ten counting conventions.

We investigate whether large language models develop analogous expectations during prompt processing. Specifically, we analyze whether models exhibit systematic preferences over the number of few-shot exemplars, even when the prompt provides no explicit signal indicating a stopping point.

Figure 12 presents a Decoder Logit Lens analysis of `Llama-3.1-8B-Instruct` and `Gemma2-9B-IT`. At each exemplar boundary (after Example k, before Example $k{+}1$), we probe the final-layer hidden state and measure the model's predicted probability of either terminating the prompt via the end-of-sequence (EOS) token or continuing with another exemplar header (`Example`). Despite the absence of any task-level incentive favoring particular exemplar counts, the model exhibits pronounced probability peaks at specific boundaries. In particular, peaks align first with multiples of ten exemplars and subsequently with multiples of five, indicating a structured bias in the model's internal expectations over exemplar number.

This behavior suggests that the model maintains a latent prior over plausible prompt lengths, which manifests as elevated confidence at certain exemplar counts. One plausible explanation is that such priors arise from statistical regularities in the pretraining data. More concretely, this bias may reflect a combination of **(1) Pretraining Artifacts,** in which few-shot prompts in the training corpus often terminate after five or ten exemplars, and **(2) Systemic Bias,** arising from the prevalence of base-ten conventions in natural language and instructional text. Results on additional models in Appendix E show that this pattern appears in four of the five evaluated models.

4 Conclusion

We showed that large language models develop systematic internal patterns during prompt interpretation that are not explicitly trained for, yet consistently emerge across model families and scales. Using sparse autoencoders, we made these patterns observable and linked them to concrete behaviors such as implicit expectations, context-sensitive modulation, and structured prompt parsing. Our results suggest that some seemingly anthropomorphic aspects of LLM behavior can be traced to measurable internal dynamics, highlighting the value of mechanistic analysis for understanding how models interpret and respond to prompts.

5 Limitations

Our analyses are designed to identify internal signals and representational patterns that correlate with long-range expectations in LLMs during prompt processing. While these signals are predictive of specific continuation behaviors, our study does not establish how these patterns are implemented at the level of individual transformer computations, nor does it show that they arise from a single, well-isolated internal mechanism. Determining how these expectation-related patterns are causally produced would require targeted interventions on model activations or training-time analyses, which we leave for future work.

In addition, our experiments focus on controlled prompt settings, primarily involving few-shot classification and structured function-calling templates. These settings allow us to isolate expectation-related effects, but they do not cover the full diversity of real-world LLM usage. Whether similar patterns emerge in noisier or more open-ended contexts—such as long-form reasoning, multi-turn dialogue, or prompts with less regular formatting—remains an open question.

References

Andre M Bastos, W Martin Usrey, Rick A Adams, George R Mangun, Pascal Fries, and Karl J Friston. 2012. Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Leonardo Berti, Flavio Giorgi, and Gjergji Kasneci. 2025. Emergent abilities in large language models: A survey. *arXiv preprint arXiv:2503.05788*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Junhao Chen, Shengding Hu, Zhiyuan Liu, and Maosong Sun. 2024a. States hidden in hidden states: Llms emerge discrete state representations implicitly. *arXiv preprint arXiv:2407.11421*.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, et al. 2024b. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *ICLR*.

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

Karl Friston. 2010. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127–138.

Karl Friston and Stefan Kiebel. 2009. Predictive coding under the free-energy principle. *Philosophical transactions of the Royal Society B: Biological sciences*, 364(1521):1211–1221.

Barbara J Grosz and Candace L Sidner. 1986. Attentions, intentions, and the structure of discourse. *Computational linguistics*, 12(3):175–204.

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

Walter Kintsch and Teun A Van Dijk. 1978. Toward a model of text comprehension and production. *Psychological review*, 85(5):363.

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

Johnny Lin. 2023. Neuronpedia: Interactive reference and tooling for analyzing neural networks. Software available from neuronpedia.org.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. 2025. On the biology of a large language model. *Transformer Circuits Thread*.

M. M. Merzenich, R. J. Nelson, M. P. Stryker, M. S. Cynader, A. Schoppmann, and J. M. Zook. 1984. Somatosensory cortical map changes following digit amputation in adult monkeys. *Journal of Comparative Neurology*, 224(4):591–605. A seminal study on cortical remapping, demonstrating how brain regions are recruited by adjacent functions when their primary input is lost.

John O'Keefe and Jonathan Dostrovsky. 1971. The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat. *Brain research*.

Shishir G. Patil, Huanzhi Mao, Charlie Cheng-Jie Ji, Fanjia Yan, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*.

Rajesh PN Rao and Dana H Ballard. 1999. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87.

David A Rosenbaum, Sandra B Kenny, and Mitchell A Derr. 1983. Hierarchical control of rapid movement sequences. *Journal of Experimental Psychology: Human Perception and Performance*, 9(1):86.

N. Sadato, A. Pascual-Leone, J. Grafman, V. Ibañez, M. P. Deiber, and M. Hallett. 1996. Activation of the primary visual cortex by braille reading in blind subjects. *Nature*, 380(6574):526–528. A foundational study demonstrating functional plasticity and neural recruitment in the human brain.

Lee Sharkey and Dan Braun Beren. 2022. [interim research report] taking features out of superposition with sparse autoencoders.

Language Model Interpretability Team. 2024. Gemma scope: Helping the safety community shed light on the inner workings of language models.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Joseph Weizenbaum. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems (NIPS)*, pages 649–657.

A AGNews and Tool-Calling Prompt Examples

For our evaluation, we utilize the AG News dataset (Zhang et al., 2015) and the Berkeley Function Calling Leaderboard (BFCL) (Patil et al., 2025). AGNews is a large-scale text classification benchmark. The dataset consists of news articles categorized into four distinct topics: World, Sports, Business, and Science/Technology. We chose this dataset because its clear semantic boundaries provide an ideal environment for observing how model features—specifically FPB and CP features—respond to shifts in topic-specific terminology and structural transitions.

The Berkeley Function Calling Leaderboard (BFCL) serves as a rigorous evaluation framework for the task of tool-calling (also known as function calling). It assesses a model's capacity to map natural language intent to structured API calls by evaluating its performance across diverse categories, including nested function calls, parallel executions, and dynamic parameter filling. We utilize the BFCL to validate the functional significance of our identified features, as it provides the high-fidelity structural complexity necessary to observe CP-feature dynamics in action.

B Cyclic Parsing (CP) Features in Other Models

We demonstrate the existence of Cyclic Parsing (CP) features across two additional models (see Figure 5, 6). Notably, these parsing behaviors are functionally congruent across all architectures, providing compelling evidence that CP-features represent an emergent, universal mechanism within diverse neural systems. This suggests that such structural decomposition is a fundamental strategy developed by Transformer-based models to navigate complex, long-context inputs.

C Extended Analysis of Predictive Coding Behaviors (PCB) in LLMs

We present additional experimental results characterizing Predictive Coding Behavior (PCB) features across two architecturally distinct models: `Llama-3.1-8B-Instruct` and `Gemma-2-2B-IT` (see Figure 7). Our findings reveal a striking consistency in activation patterns across these models, reinforcing the hypothesis that PCB is a convergent emergent behavior inherent to large-scale Transformer architectures. The conservation of these fea-

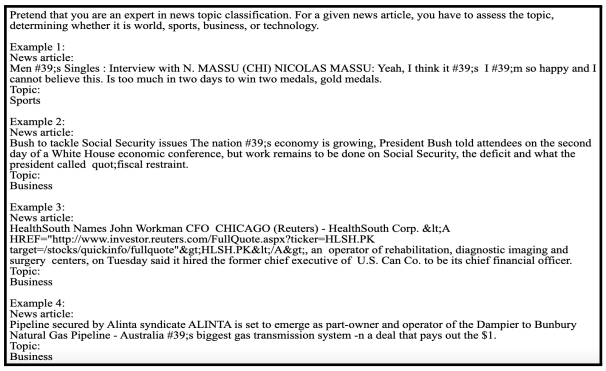


Figure 4: **AGNews and Tool-Calling Prompt Templates** (Left) AGNews prompt template (Right) Tool-Calling prompt template.

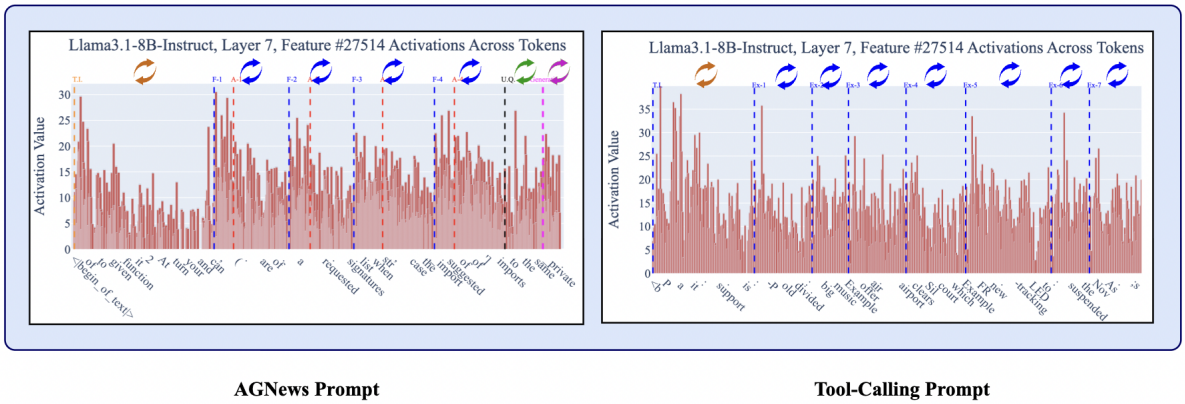


Figure 5: **Cross-Model Comparison of Cyclic Parsing Features.** The activation profile of the CP-feature in `Llama-3.1-8B-Instruct` above exhibits a functional topology nearly identical to that of `Gemma-2-9B-IT`. This high degree of structural conservation across different model families suggests that cyclic parsing is a convergent solution for managing structured input hierarchies.

tures suggests that the model's internal predictive mapping is a fundamental mechanism for managing structured information, independent of specific model scale or training lineage.

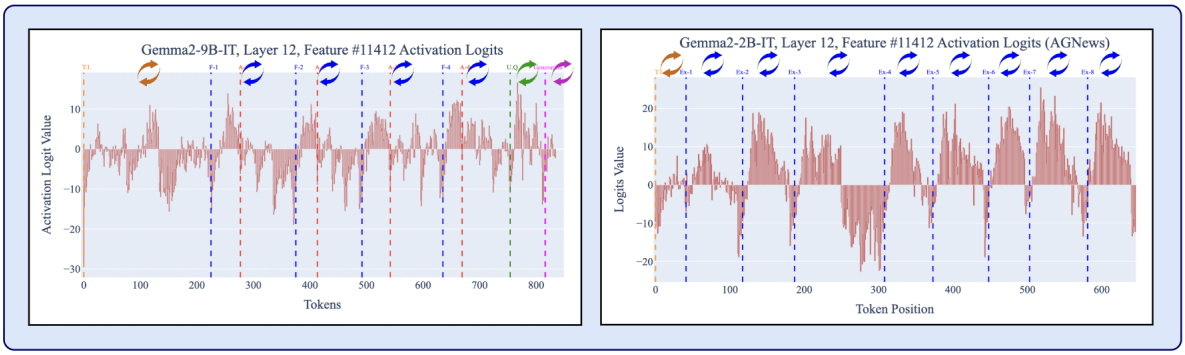
D Detailed Results on Functional Plasticity Behaviors in LLMs

We present a comprehensive analysis of Functional Plasticity–like Behavior (FPB) features in Figures 8 and 9. Our results yield several key observations. (1) Each label in the AGNews dataset is associated with a distinct cohort of label-specific features. (2) As shown in the volcano plots, these features exhibit a consistent positive shift in effect size (Cohen's d) when their primary trigger label is excluded from the input context. This pattern indicates that, in the absence of a primary semantic trigger, the affected features do not become dormant; instead, the model reallocates these specialized features to process the remaining information. This behavior provides empirical evidence for representational plasticity within the model's latent space. (3) Importantly, this pattern is consistent across all three models we evaluate.

E Detailed Results on Implicit Expectations over the Number of Few-Shot Exemplars

This section presents detailed results on model expectations over the number of few-shot exemplars. We extend the analysis from the main paper to a broader set of models, covering a range of architectures and parameter scales. Specifically, we analyze the following instruction-tuned models:

- `Llama-3.1-8B-Instruct`,
- `Llama-3.2-3B-Instruct`,
- `Llama-3.2-1B-Instruct`,



AGNews Prompt **Tool-Calling Prompt**

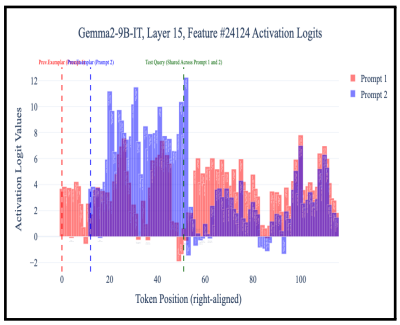
Figure 6: **Cyclic Parsing Behavior in Smaller-Scale Models.** Even the `Gemma-2-2B-IT` model exhibits a cyclic parsing mechanism, though the activation signal is significantly noisier compared to the more refined patterns observed in larger-scale models.

- `Gemma2-9B-IT`,
- and `Gemma2-2B-IT`.

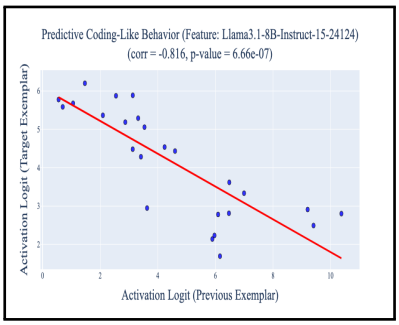
Figure 11 summarizes the results of this analysis across all five models. For each model, we perform a Decoder Logit Lens analysis using the final-layer hidden state, probing the model's predicted probability of emitting either an end-of-sequence (EOS) token or the token `Example` at each exemplar boundary (after Example k, before Example $k{+}1$). All results are averaged over 1,000 randomly constructed few-shot prompts.

Across models, we observe a consistent pattern of implicit bias toward particular exemplar counts. In four out of the five models, probability peaks align with multiples of five or ten exemplars, despite the absence of any task-level incentive favoring such counts. This suggests that these preferences are not induced by the experimental setup, but instead reflect structural or training-related regularities learned by the models.

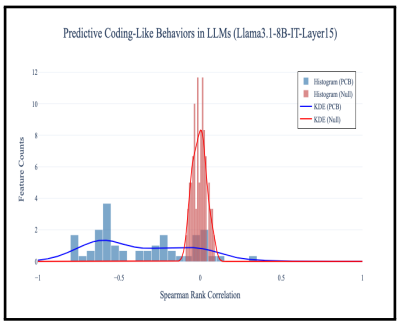
Notably, `Gemma2-2B-IT` exhibits substantially weaker bias and a comparatively uniform distribution over exemplar counts. Nevertheless, the prevalence of structured biases across the majority of models indicates that expectations over exemplar number are a common phenomenon.



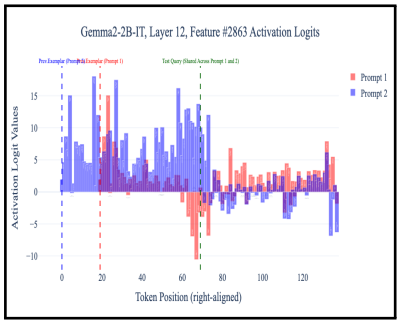
(a) PCB Example (Activation Flip)



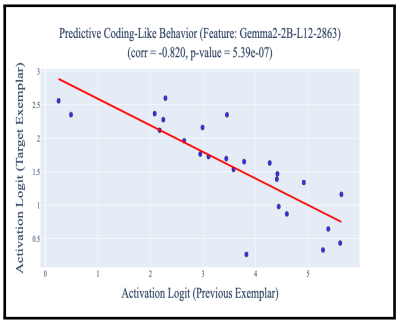
(b) PCB Example (Strong Negative Corr.)



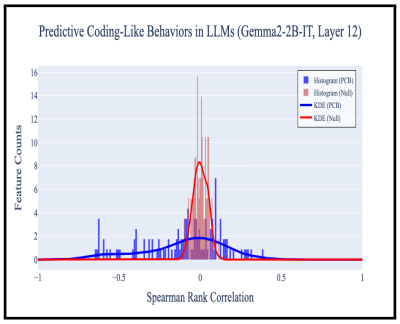
(c) Distribution of Spearman Corr.



(d) Distribution of Spearman Corr.

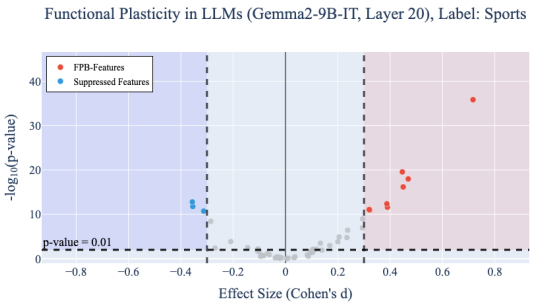


(e) Distribution of Spearman Corr.

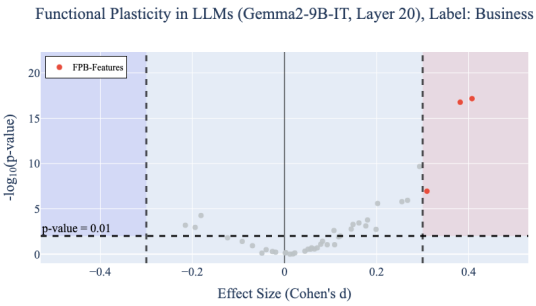


(f) Distribution of Spearman Corr.

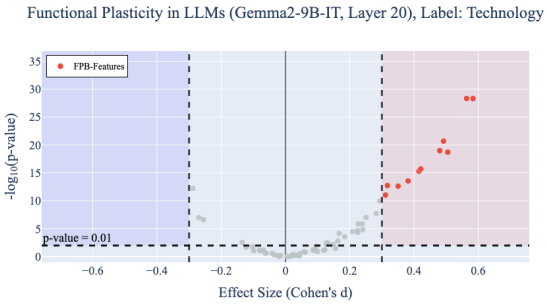
Figure 7: **Predictive Coding-Like Behavior (PCB) in `Llama3.1-8B-Instruct` and `Gemma2-2B-IT`.** Description analogous to that in Figure 2. We observe identical tendencies, yet a much stronger PCB behaviors in `Llama3.1-8B-Instruct`. These consistent results across model families and parameter scales strongly support our PCB hypothesis.



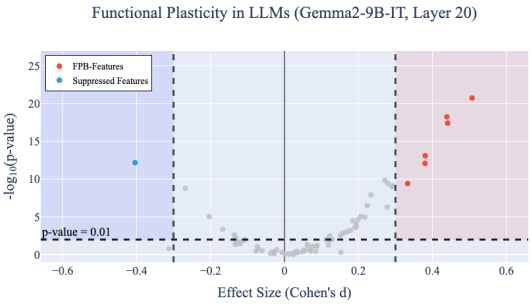
(a) Gemma2-9B-IT, L20, Removing “Sports”



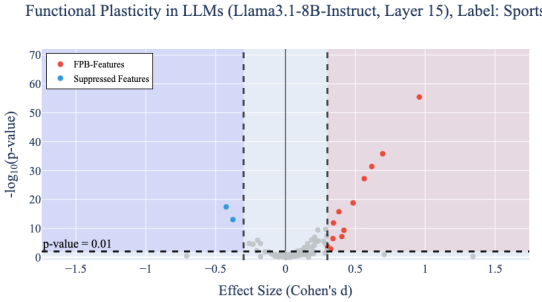
(b) Gemma2-9B-IT, L20, Removing “Business”



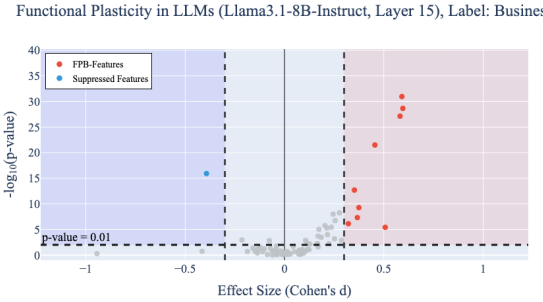
(c) Gemma2-9B-IT, L20, Removing “Technology”



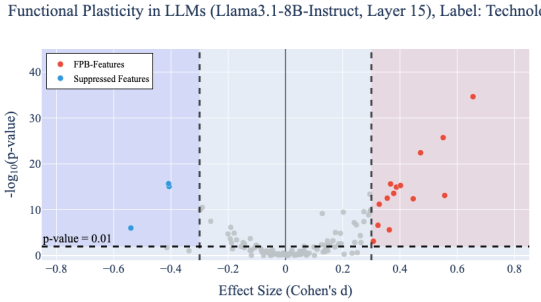
(d) Gemma2-9B-IT, L20, Removing “World”



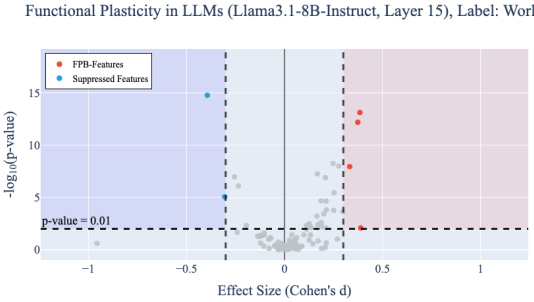
(e) Llama3.1-8B-IT, L15, Removing “Sports”



(f) Llama3.1-8B-Instruct, L15, Removing “Business”

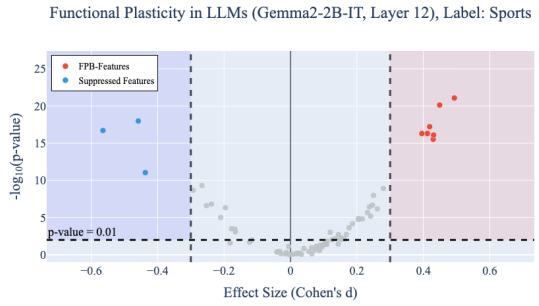


(g) Llama3.1-8B-IT, L15, Removing “Technology”

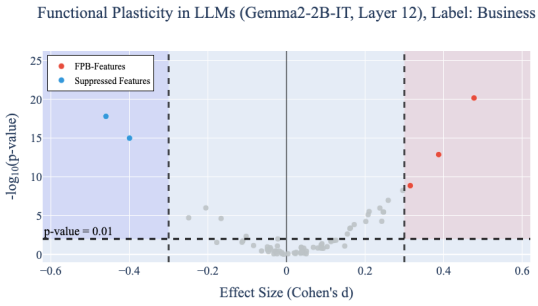


(h) Llama3.1-8B-Instruct, L15, Removing “World”

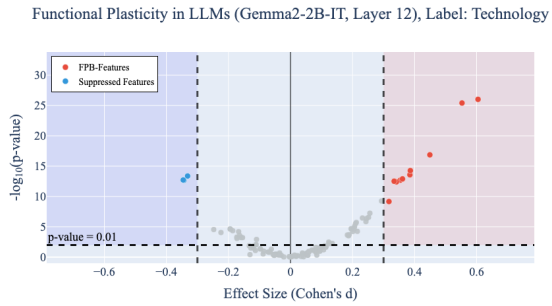
Figure 8: a



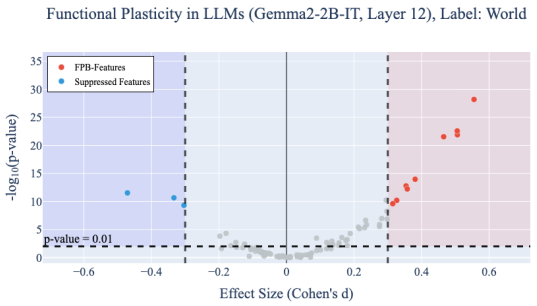
(a) Gemma2-2B-IT, L12, Removing “Sports”



(b) Gemma2-2B-IT, L12, Removing “Business”

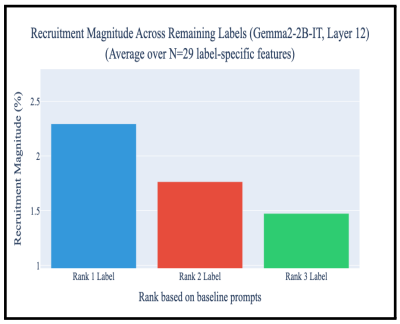


(c) Gemma2-2B-IT, L12, Removing “Technology”

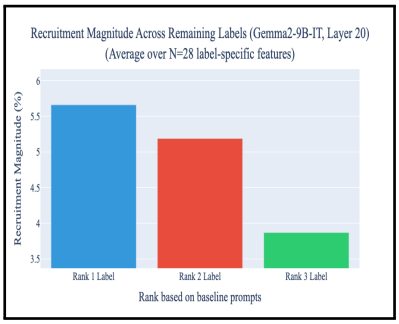


(d) Gemma2-2B-IT, L12, Removing “World”

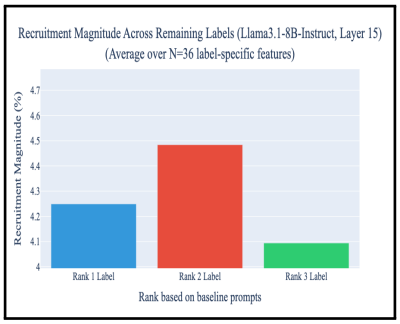
Figure 9: a



(a) Gemma2-2B-IT



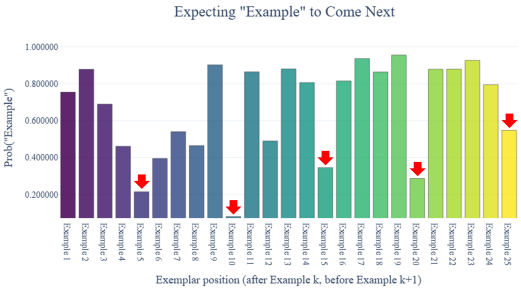
(b) Gemma2-9B-IT



(c) Llama3.1-8B-Instruct

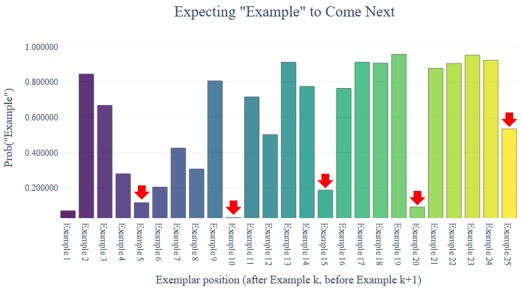
Figure 10: **Directed Migration of Functionality in FPB- Features.** When label-specific FPB-features encounter the absence of their primary trigger, they demonstrate a functional reallocation toward the remaining labels. Critically, this migration is not stochastic; instead, features preferentially shift their activity toward labels for which they exhibited an initial semantic affinity (i.e., "friendly" labels), suggesting that the migration follows a structured, latent proximity map rather than random activation.

(a) `Llama-3.2-1B-Instruct`: $p(\text{EOS})$.



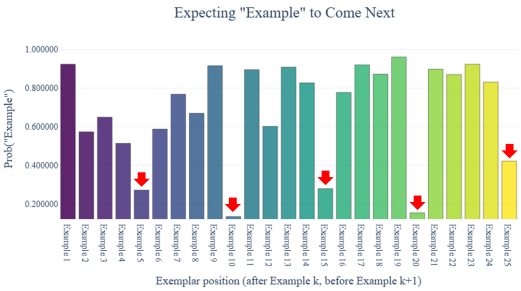
(b) `Llama-3.2-1B-Instruct`: $p(\texttt{Example})$.

(c) `Llama-3.2-3B-Instruct`: $p(\text{EOS})$.

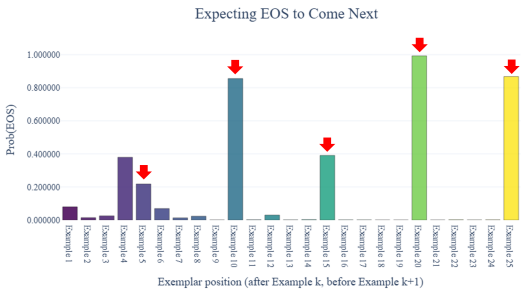


(d) `Llama-3.2-3B-Instruct`: $p(\texttt{Example})$.

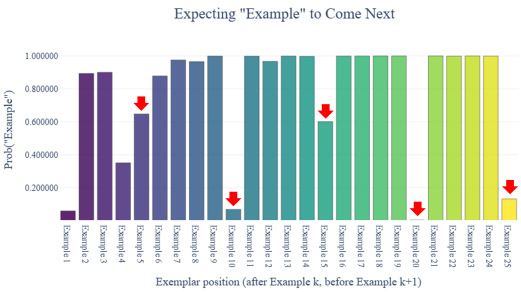
(e) `Llama-3.1-8B-Instruct`: $p(\text{EOS})$.



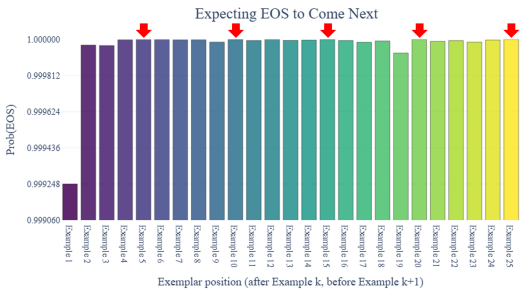
(f) `Llama-3.1-8B-Instruct`: $p(\texttt{Example})$.



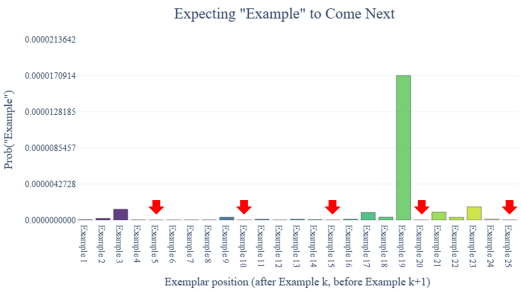
(g) `Gemma2-9B-IT`: $p(\text{EOS})$.



(h) `Gemma2-9B-IT`: $p(\texttt{Example})$.



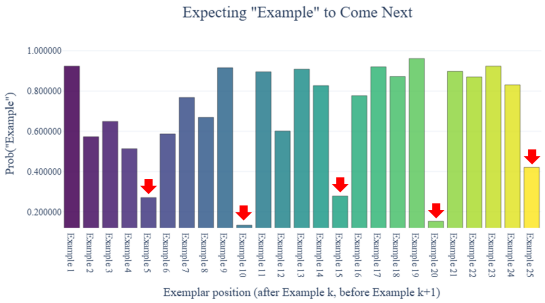
(i) `Gemma2-2B-IT`: $p(\text{EOS})$.



(j) `Gemma2-2B-IT`: $p(\texttt{Example})$.

Figure 11: Extension of Figure 12 to five models under the same setup. Most models show peaks at multiples of five or ten, whereas `Gemma2-2B-IT` is comparatively uniform.

(a) `Llama-3.1-8B-Instruct`: $p(\text{EOS})$.



(b) `Llama-3.1-8B-Instruct`: $p(\texttt{Example})$.

(c) `Gemma2-9B-IT`: $p(\text{EOS})$.

(d) `Gemma2-9B-IT`: $p(\texttt{Example})$.

Figure 12: Decoder Logit Lens analysis of implicit expectations over the number of few-shot exemplars, averaged over 1,000 randomly constructed prompts. Results are shown for `Llama-3.1-8B-Instruct` and `Gemma2-9B-IT`. Probes are taken at each exemplar boundary (after Example k, before Example $k+1$) using the final-layer hidden state. The model exhibits a bias over exemplar counts; in particular, peaks align first with multiples of ten exemplars and then with multiples of five (red arrows), even though the task setup provides no reason for such a preference in exemplar count. For Llama, the end-of-sequence (EOS) token corresponds to `<|eot_id|>` or `<|end_of_text|>`, whereas for Gemma it corresponds to `<eos>` or `<end_of_turn>`. For the `Example` panels, probabilities aggregate `Example` along with simple case/space variants. Additional results for a broader set of models are provided in Figure 11.