

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

076 exemplars, reminiscent of how humans may uncon-
077 sciously form expectations about the quantity of
078 examples to be presented.

079 It is important to emphasize that our goal is not
080 to anthropomorphize LLMs, but rather to draw on
081 well established findings from human cognition
082 as a principled and rich source of inspiration for
083 experimental design. By bridging insights from bi-
084 ological cognition with computational analysis, we
085 design experiments that expose intriguing internal
086 patterns that have not been previously documented.
087 Such patterns would be very difficult to uncover us-
088 ing purely bottom up approaches without guidance
089 from human behavioral insights. Our results indi-
090 cate that treating human cognition as a conceptual
091 mirror provides a powerful and systematic method-
092 ology for revealing hidden structure in artificial
093 intelligence systems.

094 2 Related Work

095 2.1 Studies on the Emergent Capabilities in 096 LLMs

097 Emergent capabilities are behaviors or abilities that
098 are not explicitly trained for but arise as language
099 models scale to large sizes (Berti et al., 2025). Prior
100 work has shown a wide range of such emergent cap-
101 abilities in LLMs, including in-context learning
102 (Wei et al., 2022), collaborative behaviors (Chen
103 et al., 2024b), and arithmetic ability (Chen et al.,
104 2024a). However, while these studies characterize
105 what capabilities emerge at the level of task per-
106 formance, the question of *which internal mechanisms*
107 *or processing heuristics emerge inside the model*
108 remains underexplored. In this work, we shift the
109 focus from emergent *outcomes* to emergent *mech-*
110 *anisms*, investigating how large language models
111 internally parse, organize, and integrate prompt in-
112 formation in ways that systematically shape their
113 downstream behavior.

114 2.2 Anthropomorphisms in LLMs

115 Anthropomorphism refers to the tendency to at-
116 tribute human-like understanding or intentions to
117 non-human systems, a phenomenon commonly il-
118 lustrated by the ELIZA effect (Weizenbaum, 1966).
119 Rather than merely anthropomorphizing AI, we use
120 it as a methodological lens for generating hypothe-
121 ses about internal processing mechanisms. For
122 example, insights from human cognition can offer
123 useful clues about potential computational strate-
124 gies for processing structured inputs, such as Pre-

125 Predictive Coding (Rao and Ballard, 1999).

126 Predictive coding is a prominent theory in cog-
127 nitive neuroscience that suggests that cognition is
128 shaped by how strongly the brain responds to in-
129 formation that aligns with or violates prior expecta-
130 tions. Information that matches expectations tends
131 to evoke weaker responses, while unexpected input
132 produces stronger signals that prompt updates to
133 internal representations (Friston and Kiebel, 2009;
134 Friston, 2010; Bastos et al., 2012). Despite its
135 prominence in neuroscience, whether predictive
136 coding-like behavior arises in large language mod-
137 els remains an open question.

138 In biological systems, progress in understand-
139 ing cognition has often come from identifying the
140 functional roles of specific neurons or neuronal
141 populations. A canonical example is the discov-
142 ery of place cells in the rodent hippocampus, which
143 revealed how spatial information is internally repre-
144 sented during navigation (O’Keefe and Dostrovsky,
145 1971). In contrast, despite the fact that all internal
146 activations in artificial neural networks are fully
147 accessible, our mechanistic understanding of large
148 language models remains limited.

149 Prior work has identified neurons or features that
150 respond selectively to particular concepts, such as
151 the Golden Gate Bridge or multilingualism (Tem-
152 pleton et al., 2024), and has led to resources like
153 Neuronpedia (Lin, 2023) that provide annotations
154 describing common patterns in next-token predic-
155 tions generated using LLMs as judges. However,
156 because such annotations are limited to commonali-
157 ties in next-token prediction, they offer little insight
158 into why a neuron becomes active, and our work ad-
159 dresses this limitation by drawing inspiration from
160 human subjective experience to generate mechanistic
161 hypotheses about neuron activation during task
162 execution.

163 2.3 Sparse Autoencoders

164 One of the most effective approaches for making
165 the neural activations of large language models
166 (LLMs) interpretable to humans is the Sparse Au-
167 toencoder (SAE) (Sharkey and Beren, 2022). A ma-
168 jor obstacle to understanding LLMs is that their in-
169 ternal representations are expressed as vectors that
170 are not directly human-interpretable. Features dis-
171 entangled by SAEs, however, are often regarded as
172 analogous to a *microscope* that allows us to probe
173 the internal mechanisms of large language models
(Team, 2024; Lindsey et al., 2025).

174 Specifically, while individual neurons in LLMs
175

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:

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

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:

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

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 $\mathbf{a} = (a_1, a_2, \dots, a_T)$ 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,

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

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

289 The candidate CP feature is shown in Figure
290 1. As illustrated by the representative feature
291 Gemma2-9B-L9-4919, CP features exhibit a recur-
292 ring activation pattern: activation peaks when the
293 model internally identifies the beginning of a new
294 semantic block and then gradually decreases as
295 processing proceeds toward the end of that block.
296 Note that these boundaries are not supplied by any
297 explicit supervision or annotation (although the
298 model does appear to partially rely on the newline
299 token “\n”, which is natural); instead, they arise
300 from the model’s own internal dynamics. This
301 consistent cyclic behavior (across different task types
302 and examples) suggests that the LLM implicitly
303 forms and maintains an internal notion of semantic
304 blocks during processing, reflecting how it parses
305 and organizes the structure of the input. In other
306 words, the activation magnitude of a CP-feature
307 functions as a predictive signal: a high-magnitude
308 negative activation signals the imminent conclusion
309 of the current semantic block. This suggests the
310 model internally anticipates structural transitions,
311 allowing it to prepare for the subsequent segment
312 in the input stream.

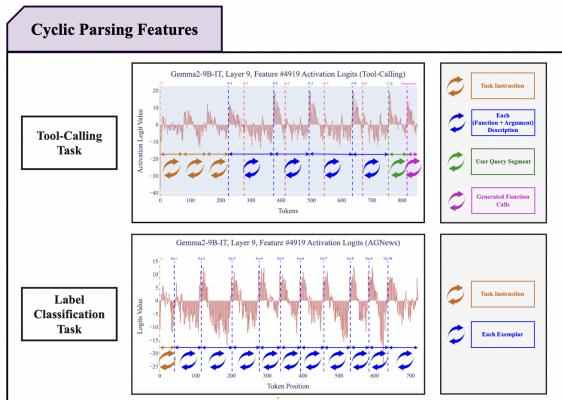
313 For instance, in the AGNews prompt (bottom
314 panel of Figure 1), which consists of (1) a task
315 instruction followed by (2) a set of few-shot ex-
316emplars, we observe that the model identifies and
317 treats each exemplar as a distinct semantic block,
318 in a manner that closely aligns with how a human
319 reader would naturally segment the prompt.

320 Similarly, in tool-calling prompts, which con-
321 tain (1) a task instruction followed by (2) a set
322 of function-description and argument-description
323 pairs, (3) a user query, and (4) the model’s gener-

324 ated output, we observe in the top panel of Figure 1
325 that the model acknowledges multiple semantic
326 blocks. In particular, the model treats the task in-
327 struction as comprising three distinct blocks, re-
328 gards each function–argument description pair as
329 a single combined semantic block, identifies the
330 user query as another block, and then recognizes
331 the generation phase as a new semantic block. This
332 parsing behavior closely mirrors how humans are
333 likely to read and structure such prompts, demon-
334 strating a high degree of alignment between the
335 model’s internal parsing dynamics and human read-
336 ing behavior.

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

343 Notably, these specialized, universal features
344 might have remained overlooked had we not drawn
345 inspiration from human cognitive behaviors. This
346 serves as a compelling case study for **human-
347 centric interpretability**: by using human behav-
348 ior patterns as a heuristic, we can uncover ‘hidden’
349 latent mechanisms in LLMs that purely auto-
350 mated discovery might miss.



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

351
352

3.2 Predictive Coding-Like Behaviors in LLMs

353
354
355
356
357
358
359
360
361
362
363
364
365
366
367

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.

368
369
370
371
372
373
374
375
376
377
378
379

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.

380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397

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.

398
399
400
401

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

402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420

of the form $(\text{Exemplar}_{\text{prev}}, \text{Exemplar}_{\text{tgt}})$ 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.

421
422
423
424
425
426
427
428
429
430
431

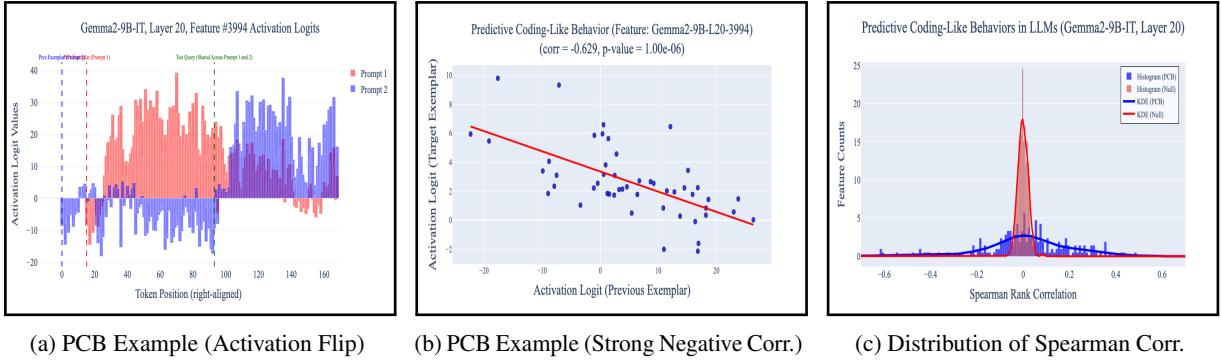
Intuitively, if a feature were strictly context-independent, its activation a_{tgt} would remain invariant across all N exemplars. However, the PCB hypothesis suggests that for a specific class of features, a_{tgt} fluctuates as an inverse function of the activation a_{prev} . To quantify this, we first 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:

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

432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450

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.*

451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
10010
10011
10012
10013
10014
10015
10016
10017
10018
10019
10020
10021
10022
10023
10024
10025
10026
10027
10028
10029
10030
10031
10032
10033
10034
10035
10036
10037
10038
10039
10040
10041
10042
10043
10044
10045
10046
10047
10048
10049
10050
10051
10052
10053
10054
10055
10056
10057
10058
10059
10060
10061
10062
10063
10064
10065
10066
10067
10068
10069
10070
10071
10072
10073
10074
10075
10076
10077
10078
10079
10080
10081
10082
10083
10084
10085
10086
10087
10088
10089
10090
10091
10092
10093
10094
10095
10096
10097
10098
10099
100100
100101
100102
100103
100104
100105
100106
100107
100108
100109
100110
100111
100112
100113
100114
100115
100116
100117
100118
100119
100120
100121
100122
100123
100124
100125
100126
100127
100128
100129
100130
100131
100132
100133
100134
100135
100136
100137
100138
100139
100140
100141
100142
100143
100144
100145
100146
100147
100148
100149
100150
100151
100152
100153
100154
100155
100156
100157
100158
100159
100160
100161
100162
100163
100164
100165
100166
100167
100168
100169
100170
100171
100172
100173
100174
100175
100176
100177
100178
100179
100180
100181
100182
100183
100184
100185
100186
100187
100188
100189
100190
100191
100192
100193
100194
100195
100196
100197
100198
100199
100200
100201
100202
100203
100204
100205
100206
100207
100208
100209
100210
100211
100212
100213
100214
100215
100216
100217
100218
100219
100220
100221
100222
100223
100224
100225
100226
100227
100228
100229
100230
100231
100232
100233
100234
100235
100236
100237
100238
100239
100240
100241
100242
100243
100244
100245
100246
100247
100248
100249
100250
100251
100252
100253
100254
100255
100256
100257
100258
100259
100260
100261
100262
100263
100264
100265
100266
100267
100268
100269
100270
100271
100272
100273
100274
100275
100276
100277
100278
100279
100280
100281
100282
100283
100284
100285
100286
100287
100288
100289
100290
100291
100292
100293
100294
100295
100296
100297
100298
100299
100300
100301
100302
100303
100304
100305
100306
100307
100308
100309
100310
100311
100312
100313
100314
100315
100316
100317
100318
100319
100320
100321
100322
100323
100324
100325
100326
100327
100328
100329
100330
100331
100332
100333
100334
100335
100336
100337
100338
100339
100340
100341
100342
100343
100344
100345
100346
100347
100348
100349
100350
100351
100352
100353
100354
100355
100356
100357
100358
100359
100360
100361
100362
100363
100364
100365
100366
100367
100368
100369
100370
100371
100372
100373
100374
100375
100376
100377
100378
100379
100380
100381
100382
100383
100384
100385
100386
100387
100388
100389
100390
100391
100392
100393
100394
100395
100396
100397
100398
100399
100400
100401
100402
100403
100404
100405
100406
100407
100408
100409
100410
100411
100412
100413
100414
100415
100416
100417
100418
100419
100420
100421
100422
100423
100424
100425
100426
100427
100428
100429
100430
100431
100432
100433
100434
100435
100436
100437
100438
100439
100440
100441
100442
100443
100444
100445
100446
100447
100448
100449
100450
100451
100452
100453
100454
100455
100456
100457
100458
100459
100460
100461
100462
100463
100464
100465
100466
100467
100468
100469
100470
100471
100472
100473
100474
100475
100476
100477
100478
100479
100480
100481
100482
100483
100484
100485
100486
100487
100488
100489
100490
100491
100492
100493
100494
100495
100496
100497
100498
100499
100500
100501
100502
100503
100504
100505
100506
100507
100508
100509
100510
100511
100512
100513
100514
100515
100516
100517
100518
100519
100520
100521
100522
100523
100524
100525
100526
100527
100528
100529
100530
100531
100532
100533
100534
100535
100536
100537
100538
100539
100540
100541
100542
100543
100544
100545
100546
100547
100548
100549
100550
100551
100552
100553
100554
100555
100556
100557
100558
100559
100560
100561
100562
100563
100564
100565
100566
100567
100568
100569
100570
100571
100572
100573
100574
100575
100576
100577
100578
100579
100580
100581
100582
100583
100584
100585
100586
100587
100588
100589
100590
100591
100592
100593
100594
100595
100596
100597
100598
100599
100600
100601
100602
100603
100604
100605
100606
100607
100608
100609
100610
100611
100612
100613
100614
100615
100616
100617
100618
100619
100620
100621
100622
100623
100624
100625
100626
100627
100628
100629
100630
100631
100632
100633
100634
100635
100636
100637
100638
100639
100640
100641
100642
100643
100644
100645
100646
100647
100648
100649
100650
100651
100652
100653
100654
100655
100656
100657
100658
100659
100660
100661
100662
100663
100664
100665
100666
100667
100668
100669
100670
100671
100672
100673
100674
100675
100676
100677
100678
100679
100680
100681
100682
100683
100684
100685
100686
100687
100688
100689
100690
100691
100692
100693
100694
100695
100696
100697
100698
100699
100700
100701
100702
100703
100704
100705
100706
100707
100708
100709
100710
100711
100712
100713
100714
100715
100716
100717
100718
100719
100720
100721
100722
100723
100724
100725
100726
100727
100728
100729
100730
100731
100732
100733
100734
100735
100736
100737
100738
100739
100740
100741
100742
100743
100744
100745
100746
100747
100748
100749
100750
100751
100752
100753
100754
100755
100756
100757
100758
100759
100760
100761
100762
100763
100764
100765
100766
100767
100768
100769
100770
100771
100772
100773
100774
100775
100776
100777
100778
100779
100780
100781
100782
100783
100784
100785
100786
100787
100788
100789
100790
100791
100792
100793
100794
100795
100796
100797
100798
100799
100800
100801
100802
100803
100804
100805
100806
100807
100808
100809
100810
100811
100812
100813
100814
100815
100816
100817
100818
100819
100820
100821
100822
100823
100824
100825
100826
100827
100828
100829
100830
100831
100832
100833
100834
100835
100836
100837
100838
100839
100840
100841
100842
100843
100844
100845



(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.

451
452
453
designed to maintain and aggregate context through
residual connections—one would expect “carry-on
semantics” to be the dominant motif.

454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
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 (re-
sults on different models provided in Appendix C).
A Kolmogorov-Smirnov test confirms that the ob-
served 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.

469
470
471
472
473
474
475
476
477
**Practical Implications of PCB for Model Steer-
ing** Beyond characterizing this tendency in
LLMs, we believe the Predictive Coding-like Be-
havior (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.

478
479
480
481
482
483
For example, standard activation steering ap-
proaches often apply feature amplification or sup-
pression uniformly across an entire prompt. How-
ever, our findings suggest that such strategies may
yield counterintuitive results: amplifying a specific
feature in a preceding exemplar may inadvertently

484
485
486
trigger a compensatory suppression of that same
feature when the model processes the target query.
This interaction could effectively diminish or nul-
lify the intended steering effect. Consequently, we
suggest that accounting for PCB dynamics is es-
sential for designing more robust steering interven-
tions. 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

494
495
496
497
498
499
500
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 sub-
strates can support diverse cognitive functions de-
pending on immediate input or environmental con-
straints, enabling robust adaptation within a fixed
underlying architecture (Sadato et al., 1996).

501
502
503
504
505
Drawing inspiration from this biological adapt-
ability, 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 contex-
tual changes. Surprisingly, we identified a substan-
tial 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 re-

maining 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 $-\log_{10}(p\text{-value})$ —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

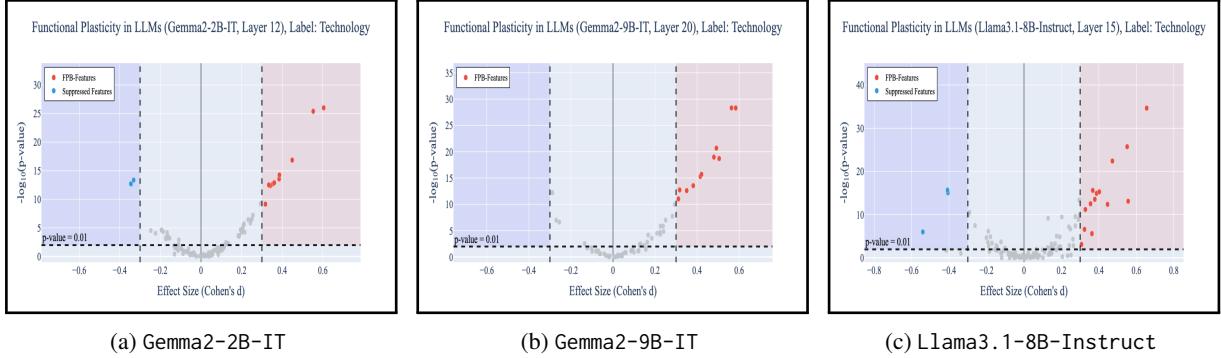


Figure 3: **Illustration of the Functional Plasticity-like Behavior (FPB) in LLMs.** Each subfigure shows ...Volcano plots of feature activation shifts exhibit a pronounced positive skew, identifying a significant population of "recruited" features. ... Aggregate recruitment magnitude as a function of 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 pre-training 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.

686 5 Limitations

687 Our analyses are designed to identify internal
688 signals and representational patterns that corre-
689 late with long-range expectations in LLMs dur-
690 ing prompt processing. While these signals are
691 predictive of specific continuation behaviors, our
692 study does not establish how these patterns are im-
693 plemented at the level of individual transformer
694 computations, nor does it show that they arise from
695 a single, well-isolated internal mechanism. De-
696 termining how these expectation-related patterns
697 are causally produced would require targeted in-
698 terventions on model activations or training-time
699 analyses, which we leave for future work.

700 In addition, our experiments focus on controlled
701 prompt settings, primarily involving few-shot clas-
702 sification and structured function-calling templates.
703 These settings allow us to isolate expectation-
704 related effects, but they do not cover the full di-
705 versity of real-world LLM usage. Whether simi-
706 lar patterns emerge in noisier or more open-ended
707 contexts—such as long-form reasoning, multi-turn
708 dialogue, or prompts with less regular format-
709 ting—remains an open question.

710 References

711 Andre M Bastos, W Martin Usrey, Rick A Adams,
712 George R Mangun, Pascal Fries, and Karl J Friston.
713 2012. Canonical microcircuits for predictive coding.
714 *Neuron*, 76(4):695–711.

715 Nora Belrose, Zach Furman, Logan Smith, Danny Ha-
716 liwi, Igor Ostrovsky, Lev McKinney, Stella Bider-
717 man, and Jacob Steinhardt. 2023. Eliciting latent
718 predictions from transformers with the tuned lens.
719 *arXiv preprint arXiv:2303.08112*.

720 Leonardo Berti, Flavio Giorgi, and Gjergji Kasneci.
721 2025. Emergent abilities in large language models:
722 A survey. *arXiv preprint arXiv:2503.05788*.

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

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

733 Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang,
734 Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu,
735 Yi-Hsin Hung, Chen Qian, et al. 2024b. Agentverse:

736 Facilitating multi-agent collaboration and exploring
737 emergent behaviors. In *ICLR*.

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

743 Karl Friston. 2010. The free-energy principle: a uni-
744 fied brain theory? *Nature reviews neuroscience*,
745 11(2):127–138.

746 Karl Friston and Stefan Kiebel. 2009. Predictive coding
747 under the free-energy principle. *Philosophical trans-
748 actions of the Royal Society B: Biological sciences*,
749 364(1521):1211–1221.

750 Barbara J Grosz and Candace L Sidner. 1986. Atten-
751 tions, intentions, and the structure of discourse. *Com-
752 putational linguistics*, 12(3):175–204.

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

759 Walter Kintsch and Teun A Van Dijk. 1978. Toward a
760 model of text comprehension and production. *Psy-
761 chological review*, 85(5):363.

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

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

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

782 M. M. Merzenich, R. J. Nelson, M. P. Stryker, M. S.
783 Cynader, A. Schoppmann, and J. M. Zook. 1984. *Somato-
784 sensory cortical map changes following digit*
785 *amputation in adult monkeys*. *Journal of Comparative*
786 *Neurology*, 224(4):591–605. A seminal study
787 on cortical remapping, demonstrating how brain re-
788 gions are recruited by adjacent functions when their
789 primary input is lost.

| | | |
|-----|--|-----|
| 790 | John O’Keefe and Jonathan Dostrovsky. 1971. The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat. <i>Brain research</i> . | 843 |
| 791 | | 844 |
| 792 | | 845 |
| 793 | | 846 |
| 794 | 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 <i>Forty-second International Conference on Machine Learning</i> . | 847 |
| 795 | | 848 |
| 796 | | 849 |
| 797 | | 850 |
| 798 | | 851 |
| 799 | | 852 |
| 800 | Rajesh PN Rao and Dana H Ballard. 1999. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. <i>Nature neuroscience</i> , 2(1):79–87. | 853 |
| 801 | | 854 |
| 802 | | 855 |
| 803 | | 856 |
| 804 | David A Rosenbaum, Sandra B Kenny, and Mitchell A Derr. 1983. Hierarchical control of rapid movement sequences. <i>Journal of Experimental Psychology: Human Perception and Performance</i> , 9(1):86. | 857 |
| 805 | | 858 |
| 806 | | 859 |
| 807 | | 860 |
| 808 | 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. <i>Nature</i> , 380(6574):526–528. A foundational study demonstrating functional plasticity and neural recruitment in the human brain. | 861 |
| 809 | | 862 |
| 810 | | 863 |
| 811 | | 864 |
| 812 | | 865 |
| 813 | | 866 |
| 814 | Lee Sharkey and Dan Braun Beren. 2022. [interim research report] taking features out of superposition with sparse autoencoders. | 867 |
| 815 | | 868 |
| 816 | | 869 |
| 817 | Language Model Interpretability Team. 2024. Gemma scope: Helping the safety community shed light on the inner workings of language models. | 870 |
| 818 | | 871 |
| 819 | | 872 |
| 820 | 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. <i>Transformer Circuits Thread</i> . | 873 |
| 821 | | 874 |
| 822 | | 875 |
| 823 | | 876 |
| 824 | | 877 |
| 825 | | 878 |
| 826 | | 879 |
| 827 | | 880 |
| 828 | | |
| 829 | | |
| 830 | 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. <i>Advances in neural information processing systems</i> , 35:24824–24837. | 881 |
| 831 | | 882 |
| 832 | | 883 |
| 833 | | 884 |
| 834 | | 885 |
| 835 | Joseph Weizenbaum. 1966. Eliza—a computer program for the study of natural language communication between man and machine. <i>Communications of the ACM</i> , 9(1):36–45. | 886 |
| 836 | | 887 |
| 837 | | 888 |
| 838 | | 889 |
| 839 | Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In <i>Advances in neural information processing systems (NIPS)</i> , pages 649–657. | 890 |
| 840 | | 891 |
| 841 | | |
| 842 | | |

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-

Pretend that you are an expert in news topic classification. For a given news article, you have to assess the topic, determining whether it is world, sports, business, or technology.

Example 1:
News article:
More #39;Singles : Interview with N. MASSU (CH) NICOLAS MASSU: Yeah, I think it #39;s I #39;m so happy and I cannot believe this. Is too much in two days to win two medals, gold medals.
Topic:
Sports

Example 2:
News article:
Bush to tackle Social Security issues The nation #39;s economy is growing. President Bush told attendees on the second day of a White House economic conference, but work remains to be done on Social Security, the deficit and what the president called quot;fiscal restraint.
Topic:
Business

Example 3:
News article:
HealthSouth names John Workman CFO CHICAGO (Reuters) - HealthSouth Corp. HLTH.PK, an operator of rehabilitation, diagnostic imaging and surgery centers, on Tuesday said it hired the former chief executive of U.S. Can Co. to be its chief financial officer.
Topic:
Business

Example 4:
News article:
Pipeline secured by Alinta syndicate ALINTA is set to emerge as part-owner and operator of the Dampier to Bunbury Natural Gas Pipeline - Australia #39;s biggest gas transmission system - in a deal that pays out the \$1.
Topic:
Business

You are an expert in composing functions. You are given a question and a set of possible functions. Based on the question, you will need to make one or more function/tool calls to achieve the purpose. If none of the functions can be used, point it out. If the given question lacks the parameters required by the function, also point it out.
You should only return the function calls in your response.

If you decide to invoke any of the function(s), you MUST put it in the format of func_name1(params_name1=params_value1, params_name2=params_value2...), func_name2(params). You SHOULD NOT include any other text in the response.

At each turn, you should try your best to complete the tasks requested by the user within the current turn. Continue to output functions to call until you have fulfilled the user's request to the best of your ability. Once you have no more functions to call, the system will consider the current turn complete and proceed to the next turn or task.

Here is a list of functions in python format that you can invoke.

```
# Function: get_class_info
# ...
# Retrieves information about the methods, properties, and constructor of a specified class if it exists within the module.

Args:
    class_name (str): The name of the class to retrieve information for, as it appears in the source code.
    include_private (bool, default=False): Determines whether to include private methods and properties, which are typically denoted by leading underscore.
    module_name (str, default=None): The name of the module where the class is defined. This is optional if the class is within the current module.

# Function: get_signature
# ...
# Retrieves the signature of a specified method within a given class, if available. The signature includes parameter names and their respective types.

Args:
    class_name (str): The name of the class that contains the method for which the signature is requested.
    method_name (str): The exact name of the method whose signature is to be retrieved.
    include_private (bool, default=False): A flag to indicate whether to include private methods' signatures in the search.
```

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

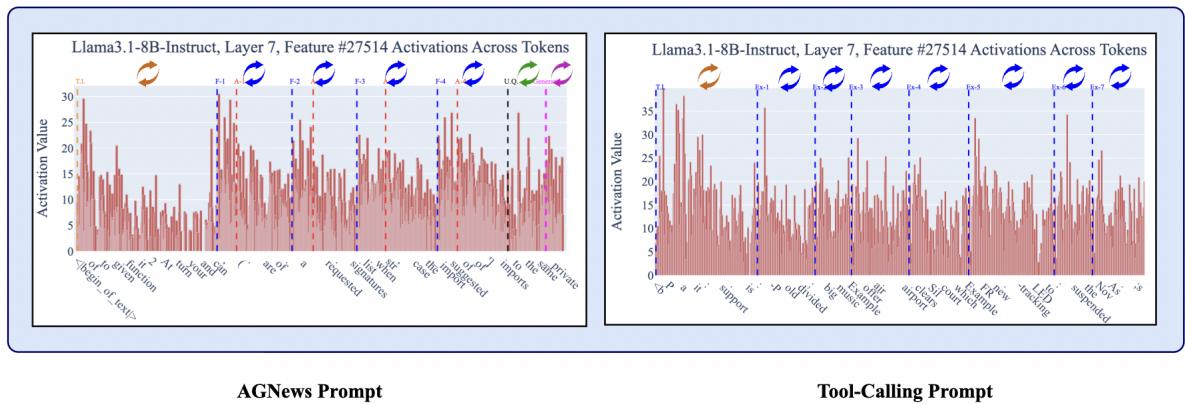


Figure 5: **Cross-Model Comparison of Cyclic Parsing Features.** The activation profile of the CP-feature in Llama-3.1-8B-Instruct above exhibits a functional topology nearly identical to that of Gemma-2-9B-IT. This high degree of structural conservation across different model families suggests that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

892 tures suggests that the model's internal predictive
893 mapping is a fundamental mechanism for man-
894 aging structured information, independent of specific
895 model scale or training lineage.

896 D Detailed Results on Functional 897 Plasticity Behaviors in LLMs

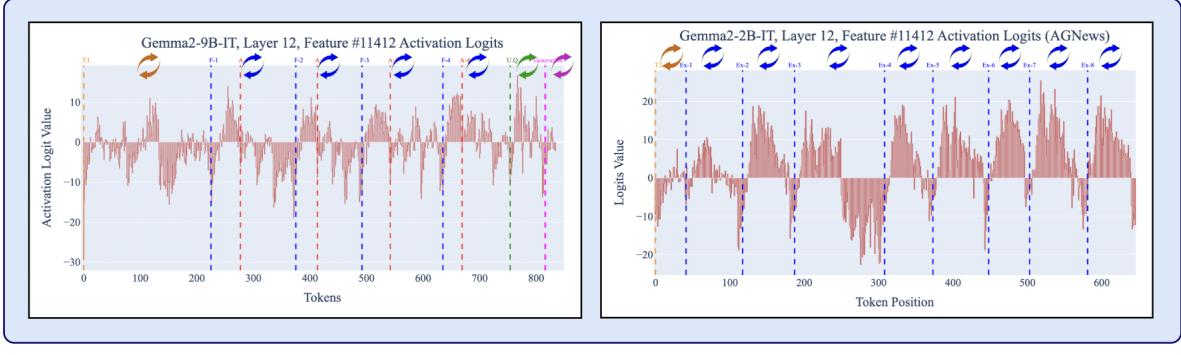
898 We present a comprehensive analysis of Functional
899 Plasticity-like Behavior (FPB) features in Figures 8
900 and 9. Our results yield several key observations.
901 (1) Each label in the AGNews dataset is associ-
902 ated with a distinct cohort of label-specific fea-
903 tures. (2) As shown in the volcano plots, these
904 features exhibit a consistent positive shift in effect
905 size (Cohen's d) when their primary trigger label
906 is excluded from the input context. This pattern
907 indicates that, in the absence of a primary semantic
908 trigger, the affected features do not become dor-
909 mant; instead, the model reallocates these special-

910 ized features to process the remaining information.
911 This behavior provides empirical evidence for rep-
912 resentational plasticity within the model's latent
913 space. (3) Importantly, this pattern is consistent
914 across all three models we evaluate.

915 E Detailed Results on Implicit 916 Expectations over the Number of 917 Few-Shot Exemplars

918 This section presents detailed results on model ex-
919 pectations over the number of few-shot exemplars.
920 We extend the analysis from the main paper to a
921 broader set of models, covering a range of ar-
922 chitectures and parameter scales. Specifically, we
923 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.

927 • Gemma2-9B-IT,

928 • and Gemma2-2B-IT.

929 Figure 11 summarizes the results of this analysis
 930 across all five models. For each model, we per-
 931 form a Decoder Logit Lens analysis using the final-
 932 layer hidden state, probing the model’s predicted
 933 probability of emitting either an end-of-sequence
 934 (EOS) token or the token Example at each exem-
 935 plar boundary (after Example k , before Example
 936 $k+1$). All results are averaged over 1,000 randomly
 937 constructed few-shot prompts.

938 Across models, we observe a consistent pattern
 939 of implicit bias toward particular exemplar counts.
 940 In four out of the five models, probability peaks
 941 align with multiples of five or ten exemplars, de-
 942 spite the absence of any task-level incentive favor-
 943 ing such counts. This suggests that these prefer-
 944 ences are not induced by the experimental setup,
 945 but instead reflect structural or training-related reg-
 946 ularities learned by the models.

947 Notably, Gemma2-2B-IT exhibits substantially
 948 weaker bias and a comparatively uniform distri-
 949 bution over exemplar counts. Nevertheless, the
 950 prevalence of structured biases across the majority
 951 of models indicates that expectations over exemplar
 952 number are a common phenomenon.

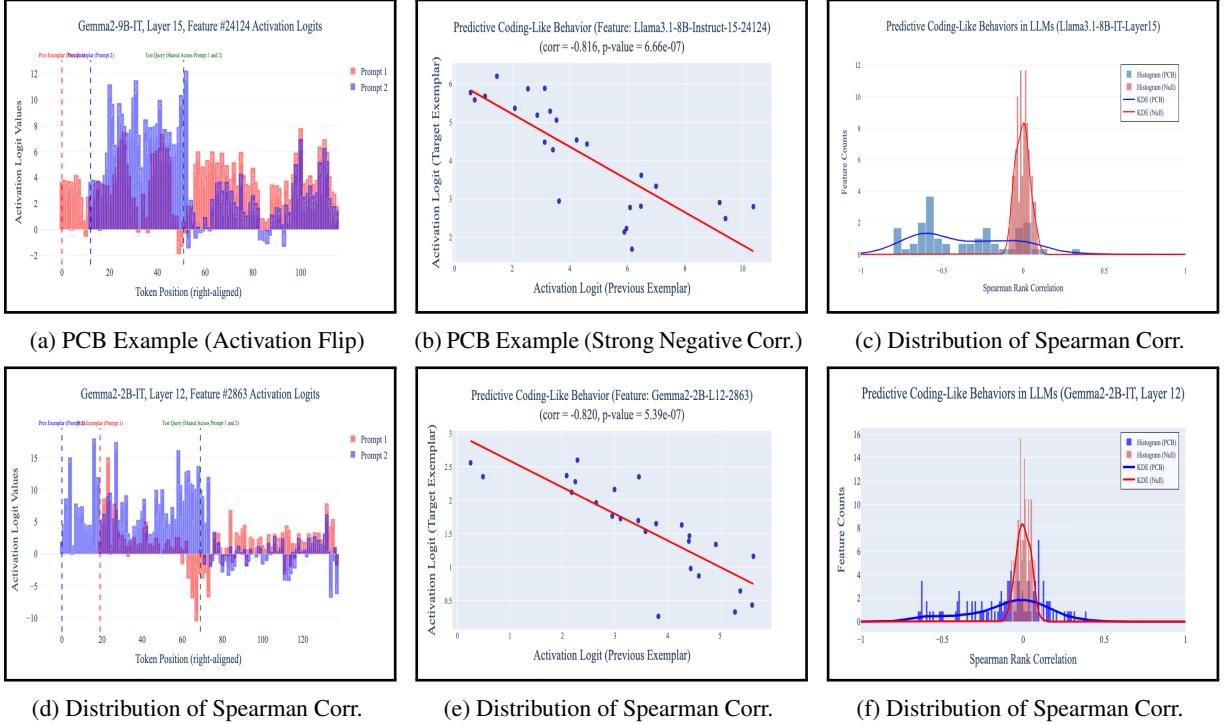
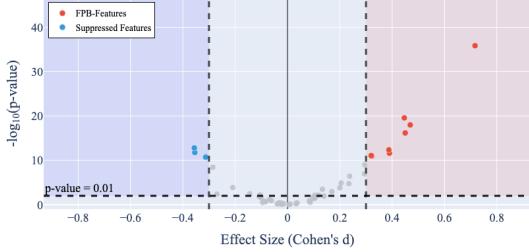


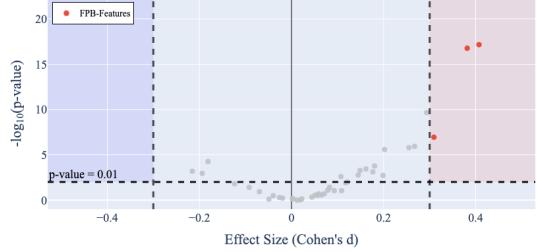
Figure 7: Predictive Coding-Like Behavior (PCB) in Llama3.1-8B-Instruct and Gemma2-2B-IT. Description analogous to that in Figure 2. We observe identical tendencies, yet a much stronger PCB behaviors in Llama3.1-8B-Instruct. These consistent results across model families and parameter scales strongly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

Functional Plasticity in LLMs (Gemma2-9B-IT, Layer 20), Label: Sports



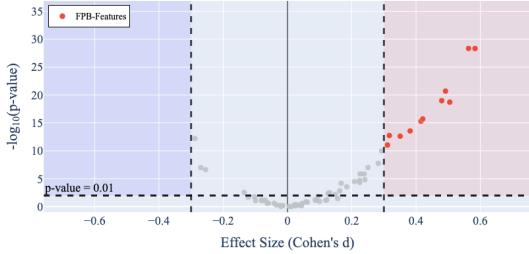
(a) Gemma2-9B-IT, L20, Removing "Sports"

Functional Plasticity in LLMs (Gemma2-9B-IT, Layer 20), Label: Business



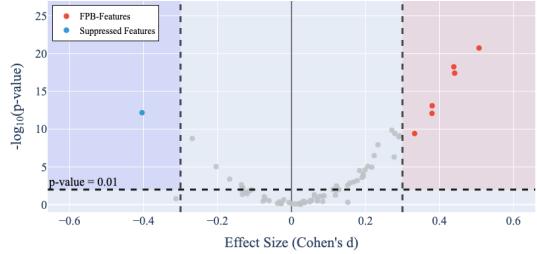
(b) Gemma2-9B-IT, L20, Removing "Business"

Functional Plasticity in LLMs (Gemma2-9B-IT, Layer 20), Label: Technology



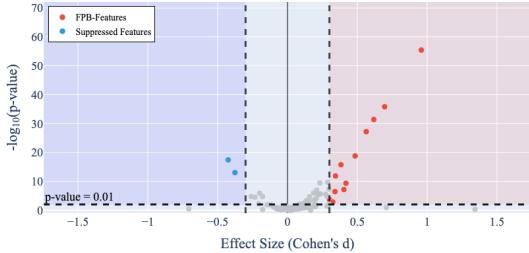
(c) Gemma2-9B-IT, L20, Removing "Technology"

Functional Plasticity in LLMs (Gemma2-9B-IT, Layer 20)



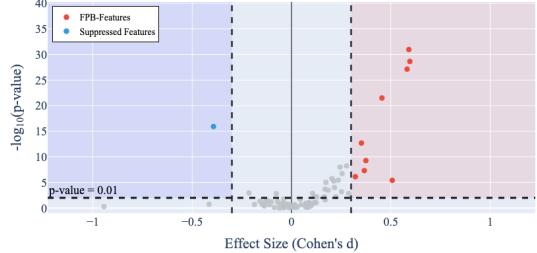
(d) Gemma2-9B-IT, L20, Removing "World"

Functional Plasticity in LLMs (Llama3.1-8B-Instruct, Layer 15), Label: Sports



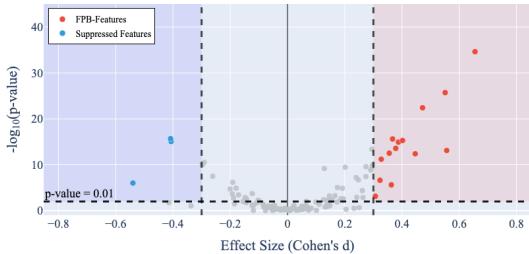
(e) Llama3.1-8B-IT, L15, Removing "Sports"

Functional Plasticity in LLMs (Llama3.1-8B-Instruct, Layer 15), Label: Business



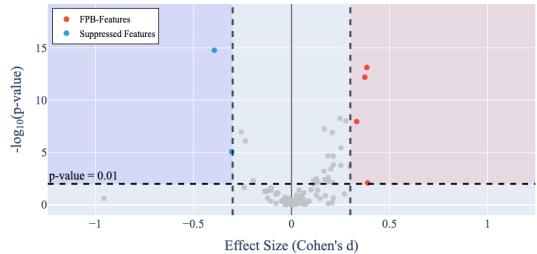
(f) Llama3.1-8B-Instruct, L15, Removing "Business"

Functional Plasticity in LLMs (Llama3.1-8B-Instruct, Layer 15), Label: Technology



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

Functional Plasticity in LLMs (Llama3.1-8B-Instruct, Layer 15), Label: World



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

Figure 8: a

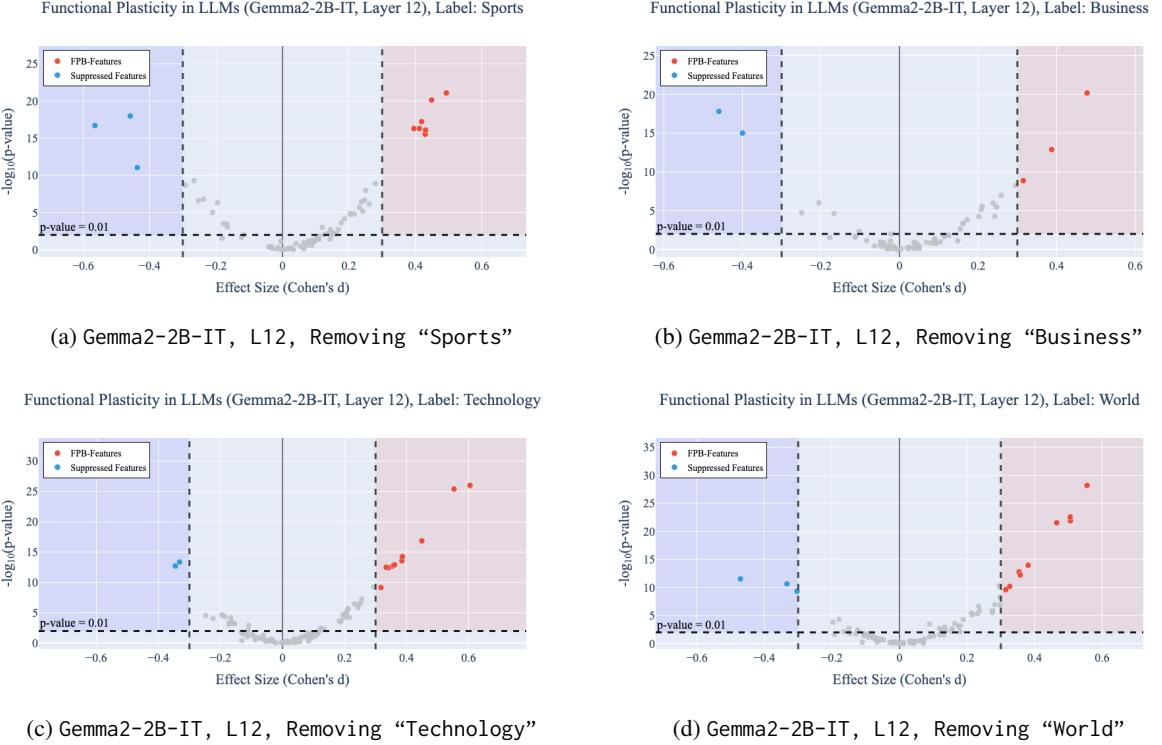


Figure 9: a

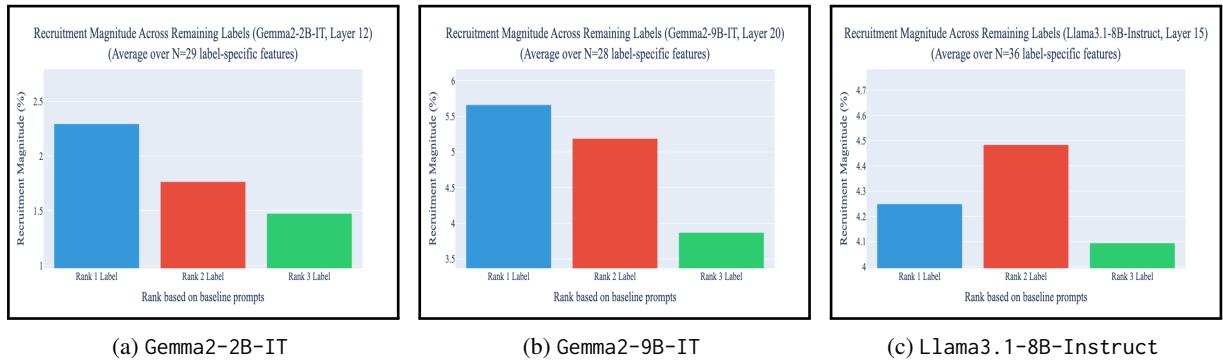


Figure 10: **Directed Migration of Functionality in FPB- Features.** When label-specific FPB-features encounter the absence of their primary trigger, they demonstrate a functional reallocation toward the remaining labels. Critically, this migration is not stochastic; instead, features preferentially 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.

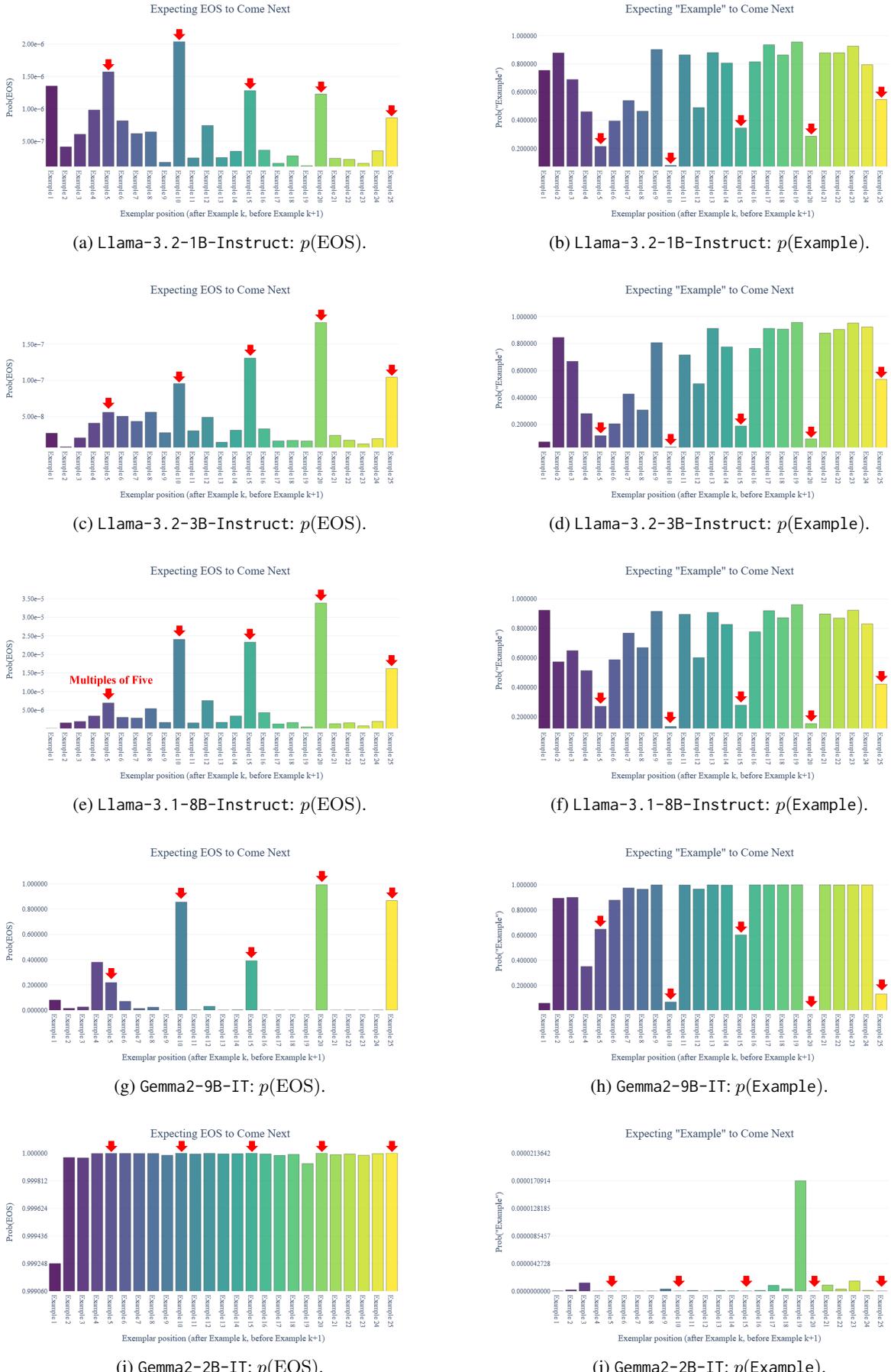


Figure 11: Extension of Figure 12 to five models under the same setup. Most models show peaks at multiples of five or ten, whereas Gemma2-2B-IT is comparatively uniform.

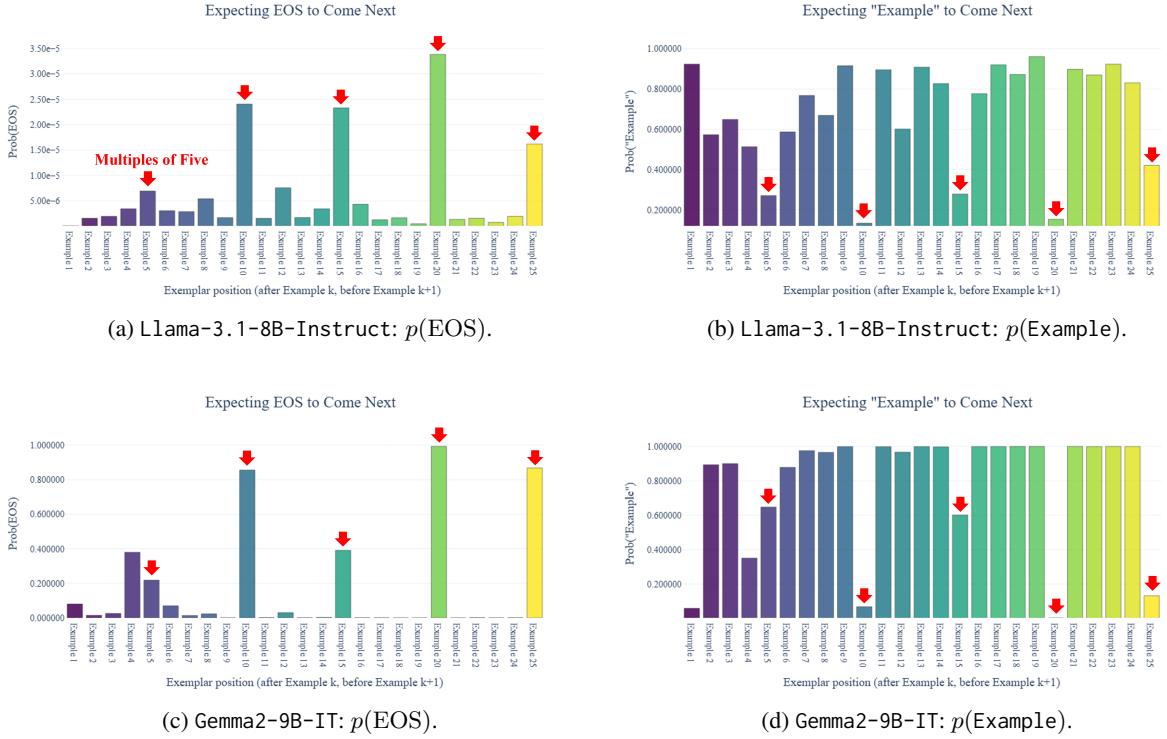


Figure 12: Decoder Logit Lens analysis of implicit expectations over the number of few-shot exemplars, averaged over 1,000 randomly constructed prompts. Results are shown for Llama-3.1-8B-Instruct and Gemma2-9B-IT. Probes are taken at each exemplar boundary (after Example k , before Example $k+1$) using 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 $\langle \text{eot_id} \rangle$ or $\langle \text{end_of_text} \rangle$, whereas for Gemma it corresponds to $\langle \text{eos} \rangle$ or $\langle \text{end_of_turn} \rangle$. 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.