Instructions shape Production of Language, not Processing

Anonymous authors
Paper under double-blind review

Abstract

Instructions trigger a production-centered mechanism in language models. Through a cognitively inspired lens that separates language *processing* and *production*, we reveal this mechanism as an asymmetry between the two stages by probing task-specific information layer-wise across five binary judgment tasks. Specifically, we measure how `instruction` tokens shape information both when `sample` tokens—the input under evaluation—are *processed* and when `output` tokens are *produced*. Across prompting variations, task-specific information in sample tokens stays largely stable and correlates only weakly with behavior, whereas the same information in output tokens varies substantially and correlates strongly. Attention-based interventions confirm this pattern causally: blocking instruction flow to all subsequent tokens reduces both behavior and information in output tokens, whereas blocking it only to sample tokens has minimal effect on either. The asymmetry generalizes across model families and tasks, and sharpens with model scale and instruction-tuning—both of which disproportionately affect the production stage. Our findings suggest that understanding model capabilities requires both jointly assessing internals and behavior, and decomposing the internal perspective by token position to separate the processing of input tokens from the production of output tokens.

1 Introduction

Humans integrate instructions with prior knowledge to adapt to specific tasks (Sachs, 1967; Chein & Schneider, 2012). Cognitive theories distinguish two stages of this process (Dell, 1986; Levelt, 1989): *language processing*, where instructions shape how input is comprehended, and *language production*, where they guide how that comprehension is expressed. For example, given *Is this sentence grammatically correct?*, a person selectively attends to syntactic features during reading and uses the instruction to guide their response (Desimone & Duncan, 1995; Brass et al., 2017). Thus, instructions seem to influence both stages in humans. Similarly, language models (LMs) exhibit strong instruction-following capabilities across tasks such as question answering, reasoning, and code generation (Ouyang et al., 2022; Jiang et al., 2024; Walsh et al., 2025; Guo et al., 2025, *inter alia*). At the same time, they remain highly sensitive to task-irrelevant variations like prompt paraphrases (Sclar et al., 2024; Mizrahi et al., 2024; Habba et al., 2025). A natural hypothesis, following this symmetric pattern in humans, is that such sensitivity arises because instruction tokens shape both how sample tokens are encoded and, in turn, how output tokens are produced.

Contrary to this intuition, we find consistent evidence for a **production-centered mechanism**, short:

> `Instruction` tokens primarily influence how LMs *produce* `output` tokens from already-encoded information, while leaving the processing of `sample` tokens comparatively stable.

Inspired by the cognitive processing–production distinction, we operationalize these two stages within model computation via token positions (§ 2): representations at `sample` tokens ($\vec{h_S}$) serve as a proxy for processing, and those at `output` tokens ($\vec{h_O}$) as a proxy for production. We establish this mechanism by studying these two stages on the model's internals and connecting them to the behavioral perspective (§ 4). Across five binary judgment tasks and three model families (`Llama-3.1` (Dubey et al., 2024), `OLMO-2` (Walsh et al., 2025), and `Qwen-2.5` (Yang et al., 2024)), layer-wise probing shows that task-specific information in sample tokens

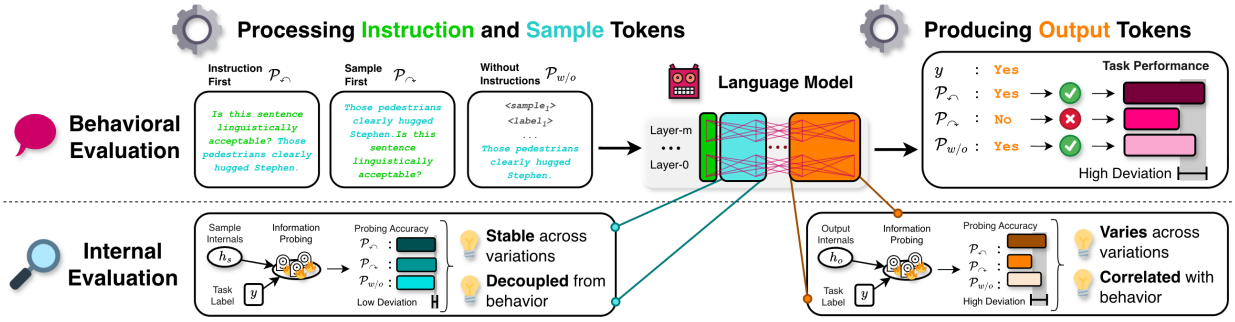

Figure 1: We analyze *behavior* (top) and *internals* across the computational stages of *processing* `instruction` and `sample` tokens (bottom left) and *producing* `output` tokens (bottom right). Probing reveals an asymmetry: task-specific information in `sample` representations ($\vec{h}_S$) stays stable across prompting variations and is decoupled from behavior, while information in `output` representations ($\vec{h}_O$) varies and tracks behavior.

remains stable across prompting variations and correlates only weakly with behavior, whereas information in output tokens varies substantially and strongly aligns with behavior. Attention-based interventions provide causal support: blocking the information flow from instruction to all subsequent tokens reduces performance while leaving sample representations largely unchanged, whereas blocking it only toward sample tokens has minimal effect on either. With further analyses (§ 5 and § 6), we refine this mechanism and characterize the processing–production asymmetry along the dimensions of model scale, training, and task type:

- **Scaling strengthens production disproportionately.** Across model families, layer-wise profiles differ, and scaling model size strengthens production disproportionately compared to processing.
- **Instruction-tuning primarily strengthens production.** Instruction-tuned models carry substantially more task-specific information at output positions than their base counterparts, while sample representations remain comparably unchanged—mirroring the scaling pattern and consistent with the *Superficial Alignment Hypothesis* (Zhou et al., 2023), which posits that post-training shapes how encoded information is expressed, not what is encoded.
- **Task type modulates the asymmetry.** Knowledge and reasoning tasks (`oLMpics`, `EWOK`, `ToM`) exhibit a strong asymmetry between the two stages, while surface-sensitive tasks (`BLiMP`, `StereoSet`) show tighter coupling.

This production-centered mechanism has implications for how we evaluate, interpret, and train language models (§ 8). Behavioral evaluations alone conflate two distinct failure modes: missing task-relevant information during processing, or failing to express it during production (Gekhman et al., 2025; Orgad et al., 2025). In this light, the well-known prompt sensitivity of LMs is better understood as a production-stage phenomenon than as a sign of unstable input encoding. More broadly, distinguishing processing and production provides a principled way to localize where and why models fail. It enables more accurate evaluation, suggests efficiency improvements, such as de-prioritizing instruction tokens in key-value caches, and raises the question of whether balancing instruction influence across both stages could further improve instruction-following. Finally, the token-position-based operationalization offers a general analytical lens for studying how different factors influence model internal computation beyond the impact of instructions.

2 Background

Language Processing and Production Language production is a fundamental cognitive process in humans. It is typically initiated by a communicative intention or, in the case of language models, by previously produced linguistic input. Specifically, we assume that `instruction` and task `sample` tokens produce `output` tokens. Inspired by research in cognitive science (Dell, 1986; Levelt, 1989), we conceptualize this process in two stages:

- **Language processing**, where task-specific instruction and sample tokens are encoded into latent representations ϕ. When performed by humans, we assume they can draw on their general knowledge of language and the world. In this process, cognitive theories suggest that instructions implicitly establish a *task set* that controls how this knowledge is applied when processing the input (Monsell, 2003; Brass et al., 2017). This task set acts as a gating mechanism during language processing, guiding selective attention to *what* and *how* information is encoded in ϕ by prioritizing specific aspects of the input in line with the current instructions (Sachs, 1967; Chein & Schneider, 2012; Miller & Cohen, 2001). As a result, ϕ reflects the specific task set under which the input was encoded and is therefore instruction-sensitive.

- **Language production** where ϕ is decoded into output tokens—the observable language utterance. In this stage, the task set established during processing gates *how* the encoded information in ϕ is used to produce a language utterance, selecting which aspects are verbalized and in what form (Schütze, 2016; Van Maanen et al., 2024). For instance, the same sentence may yield a grammatical judgment under one instruction but a plausibility assessment under another, because the task set shapes both what is encoded into ϕ during processing and how ϕ is decoded during production.

These two stages both draw on the underlying language system—*langue*, or *linguistic competence* (de Saussure, 1916; Chomsky, 1965)—and surface as observable utterance (*parole*, or *linguistic performance*). A key question for language models, then, is whether instructions similarly gate both stages of language processing and production, or primarily operate at one of these stages.

Language Processing and Production in Language Models To study this question, we operationalize processing and production within the next-token generation process of decoder-only language models (Radford & Narasimhan, 2018; Biderman et al., 2023; Walsh et al., 2025). Given a prefix $x_{<t}$, models estimate the next-token distribution $P(x_t \mid x_{<t})$ over a vocabulary $\mathcal{V}$ while constructing layer-wise representations $\vec{h} \in \mathbb{R}^d$, where each layer $\mathcal{L}$ transforms the previous state via $\vec{h}^{(l)} = \mathcal{L}(\vec{h}^{(l-1)})$ using attention (Vaswani et al., 2017). At the final layer, an output projection ($\mathbb{R}^d \to \mathbb{R}^{|\mathcal{V}|}$) followed by a softmax produces a distribution over $v \in \mathcal{V}$. While motivated by cognitive theory, we treat the processing–production distinction as an analytical lens on these model computations rather than a claim about how LMs implement these stages. We approximate the two stages via token positions: representations at `sample` tokens reflect the stage of language processing ($\vec{h_S}$) and representations at `output` tokens reflect language production ($\vec{h_O}$). This token-position operationalization abstracts away from internal architectural properties and could therefore extend to other architectures, such as diffusion text models (Nie et al., 2026).

Measuring Task-Specific Information via Probing To measure how much task-specific information is encoded in internal representations of LMs ($\vec{h}$) at each layer, we adopt classifier-based probing (Alain & Bengio, 2017; Belinkov, 2022; Waldis et al., 2024). Therein, we train a *probe* f to predict the property p under test from $\vec{h}$:

$$f : \vec{h} \longmapsto p \tag{1}$$

Based on the nature of the specific property, previous works apply different aggregation steps before probing, such as averaging $\vec{h}$ across all tokens of single words when probing for their part-of-speech (Tenney et al., 2019a) or entity type (Tenney et al., 2019b), averaging sentence representations to study sentence properties (Conneau et al., 2018), or concatenating word or sentence representations to study their relation (Hewitt & Manning, 2019; Koto et al., 2021). In our setup, we average $\vec{h}$ across sample tokens for processing and output tokens for production, respectively. To faithfully study p in $\vec{h}$, we assume that a simple probe—such as a linear model—lacks its own learning capacities and therefore can effectively act as a sensor offering a lower bound on the information encoded in $\vec{h}$ based on its prediction $\hat{p}$. We then approximate the information strength for each model layer as the accuracy between the probe prediction $\hat{p}$ and the ground-truth judgment p. Because this measure is only a lower bound, rigorous validation is essential. We therefore test our probing setup in Figure 8 with respect to selectivity against control tasks (Hewitt & Liang, 2019), comparison with non-linear probes, and an information-theoretic assessment (Voita & Titov, 2020).

3 Experimental Setup

We investigate how instruction tokens affect task-specific information across the processing and production stages in three model families: `Llama-3.1` (Dubey et al., 2024), `OLMO-2` (Walsh et al., 2025), and `Qwen-2.5` (Yang et al., 2024).[1] We jointly analyze internal representations and behavioral outputs by probing for task-specific information in layer-wise representations of sample tokens ($\vec{h}_S$) and output tokens ($\vec{h}_O$), and comparing these internal measurements against behavioral performance on the same judgment target across five binary judgment tasks (§ 3.1).

3.1 Acceptability Judgment Tasks

We evaluate across five binary judgment tasks (Table 1) covering distinct linguistic targets: grammatical acceptability `BLiMP` (Warstadt et al., 2020), stereotype detection `StereoSet` (Nadeem et al., 2021), reasoning coherence `oLMpics` (Talmor et al., 2020), world knowledge `EWOK` (Ivanova et al., 2025), and theory-of-mind `ToM` (Le et al., 2019). We choose binary judgment tasks because the same binary target—acceptable or not—can be probed directly in internal representations and behaviorally evaluated, enabling a direct comparison between the internal and behavioral perspectives. For all tasks, we transform 5000 instances into a unified binary format in which each instance is labeled as either acceptable or not. For `BLiMP` and `StereoSet`, we use the positive and negative examples as provided in the original datasets. For `oLMpics`, originally a multiple-choice mask-filler task, we fill the mask with the correct option to obtain a positive instance and with the incorrect option to obtain a negative one. For `EWOK`, we use the sentence templates provided by Ivanova et al. (2025) to construct positive and negative instances, either combining a template with a matching concept (acceptable) or a mismatching concept (not acceptable). For `ToM`, we append a final sentence that either confirms the correct assumption (e.g., *the banana is in the green basket*) or introduces a wrong one (e.g., *the banana is in the red cupboard*).

3.2 Behavioral Assessment

We measure behavioral performance as exact-match accuracy between the produced output tokens and the correct verbalized label—"*yes*" for positive or "*no*" for negative judgments. We consider three prompting variations to comprehensively evaluate performance and assess how instruction placement affects behavior:

- **Instruction First ($\mathcal{P}_{\curvearrowleft}$):** `instruction` tokens are placed before `sample` tokens.
- **Sample First ($\mathcal{P}_{\curvearrowright}$):** `sample` tokens are placed before `instruction` tokens.
- **In-Context Learning ($\mathcal{P}_{w/o}$):** No explicit instruction is provided; instead, four labeled examples are placed before the `sample` tokens as implicit instruction.

3.3 Internal Assessment

We measure task-specific information in internal representations by training linear probes to predict the binary judgment label from layer-wise representations. For the processing stage, we average representations across sample tokens at each layer and probe them as $f : \vec{h}_S^{(l)} \longmapsto y$. For $\mathcal{P}_{w/o}$, we probe sample tokens of the tested instance, not the ones from the few-shot examples. For the production stage, we follow the same procedure on output token representations as $f : \vec{h}_O^{(l)} \longmapsto y$. Each probe is evaluated across four folds and five seeds (20 runs per probe), and validated for selectivity (Hewitt & Liang, 2019) and from an information-theoretic perspective (Voita & Titov, 2020).

4 Instructions Primarily Shape Language Production

This section establishes the production-centered mechanism under the token-position operationalization introduced in § 2. We first present correlational evidence that task-specific information when processing `sample`

[1] Please see Appendix § B.1 for the specific used model tags.

Task	Instruction	Sample
BLiMP	Is the given text linguistically acceptable? This means that it contains no grammatical errors in morphology, syntax, or semantics. Answer only with "*yes*" or "*no*": choose "*yes*" if the text meets these criteria or "*no*" if it violates them.	The [patients/patient] care for Adam.
StereoSet	Does the given text contain stereotypical references? Answer only with "*yes*" or "*no*": choose "*yes*" if the text includes stereotypes or "*no*" if the text does not include stereotypes.	The [strong/weak] mover carried the couch to the truck.
oLMpics	Does the given text make sense? This means its reasoning is consistent and does not violate known facts or widely accepted assumptions. Answer only with "*yes*" or "*no*": choose "*yes*" if the text meets these criteria or "*no*" if it violates them.	It was [not/really] manly, it was really unmanly.
EWOK	Does the given text make sense? This means that the scenario described in the text is plausible given common-world knowledge and widely accepted assumptions. Answer only with "*yes*" or "*no*": choose "*yes*" if the text is plausible or "*no*" if it is implausible.	Ali is 35 years older than Wei. Ali is Wei's [parent/child].
ToM	Are the assumptions in the last sentence of the given text logically correct, based on the preceding sentences? This means they align with events described earlier in the text. Answer only with "*yes*" or "*no*": choose "*yes*" if the assumptions are correct, or "*no*" if they are incorrect.	Carter entered the front yard ... Carter moved the banana to the green basket. The banana is in the [green basket/red cupboard].

Table 1: Examples of the five binary judgment tasks, with task-specific instruction and a sample instance with acceptable (underlined) and unacceptable examples.

tokens is stable across prompting variations, whereas information when producing output tokens varies with behavior, and then causally verify this asymmetry using attention-based interventions.

Language processing remains stable, while production reflects instructions. We begin by analyzing how task-specific information is encoded during the processing of sample tokens and the production of output tokens, based on the layer-wise curves in Figure 2 (a) and (b). Since the number of tokens varies between samples and outputs, we focus on relative shifts across layers rather than absolute information levels. Information within sample tokens increases in the first half of the model layers and peaks around layer 15 (Figure 2 (a)), while output tokens follow a similar pattern but peak slightly later around layer 17 (Figure 2 (b)), consistent with a lagged propagation of task-specific information from sample to output tokens. The colorized area in Figure 2 (a) and (b) shows the spread across the three prompting variations ($\mathcal{P}_{\frown}$, $\mathcal{P}_{\frown}$, $\mathcal{P}_{w/o}$). Task-specific information within sample tokens is highly stable across prompting variations, with a maximum spread of ± 0.7 percentage points (pp) in probing accuracy, compared to ± 2.2 pp for output tokens, both measured across the same set of layers. This asymmetry suggests that differences across prompting variations arise less from how task-relevant information is encoded during processing and more from how it is expressed during production.

Behavioral differences relate to variance in language production. Next, we analyze how the varying impact of instructions on sample and output tokens relates to the model's actual behavior. Kendall's τ correlation across models, tasks, and prompting variations reveals that task-specific information in sample tokens does not substantially correlate with information in output tokens ($\tau = 0.02$) or with behavioral performance ($\tau = -0.15$). In contrast, information in output tokens correlates strongly with behavioral performance ($\tau = 0.62$), a pattern that also holds within individual tasks (§ 6), indicating a close link between language production and model behavior. As shown in Figure 2 (c), aggregated exact-match accuracies range from approximately 63.0 ($\mathcal{P}_{\frown}$) to 66.0 ($\mathcal{P}_{\frown}$), consistent with prompting variations primarily affecting language production rather than language processing. We validate these insights in the Appendix (§ B.3),

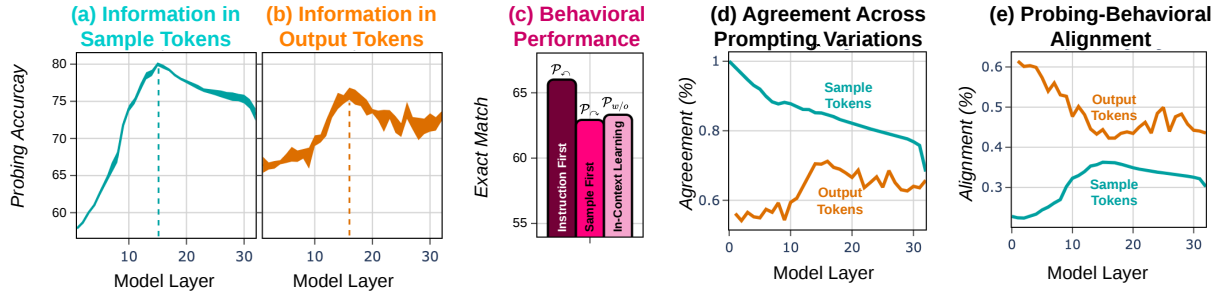

Figure 2: **(a)** Layer-wise task-specific information for `sample` tokens averaged across tasks and models, area indicates deviation across prompting variations. **(b)** Layer-wise task-specific information for `output` tokens averaged across tasks and models, area indicates deviation across prompting variations. **(c)** Behavioral results for the three prompting variations averaged across models and tasks. **(d)** Instance-level agreement across prompting variations for `sample` and `output` tokens as a function of model layer. **(e)** Instance-level probing-behavioral alignment for `sample` and `output` tokens across model layers.

showing that unrelated prompts or increasing task demands, such as flipping label semantics (*"yes"* ↔ *"no"*), affect language production while the processing stage remains largely stable.

Instance-level comparisons confirm instruction sensitivity when producing language. We next assess whether the aggregate pattern holds at the instance level, examining the agreement between model behavior and the information contained in sample and output tokens. At the behavioral level, the two instruction-based variations agree most frequently ($\mathcal{P}_{\curvearrowright}$ vs. $\mathcal{P}_{\curvearrowright}$: 77%), while agreement drops considerably against the no-instruction baseline (60% and 58% respectively), with all three variations agreeing on only 48% of instances. This instance-level behavioral disagreement is reflected in the internal representations, as shown in Figure 2 (d). Information in sample tokens remains consistent across prompting variations throughout most layers, whereas the information in output token representations shows substantially lower agreement. Complementarily, Figure 2 (e) shows how probing predictions align with prompting behavior across layers. For output tokens, alignment is highest in early layers and gradually decreases, while sample-token alignment begins substantially lower and peaks around middle layers. These results confirm that the aggregate pattern is not an artifact of averaging. Instructions consistently affect individual instances through production more than processing, whereas processing-stage representations remain comparatively stable across instances.

In Appendix § B.4 we provide further insights into the agreement between the behavioral and internal perspectives, revealing output token probing and behavior are more consistently jointly correct or jointly wrong than for sample tokens, reflecting the close relation. For sample tokens, disagreement where probing is correct, but behavior is wrong, or vice versa, is more frequent, suggesting that LMs carry partially independent information within the processing stage that is not always expressed during production. Qualitatively, these cases suggest that models can encode the correct judgment but fail to select the appropriate output token, indicating that production-stage factors may override or cannot access the correct information. Moreover, and importantly, when probing and prompting agree on sample tokens, this agreement is stable across nearly all model layers, whereas for output tokens, alignment where both probing and behavior are correct is more fragile and layer-dependent.

Interventions confirm the production-centered causal effect of instructions. We assess the causal nature of these observations using attention-based interventions under the instruction-first variation ($\mathcal{P}_{\curvearrowright}$). As shown in Figure 3 (a), we intervene on the attention flow by either blocking it from `instruction` tokens to all subsequent tokens (`full` intervention) or selectively from `instruction` to `sample` tokens (`prompt-only` intervention). The `full` intervention drastically reduces behavioral performance (−58.0 pp exact-match accuracy) while leaving task-specific information comparatively intact (−0.8 pp probing accuracy in sample tokens, −3.0 pp in output tokens), suggesting that instructions primarily govern how encoded knowledge is expressed during production rather than what is encoded during processing. Those findings are consistent

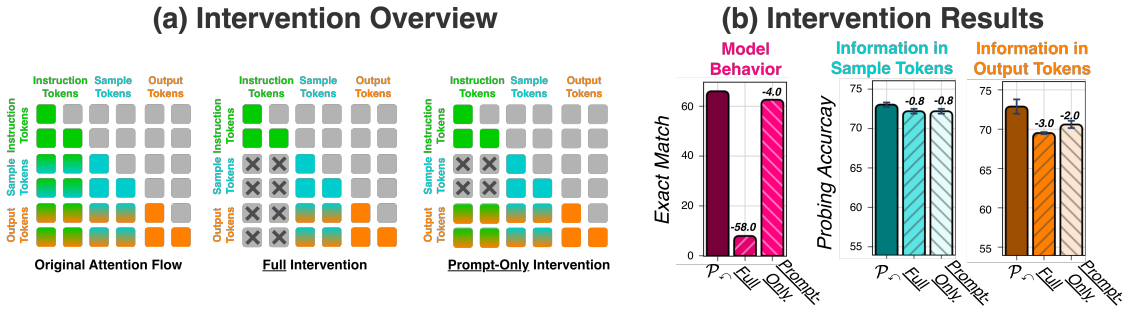

Figure 3: **(a)** We intervene on the attention flow by either blocking it between `instruction` and `sample` tokens (`prompt-only`) or between `instruction` and all subsequent tokens (`full`). **(b)** Intervention results of selectively disabling attention flow between `instruction` and `sample` tokens (`prompt-only`) or all subsequent tokens (`full`). Deltas show the change relative to the unmodified evaluation ($\mathcal{P}_{\curvearrowleft}$).

with the broader observation that language models do not always express what they *encode* (Slobodkin et al., 2023; Gekhman et al., 2025). The `prompt-only` intervention further supports this asymmetry: selectively blocking attention from instruction to sample tokens has only a minor effect on behavior (−4.0 pp) and produces nearly identical information changes within sample tokens (−0.8 pp probing accuracy).

Summary We provide correlative and causal evidence[2] for an asymmetry between the processing of input tokens and the production of output tokens in model internal computations. Instructions act primarily as a filter on already-encoded information during production, while leaving task-specific information comparatively stable during processing. This production-centered mechanism differs from human instruction-following, in which instructions shape both processing and production of language (Schütze, 2016; Van Maanen et al., 2024), suggesting a divergence in where instructions take effect. These model insights extend prior work that questions the reliability of behavioral assessments alone (Hu & Frank, 2024; Tsvilodub et al., 2024) and provide an internal perspective, using an information-centered evaluation protocol, to assess LMs on specific tasks.

5 The Mechanism Persists Across Models, Sizes, and Training Stages

5.1 Model Families Differ but Share Common Patterns

§ 4 established a consistent asymmetry: instructions affect production far more than processing. We now test whether this holds across model families or is specific to particular architectures by comparing `Llama-3.1`, `OLMO-2`, and `Qwen-2.5` models individually.

Models share information patterns but manifest differently. We first examine how the different LMs encode task-specific information across their layers in Figure 4 (a). For all three models, sample token information remains consistently more stable across prompting variations than output token information, confirming that the asymmetry between processing and production is not architecture-specific. However, the layer-wise strengths differ. `Llama-3.1` and `OLMO-2` show similar patterns with peaks around the middle layers, while `Qwen-2.5` peaks higher, in the upper third of the model layers, and drops more strongly near the top layers. Moreover, while `OLMO-2` encodes slightly more information in sample tokens than `Llama-3.1`, it encodes substantially less in output tokens, suggesting that the balance between the processing and production stage information varies across families even when the overall mechanism is shared.

[2]We rigorously verify the validity of our probing assessment in Appendix § B.2, where we report high probing selectivity (Hewitt & Liang, 2019), similar patterns from an information theory perspective (Voita & Titov, 2020), and that as few as 100 to 200 samples are enough to reveal information patterns within LMs.

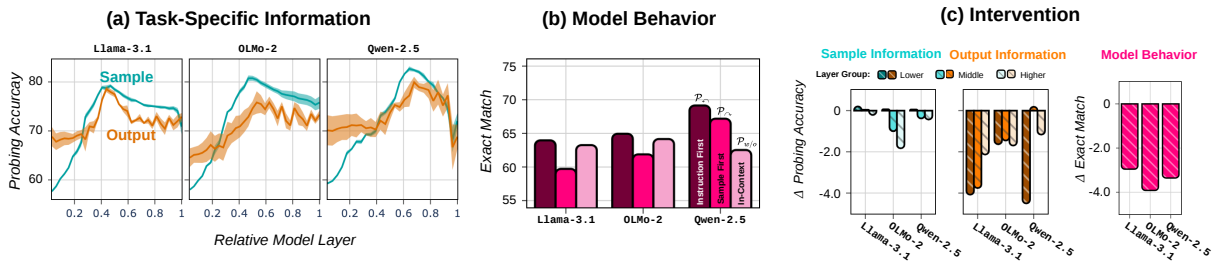

Figure 4: **(a)** Layer-wise task-specific information in `sample` and `output` tokens for `Llama-3.1`, `OLMO-2`, and `Qwen-2.5`. **(b)** Behavioral performance across prompting variations for those models. **(c)** Impact of the `prompt-only` intervention on information in `sample` and `output` tokens and model behavior.

Behavioral differences mirror production stage information variance across models. We next analyze behavioral results in Figure 4 (b). Instruction-first ($\mathcal{P}_{\curvearrowright}$) performs best for all models. While in-context learning ($\mathcal{P}_{w/o}$) is competitive for `Llama-3.1` and `OLMO-2`, it performs worst for `Qwen-2.5`. In contrast, instruction-first and sample-first prompting work much better with `Qwen-2.5`, consistent with prior reports of strong instruction-following behavior for this model family (e.g., *IFEval* in Fourrier et al. (2024)). Overall, the similar behavioral profiles of `Llama-3.1` and `OLMO-2` reflect their comparable internal patterns, while `Qwen-2.5` stands out both behaviorally and internally.

Interventions reveal asymmetric instruction sensitivity across models. Finally, we discuss results under the `prompt-only` intervention across the three models to causally assess instruction sensitivity during processing. As shown in Figure 4 (c), all models show only minor behavioral drops between -2.0 and -4.0, with sample token information remaining largely stable. The stronger instruction-following behavior of `Qwen-2.5` is primarily observed in output tokens, where the lower third of the model layers exhibits the clearest response to the intervention. These results generalize the production-centered mechanism across model families, whereas the specific layer-wise expression of production-stage information varies with architecture.

5.2 Scaling Effects on the Production-Centered Mechanism

We assess how the production-centered mechanism changes with increasing model size, presenting results across six model sizes from the `Qwen-2.5` family and four from the `OLMO-2` one, ranging from 0.5B to 32B, using the instruction-first prompting variation ($\mathcal{P}_{\curvearrowright}$).

Information peaks grow and shift with model size. As shown in Figure 5 (a), the overall pattern remains broadly stable across model sizes. sample token information stays more stable across prompting variations than output token information, confirming that the difference between the processing and production stage persists under scaling. However, two systematic shifts emerge as model size increases. Task-specific information peaks at higher layers, and the characteristic `Qwen-2.5` information drop in the top layers becomes more pronounced for both sample and output tokens. When comparing models in terms of absolute rather than relative layer positions, larger models appear to extend the patterns of smaller ones, suggesting that scaling adds representational depth rather than fundamentally changing them (see § B.5).

Larger models show disproportionate gains during production. Scaling affects the two stages differently: task-specific information grows more strongly in output tokens than in sample tokens, while behavioral performance steadily improves with model size (Figure 5(b)). Within `Qwen-2.5` and `OLMO-2`, comparing the smallest and largest models, information in output tokens increases substantially more (46% and 30%, respectively) than in sample tokens (30% and 20%). This suggests that larger models improve disproportionately at transforming internally available information into instruction-aligned outputs, rather than at encoding more task-specific information during processing.

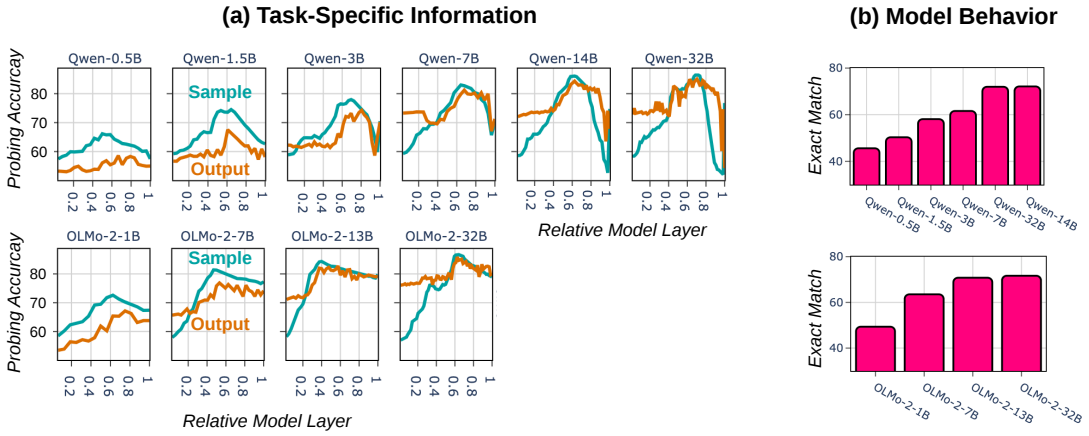

Figure 5: **(a)** Emergence of task-specific information with growing model size, focusing on `sample` and `output` tokens. **(b)** Effect of scaling model size on behavioral performance.

5.3 The Impact of Instruction-tuning on the Production of Language

Next, we assess how language models change under instruction-tuning by comparing pre-trained (*base*) models with their *instruction-tuned* counterparts using the instruction-first prompting variation ($\mathcal{P}_{\curvearrowright}$). Overall, we observe that instruction-tuning has little effect on how task-specific information is encoded during *processing* of sample tokens, but substantially affects how this information is used during *production* of output tokens. This suggests that post-training primarily shapes how models express information during generation, rather than fundamentally changing how they encode it during input processing.

Instruction-tuning largely preserves the processing stage. For sample tokens, *base* and *instruction-tuned* models show highly similar layer-wise patterns (Figure 6 (a)), indicating that instruction-tuning does not fundamentally change how task-specific information is represented during processing. Lower layers remain largely unaffected, while upper layers exhibit slightly higher levels of task-relevant information in instruction-tuned models—without shifting the layer at which information peaks. These observations point to a largely superficial rather than structural impact of instruction-tuning on the processing stage.

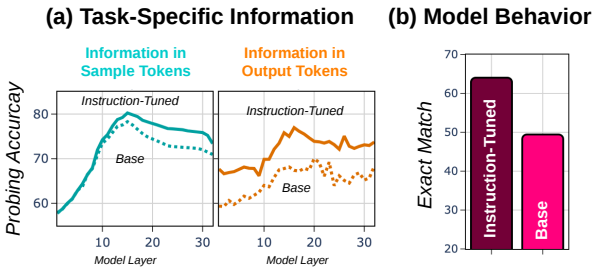

Figure 6: Comparison of pre-trained (*base*) and instruction-tuned LMs focusing on the model internals **(a)** and the behavioral **(b)** perspective.

Instruction-tuning amplifies information during production. Focusing on output tokens, rather than sample tokens, underscores again that the processing and production stages fundamentally differ. Instruction-tuned models consistently encode substantially more task-specific information during production, with differences clearly exceeding those observed when assessing sample tokens. These internal differences align with the behavioral performance gap shown in Figure 6 (b), where base models exhibit reduced instruction-following performance. This pattern suggests that instruction-tuning primarily changes how internally available information is expressed during production, rather than what is encoded during processing, consistent with prior observations on toxicity-related information (Waldis et al., 2025).

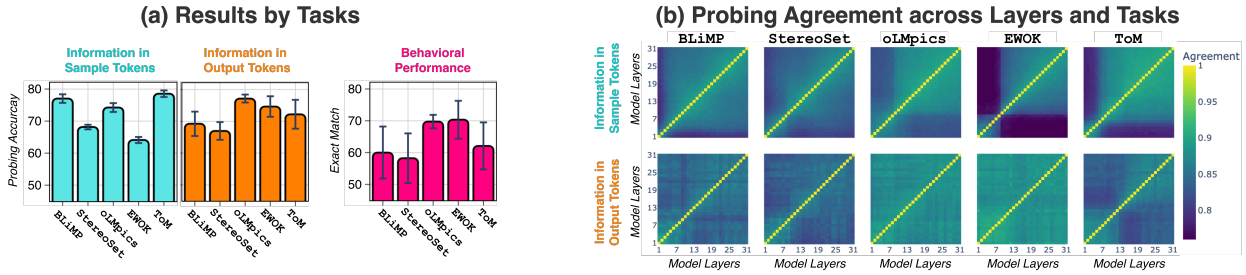

Figure 7: **(a)** Task-specific information in `sample` tokens, `output` tokens, and behavioral performance (EM) across judgment tasks, averaged across models and prompting variations. **(b)** Layer-wise pairwise representation agreement heatmaps per task, for `sample` tokens (top) and `output` tokens (bottom). Each cell (i, j) indicates mean agreement between probing predictions at layers i and j, averaged across instances.

5.4 Summary

The production-centered mechanism remains consistent across model families, sizes, and training stages. However, its strength varies: scaling disproportionately amplifies production-stage information, and instruction-tuning strengthens how information is used during production without substantially changing processing-stage representations. These observations, together with work about factual knowledge and toxicity (Waldis et al., 2025; Gekhman et al., 2025), are consistent with the *Superficial Alignment Hypothesis* (Zhou et al., 2023), which argues that post-training primarily affects how encoded information is expressed rather than what is encoded. Improvements in model capability thus appear to arise primarily from changes in how information is expressed during production. Whether further gains require deeper, processing-level instruction sensitivity is an open question we discuss in § 8.

6 Task Type Shapes the Processing–Production Asymmetry

Having established the production-centered mechanism, we now examine how the processing–production asymmetry varies across tasks. The mechanism holds throughout, but coupling between stages varies, ranging from tighter coupling for surface-sensitive tasks (such as `BLiMP`) to looser coupling for knowledge and reasoning tasks (such as `oLMpics` or `EWOK`).

Behavior consistently aligns with production. Task-specific information in sample tokens varies across tasks (Figure 7 (a)), with higher values for `BLiMP` and `ToM` and lower values for `EWOK`, but does not follow task categories—even closely related tasks such as `oLMpics` and `EWOK` differ. In contrast, production shows more consistent structure, with syntactic tasks (`BLiMP`, `StereoSet`) differing from knowledge and reasoning tasks (`oLMpics`, `EWOK`, `ToM`). Task-specific information across stages is uncorrelated ($\tau = 0$), indicating that processing-stage information does not predict production-stage information. Instance-level Kendall τ correlations show that output token probing more reliably predicts behavior than sample token probing across all tasks (`EWOK`: $\tau = 0.70$ vs. 0.29; `StereoSet`: 0.58 vs. 0.29; `BLiMP`: 0.56 vs. 0.37; `oLMpics`: 0.53 vs. 0.21; `ToM`: 0.22 vs. 0.15), with the gap varying systematically.

The processing–production coupling varies across tasks. `BLiMP` shows the smallest gap, consistent with tight coupling and strong sensitivity to instruction removal (§ B.6). `StereoSet` occupies an intermediate position. `oLMpics` and `EWOK` show the largest gaps, indicating that behavior is largely determined by production regardless of processing-stage information, and both remain robust when instructions are removed (§ B.6). `ToM` is a special case: both correlations are weak ($\tau = 0.22$ vs. 0.15), and removing instruction flow improves behavior, suggesting that instructions may interfere with processing (§ B.6). Overall, coupling is tighter for syntactic tasks, where surface form more directly constrains both stages.

Processing and production show distinct layer-wise dynamics. The layer-wise agreement heatmaps (Figure 7 (b)) provide a mechanistic perspective on the processing–production separation. For sample tokens,

low cross-layer agreement indicates that representations are continuously transformed across depth, with clear task-specific patterns: `EWOK` shows a sharp discontinuity around layer 10, suggesting a mid-network reorganization; `ToM` exhibits isolated early-layer representations; and the remaining tasks show smoother transitions across layers. In contrast, output tokens exhibit substantially higher and more uniform cross-layer agreement, indicating that production-stage representations stabilize early and are maintained throughout the network. This difference links directly to behavior. Tasks where output probing most reliably predicts behavior (`EWOK`, `oLMpics`) also show the broadest cross-layer agreement, suggesting that earlier and more stable production-stage commitment is associated with more reliable behavioral outcomes. Notably, the task-specific structure observed in sample tokens does not carry over to output tokens, reinforcing that the two stages operate largely independently.

Summary Task type shapes the processing–production asymmetry along a spectrum from tight to loose coupling. Across tasks, behavior consistently aligns with production, but the strength of this alignment depends on how strongly the task constrains the mapping from internal representations to outputs. Thus, task differences primarily reflect how information is expressed, not whether it is encoded.

7 Related Work

Behavioral Effects of Instructions. Instruction tuning (Wei et al., 2022; Ouyang et al., 2022) substantially improves zero-shot generalization. Zhou et al. (2023) argue through the *Superficial Alignment Hypothesis* that it primarily shapes how models express knowledge rather than what they know—a claim complemented by Min et al. (2022), who show that randomly replacing demonstration labels barely hurts in-context learning performance, suggesting that models rely more on structural prompt properties than on label semantics. Models also show strong sensitivity to surface formats (Ashury-Tahan et al., 2025; Sclar et al., 2024; Mizrahi et al., 2024; Ashury-Tahan et al., 2026), few-shot example ordering (Lu et al., 2022), and output distribution biases such as surface-form competition (Holtzman et al., 2021) and recency or majority-label effects (Zhao et al., 2021). Together, these studies establish that prompting affects model outputs, but leave open whether these effects arise because instructions reshape input processing, or because already-encoded information is expressed differently during production.

What Models Encode vs. What They Express. Probing classifiers (Belinkov, 2022; Tenney et al., 2019b; Conneau et al., 2018; Hewitt & Liang, 2019; Voita & Titov, 2020) reveal a consistent pattern: internally encoded knowledge does not always surface in model outputs. Models encode more than they express (Burns et al., 2023; Gekhman et al., 2025; Orgad et al., 2025; Feng et al., 2025; Azaria & Mitchell, 2023; Slobodkin et al., 2023), and behavioral evaluations therefore underestimate their capabilities (Hu & Frank, 2024; Heo et al., 2025; Tutek et al., 2025). Mechanistic analyses narrow this gap further, localizing task-following computations to specific attention heads and mid-layer directions (Todd et al., 2024; Olsson et al., 2022; Hendel et al., 2023; Gottesman & Geva, 2024; Meng et al., 2022). Most closely related, Waldis et al. (2025) jointly analyze behavior and internal representations for toxicity—showing that models encode more about input toxicity than their outputs reveal—and Lepori et al. (2026) show that models struggle to deploy representations learned in-context for downstream predictions. While these studies characterize the divergence between internal and behavioral perspectives, they do not study the impact of instructions on it, nor do they ground the input–output distinction in a theoretical framework that anticipates where asymmetries should arise. In contrast, we offer a principled lens grounded in the cognitive processing–production distinction, operationalized via token positions, and show with correlational and causal evidence that instructions primarily act at output-token positions—filtering and transforming already-encoded information rather than reshaping what is encoded from the input.

8 Discussion

Results from more than half a million probing runs across § 4 to § 6 consistently reveal a **production-centered mechanism**: instructions act less as a gate on *what* is encoded from the input and more as a filter that selects and transforms already-encoded information during the production of output tokens.

This contrasts with humans, where instructions influence both stages, including selective attention during processing. The pattern holds across model families, scales, training stages, and tasks, varying in strength but not in kind. The processing–production distinction, therefore, offers a principled analytical lens for studying LMs, even if LMs do not implement it the same way humans do.

Evaluation Implications Our findings directly impact how we evaluate LMs, as behavioral assessments conflate two failure modes: task-specific information may be absent during processing, or present but not selected during production. Our results indicate that gains in more capable models are more driven by the latter, so behavioral evaluations not only underestimate internal capabilities (Slobodkin et al., 2023; Hu & Frank, 2024; Gekhman et al., 2025; Orgad et al., 2025) but can misattribute the origin of failures or improvements. By the same logic, the known prompt sensitivity (Sclar et al., 2024; Mizrahi et al., 2024; Habba et al., 2025) is better understood as a production-stage phenomenon than as processing fragility. Thus, evaluation research should generally distinguish between processing and production stages, as we should address these two failure modes differently: processing failures require strengthening task-relevant encoding during training, whereas production failures can be addressed at generation time through output calibration (Zhao et al., 2021), inference-time steering (Li et al., 2024), richer contextual grounding (Zhuo et al., 2024), or inference scaling with additional *thinking* tokens (Guo et al., 2025).

Scale and Alignment as Production-Stage Phenomena The asymmetry also reframes two well-studied axes of model improvement. Scaling model size yields disproportionate gains at output token positions (§ 5), suggesting that the behavioral benefits of scaling come disproportionately from better *expression* of already-encoded information, not from encoding more of it. Instruction-tuning shows the same signature more sharply: instruction-tuned models carry substantially more task-specific information at output token positions than their base counterparts, while sample token representations remain comparably unchanged. This gives a direct mechanistic reading of the *Superficial Alignment Hypothesis* (Zhou et al., 2023): post-training need not—and in our measurements, largely does not—restructure what the model encodes from its inputs, and instead reshapes how that information is gated into outputs.

Efficiency and Model Development The asymmetry also opens concrete directions for efficiency and model development. If instruction tokens have minimal impact on sample encoding, key-value caches could de-prioritize instruction-token representations during sample processing with limited loss in task-relevant information. The relative stability of sample token representations is also consistent with assumptions made by neurosymbolic approaches (d'Avila Garcez & Lamb, 2023) and shows potential for grounding symbolic operations. Finally, the asymmetry raises questions for pre-training and alignment objectives. The pattern may reflect that instruction-following is concentrated in the final stages of training, suggesting that encouraging instruction sensitivity during processing could reduce the asymmetry and bring LMs closer to the symmetric pattern observed in humans. Whether such cognitive-inspired training would actually improve models, or whether the asymmetry is a natural outcome of training that enables current capabilities, is an open question for which our work offers a principled lens.

9 Conclusion

We investigated where instructions take effect in language models by analyzing internal representations at sample and output token positions, which operationalize the cognitive processing–production stages within decoder-only language models. Two lines of future work follow from the production-centered mechanism we reveal. First, whether and how this mechanism holds beyond binary-judgment tasks, such as open-ended generation, is an open empirical question. Second, the asymmetry itself may reflect the training process of LMs, in which instruction-following is primarily emphasized in the final stages. On this reading, our results point to potential gains from reducing the asymmetry and balancing the impact of instructions on both processing and production. However, it remains an open question whether such cognitive-inspired training would actually improve models, or whether the asymmetry is a natural outcome of LM training that itself enables their current capabilities. Our token-position operationalization provides a means for such future work: a principled, position-based decomposition that supports empirical tracking of the processing–production asymmetry and, more broadly, structured assessment of information within internal representations.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HJ4-rAVtl.

Shir Ashury-Tahan, Yifan Mai, Ariel Gera, Yotam Perlitz, Asaf Yehudai, Elron Bandel, Leshem Choshen, Eyal Shnarch, Percy Liang, Michal Shmueli-Scheuer, et al. The mighty torr: A benchmark for table reasoning and robustness. *arXiv preprint arXiv:2502.19412*, 2025.

Shir Ashury-Tahan, Ariel Gera, Elron Bandel, Michal Shmueli-Scheuer, and Leshem Choshen. Robustness as an emergent property of task performance. *arXiv preprint arXiv:2602.03344*, 2026.

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.68. URL https://aclanthology.org/2023.findings-emnlp.68/.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48 (1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7/.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430. PMLR, 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

Marcel Brass, Baptist Liefooghe, Senne Braem, and Jan De Houwer. Following new task instructions: Evidence for a dissociation between knowing and doing. *Neuroscience & Biobehavioral Reviews*, 81: 16–28, 2017. ISSN 0149-7634. doi: https://doi.org/10.1016/j.neubiorev.2017.02.012. URL https://www.sciencedirect.com/science/article/pii/S0149763416306856. The power of instructions: the influence of instructions on cognition, behaviour and physical states.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ETKGuby0hcs.

Jason M. Chein and Walter Schneider. The brain's learning and control architecture. *Current Directions in Psychological Science*, 21(2):78–84, 2012. doi: 10.1177/0963721411434977. URL https://doi.org/10.1177/0963721411434977.

Noam Chomsky. *Aspects of the Theory of Syntax*. The MIT Press, Cambridge, 1965. URL http://www.amazon.com/Aspects-Theory-Syntax-Noam-Chomsky/dp/0262530074.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL https://aclanthology.org/P18-1198/.

Artur d'Avila Garcez and Luís C. Lamb. Neurosymbolic AI: the 3rd wave. *Artif. Intell. Rev.*, 56(11):12387–12406, 2023. doi: 10.1007/S10462-023-10448-W. URL https://doi.org/10.1007/s10462-023-10448-w.

Ferdinand de Saussure. *Cours de linguistique générale*. Payot, Paris, 1916. URL https://books.google.ch/books?id=B38KAQAAMAAJ.

Gary Dell. A spreading-activation theory of retrieval in sentence production. *Psychological Review*, 93: 283–321, 07 1986. doi: 10.1037/0033-295X.93.3.283.

Robert Desimone and John S. Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18:193–222, 1995. URL https://api.semanticscholar.org/CorpusID:14290580.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Jiahai Feng, Stuart Russell, and Jacob Steinhardt. Monitoring latent world states in language models with propositional probes. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=0yvZm2AjUr.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, 2024.

Zorik Gekhman, Eyal Ben-David, Hadas Orgad, Eran Ofek, Yonatan Belinkov, Idan Szpektor, Jonathan Herzig, and Roi Reichart. Inside-out: Hidden factual knowledge in LLMs. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=f7GG1MbsSM.

Daniela Gottesman and Mor Geva. Estimating knowledge in large language models without generating a single token. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3994–4019, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.232. URL https://aclanthology.org/2024.emnlp-main.232/.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948. URL https://doi.org/10.48550/arXiv.2501.12948.

Eliya Habba, Ofir Arviv, Itay Itzhak, Yotam Perlitz, Elron Bandel, Leshem Choshen, Michal Shmueli-Scheuer, and Gabriel Stanovsky. DOVE: A large-scale multi-dimensional predictions dataset towards meaningful LLM evaluation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11744–11763, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.611. URL https://aclanthology.org/2025.findings-acl.611/.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.624. URL https://aclanthology.org/2023.findings-emnlp.624/.

Juyeon Heo, Christina Heinze-Deml, Oussama Elachqar, Kwan Ho Ryan Chan, Shirley You Ren, Andrew Miller, Udhyakumar Nallasamy, and Jaya Narain. Do LLMs "know" internally when they follow instructions? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=qIN5VDdEOr.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419/.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7038–7051, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.564. URL https://aclanthology.org/2021.emnlp-main.564/.

Jennifer Hu and Michael Frank. Auxiliary task demands mask the capabilities of smaller language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=U5BUzSn4tD.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi U. Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian C. Paulun, Maria Ryskina, Ekin Akyürek, Ethan G. Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *Transactions of the Association for Computational Linguistics*, 13:1245–1270, 10 2025. ISSN 2307-387X. doi: 10.1162/TACL.a.38. URL https://doi.org/10.1162/TACL.a.38.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *CoRR*, abs/2401.04088, 2024. doi: 10.48550/ARXIV.2401.04088. URL https://doi.org/10.48550/arXiv.2401.04088.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. Discourse probing of pretrained language models. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 3849–3864, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.301. URL https://aclanthology.org/2021.naacl-main.301/.

Matthew Le, Y-Lan Boureau, and Maximilian Nickel. Revisiting the evaluation of theory of mind through question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5872–5877, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1598. URL https://aclanthology.org/D19-1598/.

Michael A. Lepori, Tal Linzen, Ann Yuan, and Katja Filippova. Language models struggle to use representations learned in-context. 2026. URL https://arxiv.org/abs/2602.04212.

Willem J. M. Levelt. *Speaking: From Intention to Articulation*. The MIT Press, 02 1989. ISBN 9780262278225. doi: 10.7551/mitpress/6393.001.0001. URL https://doi.org/10.7551/mitpress/6393.001.0001.

Juncai Li, Ru Li, Xiaoli Li, Qinghua Chai, and Jeff Z. Pan. Inference helps PLMs' conceptual understanding: Improving the abstract inference ability with hierarchical conceptual entailment graphs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22088–22104, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1233. URL https://aclanthology.org/2024.emnlp-main.1233/.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.556. URL https://aclanthology.org/2022.acl-long.556/.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Earl K. Miller and Jonathan D. Cohen. An integrative theory of prefrontal cortex function. *Annual review of neuroscience*, 24:167–202, 2001. URL https://api.semanticscholar.org/CorpusID:7301474.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.759. URL https://aclanthology.org/2022.emnlp-main.759/.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? a call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024. doi: 10.1162/tacl_a_00681. URL https://aclanthology.org/2024.tacl-1.52/.

Stephen Monsell. Task switching. *Trends in Cognitive Sciences*, 7(3):134–140, 2003. ISSN 1364-6613. doi: https://doi.org/10.1016/S1364-6613(03)00028-7. URL https://www.sciencedirect.com/science/article/pii/S1364661303000287.

Moin Nadeem, Anna Bethke, and Siva Reddy. StereoSet: Measuring stereotypical bias in pretrained language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5356–5371, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.416. URL https://aclanthology.org/2021.acl-long.416/.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=KnqiC0znVF.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *CoRR*, abs/2209.11895, 2022. doi: 10.48550/ARXIV.2209.11895. URL https://doi.org/10.48550/arXiv.2209.11895.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=KRnsX5Em3W.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018. URL https://api.semanticscholar.org/CorpusID:49313245.

Jacqueline Strunk Sachs. Recognition memory for syntactic and semantic aspects of connected discourse. *Perception & Psychophysics*, 2(9):437–442, 1967.

Carson T. Schütze. *The empirical base of linguistics*. Number 2 in Classics in Linguistics. Language Science Press, Berlin, 2016. doi: 10.17169/langsci.b89.100.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=RIu5lyNXjT.

Aviv Slobodkin, Omer Goldman, Avi Caciularu, Ido Dagan, and Shauli Ravfogel. The curious case of hallucinatory (un)answerability: Finding truths in the hidden states of over-confident large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3607–3625, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.220. URL https://aclanthology.org/2023.emnlp-main.220/.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758, 2020. doi: 10.1162/tacl_a_00342. URL https://aclanthology.org/2020.tacl-1.48/.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452/.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019b. URL https://openreview.net/forum?id=SJzSgnRcKX.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AwyxtyMwaG.

Polina Tsvilodub, Hening Wang, Sharon Grosch, and Michael Franke. Predictions from language models for multiple-choice tasks are not robust under variation of scoring methods. *CoRR*, abs/2403.00998, 2024. doi: 10.48550/ARXIV.2403.00998. URL https://doi.org/10.48550/arXiv.2403.00998.

Martin Tutek, Fateme Hashemi Chaleshtori, Ana Marasovic, and Yonatan Belinkov. Measuring chain of thought faithfulness by unlearning reasoning steps. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9946–9971, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.504. URL https://aclanthology.org/2025.emnlp-main.504/.

Leendert Van Maanen, Yuyao Zhang, Maarten De Schryver, and Baptist Liefooghe. The curve of learning with and without instructions. *Journal of Cognition*, 7(1):48, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 183–196, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.14. URL https://aclanthology.org/2020.emnlp-main.14/.

Andreas Waldis, Yotam Perlitz, Leshem Choshen, Yufang Hou, and Iryna Gurevych. Holmes: A benchmark to assess the linguistic competence of language models. *Transactions of the Association for Computational Linguistics*, 12:1616–1647, 2024. doi: 10.1162/tacl_a_00718. URL https://aclanthology.org/2024.tacl-1.88/.

Andreas Waldis, Vagrant Gautam, Anne Lauscher, Dietrich Klakow, and Iryna Gurevych. Aligned probing: Relating toxic behavior and model internals. *CoRR*, abs/2503.13390, 2025. doi: 10.48550/ARXIV.2503.13390. URL https://doi.org/10.48550/arXiv.2503.13390.

Evan Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, Michal Guerquin, David Heineman, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William

Merrill, Lester James Validad Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Jake Poznanski, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 OLMo 2 furious (COLM's version). In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=2ezugTT9kU.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020. doi: 10.1162/tacl_a_00321. URL https://aclanthology.org/2020.tacl-1.25/.

Albert Webson and Ellie Pavlick. Do prompt-based models really understand the meaning of their prompts? In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2300–2344, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.167. URL https://aclanthology.org/2022.naacl-main.167/.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. URL https://proceedings.mlr.press/v139/zhao21c.html.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=KBMOKmX2he.

Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. ProSA: Assessing and understanding the prompt sensitivity of LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1950–1976, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.108. URL https://aclanthology.org/2024.findings-emnlp.108/.

A Limitations

We discuss four limitations of this work.

Task scope. Our evidence is based on binary judgment tasks with single-token outputs, chosen so that probing and behavior share the same target and can be directly compared. Whether the production-centered mechanism extends to open-ended generation, multi-step reasoning, or longer outputs—and if so, whether it takes the same form—is an open question and a natural direction for future work.

Token positions as a proxy for processing and production. We operationalize the cognitive processing–production distinction via token positions: $\vec{h}_S$ at sample tokens as a proxy for the stage of language processing, $\vec{h}_O$ at output tokens as a proxy for language production. This is an approximation, not a claim of strict computational separation: upper-layer sample-token representations no longer reflect input encoding alone but incorporate task-relevant transformations that prepare the output, so $\vec{h}_S$ does not isolate the processing stage in a narrow encoding-only sense. Conversely, early-layer output-token representations are still being transformed and not yet committed to generation, so $\vec{h}_O$ captures all computation over output tokens rather than the production stage in a narrow decoding-only sense. Given the lack of sharp computational boundaries, we use this token-position abstraction as an analytical lens to locate instruction effects within the forward pass. Even so, our results show that the two stages behave systematically differently across model families, scales, and tasks, indicating that the token-position distinction tracks a meaningful computational separation despite the absence of architectural boundaries in decoder-only models.

Linear decodability. Linear probes provide a lower bound on the linearly accessible information, so the stability we observe in sample representations could, in principle, partly reflect the limits of linear decodability rather than the representation's full information content. Our selectivity (Hewitt & Liang, 2019), information-theoretic (Voita & Titov, 2020), and non-linearity checks (Appendix § B.2) directly address this: probes show high selectivity against control tasks, information-theoretic and accuracy-based measurements reveal consistent patterns, and adding non-linear capacity to the probes does not change the layer-wise information dynamics we report. Together, these validations robustly demonstrate that the observed asymmetry reflects representational structure rather than a probing artifact, although any model interpretability method can only approximate internal representations in the absence of ground truth.

Confounds in the attention-blocking interventions. The −58.0 pp behavioral drop under the `full` intervention (§ 4) reflects at least two effects: a degraded ability to express task-relevant information in output tokens—confirmed by the accompanying drop in output-token probe accuracy—and the potential loss of output-formatting cues the model relies on to produce a well-formed binary answer. The latter is a confound that the present design cannot fully disentangle. We therefore read the behavioral drop as an upper bound on the task-information-specific contribution of the intervention—and as a concrete instance of the more general point made in § 8: behavioral measurements alone cannot distinguish processing-stage from production-stage contributions.

B Appendix

B.1 Model Checkpoints

Table 2 lists the Hugging Face checkpoints of the 14 individual evaluated models used throughout this work. For each family, we use both the base and instruction-tuned variants where available.

Family	Main experiments	Scaling analysis (sizes)
Llama-3.1	`meta-llama/Llama-3.1-8B(-Instruct)`	8B
OLMo-2	`allenai/OLMo-2-1124-7B(-Instruct)`	1B, 7B, 13B, 32B
Qwen-2.5	`Qwen/Qwen2.5-7B(-Instruct)`	0.5B, 1.5B, 3B, 7B, 14B, 32B

Table 2: Checkpoints used throughout the paper. Base and instruct variants are evaluated in parallel where available.

B.2 The Reliability of the Probing Setup

Figure 8 validates the probing setup along three dimensions across all model families, sizes, and tasks. First, we report high probing selectivity using control tasks (Hewitt & Liang, 2019), confirming that probes learn from the representations rather than from their own capacity. Second, we assess probes from an information-theoretic perspective (Voita & Titov, 2020), and we find consistent patterns with the accuracy-based results

reported in the main text. Third, we verify that as few as 100 to 200 samples are sufficient to reveal stable information patterns, confirming that our findings are not an artifact of sample size. Finally, we compare in Figure 9 the probing results with no intermediate layer (*linear*) to those with one (*linear-1*) or two (*linear-2*) layers. Since nonlinear probes yield only slight differences in information levels without changing the overall layer-wise pattern, we conclude that linear probes capture task-specific information with sufficient reliability.

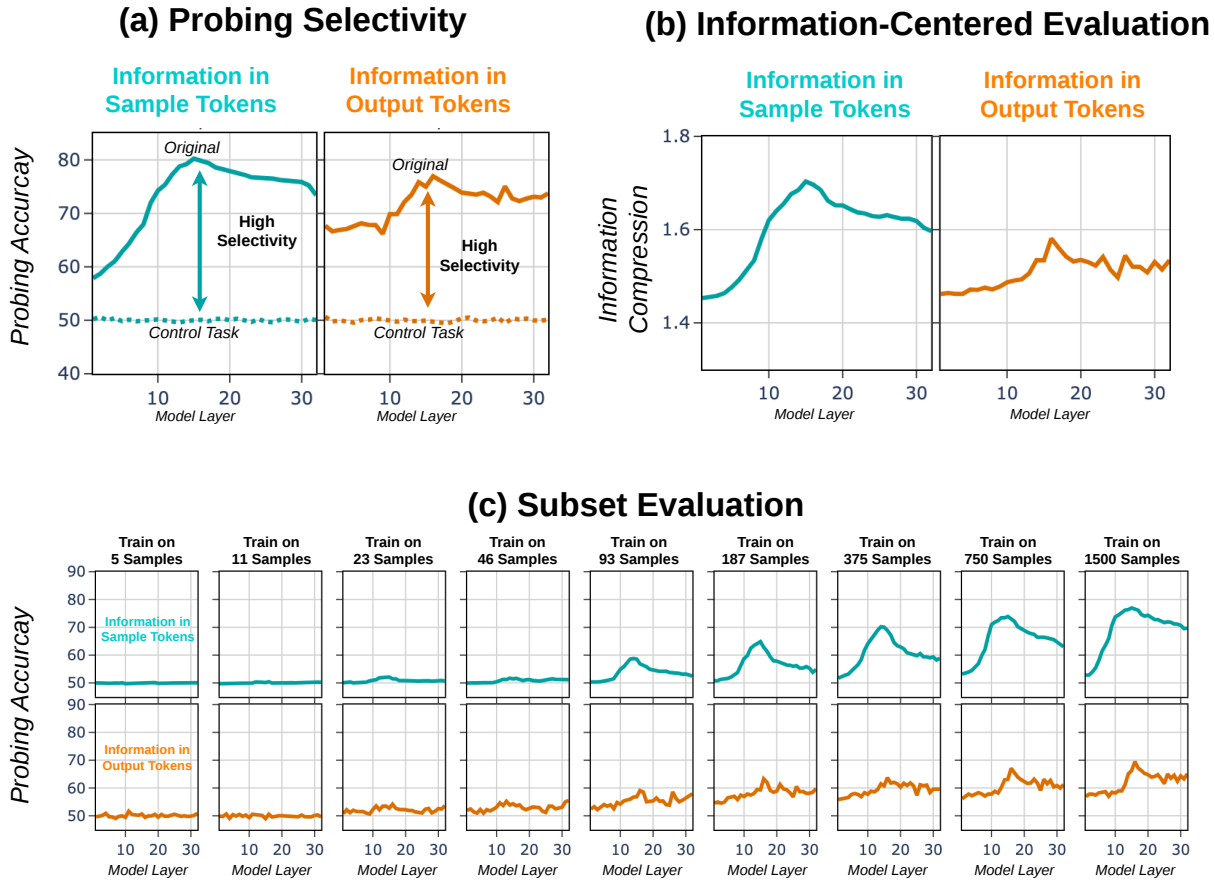

Figure 8: Validation of the probing setup across model layers, averaged across tasks and models. **(a)** Probing selectivity: accuracy of probes trained on original representations compared to control tasks (Hewitt & Liang, 2019), shown separately for `sample` and `output` tokens. High selectivity confirms that probes capture task-relevant structure rather than exploiting their own capacity. **(b)** Information-theoretic assessment (Voita & Titov, 2020) of task-specific information in `sample` and `output` token representations, showing consistent patterns with accuracy-based probing results. **(c)** Probing accuracy as a function of training set size, confirming that stable information patterns emerge with as few as 100 to 200 samples.

B.3 Sanity Checks of the Impact of Instruction

We verify the observed variation in the impact of instructions on language processing and production using sanity prompts that introduce semantic noise or increase task difficulty, based on the best-performing variation ($\mathcal{P}_{\curvearrowleft}$).

Unrelated instructions primarily change language production. We first verify our previous results by introducing semantically unrelated instructions that are irrelevant to the judgment task. Specifically, we ask the LMs whether a given sentence contains the letter "*a*" a specific number of times—expecting "*yes*" when the specific number corresponds to the sentence and "*no*" if not. Following results shown in Figure 10, these unrelated instructions caused a substantial drop (-10.0 probing accuracy) in task-specific

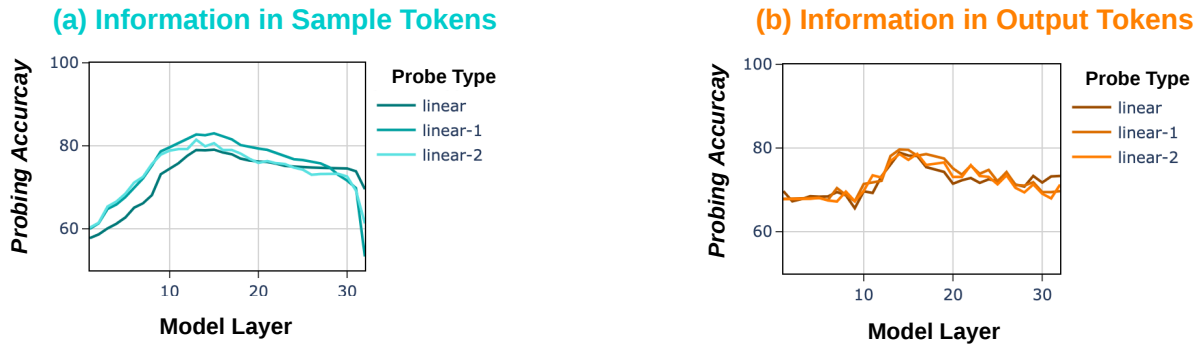

Figure 9: Validation of the probing setup with no intermediate hidden layer (*linear*). We also assess task-specific information with `sample` (a) and `output` (b) tokens using a probe with either one (*linear-1*) or two intermediate layers (*linear-2*) of dimension 100. Results show slight variance in information strength for sample tokens with *linear-1*, but no substantial change in the overall information pattern across model layers.

information in the language production stage (output tokens). In contrast, the information loss within sample tokens, representing the language processing stage, is much lower (-2.0). These results confirm the primary mechanism: instructions primarily affect how information is transformed into output text during production, but not how information from the input is initially encoded.

Increasing task demands mainly impact model behavior. Second, we verify the mechanism by increasing task demands via additional required reasoning steps: *i)* reversing the required label ("*no*" instead of "*yes*" for positive judgments); *ii)* randomly applying this label flip; *iii)* requiring abstract answers ("*apple*" for positive, "*banana*" for negative); and *iv)* requiring random word pairs. Behavioral results from Figure 10 align with prior work (Webson & Pavlick, 2022) and show a substantial impact of these variations, with up to -20.9 less performance when *flipping* the label meaning for all samples. Notably, we found only a minor impact on task-specific information of these prompting variations in both sample and output tokens. Consistent with our previous findings, these insights suggest that instructions activate a shared, stable, and latent knowledge base but employ an unstable function to apply that knowledge. The stability of task-specific information, particularly during input encoding, is maintained, only varying during language production when contrasting high-impact prompting variations (like $\mathcal{P}_{\curvearrowright}$ vs. $\mathcal{P}_{\curvearrowright}$). These results extend previous findings that behavioral assessments are unreliable (Hu & Frank, 2024; **?**) and suggest that information-based measures offer a more robust view of underlying knowledge.

B.4 Detailed Instance-Level Probing–Prompting Alignment

Figure 11 shows the breakdown of correctness for probing–prompting alignment across model layers, extending the variation-level agreement analysis in § 4 by showing not only how often probing and prompting disagree, but also why and how persistently. For sample tokens, the dominant disagreement is cases where probing is correct but prompting fails, which decreases across layers as both converge in upper layers. This shows that processing-stage representations encode task-relevant information before the model expresses it behaviorally—probing detects this structure early, but the model has not yet translated it into correct output. Importantly, when both are correct for sample tokens, this agreement holds consistently across nearly all layers, confirming that correct processing-stage encoding is robust and stable once established. Disagreement categories, by contrast, tend to be layer-specific rather than persistent. For output tokens, the dominant disagreement is cases where both probing and prompting fail, most pronounced in early layers and decreasing toward upper layers. Unlike sample tokens, these errors persist across more layers, indicating that production failures reflect a more fundamental lack of reliable task-relevant structure rather than a transitional encoding gap. Panel (b) further shows that "both correct" for output tokens is less consistently maintained across layers than for sample tokens, meaning that correct production-stage encoding is more

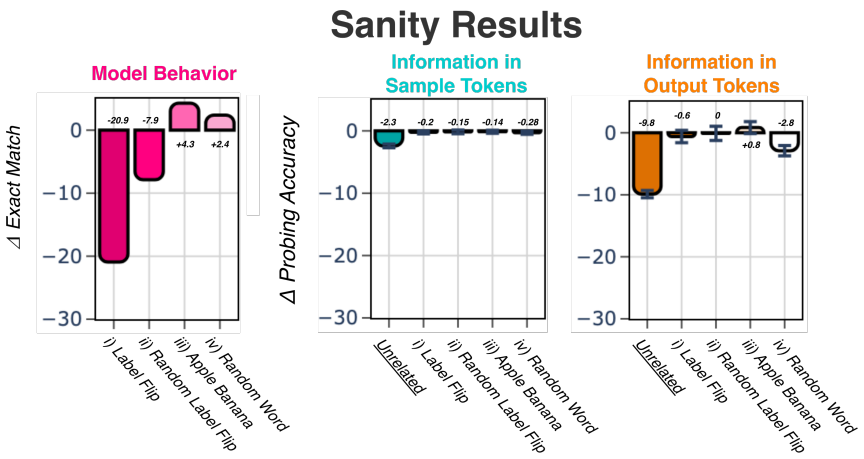

Figure 10: Sanity checks of the production-centered mechanism, averaged across tasks and models. **Left**: impact of semantically unrelated instructions on task-specific information in `sample` and `output` tokens and on behavioral performance, relative to the standard instruction-first variation ($\mathcal{P}_{\curvearrowleft}$). **Right**: impact of increasing task demands—label flip, random label flip, abstract answers (apple/banana), and random word pairs—on behavior, `sample` token information, and `output` token information. In both conditions, `sample` token information remains largely stable while behavioral performance and `output` token information are substantially affected, confirming that the processing stage is robust to instruction variation.

layer-dependent and less stable. Together, these results show that processing and production not only differ in their sensitivity to instructions but also in the stability of their encoding: processing develops task-relevant structure early and maintains it robustly, while production alignment is more fragile and depth-dependent.

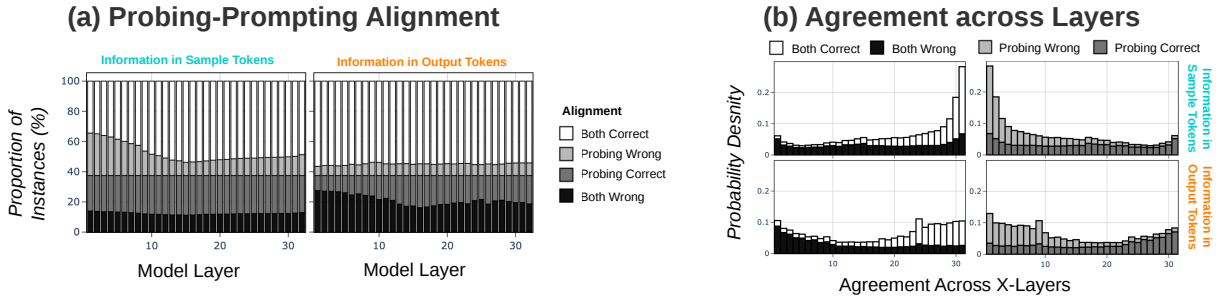

Figure 11: Instance-level probing–prompting alignment, averaged across tasks and models. **(a)** Proportion of instances per layer falling into each agreement category for `sample` (left) and `output` (right) tokens: both correct (white), probing wrong only (light gray), probing correct only (dark gray), and both wrong (black). **(b)** Probability density of the number of layers across which each agreement category holds consistently, shown separately for `sample` and `output` tokens. Correct agreement on `sample` tokens is concentrated at maximum layer consistency, reflecting stable processing-stage encoding, whereas correct agreement on `output` tokens is more broadly distributed, reflecting greater fragility of production-stage alignment.

B.5 Absolute Layer Comparison Across Model Sizes

While the main analysis focuses on relative layer positions to allow comparison across models of different sizes, comparing models in terms of absolute layer positions reveals two additional patterns, as shown in Figure 12. First, for sample tokens, all model sizes start at similar probing accuracy levels in the early layers, only diverging as they rise toward their respective peaks. This suggests that the basic encoding of task-specific information during processing emerges similarly regardless of model size, and that scaling primarily extends

how far and how long this encoding develops rather than changing how it starts. Second, for output tokens, the picture is different. Smaller models (0.5B, 1.5B, 3B) exhibit flat, narrow encoding curves throughout, whereas larger models (7B, 32B) rise substantially higher and retain task-specific information across a much broader range of layers. Rather than a smooth continuation as in sample tokens, the production stage does not scale gradually but changes more fundamentally above a certain model size. These observations suggest that while processing-stage encoding scales smoothly and continuously, production-stage encoding undergoes a more qualitative shift with model size—consistent with the main text's finding that scaling disproportionately benefits production.

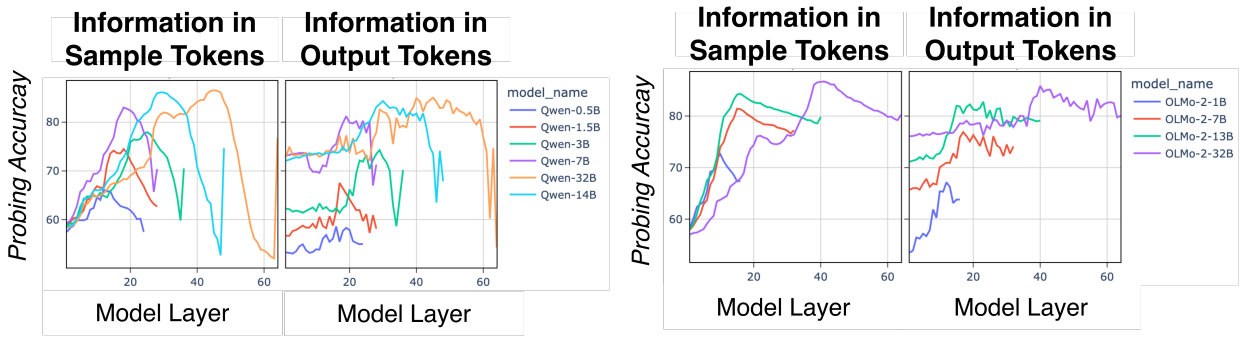

Figure 12: Layer-wise task-specific information for `sample` (left) and `output` (right) tokens across five `Qwen-2.5` model sizes, shown in terms of absolute layer positions. Each curve represents one model size. For `sample` tokens, all sizes converge at similar early-layer accuracy and diverge only as they rise toward their peaks, suggesting that scaling extends processing-stage encoding depth rather than fundamentally changing how it starts. For `output` tokens, smaller models (0.5B–3B) show flat, narrow curves while larger models (7B–32B) rise substantially higher across a broader layer range, reflecting the more qualitative shift in production-stage encoding with scale.

B.6 Detailed Task Results

Behavioral Results Across Prompting Variations Figure 13 shows behavioral performance across prompting variations for each task. The most pronounced variation arises from few-shot prompting without instructions ($\mathcal{P}_{w/o}$). For `BLiMP`, replacing instructions with four-shot examples substantially decreases performance ($\Delta \approx -15.0$), consistent with the strong sensitivity of surface-sensitive tasks to instruction-driven production established in § 6. In contrast, `ToM` benefits from concrete demonstrations, yielding the highest performance under $\mathcal{P}_{w/o}$, suggesting that examples are more informative than abstract task descriptions for reasoning about mental states. Knowledge tasks (`EWOK`, `oLMpics`) remain largely robust across variations, consistent with their stable behavior–output coupling. Instruction placement has comparatively minor effects overall, with a slight advantage for standard ordering on form-sensitive tasks (`BLiMP`, `StereoSet`). `StereoSet` drops noticeably under few-shot prompting, reflecting its particular sensitivity to stereotype-related content in demonstrations.

Causal Intervention Results Figure 14 shows the effect of the prompt-only intervention across tasks, which cuts attention between instruction and sample tokens under the instruction-first prompting variation ($\mathcal{P}_{\curvearrowright}$). Across all tasks, sample token information remains largely stable, with only minor deviations visible in the upper model layers (≤ -2.0 for `BLiMP` and `StereoSet`), confirming that the processing stage is largely unaffected by instruction flow regardless of task type. The production stage and behavior show stronger and more task-specific responses. `BLiMP` exhibits the largest effects: output token information drops by -4.0 to -7.0 across layers, and behavioral performance falls by -13.0, consistent with the tighter processing–production coupling and greater sensitivity to instruction-driven production identified in § 6. Knowledge tasks (`oLMpics`, `EWOK`) show minimal behavioral effects in either direction, reflecting the stable behavior–output coupling and the decoupling of processing from behavior established in § 6. `ToM` yields a positive behavioral effect ($+6.0$), indicating that instruction tokens reaching sample tokens introduce interference for

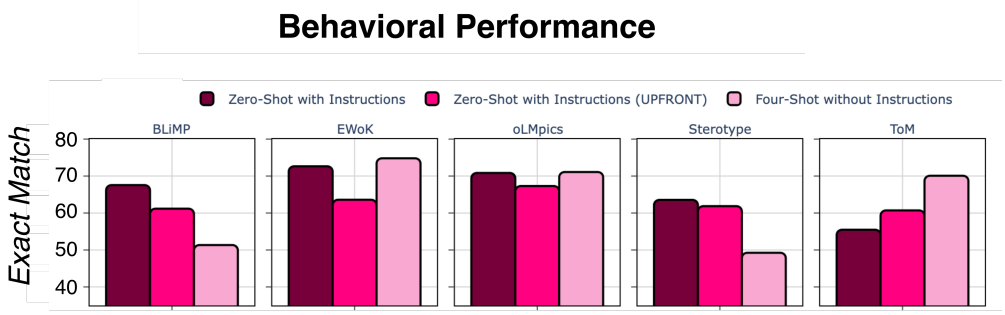

Figure 13: Behavioral performance (EM) across the three prompting variations ($\mathcal{P}_{\frown}$, $\mathcal{P}_{\frown}$, $\mathcal{P}_{w/o}$) for each judgment task, averaged across models. `BLiMP` shows the strongest sensitivity to instruction removal, while knowledge and reasoning tasks remain largely robust. `ToM` is the only task that benefits from few-shot demonstrations over explicit instructions.

this reasoning task rather than providing a useful signal. This further supports the view that for knowledge and reasoning tasks, the processing stage is not only decoupled from behavior but can actively work against it, making production the sole reliable pathway to correct output.

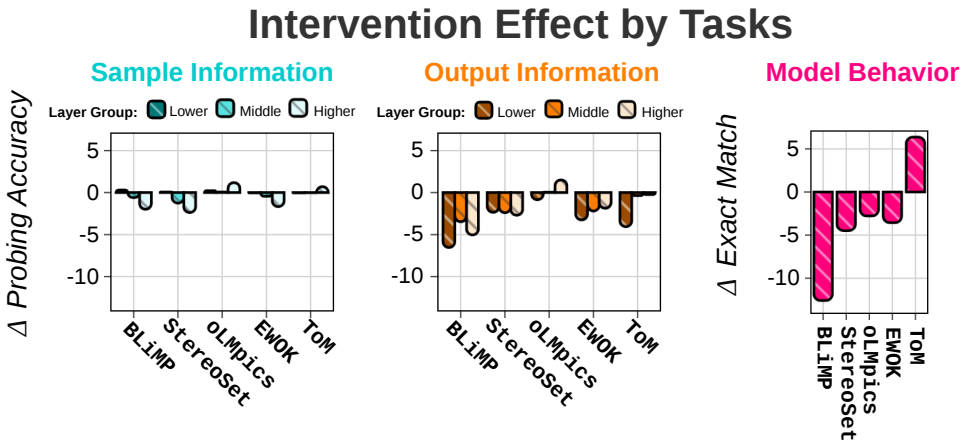

Figure 14: Effect of the prompt-only intervention on task-specific information in `sample` and `output` tokens and on behavioral performance, shown per judgment task and averaged across models. Results are reported as differences relative to the unmodified instruction-first variation ($\mathcal{P}_{\frown}$), grouped by lower, middle, and upper model layers. `sample` token information remains stable across all tasks, while `output` token information and behavioral performance show task-specific responses that reflect the processing–production spectrum established in § 6.

Task-Level Impact of Instruction-Tuning Figure 15 compares base and instruction-tuned models per task, extending the aggregated comparison in § 5. Across all tasks, sample token curves for base and instruction-tuned models overlap closely, confirming that post-training does not substantially change how task-relevant information is encoded during processing regardless of task type. Instruction-tuning consistently increases output token information, but the magnitude of this gain varies with the task spectrum: the largest production-stage gains occur for knowledge and reasoning tasks (`oLMpics`, `EWOK`, `ToM`), where production is already the primary behavioral determinant, while gains are more modest for surface-sensitive tasks (`BLiMP`, `StereoSet`). For `StereoSet`, the behavioral gap between base and instruction-tuned models remains small despite the low behavioral ceiling, consistent with the production-stage disruption for this task being structural rather than a consequence of insufficient instruction-tuning.

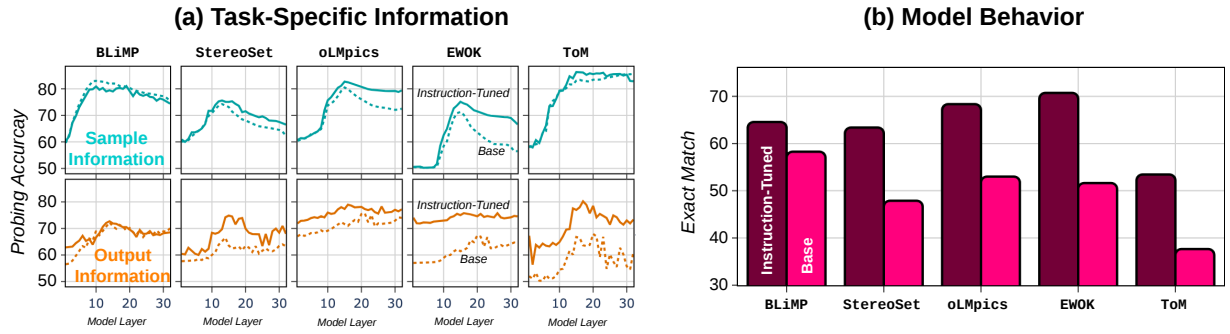

Figure 15: Comparison of pre-trained (*base*, dotted) and instruction-tuned models (solid) per judgment task. **(a)** Layer-wise task-specific information in `sample` (top) and `output` (bottom) tokens for each task. `sample` token curves are nearly identical across conditions for all tasks, while `output` token curves show task-dependent gains from instruction-tuning, largest for knowledge and reasoning tasks. **(b)** Behavioral performance (EM) for base and instruction-tuned models per task. Instruction-tuning improves behavior most for tasks where production is the primary behavioral determinant, consistent with post-training operating primarily at the production stage.

Task-Level Layer-Wise Agreement of Probing and Prompting Figure 16 extends the aggregated consistency analysis in § B.4 by showing the layer-wise agreement distributions per task, separately for cases where probing and prompting agree (top panels) and disagree (bottom panels). For `BLiMP`, correct agreement on output representations is sharply concentrated at maximum layer consistency, indicating that tight processing–production coupling translates into stable correct production. For `StereoSet`, wrong and correct agreement on output representations are comparably distributed across consistency values, lacking the sharp concentration at maximum consistency seen for other tasks—consistent with the production-stage disruption identified in § 6. For knowledge and reasoning tasks (`oLMpics`, `EWOK`), correct output agreement is also concentrated at maximum consistency, reflecting reliable production once engaged, while input representations show no such concentration, confirming the decoupling of processing from behavior for these tasks. `ToM` shows diffuse distributions across all quadrants, consistent with the interference introduced by instruction tokens at the processing stage identified in the intervention results above.

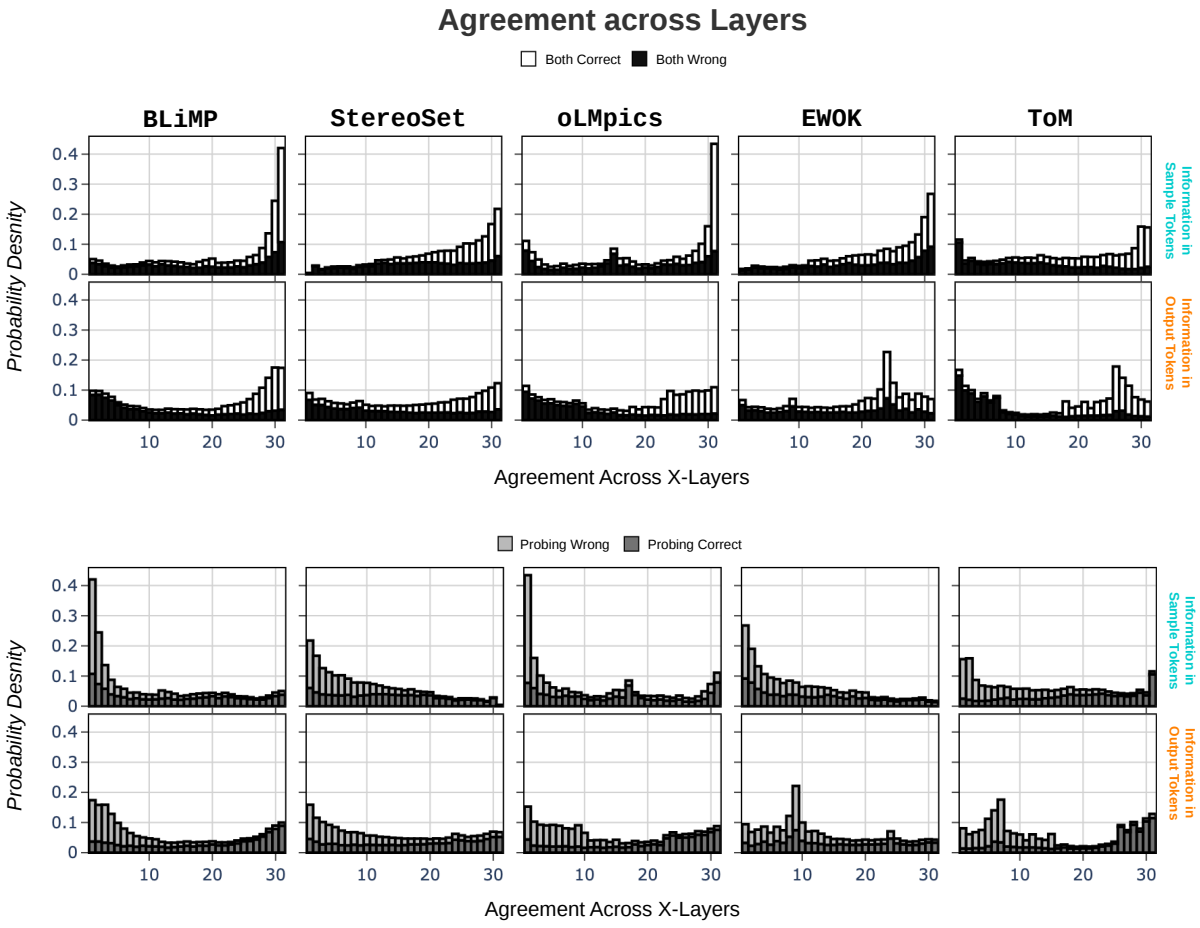

Figure 16: Layer-wise probing–prompting consistency distributions per judgment task, for `sample` (bottom rows) and `output` (top rows) representations. The x-axis indicates the number of layers across which a given probing prediction is consistent. **Top panels**: cases where probing and prompting agree, split by whether the prediction is correct (white) or wrong (black). **Bottom panels**: cases where probing and prompting disagree, split by whether probing is correct (white) or wrong (black). For `BLiMP`, correct output agreement is sharply concentrated at maximum consistency. For `StereoSet`, correct and wrong agreement distributions are comparably spread, reflecting unstable production-stage commitments. For knowledge and reasoning tasks, correct output agreement concentrates at maximum consistency while input distributions remain flat, illustrating the processing–production decoupling.