Knowing Before Answering: Decoding Language Models for Reliable RAG

Syed Mahbubul Huq[1], Chris Child[1], Tillman Weyde[1] & Pranava Madhyastha[1,2]
[1] City St George's, University of London
[2] The Alan Turing Institute
Correspondence: {syed-mahbubul.huq.2,pranava.madhyastha}@city.ac.uk

Abstract

In Retrieval-Augmented Generation (RAG), retrieval may provide insufficient or conflicting information needed to answer a question. The system should not only know when to answer but also be able to identify cases in which the documents provided in RAG are insufficient or contain conflicting information. This can be framed as a three-way classification problem, where we use the model's internal signals to determine whether the provided information in the input can be classified as sufficient, insufficient, or conflicting. We create a controlled benchmark dataset that replicates a RAG setup with fictitious information and labels each instance as answerable, insufficient, or conflicting. We use hidden activations and attention-derived features as inputs to train a lightweight linear model to distinguish among the three classes. Across 16 language models spanning different architectures and a range of model sizes, our feature-based router consistently outperforms prompting-based baselines and the performance of specialised RAG-models. We further conduct analyses into the information dynamics of the models. We show that the most informative signals for the classification are available in the middle layers, with hidden activation states being more effective than attention values or the MLP-feature outputs in most of the tested models. Overall, our results suggest that language models internally encode whether retrieved evidence is sufficient to support answering, and that this signal can be decoded reliably for RAG triage.

1 Introduction

Retrieval-Augmented Generation (RAG) extends language models with external evidence at inference time, improving factuality and reducing hallucination (Lewis et al., 2020). But retrieval is imperfect. In practice, retrieved documents may lack the information needed to answer a question, or may contain mutually contradictory answers. Prior work shows that in these cases, models tend to generate unsupported or contradictory responses rather than acknowledging the limitations of the evidence (Niu et al., 2024; Joren et al., 2025).

Existing methods address this as a two-way problem that involves either answering the question or refusing to answer the question. These predominantly rely on prompting (Zhang et al., 2025b; Zhou et al., 2023), confidence-based abstention (Kadavath et al., 2022), retrieval-side relevance estimation (Yoran et al., 2024), or external entailment (Honovich et al., 2022). These approaches share two limitations: they operate on the output side of the model, and they collapse the problem to two classes, ignoring the distinct case where conflicting evidence is present. A model that confidently answers from contradictory documents is failing in a qualitatively different way from one that answers without any supporting evidence. We highlight here that this distinction is an important one, especially for downstream tasks, trust and system design.

Code and dataset are available at https://github.com/SyedHuq28/Knowing-Before-Answering-Decoding-Language-Models-for-Reliable-RAG.

In this work, we propose a three-way framing: for a given question and retrieved context, the evidence state is either *Answer* (sufficient, consistent evidence), *Refuse* (insufficient evidence), or *Conflict* (contradictory evidence). We then present a methodology that exploits encoded information present in internal representations for prediction. Specifically, we train a lightweight *triage router* (which is a logistic regression based prediction module) on activations extracted from a single hidden layer of the language model, using a controlled diagnostic benchmark that isolates evidence-state decisions under matched inputs. We evaluate across 16 transformer-based language models spanning 90M to 32B parameters, diverse architectures, and training regimes. The router consistently and substantially outperforms prompt-based and RAG-specialised baselines across almost all models, achieving up to 0.91 accuracy and reducing false answer rates by up to 75% relative to the strongest prompt-based competitor, with no additional tokens or inference cost.

Furthermore, we analyse *where* evidence-state information is linearly decodable from the model's internal representations. Layer probing experiments show that the most informative representations consistently emerge in the middle layers of the network, a pattern that holds across model families and sizes. Hidden-state patching confirms that targeted intervention at the best probe layer reliably reshapes downstream model behaviour.

Our core contributions in the paper are: (i) A three-way RAG triage formulation that explicitly distinguishes *Answer*, *Refuse*, and *Conflict* evidence states, with a controlled diagnostic benchmark designed to allow us to evaluate the decisions; (ii) A lightweight triage router that decodes evidence state from a single hidden layer, outperforming all prompt-based and RAG-specialised baselines across 16 models without additional inference cost; and (iii) An analysis of where evidence-state information is linearly decodable, together with hidden-state interventions that test its influence on downstream behaviour.

2 Methodology

In our experimentation, for a given question q and a set of retrieved documents $D = d_1, \ldots, d_k$, the system is expected to decide whether the retrieved evidence (i) answers a question, (ii) refuses to answer, or (iii) identifies conflicting answers within the provided context. We denote the triage label by

$$y \in \{\textsc{Answer}, \textsc{Refuse}, \textsc{Conflict}\}\,. \tag{1}$$

Before the language model generates the answer, we propose a system to reliably predict y to reduce instances where the model should refrain from answering or report conflicting cases. An instance is labelled ANSWER if the retrieved documents provide sufficient evidence and have only ONE grounded answer to the question. If the instance has MORE THAN ONE grounded answer to the question, it is labelled as CONFLICT. Finally, the instance is labelled REFUSE if the retrieved documents do not contain enough information to answer the question. We can utilise signals of LMs as distinguishing features using a lightweight classifier that can be used to study the internal states and distinguish decisions.

2.1 Dataset Construction

Our objective is to isolate a specific decision problem: whether the provided evidence is sufficient to answer, insufficient, or conflicting. We therefore use and modify three popular datasets: TriviaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018), and Natural Questions (Kwiatkowski et al., 2019). This design focuses on interpretability by reducing parametric knowledge leakage and maintaining variability in the dataset. We randomly sample 797 question–answer–evidence triples from each source, resulting in 2,391 base questions in total. For each question, we keep the supporting document associated with the answer and treat it as the evidence document.

To reduce parametric knowledge leakage, we apply counterfactual entity substitution following the strategy of (Xie et al., 2024), implemented via Gemini 2.5 Pro. We note that the substitution is applied globally and consistently: i.e., whenever an original entity appears

across any document in the dataset, it is replaced with the same counterfactual, preventing models from recovering the original answer through cross-document consistency. Each substituted entity is unique within the dataset to prevent repetition-based inference.

To emulate a standard RAG setting, we retrieve candidate documents using BM25, a popular lexical retrieval baseline that is widely used in RAG pipelines (Wang et al., 2024). For each question, we retrieve documents from other contexts in the dataset, which can be gold contexts for other questions, and create three evidence configurations.

For each question, we construct three five-document context sets. The ANSWER set contains the gold document together with four BM25-retrieved distractor documents, with the gold document placed at a random position for all ANSWER, and CONFLICT cases. The REFUSE set contains five BM25-retrieved non-gold documents. The CONFLICT set contains two gold documents, where the original gold document and a conflicting version created by replacing the answer entity with a different counterfactual conflicting entity using Gemini 2.5 Pro are combined with three distractor documents retrieved using BM25.

Each question gives us three aligned instances, one for each expected label. The final dataset contains $2{,}391 \times 3 = 7{,}173$ instances in total, with every instance containing exactly five documents. We use a question-level split of 70% training, 10% validation, and 20% test data that avoids split leakage and remains constant across our experiments.

Quality control. We apply automated and human verification to ensure dataset quality. First, Gemini 2.5 Pro is used to verify that, outside of the gold documents, no distractor document contains the substituted answer entity. Second, an internal consistency check was performed by a co-author on a stratified sample of 10%, verifying entity replacement correctness and document coherence. We report an internal agreement rate of 100% on this sample set. The resulting benchmark is intentionally controlled and should be viewed as a diagnostic testbed rather than as a replacement for naturalistic RAG evaluation. Our aim is to situate the dataset to isolate whether models internally distinguish answerable, unanswerable, and conflicting evidence conditions under matched inputs.

On Parametric Knowledge Leakage A core assumption of our benchmark is that entity substitution prevents models from answering questions from parametric memory alone. To verify this empirically, we measure each model's accuracy on the ANSWER instances when *no context documents are provided* (**NO_CTX** condition). Note here that if substitution is indeed effective, models should perform near chance ($\approx$0.33 for a three-way problem, or near 0.0 if measured only against the substituted answer string).

2.2 Evaluation Metrics

The overall classification accuracy over the three labels is our **Accuracy** metric. **Macro-F1** is the macro-averaged F1 across ANSWER, REFUSE, and CONFLICT. Lastly, **False Answer Rate (FAR)** is our reliability metric, which is defined as the fraction of examples whose true label is either REFUSE or CONFLICT, but are incorrectly predicted as ANSWER. FAR captures the cases where the model generates an answer when it should not do so. Additionally, for evaluation, we perform a substring match to determine whether the language models produced a refusal/conflict string, validating the accuracy of the baselines. Furthermore, we report 95% confidence intervals using 10,000 bootstrap resamples with replacement from the 1,435 test instances, recomputing metrics on fixed test predictions without retraining; intervals are the 2.5th–97.5th empirical percentiles, using a fixed random seed of 42.

2.3 Baselines

We compare our router against prompt-based and RAG-specialised baselines: **ATTR**: The model is instructed to answer using only the retrieved documents with an explicit refusal or conflict policy. We construct and use the prompt, taking inspiration from the *Attribute* style structure presented by Zhou et al. (2023). **KRE**: Based on Ying et al. (2024), we adapt their *instruction-with-hint* prompting design and construct the baseline prompt. **OPIN_INSTR**: We adapt the opinion based prompting framework of Zhou et al. (2023) and construct the

baseline. **FaithfulRAG style CoT Prompt:** Inspired by Zhang et al. (2025b), we adapt their CoT style prompting method and construct this baseline. **Chat-QA-1.5:** We utilise a specialised pre-trained RAG model introduced by Liu et al. (2024), with Llama3-8B model as its backbone. **Self-RAG:** We evaluate another specialised pre-trained RAG model introduced by Asai et al. (2024). The backbone of the model is the Llama2-7B model. We additionally evaluate two surface-text baselines without hidden states: **TF-IDF with logistic regression** over the concatenated question and documents, and a zero-shot **NLI DeBERTa cross-encoder**. Moreover, in addition to the above-mentioned baselines, we also test the language models on vanilla prompts and RAG setup without providing any context (**NO_CTX**). These two baselines can be considered diagnostic lower bounds.

2.4 Primary Router: Activation as Feature

Feature Extraction: To train our classifier, instead of looking only at the output produced by the language model, we use a forward hook and extract features for every layer of the model for the last token of the prompt. We use the Naive Prompt of our baseline for feature extraction. Based on previous research, the last prompt token has shown characteristics of carrying information for the entire input sequence (Alain & Bengio, 2017; Wendler et al., 2024).

For an input instance $x = (q, D, p)$ with documents D, and prompt template p, we extract features for every layer of the model from Hidden States (H), and the MLP Outputs (M) of each layer l of the language model.

Training: We train multinomial logistic regression models and perform a sweep over all layers (l). For each layer of the language model, we developed two distinct logistic regression models: one utilising H features and the other utilising M features. The specific layer and feature set (either H or M) that achieved the highest classification accuracy on the validation set was selected for the final router (r).

2.5 Secondary Router: Attention Weights as Feature

To compare with our first router, we create an additional router, a lightweight model based on attention weights that is trained as a classifier on features derived from the alignment between the retriever score distribution and the LM's attention over documents. This is motivated by previous work exploring attention weights as proxy signals for relevance and faithfulness (Tian et al., 2026).

Feature Extraction: We extract seven scalar features per layer from the last query token's attention row over retrieved document segments: the top-1 attention mass, top-2 attention mass, the gap between them, attention entropy, retriever score alignment, attention coverage, and the attention margin.

Training: Similar to our main router discussed in 2.4, we train each layer on the validation data, and based on the highest accuracy, we select the particular layer for training on the training data. The training configuration is identical to our primary router.

2.6 Language Models

For our experiment, to ensure generalisation of our method, we chose 16 different transformer-based language models spanning different parameter sizes (90M to 32B), architectural designs, and training methods summarised in Table 4 in the Appendix.

2.7 Implementation Details

All experiments were conducted using NVIDIA A100 (80GB) GPUs hosted on a High-Performance Computing (HPC) cluster. For training our routers, we used Multinomial Logistic Regression along with StandardScaler with Sigmoid calibration. For all of our experiments, we use greedy decoding with a fixed maximum length of 64 tokens to maintain consistency across models. Only for the language model gpt-oss-20b, we keep an exception.

Table 1: Comparison of accuracy with prompt-based, RAG-specialised, and surface-text baselines. ATTR, KRE, FaithRAG, OPIN_INSTR, Vanilla, and NO_CTX are prompt-based baselines. RA denotes our Router (Activation Features). For FAR, lower values are better.

	Prompt-based Techniques						Ours		
Model	**ATTR** [Zhou et al., 2023]	**KRE** [Ying et al., 2024]	**FaithRAG** [Zhang et al., 2025b]	**OPIN_INSTR** [Zhou et al., 2023]	**Vanilla**	**NO_CTX**	**Acc.**	**Macro-F1**	**FAR**
Falcon-H1-Tiny-90M	0.34	0.34	0.35	0.34	0.33	0.33	0.69	0.68	0.13
Falcon3-7B-1.58bit	0.35	0.35	0.31	0.33	0.22	0.32	0.76	0.73	0.12
OLMo-2-1B	0.31	0.36	0.33	0.33	0.29	0.33	0.75	0.73	0.14
OLMo-2-1B-SFT	0.35	0.32	0.31	0.35	0.27	0.32	0.77	0.76	0.11
OLMo-2-1B-DPO	0.35	0.32	0.31	0.35	0.27	0.32	0.77	0.76	0.11
OLMo-2-1B-Instruct	0.36	0.33	0.34	0.33	0.28	0.33	0.78	0.78	0.12
OLMo-3-7B-Instruct	0.34	0.34	0.36	0.34	0.34	0.33	0.81	0.81	0.13
OLMo-3.1-32B-Instruct	0.64	0.64	0.67	0.54	0.44	0.33	0.90	0.90	0.06
Qwen3.5-2B	0.33	0.34	0.33	0.34	0.33	0.33	0.85	0.80	0.08
Qwen3-4B-Instruct	0.59	0.80	0.62	0.80	0.51	0.33	0.89	0.88	0.08
Qwen3.5-9B	0.34	0.32	0.34	0.33	0.33	0.36	0.91	0.89	0.07
Qwen3.5-27B	0.50	0.37	0.36	0.35	0.34	0.33	0.91	0.90	0.08
Llama-3.2-3B	0.34	0.34	0.33	0.34	0.32	0.33	0.82	0.81	0.09
Phi-mini-MoE-Instruct	0.42	0.42	0.37	0.44	0.34	0.33	0.83	0.82	0.11
Granite-3.1-8B-Instruct	0.49	0.50	0.55	0.39	0.34	0.33	0.88	0.87	0.07
gpt-oss-20b	0.80	0.77	0.82	0.61	0.34	0.24	0.82	0.81	0.07

	Specialised and Surface-text Baselines			
	Self-RAG [Asai et al., 2024]	**Chat-QA-1.5** [Liu et al., 2024]	**TF-IDF + LR**	**NLI DeBERTa**
Acc.	0.340	0.370	0.356	0.369

Table 2: Comparison of average accuracy of the best performing baselines (ATTR, KRE, FaithRAG, and OPIN_INSTR) and our router-based model (RA) on the top three highest accuracy language models. The table also compares category-wise accuracy for each language model in correctly classifying Answer, Refuse, and Conflict states. For each metric, the best value is highlighted (Higher value is better for Acc. and Macro-F1. Lower value is better for FAR).

Model	Overall						Answer		Refuse		Conflict	
	Acc.		Macro-F1		FAR							
	Base	RA	Base	RA	Base	RA	Base	RA	Base	RA	Base	RA
gpt-oss-20b	0.73	**0.82**	0.75	**0.81**	0.19	**0.07**	0.66	**0.69**	**0.94**	0.93	0.58	**0.84**
Qwen3-4B-Instruct	0.70	**0.88**	0.70	**0.88**	0.28	**0.08**	0.73	**0.85**	0.66	**0.87**	0.76	**0.94**
Olmo-3.1-32B-Instruct	0.61	**0.90**	0.70	**0.90**	0.19	**0.06**	0.41	**0.86**	**0.97**	0.93	0.44	**0.90**

As this is the only reasoning model in our test, and it generates reasoning traces and tokens, along with the actual answer, the maximum length of 1024 tokens was used. We additionally evaluate selective prediction at 80% and 50% coverage, with full results reported in (Appendix I).

3 Results

3.1 Model-wise Accuracy

Our primary findings are presented in Table 1. Across all 16 models, our activation-feature router (RA in the table) consistently outperforms all prompt-based and RAG-specialised baselines on accuracy (we also report our Macro-F1, and FAR). We additionally evaluated specialised RAG models, **Self-RAG** (Asai et al., 2024) and **Chat-QA-1.5** (Liu et al., 2024), which gained overall accuracies of 0.34 and 0.37, respectively, indicating near-chance performance. The surface-text baselines also remain near chance: **TF-IDF+LR** achieves 0.356 accuracy and 0.352 Macro-F1, while zero-shot **NLI DeBERTa** achieves 0.369 accuracy and 0.338 Macro-F1.

We highlight an important pattern where most prompt-based baselines produce accuracy near chance (0.33) for the majority of tested models. To further understand why, we analyse the failure cases and we observe that smaller and non-instruction-tuned models tend to *always answer*, ignoring refusal or conflict instructions entirely which indicates that the instruction following capability required to produce structured refusal behaviour is simply absent. Second, larger instruction-tuned models that do follow refusal instructions tend to *over-refuse*: they identify insufficient evidence correctly but also refuse on ANSWER instances, collapsing into a two-class behaviour that inflates REFUSE predictions at the expense of ANSWER and CONFLICT. This over-refusal pattern is consistent with findings in prior work (Röttger et al., 2024).

We present further evidence of this in Table 2, considering the top-performing models in terms of overall accuracy. We further note that neither failure mode is fixable by prompt engineering alone, as they reflect model-level behavioural biases rather than prompt interpretation errors. This further motivates our approach, where we are able to obtain information directly from the internal activations before generation begins. We note that two broad patterns show up consistently.

We also highlight the NO_CTX accuracy for all 16 models. Across the full model suite, NO_CTX accuracy ranges from 0.24 to 0.36, confirming that models cannot reliably recover the substituted answers from parametric knowledge alone. Notably, even the largest models tested (OLMo-3.1-32B-Instruct, gpt-oss-20b) do not exceed 0.33 and 0.24 respectively in the NO_CTX condition, suggesting that entity substitution is effective across model scales. We therefore treat the NO_CTX condition as a leakage audit.

3.2 Layer Probe

For each model, we train each layer using hidden-state, MLP, and attention-weight features. We visualise and compare all the layers of the language models and the validation accuracy for each layer in Figure 1.

The heatmap portion in Figure 1 demonstrates the depth and region within the language model, based on the layer on which the best layer based on the extracted features exists for each model. For most of the models, we can see peak dominance mostly in the middle region. Although hybrid and quantised structured models, Falcon-H1-Tiny-90M and Falcon3-7B-1.58bit, show different characteristics in terms of best-layer performance, based on the extracted feature.

4 Mechanistic Analysis

4.1 Router-type hierarchy: hidden state is usually best

A consistent result is that hidden-state features are the strongest in almost all the models, followed by MLP feature-based routers and attention-weight-feature-based ones. Since hidden states reflect the integrated residual-stream representation after passing through the attention and MLP blocks, they are expected to provide more features for our routers. MLP outputs reflect only the feedforward branch, and attention weights primarily indicate where the model attends rather than the full content of what has been computed. As a result, hidden states in most cases provide rich and transferable routing signals.

4.1.1 The OLMo-2-1B base exception

For OLMo-2-1B base, the MLP router slightly outperforms the hidden-state router. However, the behaviour reverses in the SFT, DPO, and Instruct variants. This indicates that although in base variants the MLP router has rich distinguishing features, after post-training, it becomes more integrated into the hidden states without altering the depth region.

Figure 1: Layer-wise validation accuracy of three routing strategies across sixteen language models. Each subplot shows validation accuracy (y-axis) as a function of transformer layer index (x-axis) for three router types: Hidden State Router (blue), MLP Router (red), and Attention Weight Router (green). Filled circles mark the best-performing layer per router. The heatmap strip below each plot shows the relative best-layer position (best layer / total layers), where darker colours indicate earlier layers and brighter colours indicate later layers. H, M, and A denote Hidden State, MLP, and Attention routers, respectively. The annotation box reports the total layer count and peak validation accuracy with its layer index for each router. An enlarged version of this figure is provided in Figure 7, Appendix N.

4.2 Relative layer position: where models peak

Overall, among the 16 evaluated language models, most of the feature-extracted models show an inverted-U trajectory that peaks in the middle-to-upper-middle portion of the network. The relative position of the peak remains broadly similar, irrespective of the size of the language models.

4.2.1 Falcon-H1-Tiny-90M and OLMo-3.1-32B-Instruct peak unusually early

A distinct result in the heatmap portion of the Figure 1 is that Falcon-H1-Tiny-90M, the smallest of the language models tested, peaks earlier than all other models for all three router types. Because of its small structure and hybrid Mamba/attention design, the behaviour is different compared to standard dense transformers.

For OLMo-3.1-32B-Instruct, it is evident that task-specific information is linearly decodable in the early layers as well. The instruction-tuned OLMo model is initialised from an OLMo-3-Think-SFT checkpoint and is then optimised for non-reasoning chat, multi-turn preference alignment, and function calling (Olmo et al., 2026). This optimisation could be a factor to make the model linearly accessible in the early layers, and can be hypothesised based on our findings.

4.3 Model-specific sources of layerwise variation

The OLMo-2 family shows the most stable pattern across variants. The base, SFT, DPO, and Instruct models have nearly identical curve shapes and peak in a similar layer range, with fine-tuning mainly increasing peak accuracy rather than moving the optimal layer. This suggests that the backbone architecture, more than the post-training stage, determines where the most informative representations lie.

Falcon3-7B-1.58bit is unusual in that its attention router peaks extremely early, while its hidden-state and MLP routers retain a more standard middle-layer optimum. A plausible explanation is that low-bit quantisation affects attention-derived signals more strongly than activation-derived signals, making attention weights less stable across depth.

4.3.1 Granite-3.1-8B-Instruct: why the curve is more irregular

Granite-3.1-8B-Instruct exhibits the strongest layer-to-layer fluctuation among the dense models. Several architectural features likely contribute to this behaviour. First, Granite uses explicit scaling terms such as `residual_multiplier`, `attention_multiplier`, `embedding_multiplier`, and `logits_scaling`, which suggest a more tightly regulated signal flow through the network.

Second, Granite is one of the deepest models in the comparison, which increases the chance that representations are repeatedly re-encoded rather than smoothly refined. Third, it is the only model here with non-zero attention dropout. Finally, its grouped-query attention may reduce the distinctiveness of raw attention outputs relative to the hidden state. Together, these properties likely explain why Granite shows a more jagged and inconsistent layer-wise profile. A comparison of the architectures of the tested models is presented in Figure 6 in the Appendix.

5 Ablation

We evaluate the robustness of our router design through cross-model transfer, prompt robustness, hidden state interventions, and additional diagnostic ablations.

5.1 Cross-Model Transferability

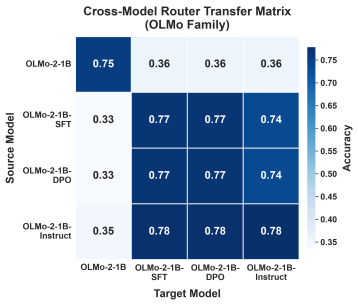

Figure 2: OLMo Model Transferability.

To conduct cross-model transfer within the OLMo-2 family (OLMo-2-1B, OLMo-2-1B-SFT, OLMo-2-1B-DPO, and OLMo-2-1B-Instruct), we train a router on one model and test it on the other corresponding models. Figure 2 indicates the results. Routers trained on the fine-tuned variants SFT, DPO, and Instruct transfer well across each other, indicating that these model variants preserve a similar geometry for our task classification. In contrast, transfer to and from the base variant does not generalise, showing that fine-tuning introduces a linear representation that the base variants lack.

5.2 Prompt Robustness

To evaluate the robustness of our system on different prompts, we train our router on one specific prompt and test the accuracy on different prompts on a subsample of validation data. The details of the prompt, along with the produced heatmap, are provided in Appendix D. By analysing Figure 4, we can see that the performance is largely stable across Prompts P0 to P3, where minimal lexical variations are applied. The performance drops on Prompt P4, which uses question-first ordering, suggesting that the system is sensitive to the position

of the questions within the context of the prompt. When a style wrapper (Prompt P5) is applied, it affects smaller base models more than larger ones.

5.3 Hidden-state Patching

To validate that the router layer is a functionally active control point, we perform a hidden-state patching experiment following Meng et al. (2022). For each of 100 source–target pairs drawn from different triage classes (ANSWER, CONFLICT, and REFUSE), we replace the target example's hidden state at the selected router layer with the corresponding hidden state from the source example, then measure how downstream router predictions shift. The aim of this is to establish that the selected layer is a fair control probe such that intervening at this layer is sufficient to reshape the downstream prediction. This provides evidence that the selected layer acts as a functional control point for downstream routing behaviour.

We present the results in Figure 5 across 11 language models. Patching the best probe layer produces a substantial increase in downstream source-class prediction rates across all patch directions and models. The effect is strongest for refuse→answer and conflict→answer directions, consistently exceeding 0.70 in instruction-tuned models. The answer→refuse direction produces smaller but reliable shifts, above chance in all tested models. These results confirm that the selected middle layer is directly relevant to the triage decision.

We additionally test whether the intervention propagates to free-form generation, rather than only changing the downstream router prediction. Results show that the identified hidden-state signal can influence generated text, but does not provide a complete explanation of the model's generation process (results presented in Appendix M).

5.4 Binary vs. Three-Way Formulation

We compare the three-way router with a simpler binary version that does not treat CONFLICT as a separate class. The two settings achieve similar overall accuracy (0.829 vs. 0.832), but the three-way router gives better Macro-F1 (0.829 vs. 0.811) and a lower FAR (0.104 vs. 0.127). This suggests that keeping CONFLICT as a separate class helps reduce false answers without hurting overall performance. Full model-wise results are provided in Appendix L. In addition, as a control, we train the probe with randomised labels. Accuracy drops to 0.332 ± 0.013 on average, close to chance; full results are reported in Appendix K.

5.5 Long-Context Feature-Source Ablation

We compare the final prompt-token representation with two simple pooling strategies: averaging the final content-token representation from each document (Doc-end mean), and averaging the representations of all document content tokens (Doc-token mean) on Qwen3-4B-Instruct and Llama-3.2-3B. Performance generally drops as the context becomes longer, while the final-token representation remains stronger than the pooling alternatives. Results are presented in Table 7 of the Appendix.

Overall, the final-token representation remains stronger than the two simple pooling alternatives, but its performance declines as the number of retrieved documents increases. The drop is larger when a router trained at $k = 5$ is tested on longer contexts, indicating sensitivity to context-length distribution shift.

5.6 Natural-Domain Transfer

We additionally evaluate zero-shot transfer to RAGTruth (Niu et al., 2024), where routers trained without any RAGTruth supervision achieve 0.84–0.92 accuracy across four model families; full results and limitations are reported in Appendix G.

6 Related Work

Recent work on improving RAG faithfulness has primarily focused on different prompt-based techniques and specialised RAG models (Zhou et al., 2023; Ying et al., 2024; Zhang et al., 2025b; Asai et al., 2024; Liu et al., 2024). While these methods are effective in some cases, this comes with additional token, reasoning, and text generation costs. Moreover, prompt optimisation does not necessarily generalise across language models.

Research by Kadavath et al. (2022) suggests that model outcomes can be predicted internally. Extending on this, recent interpretability work also shows that hidden states can encode useful decision signals (Huang et al., 2025; Zhou et al., 2026; Liu et al., 2025b; Wang et al., 2025). Researchers have shown how these signals can be utilised to probe correctness during reasoning, detect hallucinations from single responses, and study how retrieved and parametric knowledge interact inside RAG systems (Alain & Bengio, 2017; Zhang et al., 2025a; Sriramanan et al., 2024; Sun et al., 2025; Zhou et al., 2026).

Our work is also related to recent approaches that use internal model states for RAG decisions. Probing-RAG (Baek et al., 2025) uses intermediate hidden states for a binary retrieve-more decision during generation, while SeaKR (Yao et al., 2025) uses internal uncertainty to guide retrieval and reasoning. RedeEP (Sun et al., 2025) instead detects hallucinations after generation by analysing parametric and retrieved-context contributions. LLM Microscope (Liu et al., 2025a) studies incorrect context using a single perturbed document, whereas our CONFLICT setting captures disagreement between retrieved sources. In contrast, our router makes a three-way decision before generation begins, with CONFLICT treated as a separate prediction target.

In addition to prompting and verifier-based approaches, we extend interpretability research and study whether the RAG evidence state itself ANSWER, REFUSE, or CONFLICT is linearly decodable from internal activations before generation. Previous literature has not really addressed the three-way problem through layer probing and mechanistic exploration. Besides, work on activation steering (Rimsky et al., 2024; Subramani et al., 2022) suggests that vector steering can be a potential research direction for studying language models; we expand on it to explore RAG faithfulness. We conducted an ablation in applying Contrastive Activation Steering in Appendix C, and aim to conduct future research on it.

7 Conclusion

In this paper, our focus was to examine whether language models internally represent the evidential state of their retrieved context before generating a response. Our experiments suggest that this is indeed possible and that the internal representations contain the necessary information. A simple logistic regression-based linear model trained on activations from a single hidden layer is reliably capable of distinguishing whether retrieved evidence is sufficient to answer, insufficient, or contradictory, across 16 language models spanning diverse architectures, sizes, and training regimens. We showcase that prompt-based approaches to RAG usually fail mostly because of model-level behavioural biases, suggesting that prompting alone is perhaps suboptimal. We observe that smaller models tend to always answer, larger models tend to over-refuse, and neither of these is capable of handling conflicting evidence. Our approach, on the other hand, bypasses these limitations entirely by operating on internal representations before generation begins, achieving up to 0.91 accuracy and reducing false answer rates by up to 75% relative to the strongest prompt-based baseline with no additional token budget, while remaining extremely computationally efficient.

We also provide mechanistic findings, where we show that the most informative representations for the triage decision emerge consistently in the middle layers of the network, a pattern that holds across model families and is stable under post-training. Hidden-state patching confirms these layers are important control points and targeted intervention there reliably reshapes downstream behaviour. Among the three evidence states, *Conflict* is the most practically dangerous as models generate confidently from contradictory documents, and still our approach shows that it is linearly decodable internally at the right layer.

Acknowledgements

This work was partially supported by the Mike Hudson Foundation, a non-profit AI donor fund, the City St George's, University of London, and was supported in part by the Alan Turing Institute under Fundamental Research Project No. PP00029.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2017. URL https://openreview.net/forum?id=ryF7rTqgl.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

Ingeol Baek, Hwan Chang, ByeongJeong Kim, Jimin Lee, and Hwanhee Lee. Probing-RAG: Self-probing to guide language models in selective document retrieval, April 2025. URL https://aclanthology.org/2025.findings-naacl.181/.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. TRUE: Re-evaluating factual consistency evaluation. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3905–3920, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.287. URL https://aclanthology.org/2022.naacl-main.287/.

Zihan Huang, Junda Wu, Rohan Surana, Raghav Jain, Tong Yu, Raghavendra Addanki, David Arbour, Sungchul Kim, and Julian McAuley. Traceable and explainable multimodal large language models: An information-theoretic view. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=pQm66IPmeE.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. Sufficient context: A new lens on retrieval augmented generation systems. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Jjr2Odj8DJ.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

Jiarui Liu, Jivitesh Jain, Mona Diab, and Nishant Subramani. Llm microscope: What model internals reveal about answer correctness and context utilization, 2025a. URL https://arxiv.org/abs/2510.04013.

Qi Liu, Haozhe Duan, Jiaxin Mao, and Ji-Rong Wen. How do large language models understand relevance? a mechanistic interpretability perspective. *ACM Trans. Inf. Syst.*, 44(2), December 2025b. ISSN 1046-8188. doi: 10.1145/3774942. URL https://doi.org/10.1145/3774942.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. Chatqa: surpassing gpt-4 on conversational qa and rag. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10862–10878, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.585. URL https://aclanthology.org/2024.acl-long.585/.

Team Olmo, :, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2026. URL https://arxiv.org/abs/2512.13961.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. pp. 5377–5400, June 2024. doi: 10.18653/v1/2024.naacl-long.301. URL https://aclanthology.org/2024.naacl-long.301/.

Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. Llm-check: Investigating detection of hallucinations in large language models. *Advances in neural information processing systems*, 37:34188–34216, 2024.

Nishant Subramani, Nivedita Suresh, and Matthew Peters. Extracting latent steering vectors from pretrained language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 566–581, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi:

10.18653/v1/2022.findings-acl.48. URL https://aclanthology.org/2022.findings-acl.48/.

ZhongXiang Sun, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Weijie Yu, Yang Song, and Han Li. RedeEP: Detecting hallucination in retrieval-augmented generation via mechanistic interpretability. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ztzZDzgfrh.

Yuxing Tian, Fengran Mo, Weixu Zhang, Yiyan Qi, and Jian-Yun Nie. ReAttn: Improving attention-based re-ranking via attention re-weighting. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 1282–1295, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-386-9. doi: 10.18653/v1/2026.findings-eacl.66. URL https://aclanthology.org/2026.findings-eacl.66/.

Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, Ruicheng Yin, Changze Lv, Xiaoqing Zheng, and Xuanjing Huang. Searching for best practices in retrieval-augmented generation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17716–17736, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.981. URL https://aclanthology.org/2024.emnlp-main.981/.

Yuhao Wang, Ruiyang Ren, Yucheng Wang, Wayne Xin Zhao, Jing Liu, Hua Wu, and Haifeng Wang. Unveiling knowledge utilization mechanisms in llm-based retrieval-augmented generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, pp. 1262–1271, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715921. doi: 10.1145/3726302.3730112. URL https://doi.org/10.1145/3726302.3730112.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Do llamas work in English? on the latent language of multilingual transformers. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15366–15394, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.820. URL https://aclanthology.org/2024.acl-long.820/.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=auKAUJZMO6.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Liu Weichuan, Lei Hou, and Juanzi Li. SeaKR: Self-aware knowledge retrieval for adaptive retrieval augmented generation, July 2025. URL https://aclanthology.org/2025.acl-long.1312/.

Jiahao Ying, Yixin Cao, Kai Xiong, Long Cui, Yidong He, and Yongbin Liu. Intuitive or dependent? investigating LLMs' behavior style to conflicting prompts. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4221–4246, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.232. URL https://aclanthology.org/2024.acl-long.232/.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ZS4m74kZpH.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification. In *Second Conference on Language Modeling*, 2025a. URL https://openreview.net/forum?id=O6I0Av7683.

Qinggang Zhang, Zhishang Xiang, Yilin Xiao, Le Wang, Junhui Li, Xinrun Wang, and Jinsong Su. FaithfulRAG: Fact-level conflict modeling for context-faithful retrieval-augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21863–21882, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1062. URL https://aclanthology.org/2025.acl-long.1062/.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. Context-faithful prompting for large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14544–14556, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.968. URL https://aclanthology.org/2023.findings-emnlp.968/.

Youchao Zhou, Heyan Huang, Yicheng Liu, Rui Dai, Xinglin Wang, Xingchen Zhang, Shumin Shi, and Yang Deng. Do retrieval augmented language models know when they don't know? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35158–35166, 2026.

A Artefact Robustness

To test the robustness of our router and assess its reliability on formatting artefacts in the retrieved documents, we sample 20 examples from each of the ANSWER, REFUSE, and CONFLICT classes, for a total of 60 instances. We use automatic back-translation through a non-English language as a pivot using MarianMT/OPUS-MT and translate it back to English, which changes the structure of the retrieved document while preserving the semantics. We test our router, trained on the original data, on this rewritten set of retrieved documents using two language models. The router maintains an accuracy of **0.83** and **0.80** for Qwen3-4B-Instruct and Granite-3.1-8B-Instruct, respectively, suggesting that the trained router is independent of lexical and stylistic cues.

B Router-Design

Table 3: Validation accuracy comparison between the baseline setup and ablation variants across three models. H means hidden state features, M means MLP block outputs from transformer. Top N suggests selecting the top-performing layers for training and testing the router. All-H/M suggests combining all the layers for training and testing the router.

Setting	Granite-3.1-8B-Instruct	OLMo-2-1B	Qwen3-4B-Instruct
Previous Val Acc (baseline)	0.90	0.73	0.91
Layer aggregation			
Top3-H	0.91	0.80	0.89
Top3-M	0.90	0.81	0.91
All-H	0.90	0.78	0.91
All-M	0.90	0.80	0.90
Feature combination			
Top1-(H+M)	0.90	0.76	0.91
Classifier capacity			
MLP-Top1-H	0.91	0.83	0.92
MLP-Top1-M	0.90	0.80	0.91

The results presented in Table 3 support our claim that the routing signal is linearly decodable from a selected layer. We evaluate three language models under different architectural and design choices. Aggregating the top few layers or all layers yields only small and inconsistent gains relative to the best single-layer setup, especially for Granite-3.1-8B-Instruct and Qwen3-4B-Instruct, while OLMo-2-1B benefits more from broader aggregation. Combining hidden-state and MLP features also provides limited improvement. Furthermore, we evaluate a non-linear router based on a multilayer perceptron (MLP), and find that the gain is most pronounced for OLMo-2-1B compared to the other models. This suggests that smaller base models may benefit more from a slightly more expressive classifier, possibly because their routing signal is less directly linearly separable across transformer layers.

C Activation Steering

The triage representation by probing can be represented by moving the model along the representation, which should shift its response. We test this state, which can be usable for control, using Contrastive Activation Addition, inspired by the work of Rimsky et al. (2024)

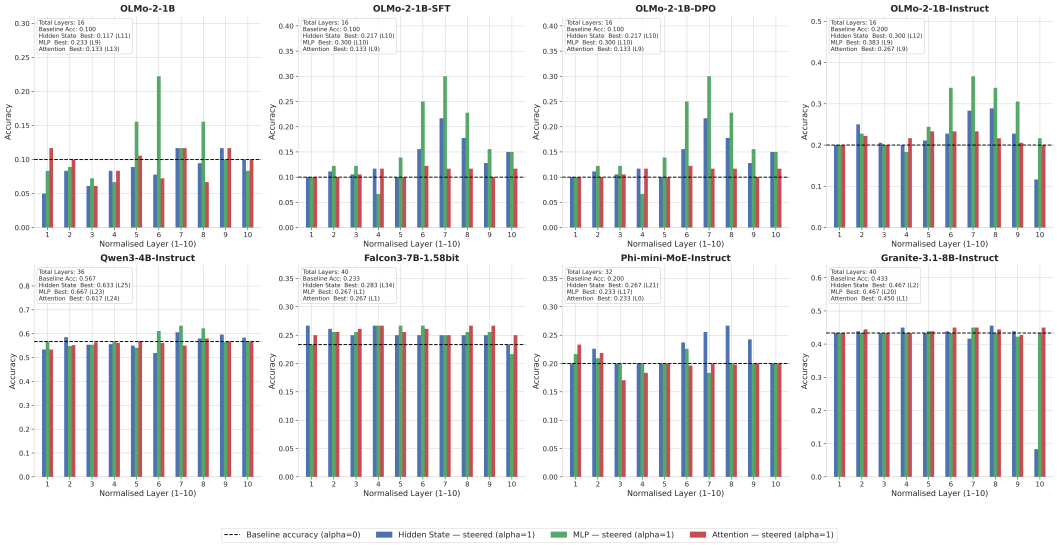

Figure 3: Mean accuracy across language models on a subset of 80 examples (40 Refuse, 40 Conflict) under vector steering on hidden state, MLP output, and attention output. The dashed horizontal line indicates the baseline accuracy. Layers are downsampled to 10 layers for ease of comparison.

Figure 3 presents our findings on steering activation. For each model, we compute steering vectors utilising a mean-difference estimator between positive and negative examples at each layer. We construct two vector types, REFUSE-vs-ANSWER and CONFLICT-vs-ANSWER, so that each direction represents how the internal state differs.

We compute vectors at three intervention sites: the residual stream, the MLP output, and the attention output. During generation, for layer l, site s, and steering strength α, we add the steering vector to the activation as

$$\tilde{h}_l^{(s)} = h_l^{(s)} + \alpha v_l^{(s)}.$$

We use $\alpha \in \{0, 1\}$, where $\alpha = 0$ is the unsteered baseline and $\alpha = 1$ is the steered setting, and sweep all layers of each model on a validation subset of 40 refuse and 40 conflict examples.

Baseline prompting and steering did not affect Falcon-H1-Tiny-90M and Llama-3.2-3B, as these models always generated an answer instead of producing refusal or conflict behaviour because of not being instruction-tuned, so we keep these out of our analysis. Figure 3 shows that steering can partially control language model behaviours. Similar to our finding in the layer probe experiment, the strongest effects usually appear in middle-to-late layers. In contrast, in probing, hidden states were typically most decodable, but in steering, the gains mainly come from MLP features. On the other hand, attention steering is generally weaker and less consistent throughout all the tested language models.

The variation of gains after steering also varies by model and architecture. In OLMo, post-training increases steerability without changing the depth of the layer position, suggesting that it strengthens an existing feature rather than relocating it. In instruction-tuned models like Qwen and Granite, gains are smaller, likely because the baseline prompt already implemented stronger refusal/conflict behaviour. Very late-layer interventions can also hurt performance, especially in Granite, indicating that steering at the final layers can degrade performance badly. Overall, these results suggest that language models in RAG setups are both decodable and partly controllable.

D Prompt Templates & Robustness

Figure 4 provides the full prompt-level result. Performance remains relatively stable across P0–P3, where the changes are mainly lexical, while a larger drop is observed for P4 with question-first ordering. P5 also affects some of the smaller models more strongly. This suggests that the router is reasonably robust to small prompt variations, but can still be affected by larger changes in prompt structure.

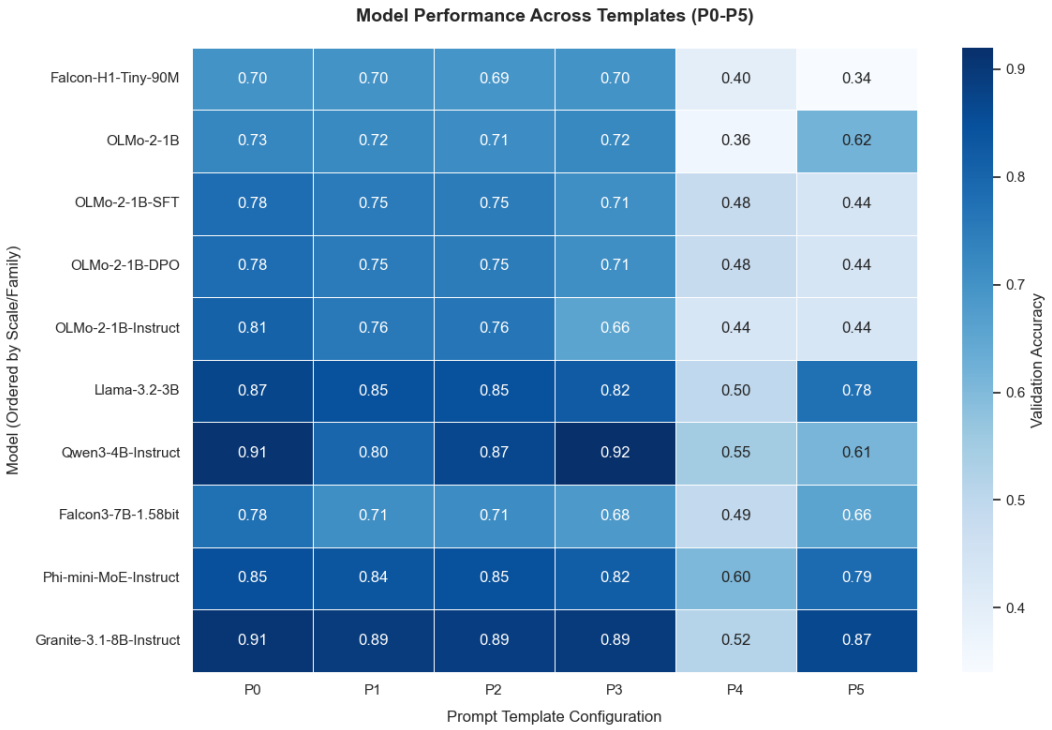

Figure 4: Prompt Robustness Heatmap

The prompt set consists of six variants, `P0` to `P5`, where each `P` denotes a different prompt template. The complete baseline prompts and deterministic output-parsing configuration are provided with the released code.

- **P0 (Baseline):**

  ```
  You are a helpful assistant. Use only the provided documents to answer.

  [Document 1]:


  Question:
  Answer:
  ```

- **P1 (Instruction Paraphrase):**

  ```
  Answer using only the provided documents. Do not use outside knowledge.

  [Document 1]:


  Question:
  Answer:
  ```

- **P2 (Boundary Change):**

```
You are a helpful assistant. Use only the provided documents to answer.

Passage 1:


Question:
Answer:
```

- **P3 (No System Grounding):**

```
[Document 1]:


Question:
Answer:
```

- **P4 (Question-First):**

```
You are a helpful assistant. Use only the provided documents to answer.

Question:

[Document 1]:


Answer:
```

- **P5 (Structured Format):**

```
You are a helpful assistant. Use only the provided documents to answer.

<doc id="1">
</doc>

<question>...</question>
<answer>
```

E Model Details

Table 4 summarises the 16 language models used in our experiments, covering different model families, parameter sizes, training stages, and architectures.

Table 4: Model Suite Overview

Model Family	Variant	Parameters	Type	Architecture
Falcon	Falcon-H1-Tiny-90M	90M	Instruct	Hybrid
OLMo	OLMo-2-1B	1B	Base	Dense
OLMo	OLMo-2-1B-SFT	1B	SFT	Dense
OLMo	OLMo-2-1B-DPO	1B	DPO	Dense
OLMo	OLMo-2-1B-Instruct	1B	Instruct	Dense
Qwen	Qwen3.5-2B	2B	Base	Dense
Llama	Llama-3.2-3B	3B	Base	Dense
Qwen	Qwen3-4B-Instruct	4B	Instruct	Dense
Falcon	Falcon3-7B-1.58bit	7B	Instruct	quantised
OLMo	OLMo-3-7B-Instruct	7B	Instruct	Dense
Phi	Phi-mini-MoE-Instruct	7.6B	Instruct	Sparse MoE
Granite	Granite-3.1-8B-Instruct	8B	Instruct	Dense
Qwen	Qwen3.5-9B	9B	Base	Dense
GPT-OSS	gpt-oss-20b	20B	Reasoning	Dense
Qwen	Qwen3.5-27B	27B	Base	Dense
OLMo	OLMo-3.1-32B-Instruct	32B	Instruct	Dense

F Hidden-State Patching

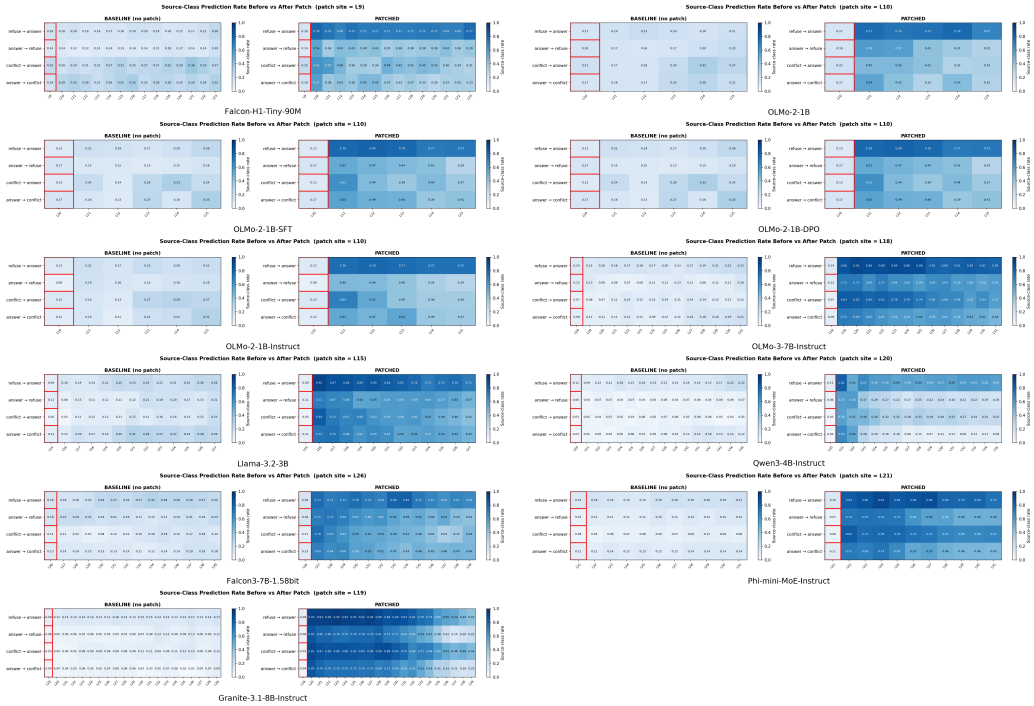

Figure 5: Patching on different tested language models. Label on y-axis, such as *refuse* → *answer* means that a hidden state from a *refuse* example is inserted into an *answer* example at the patch site. The x-axis shows the patch layer and all subsequent layers, while each cell reports the proportion of examples whose downstream router prediction matches the source class. Higher values after patching indicate that the selected layer acts as a causal control point.

G Zero-Shot Transfer to RAGTruth

To examine whether the routing signal transfers beyond our controlled benchmark, we evaluate the trained routers on RAGTruth (Niu et al., 2024) without any additional training or adaptation on RAGTruth. We use the MS MARCO-derived portion of RAGTruth and evaluate four model families. The results are shown in Table 5.

Model	Accuracy	Macro-F1
Falcon3-7B-1.58bit	0.91	0.63
Granite-3.1-8B-Instruct	0.92	0.65
Llama-3.2-3B	0.84	0.65
Qwen3-4B-Instruct	0.91	0.63

Table 5: Zero-shot transfer to RAGTruth. The routers are evaluated without any RAGTruth training signal or in-domain adaptation.

All four routers retain strong accuracy under this transfer setting, with accuracy ranging from 0.84 to 0.92. This provides evidence that the learned routing signal is not limited to the controlled construction used in our main experiments and can transfer to naturally occurring RAG examples.

There is an important limitation to this evaluation. First, RAGTruth does not provide a separate annotation for conflicting retrieved evidence. We therefore evaluate transfer in a binary setting, and this experiment does not test transfer of the three-way ANSWER–REFUSE–CONFLICT formulation directly. We treat the RAGTruth experiment as evidence of natural-domain transfer rather than as a replacement for the controlled three-way evaluation. A fully annotated naturalistic benchmark containing answerable, insufficient, and conflicting evidence would allow a more direct test of three-way transfer.

H Additional Results

Figure 6 provides an architectural comparison of the models used in the main experiments. The figure provides additional context for the model-specific layer-wise differences.

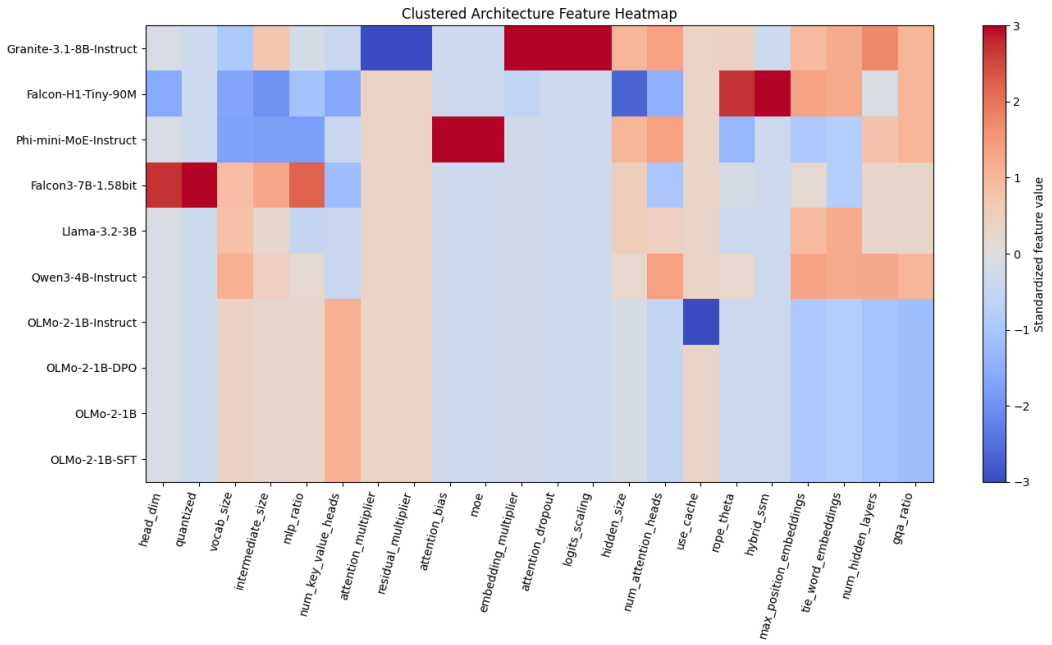

Figure 6: Architectural Feature Heatmap

I Calibration and Selective Abstention

We evaluate whether calibrated router confidence can support selective abstention by retaining only the most confident predictions. Table 6 reports Accuracy and False Answer Rate (FAR) at 80%, and 50% coverage. On average, reducing coverage from 100% to 80% increases accuracy while reducing FAR. At 50% coverage, average accuracy reaches 0.943 and FAR decreases to 0.013.

Model	**Acc.@80%**	**FAR@80%**	**Acc.@50%**	**FAR@50%**
Falcon-H1-Tiny-90M	0.754	0.064	0.833	0.021
Falcon3-7B-1.58bit	0.819	0.080	0.901	0.032
Llama-3.2-3B	0.904	0.046	0.957	0.012
Phi-mini-MoE-Instruct	0.904	0.054	0.964	0.017
OLMo-2-1B	0.827	0.058	0.890	0.024
OLMo-2-1B-SFT	0.860	0.055	0.922	0.012
OLMo-2-1B-Instruct	0.895	0.060	0.960	0.022
Qwen3-4B-Instruct	0.963	0.024	0.993	0.005
OLMo-3.1-32B-Instruct	0.947	0.023	0.988	0.002
Qwen3.5-2B	0.878	0.035	0.944	0.007
Qwen3.5-9B	0.943	0.032	0.965	0.005
Qwen3.5-27B	0.967	0.018	0.994	0.002
Average	**0.888**	**0.046**	**0.943**	**0.013**

Table 6: Accuracy and FAR across confidence thresholds.

J Long-Context Feature-Source Ablation

Table 7 shows that the final-token representation remains stronger than the document-level pooling alternatives as the number of retrieved documents increases, although performance generally decreases for longer contexts. The additional drop when a router trained at $k = 5$ is evaluated on larger values of k also indicates sensitivity to context-length distribution shift.

Table 7: Long-context ablation as the number of retrieved documents increases. Matched-k trains and tests at the same context size, while train $k = 5$ trains only on five-document contexts and evaluates on progressively longer contexts. FAR and Conflict-F1 are reported at $k = 20$.

Model	**Setting**	**Feature**	**Acc.** $k = 5$	**Acc.** $k = 10$	**Acc.** $k = 15$	**Acc.** $k = 20$	**FAR** $k = 20$	**Conflict-F1** $k = 20$
Qwen3-4B-Instruct	matched-k	Last token	0.858	0.842	0.804	0.821	0.113	0.843
Qwen3-4B-Instruct	matched-k	Doc-end mean	0.575	0.554	0.454	0.500	0.312	0.639
Qwen3-4B-Instruct	matched-k	Doc-token mean	0.637	0.550	0.533	0.454	0.325	0.608
Llama-3.2-3B	matched-k	Last token	0.838	0.825	0.746	0.717	0.169	0.724
Llama-3.2-3B	matched-k	Doc-end mean	0.588	0.537	0.487	0.517	0.312	0.616
Llama-3.2-3B	matched-k	Doc-token mean	0.613	0.537	0.487	0.450	0.350	0.646
Qwen3-4B-Instruct	train $k = 5$	Last token	0.858	0.846	0.808	0.808	0.181	0.792
Llama-3.2-3B	train $k = 5$	Last token	0.838	0.796	0.738	0.675	0.163	0.632

K Random-Label Control

We also train the probe after randomly shuffling the class labels. Table 8 shows that after randomly shuffling the class labels, average accuracy drops to 0.332 ± 0.013, which is close to chance for the three-way task. The large difference from the real-label results indicates that the probe performance is dependent on meaningful evidence-state labels rather than the classifier simply separating arbitrary classes.

Table 8: Random-label control. Real-label accuracy is taken from the main results in Table 1; random-label accuracy is reported as mean $\pm$ standard deviation.

Model	Real-label Acc.	Random-label Acc.
Granite-3.1-8B-Instruct	0.88	0.333 ± 0.012
OLMo-3.1-32B-Instruct	0.90	0.328 ± 0.012
gpt-oss-20b	0.82	0.338 ± 0.014
Qwen3.5-27B	0.91	0.330 ± 0.009
Qwen3.5-2B	0.85	0.328 ± 0.011
Qwen3.5-9B	0.91	0.326 ± 0.019
Falcon-H1-Tiny-90M	0.69	0.328 ± 0.024
Llama-3.2-3B	0.82	0.335 ± 0.012
OLMo-2-1B	0.75	0.337 ± 0.005
OLMo-2-1B-SFT	0.77	0.333 ± 0.014
OLMo-2-1B-DPO	0.77	0.333 ± 0.014
OLMo-2-1B-Instruct	0.78	0.332 ± 0.011
Qwen3-4B-Instruct	0.89	0.341 ± 0.010
Random-label average	–	$\mathbf{0.332 \pm 0.013}$

L Binary vs. Three-Way Formulation

Table 9 presents the model-wise comparison between the binary and three-way formulations. The two settings have similar average accuracy, but the three-way formulation gives higher Macro-F1 and a lower FAR. This supports treating CONFLICT as a separate state, since doing so improves reliability without substantially reducing overall classification accuracy.

Table 9: Comparison between the three-way and binary router formulations. For FAR, lower values are better. The final column reports CONFLICT accuracy for the three-way router.

	Three-Way			Binary			Three-Way
Model	**Acc.**	**Macro-F1**	**FAR**	**Acc.**	**Macro-F1**	**FAR**	**CONFLICT Acc.**
Granite-3.1-8b-Ins	0.879	0.870	0.072	0.877	0.863	0.099	0.900
OLMo-32B	0.904	0.904	0.062	0.900	0.887	0.072	0.919
GPT-OSS	0.823	0.814	0.075	0.874	0.859	0.097	0.915
Qwen-27B	0.912	0.902	0.080	0.896	0.884	0.091	0.919
Qwen-2B	0.846	0.803	0.083	0.833	0.813	0.125	0.852
Qwen-9B	0.907	0.887	0.071	0.881	0.866	0.096	0.892
Falcon-tiny	0.696	0.684	0.138	0.757	0.713	0.140	0.733
Llama-3.2-3b	0.816	0.814	0.093	0.805	0.783	0.156	0.856
OLMo-2B-1b	0.752	0.731	0.143	0.747	0.717	0.198	0.756
Qwen-3-4B	0.888	0.888	0.081	0.891	0.880	0.101	0.910
Average	**0.829**	**0.829**	**0.104**	**0.832**	**0.811**	**0.127**	**0.850**

M Generation-Level Patching

To test whether hidden-state interventions also affect the model's autoregressive output, we additionally perform generation-level patching. The model receives the same RAG prompt used in the main experiments and generates a free-form response after intervention at the selected layer.

Across models and patching directions, 63–83% of patched generations have Jaccard divergence ≥ 0.30 from their corresponding unpatched generations, while the degeneration rate is ≤ 0.01. Sentence-level comparisons further show a semantic-similarity gain of $+0.47$ to $+0.93$ over the within-class control, with BERTScore F1 ≥ 0.87.

N Enlarged Figures

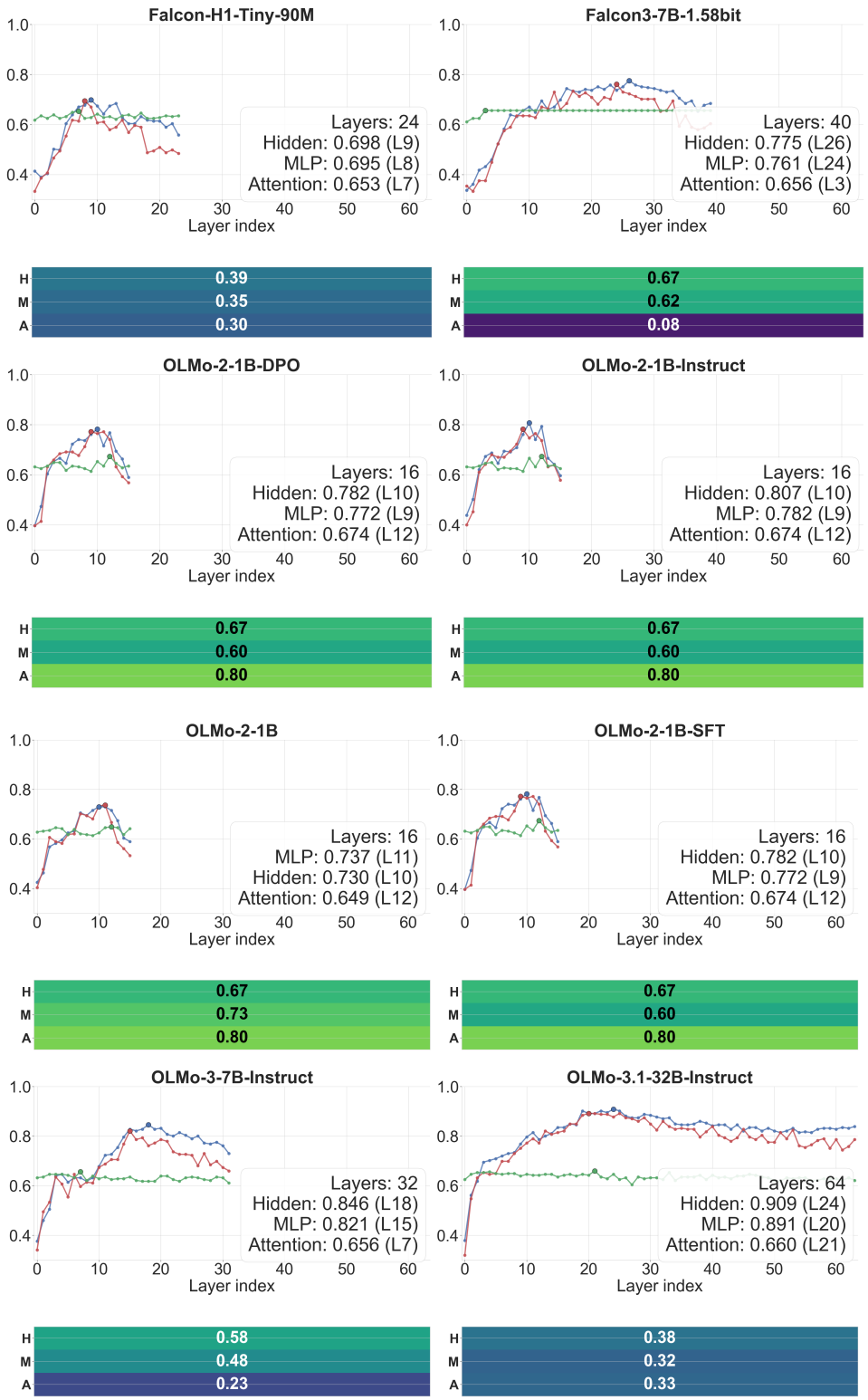

Figure 7: Zoomed layer-wise validation accuracy

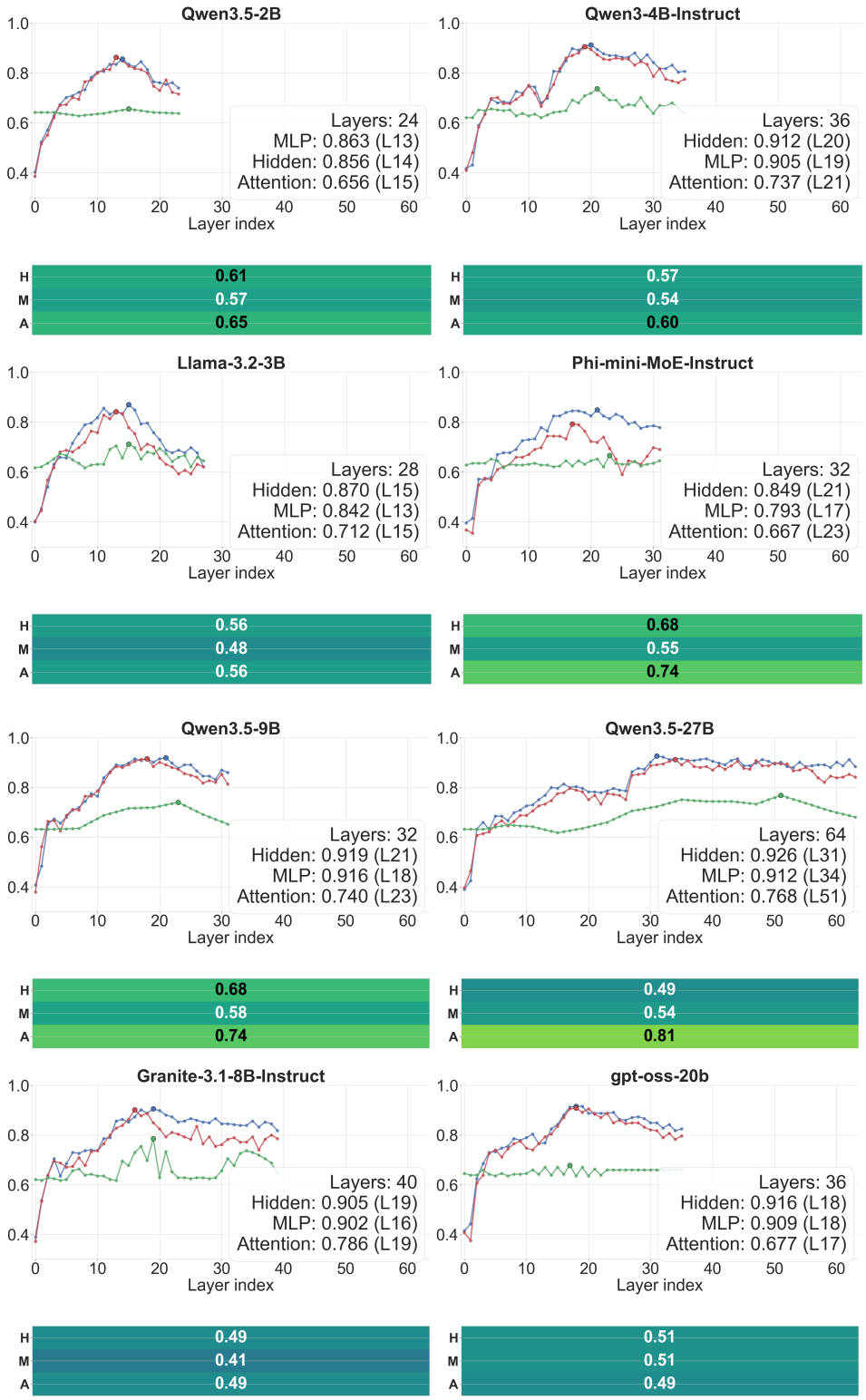

Figure 7: Zoomed layer-wise validation accuracy (continued)