The Hard Decision Layer: Evidence for Committed Inference in Transformers

Anonymous ACL submission

Abstract

We investigate where and how transformer-based language models commit to predictions in multiple-choice question answering. We identify the *Hard Decision Layer* (HDL), a natural architectural property where answer option rankings stabilize abruptly during inference. Empirical validation across four language models (Qwen, Llama, Granite, Mistral) and four benchmark datasets demonstrates consistent HDL emergence without learned routing policies. We also show that the HDL is invariant to fine-tuning. Our results reveal striking accuracy improvements at the HDL: up to +0.61 (Qwen on CommonsenseQA), after which performance stabilizes. Systematic ablations on label formats and problem complexity confirm the phenomenon is fundamental to model architecture. These findings offer mechanistic insights into transformer inference and suggest opportunities for efficient reasoning and model steering. All code and results required to reproduce this work are available in `https://anonymous.4open.science/r/hard-decision-layer-E33D/`

1 Introduction

Understanding where and how transformer-based language models commit to their predictions is fundamental to both mechanistic interpretability (Bereska and Gavves, 2024) and computational efficiency. Recent work shows that models allocate computational depth dynamically: information flows through the residual stream (Elhage et al., 2021) and is iteratively refined by attention and feed-forward networks (Geva et al., 2022), with early layers performing vocabulary filtering while later layers refine candidates (Gupta et al., 2025). A key insight is that model predictions stabilize mid-network (Lioubashevski et al., 2024), motivating research into when and how models make their final decisions.

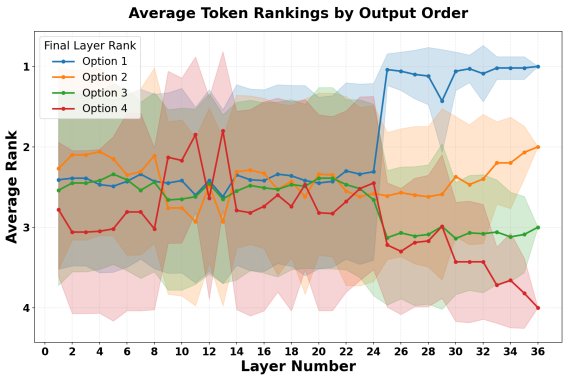


Figure 1: Average token rankings for answer option tokens across transformer layers of the Qwen model. Each line represents one of the four answer options, with average rankings computed at each layer over all questions in the QASC dataset. The faded regions surrounding each ranking line indicate the standard deviation of rankings across the dataset. Per our definition, the *hard decision layer* (HDL) is identified at Layer 25. Following this layer, the average rankings are stabilized all the way until the last layer.

Existing approaches to understanding layer-wise decision-making fall into two categories: dynamic methods that identify when to exit early through learned routing policies (Din et al., 2024), and static analyses that examine post-hoc layer pruning (Fan et al., 2024). However, neither characterizes how models inherently organize their reasoning. What is missing is an understanding of whether there exists a natural architectural property that governs when predictions stabilize, independent of learned routing mechanisms or post-hoc modifications.

We investigate this question in the context of multiple-choice question answering (MCQA) (Hendrycks et al., 2021), a controlled setting that isolates discrete reasoning from open-ended generation. We discover that a fixed *Hard Decision Layer* (HDL) emerges in transformer models, beyond which the relative ranking of an-

swer options remains stable. The HDL represents a natural, static property of how models organize their reasoning, consistent across diverse datasets and model architectures.

We make the following key contributions: (i) we identify and characterize the *Hard Decision Layer (HDL)*, a natural architectural property in transformer models where answer option rankings stabilize during multiple-choice reasoning, emerging without learned routing policies. (ii) Through a comprehensive experimental study, we demonstrate that HDL is invariant to model fine-tuning (via LoRA) and robust to the number of options, confirming it reflects an intrinsic architectural property rather than a training artifact. (iii) Through systematic evaluation across four language models and four datasets, we also show that HDL prominence depends critically on option label representation, with alphabetic and arabic numeral labels producing sharper decision boundaries than roman numerals.

2 Related Work

Understanding how LLMs process and generate predictions is fundamental to both mechanistic interpretability and computational efficiency. Our work bridges these two areas by examining layer-wise decision-making in transformers, specifically through the lens of multiple-choice question answering (MCQA).

2.1 Mechanistic Interpretability

Mechanistic interpretability seeks to understand the internal computations underlying transformer predictions (Bereska and Gavves, 2024). Information flows through the *residual stream* (Elhage et al., 2021), iteratively refined by attention and feed-forward networks (FFN) across layers. Projection-based techniques like the *logit lens* (nostalgebraist, 2020) and *tuned lens* (Belrose et al., 2023) enable "peeking" at intermediate predictions. Complementary approaches such as activation patching and causal tracing (Meng et al., 2022) identify causal components and factual associations (Hewitt and Manning, 2019; Haviv et al., 2023; Dar et al., 2023).

In transformers, feed-forward layers function as key-value memories (Geva et al., 2022, 2021), promoting concepts to affect outputs. Early layers perform vocabulary filtering while later layers refine candidates (Gupta et al., 2025). Notably, models allocate computational depth dynamically: tokens at the start of factual sequences use more layers for decision-making, while those naturally following use fewer layers, suggesting that models self-organize reasoning based on task context.

2.2 Token Saturation Phenomenon

Model predictions stabilize mid-network, motivating early exit research. (Din et al., 2024) use linear transformations to shortcut intermediate computation, achieving 13.8% layer savings on GPT-2, though this requires dynamic decisions at each layer. More structured approaches show that token ranking saturation follows a predictable order across architectures (Lioubashevski et al., 2024), suggesting principled decision progression and natural boundaries. (Fan et al., 2024) further demonstrate that not all layers are equally essential, achieving layer pruning ratios up to 17.8% in large models like Llama2 and OPT.

2.3 Multiple-Choice Question Answering and Text Classification using LLMs

MCQA benchmarks have become standard for evaluating language models (Hendrycks et al., 2021), offering a balanced evaluation framework that avoids open-ended generation complexities while probing model capabilities. The use of large language models (LLMs) as classifiers through in-context learning (ICL) was established by Brown et al. (2020), who demonstrated that GPT-3 (175B parameters) could perform sentiment analysis, natural language inference, and topic classification by conditioning on a few labeled demonstrations in the prompt and reading the next predicted token as the class label, without any gradient updates. This one-token prediction paradigm, however, was shown by Zhao et al. (2021) to suffer from systematic biases: the model's probability distribution over candidate label tokens is skewed by majority-label bias (favoring whichever class appears most in the demonstrations), recency bias (favoring the label of the last demonstration), and common-token bias (favoring label words that are more frequent in pretraining). Their proposed fix, *contextual calibration*, estimates these biases by feeding a content-free input (e.g., "N/A") and learning an affine transformation on the label logits, recovering up to 30 percentage points of accuracy.

Our central research question is: How do modern LLMs make their answer choice in MCQA tasks? This study also extends to classification tasks using

LLMs. We address this question by identifying a fixed *Hard Decision Layer* (HDL)—the layer beyond which answer option rankings remain stable. Unlike prior work that relies on dynamic early exit policies or post-hoc layer pruning, the HDL emerges as a natural, static property of transformer inference without requiring learned routing mechanisms. We demonstrate that this phenomenon generalizes consistently across model architectures and evaluation datasets. These findings reveal that transformers inherently organize their inference according to architectural and task characteristics, offering mechanistic insights into how depth is utilized during reasoning.

3 Methodology

3.1 Preliminary

3.1.1 Logit Lens

Logit Lens is a technique where intermediate layer representations from a transformer model are projected into the vocabulary space, allowing us to observe what output the model "predicts" at each layer. This technique enables us to track how the model's predictions evolve across layers and identify key decision points in the reasoning process.

Let $\mathcal{M}$ denote a transformer-based large language model with L layers, hidden dimension d, and vocabulary $\mathcal{V}$ of size $|\mathcal{V}| = V$. Given an input sequence of T tokens $\boldsymbol{x} = (x_1, x_2, \ldots, x_T)$ with $x_t \in \mathcal{V}$, the embedding layer produces an initial representation $\boldsymbol{h}^{(0)} \in \mathbb{R}^{T \times d}$. Each transformer block $\ell \in \{1, \ldots, L\}$ updates the residual stream as

$$\boldsymbol{h}^{(\ell)} = f_\ell\big(\boldsymbol{h}^{(\ell-1)}\big), \tag{1}$$

where f_ℓ encapsulates multi-head self-attention followed by a position-wise feedforward network, with residual connections and layer normalization applied according to the model's specific architecture. We capture $\boldsymbol{h}^{(\ell)}$ for every ℓ, storing the output tensor during a single forward pass with gradient computation disabled.

To interpret the intermediate representations, we project each layer's residual stream into vocabulary space using the model's unembedding head. For the token at position T, we first extract the hidden-state vector $\boldsymbol{h}_t^{(\ell)} \in \mathbb{R}^d$ from layer ℓ. Because the unembedding matrix $\boldsymbol{W}_U \in \mathbb{R}^{V \times d}$ is trained to operate on the output of the final layer normalization, we apply the model's final-layer normalization function $\mathrm{LN}_{\text{final}}(\cdot)$ before projection:

$$\hat{\boldsymbol{h}}_t^{(\ell)} = \mathrm{LN}_{\text{final}}\big(\boldsymbol{h}_t^{(\ell)}\big). \tag{2}$$

The logit vector over the vocabulary at layer ℓ and position t is then

$$\boldsymbol{z}_t^{(\ell)} = \boldsymbol{W}_U\, \hat{\boldsymbol{h}}_t^{(\ell)} + \boldsymbol{b}_U, \tag{3}$$

where $\boldsymbol{b}_U \in \mathbb{R}^V$ is the bias term if present. For each layer, we record the logits for the answer option tokens $S = \{A, B, C, D, ...\}$, yielding the pairs $\big\{\big(v,\, z_{t,v}^{(\ell)}\big) : v \in \mathcal{S}\big\}$.

3.1.2 Logit Lens Extraction and Analysis

To investigate how models arrive at answers across layers, we apply Logit Lens to extract layer-wise logits, and thereby, token rankings for the answer options in response to the prompt[1]. For each layer, we record the logits for the answer option tokens and rank them to determine the model's preference at that depth. These layer-wise rankings allow us to compute accuracy at each layer and track how the model's answer evolves throughout the network, providing mechanistic insight into the internal reasoning process.

We now formalize the ranking and accuracy computations used in our analysis. For a question s with four answer options represented by vocabulary tokens $v_1, v_2, v_3, v_4 \in \mathcal{V}$, we extract logits at each layer ℓ using the Logit Lens procedure.

At layer ℓ and position T (the position of the answer token in the sequence), let $\boldsymbol{z}_T^{(\ell)}$ denote the full logit vector over vocabulary, as defined in Equation (3). We extract the logits corresponding to the four answer option tokens as

$$\mathbf{z}_\ell = \big(z_{T,v_1}^{(\ell)}, z_{T,v_2}^{(\ell)}, z_{T,v_3}^{(\ell)}, z_{T,v_4}^{(\ell)}\big), \tag{4}$$

where $z_{T,v_i}^{(\ell)}$ is the v_i-th component of $\boldsymbol{z}_T^{(\ell)}$. We compute the ranking by sorting these logits in descending order:

$$\mathrm{rank}_\ell(v_i) = \mathrm{argsort}_{\text{descending}}(\mathbf{z}_\ell)(v_i), \tag{5}$$

where $\mathrm{rank}_\ell(v_i) \in \{1, 2, 3, 4\}$ represents the position of option token v_i in the sorted logits at layer ℓ, with rank 1 indicating the highest logit.

To aggregate layer-wise rankings across the dataset, we group tokens by their rank at the final output layer. Let $\mathcal{D}$ denote the set of all

[1]We reproduce a standard multiple-choice prompt template in Figure 3 in Appendix A, demonstrating the structured format used in our experiments.

questions, and for each question $s \in \mathcal{D}$, let output_rank(s, v_i) denote the output option rank (position in the final-layer ranking) of token v_i. Let $I_k(s, v_i) := \mathbb{1}[\text{output_rank}(s, v_i) = k]$, where $\mathbb{1}[\cdot]$ is the indicator function, be the indicator that token v_i achieves rank k in question s. The average rank of tokens with output option rank k at layer ℓ is computed as:

$$\overline{\text{rank}}_\ell(k) = \frac{1}{|\mathcal{D}|} \sum_{s \in \mathcal{D}} \sum_{i=1}^{4} \text{rank}_\ell(v_i) \cdot I_k(s, v_i), \quad (6)$$

with standard deviation $\sigma_\ell(k)$ computed across questions to capture variability.

Finally, layer-wise accuracy measures the fraction of questions for which the correct answer token achieves the highest ranking at layer ℓ. Let $v_{\text{correct}}(s)$ denote the correct answer token for question s. The accuracy at layer ℓ is:

$$\text{Acc}_\ell = \frac{1}{|\mathcal{D}|} \sum_{s \in \mathcal{D}} \mathbb{1}[\text{rank}_\ell(v_{\text{correct}}(s)) = 1], \quad (7)$$

where $\mathbb{1}[\cdot]$ is the indicator function.

3.1.3 Hard-Decision Layer

We identify a unique layer - called the Hard-Decision Layer (HDL) - in the modern LLMs tested, which causes the steepest drop in the average ranking of the answer option that eventually ends up as the predicted answer by the model. Let output_rank$(s, v_i) = 1$ identify the final predicted answer option for question s. HDL is formally defined as the layer where the largest decrease in average rank occurs:

$$\ell_{\text{HDL}} = \arg\max_{\ell \in \{1,\ldots,L\}} \left[\overline{\text{rank}}_\ell(1) - \overline{\text{rank}}_{\ell-1}(1)\right], \quad (8)$$

where $\overline{\text{rank}}_\ell(1)$ is the average rank of the predicted answer token (output rank 1) at layer ℓ, as defined in Equation 6. This layer represents the critical point in the network where the model commits to its final answer, with the steepest improvement in answer ranking indicating the strongest refinement in the decision process.

3.2 Experimental Setup

3.2.1 Models

We evaluate four instruction-tuned language models: Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), IBM Granite-3.3-2B-Instruct[2], and Qwen3-4B-Instruct-2507 (Yang et al., 2025). These models range from 2B to 8B parameters and represent diverse architectures and training methodologies. More information about the models used is provided in Appendix A (Table 5).

3.2.2 Datasets

We evaluate model behavior across four multiple-choice question-answering datasets: CommonsenseQA (Talmor et al., 2019), a benchmark for evaluating commonsense reasoning; QASC (Khot et al., 2020), which requires compositional reasoning and multi-step inference; MMLU-Pro (Wang et al., 2024), an enhanced MMLU benchmark covering STEM, social sciences, and humanities; and SuperGPQA (Du et al., 2025), a graduate-level benchmark covering advanced concepts in physics, chemistry, and biology. For each model and dataset, we sample 100 random questions and normalize them to contain exactly four options per question. When a question contains more than four options, we randomly select four while ensuring the correct answer remains in the pool. A detailed summary of the datasets is provided in Appendix A (Table 6).

3.2.3 Fine-tuning with Low-Rank Adaptation

To understand whether HDL persists after fine-tuning, we fine-tune two models (Qwen and Llama) using low-rank adaptation (LoRA) (Hu et al., 2022) for 1 epoch on a separate training set consisting of 8000 to 10000 samples for each evaluated dataset. Given base parameters $\boldsymbol{W}_0$ and low-rank matrices $\boldsymbol{A} \in \mathbb{R}^{d \times r}$, $\boldsymbol{B} \in \mathbb{R}^{r \times d}$ with rank $r \ll d$, a model with the LoRA adapter behaves identically to the model with $\boldsymbol{W} = \boldsymbol{W}_0 + \boldsymbol{AB}$ for each adapted weight matrix. Details regarding the hyperparameters used for model training are provided in Appendix B.

3.2.4 Task Variants

To examine whether other aspects of the multiple-choice format affect the HDL, we conduct experiments varying both the option labeling scheme and the number of available choices. We test three labeling formats on the QASC dataset for both Qwen and Llama models: alphabetic labels (A/B/C/D), arabic numerals (1/2/3/4), and roman numerals (i/ii/iii/iv). It was ensured that, for the models tested, the new option labels are still tokenized

[2] https://huggingface.co/ibm-granite/granite-3.3-2b-instruct

to 1 token. We additionally vary the number of options from three to five choices (using a subset or extension of the original options) to understand how the HDL manifests across different problem complexities. For each configuration, we extract layer-wise logits and compute both option token rankings and layer-wise accuracy. This allows us to isolate the effects of notational convention and problem structure from the underlying model reasoning and assess whether the HDL is robust across different task variants.

3.2.5 Open-ended Generation

To investigate whether HDL as a detectable phenomenon extends beyond multiple-choice reasoning to general sequence generation, we conduct preliminary experiments on open-ended question answering using the GSM8K (Cobbe et al., 2021) dataset of mathematical word problems. Rather than tracking option token rankings, we extract residual states during generation of the n-th output token at various generation steps ($n = 10, 50, 100, 200$), allowing us to observe layer-wise dynamics across different points in the generation process. For each step, we capture the top-10 most likely output tokens at each layer by applying Logit Lens to the intermediate representations. This analysis allows us to assess whether intermediate layers exhibit stabilization patterns similar to those observed in multiple-choice settings and to examine how the rank of the generated token evolves across layers in a free-form generation context.

4 Results

Finding 1: HDL exists in all models and is invariant across MCQA datasets.

As shown in Table 1, all models have a fixed HDL for the MCQA datasets tested. Figure 4 and Figure 5 in Appendix C show the generalization of this behaviour across datasets and models respectively. Some models position their HDL early in the network (e.g., Llama at layer 18 of 32 total layers, or 56.25% depth), while others place it much later (e.g., Granite at layer 32 of 40 total layers, or 80% depth). Table 2 summarizes the model parameter counts and the fraction of layers up to the HDL, representing where most computation occurs. A trend is observed where parameter count appears inversely related to the fraction of layers until the HDL. This suggests that higher-capacity models accomplish answer selection in earlier layers.

Table 1 also demonstrates a sharp jump in the accuracy of intermediate layer outputs at the HDL (also illustrated in Figures 7 and 9 in Appendix D). For some models, such as Qwen, the accuracy at the HDL nearly matches that of the final layer. The accuracy spike at HDL is much larger than the accuracy improvement from HDL till the final output layer. Remarkably, the accuracy of layers before the HDL hovers around 25%, which equals random chance for four-option multiple-choice questions. This suggests the model essentially selects its answer, for most questions, at the HDL.

Citing a specific example of one dataset and model, Figure 1 shows the aggregated layer-wise rankings of the answer option tokens by the Qwen model across the entire QASC dataset. In the plot, Options 1, 2, 3, and 4 represent the options ranked 1st, 2nd, 3rd, and 4th respectively in the model's final output layer. For most questions, the model initially exhibits frequent fluctuations in the ordering of answer tokens. However, at the HDL (layer 25 for the Qwen model), the model produces a decisive ranking that remains largely stable through the remaining layers. On average, option tokens maintain their ranking from the HDL onward, suggesting that the model has completed most of its reasoning by this point. Notably, the top answer token shows greater stability than other option choices. As shown in Figure 4, the HDL position for a given model remains consistent across different datasets. This suggests the HDL is unlikely to be a memory storage layer, in which case we would expect its position to vary with different datasets. Rather, the HDL likely plays a role specific to multiple-choice question answering. The results also show a sharp drop in the mean and standard deviation of Option 1's rank at the HDL which shows the model committing to a specific answer.

Finding 2: HDL is invariant both to finetuning and to the number of choice options. The Qwen and Llama models were finetuned on a separate training set for each of the datasets and evaluated again. The HDL position for these models remains stable even after finetuning. While the models demonstrate clear accuracy improvements (as expected) following finetuning, the HDL itself does not shift across base and finetuned variants, as confirmed quantitatively in Table 3, and visualized in Figures 6 and 8 (Appendix D).

Similarly, when examining whether the number of options in multiple-choice questions affects the HDL position by comparing three-option and five-

Model	Dataset	HDL Layer (Predicted)	Accuracy (Pre-HDL)	Accuracy (Post-HDL)	Accuracy (Final)	Option-1 Rank (Pre-HDL)	Option-1 Rank (Post-HDL)
Qwen	QASC	25	0.31	0.70 (+0.39)	0.72 (+0.02)	2.31 ± 1.09	1.04 ± 0.20
	MMLU-Pro	25	0.23	0.58 (+0.35)	0.61 (+0.03)	2.52 ± 1.17	1.09 ± 0.35
	CommonsenseQA	25	0.26	0.87 (+0.61)	0.86 (-0.01)	2.42 ± 1.07	1.01 ± 0.10
	SuperGPQA	25	0.24	0.40 (+0.16)	0.41 (+0.01)	2.22 ± 1.11	1.05 ± 0.26
Llama	QASC	18	0.28	0.59 (+0.31)	0.74 (+0.15)	2.35 ± 1.01	1.42 ± 0.62
	MMLU-Pro	18	0.25	0.45 (+0.20)	0.59 (+0.14)	2.72 ± 1.11	1.97 ± 1.05
	CommonsenseQA	18	0.24	0.63 (+0.39)	0.80 (+0.17)	2.54 ± 1.08	1.52 ± 0.81
	SuperGPQA	23	0.29	0.41 (+0.12)	0.32 (-0.09)	2.11 ± 0.96	1.58 ± 0.89
Granite	QASC	32	0.27	0.51 (+0.24)	0.67 (+0.16)	2.41 ± 1.21	1.56 ± 0.88
	MMLU-Pro	32	0.32	0.45 (+0.13)	0.47 (+0.02)	2.54 ± 1.12	1.90 ± 1.01
	CommonsenseQA	32	0.30	0.56 (+0.26)	0.72 (+0.16)	2.45 ± 1.20	1.73 ± 1.01
	SuperGPQA	33	0.27	0.36 (+0.09)	0.41 (+0.05)	2.12 ± 1.18	1.74 ± 1.05
Mistral	QASC	20	0.38	0.56 (+0.18)	0.62 (+0.06)	1.86 ± 0.96	1.32 ± 0.72
	MMLU-Pro	20	0.30	0.46 (+0.16)	0.49 (+0.03)	2.09 ± 1.03	1.35 ± 0.77
	CommonsenseQA	20	0.45	0.68 (+0.23)	0.76 (+0.08)	1.97 ± 1.08	1.36 ± 0.66
	SuperGPQA	20	0.27	0.33 (+0.06)	0.37 (+0.04)	2.08 ± 0.89	1.50 ± 0.69

Table 1: Descriptive HDL summary across models and datasets. Accuracy (Pre-HDL), Accuracy (Post-HDL), and Accuracy (Final) are mean accuracies of the logit-lens projection at layers before the predicted HDL, at and after the HDL, and at the final layer, respectively. The HDL Layer (Predicted) is predicted as the layer that causes the largest drop in Option 1's average rank. Parenthetical values in the accuracy columns indicate absolute increases: Accuracy (Post-HDL) shows the increase over Accuracy (Pre-HDL), and Accuracy (Final) shows the increase over Accuracy (Post-HDL). A small final-post gap indicates that post-HDL behavior already matches the model's final output. Option-1 Rank (Pre-HDL) and Option-1 Rank (Post-HDL) show the mean and standard deviation of the rank of Option 1 at the layer before HDL and at the HDL layer, respectively. Lower rank values indicate greater concentration on the correct answer, demonstrating the rank collapse that shows the model committing to an answer.

Table 2: Depth of Hard Decision Layer (as percentage of layers) across models

Model	Parameters	Depth of HDL (as % of layers)
Llama	8B	56.25% (18/32)
Mistral	7B	62.50% (20/32)
Qwen	4B	69.44% (25/36)
Granite	2B	80.00% (32/40)

option variants on the QASC dataset for both Qwen and Llama models, we found the HDL position to remain invariant across different option counts (Figure 10 in Appendix E). While the pre-HDL accuracy differs according to random chance (33% for three options, 20% for five options), the HDL location and the accuracy patterns (Figure 11 in Appendix E) do not significantly differ across these conditions - quantified in Table 4.

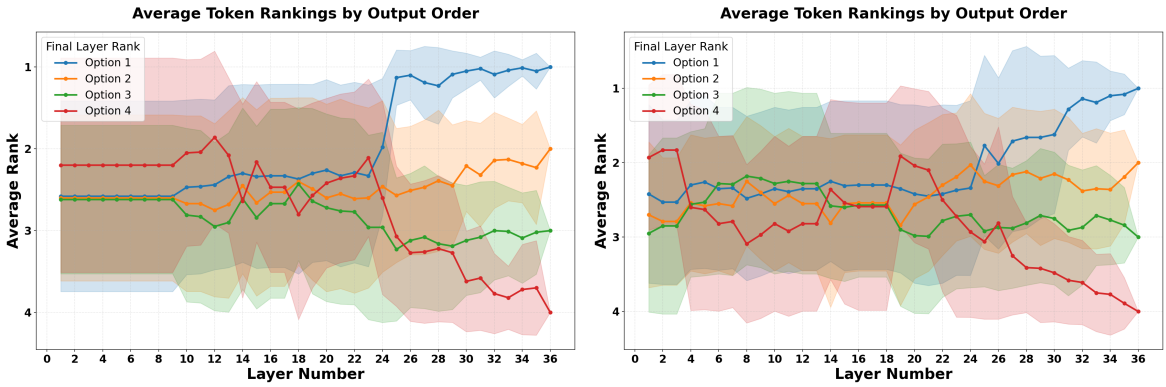
Finding 3: HDL prominence is affected by option labels used. Previous experiments used alphabets (A/B/C/D) as option labels in the multiple-choice questions. To determine whether the choice of option labels affects the location or prominence of the HDL, we tested alternatives: arabic numerals (1/2/3/4) and roman numerals (i/ii/iii/iv). As shown in both Figure 2 (also in Table 7 and Figure 12 in Appendix D), the HDL appears at the same position across all labeling schemes. However, the pattern differs in clarity: the HDL is well-pronounced for both arabic numerals and alphabets, whereas for roman numerals it is less distinct. The top-1 option remains stable post-HDL across all schemes, but other option probabilities show greater volatility with roman numerals compared to the other formats. The layer-wise accuracy plots in Figure 13 - also in Appendix E - reveal more pronounced differences. Arabic numerals exhibit the steepest accuracy jump at the HDL, with alphabets showing nearly identical behavior. In contrast, roman numerals display a more gradual accuracy increase beginning at the HDL. This might be indicative of MCQs in training data having more of alphabets and arabic numerals for option labels.

4.1 Open-ended generation

We also briefly explore open-ended question answering by evaluating models on GSM8K math word problems. Residuals during generation of nth token were captured ($n = 10, 50, 100, 200$). Figure 14 in Appendix F displays average token rankings for the top 10 output tokens from the Qwen model; tokens not in the top-10 at intermediate lay-

Model	Dataset	Variant	HDL Layer (Predicted)	Accuracy (Pre-HDL)	Accuracy (Post-HDL)	Accuracy (Final)
Llama	CommonsenseQA	Base	18	0.24	0.63 (+0.39)	0.80 (+0.17)
		Finetuned	18	0.25	0.60 (+0.35)	0.86 (+0.26)
Llama	MMLU-Pro	Base	18	0.25	0.45 (+0.20)	0.59 (+0.14)
		Finetuned	18	0.21	0.40 (+0.19)	0.65 (+0.25)
Llama	QASC	Base	18	0.28	0.59 (+0.31)	0.74 (+0.15)
		Finetuned	18	0.29	0.61 (+0.32)	0.77 (+0.16)
Llama	SuperGPQA	Base	23	0.29	0.41 (+0.12)	0.32 (-0.09)
		Finetuned	18	0.23	0.31 (+0.08)	0.40 (+0.09)
Qwen	CommonsenseQA	Base	25	0.26	0.87 (+0.61)	0.86 (-0.01)
		Finetuned	25	0.27	0.86 (+0.59)	0.86 (+0.00)
Qwen	MMLU-Pro	Base	25	0.23	0.58 (+0.35)	0.61 (+0.03)
		Finetuned	25	0.22	0.65 (+0.43)	0.69 (+0.04)
Qwen	QASC	Base	25	0.31	0.70 (+0.39)	0.72 (+0.02)
		Finetuned	25	0.30	0.79 (+0.49)	0.80 (+0.01)
Qwen	SuperGPQA	Base	25	0.24	0.40 (+0.16)	0.41 (+0.01)
		Finetuned	25	0.23	0.45 (+0.22)	0.42 (-0.03)

Table 3: Descriptive HDL summary for base vs fine-tuned models. Each (model, dataset) pair appears on two rows: base and finetuned variants. HDL Layer (Predicted), Accuracy (Pre-HDL), Accuracy (Post-HDL), and Accuracy (Final) are mean logit-lens accuracies at layers before the predicted HDL, at and after the HDL, and at the final layer, respectively. Parenthetical values in the accuracy columns indicate absolute increases: Accuracy (Post-HDL) shows the increase over Accuracy (Pre-HDL), and Accuracy (Final) shows the increase over Accuracy (Post-HDL). Comparing base and finetuned variants within a (model, dataset) pair reveals whether fine-tuning shifted the HDL position or only its accuracy levels.



(a) Qwen (Arabic)

(b) Qwen (Roman)

Figure 2: Impact of option labeling schemes on token rankings across layers. Each subfigure shows aggregate token ranking plots for the Qwen model across alternate labeling schemes: Arabic numerals (1/2/3/4), and Roman numerals (i/ii/iii/iv), all evaluated on the QASC dataset. The plots reveal whether the location and prominence of the Hard Decision Layer (HDL) varies across different labeling schemes. Generally, the HDL manifests at a similar layer position regardless of labeling scheme, though its clarity differs: Alphabets (in Figure 1) and Arabic numerals produce a well-pronounced HDL, while Roman numerals show a less distinct pattern.

ers appear blank. The HDL is less pronounced in these plots than in multiple-choice settings. However, knowing that layer 25 is the HDL for Qwen, we observe that the top-1 token stabilizes around this layer across all values of n, while other tokens show no clear pattern. This suggests the HDL may extend beyond multiple-choice answering to general token generation, though further investigation is needed.

5 Discussion

Our findings reveal interesting patterns in how language models process multiple-choice questions across their layers. At first glance, the observed

Model	Number of Options	HDL Layer (Predicted)	Accuracy (Pre-HDL)	Accuracy (Post-HDL)	Accuracy (Final)
Qwen	3	25	0.32	0.78 (+0.46)	0.77 (−0.01)
Qwen	4	25	0.31	0.70 (+0.39)	0.72 (+0.02)
Qwen	5	25	0.22	0.64 (+0.42)	0.64 (+0.00)
Llama	3	18	0.34	0.62 (+0.28)	0.78 (+0.16)
Llama	4	18	0.28	0.59 (+0.31)	0.74 (+0.15)
Llama	5	18	0.24	0.55 (+0.31)	0.70 (+0.15)

Table 4: Descriptive HDL summary across QASC option-count variants. All variants use alphabetic labels (A/B/C/D) on the same QASC questions; only the Number of Options differs. The default QASC cohort uses 4 options. Each row shows the HDL Layer (Predicted) and accuracies at the layer just before HDL (Accuracy (Pre-HDL)), at the HDL layer (Accuracy (Post-HDL)), and at the final layer (Accuracy (Final)) for one (model, number of options) combination; parenthetical values on the accuracy columns are absolute increases over the previous column. Results are restricted to Qwen and Llama because the 3- and 5-option residual-trace pickles are only available for those models.

behavior might be attributable to models storing dataset-specific information in particular layers, as suggested by prior work on model internals (Meng et al., 2022). However, this explanation appears unlikely for several reasons. The behavior remains consistent across diverse datasets, suggesting it is not tied to specific memorization of training data. Furthermore, the phenomenon occurs consistently across four different independently-trained models, making it improbable that every model independently chose to localize all dataset-specific knowledge for four different datasets in one layer. Additionally, the inclusion of SuperGPQA dataset, released after the Llama and Mistral models were trained, rules out dataset memorization as the primary driver, while still allowing for the remote possibility that relevant knowledge is localized within these layers.

A more compelling interpretation is that this behavior reflects the inherent structure of multiple-choice question answering itself. If this hypothesis holds, it would suggest that models may self-organize their computations in a task-specific manner, with later layers dedicated to selecting among the available options. Consistent with this view, our analysis of open-ended generation shows that while this decision-making pattern is less pronounced, it remains present. This aligns with recent findings in the literature showing that models identify likely candidate tokens in earlier layers and refine token distributions progressively through deeper layers (Gupta et al., 2025).

Our results have significant implications for model compression and efficiency. Since the HDL achieves accuracy comparable to the final layer, layers beyond the HDL can be pruned without sacrificing performance on multiple-choice tasks. Larger models like Llama demonstrate a shallower HDL position, enabling up to 43.75% of layers to be removed while retaining competitive accuracy. This reveals an interesting interplay between model capacity and computational depth: a larger model with substantial layer pruning post-HDL may outperform a smaller, naturally efficient model with a deeper HDL. These results suggest that model size and HDL position are jointly important considerations for inference efficiency, providing a novel approach to trading off computational cost against model capacity. However, the findings also suggest a critical caveat to layer pruning: when prompt formatting differs, the final layers become essential for maintaining accuracy, and aggressive pruning post-HDL could substantially degrade performance.

6 Limitations

This study has several important limitations that should be considered when interpreting the results. First, our evaluation restricts models to direct answer selection without access to chain-of-thought reasoning, which is a more natural and often more effective way for models to approach complex problems. Additionally, our analysis focuses only on the canonical answer labels (A/B/C/D), but intermediate layers may activate alternative surface forms that correspond to the same option (e.g., “b”, “second”, “B)” and other variants), which are not accounted for in the current analysis. Furthermore, while our analysis includes some exploration of open-ended generation, this setting requires substantially more investigation to understand whether the patterns we observe in multiple-choice tasks generalize to free-form generation. Future work

should extend this analysis to chain-of-thought reasoning, more diverse generation tasks, account for answer form variations, and investigate the underlying mechanisms driving the observed layer-wise decision patterns.

References

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

L. Bereska and E. Gavves. 2024. Mechanistic interpretability for ai safety: A review. arXiv preprint arXiv:2404.14082.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*.

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

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2023. Analyzing transformers in embedding space. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 16124–16170.

Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. 2024. Jump to conclusions: Short-cutting transformers with linear transformations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9615–9625.

Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, and 1 others. 2025. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, and 1 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Siqi Fan, Xin Jiang, Xiang Li, Xuying Meng, Peng Han, Shuo Shang, Aixin Sun, Yequan Wang, and Zhongyuan Wang. 2024. Not all layers of llms are necessary during inference. *arXiv preprint arXiv:2403.02181*.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 30–45.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

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*.

Akshat Gupta, Jay Yeung, Gopala Anumanchipalli, and Anna Ivanova. 2025. How do llms use their depth? *arXiv preprint arXiv:2510.18871*.

Adi Haviv, Ido Cohen, Jacob Gidron, Roei Schuster, Yoav Goldberg, and Mor Geva. 2023. Understanding transformer memorization recall through idioms. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 248–264.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

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*, pages 4129–4138.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. *arXiv:1910.11473v2*.

Daria Lioubashevski, Tomer Schlank, Gabriel Stanovsky, and Ariel Goldstein. 2024. Looking beyond the top-1: Transformers determine top tokens in order. *arXiv preprint arXiv:2410.20210*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372.

nostalgebraist. 2020. interpreting gpt: the logit lens. *LessWrong*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

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. *Advances in Neural Information Processing Systems*, 37:95266–95290.

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

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Appendix Contents

A Prompt, Models and Datasets 11

B LoRA Hyperparameter Details 11

C Supplementary Results for Finding 1 11

D Supplementary Results for Finding 2 11

E Supplementary Results for Finding 3 11

F Open-Ended Generation 11

A Prompt, Models and Datasets

Figure 3 shows the prompt template used for the multiple-choice question answering used in experiments. Tables 5 and 6 provide more information about the models and datasets used respectively.

> You are given a question and some options. Output the correct option letter only and nothing else.
> <question>
> For the two linear equations 2 * x + 3 * y = 10 and 4 * x + 4 * y = 12 with variables x and y. Use cramer's rule to solve these two variables.
> </question>
> <options>
> (A) [4, 1]
> (B) [-2, 6]
> (C) [3, 2]
> (D) [-1, 4]
> </options>
> The correct option is: (

Figure 3: Prompt template used in the MCQA tasks.

B LoRA Hyperparameter Details

Training Hyperparameters Hyperparameters for all LoRA model trainings are set as follows:

- Learning rate: 2×10^{-4}
- Maximum prompt length: 1024
- Batch size: 64
- LoRA rank: 32

Implementation We use the Tinker library for implementing LoRA fine-tuning.

Dataset-specific Sample Counts The number of samples used for fine-tuning varies by dataset:

- CommonsenseQA: 9000 samples
- MMLU-Pro: 10000 samples
- QASC: 8000 samples
- SuperGPQA: 10000 samples

C Supplementary Results for Finding 1

Figure 4 shows the average token rankings through the layers for the Qwen model on all the four datasets evaluated. Figure 5 shows the average token rankings through the layers for all four models on the QASC dataset.

D Supplementary Results for Finding 2

Figure 6 and Figure 7 show the average token ranking and accuracy through the layers for the Qwen model in comparison to its finetuned versions on QASC and MMLU-Pro datasets, while Figure 8 and Figure 9 show the corresponding results for the Llama model.

E Supplementary Results for Finding 3

Figures 10 and 11 show the average token rankings and accuracy through the layers for the Qwen and Llama models when evaluated on the QASC dataset with varying number of answer choices. Figures 12 and 13 along with Table 7 show the results of Qwen and Llama models when the option labels were varied.

F Plots for Open-Ended Generation

Figure 14 shows the average token rankings through the layers for the Qwen model on the open-ended generation task evaluated on different token positions $n = 10, 50, 100, 200$

Table 5: List of models evaluated in this study. Release dates are approximate and sourced from Hugging Face and arXiv.

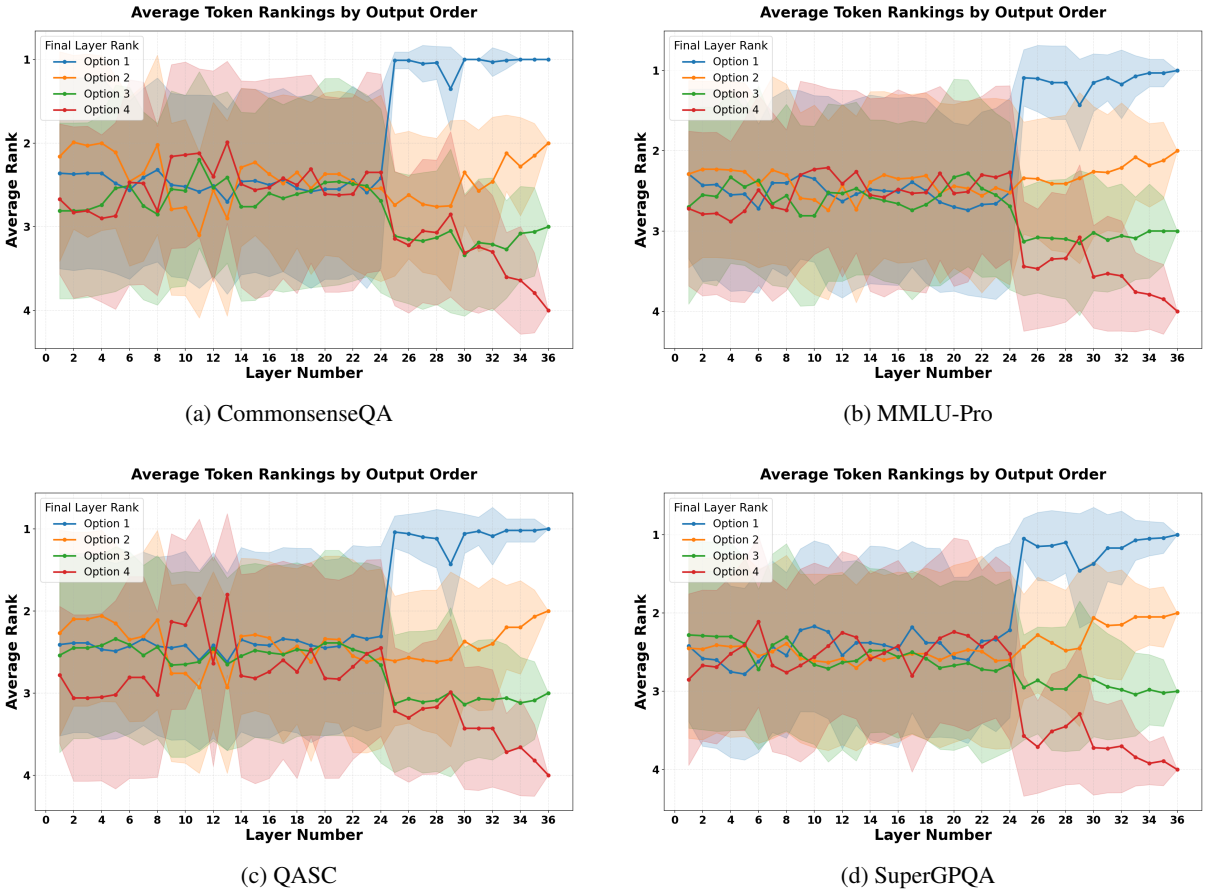
Model	Parameter Count	Release Date	License
mistralai/Mistral-7B-Instruct-v0.3	7B	May 22, 2024	Apache 2.0
meta-llama/Llama-3.1-8B-Instruct	8B	July 23, 2024	Llama 3.1 Community License Agreement
ibm-granite/granite-3.3-2b-instruct	2B	April 16, 2025	Apache 2.0
Qwen/Qwen3-4B-Instruct-2507	4B	August 5, 2025	Apache 2.0

Table 6: Summary of datasets used in this study. Release dates are approximate and sourced from Hugging Face and arXiv.

Dataset	Description	Release Date	License
CommonsenseQA	Benchmark for evaluating commonsense reasoning capabilities through multiple-choice questions.	November 2, 2018	MIT
QASC	Question-answering dataset requiring compositional reasoning and multi-step inference across multiple facts.	October 21, 2019	Creative Commons Attribution 4.0
MMLU-Pro	Enhanced MMLU benchmark covering STEM, social sciences, and humanities with challenging multiple-choice questions.	May 8, 2024	MIT
SuperGPQA	Graduate-level question-answering benchmark covering advanced concepts in physics, chemistry, and biology.	February 20, 2025	Open Data Commons License Attribution

Model	Label Type	HDL Layer (Predicted)	Accuracy (Pre-HDL)	Accuracy (Post-HDL)	Accuracy (Final)
Qwen	alpha	25	0.31	0.70 (+0.39)	0.72 (+0.02)
Qwen	arabic	25	0.31	0.73 (+0.42)	0.75 (+0.02)
Qwen	roman	25	0.27	0.57 (+0.30)	0.77 (+0.20)
Llama	alpha	18	0.28	0.59 (+0.31)	0.74 (+0.15)
Llama	arabic	18	0.43	0.68 (+0.25)	0.71 (+0.03)
Llama	roman	18	0.42	0.56 (+0.14)	0.73 (+0.17)

Table 7: Descriptive HDL summary across QASC option-label variants. All variants present 4 options on the same QASC questions; only the Label Type differs (alpha = A/B/C/D, arabic = 1/2/3/4, roman = i/ii/iii/iv). Each row shows the HDL Layer (Predicted) and accuracies at the layer just before HDL (Accuracy (Pre-HDL)), at the HDL layer (Accuracy (Post-HDL)), and at the final layer (Accuracy (Final)) for one (model, label type) combination; parenthetical values on the accuracy columns are absolute increases over the previous column.



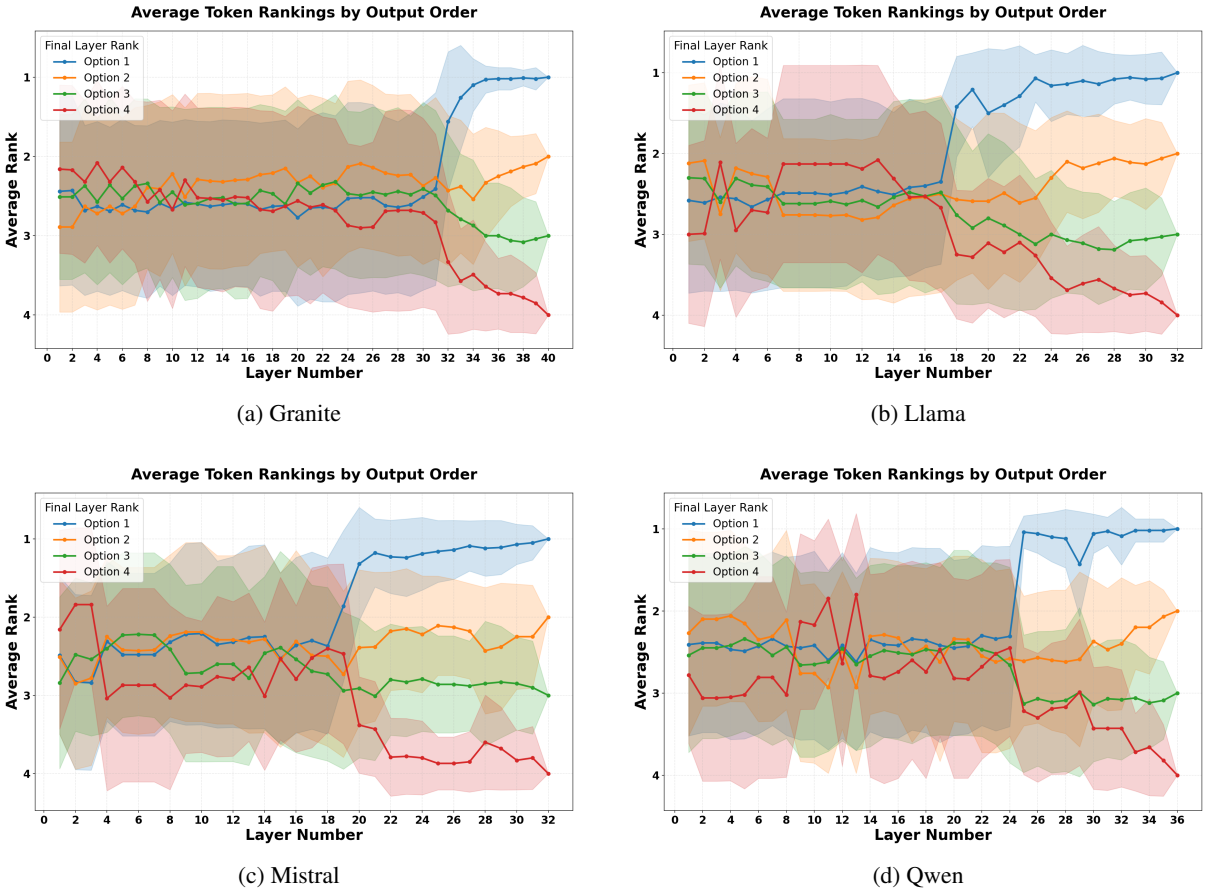
(a) CommonsenseQA

(b) MMLU-Pro

(c) QASC

(d) SuperGPQA

Figure 4: Average token rankings plots of the Qwen model when evaluated on different datasets.



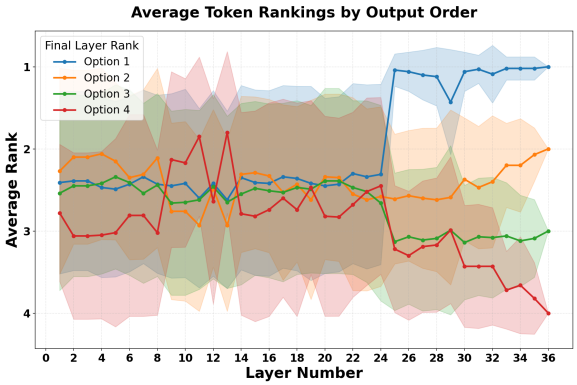
(a) Granite

(b) Llama

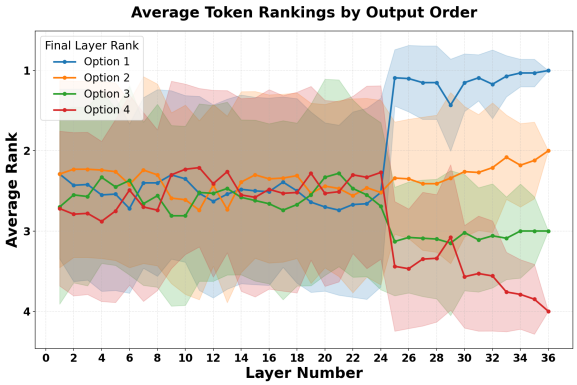
(c) Mistral

(d) Qwen

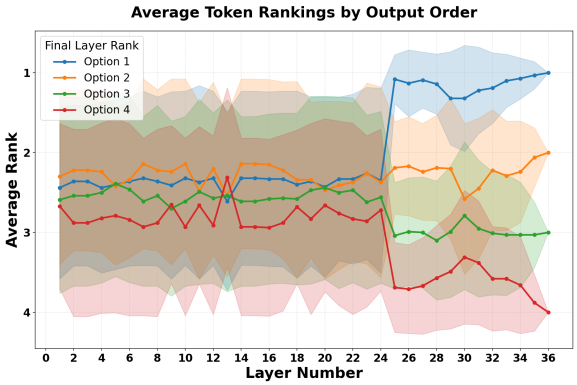
Figure 5: Average token rankings plots across different models when evaluated on the QASC dataset.



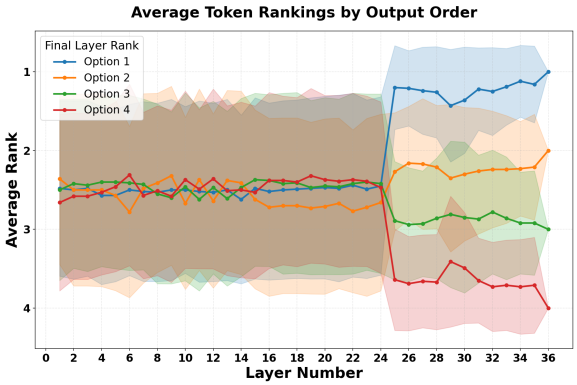
(a) QASC Qwen (Base)



(b) MMLU-Pro Qwen (Base)

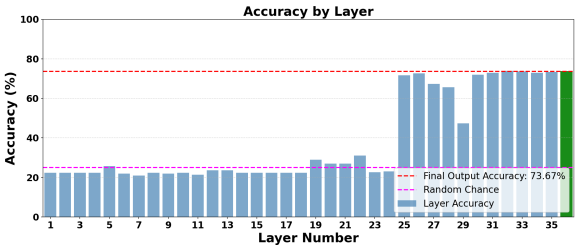


(c) QASC Qwen (Ft)

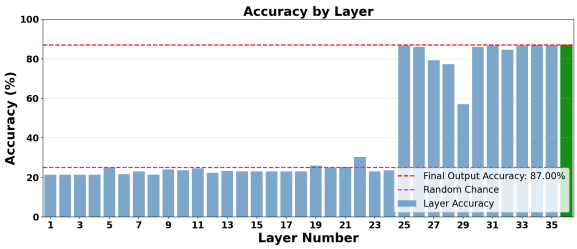


(d) MMLU-Pro Qwen (Ft)

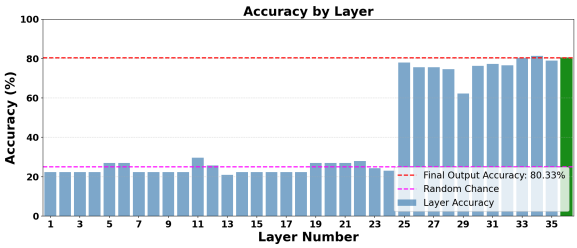
Figure 6: Average token rankings plots comparing base and finetuned (Ft) Qwen models across datasets.



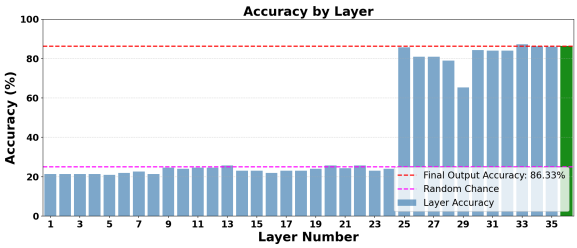
(a) Qwen QASC (Base)



(b) Qwen CommonsenseQA (Base)

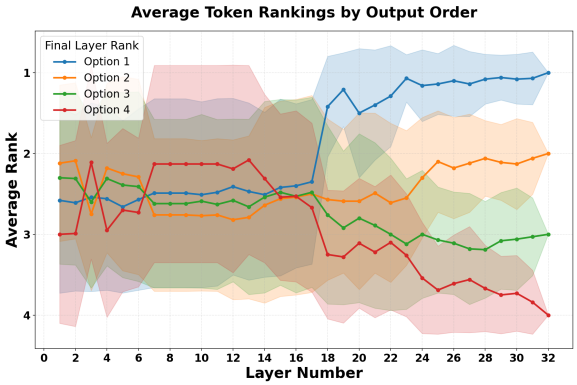


(c) Qwen QASC (Ft)

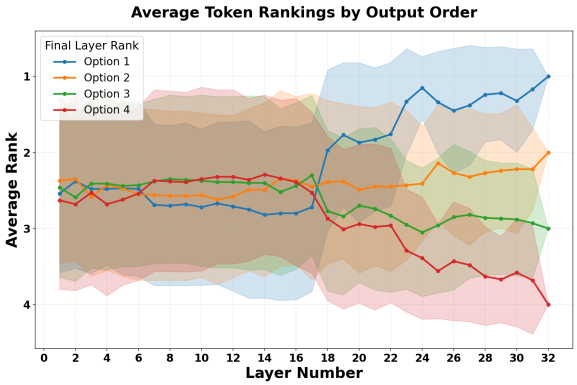


(d) Qwen CommonsenseQA (Ft)

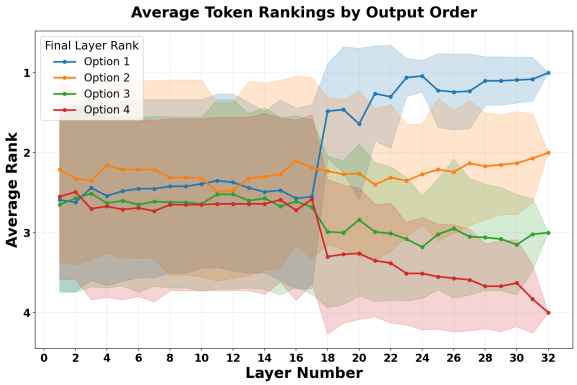
Figure 7: Accuracy of intermediate layer outputs across model layers for base and finetuned (Ft) Qwen models.



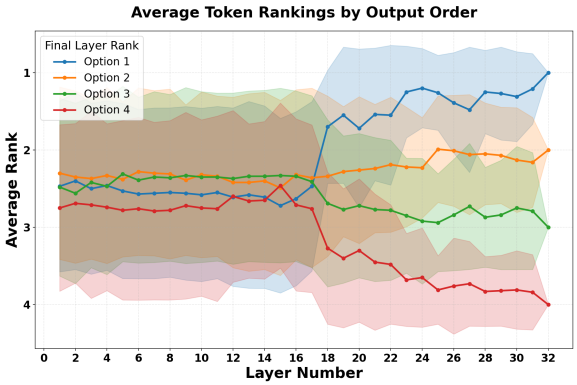
(a) QASC Llama (Base)



(b) MMLU-Pro Llama (Base)

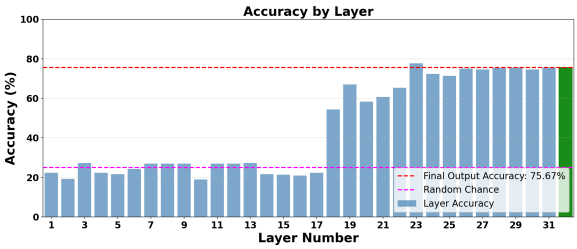


(c) QASC Llama (Ft)

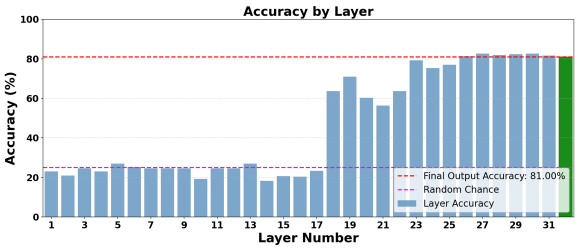


(d) MMLU-Pro Llama (Ft)

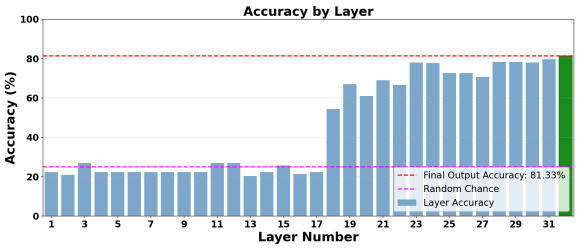
Figure 8: Average token rankings plots comparing base and finetuned (Ft) Llama models across datasets.



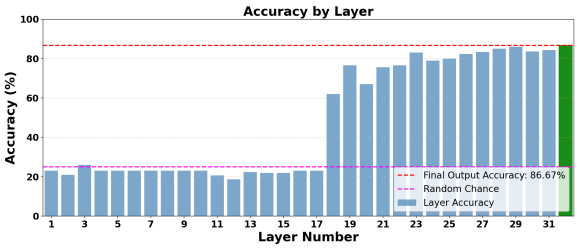
(a) Llama QASC (Base)



(b) Llama CommonsenseQA (Base)

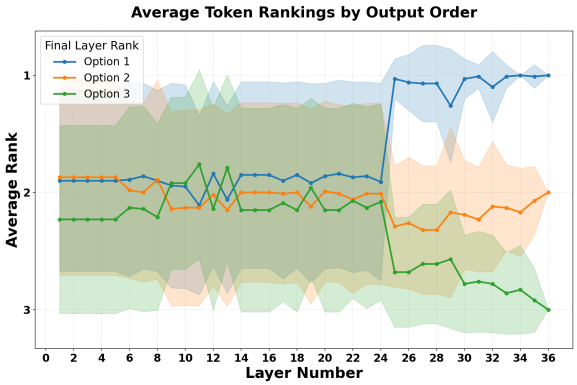


(c) Llama QASC (Ft)

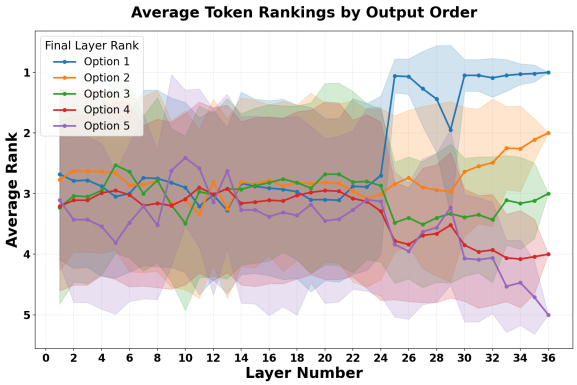


(d) Llama CommonsenseQA (Ft)

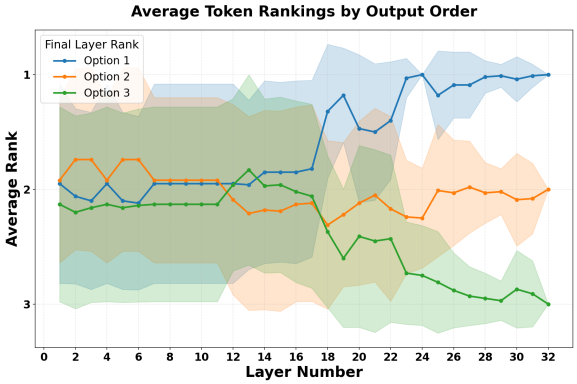
Figure 9: Accuracy of intermediate layer outputs across model layers for base and finetuned (Ft) Llama models.



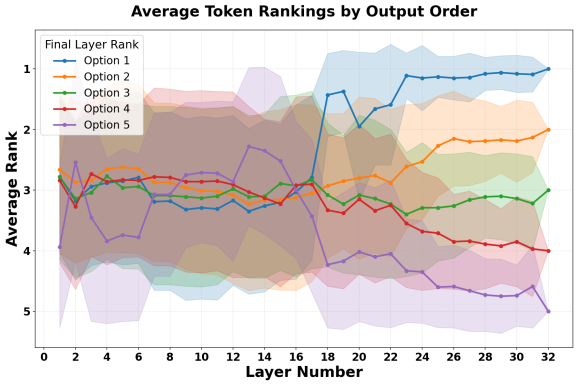
(a) Qwen (3 Options)



(b) Qwen (5 Options)

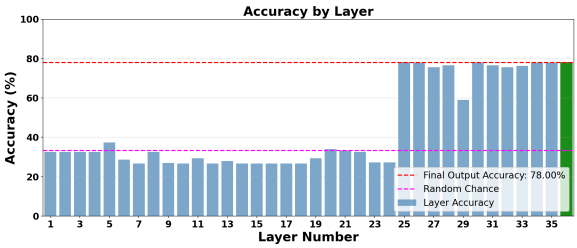


(c) Llama (3 Options)

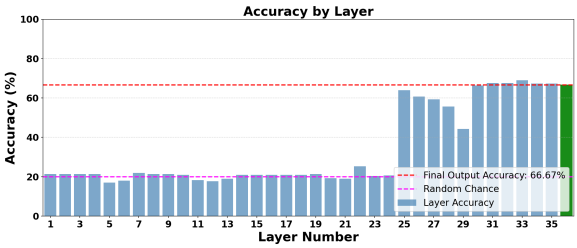


(d) Llama (5 Options)

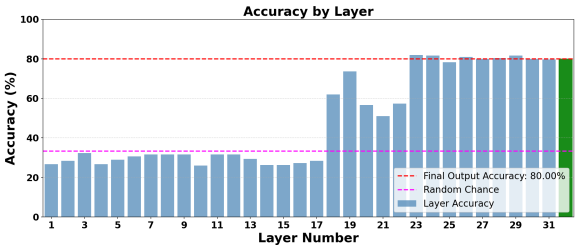
Figure 10: Average token ranking plots for varying numbers in the multiple-choice options for Qwen and Llama models on the QASC dataset.



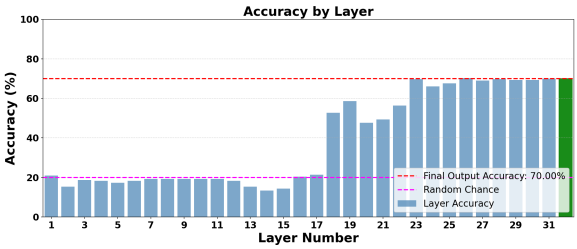
(a) Qwen (3 Options)



(b) Qwen (5 Options)

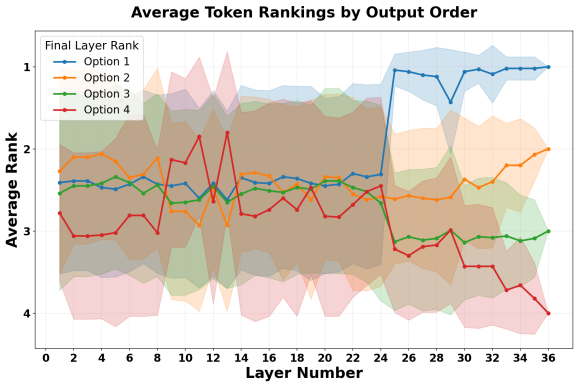


(c) Llama (3 Options)

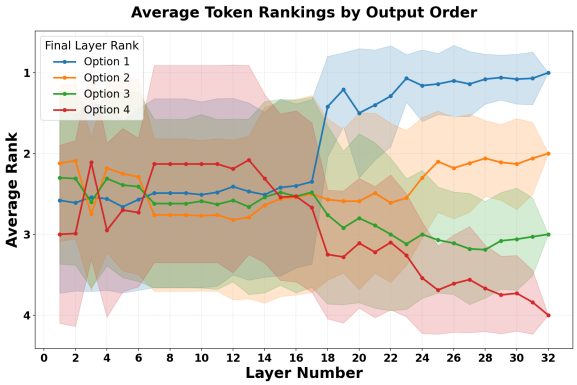


(d) Llama (5 Options)

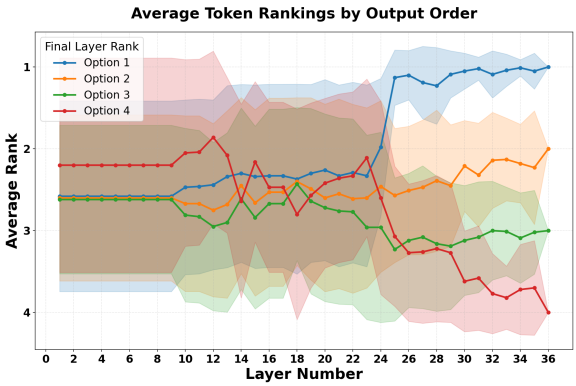
Figure 11: Layer-wise accuracy comparison across different numbers of multiple-choice options.



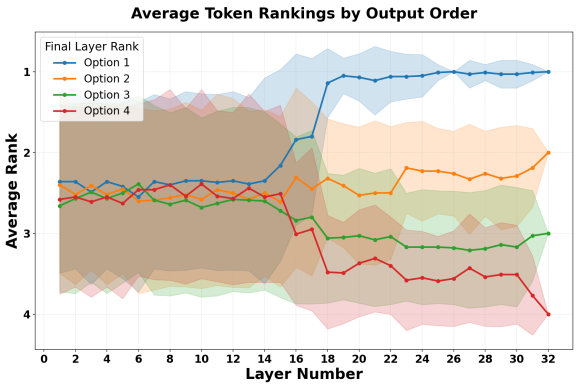
(a) Qwen (Alphabets)



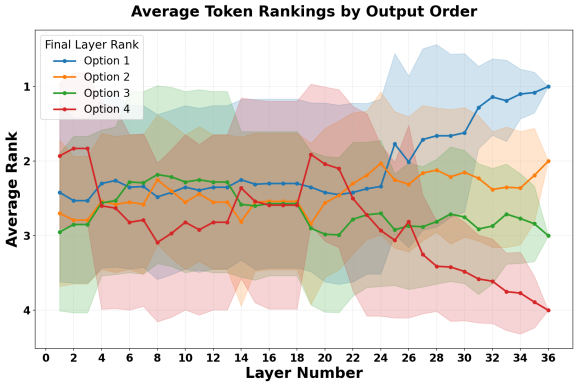
(b) Llama (Alphabets)



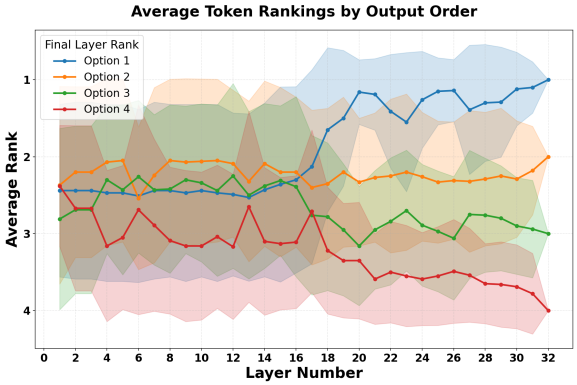
(c) Qwen (Arabic)



(d) Llama (Arabic)

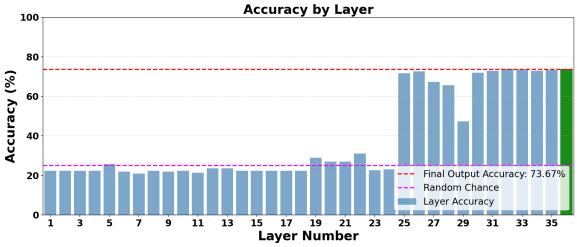


(e) Qwen (Roman)

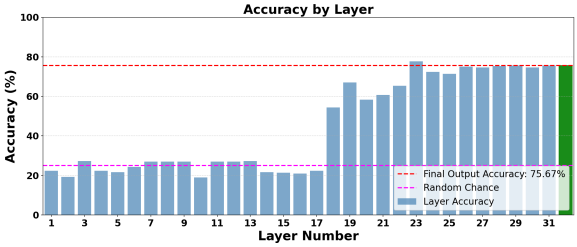


(f) Llama (Roman)

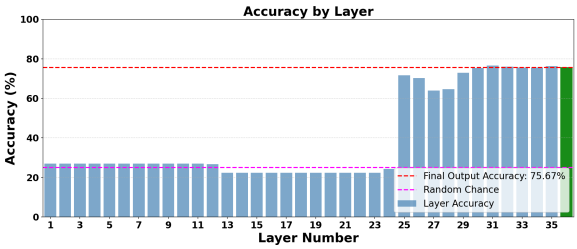
Figure 12: Impact of option labeling schemes on token rankings across layers. Each subfigure shows aggregate token ranking plots for either Qwen or Llama across three labeling schemes: Alphabets (A/B/C/D), Arabic numerals (1/2/3/4), and Roman numerals (i/ii/iii/iv), all evaluated on the QASC dataset. The plots reveal whether the location and prominence of the Hard Decision Layer (HDL) varies across different labeling schemes. Generally, the HDL manifests at a similar layer position regardless of labeling scheme, though its clarity differs: Alphabets and Arabic numerals produce a well-pronounced HDL, while Roman numerals show a less distinct pattern.



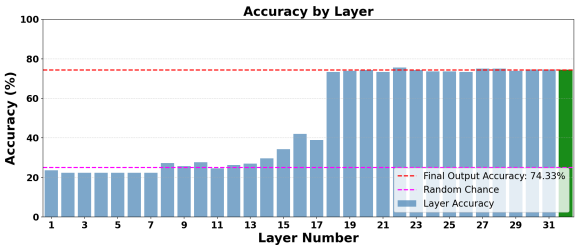
(a) Qwen (Alphabets)



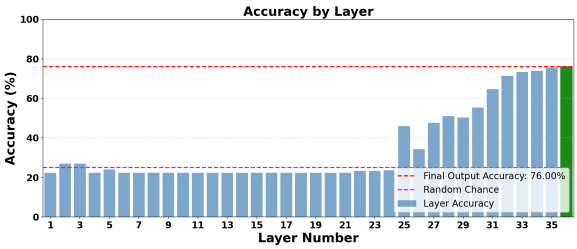
(b) Llama (Alphabets)



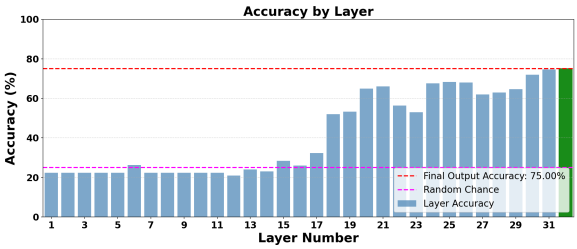
(c) Qwen (Arabic)



(d) Llama (Arabic)

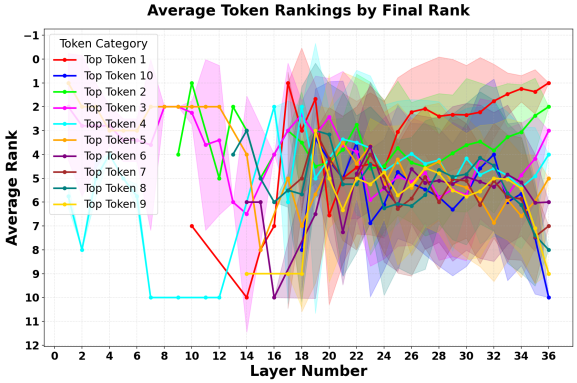


(e) Qwen (Roman)

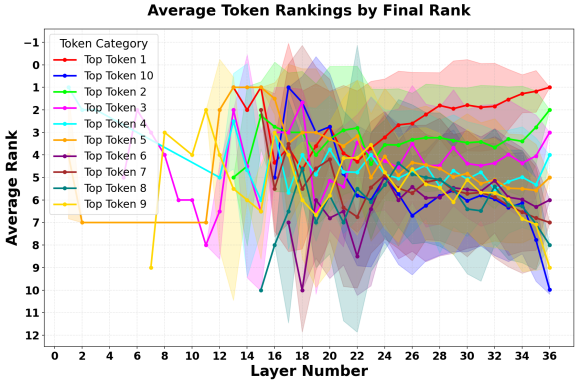


(f) Llama (Roman)

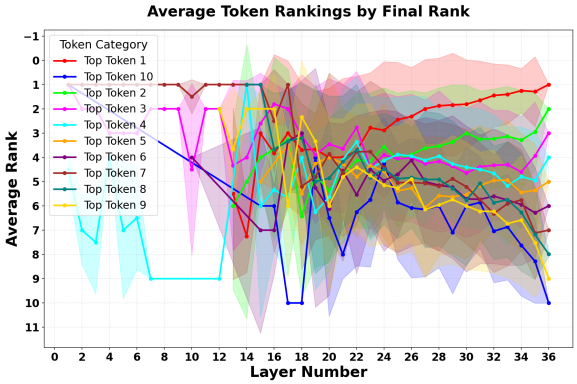
Figure 13: Layer-wise accuracy comparison across different option label types.



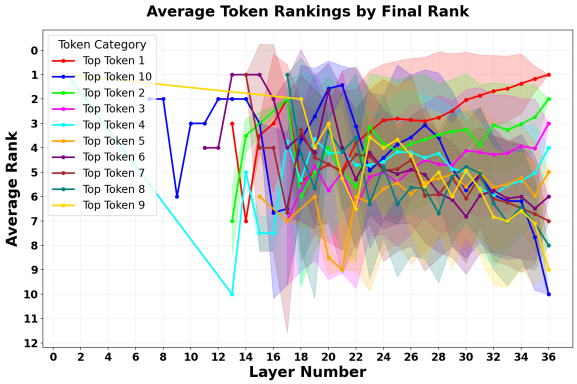
(a) $n = 10$



(b) $n = 50$



(c) $n = 100$



(d) $n = 200$

Figure 14: Aggregate plots of nth token generation in the open-ended GSM8K evaluation for Qwen.