Do Correctness Probes Transfer? A Cross-Family Study of LLM Hidden States

Anonymous ACL submission

Abstract

Linear probes on language-model hidden states can predict output correctness, but it remains unclear whether they capture transferable correctness representations. We study out-of-distribution correctness-probe transfer across seven datasets, four task families, and three open-weight LLMs: Qwen3-8B, Llama-3.1-8B, and Gemma-2-9B, with all findings validated across five random seeds. Within-family transfer is consistently stronger than cross-family transfer (gap 0.075 AUC), with code generation showing the clearest pattern across MBPP-test, MBPP-train, and HumanEval. We further find that probe weight directions are nearly orthogonal across most dataset pairs despite strong predictive transfer, suggesting that probes do not encode a single canonical correctness direction. Comparing probes to logprob confidence baselines, we find probe advantage is concentrated in in-domain and within-family transfer cells. We also identify a layer-selection confound where early-layer shortcuts can appear transferable in-domain but fail cross-source. Overall, correctness probes transfer selectively rather than universally, making them most useful in-domain and within task families.

1 Introduction

Large language models exhibit substantial variation in correctness across domains: a model that performs strongly on grade-school arithmetic may perform much worse on competition mathematics, code generation, or factual recall (Zhang et al., 2024; Ding et al., 2024; Yang et al., 2025b). Knowing when a model is likely to be wrong has become essential for deployment, calibration, and downstream pipelines that depend on selective prediction (Liu et al., 2025). A growing body of work suggests that LLM hidden states encode correctness-related information, as linear probes frequently achieve strong in-domain discrimination when evaluated on the same dataset used for training (Kadavath et al., 2022; Azaria and Mitchell, 2023; Su et al., 2024; Ji et al., 2024).

Do correctness probes learn a general representation of correctness, or dataset-specific features? If correctness is general, probes should transfer across math, factual QA, code, and multiple-choice reasoning. If not, they are useful diagnostics but limited evidence about broader epistemic representations.

Prior work has shown that LLM internal states can predict answer correctness and, in recent work, that question-only correctness probes can generalize across some out-of-distribution knowledge datasets (Moreno Cencerrado et al., 2025; Zhang et al., 2025). However, to our knowledge, no prior work has constructed full source-target transfer matrices of response-level correctness probes spanning mathematical reasoning, multiple-choice reasoning, code synthesis, and factual recall across multiple open-weight model families.

This paper systematically studies out-of-distribution probe transfer across seven datasets and four task families: mathematical reasoning (GSM8K, MATH-500), multiple-choice reasoning (MMLU-Pro), code synthesis (MBPP-test, MBPP-train, HumanEval), and factual recall (TriviaQA). Using three open-weight instruction-tuned models from different families—Qwen3-8B, Llama-3.1-8B, and Gemma-2-9B—we train a linear probe for each source dataset and evaluate AUC on every target dataset, producing a 7×7 transfer matrix for each model. We further analyze probe direction similarity, logprob-based baselines, per-layer trajectories, and multi-seed stability. Our contributions are:

- **Transfer matrices.** We construct systematic source-target transfer matrices for response-level correctness probes across multiple open-weight LLM families and task domains, showing that correctness-predictive information is

linearly accessible across all evaluated settings.

- **Within-task vs. cross-task transfer.** We show that probe transfer is consistently stronger within task families than across them, with the clearest pattern emerging in code, where transfer among coding datasets is substantially stronger than code-to-non-code transfer.
- **Model-dependent math asymmetry.** We find that transfer between grade-school and competition-style mathematics is asymmetric and model dependent, suggesting that transfer reflects not only task structure but also model-specific factors such as accuracy distributions.
- **Orthogonal probe directions despite transfer.** We show that predictive transfer often occurs despite weak alignment of probe directions in weight space, suggesting that correctness probes do not encode a single universal correctness direction.
- **Layer-selection confound.** We identify a methodological confound in naive best-layer selection, where early-layer signals can capture dataset-specific difficulty rather than transferable correctness information, and propose a simple constraint to mitigate this issue.

2 Related Work

Probing language model representations. Probing is widely used to analyze what information is encoded in neural language model hidden states. (Alain and Bengio, 2016; Hewitt and Manning, 2019; Tenney et al., 2019; Jawahar et al., 2019). High probe accuracy does not imply that the model robustly represents the target concept, since probe capacity, dataset artifacts, and control-task performance can affect results (Hewitt and Liang, 2019; Pimentel et al., 2020; Voita and Titov, 2020; Belinkov, 2022).

Correctness and reliability probes. Recent work reveals LLM internal states contain signals related to correctness, truthfulness, hallucination risk, and self-verification (Kadavath et al., 2022; Azaria and Mitchell, 2023; Su et al., 2024; Ji et al., 2024; Zhang et al., 2025). Closest to our work, Moreno Cencerrado et al. (2025) train question-only probes before generation and find transfer across knowledge datasets, but weaker generalization on mathematical reasoning.

Dataset	Family	N
GSM8K	Math	1000
MATH-500	Math	500
MMLU-Pro	MC	1000
MBPP-test	Code	257
MBPP-train	Code	120
HumanEval	Code	164
TriviaQA	Factual	1000

Table 1: Datasets and task families. MC denotes multiple-choice; full descriptions, citations, and grading details are in Appendix D.

Out-of-distribution transfer and representation geometry. Prior work shows that representations become less transferable as task distance increases and features become more specialized (Yosinski et al., 2014; Mou et al., 2016; Liu et al., 2019). This motivates our analysis of whether correctness probes trained on different datasets learn aligned weight directions (Kornblith et al., 2019; Hewitt and Manning, 2019).

Confidence baselines. Confidence can also be estimated without hidden-state probing, using verbalized confidence, log probabilities, entropy, semantic uncertainty, or self-evaluation (Lin et al., 2022; Tian et al., 2023; Kuhn et al., 2023; Farquhar et al., 2024; Geng et al., 2024).

3 Experimental Setup

3.1 Models

We evaluate three open-weight instruction-tuned decoder-only models from different families: Qwen3-8B (Yang et al., 2025a), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Gemma-2-9B-it (Team et al., 2024). These models are chosen to provide architectural and training-data diversity within a comparable 7–9B parameter range. We use each model in standard inference mode with its default chat template and without system-prompt modification. All generations use greedy decoding (`do_sample=False`) with a maximum of 2048 new tokens. Model architecture details and HuggingFace identifiers are reported with the reproducibility details in Appendix I.

3.2 Datasets

We evaluate transfer across seven datasets spanning four task families, as summarized in Table 1. MBPP-test, MBPP-train, and HumanEval use their full available splits; the remaining datasets

use 500 or 1000 sampled test examples with `random_state=42`. Math and code include within-family pairs for transfer analysis, while MMLU-Pro and TriviaQA are singleton families. Dataset accuracy and correctness-label balance for each model-dataset pair are reported in Appendix D.

3.3 Prompts and Generation

Each dataset uses a tailored prompt format with an explicit output anchor to support deterministic answer extraction. GSM8K asks the model to solve step by step and end with "Final Answer:"; MATH-500 asks for a final answer in `\boxed{...}`; MMLU-Pro asks for reasoning followed by "Answer:"; MBPP-test and MBPP-train ask for only Python code in a Python block; HumanEval uses the same code prompt while preserving the required function signature; and TriviaQA asks for a final answer after the same "Final Answer:" anchor. All generations use greedy decoding (`do_sample=False`) with a maximum of 2048 new tokens. Full prompt templates are provided in Appendix F.

3.4 Correctness Labeling

Each generated answer is graded using dataset-specific automatic evaluators. Math tasks use multi-pattern answer extraction with symbolic equivalence checking when possible; MMLU-Pro uses constrained letter parsing; code tasks execute generated functions against unit tests in a sandboxed temporary directory with a 10-second timeout; and TriviaQA uses normalized alias matching against all provided answer aliases. To reduce the effect of remaining label noise or imbalance, probe training uses stratified cross-validation and class-balanced logistic regression. Additional labeling details, including answer extraction and code-execution rules, are provided in Appendix G.

4 Methodology

4.1 Probe Architecture and Training

For each model-dataset pair, we train a binary logistic regression probe on hidden states from a single layer. The probe predicts the probability that the model's generated answer is correct:

$$f_\theta(\mathbf{h}) = \sigma(\mathbf{w}^\top \mathbf{h} + b), \tag{1}$$

where $\mathbf{h} \in \mathbb{R}^d$ is a hidden state, $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are learned parameters, σ is the sigmoid function, and d is the model's hidden dimension. We train using scikit-learn's `LogisticRegression` with `class_weight='balanced'` to account for class imbalance across datasets, `C=1.0` for L2 regularization, and `max_iter=2000`. We use 5-fold stratified cross-validation with `random_state=42`, reporting mean AUC across folds. Each probe is trained on hidden states from a single layer per dataset, with layer selection described in Section 4.3.

4.2 Hidden State Extraction

We extract hidden states with a generate-then-replay procedure. First, the model generates up to 2048 tokens with greedy decoding while we record token-level log probabilities and entropies for baselines (Section 4.6); hidden states are not stored. Second, we replay the concatenated prompt and generation in a single forward pass with `output_hidden_states=True`.

From each layer, we extract four positions: end of question, middle of generation, pre-answer marker, and final generated token. If no answer marker is found, the pre-answer position falls back to the penultimate generated token. All reported results use the pre-answer-marker position. Hidden states are stored as `float16`, and layer indices refer to transformer block outputs excluding the embedding layer.

4.3 Layer Selection

We first selected probe layers by unconstrained cross-validation, which produced a Qwen3-8B anomaly: MATH-500 selected layer 2 despite other datasets peaking at layers 20–29. Cross-source probes at MATH-500 layer 2 often fell below chance. This suggests a dataset-specific shortcut rather than transferable correctness signal.

We therefore constrain best-layer selection to the upper half of each network:

$$\ell^*(\mathcal{D}) = \arg\max_{\ell \in \{\lfloor L/2 \rfloor, \ldots, L\}} \mathrm{AUC}_{\mathrm{CV}}(\mathcal{D}, \ell), \tag{2}$$

where L is the number of layers and $\mathrm{AUC}_{\mathrm{CV}}(\mathcal{D}, \ell)$ is 5-fold cross-validated AUC for dataset $\mathcal{D}$ at layer ℓ. This shifts Qwen3-8B MATH-500 from layer 2 to 29 with only a 0.009 AUC drop. Transfer uses dataset-specific constrained best layers; direction similarity uses shared cosine layers selected under the same constraint (Appendix B, Table B1).

4.4 Transfer Matrix Construction

For each model $\mathcal{M}$ and ordered pair of source and target datasets $(\mathcal{S}, \mathcal{T})$, we train a probe on source-

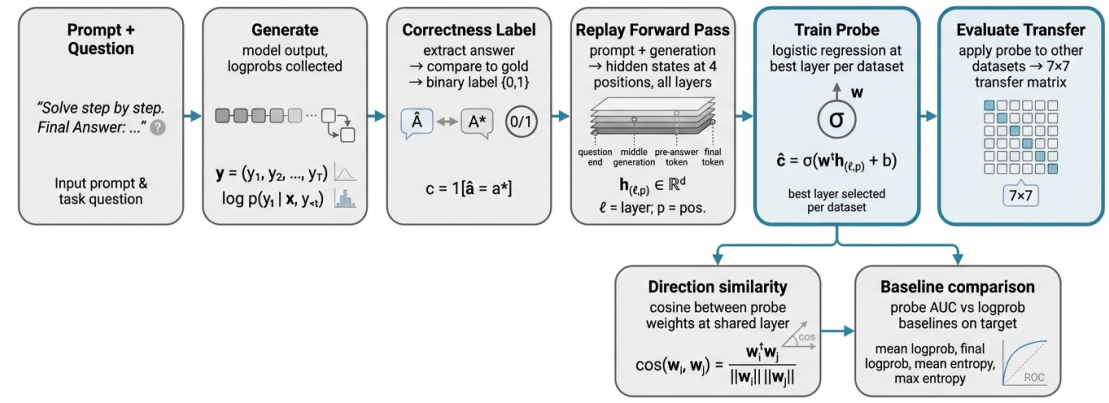


Figure 1: Experimental pipeline used to construct the transfer matrices and auxiliary analyses.

dataset activations at the source's constrained best layer $\ell^*(\mathcal{S})$, then evaluate that probe on target-dataset activations at the same layer:

$$M_{\mathcal{M}}[\mathcal{S},\mathcal{T}] = \mathrm{AUC}\left(f_{\theta_{\mathcal{S}}}\left(\mathbf{H}_{\mathcal{T}}^{(\ell^*(\mathcal{S}))}\right), \mathbf{y}_{\mathcal{T}}\right), \quad (3)$$

where $\theta_{\mathcal{S}}$ are probe parameters fitted on $\mathcal{S}$, $\mathbf{H}_{\mathcal{T}}^{(\ell^*(\mathcal{S}))}$ are target-dataset hidden states at layer $\ell^*(\mathcal{S})$, and $\mathbf{y}_{\mathcal{T}}$ are target-dataset correctness labels. For diagonal cells where $\mathcal{S} = \mathcal{T}$, we report the 5-fold cross-validated AUC from Section 4.1 rather than evaluating the probe on its own training data. This yields a 7×7 transfer matrix $M_{\mathcal{M}}$ per model. We average matrices across the three models to produce a cross-model transfer matrix.

4.5 Direction Similarity

To compare probe geometry, we compute cosine similarity between probe weight vectors. Since probes trained at different layers live in different representation spaces, we train an additional set of probes at a model-specific shared layer: 26 for Qwen3-8B, 16 for Llama-3.1-8B, and 41 for Gemma-2-9B. For probes with weights $\mathbf{w}_{\mathcal{A}}$ and $\mathbf{w}_{\mathcal{B}}$, we compute:

$$\cos(\mathbf{w}_{\mathcal{A}}, \mathbf{w}_{\mathcal{B}}) = \frac{\mathbf{w}_{\mathcal{A}}^{\top}\mathbf{w}_{\mathcal{B}}}{\|\mathbf{w}_{\mathcal{A}}\|_2 \|\mathbf{w}_{\mathcal{B}}\|_2}. \quad (4)$$

Values near 1 indicate aligned directions; values near 0 indicate orthogonal directions.

4.6 Baselines

We compare probe-based correctness predictions to four target-dataset logprob baselines derived from generation-time statistics (Section 4.2): mean log probability across generated tokens, log probability of the final generated token, mean entropy across generated tokens, and maximum entropy across generated tokens. For the entropy baselines, we report AUC of the negated entropy, as higher entropy corresponds to lower expected correctness. For each cell $(\mathcal{S},\mathcal{T})$ in the transfer matrix, we compute the probe-vs-baseline gap:

$$\Delta_{\mathcal{M}}[\mathcal{S},\mathcal{T}] = M_{\mathcal{M}}[\mathcal{S},\mathcal{T}] - \max_{b} \mathrm{AUC}(b, \mathbf{y}_{\mathcal{T}}), \quad (5)$$

where the maximum is taken over the four baselines b. Positive values indicate the probe outperforms the best target-side baseline; negative values indicate the baseline outperforms the probe.

4.7 Multi-Seed Robustness

We repeat probe training and evaluation with five seeds $\{42, 7, 13, 100, 0\}$, each controlling cross-validation splits and probe initialization. We report the per-cell mean and standard deviation across seeds.

5 Results

5.1 In-Domain Probe Performance

Linear probes extract above-chance correctness signals across all evaluated model-dataset combinations. In-domain AUCs in the averaged transfer matrix range from 0.692 on HumanEval to 0.868 on TriviaQA, with mean 0.791 across the seven datasets. In the averaged matrix, all diagonal AUCs exceed 0.69, confirming that correctness-relevant

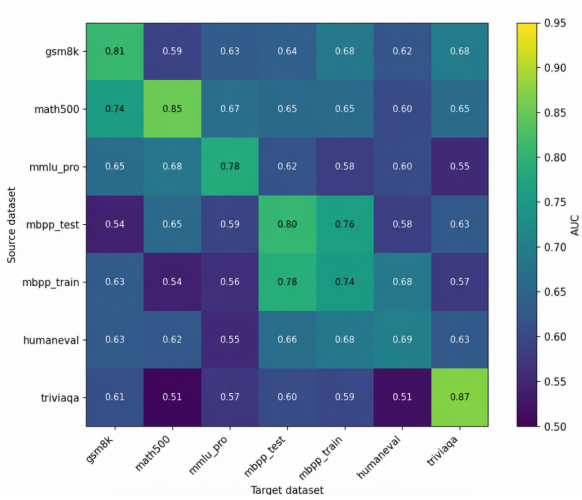


Figure 2: Averaged 7×7 probe transfer matrix. Rows are source datasets, columns are target datasets, and cells report mean AUC.

Dataset	Qwen3	Llama	Gemma	Avg.
GSM8K	0.770	0.827	0.818	0.805
MATH-500	0.852	0.849	0.850	0.850
MMLU-Pro	0.791	0.788	0.750	0.776
MBPP-test	0.796	0.798	0.808	0.801
MBPP-train	0.743	0.696	0.789	0.743
HumanEval	0.640	0.729	0.707	0.692
TriviaQA	0.842	0.945	0.818	0.868

Table 2: In-domain probe AUCs at constrained best layers. Values report 5-fold cross-validated AUC for probes trained and evaluated on the same dataset.

information is linearly accessible in hidden states across task domains (Table 2).

5.2 Cross-Dataset Transfer Patterns

Group	Mean AUC	Std.	Min	Max	Count
Cross-family	0.609	0.047	0.507	0.684	34
Diagonal	0.791	0.061	0.692	0.868	7
Within-family	0.684	0.073	0.584	0.778	8

Table 3: Within-family versus cross-family transfer summary.

Figure 2 shows the averaged 7×7 transfer matrix across the three models. Off-diagonal cells span a wide range (0.51–0.78), with mean 0.62. We observe several structural patterns.

Within-family transfer is consistently stronger than cross-family transfer. In the averaged 7×7 matrix, within-family off-diagonal cells have mean AUC 0.684, compared to 0.609 for cross-family off-diagonal cells, a gap of 0.075 (Table 3). This pattern is consistent in sign across all three models.

Code within-family transfer is the cleanest pattern. Pairwise transfer among MBPP-test, MBPP-train, and HumanEval averages 0.667 on Qwen3, 0.730 on Llama, and 0.671 on Gemma, compared to code-to-non-code transfer of 0.604, 0.627, and 0.557, respectively. The code-family within-vs-cross gap is 0.063 for Qwen3, 0.103 for Llama, and 0.114 for Gemma.

Math within-family transfer is highly asymmetric. The direction of the asymmetry varies across models. On Llama and Gemma, MATH-500-trained probes transfer better to GSM8K than the reverse direction: AUCs of 0.766 versus 0.577 on Llama (asymmetry +0.189), and 0.768 versus 0.407 on Gemma (asymmetry +0.361). On Qwen3, the asymmetry reverses: GSM8K-trained probes transfer to MATH-500 at 0.789, while MATH-500-trained probes transfer to GSM8K at 0.698 (asymmetry -0.091). The direction of asymmetry thus depends on the model rather than being a universal property.

Below-chance transfer appears in some individual model cells. This is especially visible for cells involving Gemma and MATH-500 as the target. These cells suggest that source-trained probes can become anti-predictive on certain target distributions, although the effect disappears in the averaged matrix.

5.3 Probe Direction Similarity

At each model's shared cosine layer, we compute pairwise cosine similarities between probe weight vectors. Results show that cosine similarity between probes from different datasets is near zero across most pairs, regardless of whether the probes transfer well predictively (Figure 3).

Qwen3-8B. All 21 off-diagonal cosines fall in $[-0.029, 0.086]$. The largest within-family value is 0.086 (MBPP-test ↔ MBPP-train); within-family math (GSM8K ↔ MATH-500) is only 0.016. Probes encode correctness through nearly orthogonal weight directions across all dataset pairs.

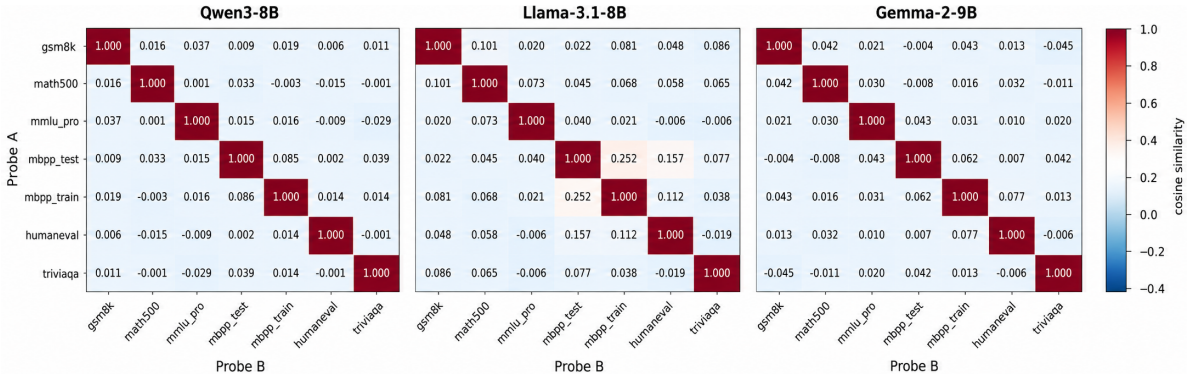


Figure 3: Probe direction cosine similarity matrices. Cells report pairwise cosine similarity between probe weight vectors trained at each model's shared cosine layer.

Summary type	Model	Average gap	Max gap	Cells with gaps ≥ 0.05	Valid cells
Overall	All	-0.0872	0.1332	14	147
Per-model	Gemma	-0.0973	0.1332	6	49
Per-model	Llama	-0.0740	0.1191	4	49
Per-model	Qwen3	-0.0902	0.1329	4	49

Table 4: Probe versus baseline gap summary. Gap = probe AUC − best logprob baseline AUC; values ≥ 0.05 favor probes. Full table in Appendix C.

Llama-3.1-8B. Off-diagonal cosines range from -0.019 to 0.252. The code family shows the clearest alignment: MBPP-test ↔ MBPP-train cosine 0.252, MBPP-test ↔ HumanEval cosine 0.157, and MBPP-train ↔ HumanEval cosine 0.112. The math pair (GSM8K ↔ MATH-500) is 0.101, while most remaining within- and cross-family pairs are near 0.10 or below.

Gemma-2-9B. Off-diagonal cosines range from -0.045 to 0.077. No within-family pair shows meaningful alignment; even MBPP-test ↔ MBPP-train reaches only 0.062. Probes are therefore nearly orthogonal across all off-diagonal dataset pairs.

Overall, predictive transfer can occur even when probe weight vectors are nearly orthogonal. For example, MATH-500 → GSM8K on Llama achieves AUC 0.766 despite cosine similarity of only 0.101. This suggests that predictive transfer and weight-space alignment are partially decoupled.

5.4 Comparison to Logprob Baselines

We compute the gap between probe AUC and the best target-side logprob baseline for each cell of the transfer matrix. Across all 147 cells, the average gap is -0.087 (Table 4), with per-model averages ranging from -0.074 (Llama) to -0.097 (Gemma).

The cells where probes outperform baselines show a clear structure. Probes exceed the best baseline by at least 0.05 AUC in 14 cells, concentrated in diagonal (in-domain) settings and within-family transfer. Notable examples include in-domain GSM8K and MMLU-Pro across all three models, code-family transfer such as Qwen3 MBPP-train → MBPP-test (+0.122), Qwen3 MBPP-test → MBPP-train (+0.091), and Gemma MBPP-train → MBPP-test (+0.096), and math transfer from MATH-500 → GSM8K on Llama (+0.058) and Gemma (+0.078). Most cross-family cells, by contrast, fall below their corresponding target-side baselines, reinforcing that probe advantages are concentrated in-domain and within task families.

5.5 Layer-Wise Trajectories

Figure 4 shows the per-layer in-domain AUC trajectory for each model-dataset pair. Three distinct patterns emerge.

Late-layer emergence. GSM8K, MMLU-Pro, MBPP-test, HumanEval, and TriviaQA exhibit late-layer emergence on most models. AUC rises from near chance at early layers to peak performance at the layer ranges identified as constrained best layers in Section 4.3.

Wide-plateau pattern. MATH-500 shows a wide plateau across all three models. MATH-500 in-domain AUC sits between 0.79 and 0.87 from layer 0 onward and remains roughly stable across the

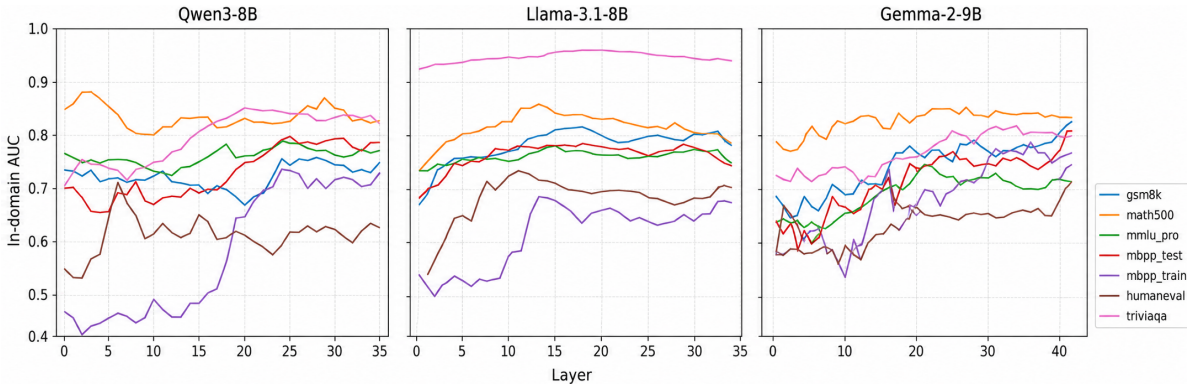


Figure 4: Per-layer in-domain probe AUC trajectories. Each panel shows in-domain probe AUC by layer for one model.

network. This pattern motivated the cross-source ablation in Section 4.3 that identified the input-difficulty confound at Qwen3 layer 2.

Late-layer steep emergence. MBPP-train exhibits late-layer steep emergence. Across all three models, MBPP-train in-domain AUC remains near chance through early and middle layers, then rises sharply in late layers. This pattern likely reflects the dataset's small sample size ($N = 120$) and class imbalance: probes require deep, compressed representations to extract signal from limited data.

5.6 Multi-Seed Stability

Transfer estimates are stable across five random seeds $\{42, 7, 13, 100, 0\}$. Average per-cell standard deviation is approximately 0.02 AUC across all three models, with the largest variance concentrated in cells involving MBPP-train, consistent with its small sample size. Full multi-seed statistics are reported in Appendix H.

6 Analysis

6.1 Transfer is Selective and Model-Dependent

The within-family advantage suggests that correctness signals are not fully domain-general. Instead, probes appear to transfer best when the source and target tasks share output structure, evaluation criteria, or reasoning format. This is clearest in code generation, where unit-test correctness, syntactic validity, and function-level completion create similar failure modes across MBPP-test, MBPP-train, and HumanEval. The fact that this transfer occurs even without aligned probe directions (Section 6.2) suggests that related tasks can share predictive information without relying on a single common correctness direction.

Math transfer is less uniform, which shows that task family alone does not determine transfer. Llama and Gemma transfer better from MATH-500 to GSM8K, while Qwen3 shows the reverse. One plausible explanation is that each model separates correct and incorrect math responses differently depending on its accuracy distribution: MATH-500 is a weaker, more balanced source for Llama and Gemma, whereas Qwen3 performs strongly on both math datasets. Thus, transfer reflects both task structure and model-specific representational organization. Averaged transfer matrices reveal robust trends, but per-model behavior remains necessary for interpreting probe reliability. These model-specific patterns are partially hidden by cross-model averaging: Qwen3's reverse math asymmetry offsets the MATH-500 $\rightarrow$ GSM8K asymmetry observed in Llama and Gemma, while Llama's code-direction alignment is weakened by weaker alignment in Qwen3 and Gemma.

6.2 Probes Transfer Predictively Through Orthogonal Directions

Probe direction similarity is near zero for most dataset pairs, even when predictive transfer is above chance. For example, on Qwen3, MATH-500 $\rightarrow$ GSM8K achieves transfer AUC 0.698, while the corresponding probe-direction cosine similarity is only 0.016. This suggests that predictive transfer and weight-space alignment are not tightly coupled.

The clearest exception is Llama's code family, where code-pair cosine similarities range from 0.112 to 0.252, peaking for MBPP-test $\leftrightarrow$ MBPP-train. This is the only block with clear direction alignment, and even there the absolute cosine values remain modest.

Several mechanisms could explain transfer despite near-orthogonal directions. Correctness may be encoded in a shared low-dimensional subspace while dataset-specific components dominate the full probe vector, reducing cosine similarity. Alternatively, correctness may be distributed across multiple directions, allowing probes trained on different datasets to extract overlapping but non-identical signals. Regularization and limited data may also lead probes to choose different weight configurations even when some shared structure exists.

The key distinction is between predictive transfer and geometric identity. A probe may generalize to a target dataset because its decision boundary captures some transferable feature, but this does not mean its full weight vector aligns with probes trained on that target. Thus, transfer AUC alone is insufficient evidence for a shared correctness direction; geometric claims require direct alignment or subspace analyses.

6.3 Probes Beat Baselines Where Task Structure is Shared

Recall that probes exceed baselines by at least 0.05 AUC mainly in in-domain and within-family transfer cells (Section 5.4). Cross-family cells almost universally lose to the strongest target-side logprob baseline, showing that hidden-state probes should not be treated as universal correctness detectors.

The pattern is still informative: probes add value when source and target share task structure. In-domain gains are expected because the probe is trained on the target distribution, but within-family transfer, especially in code, suggests that probes capture task-family-specific correctness features not fully summarized by token-level confidence.

For broad cross-domain deployment, probe-based correctness detection should therefore be benchmarked against target-side confidence baselines rather than assumed to improve reliability by default.

6.4 The Layer-2 MATH-500 Confound is Model-Specific

The layer-2 anomaly appears specific to Qwen3-8B. On Llama-3.1-8B and Gemma-2-9B, MATH-500's best layer already falls in the upper half of the network, while Qwen3 shifts from layer 2 to layer 29 under the late-layer constraint. We therefore treat the Qwen3 layer-2 result as a model-specific layer-selection confound rather than a universal property of MATH-500 probing (Appendix B.1).

This layer-2 result behaves like an input-distribution confound: several source probes evaluated on MATH-500 layer-2 activations fall below chance, while layer 29 restores above-chance transfer for most sources with only a 0.009 in-domain AUC drop. TriviaQA is the main exception, performing better at layer 2 than layer 29. Overall, this suggests that early layers can capture dataset-specific shortcut features rather than transferable correctness signals.

Future probing studies should either apply a similar late-layer constraint or verify that cross-source probes remain above chance at the selected layer.

7 Conclusion

We study out-of-distribution transfer of response-level correctness probes across three open-weight LLM families, seven datasets, and four task families. Three findings emerge. First, within-family transfer is stronger than cross-family transfer, with a 0.075 AUC gap and the clearest pattern in code generation. Second, predictive transfer can occur even when probe weight vectors are nearly orthogonal, suggesting that correctness is not encoded along a single canonical direction. Third, probes outperform logprob baselines mainly in in-domain and within-family settings, while cross-family probes typically lose to target-side confidence measures. We also identify a layer-selection confound in which early-layer shortcuts can appear predictive in-domain but fail cross-source.

These results suggest that correctness probes should be used as domain-sensitive reliability tools rather than universal correctness detectors. Practically, a probe trained on one domain should not be deployed as a correctness detector for another domain, such as code or factual QA, without target-domain validation. However, probes can add value when source and target share structure, especially within code generation. Future work should test whether alignment can be encouraged during probe training, whether these patterns hold at larger scales, and whether step-level correctness signals transfer differently from response-level signals.

8 Limitations

Several limitations qualify our findings. Many of these limitations reflect the compute and budget constraints of running a systematic transfer study across multiple models, datasets, layers, and random seeds.

Model scale. We evaluate three open-weight models in the 7–9B parameter range. Larger models may organize correctness signals differently, so the orthogonality finding and within-family transfer pattern may not hold unchanged at larger scales. Extending the study to 70B+ models would require substantially more compute.

Random seeds. Our primary transfer matrices use seed 42, with multi-seed validation reported in Appendix H. Although average per-cell variance is low, cells near the cross-family/within-family boundary may be more sensitive to random variation. A larger seed sweep would strengthen the robustness analysis but would multiply the cost of probe training and evaluation.

Singleton task families. MMLU-Pro and TriviaQA are singleton families in our dataset selection, so we cannot directly assess within-family transfer for multiple-choice reasoning or factual recall. Adding more datasets per family would improve coverage but would also require additional generation, grading, hidden-state extraction, and transfer-matrix evaluation.

Small MBPP-train sample. MBPP-train has only 120 examples and class imbalance. Cells involving MBPP-train as source or target therefore show higher variance and should be interpreted cautiously.

Response-level correctness only. Our probes target binary response-level correctness. Step-level correctness signals, intermediate reasoning trajectories, and partial-correctness measures may exhibit different transfer behavior. Studying these alternatives would require more detailed annotation or process-level grading.

Greedy decoding. All generations use greedy decoding with one sample per question. Multi-sample or temperature-varied generation may produce different probe behavior, but would substantially increase the number of generations and hidden-state replays.

English-only datasets. All seven datasets are English, so cross-lingual transfer of correctness probes remains untested. Extending to multilingual datasets is an important direction for future work.

Correlational analysis. Our analysis is correlational. We do not perform causal interventions, such as probe-direction ablation or steering, to test whether probe directions are functionally tied to correctness.

9 Ethical Considerations

Our work studies whether linear probes on LLM hidden states can predict response correctness. Probes have potential downstream applications including selective prediction, calibration, and reliability detection. Our finding that probes are domain-sensitive and do not encode a universal correctness direction has practical implications: deploying probes as general-purpose correctness detectors without domain-specific validation could produce miscalibrated reliability estimates and lead users to trust model outputs in settings where the probe has not been validated. We caution against treating probe AUCs from one domain as evidence of reliable correctness detection in another, particularly across task families.

References

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976.

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

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ning Ding, Yulin Chen, Ganqu Cui, Xingtai Lv, Weilin Zhao, Ruobing Xie, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. 2024. Mastering text, code and math simultaneously via fusing highly specialized language models. *arXiv preprint arXiv:2403.08281*.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large

language models using semantic entropy. *Nature*, 630(8017):625–630.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *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)*, pages 4129–4138.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657.

Ziwei Ji, Delong Chen, Etsuko Ishii, Samuel Cahyawijaya, Yejin Bang, Bryan Wilie, and Pascale Fung. 2024. LLM internal states reveal hallucination risk faced with a query. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 88–104.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

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

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *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)*, pages 1073–1094.

Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. 2025. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6107–6117.

Iván Vicente Moreno Cencerrado, Arnau Padrés Masdemont, Anton Gonzalvez Hawthorne, David Demitri Africa, and Lorenzo Pacchiardi. 2025. No answer needed: Predicting LLM answer accuracy from question-only linear probes. *arXiv preprint arXiv:2509.10625*.

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in NLP applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 479–489.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622.

Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024. Unsupervised real-time hallucination detection based on the internal states of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14379–14391.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

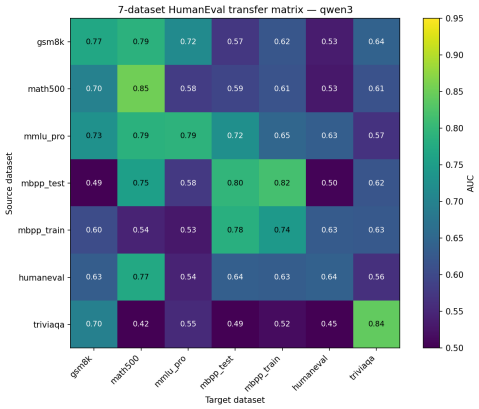
Lei Yang, Renren Jin, Ling Shi, Jianxiang Peng, Yue Chen, and Deyi Xiong. 2025b. ProBench: Benchmarking large language models in competitive programming. *arXiv preprint arXiv:2502.20868*.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 27.

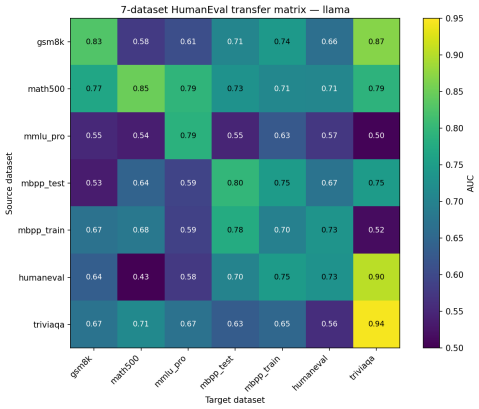
Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025. Reasoning models know when they're right: Probing hidden states for self-verification. *arXiv preprint arXiv:2504.05419*.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Charlotte Zhuang, and Dylan Slack. 2024. A careful examination of large language model performance on grade school arithmetic. *Advances in Neural Information Processing Systems*, 37:46819–46836.

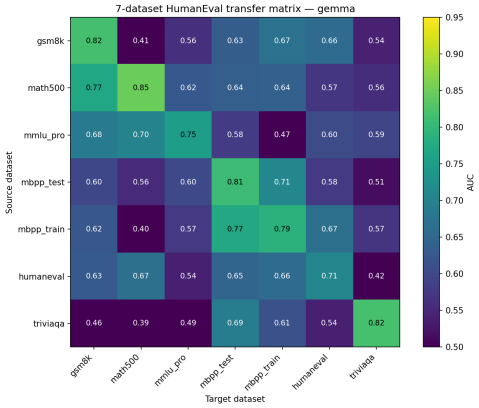
A Per-Model 7×7 Transfer Matrices



Qwen3-8B



Llama-3.1-8B



Gemma-2-9B

Figure A1: Per-model 7×7 probe transfer matrices. Rows indicate source datasets used to train probes, columns indicate target datasets used for evaluation, and cells report AUC for each model.

B Best Layers and Shared Layers

The constrained best layer selected for each model-dataset pair and the shared layer used for cosine-similarity analysis. Constrained layers are selected from the upper half of each model's transformer blocks.

Model	Dataset	Best	Shared
Qwen3-8B	GSM8K	24	26
Qwen3-8B	MATH-500	29	26
Qwen3-8B	MMLU-Pro	24	26
Qwen3-8B	MBPP-test	25	26
Qwen3-8B	MBPP-train	24	26
Qwen3-8B	HumanEval	27	26
Qwen3-8B	TriviaQA	20	26
Llama-3.1-8B	GSM8K	18	16
Llama-3.1-8B	MATH-500	16	16
Llama-3.1-8B	MMLU-Pro	30	16
Llama-3.1-8B	MBPP-test	18	16
Llama-3.1-8B	MBPP-train	30	16
Llama-3.1-8B	HumanEval	30	16
Llama-3.1-8B	TriviaQA	19	16
Gemma-2-9B	GSM8K	41	41
Gemma-2-9B	MATH-500	26	41
Gemma-2-9B	MMLU-Pro	21	41
Gemma-2-9B	MBPP-test	41	41
Gemma-2-9B	MBPP-train	36	41
Gemma-2-9B	HumanEval	41	41
Gemma-2-9B	TriviaQA	34	41

Table B1: Constrained best layers and shared cosine layers.

B.1 Qwen3 MATH-500 Layer-2 Diagnostic

For Qwen3-8B with MATH-500 as the target, we compare transfer from each non-MATH-500 source at the unconstrained MATH-500 layer 2 and the constrained MATH-500 layer 29. The layer-2 setting is often anti-predictive or weak under cross-source transfer, while the constrained late layer recovers stronger transfer for most sources.

Source	Source best layer	AUC @ layer 2	AUC @ layer 29
GSM8K	24	0.280	0.789
MMLU-Pro	24	0.442	0.787
MBPP-test	25	0.606	0.753
MBPP-train	24	0.318	0.545
HumanEval	27	0.341	0.772
TriviaQA	20	0.677	0.422

Table B2: Qwen3-8B MATH-500 layer-2 diagnostic. AUCs compare transfer to the MATH-500 target when evaluated at MATH-500 layer 2 versus the constrained MATH-500 layer 29.

C Probe-vs-Baseline Results

For each source-target-model cell, we compare probe AUC against the strongest target-side log-prob baseline. The gap is defined as probe AUC minus best baseline AUC.

D Dataset Accuracy and Class Balance

Model accuracy and correctness-label balance for each dataset.

Source	Target	Probe AUC	Best baseline	Baseline AUC	Gap	Beat ≥ 0.05
GSM8K	GSM8K	0.770	mean_ent	0.739	0.031	No
GSM8K	MATH-500	0.789	max_ent	0.827	-0.038	No
GSM8K	MMLU-Pro	0.715	max_ent	0.754	-0.038	No
GSM8K	MBPP-test	0.573	max_ent	0.663	-0.090	No
GSM8K	MBPP-train	0.620	mean_ent	0.726	-0.106	No
GSM8K	HumanEval	0.534	min_logp	0.597	-0.064	No
GSM8K	TriviaQA	0.642	max_ent	0.810	-0.169	No
MATH-500	GSM8K	0.698	mean_ent	0.739	-0.041	No
MATH-500	MATH-500	0.852	max_ent	0.827	0.025	No
MATH-500	MMLU-Pro	0.580	max_ent	0.754	-0.173	No
MATH-500	MBPP-test	0.594	max_ent	0.663	-0.068	No
MATH-500	MBPP-train	0.615	mean_ent	0.726	-0.112	No
MATH-500	HumanEval	0.528	min_logp	0.597	-0.069	No
MATH-500	TriviaQA	0.614	max_ent	0.810	-0.197	No
MMLU-Pro	GSM8K	0.728	mean_ent	0.739	-0.011	No
MMLU-Pro	MATH-500	0.787	max_ent	0.827	-0.040	No
MMLU-Pro	MMLU-Pro	0.791	max_ent	0.754	0.037	No
MMLU-Pro	MBPP-test	0.723	max_ent	0.663	0.061	Yes
MMLU-Pro	MBPP-train	0.648	mean_ent	0.726	-0.078	No
MMLU-Pro	HumanEval	0.634	min_logp	0.597	0.037	No
MMLU-Pro	TriviaQA	0.566	max_ent	0.810	-0.244	No
MBPP-test	GSM8K	0.485	mean_ent	0.739	-0.254	No
MBPP-test	MATH-500	0.753	max_ent	0.827	-0.074	No
MBPP-test	MMLU-Pro	0.575	max_ent	0.754	-0.179	No
MBPP-test	MBPP-test	0.796	max_ent	0.663	0.133	Yes
MBPP-test	MBPP-train	0.817	mean_ent	0.726	0.091	Yes
MBPP-test	HumanEval	0.504	min_logp	0.597	-0.093	No
MBPP-test	TriviaQA	0.618	max_ent	0.810	-0.192	No
MBPP-train	GSM8K	0.604	mean_ent	0.739	-0.135	No
MBPP-train	MATH-500	0.545	max_ent	0.827	-0.283	No
MBPP-train	MMLU-Pro	0.529	max_ent	0.754	-0.225	No
MBPP-train	MBPP-test	0.785	max_ent	0.663	0.122	Yes
MBPP-train	MBPP-train	0.743	mean_ent	0.726	0.016	No
MBPP-train	HumanEval	0.630	min_logp	0.597	0.032	No
MBPP-train	TriviaQA	0.626	max_ent	0.810	-0.184	No
HumanEval	GSM8K	0.633	mean_ent	0.739	-0.107	No
HumanEval	MATH-500	0.772	max_ent	0.827	-0.055	No
HumanEval	MMLU-Pro	0.545	max_ent	0.754	-0.209	No
HumanEval	MBPP-test	0.636	max_ent	0.663	-0.027	No
HumanEval	MBPP-train	0.632	mean_ent	0.726	-0.094	No
HumanEval	HumanEval	0.640	min_logp	0.597	0.043	No
HumanEval	TriviaQA	0.562	max_ent	0.810	-0.248	No
TriviaQA	GSM8K	0.697	mean_ent	0.739	-0.043	No
TriviaQA	MATH-500	0.422	max_ent	0.827	-0.405	No
TriviaQA	MMLU-Pro	0.554	max_ent	0.754	-0.199	No
TriviaQA	MBPP-test	0.487	max_ent	0.663	-0.176	No
TriviaQA	MBPP-train	0.516	mean_ent	0.726	-0.211	No
TriviaQA	HumanEval	0.448	min_logp	0.597	-0.149	No
TriviaQA	TriviaQA	0.842	max_ent	0.810	0.032	No

Table C1: Qwen3-8B probe-vs-baseline results.

Source	Target	Probe AUC	Best baseline	Baseline AUC	Gap	Beat ≥ 0.05
GSM8K	GSM8K	0.827	max_ent	0.708	0.119	Yes
GSM8K	MATH-500	0.577	max_ent	0.810	-0.233	No
GSM8K	MMLU-Pro	0.608	max_ent	0.692	-0.083	No
GSM8K	MBPP-test	0.705	max_ent	0.757	-0.052	No
GSM8K	MBPP-train	0.742	mean_ent	0.726	0.015	No
GSM8K	HumanEval	0.664	mean_logp	0.691	-0.027	No
GSM8K	TriviaQA	0.869	final_logp	0.909	-0.040	No
MATH-500	GSM8K	0.766	max_ent	0.708	0.058	Yes
MATH-500	MATH-500	0.849	max_ent	0.810	0.040	No
MATH-500	MMLU-Pro	0.794	max_ent	0.692	0.103	Yes
MATH-500	MBPP-test	0.729	max_ent	0.757	-0.029	No
MATH-500	MBPP-train	0.708	mean_ent	0.726	-0.018	No
MATH-500	HumanEval	0.707	mean_logp	0.691	0.016	No
MATH-500	TriviaQA	0.791	final_logp	0.909	-0.118	No
MMLU-Pro	GSM8K	0.551	max_ent	0.708	-0.157	No
MMLU-Pro	MATH-500	0.539	max_ent	0.810	-0.271	No
MMLU-Pro	MMLU-Pro	0.788	max_ent	0.692	0.096	Yes
MMLU-Pro	MBPP-test	0.546	max_ent	0.757	-0.211	No
MMLU-Pro	MBPP-train	0.626	mean_ent	0.726	-0.101	No
MMLU-Pro	HumanEval	0.574	mean_logp	0.691	-0.117	No
MMLU-Pro	TriviaQA	0.504	final_logp	0.909	-0.405	No
MBPP-test	GSM8K	0.535	max_ent	0.708	-0.174	No
MBPP-test	MATH-500	0.643	max_ent	0.810	-0.167	No
MBPP-test	MMLU-Pro	0.592	max_ent	0.692	-0.099	No
MBPP-test	MBPP-test	0.798	max_ent	0.757	0.040	No
MBPP-test	MBPP-train	0.747	mean_ent	0.726	0.021	No
MBPP-test	HumanEval	0.672	mean_logp	0.691	-0.019	No
MBPP-test	TriviaQA	0.747	final_logp	0.909	-0.162	No
MBPP-train	GSM8K	0.670	max_ent	0.708	-0.039	No
MBPP-train	MATH-500	0.681	max_ent	0.810	-0.129	No
MBPP-train	MMLU-Pro	0.587	max_ent	0.692	-0.105	No
MBPP-train	MBPP-test	0.777	max_ent	0.757	0.019	No
MBPP-train	MBPP-train	0.696	mean_ent	0.726	-0.031	No
MBPP-train	HumanEval	0.733	mean_logp	0.691	0.043	No
MBPP-train	TriviaQA	0.516	final_logp	0.909	-0.393	No
HumanEval	GSM8K	0.635	max_ent	0.708	-0.073	No
HumanEval	MATH-500	0.435	max_ent	0.810	-0.375	No
HumanEval	MMLU-Pro	0.575	max_ent	0.692	-0.116	No
HumanEval	MBPP-test	0.701	max_ent	0.757	-0.056	No
HumanEval	MBPP-train	0.747	mean_ent	0.726	0.021	No
HumanEval	HumanEval	0.729	mean_logp	0.691	0.039	No
HumanEval	TriviaQA	0.903	final_logp	0.909	-0.005	No
TriviaQA	GSM8K	0.672	max_ent	0.708	-0.037	No
TriviaQA	MATH-500	0.714	max_ent	0.810	-0.096	No
TriviaQA	MMLU-Pro	0.670	max_ent	0.692	-0.022	No
TriviaQA	MBPP-test	0.631	max_ent	0.757	-0.126	No
TriviaQA	MBPP-train	0.650	mean_ent	0.726	-0.076	No
TriviaQA	HumanEval	0.558	mean_logp	0.691	-0.132	No
TriviaQA	TriviaQA	0.945	final_logp	0.909	0.036	No

Table C2: Llama-3.1-8B probe-vs-baseline results.

Source	Target	Probe AUC	Best baseline	Baseline AUC	Gap	Beat ≥ 0.05
GSM8K	GSM8K	0.818	mean_ent	0.690	0.128	Yes
GSM8K	MATH-500	0.407	max_ent	0.836	-0.430	No
GSM8K	MMLU-Pro	0.561	max_ent	0.648	-0.086	No
GSM8K	MBPP-test	0.632	final_logp	0.675	-0.043	No
GSM8K	MBPP-train	0.667	mean_ent	0.680	-0.013	No
GSM8K	HumanEval	0.658	mean_logp	0.708	-0.050	No
GSM8K	TriviaQA	0.541	max_ent	0.777	-0.236	No
MATH-500	GSM8K	0.768	mean_ent	0.690	0.078	Yes
MATH-500	MATH-500	0.850	max_ent	0.836	0.013	No
MATH-500	MMLU-Pro	0.625	max_ent	0.648	-0.023	No
MATH-500	MBPP-test	0.636	final_logp	0.675	-0.039	No
MATH-500	MBPP-train	0.637	mean_ent	0.680	-0.043	No
MATH-500	HumanEval	0.571	mean_logp	0.708	-0.136	No
MATH-500	TriviaQA	0.559	max_ent	0.777	-0.218	No
MMLU-Pro	GSM8K	0.684	mean_ent	0.690	-0.006	No
MMLU-Pro	MATH-500	0.702	max_ent	0.836	-0.134	No
MMLU-Pro	MMLU-Pro	0.750	max_ent	0.648	0.103	Yes
MMLU-Pro	MBPP-test	0.580	final_logp	0.675	-0.095	No
MMLU-Pro	MBPP-train	0.474	mean_ent	0.680	-0.206	No
MMLU-Pro	HumanEval	0.603	mean_logp	0.708	-0.105	No
MMLU-Pro	TriviaQA	0.594	max_ent	0.777	-0.183	No
MBPP-test	GSM8K	0.598	mean_ent	0.690	-0.092	No
MBPP-test	MATH-500	0.556	max_ent	0.836	-0.280	No
MBPP-test	MMLU-Pro	0.599	max_ent	0.648	-0.048	No
MBPP-test	MBPP-test	0.808	final_logp	0.675	0.133	Yes
MBPP-test	MBPP-train	0.708	mean_ent	0.680	0.028	No
MBPP-test	HumanEval	0.576	mean_logp	0.708	-0.132	No
MBPP-test	TriviaQA	0.515	max_ent	0.777	-0.262	No
MBPP-train	GSM8K	0.622	mean_ent	0.690	-0.068	No
MBPP-train	MATH-500	0.401	max_ent	0.836	-0.435	No
MBPP-train	MMLU-Pro	0.574	max_ent	0.648	-0.074	No
MBPP-train	MBPP-test	0.771	final_logp	0.675	0.096	Yes
MBPP-train	MBPP-train	0.789	mean_ent	0.680	0.109	Yes
MBPP-train	HumanEval	0.665	mean_logp	0.708	-0.042	No
MBPP-train	TriviaQA	0.570	max_ent	0.777	-0.207	No
HumanEval	GSM8K	0.627	mean_ent	0.690	-0.063	No
HumanEval	MATH-500	0.667	max_ent	0.836	-0.170	No
HumanEval	MMLU-Pro	0.540	max_ent	0.648	-0.108	No
HumanEval	MBPP-test	0.647	final_logp	0.675	-0.028	No
HumanEval	MBPP-train	0.658	mean_ent	0.680	-0.022	No
HumanEval	HumanEval	0.707	mean_logp	0.708	-0.000	No
HumanEval	TriviaQA	0.420	max_ent	0.777	-0.357	No
TriviaQA	GSM8K	0.459	mean_ent	0.690	-0.232	No
TriviaQA	MATH-500	0.386	max_ent	0.836	-0.450	No
TriviaQA	MMLU-Pro	0.495	max_ent	0.648	-0.153	No
TriviaQA	MBPP-test	0.689	final_logp	0.675	0.014	No
TriviaQA	MBPP-train	0.611	mean_ent	0.680	-0.069	No
TriviaQA	HumanEval	0.539	mean_logp	0.708	-0.169	No
TriviaQA	TriviaQA	0.818	max_ent	0.777	0.041	No

Table C3: Gemma-2-9B probe-vs-baseline results.

Dataset	Family	N	Description	Grading
GSM8K (Cobbe et al., 2021)	Math	1000	Grade-school arithmetic word problems	Numeric exact-match
MATH-500 (Hendrycks et al., 2021; Lightman et al., 2023)	Math	500	Competition mathematics	SymPy-based equivalence
MMLU-Pro (Wang et al., 2024)	Multiple-choice	1000	Professional-level domain knowledge	Letter exact-match
MBPP-test (Austin et al., 2021)	Code	257	Python programming, sanitized test split	Unit-test execution
MBPP-train (Austin et al., 2021)	Code	120	Python programming, sanitized train split	Unit-test execution
HumanEval (Chen et al., 2021)	Code	164	Python function-synthesis problems with docstrings and required entry points	Unit-test execution
TriviaQA (Joshi et al., 2017)	Factual	1000	Open-domain factual question answering	Normalized alias match

Table D1: Datasets used in the experiment. MBPP-test, MBPP-train, and HumanEval use their full available splits; the remaining datasets use sampled test subsets of 500 or 1000 examples with `random_state = 42`.

Model	Dataset	N	Accuracy	Correct	Incorrect
Qwen3-8B	GSM8K	1000	0.927	927	73
Qwen3-8B	MATH-500	500	0.772	386	114
Qwen3-8B	MMLU-Pro	1000	0.665	665	335
Qwen3-8B	MBPP-test	257	0.774	199	58
Qwen3-8B	MBPP-train	120	0.775	93	27
Qwen3-8B	HumanEval	164	0.823	135	29
Qwen3-8B	TriviaQA	1000	0.619	619	381
Llama-3.1-8B	GSM8K	1000	0.823	823	177
Llama-3.1-8B	MATH-500	500	0.438	219	281
Llama-3.1-8B	MMLU-Pro	1000	0.428	428	572
Llama-3.1-8B	MBPP-test	257	0.665	171	86
Llama-3.1-8B	MBPP-train	120	0.708	85	35
Llama-3.1-8B	HumanEval	164	0.628	103	61
Llama-3.1-8B	TriviaQA	1000	0.618	618	382
Gemma-2-9B	GSM8K	1000	0.869	869	131
Gemma-2-9B	MATH-500	500	0.460	230	270
Gemma-2-9B	MMLU-Pro	1000	0.490	490	510
Gemma-2-9B	MBPP-test	257	0.673	173	84
Gemma-2-9B	MBPP-train	120	0.742	89	31
Gemma-2-9B	HumanEval	164	0.567	93	71
Gemma-2-9B	TriviaQA	1000	0.754	754	246

Table D2: Dataset accuracy and class balance for each model-dataset pair.

E Detailed Methodology Equations

E.1 Probe Training Objective

For each model-dataset pair, the logistic regression probe defined in Section 4.1 is trained by minimizing the class-balanced cross-entropy loss with L2 regularization:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \alpha_{y_i} \left[y_i \log f_\theta(\mathbf{h}_i) + (1 - y_i) \log \left(1 - f_\theta(\mathbf{h}_i)\right)\right] + \frac{1}{2C}\|\mathbf{w}\|_2^2. \tag{E1}$$

where $y_i \in \{0, 1\}$ is the correctness label, $\mathbf{h}_i$ is the hidden state for example i, $C = 1.0$ is the inverse regularization strength, and $\alpha_{y_i} = N/(2N_{y_i})$ is the class-balanced weight, with N_{y_i} the count of examples with label y_i. Optimization is performed with L-BFGS via scikit-learn's `LogisticRegression(max_iter=2000, C=1.0, class_weight='balanced')`.

E.2 Token-Level Log Probabilities and Entropies

The first-pass generation in Section 4.2 produces token-level statistics used as baselines. For a generated token g_k at position k in the sequence:

$$\ell_k = \log p_{\mathcal{M}}\left(g_k \mid x_{1:T}, g_{1:k-1}\right), \tag{E2}$$

$$H_k = -\sum_{v \in \mathcal{V}} p_{\mathcal{M}}\left(v \mid x_{1:T}, g_{1:k-1}\right) \times \log p_{\mathcal{M}}\left(v \mid x_{1:T}, g_{1:k-1}\right). \tag{E3}$$

where $\mathcal{V}$ is the vocabulary, ℓ_k is the log probability of the actually generated token, and H_k is the Shannon entropy of the full next-token distribution at position k.

E.3 Hidden State Position Indexing

The four extraction positions in Section 4.2 correspond to the following indices into the per-layer hidden-state buffer $\mathbf{H}^{(\ell)} \in \mathbb{R}^{(T+K)\times d}$, where T is prompt length and K is generation length:

$$\mathbf{h}_1^{(\ell)} = \mathbf{H}_T^{(\ell)} \quad \text{(end of input question)}, \tag{E4}$$

$$\mathbf{h}_2^{(\ell)} = \mathbf{H}_{T+\lfloor K/2 \rfloor}^{(\ell)} \quad \text{(middle of generation)}, \tag{E5}$$

$$\mathbf{h}_3^{(\ell)} = \mathbf{H}_{T+m-1}^{(\ell)} \quad \text{(pre-answer marker)}, \tag{E6}$$

$$\mathbf{h}_4^{(\ell)} = \mathbf{H}_{T+K}^{(\ell)} \quad \text{(final generated token)}. \tag{E7}$$

where m is the position of the first answer-format marker in the generated text. If no marker is found, $\mathbf{h}_3^{(\ell)}$ falls back to position $T + K - 1$, the penultimate generated token.

E.4 Cross-Model Averaging

The cross-model transfer matrix in Section 4.4 is computed by averaging per-model matrices element-wise:

$$\bar{M}[\mathcal{S}, \mathcal{T}] = \frac{1}{|\mathfrak{M}|} \sum_{\mathcal{M} \in \mathfrak{M}} M_{\mathcal{M}}[\mathcal{S}, \mathcal{T}]. \tag{E8}$$

where $\mathfrak{M}$ contains `Qwen3-8B`, `Llama-3.1-8B`, and `Gemma-2-9B`.

E.5 Shared Layer Selection for Direction Similarity

The shared layer used for direction similarity in Section 4.5 is selected per model as the layer with the highest average in-domain AUC across all datasets, constrained to the upper half of the network:

$$\ell^*_{\text{shared}}(\mathcal{M}) = \arg\max_{\ell \in \{\lfloor L/2 \rfloor, \ldots, L\}} \frac{1}{|\mathfrak{D}|} \sum_{\mathcal{D} \in \mathfrak{D}} \text{AUC}_{\text{CV}}(\mathcal{D}, \ell). \tag{E9}$$

where $\mathfrak{D}$ is the set of seven datasets. The resulting shared layers are 26 for `Qwen3-8B`, 16 for `Llama-3.1-8B`, and 41 for `Gemma-2-9B`.

E.6 Logprob Baseline Definitions

The four baselines defined in Section 4.6 are computed per example i from the token-level statistics in Appendix E.2:

$$b_i^{\text{mean_logp}} = \frac{1}{K_i} \sum_{k=1}^{K_i} \ell_{i,k}, \tag{E10}$$

$$b_i^{\text{final_logp}} = \ell_{i,K_i}, \tag{E11}$$

$$b_i^{\text{mean_ent}} = -\frac{1}{K_i} \sum_{k=1}^{K_i} H_{i,k}, \tag{E12}$$

$$b_i^{\text{max_ent}} = -\max_{k \in \{1, \ldots, K_i\}} H_{i,k}. \tag{E13}$$

The entropy baselines are negated because higher entropy corresponds to lower expected correctness. For each baseline b, the per-cell AUC is computed against the target dataset's correctness labels $\mathbf{y}_{\mathcal{T}}$.

E.7 Multi-Seed Mean and Standard Deviation

For each cell of the transfer matrix, the multi-seed mean and standard deviation across $|S| = 5$ seeds are computed as:

$$\mu_{\mathcal{M}}[\mathcal{S}, \mathcal{T}] = \frac{1}{|S|} \sum_{s \in S} M_{\mathcal{M}}^{(s)}[\mathcal{S}, \mathcal{T}], \tag{E14}$$

$$\sigma_{\mathcal{M}}[\mathcal{S}, \mathcal{T}] = \Bigg(\frac{1}{|S|} \sum_{s \in S} \big(M_{\mathcal{M}}^{(s)}[\mathcal{S}, \mathcal{T}] - \mu_{\mathcal{M}}[\mathcal{S}, \mathcal{T}] \big)^2 \Bigg)^{1/2}. \tag{E15}$$

where $M_{\mathcal{M}}^{(s)}$ is the transfer matrix computed with seed s and $S = \{42, 7, 13, 100, 0\}$.

F Prompt Templates

G Labeling and Grading Details

We assign binary correctness labels using dataset-specific automatic graders. Each generated response is parsed into a final answer or executable solution and compared against the dataset gold label using the appropriate evaluation method.

G.1 Math Grading

For GSM8K, we extract the final numeric answer from the model response using the output anchor `Final Answer:` and compare it against the gold numeric answer with exact matching after basic normalization. For MATH-500, we extract the final answer from the model response, prioritizing answers inside `\boxed{...}` when present. Extracted answers are normalized and compared against the gold answer using symbolic equivalence when possible. If symbolic parsing fails, the evaluator falls back to normalized string comparison.

G.2 Multiple-Choice Grading

For MMLU-Pro, we parse the model's final answer as a multiple-choice letter. The evaluator prioritizes answers following the output anchor `Answer:` and constrains valid predictions to the available option letters. A response is marked correct if the extracted letter matches the gold answer.

G.3 Code Grading

For MBPP-test, MBPP-train, and HumanEval, we extract generated Python code from the model response and execute it against the dataset-provided unit tests. Code is run in a sandboxed temporary environment with a timeout to prevent non-terminating executions. A response is labeled correct only if the extracted program passes all required tests. For HumanEval, the evaluator additionally preserves the required function signature and checks the generated implementation against the official task tests.

G.4 Factual QA Grading

For TriviaQA, we extract the final answer using the `Final Answer:` anchor and compare it against all provided gold aliases. Both prediction and aliases are normalized before comparison by lowercasing, removing punctuation, and stripping extra whitespace. A response is marked correct if the normalized prediction matches any normalized alias.

H Multi-Seed Stability

Model	Avg. std	Max std	Max-std cell	Mean AUC
Qwen3	0.021	0.061	MMLU-Pro → MBPP-train	0.584
Llama	0.019	0.056	MBPP-train → TriviaQA	0.642
Gemma	0.022	0.079	MMLU-Pro → MBPP-train	0.564

Table H1: Multi-seed stability across five random seeds. Average cell standard deviation is around 0.02 AUC, with the largest variance concentrated in cells involving MBPP-train.

Dataset	Prompt template
GSM8K	`Solve step by step. End with: "Final Answer: "`
MATH-500	`Solve step by step. Put final answer in \boxed{...}.`
MMLU-Pro	`Reason step by step, then end with "Answer: ".`
MBPP-test	`Write the Python solution. Return only the code in a python block.`
MBPP-train	`Write the Python solution. Return only the code in a python block.`
HumanEval	`Write the Python solution. Return only the code in a python block. Preserve the required function signature.`
TriviaQA	`Answer the following. End with: "Final Answer: ".`

Table F1: Prompt templates used for each dataset.

I Hardware, Model and Reproducibility

All experiments were run on a single NVIDIA H100 80GB GPU, requiring approximately 35 total GPU hours and costing about 150 USD in rental compute.Dataset sampling uses `random_state=42`. Multi-seed robustness uses seeds $\{42, 7, 13, 100, 0\}$. Generation uses greedy decoding with a maximum of 2048 new tokens. Probes use scikit-learn logistic regression with `C=1.0`, `max_iter=2000`, and `class_weight="balanced"`. We store generated outputs, correctness labels, hidden-state activations, transfer matrices, and probe results for reproducibility. All artifacts are used solely for research evaluation of LLM correctness signals, consistent with the intended use of public benchmark datasets and open-weight models; we do not redistribute raw benchmark datasets or model weights.

Code availability. To preserve anonymity during peer review, we do not include a public repository link in the submission. Upon publication, we will release the code on GitHub, including configuration files, prompt templates, evaluation scripts, and analysis notebooks. Generated outputs, correctness labels, transfer matrices, and probe results will also be released where permitted by dataset licenses.

Model	Family	Params	Layers	Hidden	HuggingFace ID
Qwen3-8B	Qwen3	8B	36	4096	`Qwen/Qwen3-8B`
Llama-3.1-8B-Instruct	Llama 3.1	8B	32	4096	`meta-llama/Llama-3.1-8B-Instruct`
Gemma-2-9B-it	Gemma 2	9B	42	3584	`google/gemma-2-9b-it`

Table I1: Models used in the study. Open-weight instruction-tuned dense decoder-only models in the 7–9B parameter range.

J AI Assistance Statement

The authors used AI assistance for editing, academic wording, and LaTeX formatting during manuscript drafting. AI tools were also used for debugging support. All research questions, experimental design choices, implementation decisions, numerical results, analyses, and final claims were developed, reviewed, and verified by the authors. AI assistance was used to improve presentation and clarity, not to replace author judgment or independent verification.