UNDERSTANDING THE KNOWLEDGE–BEHAVIOR GAP IN LANGUAGE MODELS THROUGH ANSWER MEASUREMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

The knowledge–behavior gap refers to the discrepancy between the answers a language model's internal representations support and the answers it actually gives. Prior work identifies this gap by comparing what probes recover from hidden states with what models answer, and shows that probing can partly mitigate it. However, how the model's answering process loses internal answer information remains unclear. We develop a measurement theory of answer support, the total probability mass of continuations expressing the same answer. Answering estimates this support through probability readings and commits to the answer with the highest reading. Two measurement properties are key: the **measurement unit**, which determines whether readings concern complete answers or individual expressions, and the **measurement scale**, which determines whether they share a common reference. Mismatch can distort support ordering, preventing supported answers from becoming commitments. Accordingly, we construct native measurements that vary in measurement unit and scale. We further prove that, when answer support is measurable from complete-answer representations, a shared map places candidates formed along separate generation paths on a common answer-level scale. Across four frozen models, aligning either property improves answering accuracy and robustness to answer rewording; aligning both yields the highest accuracy throughout and the lowest flip rate in most settings. Supervised by the model's own support ordering, shared linear maps improve native measurements in nearly all settings and generalize even when trained only on support relations among incorrect answers. Used as generation feedback, the learned readings also improve free-form answers. Measuring answer support and converting it into commitments thus offers a route to understanding and narrowing the knowledge–behavior gap.

1 INTRODUCTION

Large language models acquire extensive knowledge during training (Petroni et al., 2019) and perform well on a wide range of tasks (Brown et al., 2020; OpenAI, 2023). Probing their hidden states, however, reveals that even when a model's internal representations encode the correct answer, the model may still give an incorrect one (Burns et al., 2023; Li et al., 2023; Orgad et al., 2025). This phenomenon is known as the **knowledge–behavior gap**. It raises a question: **why does internally available answer information fail to become a correct answer commitment?**

Existing work mitigates this gap with probes or activation interventions that read or steer the correct answer from hidden states (Burns et al., 2023; Li et al., 2023). However, the mechanism by which a model's own answering loses internally represented information remains unclear.

We develop a measurement theory of answer support to study how a model converts answer information into commitments. We use knowledge in a model-relative sense: the information the model encodes about possible answers to a question, including both correct and incorrect answers. An autoregressive language model defines a probability measure over continuations through token-level conditional probabilities. For a given question, **answer support** is the total probability mass of all continuation trajectories expressing the same answer. A probability reading for an individual response estimates this support, and answering commits to the candidate with the largest reading. Our account

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

locates the gap in this conversion: when the measurement used for commitment distorts the ordering of answer support, internally supported answers fail to appear in behavior. The mismatch arises in two properties of the measurement, its unit and its scale.

The **measurement unit** determines whether a reading concerns a complete answer or one of its textual expressions. The probability of a specific text factors into the support of its answer and the probability of that expression given the answer, so a token-level reading mixes the two. The **measurement scale** determines whether competing answers are compared against a common reference. Candidates measured separately each face an implicit reference of other possible continuations, whereas candidates measured together share an explicit answer-level reference. Autoregressive generation connects the two properties: the model commits token by token along separate histories while complete answers gradually take shape. The final commitment requires comparing these separately formed answers on a common support scale.

The theory characterizes a faithful answering procedure by preservation of answer-support ordering. Under a separable measurement model, answer-level labels and balanced presentation orders yield a common readout that preserves this ordering. It also unifies separate effects: competition among surface forms of the same answer (Holtzman et al., 2021) can be understood as a mismatch in unit, and miscalibrated readouts (Liu et al., 2023) as a mismatch in scale.

To test the framework's predictions for answering behavior, we construct four native measurements that vary the two properties and compare them on ARC-C and MMLU across four frozen pretrained models. Aligning either property improves accuracy over the token-level measurement with separate references in every setting, and aligning both yields the highest accuracy throughout. If the gap stems from measurement mismatch, aligned measurements should also be robust to how an answer is worded. When the correct answer is reworded, aligning either property reduces how often the selected answer changes in every setting, and aligning both gives the lowest or joint-lowest flip rate in most settings.

Answer support also indicates how to repair a mismatched measurement. We prove that, when answer support is measurable from complete-answer representations, a map shared across questions and candidates places these representations on a common support scale (Section 3.4). Linear maps over frozen representations implement a learnable version of this construction and improve on native measurements in nearly every setting. They remain effective when trained only on support relations among incorrect answers: model-derived support comparisons thus teach a transferable map even when correct candidates are excluded from training. The learned measure also supplies answer-level feedback for free generation: propagating its score through gradients to the hidden states of generation guides a fresh response and improves accuracy on mathematical question answering.

Our main contributions are:

- We develop a measurement theory of answer support, characterize unit and scale distortions, and establish support-order preservation for balanced answer-label readings under a separable measurement model.
- Through four native measurements and meaning-preserving rewording on ARC-C and MMLU, we show that aligning either property from a token-level, separate-reference baseline improves accuracy and stability; aligning both yields the highest accuracy throughout and the lowest flip rate in most settings.
- We develop a shared-map construction for reading separately formed candidate representations on a common answer-level scale, implement it with linear maps learnable from incorrect answers alone, and use the learned readings to guide free generation.

2 Related Work

The knowledge–behavior gap. A growing body of work finds that language models represent more than their answers express. Linear probes and unsupervised methods recover the truthfulness of statements from hidden states (Burns et al., 2023; Azaria & Mitchell, 2023), and probes on internal activations identify correct answers more accurately than the model's own generations (Li et al., 2023; Orgad et al., 2025). Models also assess the correctness of their own answers better than their generations suggest (Kadavath et al., 2022), and some answers that are recoverable internally are rarely or never generated (Gekhman et al., 2025). Related inconsistencies appear between generating

and validating an answer (Li et al., 2024) and between prompted judgments and direct probability measurements (Hu & Levy, 2023). Liu et al. (2023) find that probes outperform queries to the model and that the two are often accurate on different inputs. These studies document the gap or exploit it with external readouts. We instead explain its recoverable part through the model's native answer measurements: the gap arises when the commitment measurement is misaligned with answer support in unit and scale, and aligning both recovers correct commitments without changing the model.

Answer measurement. Prior work studies surface-form competition among answer expressions (Holtzman et al., 2021; Wiegreffe et al., 2023) and selection biases from option identifiers and positions (Pezeshkpour & Hruschka, 2024; Zheng et al., 2024). These works characterize particular sources of distortion in answer scoring. We place them in a common measurement framework: every native measurement has a measurement unit and a measurement scale, and faithful answering requires its readings to preserve the ordering of answer support.

3 A Measurement Theory of Answer Support

Research Question

How does the measurement of answer support shape answer commitments and give rise to the knowledge–behavior gap?

We distinguish answer support, its measurement, and the commitment induced by the measurement. Fix a question q and an answer space $\mathcal{A}_q$. The model defines a probability measure P_q over complete continuations Y. An answer map ϕ_q assigns each continuation an answer in $\mathcal{A}_q$. This gives a measure on answers (Kallenberg, 2002):

$$\mu_q = (\phi_q)_{\#} P_q, \qquad \mu_q(\{a\}) = P_q(\phi_q(Y) = a)\,. \tag{1}$$

The mass $m_q(a) := \mu_q(\{a\})$ is the model's support for the complete answer a: it aggregates all continuation trajectories expressing that answer. For fixed model parameters θ and question input q, the answer support induced by $P_q = p_\theta(\cdot \mid q)$ serves as the common target across measurement interfaces. Each interface combines a prompt construction, output encoding, and scoring rule to produce a reading for an individual candidate, estimating its support and its ordering relative to other candidates. We study how prompting and readout choices change the event being measured and the comparison reference under which readings are formed, corresponding to the measurement unit and measurement scale, respectively.

The autoregressive model represents P_q through token probabilities (Bengio et al., 2003). For a complete continuation $y = (y_1, \ldots, y_T)$, including termination,

$$\log P_q(Y = y) = \sum_{t=1}^{T} \log P_q(y_t \mid q, y_{<t}). \tag{2}$$

Log probability is thus the model's native reading of a particular continuation. Turning such readings into an ordering of answer support depends on two properties: the **measurement unit**, which specifies the event being read, and the **measurement scale**, which sets the reference under which competing readings are formed and compared.

3.1 Answer Support and Commitment

For a finite candidate set $\mathcal{C}_q \subseteq \mathcal{A}_q$, answer support defines the preferred answer

$$a_q^\star = \arg\max_{a \in \mathcal{C}_q} m_q(a). \tag{3}$$

A measurement procedure r gives each candidate a reading $s_r(q, a)$ and commits to

$$\widehat{a}_r(q) = \arg\max_{a \in \mathcal{C}_q} s_r(q, a). \tag{4}$$

We call a measurement *support-order preserving* when, for every candidate pair,

$$s_r(q, a_i) > s_r(q, a_j) \quad \Longleftrightarrow \quad m_q(a_i) > m_q(a_j). \tag{5}$$

This condition also preserves ties; a common tie-breaking rule then gives $\widehat{a}_r(q) = a_q^\star$. When $a_q^\star$ is correct but $\widehat{a}_r(q) \neq a_q^\star$, the model's answer support has not become its answering behavior.

3.2 MEASUREMENT UNITS

Definition 3.1 (Measurement unit)**.** The measurement unit is the object of a probability reading. A token-level measurement follows the generation of one answer text, using the conditional log probabilities of its tokens. An answer-level measurement directly measures a complete candidate answer.

The unit difference appears in an exact probability identity. If a positive-probability complete continuation y expresses answer a, then

$$\log P_q(Y = y) = \log m_q(a) + \log P_q(Y = y \mid \phi_q(Y) = a)\,. \tag{6}$$

The first term is answer support. The second describes how that support is distributed across texts that express the same answer. One answer can therefore retain the same support while its textual realizations have different log probabilities (Holtzman et al., 2021). An expression change can reverse a pair of text readings while the underlying support ordering stays fixed (Appendix A.1).

An answer-level unit reads the event to which the commitment refers. In a label protocol, one token can uniquely encode a complete answer: its next-token probability marginalizes all subsequent continuations following that commitment. The label's referent therefore determines the measurement unit. Section 3.3 specifies the measurement model connecting these label readings to m_q.

3.3 MEASUREMENT SCALES

Definition 3.2 (Measurement scale)**.** The measurement scale is determined by the comparison reference used when a reading is formed. Scale alignment requires that readings of competing answers express their support on a common scale. For a question q, this requirement is satisfied when one strictly increasing function $T_{r,q}$ applies to every candidate with positive support:

$$s_r(q, a) = T_{r,q}(\log m_q(a))\,, \qquad a \in \mathcal{C}_q, \quad m_q(a) > 0. \tag{7}$$

This condition concerns within-question comparisons.

A reading formed along an individual candidate's continuation uses its evolving context as the reference; a separate answer query likewise uses a candidate-specific context. The alternatives against which its output probability is computed are implicit in that context. These references can introduce candidate-dependent distortions of the target ordering.

Presenting the full candidate set supplies an explicit answer-level reference before readings are formed. When all candidates are compared at one output position, their readings also share the same model state. We formalize this answer-level, common-reference construction with a separable measurement model. Let $\ell^{\pi}_{q,i}$ be the log probability of the label assigned to a_i in presentation order π, and $k_\pi(i)$ its position. Assume

$$\ell^{\pi}_{q,i} = \log m_q(a_i) + c^{\pi}_q + b_q(k_\pi(i)),$$

where c^{π}_q is shared across candidates and b_q is a position effect shared across answer identities. Averaging over balanced orders, in which every answer occupies every position equally often, gives

$$s_{A_C}(q, a_i) = \log m_q(a_i) + K_q.$$

Here K_q is common to all candidates. Under this model, balanced A_C is a support-order-preserving measurement: its pairwise differences recover log support ratios. Appendix A.2 gives the proof and a residual-distortion bound.

3.4 A SHARED MAP FROM SEPARATE REPRESENTATIONS

Native response probabilities are read from model representations to estimate answer support. Unit and scale mismatch can distort this readout. A shared map provides a second measurement construction, reading complete-answer representations directly on a common support scale. Let $h_r(q, a) \in \mathbb{R}^d$ be a representation formed after processing the complete candidate, where r fixes its encoding and representation extraction. For a random question Q, draw a candidate A uniformly from $\mathcal{C}_Q$ and write

$$H_r = h_r(Q, A), \qquad M = m_Q(A). \tag{8}$$

We express the representation premise by taking M to be measurable with respect to $\sigma(H_r)$. The Doob–Dynkin factorization lemma (Kallenberg, 2002) then gives the following shared map.

Proposition 3.1 (Shared answer-support map)**.** *Fix the encoding and layer r. If M is measurable with respect to $\sigma(H_r)$, there exists a measurable map $f_r^\star : \mathbb{R}^d \to [0,1]$, shared across questions and candidates, such that*

$$m_q(a) = f_r^\star(h_r(q,a)) \quad \textit{almost surely}. \tag{9}$$

For almost every question q, let $g_r^\star = \log f_r^\star$ on positive-support candidate states. For every such pair a_i, a_j,

$$g_r^\star(h_r(q,a_i)) - g_r^\star(h_r(q,a_j)) = \log \frac{m_q(a_i)}{m_q(a_j)}. \tag{10}$$

The proof is given in Appendix A.3.

Key Result: Shared map

This construction brings the measurement unit and scale together. Each evaluation of $g_r^\star$ measures a complete answer, and its within-question differences recover log support ratios. Candidates can thus follow separate autoregressive paths yet admit an answer-level measurement on a common scale.

The factorization establishes the existence of a shared measurable map. We study a linear map $g_{w,r}(h) = w_r^\top h$ as a simple learnable realization of this construction:

$$\widehat{a}_{w,r}(q) = \arg\max_{a \in \mathcal{C}_q} w_r^\top h_r(q,a). \tag{11}$$

Section 5.1 uses balanced A_C ordering as model-derived supervision and evaluates the resulting commitments on held-out questions. This tests whether a shared linear realization can use support comparisons to improve native answer readings.

4 Measurement Interfaces and Answering Behavior

Section 3 connects answer-support measurement to answer commitment. We test its behavioral predictions with four native measurements organized by unit and scale. Finite candidate sets make the competing answer identities explicit; accuracy and robustness to rewording assess the resulting commitments.

4.1 Four Native Measurements

A native measurement forms an answer reading directly from the model's next-token probabilities, without training any additional parameters. We construct four such measurements by varying the measurement unit and the measurement scale. The unit is either the supplied answer text (token level, T) or the complete answer (answer level, A). The scale is formed either from each candidate in isolation (separate reference, S) or from the full candidate set (candidate-set reference, C), giving the four measurements in Table 1.

Table 1: **Four native measurements.** T_S and A_C lie at opposite corners, while T_C and A_S each align one of the two properties.

	Scale	
Unit	Separate ($\boldsymbol{S}$)	Set ($\boldsymbol{C}$)
Token ($\boldsymbol{T}$)	$\boldsymbol{T_S}$	$\boldsymbol{T_C}$
Answer ($\boldsymbol{A}$)	$\boldsymbol{A_S}$	$\boldsymbol{A_C}$

For question q, let $\mathcal{C}_q = \{a_1, \ldots, a_4\}$ be the candidate answers and y_i the supplied text of a_i. A token-level measurement scores y_i by its mean token log probability after a prefix x (Equation 19).

T_S: token unit, separate reference. The prefix $x_{T_S}(q)$ consists of the question, an instruction to give the complete answer text, and an answer cue. We score each candidate text in its own continuation after this prefix (Robinson et al., 2023).

T_C: token unit, candidate-set reference. The prefix $x^\pi_{T_C}(q, \mathcal{C}_q)$ additionally lists all candidates in order π and asks for the complete answer text (Cho et al., 2025). With the full set available before generation, we score each candidate in its own continuation and average over orders.

A_S: answer unit, separate reference. The prefix $x_{A_S}(q, a_i)$ presents a_i and asks whether an independently generated answer to q would express the same answer. This recurrence query asks about the answer event $\phi_q(Y) = a_i$, whose target probability is $m_q(a_i)$. We use the log-probability difference between `YES` and `NO` as its support reading.

A_C: answer unit, candidate-set reference. The prefix $x^{\pi}_{A_C}(q, \mathcal{C}_q)$ lists all four answers, assigns each a single-token label, and asks for one label (Robinson et al., 2023). In both A_S and A_C, a single output token stands for a complete answer. The order set Π contains four cyclic orders, and readings are aligned by answer identity before averaging (Pezeshkpour & Hruschka, 2024). Each answer appears once at each position. In A_C, the four labels compete at a single output position. Each measurement commits to the candidate with the largest reading. The scoring formulas and prompt templates are given in Appendix B.

4.2 Answering Behavior under Four Measurements

We evaluate four frozen pretrained models, Qwen2.5-7B, Qwen2.5-14B (Qwen Team, 2024), Llama-3.1-8B (Grattafiori et al., 2024), and Gemma-3-12B-PT (Gemma Team, 2025), on 300 ARC-C (Clark et al., 2018) and 300 MMLU (Hendrycks et al., 2021a) questions each, using plain-text prompts. All four interfaces use the same questions for each model, and we report accuracy (%) with 95% bootstrap confidence intervals. Evaluation details appear in Appendix B.

Table 2: **Answering accuracy under four native measurements.** Aligning either the unit or the scale improves accuracy over T_S in every setting, and aligning both (A_C) gives the highest accuracy.

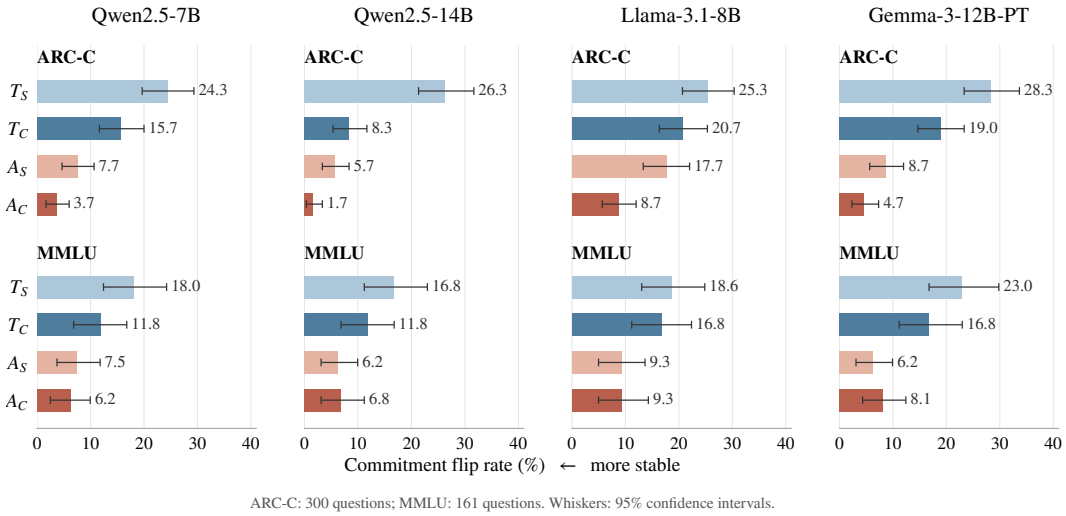
Model	T_S	T_C	A_S	A_C	Gap (pp)
Measurement unit	×	×	✓	✓	$A_C - T_S$
Measurement scale	×	✓	×	✓	
ARC-C					
Qwen2.5-7B	54.67 [49.0, 60.3]	81.33 [77.0, 85.7]	85.00 [80.7, 89.0]	**87.33** [83.7, 91.0]	+32.7
Qwen2.5-14B	62.33 [56.7, 67.7]	89.00 [85.3, 92.3]	88.00 [84.3, 91.7]	**91.67** [88.3, 94.7]	+29.3
Llama-3.1-8B	57.67 [52.0, 63.3]	65.67 [60.3, 71.0]	67.33 [62.0, 72.7]	**76.00** [71.0, 80.7]	+18.3
Gemma-3-12B PT	63.00 [57.7, 68.3]	77.00 [72.3, 81.7]	80.33 [75.7, 84.7]	**87.67** [83.7, 91.3]	+24.7
MMLU					
Qwen2.5-7B	44.33 [38.7, 50.0]	59.67 [54.0, 65.3]	60.00 [54.3, 65.3]	**68.67** [63.3, 73.7]	+24.3
Qwen2.5-14B	49.00 [43.3, 54.7]	66.33 [61.0, 71.7]	65.33 [60.0, 70.7]	**76.67** [71.7, 81.3]	+27.7
Llama-3.1-8B	44.33 [38.7, 50.0]	44.67 [39.0, 50.3]	46.67 [41.0, 52.3]	**62.67** [57.0, 68.0]	+18.3
Gemma-3-12B PT	50.33 [44.7, 56.0]	52.67 [47.0, 58.3]	56.33 [50.7, 61.7]	**73.00** [68.0, 78.0]	+22.7

Table 2 shows that aligning either the unit ($T_S \to A_S$) or the scale ($T_S \to T_C$) improves accuracy in every model–dataset setting, and A_C, which aligns both, is the most accurate throughout. Therefore, we attribute the higher accuracy of aligned measurements to a more faithful translation of answer support into commitments.

4.3 Stability under Answer Rewording

We test whether measurements preserve their selections when answer meaning stays fixed. Equation 6 separates answer support from the probability of a particular expression. Meaning-preserving

rewording therefore tests whether commitments remain stable as that expression changes. We replace the correct candidate with one fixed, reviewed, meaning-preserving rewording, keeping the question, other candidates, and prompt template unchanged. For example, `Bubbles appeared on the eggshell.` becomes `Bubbles formed on the eggshell.` Questions and rewrites are shared across all models and measurements: 300 questions from ARC-C and 161 from MMLU that admit local wording edits. We report the flip rate, the fraction of questions whose selected answer identity changes after rewording (Appendix B.3).



ARC-C: 300 questions; MMLU: 161 questions. Whiskers: 95% confidence intervals.

Figure 1: **Commitment flip rates under answer rewording (lower is more stable).** Aligning either the unit or the scale stabilizes commitments, and A_C has the lowest or joint-lowest flip rate in most settings.

As shown in Figure 1, aligning the unit ($T_S \to A_S$) reduces flips in every model–dataset setting, and aligning the scale ($T_S \to T_C$) also reduces them throughout. A_C, which aligns both, has the lowest or joint-lowest flip rate in most settings. Therefore, we attribute the reduced sensitivity to meaning-preserving rewording to answer-level units and common references, which stabilize commitments across expressions of the same answer.

5 Learning a Shared Answer Measure

5.1 Recovering Answer Support from Frozen Representations

We test shared linear readouts of frozen T_S, T_C, and A_S representations, following Section 3.4. For each interface, we keep the pretrained model and its forward inputs fixed and replace only the native score with a learned answer-level reading. This tests whether the representations already computed for each candidate support a more effective commitment.

From representations to an answer measure. For each measurement $r \in \{T_S, T_C, A_S\}$ and candidate a_i, we take a representation $h_r(q, a_i)$ at a selected layer: the answer-final state for T_S and T_C, averaged over the four cyclic orders for T_C, and the state after appending `YES` to the recurrence prompt for A_S (Appendix B.4). A linear map shared across candidates defines the answer measure

$$s_{w,r}(q, a_i) = w_r^\top h_r(q, a_i), \qquad \widehat{\mu}_{w,r}(\{a_i\} \mid q) = \frac{\exp s_{w,r}(q, a_i)}{\sum_j \exp s_{w,r}(q, a_j)}, \tag{12}$$

and the model commits to $\arg\max_i \widehat{\mu}_{w,r}(\{a_i\} \mid q)$. The map assigns one reading to each complete answer and uses the same learned function across candidates. Learning from a common reference ordering brings separately formed representations onto a shared answer-support scale.

Table 3: **Native measurements and their learned shared maps.** A shared linear map over frozen representations improves the native measurement in nearly every setting, with the largest gains for the token-level measurements.

Model	T_S		T_C		A_S		A_C
	Native	Map	Native	Map	Native	Map	Native
ARC-C							
Qwen2.5-7B	54.67	**80.93** ± 2.18	81.33	**86.60** ± 0.49	**85.00**	83.40 ± 0.80	87.33
Qwen2.5-14B	62.33	**85.00** ± 0.94	89.00	**92.20** ± 0.45	88.00	**88.93** ± 0.60	91.67
Llama-3.1-8B	57.67	**68.80** ± 1.85	65.67	**73.87** ± 1.10	67.33	**68.20** ± 1.12	76.00
Gemma-3-12B PT	63.00	**86.40** ± 0.64	77.00	**87.13** ± 1.26	80.33	**82.33** ± 1.03	87.67
MMLU							
Qwen2.5-7B	44.33	**58.60** ± 1.32	59.67	**65.27** ± 1.62	60.00	**61.07** ± 0.72	68.67
Qwen2.5-14B	49.00	**62.00** ± 0.33	66.33	**70.53** ± 1.35	65.33	**67.20** ± 0.90	76.67
Llama-3.1-8B	44.33	**49.67** ± 0.94	44.67	**55.47** ± 1.46	46.67	**49.00** ± 2.47	62.67
Gemma-3-12B PT	50.33	**62.53** ± 1.12	52.67	**66.20** ± 1.26	56.33	**61.53** ± 1.66	73.00

Learning and evaluation. We use balanced A_C ordering as supervision, justified by its support-order preservation under the measurement model of Proposition A.1. Each training question contributes its highest- and lowest-scoring candidates under A_C; their representation difference and reference ordering form a pairwise training example. We fit all maps by regularized pairwise logistic regression (Hastie et al., 2009), using 300 training and 300 validation questions per dataset for each of five random seeds defining independently sampled training/validation splits. Map accuracy (%) is reported as mean ± standard deviation across the five seeds, and native accuracies are shared with Table 2. For each seed, layer and regularization selection maximize agreement with A_C orderings on validation questions. A_C supplies reference orderings during training and validation; at test time, the learned map ranks candidates directly from their representations under the corresponding interface, without computing A_C readings. All maps and native measurements use the same fixed 300 test questions per dataset as Table 2. Fitting details appear in Appendix B.5.

Support-supervised maps improve native commitments. As shown in Table 3, the learned map improves on its native measurement in nearly every setting, with the largest gains for the token-level measurements. Therefore, we attribute these gains to support comparisons from the common A_C reference, which train more effective readings of frozen candidate representations and narrow the knowledge–behavior gap.

5.2 Learning the Shared Mapping from Incorrect Answers

Does learning the shared map require correct training candidates? We test this by retaining the A_C reference and training exclusively on its orderings between pairs of incorrect answers.

Learning from incorrect answers. We use T_S representations from Qwen2.5-7B and Qwen2.5-14B. For each training question, we exclude the correct option using the gold label and select, among the remaining answers, those with the highest and lowest A_C readings; their ordering labels a shared logistic-regression map. We reuse the seeds and data splits from Section 5.1 and report mean ± standard deviation across seeds. Layer and regularization selection use support orderings among incorrect validation answers. Evaluation includes all candidates of each test question.

Incorrect answers suffice to learn the shared mapping. The learned map improves on native T_S in all four settings (Table 4). Therefore, we attribute the improvement to support comparisons among incorrect answers, which teach a transferable readout that improves selection among all candidates, including correct answers, on new questions.

Table 4: **Learning from incorrect answers.** A map trained only on incorrect answers improves on native T_S in every setting.

Dataset	Native T_S	**Wrong-only map**
Qwen2.5-7B		
ARC-C	54.67	**75.93** $\pm$ 1.01
MMLU	44.33	**55.13** $\pm$ 1.88
Qwen2.5-14B		
ARC-C	62.33	**78.40** $\pm$ 2.35
MMLU	49.00	**59.93** $\pm$ 0.68

Table 5: **Map-guided free generation.** Map-guided regeneration improves accuracy in every setting.

Dataset	Initial	**Map-guided**
Llama-3.1-8B-Instruct		
GSM8K	78.00 $\pm$ 4.14	**82.00** $\pm$ 3.84
MATH500	37.00 $\pm$ 4.82	**41.00** $\pm$ 4.90
DeepSeek-Math-7B-Instruct		
GSM8K	73.00 $\pm$ 4.43	**78.00** $\pm$ 4.15
MATH500	38.00 $\pm$ 4.86	**39.00** $\pm$ 4.89

5.3 Guiding Free Generation with the Answer Measure

Sections 5.1 and 5.2 read answer support among the given candidates of multiple-choice questions. We extend the shared map to free generation, where no candidate set is given, and evaluate it with Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and DeepSeek-Math-7B-Instruct (Shao et al., 2024) on GSM8K (Cobbe et al., 2021) and MATH500 (Hendrycks et al., 2021b; Lightman et al., 2024). We use disjoint splits of 500 and 400 training questions, respectively, and 300 evaluation questions per dataset. Each model learns its own map, and we report mean $\pm$ standard deviation across random seeds.

Without a candidate set, answer support must be estimated from the model's own samples. For each training question, we sample 64 responses, group them by answer identity $\phi_q(y)$, and use the group frequencies as Monte Carlo estimates of answer support (Wang et al., 2023). This support ordering supervises a shared linear map $g_w(h) = w^\top h$, fitted to answer-group mean representations. At test time, the map scores the answer-final representation of each initial response $y^{(0)}$; the lowest-scoring 25% of responses are regenerated and the rest are kept. For each selected response, we backpropagate the map score through its replayed trajectory to the hidden states and apply the resulting score-increasing perturbation while generating a fresh response from the original question (Dathathri et al., 2020). Appendix B.6 gives the gradient formulation and intervention positions.

The answer measure improves free generation. As shown in Table 5, map-guided regeneration improves accuracy on both datasets for both models. The map is learned from the model's own samples without a multiple-choice reference, and it acts only on the responses it scores as poorly supported. Therefore, we attribute the improvement to answer-level feedback that steers generation toward better-supported answers, which extends the shared map from candidate selection to open-ended answering.

6 Conclusion

We develop a measurement theory of answer support that connects measurement unit and scale to answer commitments. The theory establishes order preservation for balanced answer-label readings under a separable model and the existence of a shared support map under a representation measurability premise. Native-measurement experiments on ARC-C and MMLU show that adjusting these properties improves answer accuracy and stability under rewording. Shared-map experiments further show that answer representations from separate trajectories can support more effective answer-level measurement on a common scale, and that support relations between incorrect answers suffice to learn such a map. Using the learned measure for feedback control connects answer-level measurement to improved generation behavior. Together, these results show that whether knowledge is expressed as answering capability depends on how internal answer support is measured and converted into commitments. Improving this conversion provides a systematic path to realizing a model's existing answering capability.

AI USE STATEMENT

Generative AI tools from OpenAI (ChatGPT/Codex) assisted with experimental code, technical discussion, and manuscript revision, including English-language editing. The authors take responsibility for the final manuscript, experimental procedures, reported results, and artifacts developed with AI assistance.

ETHICS STATEMENT

This work studies answer measurement in publicly released language models from the Qwen2.5, Llama-3.1, Gemma-3, and DeepSeek-Math families using public benchmarks: ARC-Challenge, MMLU, GSM8K, and MATH500. The experiments concern model outputs and internal representations. Better understanding how measurement affects answer selection can support more reliable evaluation and use of language models. Answer support reflects a model's preferences, which can include incorrect or biased answers; practical applications should therefore assess both factual accuracy and task-specific safety. The activation-guidance procedure requires white-box access and can also be used to favor undesirable outputs. Its deployment warrants evaluation of the chosen feedback objective and the resulting behavior.

REPRODUCIBILITY STATEMENT

Section 3 defines answer support, measurement units and scales, and shared representation-level readouts. Appendix A provides the probability decomposition and proofs for balanced label measurements and shared maps, together with their assumptions and residual-error analysis. Sections 4 and 5 describe the experimental comparisons. Appendix B specifies the prompt templates, cyclic option orders, rewording protocol, representation extraction, and shared-map fitting procedure. In particular, Appendix B.5 records the five random seeds, training/validation splits, fixed test sets, layer and regularization selection, and aggregation of results across seeds. Appendix B.6 describes the gradient feedback and intervention positions used for free generation. The models and benchmark datasets used in the experiments are publicly available. The supplementary material provides anonymized core implementation excerpts for answer measurement and shared-map fitting, together with execution-log excerpts and per-seed results. We will release the complete experimental code and logs publicly upon publication.

REFERENCES

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023. URL https://aclanthology.org/2023.findings-emnlp.68/.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003. URL https://www.jmlr.org/papers/v3/bengio03a.html.

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

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.

Gyeongje Cho, Yeonkyoung So, and Jaejin Lee. Choices speak louder than questions. *arXiv preprint arXiv:2502.18798*, 2025. URL https://arxiv.org/abs/2502.18798.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1912.02164.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7 (1):1–26, 1979. doi: 10.1214/aos/1176344552. URL https://doi.org/10.1214/aos/1176344552.

Zorik Gekhman et al. Inside-out: Hidden factual knowledge in LLMs. *arXiv preprint arXiv:2503.15299*, 2025.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2 edition, 2009. doi: 10.1007/978-0-387-84858-7. URL https://doi.org/10.1007/978-0-387-84858-7.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://arxiv.org/abs/2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021b. URL https://arxiv.org/abs/2103.03874.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7038–7051. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.564. URL https://aclanthology.org/2021.emnlp-main.564/.

Jennifer Hu and Roger Levy. Prompting is not a substitute for probability measurements in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://aclanthology.org/2023.emnlp-main.306/.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Olav Kallenberg. *Foundations of Modern Probability*. Springer, 2 edition, 2002. doi: 10.1007/978-1-4757-4015-8. URL https://doi.org/10.1007/978-1-4757-4015-8.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2306.03341.

Xiang Lisa Li, Vaishnavi Shrivastava, Siyan Li, Tatsunori Hashimoto, and Percy Liang. Benchmarking and improving generator-validator consistency of language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=phBS6YpTzC.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.20050.

Kevin Liu, Stephen Casper, Dylan Hadfield-Menell, and Jacob Andreas. Cognitive dissonance: Why do language model outputs disagree with internal representations of truthfulness? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://aclanthology.org/2023.emnlp-main.291/.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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 *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.02707.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 2463–2473. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250/.

Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2006–2017. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-naacl.130. URL https://aclanthology.org/2024.findings-naacl.130/.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. Leveraging large language models for multiple choice question answering. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.12353.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2203.11171.

Sarah Wiegreffe, Matthew Finlayson, Oyvind Tafjord, Peter Clark, and Ashish Sabharwal. Increasing probability mass on answer choices does not always improve accuracy. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *International Conference on Learning Representations*, 2024.

A Supporting Derivations

This appendix derives the answer-support decomposition in Section 3.2, establishes order preservation under a separable label-measurement model, and proves the shared-map proposition in Section 3.4.

A.1 Answer Support and Text Probability

Fix a question q and the answer map ϕ_q from Section 3. For a complete continuation y with $P_q(Y = y) > 0$ and $\phi_q(y) = a$, define

$$\rho_q(y \mid a) = P_q(Y = y \mid \phi_q(Y) = a).$$

Since $\{Y = y\}$ is contained in $\{\phi_q(Y) = a\}$, the definition of conditional probability gives $P_q(Y = y) = m_q(a)\rho_q(y \mid a)$. Taking logarithms yields Equation 6:

$$\log P_q(Y = y) = \log m_q(a) + \log \rho_q(y \mid a). \tag{13}$$

Consequently, for two positive-probability continuations $y^{(a)}$ and $y^{(b)}$ expressing answers a and b, respectively,

$$\log \frac{P_q(Y = y^{(a)})}{P_q(Y = y^{(b)})} = \log \frac{m_q(a)}{m_q(b)} + \log \frac{\rho_q(y^{(a)} \mid a)}{\rho_q(y^{(b)} \mid b)}. \tag{14}$$

The expression-dependent term can change the ordering induced by answer support.

Length normalization. If L_a is the token length of $y^{(a)}$, its mean log probability is

$$\frac{\log P_q(Y = y^{(a)})}{L_a} = \frac{\log m_q(a)}{L_a} + \frac{\log \rho_q(y^{(a)} \mid a)}{L_a}. \tag{15}$$

Length normalization rescales both terms, so the expression-dependent term remains. In addition, different lengths rescale answer support by different factors.

An equal-length example. Consider a continuation space containing three texts of the same token length, y_1, y_2, y_3, with probabilities $0.30, 0.30, 0.40$. Let $\phi_q(y_1) = \phi_q(y_2) = a$ and $\phi_q(y_3) = b$. The answer-support measure assigns $m_q(a) = 0.60$ and $m_q(b) = 0.40$. Each individual text expressing a nevertheless has lower probability than the text expressing b. Comparing either y_1 or y_2 with y_3 therefore selects b under both summed and mean log probability, while the answer-support measure selects a. Equal token lengths isolate the effect of distributing one answer's support across multiple expressions.

Text prefixes and answer labels. For a fixed interface prompt x, let P_x denote the continuation measure, with trajectories completed under its stopping rule. A prefix u defines the event $[u] = \{y : y_{1:|u|} = u\}$, whose probability is

$$P_x([u]) = \sum_{y \in [u]} P_x(Y = y).$$

The token products used to score supplied answer text measure this prefix event; including termination gives the complete-trajectory event used in Equation 13. In the answer-label protocol, commitment is determined by the first label L. Thus $\{L = \lambda_i\} = [\lambda_i]$ is the event of committing to a_i, and its probability already sums all subsequent trajectories following that commitment. A single-token label therefore has an answer-level unit. The measurement model below connects this protocol's commitment probabilities to the target support m_q.

A.2 Order Preservation under Balanced Label Measurements

Fix a question q and a candidate set $\mathcal{C}_q = \{a_1, \ldots, a_n\}$ with $m_q(a_i) > 0$ for every i. For an order π, let $k_\pi(i)$ be the position of a_i and $\lambda_\pi(i)$ its assigned label. Write

$$\ell^\pi_{q,i} = \log p_\theta\big(\lambda_\pi(i) \mid x^\pi_{A_C}(q, \mathcal{C}_q)\big)\,.$$

An order set Π is balanced if each answer occupies each position equally often. The four cyclic orders used in Equation 21 satisfy this condition for $n = 4$.

Separable measurement assumption. Assume that, for every candidate and order,

$$\ell^{\pi}_{q,i} = \log m_q(a_i) + c^{\pi}_q + b_q(k_\pi(i)). \tag{16}$$

Here c^{π}_q is shared across candidates and may depend on the fixed candidate set and its presentation order, including the output normalization. The position term $b_q(k)$ depends on the question and position, and is shared across answers occupying that position. This assumption links the label interface to the answer-support measure defined in Section 3.

Equation 16 combines answer support $m_q(a_i)$ with multiplicative position factors $\exp b_q(k_\pi(i))$. Under this assumption, the candidate-normalized label probabilities satisfy

$$\frac{\exp \ell^{\pi}_{q,i}}{\sum_j \exp \ell^{\pi}_{q,j}} = \frac{m_q(a_i) \exp b_q(k_\pi(i))}{\sum_j m_q(a_j) \exp b_q(k_\pi(j))}.$$

The model separates answer support from position effects; candidate interactions and answer-dependent recognition effects enter the residual formulation below.

Proposition A.1 (Order preservation under balanced label measurements)**.** *Under Equation 16 and a balanced order set* Π*, the answer-aligned mean log probability satisfies*

$$s_{A_C}(q, a_i) = \frac{1}{|\Pi|} \sum_{\pi \in \Pi} \ell^{\pi}_{q,i} = \log m_q(a_i) + K_q, \tag{17}$$

where K_q *is common to all candidates in the fixed set. Thus it satisfies Equation 7 with* $T_{A_C,q}(z) = z + K_q$ *and preserves the answer-support ordering, including ties.*

Proof. Averaging Equation 16 gives

$$s_{A_C}(q, a_i) = \log m_q(a_i) + \frac{1}{|\Pi|} \sum_{\pi \in \Pi} c^{\pi}_q + \frac{1}{n} \sum_{k=1}^{n} b_q(k).$$

Balance makes the final term identical for every candidate. The last two terms define K_q, so within-question score differences equal $\log[m_q(a_i)/m_q(a_j)]$, proving the claim. □

Normalizing the exponentiated balanced readings consequently gives

$$\frac{\exp s_{A_C}(q, a_i)}{\sum_j \exp s_{A_C}(q, a_j)} = \frac{m_q(a_i)}{\sum_j m_q(a_j)}.$$

Thus the same construction recovers the candidate-normalized support under the separable model, with the normalization preserving its ordering.

Bounded residual distortions. More generally, add a residual $e^{\pi}_{q,i}$ to the right-hand side of Equation 16, and define $\bar{e}_{q,i} = |\Pi|^{-1} \sum_{\pi \in \Pi} e^{\pi}_{q,i}$. The same averaging argument yields

$$s_{A_C}(q, a_i) - s_{A_C}(q, a_j) = \log \frac{m_q(a_i)}{m_q(a_j)} + \bar{e}_{q,i} - \bar{e}_{q,j}. \tag{18}$$

If $\max_i |\bar{e}_{q,i}| \le \varepsilon_q$, the error in each pairwise log-support contrast is at most $2\varepsilon_q$. Consequently, $\log[m_q(a_i)/m_q(a_j)] > 2\varepsilon_q$ implies $s_{A_C}(q, a_i) > s_{A_C}(q, a_j)$; a unique support maximizer is preserved when its log-support gap to every other candidate exceeds this bound. The residual includes candidate-specific context and recognition effects, as well as content–position interactions. Its answer-dependent average determines the remaining ordering distortion; the bound is a condition on these effects. The same ordering statements hold for averaged label logits, since logits and log probabilities differ by a shared constant within each order.

A.3 Proof of the Shared Answer-Support Map

Proof of Proposition 3.1. The random variable $M = m_Q(A)$ takes values in $[0, 1]$ and is $\sigma(H_r)$-measurable by assumption. The Doob–Dynkin factorization lemma (Kallenberg, 2002) therefore gives a measurable function $f^\star_r : \mathbb{R}^d \to [0, 1]$ satisfying $M = f^\star_r(H_r)$ almost surely. Since the

lemma is applied to the joint distribution of (Q, A), this single function is shared across questions and candidates, proving Equation 9.

It remains to prove Equation 10. Condition on $Q = q$. Since A is uniform on the finite set $\mathcal{C}_q$, the conditional probability of a failed factorization is

$$\frac{1}{|\mathcal{C}_q|} \sum_{a \in \mathcal{C}_q} \mathbf{1}\{f_r^\star(h_r(q,a)) \neq m_q(a)\}\,.$$

By Equation 9, its expectation over Q is zero, so this nonnegative quantity is zero for almost every question. Thus, the same factorization holds for all candidates of each such question. For candidates with positive support, taking logarithms gives $g_r^\star(h_r(q,a)) = \log m_q(a)$. Subtracting this identity for a_i and a_j gives Equation 10, which completes the proof. □

B Experimental Interface Constructions

This appendix specifies the prompts and measurement positions used for the four native interfaces in Section 4.1, their option-order control, and the representation positions used by the shared maps in Section 5.1. Prompts are plain text. Braced fields below are replaced with the question and candidate texts; displayed blank lines are retained.

B.1 Native Prompts and Measurement Positions

Scoring rules. For question q, let $\mathcal{C}_q = \{a_1, \ldots, a_4\}$ be the candidate answers and y_i the supplied text of a_i. A token-level measurement scores y_i by its mean token log probability after a prefix x:

$$L_\theta(y \mid x) = \frac{1}{n(y;x)} \sum_{t=1}^{n(y;x)} \log p_\theta(y_t \mid x, y_{<t}), \tag{19}$$

where $n(y;x)$ is the number of tokens in y under the tokenization that follows x.

T_S: token unit, separate reference. The prefix $x_{T_S}(q)$ consists of the question, an instruction to give the complete answer text, and an answer cue. We score each candidate text in its own continuation after this prefix (Robinson et al., 2023).

T_C: token unit, candidate-set reference. The prefix $x^\pi_{T_C}(q, \mathcal{C}_q)$ additionally lists all candidates in order π and asks for the complete answer text (Cho et al., 2025). With the full set available before generation, we score each candidate in its own continuation and average over orders:

$$\begin{aligned} s_{T_S}(q, a_i) &= L_\theta(y_i \mid x_{T_S}(q)), \\ s_{T_C}(q, a_i) &= \frac{1}{|\Pi|} \sum_{\pi \in \Pi} L_\theta(y_i \mid x^\pi_{T_C}(q, \mathcal{C}_q)). \end{aligned} \tag{20}$$

A_S: answer unit, separate reference. The prefix $x_{A_S}(q, a_i)$ presents a_i and asks whether an independently generated answer to q would express the same answer. This recurrence query asks about the answer event $\phi_q(Y) = a_i$, whose target probability is $m_q(a_i)$. We use the log-probability difference between `YES` and `NO` as its support reading.

A_C: answer unit, candidate-set reference. The prefix $x^\pi_{A_C}(q, \mathcal{C}_q)$ lists all four answers, assigns each a single-token label, and asks for one label (Robinson et al., 2023). In both A_S and A_C, a single output token stands for a complete answer. With $\lambda_\pi(i)$ the label of a_i under order π, the two answer-level measurements are

$$\begin{aligned} s_{A_S}(q, a_i) &= \log p_\theta(\texttt{YES} \mid x_{A_S}(q, a_i)) - \log p_\theta(\texttt{NO} \mid x_{A_S}(q, a_i)), \\ s_{A_C}(q, a_i) &= \frac{1}{|\Pi|} \sum_{\pi \in \Pi} \log p_\theta(\lambda_\pi(i) \mid x^\pi_{A_C}(q, \mathcal{C}_q)). \end{aligned} \tag{21}$$

The order set Π contains four cyclic orders, and readings are aligned by answer identity before averaging (Pezeshkpour & Hruschka, 2024). In A_C, the four labels compete at a single output position. Each measurement commits to the candidate with the largest reading.

Numerical precision and aggregation. The native-interface evaluations in Section 4.2 use FP32 inference with TF32 disabled. MMLU accuracy is question-weighted across the 57 represented subjects. The 95% confidence intervals in Table 2 and Figure 1 use the 2.5th and 97.5th percentiles from 20,000 bootstrap resamples of questions (Efron, 1979).

T_S: token unit, separate reference. Each candidate is evaluated after the following prefix:

```
Question: {question}

Answer with the complete answer text.
Answer:
```

We append one space followed by the candidate text and measure its mean token log probability, as defined in Equation 19. Each token is conditioned on the prefix and the preceding candidate tokens.

T_C: token unit, candidate-set reference. For one presentation order, the prefix is:

```
Question: {question}
A. {option at position 1}
B. {option at position 2}
C. {option at position 3}
D. {option at position 4}

Answer with the complete answer text.
Answer:
```

We append one space followed by each candidate text in turn. The measured continuation is the candidate text following the final `Answer:`; the displayed options supply its context.

A_S: answer unit, separate reference. Each candidate is inserted into this prompt:

```
Question: {question}

Proposed answer: {candidate}
If you independently generated your own answer to the original question,
would it express the same answer as the proposed answer?
Answer only YES or NO.
Answer:
```

At the next output position, we read the single-token continuations `YES` and `NO`. Their log-probability difference gives the candidate's native measurement.

A_C: answer unit, candidate-set reference. For the same presentation order as T_C, we use:

```
Question: {question}
A. {option at position 1}
B. {option at position 2}
C. {option at position 3}
D. {option at position 4}

Answer with the option letter.
Answer:
```

At the next output position, we read the four single-token label continuations `A`, `B`, `C`, and `D`. Each label's reading is assigned to the answer occupying that position.

Logit-based computation. Both answer-level measurements are computed from logits. The `YES`–`NO` logit difference equals the log-probability difference. Within each A_C order, label logits differ from log probabilities by a normalizer shared across labels, so averaging either after answer-identity alignment yields the same ranking.

B.2 Cyclic Orders and Answer Identity

For candidates (a_1, a_2, a_3, a_4), both T_C and A_C use

$$\Pi = \{(1,2,3,4),\ (2,3,4,1),\ (3,4,1,2),\ (4,1,2,3)\}.$$

The position labels remain `A`, `B`, `C`, and `D`, while candidate texts rotate. Thus each answer appears once at each position. For example, a_1 receives labels `A`, `D`, `C`, and `B` across these orders.

For T_C, we average the four mean token log probabilities of the same answer text. For A_C, we average the readings of that answer's assigned label in each order. Both averages are taken after aligning by answer identity, as in Equations 20 and 21.

B.3 Answer-Rewording Protocol

Section 4.3 replaces the correct candidate with one fixed, reviewed, meaning-preserving rewording. For example, `Bubbles appeared on the eggshell.` becomes `Bubbles formed on the eggshell.` We keep the question and the other three candidates unchanged. ARC-C uses all 300 evaluation questions, and MMLU uses the 161 questions that admit local wording edits. Questions and rewrites are shared across models and measurements; each measurement's prompt template is held fixed within a before–after pair. In T_C, the rewording replaces the candidate in both the list and the scored continuation, and we recompute all candidate readings.

For measurement r, let $\widehat{a}_r(q)$ and $\widehat{a}'_r(q)$ be the answer identities selected before and after rewording. The flip rate is

$$F_r = \frac{1}{N} \sum_{q=1}^{N} \mathbf{1}\{\widehat{a}_r(q) \neq \widehat{a}'_r(q)\}\,. \tag{22}$$

It counts every change of selected answer, including changes between two incorrect candidates.

B.4 Candidate Representation Positions

For the shared maps, each candidate contributes one hidden vector at the selected layer. For T_S, the vector is read after processing the final token of the supplied answer text. Its candidate encoding is:

```
Question: {question}

Answer with the complete answer text.
Answer: {candidate}
```

For T_C, we read the final candidate-text token after the candidate-set prefix above and average the four vectors for the same answer across cyclic orders. For A_S, we append `YES` to the proposal prompt and read the state at that token. The native A_S measurement uses the output distribution before this token; the learned map uses its processed hidden state. These vectors are the inputs $h_r(q, a_i)$ in Equation 12.

B.5 Shared-Map Fitting

Sections 5.1 and 5.2 use a fixed pool of 600 non-test questions per dataset and seeds 202609260–202609264. Each seed independently samples 300 training questions without replacement and uses the remaining 300 for validation. The splits are shared across models, interfaces, and the standard and wrong-only conditions. All runs are evaluated on the same 300 held-out test questions per dataset used for the native measurements.

We fit L_2-regularized pairwise logistic regression for every map (Hastie et al., 2009). Each training question contributes the highest- and lowest-scoring candidates under the four-order-averaged A_C reference. We standardize features using these training vectors and fit their differences in both directions with opposite binary labels and no intercept. In the wrong-only condition, the gold answer is excluded before selecting each pair. The resulting linear score is applied to every test candidate.

For each seed, the search covers every Transformer layer and the final normalized state, with inverse regularization strengths $C \in \{0.001, 0.01, 0.1\}$. Selection maximizes validation pair-order agreement

with A_C; ties are resolved by top-choice agreement, then smaller C, then earlier layer. Wrong-only validation compares only incorrect candidates. We use the `liblinear` solver with a 5,000-iteration limit and tolerance 10^{-6}. Tables 3 and 4 report the mean and sample standard deviation of test accuracy across the five independently sampled training sets, with layer and regularization selected separately for each seed.

B.6 Free-Generation Implementation

For a response selected for regeneration in Section 5.3, let $H_{q,y^{(0)}}(\delta h)$ denote its answer-final representation when replaying the initial trajectory with perturbation δh. With the linear map $g_w(h) = w^\top h$, the feedback and its gradient are

$$\begin{aligned} S_q(\delta h) &= g_w\big(H_{q,y^{(0)}}(\delta h)\big)\,, \\ \nabla_{\delta h} S_q(\delta h) &= J_{H_{q,y^{(0)}}}(\delta h)^\top w. \end{aligned} \tag{23}$$

We use this gradient to raise the feedback score, then generate a fresh response from the original question with the resulting perturbation.

For Section 5.3, the perturbation is added at block 20 from the final prompt token onward, including subsequent generated tokens. The map reads the answer-final state at block 24. We repeat the full procedure across multiple random seeds; Table 5 reports the mean and standard deviation across seeds.