

ALW: Adaptive Layer-Wise contrastive decoding enhancing reasoning ability in Large Language Models

Yuechi Zhou, Chuyue Zhou, Jianxin Zhang, Juntao Li*, Min Zhang

School of Computer Science and Technology, Soochow University

Key Laboratory of Data Intelligence and Advanced Computing, Soochow University

{yczhou, cyzhou, 20204027008}@stu.suda.edu.cn

{ljt, minzhang}@suda.edu.cn

Abstract

Large language models (LLMs) have achieved remarkable performance across various reasoning tasks. However, many LLMs still encounter challenges in reasoning, especially for LLMs with fewer parameters or insufficient pre-training data. Through our experiments, we identify that noise accumulation across layers often leads to unstable token predictions during reasoning. We find that contrasting the probability distributions across layers effectively mitigates this interference. Building on this insight, we propose Adaptive Layer-Wise contrastive decoding (ALW), a novel framework that enhances reasoning ability by dynamically disentangling noise in shallow layers from critical signals in deep layers. Extensive experiments on several reasoning benchmarks demonstrate that ALW consistently improves answer accuracy across multiple LLMs while maintaining inference efficiency. For example, we achieve a 48% improvement on the Gsm8k using the LLaMA-7B model and an absolute accuracy increase of 5.2 points on the BBH evaluation benchmark with the LLaMA-65B model¹.

1 Introduction

Recent years have witnessed significant advancements in reasoning tasks (Qiao et al., 2023; Hong et al., 2024) due to the development and deployment of large language models (LLMs) (Yang et al., 2024; Zhao et al., 2024; Guo et al., 2025). However, not all language models (LMs) possess strong reasoning abilities. They encounter a variety of reasoning issues, such as arithmetic mistakes and semantic errors (Wang et al., 2023).

Existing research enhances reasoning capabilities by linking to external knowledge and retrieving reasoning evidence (Zhu et al., 2023; Wang et al., 2024). Recent works find that contrast decoding of two different models can improve text quality

in open-ended text generation (Li et al., 2023) and have great potential in reasoning tasks (O’Brien and Lewis, 2023). Chuang et al. (2024) further achieve performance improvement on open-ended generation tasks through self-contrast decoding but with limited gains on reasoning tasks. In view of this, we start from recent observations on noise signals within LLMs (Ye et al., 2024), in which LLMs assign disproportionately high scores to non-critical tokens, and directly subtracting the attention distributions of different attention heads can effectively diminish the influence of noisy irrelevant tokens to the generation results. This insight leads us to question whether the noise from over-attended irrelevant tokens accumulates across different layers and results in diverse reasoning errors observed from the final layer, e.g., repetitive generation. Figure 1 preliminarily confirms that subtracting the prediction distribution of the shallow layer from that of the final layer can weaken noisy context and produce the correct answer “No”.

Therefore, we mitigate noise at the layer level and propose adaptive layer-wise contrastive decoding, a framework designed to enhance reasoning capabilities by explicitly disentangling shallow noise from deep reasoning. To achieve this, we develop a quantification strategy to determine the optimal early exit layer for noise suppression. Furthermore, we introduce a lightweight LM that learns which layer of the LLM requires noise suppression in an end-to-end manner and dynamically adjusts the contrastive strategy during the inference stage. Additionally, we extend ALW by developing a tuning-free method based on semantic similarity, reducing resource consumption.

Specifically, we first identify the optimal noise layer that requires the most suppression for different inputs and tasks. Then, a predictor, either LM-based or similarity-based, is employed to capture the mapping between the prompt and the optimal contrastive layer. During inference, the prompt is

*Corresponding author.

¹Code is available at <https://github.com/yczhoualp/ALW>

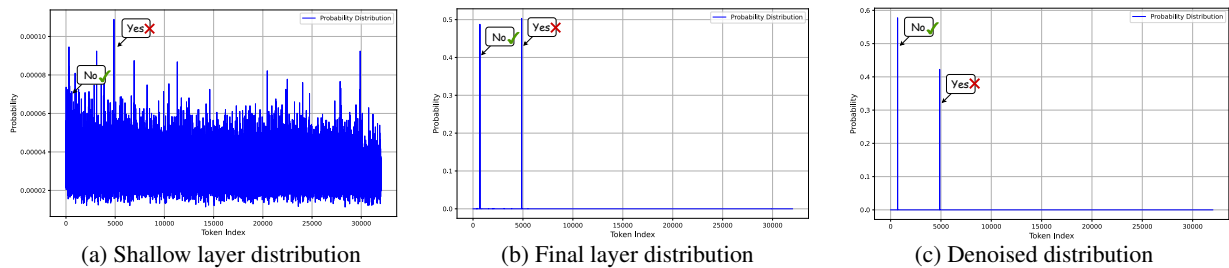


Figure 1: Illustration of the probability distributions during the decoding of the final answer token. The model prompt is: “Q: Would a pear sink in water? A: ... CoT... The answer is”. The model used is LLaMA-7B.

processed by both the LLM and the predictor. The LLM contrasts the final layer with the early exit layer selected by the predictor, effectively filtering noise from shallow layers while preserving key signals in the final layer. Extensive experiments demonstrate that ALW achieves stable performance improvements and significantly enhances the reasoning abilities of LLMs.

Our contributions are as follows:

- We introduce ALW, a novel decoding framework designed to enhance reasoning capabilities by dynamically disentangling shallow noise from deep signals.
- We evaluate ALW on several LLM benchmark datasets across multiple choice tasks and open-ended generation tasks, demonstrating its effectiveness in enhancing reasoning abilities.
- We conduct a comprehensive analysis of ALW’s generalization ability, computational cost, effectiveness, and limitations.

2 Preliminary

2.1 Noise across Different Layers of LLMs

Inspired by noise suppression across attention heads (Ye et al., 2024), we explore the issue of vertical noise propagation (across different layers). Noise can be understood as task-irrelevant interference, such as incorrect attention allocation, which may cause the model to hesitate between multiple unrelated tokens, manifesting as high entropy in the probability distribution. In contrast, lower entropy indicates greater confidence in the next prediction and a reduction in noise.

We use entropy to quantify the noise at each layer. Figure 2 presents the results of a randomly sampled case (more cases are given in Appendix A). We can observe that shallow layers consistently exhibit higher entropy (~ 10), indicating greater noise

and uncertainty, while deeper layers show significantly lower entropy (~ 1), reflecting more stable and refined representations. After contrasting the probability distributions of the final layer and the shallow layer, we obtain a new distribution (“Diff” row in Figure 2) and observe that the entropy at each step is consistently lower than that of the final layer. We attribute this to the information redundancy and noise differences between shallow and deep layers, where the contrastive operation further suppresses the noise, making the probability distribution more focused on the target token.

This is also consistent with the observation in Figure 1, where we examine the distribution just before decoding the final answer. In the final layer, the model struggles to choose between “yes” and “no”, with their probabilities being very close. In fact, this decision is influenced by noise from the shallow layers where the probability of “yes” is much higher than “no”, leading to uncertainty in the final layer. After denoising, the model is able to confidently select the correct token.

2.2 Noise Suppression

From cases to general tasks, we initially attempt to randomly select contrastive layers for noise suppression. However, this approach clearly failed, as different inputs exhibit varying sensitivities to noise distributions. Therefore, we quantify the optimal layer for each input (Section 3.2), where the optimal layer is the one that maximizes the probability of the gold token after denoising. Our goal is not only to generate the correct final answer but also to produce a reasonable reasoning process.

Next, we focus on the effective utilization of the generated data pairs. We explored rule-based methods, such as denoising with the most frequent layer, but these proved unsuccessful (Appendix B). We believe the task of layer selection is complex and requires capturing the implicit relationships between prompts and layers, leading us to utilize

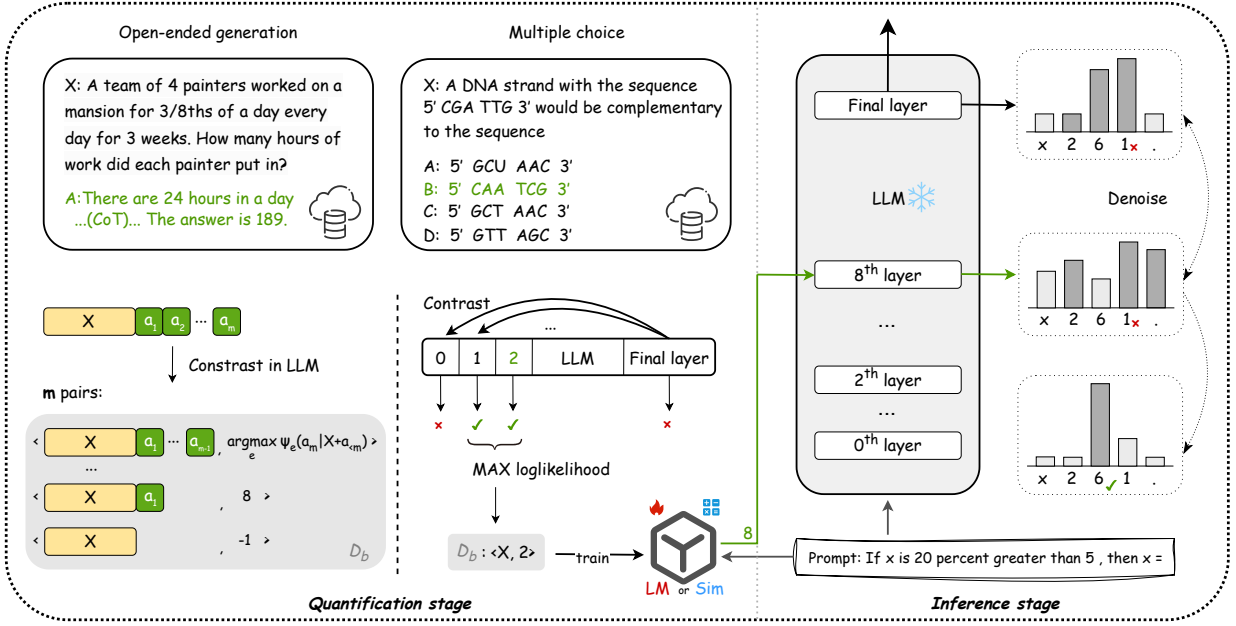


Figure 3: The workflow illustration of ALW in the quantification and inference stage.

such contrastive layers, formulated as follows:

$$\begin{aligned}
 l &= \arg \max_e s_e(T|X), e \in [0, M//2], \\
 \text{st. } s_e(T|X) &\geq \max \{s_e(F_i|X)\}_{i=0}^{K-1}. \quad (5)
 \end{aligned}$$

However, if the score of the option T computed by the final layer matches that of the optimal layer l , the contrastive operation is deemed unnecessary, and l is assigned a value of -1 . Similarly, in generation tasks, if $p_M \geq \text{softmax}(\psi_l)$, l is also set to -1 . Furthermore, to reduce the search space, we focus on the first half of the model’s layers. For example, in the LLaMA-7B model, which consists of 33 early exit layers, the possible values for l range from -1 to 16. Figure 3 illustrates the process of quantifying the optimal layer. In generation tasks, given an answer of length m , we can generate m pairs, whereas in multiple choice tasks, only a single data pair can be generated.

3.3 Layer Prediction

As described above, for each task, we obtain multiple $\langle X, l \rangle$ data pairs. These pairs are stored in a database $\mathcal{D}_b = \{\langle X_i, l_i \rangle\}_{i=0}^{N-1}$, where N is the total number of pairs. We formulate layer prediction as a classification task and train a lightweight model, such as RoBERTa, end-to-end.

Furthermore, to balance accuracy and recall, we assign appropriate weights to each category. Let p_θ represent the prediction probability of the LM. The

model is optimized as follows:

$$\begin{aligned}
 \mathcal{L} &= -\frac{1}{b} \sum_{i=1}^b \sum_j w_j \cdot l_{ij} \log p_\theta(y_{ij} | X_i), \\
 w_j &= \frac{1/n_j}{\sum_j 1/n_j} \propto \frac{1}{n_j}, \quad (6)
 \end{aligned}$$

where b is the batch size, i indexes the samples, j indexes the categories, and n_j represents the frequency of category j in \mathcal{D}_b . An example illustrating the categories is provided in Appendix C.

In addition to the LM-based layer prediction strategy, we introduce a tuning-free method that selects the optimal layer based on the cosine similarity between the input embeddings and the data pairs in \mathcal{D}_b . We refer to this approach as similarity-based layer prediction, denoted as ALW-sim. In contrast, the method that requires LM training to fully activate ALW is termed ALW-lm.

3.4 Decoding Strategy

Drawing on prior research (Li et al., 2023) that used adaptive plausibility constraints to address false positive and false negative errors in contrastive objectives, we extend this to our inference stage. If certain tokens are characterized by high probabilities, they will be grouped into $\mathcal{V}_{\text{head}(x_n|x_{<n})}$:

$$\{x_t \in \mathcal{V} : p_M(x_n|x_{<n}) \geq \alpha \max p_M(x_n|x_{<n})\}, \quad (7)$$

where α is a hyperparameter ranging from 0 to 1, which truncates the distribution of low-probability tokens. During the inference stage, we leverage the final layer to ensure token reliability. If a token

achieves a high probability after contrast but is not recognized by the final layer (not in $\mathcal{V}_{\text{head}}$), we exclude it during the decoding process. Finally, the LLM decodes with ALW-lm or ALW-sim:

$$\psi(x_n|X) = \begin{cases} \log \frac{p_M(x_n|X)}{p_l(x_n|X)} & \text{if } x_n \in \mathcal{V}_{\text{head}}, \\ -\infty & \text{Otherwise,} \end{cases} \quad (8)$$

$l = \arg \max_k p_\theta(y_k|X)$ or $l = \text{Sim}(X, \mathcal{D}_b)$,

where the function $\text{Sim}(\cdot)$ computes the cosine similarity between X and all prompts in \mathcal{D}_b , selecting the prompt with the highest similarity and assigning its corresponding l as the contrastive layer. A detailed procedure is provided in Algorithm 1.

4 Experiments

4.1 Experimental Setup

Tasks. Our framework is evaluated across various tasks. We assess arithmetic reasoning performance on Gsm8k (Cobbe et al., 2021) MathQA (Aida et al., 2019). For commonsense reasoning evaluation, we employ StrategyQA (Geva et al., 2021), and PiQA (Bisk et al., 2020). Additionally, we utilize the logically complex and demanding dataset Folio (Han et al., 2022) for natural language reasoning. Finally, we adopt three widely recognized testing benchmarks, MMLU (Hendrycks et al., 2021), BBH (Suzgun et al., 2022), and ARC-Challenge (Clark et al., 2018). These benchmarks are respected in the field for their diversity and breadth of tasks (details in Appendix D), allowing us to comprehensively analyze the model’s abilities to manage intricate reasoning across varied domains. Among these tasks, Gsm8k, StrategyQA, and Folio are designed as open-ended generation tasks, and the others are multiple-choice tasks.

Models and Baselines. We employ the LLaMA family (Touvron et al., 2023) as our fundamental models, covering sizes of 7B, 13B, 30B, and 65B. In addition, we also consider the updated and more powerful model LLaMA3-8B-Instruct (Dubey et al., 2024). We test our framework against three baselines: Greedy Decoding, Decoding by Contrasting Layers (DoLa) (Chuang et al., 2024), and Contrastive Decoding (CD) (Li et al., 2023). Greedy decoding selects the token with the highest probability at each step, DoLa leverages the Jensen-Shannon divergence between transformer layers within Language Models to enhance factuality, and CD ensures plausible outputs by making a comparison between an LLM and a smaller one.

Algorithm 1 ALW Decoding Strategy

Input: Prompt $X = \{x_0, x_1, \dots, x_{n-1}\}$, LLM, LM or $\text{Sim}(\cdot)$, Output Layer W_o , database \mathcal{D}_b

Output: Next token x_n

- 1: Hidden states $\{h^e\}_{e=0}^M \leftarrow \text{Forward LLM}(X)$
 - 2: $l \leftarrow \text{Forward LM}(X)$ or $\text{Sim}(X, \mathcal{D}_b)$
 - 3: **if** $l = -1$ **then**
 - 4: $\psi(x) = W_o(h^M)$
 - 5: **else**
 - 6: Final prob $p_M = \text{softmax}(W_o(h^M))$
 - 7: Early prob $p_l = \text{softmax}(W_o(h^l))$
 - 8: $\psi(x) = \log(p_M) - \log(p_l)$
 - 9: $Mask \leftarrow \text{Token indices not in } \mathcal{V}_{\text{head}}$
 - 10: $\psi(Mask) \leftarrow -\infty$
 - 11: **end if**
 - 12: **return** $x_n = \arg \max \text{softmax}(\psi(x))$
-

Details. In our experiments, the variable α in Equation 7 is set to 0.1, a value that has shown robustness across various domains (Li et al., 2023). Our predictor utilizes a RoBERTa-base model (Liu et al., 2019) or a sentence-transformer (Reimers and Gurevych, 2019) to compute semantic similarity. For open-ended generation tasks, we implement Chain-of-Thought (Wei et al., 2022). Additional details can be found in Appendix E.

4.2 Results

4.2.1 Arithmetic Reasoning

Results are illustrated in Table 1, where the open-end generation task is indicated by (*). Notably, we maintain the layer search space for DoLa within the range $[0, M//2]$, matching the strategy used in ALW. For CD, we employ LLaMA-7B as the amateur model, and larger models (13B, 30B, 65B) are utilized as expert models. However, due to the absence of LLaMA models smaller than LLaMA-7B, the baseline CD is not included in the experiments conducted at the 7B scale, and the same as LLaMA3-8B-Instruct. ALW significantly enhances the model’s arithmetic reasoning capabilities. Our approach leads the other three baselines across four model scales on Gsm8k and MathQA. Compared to greedy decoding, ALW achieves a substantial increase in performance, which is more pronounced in smaller models. For example, there is a 48% increase on Gsm8k for LLaMA-7B (28% for LLaMA-30B and 9% for 65B).

Models	Arithmetic		Commonsense		General				Average	
	Gsm8k*	MathQ	StrQA*	PiQA	Folio*	MMLU	BBH	ARC-C		
LLaMA-7B-HF	Greedy	13.6	22.3	66.4	78.5	16.5	34.8	34.1	37.0	37.9
	DoLa	12.9	24.8	65.1	76.8	29.8	35.8	38.7	37.9	40.2
	Ours	20.1	36.3	70.3	78.1	29.8	40.6	41.7	41.3	44.7
LLaMA3-8B-Inst	Greedy	76.5	38.1	71.6	79.8	52.5	38.4	43.7	41.7	55.3
	DoLa	73.2	50.2	69.9	77.7	52.1	38.2	49.5	43.5	56.8
	Ours	78.8	56.3	71.6	80.1	52.9	44.2	50.8	47.8	60.3
LLaMA-13B-HF	Greedy	18.9	24.9	67.2	80.1	34.7	36.2	37.2	36.5	42.0
	CD	8.3	22.3	66.4	69.9	37.2	29.5	36.6	34.2	38.1
	DoLa	22.0	30.2	69.9	78.1	39.7	38.0	43.1	40.6	45.2
	Ours	22.7	37.8	72.5	79.6	50.4	38.2	44.5	42.1	48.5
LLaMA-30B-HF	Greedy	32.6	26.3	72.9	81.2	44.6	37.8	39.5	37.2	46.5
	CD	32.6	24.8	70.7	72.2	20.5	31.5	39.2	33.4	40.6
	DoLa	28.0	29.0	73.4	77.0	41.3	37.5	43.9	40.6	46.3
	Ours	41.7	35.7	76.0	81.3	44.6	40.4	45.0	41.0	50.7
LLaMA-65B-HF	Greedy	50.8	27.2	76.0	81.3	42.1	40.4	37.9	39.6	49.4
	CD	43.9	25.1	74.2	72.2	21.5	32.0	35.1	34.9	42.4
	DoLa	54.0	30.4	72.9	66.8	41.3	37.1	42.9	45.1	48.8
	Ours	55.3	38.5	77.7	81.3	43.8	41.4	43.1	47.8	53.6

Table 1: The overall performance of ALW for three types of reasoning tasks across LLaMA-7B, 13B, 30B, 65B and LLaMA3-8B-Instruct. Bold indicates the best results. ALW outperforms three baselines consistently.

4.2.2 Commonsense Reasoning

For commonsense reasoning tasks, LLMs effectively suppress noises from shallow layers, thereby enhancing their commonsense reasoning capabilities. Results show that ALW outperforms other baselines on the StrategyQA benchmark. However, the improvement on PiQA is less pronounced. A closer analysis reveals that in over 90% of instances, the contrast operation was not applied, suggesting that decoding in the final layer is more fruitful than employing the contrast operation. We hypothesize that this could be due to the inherent simplicity of the PiQA dataset, which enables the task to be effectively handled by the shallow layers. As a result, less noise will present in the shallow layers. Further details can be found in Appendix F.

4.2.3 Comprehensive Evaluation

We verify the reasoning capabilities of LLMs on Folio, MMLU, BBH, and ARC-C. In each case, ALW consistently surpasses all baselines. ALW proves particularly efficient in Folio, which involves first-order logic. This efficacy arises from the inadequacies of the shallow layers in grasping deep logical relationships, thereby enabling the deeper layers to circumvent various reasoning errors. Notably, after using ALW, LLaMA-13B saw a significant increase in reasoning abilities on Folio, with a result of 50.4 surpassing all other model settings. Moreover, we present the results of the MMLU and BBH sub-

	BBH	MathQ	PiQA	ARC	MMLU	Gsm8k	Folio	StrQA
7B	59.6	79.8	93.5	80.5	85.5	80.2	82.1	71.2
8B	73.8	91.6	95.4	81.6	90.1	81.3	84.9	75.2
13B	51.0	69.7	93.1	77.2	81.1	77.1	81.6	70.6
30B	46.1	50.1	87.2	62.3	80.3	79.1	84.2	71.3
65B	40.8	43.6	80.2	50.0	75.4	77.0	83.1	71.0

Table 2: Accuracy of predictor in multiple-choice tasks (left); in open-ended generation tasks (right).

tasks in Appendix K and analyze the differences across various sub-tasks and models.

4.2.4 Accuracy of Predictor

For multiple-choice tasks, we report the accuracy of the predictor (lm-based) in the test set, as shown in Table 2 (left). For open-ended generation tasks, we cannot definitively determine whether a generated token is correct or incorrect during generation. However, during the training process, we can maximize the probability of golden tokens after contrasting. Thus, we report the results on the validation set, which is shown in Table 2 (right). It can be observed that in all settings, the accuracy rates are almost always greater than 50%, indicating that the predictor can provide relatively accurate predictions of the optimal layer for a given prompt. In addition, we present the distribution of the predictor output in Appendix G.

Datasets	BERT-base	RoBERTa-base	RoBERTa-large
MathQA	36.1 (-0.2)	36.3	27.4(-8.9)
Folio*	31.4 (+1.6)	29.8	16.5(-13.3)
BBH	40.5 (-1.2)	41.7	40.7(-1.0)

Table 3: Ablation study of predictors. LLaMA-7B with RoBERTa-base achieves the best performance.

4.3 Ablation

Impact of Predictor. We carry out experiments with different pre-trained LMs as predictors to probe the configuration of the ideal model. Table 3 suggests that the application of BERT-base (Devlin et al., 2019) as a predictor produces slightly subpar results in contrast to the use of RoBERTa-base. This is substantiated by the relatively diminished accuracy on MathQA and BBH. However, larger models do not guarantee superior performance. In fact, the use of RoBERTa-large leads to a decline in the overall reasoning competency of LLMs. This may arise from a mismatch between the volume of training data and the size of the model.

Generalization. In Table 4, we present ALW’s generalization performance in-domain and out-of-domain in LLaMA-7B. The mathematical reasoning domain (Domain 1) includes Gsm8k and MathQA; the comprehensive reasoning domain (Domain 2) includes MMLU and BBH; and StrategyQA serves as the commonsense reasoning domain (Domain 3). The arrows indicate values that are lower than those of greedy decoding. Observing the position where the arrow appears, we conclude as follows: whether within the domain or across domains, transferring from multiple-choice tasks to generation tasks is not ideal; conversely, the effect can exceed greedy decoding but does not reach the performance specific to the dataset. Besides this, ALW has demonstrated exceptional generalization performance. For example, in Domain 2, when trained solely on MMLU, ALW achieved an accuracy of 40.4 when reasoning on BBH, reaching 96% of the performance (41.7) in a specific dataset.

Similarity VS LM-based selection. We compare the tuning-free ALW-sim with the lightly fine-tuned ALW-lm. As shown in Figure 5, the simple similarity-based method, while easy to deploy, cannot fully unlock the potential of ALW. Additional experimental results in Table 7 show that ALW-sim outperforms greedy decoding on average, and in most settings it is not weaker than the latter. Furthermore, although ALW-sim performs well, we

Domain	Domain1		Domain2		Domain3
	Gsm8k*	MathQA	MMLU	BBH	StrQA*
Gsm8k*	-	28.4	37.2	40.2	67.7
MathQA	12.1↓	-	37.2	40.7	65.9↓
MMLU	9.85↓	27.2	-	40.4	66.8
BBH	15.9	27.2	37.0	-	65.9↓
StrQA*	13.6	27.5	35.3	40.9	-
Base	13.6	22.3	34.8	34.1	66.4

Table 4: In-domain and out-domain generalization.

Prompt	Would a pear sink in water?
Greedy (×)	A pear is a fruit. Fruits float in water. Thus, a pear would sink in water. So the answer is yes.
Case 1 (×)	... Thus, a pear would float in water. So the answer is yes.
Case 2 (×)	... Thus, a pear would float in water. So the answer is no .
ALW (✓)	A pear would float in water because its density is lower than that of water. So the answer is no.

Table 5: A case demonstrating the effectiveness of ALW.

still recommend using the more advanced ALW, but if training resources are extremely limited, ALW-sim can serve as a substitute.

Random layer. We verify the importance of the predictor by randomly selecting the contrastive layer. The results in Table 8 show that random selection led to a significant decrease in the reasoning accuracy across all settings, with only 25% of the cases outperforming the results of using greedy decoding (highlighted in bold). Therefore, a predictor that learns the patterns of optimal layer occurrence with a quantification strategy is necessary.

The effect of ALW. We explore why ALW is effective, attributing its success to the dynamic correction of reasoning paths. As shown in Table 5, greedy decoding makes an error when reasoning through “pear is fruit” → “fruit float” → “pear sink”. Applying ALW only at the erroneous step corrects the prediction to “would float” (Case 1). However, continuing with greedy decoding afterward still fails to yield the correct answer, while suppressing noise consistently results in a correct answer (Case 2). By analyzing the probability distribution when predicting “yes” or “no” (Figure 1), we observe that the model cannot confidently select an answer, and denoising significantly enhances the confidence while steering the model toward the correct choice.

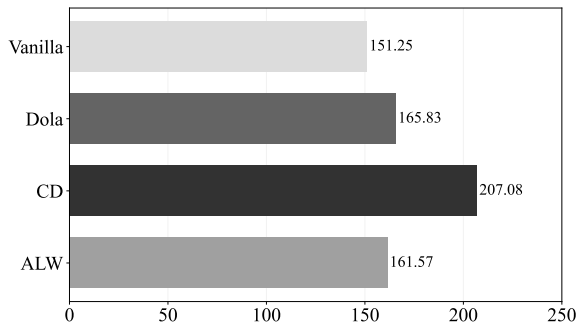


Figure 4: Averaged decoding latency per token in milliseconds (ms/token) for LLaMA-30B.

This suggests that ALW performs probability recalibration at key decision points, amplifying the signal of the correct answer. Notably, if ALW is applied throughout the entire reasoning process, it suppresses the shallow statistical association of “fruit \rightarrow float” while reinforcing the deeper physical rule of “density $<$ water \rightarrow float”. Consequently, ALW guides the reasoning path in a more principled direction from the outset by mitigating noise.

Latency. We further examine the decoding latency. As illustrated in Figure 4, ALW registers only a 6% increase compared to greedy decoding. Compared to DoLa (+10%) and CD (+37%), ALW results in almost no additional latency. Please refer to Appendix H for further analysis and explanation.

Elimination of Errors. Building on the research by Wang et al. (2023), we quantitatively investigate whether reasoning errors have been eliminated in the final layer and, if so, what types of errors are addressed. We analyze on Gsm8k before and after denoising, as shown in Table 9. Aside from “miss step” errors, the noise suppression significantly reduces various errors, particularly those related to model degeneration and arithmetic errors.

In addition, we conduct further analyses, including the rationale for selecting the first half of the layers (Appendix I) and case studies (Appendix J).

5 Related work

5.1 Early-Exiting Approaches

Panda et al. (2016) pioneered the exploration of early exiting in neural networks to accelerate the decoding process. Recently, early exit methods have become integral to efficient language models, as summarized by Khoshnoodi et al. (2024). Layer-Skip (Elhoushi et al., 2024) employs early exit layers as a speculative decoding model, while SkipDecode (Corro et al., 2023) establishes a single exit

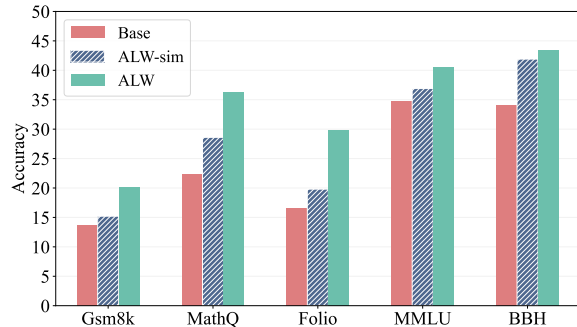


Figure 5: ALW-sim vs ALW across five different datasets in LLaMA-7B.

point for each token in a batch at every sequence position. Contemporaneous work includes EE-LLM (Chen et al., 2024) and ConsistentEE (Zeng et al., 2024), as well as studies summarized by Sajjad et al. (2023). However, these studies emphasize extreme efficiency, such as pruning 40% of the layers while retaining up to 98% of the original performance (Sajjad et al., 2023). In our study, early exiting serves as an *intermediate step*. We use early exit alongside the final layer to capture exploitable patterns for enhancing reasoning, focusing more on performance than efficiency.

5.2 Contrastive Decoding

Su et al. (2022) put forward the notion of Contrastive Search, a method dedicated to accentuating semantic distinctions among tokens within the representational space. Furthermore, Li et al. (2023) introduced Contrastive Decoding, exploiting the disparities between expert and amateur models. In addition, Gera et al. (2023); Chuang et al. (2024) involved experimental contrastive measures between different layers of a single backbone model, producing high-quality text by capitalizing on pre-existing language models. Recently, researchers Arias et al. (2024); Waldendorf et al. (2024); Sennrich et al. (2024) conducted numerous novel studies, primarily focusing on the open-text generation or contrast decoding to mitigate hallucinations. However, we focus on downstream reasoning tasks and offer a broader framework with novel explanations for effectiveness. Additionally, we leverage layer-wise noise patterns to enhance LLMs internally.

6 Conclusion

In this paper, we start from the assumptions of attention noise and explore the noise across different layers of LLMs. We observe that noise accumula-

tion across layers often leads to unstable token predictions during reasoning. To address this issue, we propose adaptive layer-wise contrastive decoding, a search-based decoding approach that enhances reasoning ability by dynamically disentangling noise in shallow layers from critical signals in deep layers. We design a quantification strategy and optimal layer selection methods based on LMs or similarity. The experimental results demonstrate that ALW significantly enhances the reasoning capabilities of LLMs, and further analysis reveals that ALW improves reasoning by focusing on the correct reasoning path through noise suppression.

Limitation

While the adaptive layer-wise contrastive decoding method has significantly enhanced the reasoning capabilities of Large Language Models, it still has some limitations:

- ALW is well-suited for challenging tasks. When the task is relatively simple, the shallow layers are highly confident in the prediction, leading to less noise present in the shallow layers, which can render ALW ineffective.
- We recommend using the more advanced ALW other than ALW-sim, while ALW necessitates training preparation before inference. Although the predictor is small-scale, it still adds additional computational overhead.
- Although ALW enhances reasoning ability by focusing on the correct reasoning path, it cannot fully resolve the issue of correct answers with incorrect paths. While ALW alleviates this, a small residue effect still remains.

7 Acknowledgments

We want to thank all the anonymous reviewers for their valuable comments. This work was supported by the National Science Foundation of China (NSFC No. 62206194), the Natural Science Foundation of Jiangsu Province, China (Grant No. BK20220488), and the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001).

References

Amini Aida, Gabriel Saadia, Lin Peter, Koncel-Kedziorski Rik, Choi Yejin, and Hajishirzi Hannaneh. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

Esteban Garces Arias, Julian Rodemann, Meimingwei Li, Christian Heumann, and Matthias Aßenmacher. 2024. Adaptive contrastive search: Uncertainty-guided decoding for open-ended text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 15060–15080. Association for Computational Linguistics.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Yanxi Chen, Xuchen Pan, Yaliang Li, Bolin Ding, and Jingren Zhou. 2024. EE-LLM: large-scale training and inference of early-exit large language models with 3d parallelism. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

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*.

Luciano Del Corro, Allie Del Giorno, Sahaj Agarwal, Bin Yu, Ahmed Awadallah, and Subhabrata Mukherjee. 2023. Skipdecode: Autoregressive skip decoding with batching and caching for efficient LLM inference. *CoRR*, abs/2307.02628.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

- Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed A Aly, Beidi Chen, and Carole-Jean Wu. 2024. [Layerskip: Enabling early exit inference and self-speculative decoding](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 12622–12642. Association for Computational Linguistics.
- Ariel Gera, Roni Friedman, Ofir Arviv, Chulaka Gunasekara, Benjamin Sznajder, Noam Slonim, and Eyal Shnarch. 2023. The benefits of bad advice: Autocontrastive decoding across model layers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10406–10420. Association for Computational Linguistics.
- Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics (TACL)*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *arXiv preprint arXiv:2501.12948*.
- Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, David Peng, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Shafiq Joty, Alexander R. Fabri, Wojciech Kryscinski, Xi Victoria Lin, Caiming Xiong, and Dragomir Radev. 2022. [Folio: Natural language reasoning with first-order logic](#). *arXiv preprint arXiv:2209.00840*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Ruixin Hong, Xinyu Pang, and Changshui Zhang. 2024. [Advances in reasoning by prompting large language models: A survey](#). *Cybernetics and Intelligence*, pages 1–15.
- Mahsa Khoshnoodi, Vinija Jain, Mingye Gao, Malavika Srikanth, and Aman Chadha. 2024. [A comprehensive survey of accelerated generation techniques in large language models](#). *CoRR*, abs/2405.13019.
- Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. [Contrastive decoding: Open-ended text generation as optimization](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12286–12312. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Sean O’Brien and Mike Lewis. 2023. [Contrastive decoding improves reasoning in large language models](#). *CoRR*, abs/2309.09117.
- Priyadarshini Panda, Abhronil Sengupta, and Kaushik Roy. 2016. Conditional deep learning for energy-efficient and enhanced pattern recognition. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 475–480. IEEE.
- Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. [Reasoning with language model prompting: A survey](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5368–5393. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2023. [On the effect of dropping layers of pre-trained transformer models](#). *Comput. Speech Lang.*, 77:101429.
- Rico Sennrich, Jannis Vamvas, and Alireza Mohammadshahi. 2024. [Mitigating hallucinations and off-target machine translation with source-contrastive and language-contrastive decoding](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 2: Short Papers, St. Julian’s, Malta, March 17-22, 2024*, pages 21–33. Association for Computational Linguistics.
- Jovan Stojkovic, Chaojie Zhang, Íñigo Goiri, Josep Torrellas, and Esha Choukse. 2024. [Dynamollm: Designing LLM inference clusters for performance and energy efficiency](#). *CoRR*, abs/2408.00741.
- Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. *Advances in Neural Information Processing Systems*, 35:21548–21561.

- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Jonas Waldendorf, Barry Haddow, and Alexandra Birch. 2024. [Contrastive decoding reduces hallucinations in large multilingual machine translation models](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024*, pages 2526–2539. Association for Computational Linguistics.
- Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023. [Towards understanding chain-of-thought prompting: An empirical study of what matters](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 2717–2739. Association for Computational Linguistics.
- 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. 2024. [Searching for best practices in retrieval-augmented generation](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 17716–17736. Association for Computational Linguistics.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, and Na Ni. 2024. [Qwen2 technical report](#). *CoRR*, abs/2407.10671.
- Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. 2024. Differential transformer. *arXiv preprint arXiv:2410.05258*.
- Ziqian Zeng, Yihuai Hong, Hongliang Dai, Huiping Zhuang, and Cen Chen. 2024. [Consistentee: A consistent and hardness-guided early exiting method for accelerating language models inference](#). In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 19506–19514. AAAI Press.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024. [A survey of large language models](#). *Preprint*, arXiv:2303.18223.
- Minjun Zhu, Yixuan Weng, Shizhu He, Cunguang Wang, Kang Liu, Li Cai, and Jun Zhao. 2023. [Learning to build reasoning chains by reliable path retrieval](#). In *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*, pages 1–5. IEEE.

A More Preliminary Cases

We provide more preliminary cases in Figure 6 and 7. The observed phenomenon is consistent with the description in Figure 2. Additionally, the contrast operation helps alleviate prediction uncertainty in the final layer, enhancing the model’s confidence in the correct token. This effect arises from removing noise related to shallow layers in the final layer. The token-level representation, rather than the distribution-level representation, is shown in Figure 1.

B Most Frequent Layer

Based on the frequency of layer occurrence in the training set, we selected a common layer to be contrasted during testing. Taking the use of LLaMA-65B for testing on BBH as an example, we considered layers 0 to 40 and the case without contrasting (-1), and tallied the occurrences of the optimal layer. Details are shown in Table 10. We observe that the layer requiring no contrast operation appears most frequently, while other layers are also involved, indicating that denoising is dynamically related to the input. Additionally, shallow layers appear more frequently than higher layers.

As layer 1 appeared 147 times, we fixed the contrast layer to the layer labeled as 1 during testing. We conduct experiments on Gsm8k, StraQA, and PiQA and the results are shown in Table 6. The numbers in parentheses indicate the change in accuracy when using common layer selection compared to using the predictor. The results in bold represent outcomes that exceed the greedy decoding results.

In fact, this idea is similar to the static mode in DoLa. DoLa employs a static method, consistently using the 0th layer to enhance the factuality of the LLMs. They believe that the static method is inferior to the dynamic method (baseline in our paper). However, our main experiments show that ALW outperforms DoLa’s dynamic method. The results just given also prove this point.

	7B	13B	30B	65B
Gsm8k	0/ 12.1(-8.0)	0/ 15.9(-6.8)	0/ 40.9 (-0.8)	0/ 52.3 (-3.0)
PiQA	1/ 78.0(-0.1)	1/ 79.2(-0.4)	1/ 81.2 (-0.1)	2/ 80.5(-0.8)
StraQA	0/ 65.1(-5.2)	0/ 70.3 (-2.2)	0/ 73.4 (-2.6)	0/ 75.5(-2.2)

Table 6: The result of using the most frequent layer as contrast layer. On the left of / is the layer number that appears the most, and on the right is the result.

C A Description about Category

The LLM has an embedding layer and M transformer decode layers, labeled 1 to M . We employ a quantification strategy to find the most important layer l among the first half of the early exit layers. Taking LLaMA-7B as an example, there are 32 decode layers labeled 1 to 32, and we consider layers 1 to 16 and the embedding layer, so the range of l is $[0, 16]$. However, we recognize that there are cases where the final layer itself has sufficient confidence to decode the gold token. In such cases, l is labeled -1 . This is indeed equivalent to the final layer, but we want to highlight that no contrast operation was used, to avoid any confusion among researchers about how layer M could be contrasted with layer M , since l refers to the contrasting layer. The notation $l = -1$ clearly indicates that this is a special case worth noting.

Based on the explanation above, the range of l is $[-1, 16]$, and these 18 categories are the number of classes that the predictor needs to learn to classify.

D Concerns about Evaluation Scope

In Section 4.1, we introduced that in addition to arithmetic and commonsense reasoning datasets, we also tested MMLU and BBH. They have been selected as standard evaluation benchmarks by the Open LLM Leaderboard for assessing large language models, which also proves their versatility and extensive applicability. In addition to MMLU and BBH, we evaluated Folio to assess the logical reasoning capabilities of LLMs, not limiting the scope to just arithmetic and commonsense reasoning.

E More Experiment Details

Alignment of LM and LLM. We believe that a perfect match between RoBERTa tokenizer and the LLM tokenizer is not necessary. When RoBERTa predicts the optimal contrast layer, its input is the full, raw text, rather than the ID sequence tokenized by LLM. The task for RoBERTa is to predict which layer’s noise needs to be suppressed most for the current input, which is a sentence-level or paragraph-level semantic decision, rather than a fine-grained subword match. In fact, this approach is similar to the length prediction method used in LLM scheduling (Stojkovic et al., 2024), where BERT is employed to predict the possible output length for an LLM given the input to plan mem-

Question: When was the Internet first made available to the public?
 Answer: The Internet was first made available to the public in the early 1990s.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Diff	1.69	2.74	2.42	2.02	1.07	0.29	0.41	0.01	0.32	0.97	0.85	1.57	0.01	0.50	0.01	1.36	0.01	1.09	0.97
32	2.60	4.05	3.27	2.61	2.32	0.58	0.74	0.50	0.42	1.38	1.28	1.87	0.42	0.46	0.02	1.41	0.02	0.87	1.39
30	0.01	0.21	0.57	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.03	1.00	0.19	0.01	0.01	1.00	0.01	0.06	0.04
28	0.01	0.08	0.36	0.06	0.04	0.01	0.01	0.04	0.01	0.01	3.14	0.97	0.35	2.04	0.68	3.60	0.21	0.01	0.04
26	0.63	0.48	1.78	0.15	0.67	0.01	0.01	0.02	0.01	0.01	4.42	0.60	1.11	3.57	4.51	5.82	1.77	0.04	0.07
24	0.72	0.36	1.67	0.58	0.68	0.01	0.04	0.01	0.01	0.75	4.07	0.39	2.08	2.90	6.54	6.70	8.29	0.01	0.18
22	2.03	0.05	1.99	0.14	0.66	0.02	0.32	0.02	0.01	2.41	5.57	0.21	2.13	2.19	7.73	2.45	3.86	0.01	1.05
20	1.45	0.03	4.28	0.11	1.07	0.28	0.44	0.09	0.03	8.02	6.44	2.87	4.89	3.32	8.52	3.10	3.44	0.19	7.78
18	8.29	0.35	8.51	5.83	6.13	2.73	2.94	3.38	0.95	9.27	9.05	9.25	9.38	8.99	9.64	4.25	9.60	8.98	9.44
16	9.77	8.76	9.75	9.63	9.80	9.91	9.78	9.90	9.85	9.90	9.98	10.00	9.88	9.97	10.15	10.05	10.02	9.76	9.88
14	10.09	9.69	10.05	10.05	10.06	10.09	10.08	10.10	10.08	10.10	10.16	10.15	10.12	10.13	10.21	10.18	10.13	10.05	10.06
12	10.21	10.19	10.19	10.18	10.18	10.21	10.20	10.21	10.23	10.18	10.22	10.24	10.20	10.21	10.28	10.24	10.21	10.16	10.18
10	10.28	10.28	10.23	10.26	10.25	10.27	10.28	10.27	10.29	10.26	10.26	10.29	10.27	10.27	10.32	10.29	10.27	10.25	10.25
8	10.30	10.32	10.29	10.30	10.31	10.30	10.31	10.31	10.32	10.31	10.31	10.32	10.30	10.30	10.33	10.33	10.31	10.31	10.30
6	10.34	10.34	10.33	10.34	10.34	10.34	10.33	10.34	10.35	10.34	10.34	10.35	10.33	10.33	10.35	10.34	10.33	10.34	10.33
4	10.36	10.36	10.35	10.36	10.36	10.36	10.35	10.36	10.36	10.35	10.36	10.36	10.35	10.35	10.36	10.36	10.35	10.35	10.35
2	10.37	10.36	10.37	10.37	10.37	10.37	10.37	10.37	10.35	10.37	10.37	10.35	10.37	10.37	10.37	10.37	10.37	10.37	10.37
0	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37

⌘ 'The' 'Internet' 'was' 'first' 'made' 'available' 'to' 'the' 'public' 'in' 'the' 'early' '1' '9' '9' '0' 's'

Figure 6: More preliminary cases (case 1).

Question: When did the first human land on the Moon?
 Answer: The first humans landed on the Moon on July 20, 1969, during NASA's Apollo 11 mission.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Diff	2.82	1.74	1.08	1.01	0.01	0.13	0.18	0.88	1.02	0.01	0.12	0.55	0.37	0.37	0.01	0.01	0.01	0.01	0.01	1.26	2.39	0.74	1.34	0.01	0.01	0.01	0.18	0.01	0.01	0.26	0.15
32	4.61	4.22	2.12	2.15	0.09	0.45	0.71	0.41	1.88	1.37	0.22	0.14	0.57	0.71	1.08	0.04	0.03	0.03	0.02	1.91	3.11	1.09	1.79	0.13	0.11	0.02	0.31	0.01	0.06	0.76	0.64
30	1.64	1.28	0.10	0.05	0.02	0.01	0.28	0.01	0.15	0.85	0.08	0.01	0.14	0.57	1.12	0.01	0.01	0.01	0.02	0.81	1.31	0.64	0.09	1.24	0.01	0.01	0.87	0.01	0.01	1.06	0.01
28	1.50	1.01	0.06	0.56	0.01	0.01	0.65	0.01	0.24	1.60	0.34	0.13	0.30	0.01	0.75	0.43	0.27	0.42	1.20	2.12	1.91	0.13	0.46	1.46	0.04	0.01	1.07	0.96	1.78	1.00	0.86
26	2.22	1.07	0.15	0.01	0.01	0.58	0.71	0.03	0.45	0.77	2.02	0.23	1.21	0.01	1.93	0.39	0.05	0.48	0.68	4.12	1.86	0.06	1.32	1.75	0.03	0.65	0.72	1.10	0.39	1.00	0.22
24	1.64	0.90	0.65	0.28	0.01	0.34	0.99	0.32	1.80	0.58	3.95	0.07	1.08	0.01	2.01	0.26	0.33	0.90	0.90	3.85	1.76	0.15	0.85	3.47	0.11	4.11	0.85	0.91	0.39	1.01	1.22
22	1.97	1.02	0.25	1.88	0.12	0.63	0.94	0.51	3.19	0.53	4.26	0.09	0.58	0.14	3.56	0.25	0.60	0.79	0.69	6.64	4.23	0.69	1.94	8.00	0.28	6.49	0.89	0.82	0.50	1.35	3.46
20	1.72	0.13	0.66	7.46	6.75	3.73	0.77	0.69	8.98	2.24	3.22	1.52	0.70	2.00	2.16	2.01	2.12	8.72	8.37	6.30	8.47	8.14	1.81	8.67	7.14	2.94	5.01	8.11	9.42		
18	8.08	3.31	8.58	9.56	9.23	9.74	9.09	7.55	9.79	9.70	9.54	9.57	8.70	9.34	9.82	9.57	9.71	8.03	8.84	9.84	9.81	9.75	9.57	9.84	9.13	9.57	9.71	8.78	9.41	9.87	9.92
16	9.76	9.68	9.86	10.01	9.91	10.00	9.97	9.94	10.00	10.04	9.83	10.07	9.96	10.02	10.05	10.08	10.14	10.09	10.00	10.03	10.00	10.04	9.94	10.08	10.01	9.90	9.96	10.02	10.12	10.09	10.03
14	10.12	10.04	10.15	10.17	10.13	10.15	10.15	10.13	10.14	10.18	10.12	10.19	10.19	10.15	10.15	10.19	10.20	10.21	10.16	10.14	10.17	10.17	10.08	10.11	10.13	10.11	10.10	10.12	10.20	10.15	10.15
12	10.24	10.24	10.23	10.22	10.21	10.22	10.24	10.23	10.26	10.24	10.25	10.27	10.21	10.23	10.25	10.28	10.26	10.25	10.22	10.26	10.25	10.17	10.12	10.22	10.20	10.18	10.21	10.24	10.20	10.22	
10	10.29	10.30	10.28	10.27	10.26	10.28	10.29	10.29	10.28	10.30	10.29	10.29	10.30	10.28	10.29	10.29	10.33	10.30	10.28	10.28	10.31	10.30	10.24	10.12	10.28	10.28	10.24	10.27	10.28	10.27	10.29
8	10.33	10.33	10.32	10.30	10.29	10.31	10.31	10.32	10.31	10.32	10.32	10.31	10.33	10.32	10.33	10.32	10.34	10.33	10.31	10.32	10.33	10.32	10.28	10.20	10.29	10.31	10.29	10.30	10.30	10.30	10.32
6	10.35	10.35	10.35	10.34	10.33	10.34	10.34	10.35	10.34	10.34	10.34	10.34	10.35	10.35	10.35	10.34	10.35	10.35	10.34	10.34	10.35	10.34	10.33	10.34	10.32	10.34	10.33	10.33	10.34	10.33	10.34
4	10.36	10.36	10.35	10.36	10.36	10.35	10.36	10.36	10.36	10.36	10.36	10.36	10.36	10.36	10.36	10.36	10.36	10.36	10.35	10.36	10.37	10.36	10.35	10.36	10.35	10.35	10.35	10.36	10.36	10.36	10.35
2	10.36	10.36	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.36	10.37	10.36	10.37	10.37	10.37	10.37	10.37	10.37
0	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37	10.37

⌘ 'The' 'first' 'humans' 'land' 'ed' 'on' 'the' 'Moon' 'on' 'July' '2' '0' '1' '9' '6' '9' 'd' 'uring' 'NASA' 's' 'A' 'pol' 'lo' '1' '1' 'mission'

Figure 7: More preliminary cases (case 2).

ory usage, and such methods often do not require alignment between PLM and LLM tokenizers.

α for multiple choice tasks. When handling multiple choice questions, we compute the log probability sum of a given sentence. However, if any token in the sentence is not present in \mathcal{V}_{head} , its score would be $-\infty$ according to Equation 8, which would make the log probability sum of the entire sentence $-\infty$, which is unreasonable. To address this, for tokens not found in \mathcal{V}_{head} , we apply a penalty of -1000 instead of $-\infty$, ensuring a more stable and reasonable computation.

GPUs. The main experiments were conducted on an NVIDIA GeForce RTX 3090. When running ALW with LLaMA-7B as the LLM and RoBERTa-base as the predictor, one RTX 3090 is required. The predictor remains the same. For the 13B model, two 3090 GPUs are required; for the 30B model, four GPUs; and for the 65B model, eight GPUs.

Since the predictor’s inference speed is very fast and does not require a lot of GPU communication, the budget is essentially consistent with simply inferring the LLaMA model.

Others. The number of shots assigned for the three open-ended generation datasets is six shots for StrategyQA, eight for Gsm8k, and three for Folio. Moreover, we present the CoT prompts for three open-ended generation datasets: Gsm8k, StrategyQA, and Folio. The CoT prompts for Gsm8k and StrategyQA are constructed according to the method outlined in (Wei et al., 2022). For the Folio dataset, prompts are constructed by a random selection and construction process, which is described in Table 13. In addition, when using ALW-sim, in order to balance the data distribution, we discard some data labeled -1 to maintain balance. Relatively, we added learning weight when using ALW-lm (Sec 3.3).

F Concerns about Performance Variability

For concerns about variability in performance gains across different tasks and model sizes, we believe it is normal for some variation in gains across different settings. What we have developed is a universal reasoning framework that is applicable to various situations and having the added benefit is more important. We have achieved this goal.

In Section 4, we demonstrate the robust capabilities of ALW across eight datasets and five model sizes, ensuring that it can enhance reasoning abilities in most cases. It shows strong performance on datasets such as BBH, ARC-C, Gsm8k, and MathQA. For PiQA, where the gain in performance is smaller, we attempt to analyze in Section 4.3 that this may be because, for easier tasks, fewer noises are presented in shallow layers.

G The Distribution of Predictor Outputs

We present the distribution of layers predicted by the predictor when testing Gsm8k with LLaMA-7B in Table 11. On 5,621 occasions, the predictor chose to skip the contrast operation. When a contrast is required, the predictor does not have an extreme tendency to predict a specific layer, and the overall distribution shows a trend of decreasing predicted frequency as the layer number increases. This shows that modeling the relationship between prompt and layer is not a simple task.

H Concerns about Decoding Latency

We test the average time needed to decode each token using decoding methods on four NVIDIA GeForce RTX 3090 GPUs for the LLaMA-30B. As shown in Figure 4, ALW incurs the least time expenditure, being only 1.06 times that of the base, with virtually no increase in time cost.

Moreover, we consider additional operations from a mathematical perspective. For ALW-sim, we need to additionally calculate the cosine similarity of each row in the semantic representation vector $[1, d]$ and the precomputed representation vector $[n, d]$, which can be approximated as matrix multiplication of two $[n, d]$ vectors, with a computational cost of only dn ($d=384$ in our experiment).

For ALW-lm, the additional operation is the forward process of LM and contrast operation. The RoBERTa-base model in our experiment has 110 million parameters, and if we use LLaMA-7B, its parameter count is 64 times that of RoBERTa-base.

Thus, for LLM inference, the inference time of the LM can be considered negligible. Furthermore, since LM inference runs in parallel with LLM, as long as the forward pass of the LM is shorter than the inference time of the LLM, the time impact introduced by the LM can be completely disregarded. In this case, the additional computations involve only **the subtraction of two hidden states and the softmax operation**.

I Why First Half of All Layers?

As shown in Figure 2, 6 and 7, after surpassing half of the layers, the layers contain less noise and the predictions become more focused. That is, deeper layers tend to retain clearer signals, distinguishing them from the noise in the shallow layers. Therefore, theoretically, subtracting the deeper layers in the final stage is unlikely to yield better results. Moreover, if we consider only the latter half of the layers, the search space is consistent with the first half but the occurrence is less frequent, covering a narrower range of scenarios. On the other hand, using all layers from 0 to 32 undoubtedly greatly increases the search space, and the increased number of classifications makes it more difficult for the predictor to train. We conducted supplementary experiments in Table 12. We searched within the latter half of the layers, and the results indicated a significant decrease in performance. This is consistent with the phenomena observed in Dola.

J Qualitative Study

This section presents generation cases from ALW and baselines across three open-ended generation datasets: Gsm8k (Table 16), StrategyQA (Table 17), and Folio (Table 18). Using Folio as an example, in the case shown in Table 18, ALW and greedy decoding produced entirely opposite generations when decoding $\forall x (\text{Student}(x) \rightarrow \text{Teacher}(x))$. The erroneous reasoning in the greedy decoding led to $\neg \text{Student}(\text{leon}) \rightarrow \neg \text{Teacher}(\text{leon})$, resulting in the conclusion “unknown”. In contrast, ALW accurately expressed $\forall x (\text{Teacher}(x) \rightarrow \neg \text{Student}(x))$ according to the given description, correcting the erroneous reasoning path of the greedy decoder and leading to the correct result “True”.

However, there are issues where the result is correct but the reasoning path is incorrect, and we analyze this situation in Table 19. In fact, we have explained in Section 4.3 that the reason ALW improves reasoning is that LLMs often struggle with

Models		Arithmetic		Commonsense		General				Average
		Gsm8k*	MathQ	StrQA*	PiQA	Folio*	MMLU	BBH	ARC-C	
LLaMA-7B-HF	Greedy	13.6	22.3	66.4	78.5	16.5	34.8	34.1	37.0	37.9
	Ours-sim	15.2	28.6	66.4	78.0	19.8	36.9	39.5	40.1	41.0
	Ours-lm	20.1	36.3	70.3	78.1	29.8	40.6	41.7	41.3	45.5
LLaMA-8B-HF	Greedy	76.5	38.1	71.6	79.8	52.5	38.4	43.7	41.7	55.3
	Ours-sim	76.5	44.7	70.2	78.3	52.5	40.4	46.6	42.3	56.4
	Ours-lm	78.8	56.3	71.6	80.1	52.9	44.2	50.8	47.8	60.3
LLaMA-13B-HF	Greedy	18.9	24.9	67.2	80.1	34.7	36.2	37.2	36.5	42.0
	Ours-sim	14.4	30.2	70.2	79.7	38.8	37.2	40.1	37.9	43.9
	Ours-lm	22.7	37.8	72.5	79.6	50.4	38.2	44.8	42.1	48.5
LLaMA-30B-HF	Greedy	32.6	26.3	72.9	81.2	44.6	37.8	39.5	37.2	46.5
	Ours-sim	32.6	28.3	70.2	81.2	44.6	39.6	41.7	38.5	47.4
	Ours-lm	41.7	35.7	76.0	81.3	44.6	40.4	45.0	41.0	50.7
LLaMA-65B-HF	Greedy	50.8	27.2	76.0	81.3	42.1	40.4	37.9	39.6	49.4
	Ours-sim	46.2	31.5	74.2	80.1	42.1	41.0	39.2	43.1	50.1
	Ours-lm	55.3	38.5	77.7	81.3	43.8	41.4	43.1	47.8	53.6

Table 7: The performance of ALW-sim and ALW-lm across all settings. The non-bold entries indicate that ALW-sim is weaker than greedy decoding. The performance of ALW-sim lies between the base and ALW. We recommend using the more advanced ALW, but if training resources are extremely limited, ALW-sim can serve as a substitute.

Datasets	LLaMA-7B-HF	LLaMA-13B-HF	LLaMA-30B-HF	LLaMA-65B-HF
Gsm8k*	11.4 (-8.7)	18.9 (-3.8)	32.6 (-9.1)	50.0 (-5.3)
PiQA	74.4 (-3.7)	72.8 (-6.8)	74.9 (-6.4)	73.9 (-7.4)
Folio*	26.4 (-3.4)	38.8 (-11.6)	33.1 (-11.5)	32.2 (-11.6)
MMLU	35.9 (-4.7)	34.4 (-3.8)	37.9 (-2.5)	36.0 (-5.4)

Table 8: The importance of layer predictor. The numbers in parentheses indicate the change in accuracy when using random layer selection compared to using the predictor. The results in bold represent outcomes that exceed the greedy decoding results.

Contrast	Arithmetic	Missing Step	Semantic	Degeneration	Total Errors
×	6.1%	30.3%	42.4%	7.6%	86.4%
✓	5.3%	30.3%	40.9%	3.0%	79.5%

Table 9: Proportion of errors in of a set of 132 **Gsm8k** questions. Contrast significantly mitigates various errors, especially the model’s degeneration and arithmetic errors.

layer	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
num	1742	30	147	131	29	36	29	20	27	48	28	31	46	19	82	51	47	40	37	33	67
layer	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
num	58	25	23	9	21	16	23	29	22	28	35	25	18	50	30	33	27	16	21	9	16

Table 10: The statistics of the optimal layer. Using LLaMA-65B and BBH datasets. The layer requiring no contrast operation appears most frequently, while other layers are also involved. Moreover, shallow layers appear more frequently than higher layers.

layer	n	layer	n
-1	5621	8	236
0	1358	9	130
1	1114	10	256
2	928	11	161
3	964	12	154
4	825	13	512
5	335	14	186
6	448	15	86
7	741	16	252

Table 11: The distribution of predictor outputs

	7B	13B	30B
[0, 16]	20.1	22.7	41.7
[16, 32]	12.1	15.2	30.3
Vanilla	13.6	18.9	32.6

Table 12: The comparison between results of using different model sizes in Gsm8k.

path decisions, and ALW enables the model to focus confidently on deep reasoning signals, thereby making the correct next-step choice.

Observing an example from Gsm8k in Table 19, ALW incorrectly multiplies the task time by the total quantity. However, the erroneous calculations for small and large holes happen to cancel each other out ($30 \times 3 + 15 \times 10 = 240$ minutes = 4 hours), which coincidentally matches the correct calculation ($30 \times 3 = 90$ minutes, $15 \times 10 = 150$ minutes, total = 240 minutes). Contrastive decoding works when the model struggles to choose between the correct or incorrect token. That is, if the model strongly favors an incorrect token (e.g., 95% probability), it is difficult to change the relative probabilities between the incorrect and correct tokens through contrastive decoding. We believe the correctness of the reasoning path depends on the model’s understanding of the problem. If, for a given model, there is no correct path in the path space or if its probability is very low, it is indeed difficult to correct it using ALW. In fact, when comparing the outputs from ALW and greedy decoding, the vanilla output is repetitive, indicating that the model was already confused about the problem. In this case, ALW can only improve the quality of the content in these cases. Observing an example from Folio, all reasoning paths are correct except for “ $\forall x$ (BasketballPlayer(x) \rightarrow American(x))”. In fact, because ALW has already inferred that Yuri is not a

basketball player, it can already answer that Yuri is not an American basketball player. The statement “ $\forall x$ (BasketballPlayer(x) \rightarrow American(x))” should have been correctly modeled as “ \exists ”, which might be due to the very limited presence of data including “ \exists ” in the training set.

K Sub-task Results

BBH Sub-tasks. As shown in Table 14, all models on the “penguins_in_a_table” task show significant improvement (+40.0 to +67.0), with an average score of 82.6 (+53.4). This task involves table-based data reasoning, where ALW can more effectively extract structured information by denoising, thereby reducing redundant interference. Additionally, all models exhibit significant improvement on “snarks”, with an average increase of 13.1 points. The “snarks” typically involves implicit meaning in complex contexts, which heavily relies on the model’s understanding of deep semantic and contextual relationships. ALW enhances the model’s ability to capture deep semantic features by suppressing shallow noise (e.g., literal ambiguity, surface feature interference).

However, performance is suboptimal on some tasks, such as “movie_recommendation”, where the average score drops to 68.0 (−7.2) after using ALW, significantly lower than other semantic tasks. Movie recommendation requires integrating multi-dimensional information, such as user preferences (e.g., actors, directors, genres) and context (e.g., time, ratings). Shallow layers may capture local high-frequency features (e.g., “sci-fi”, “high rating”), but ALW mistakenly treats them as noise to be suppressed, leading to the loss of key signals.

MMLU Sub-tasks. As shown in Table 15, there are differences in the performance of various sub-tasks across different models, particularly in more complex domains (such as economics and law), where performance fluctuations are more pronounced. In certain sub-tasks, ALW significantly improves reasoning performance through shallow layer adaptive selection and final layer denoising strategies. For example, more noticeable improvements can be observed in tasks related to mathematics and medicine. However, in areas related to philosophy and management, while some improvement is seen, the overall enhancement may be smaller, indicating that the reasoning process in these tasks may inherently be more straightforward or simple, requiring less denoising.

	prompt
shot-1	<p>Context: All kids love animals. All toddlers are kids. If someone loves animals, then they are not bad. All pirates are bad. If Nancy is not a pirate, then Nancy loves animals. If Nancy is not a toddler, then Nancy is bad.</p> <p>Question: The statement "Nancy is a pirate" is True, False or Unknown?</p> <p>A: $\forall x (\text{Kid}(x) \rightarrow \text{LoveAnimals}(x)), \forall x (\text{Toddler}(x) \rightarrow \text{Kid}(x)), \forall x (\text{LoveAnimals}(x) \rightarrow \neg \text{Bad}(x)), \forall x (\text{Pirate}(x) \rightarrow \text{Bad}(x)), \neg \text{Pirate}(\text{nancy}) \rightarrow \text{LoveAnimals}(\text{nancy}), \neg \text{Toddler}(\text{nancy}) \rightarrow \text{Bad}(\text{nancy})$. The statement is Unknown.</p>
shot-2	<p>Context: Oxford Circus is a road junction connecting Oxford Street and Regent Street. Oxford Street is in London. John Nash designed the construction on Regent Street. John Nash designed Oxford Circus. John Nash is an architect in British. Oxford Circus is the entrance to Oxford Circus tube station which is a part of the Central line in 1900.</p> <p>Question: The statement "Oxford Circus is in London" is True, False or Unknown?</p> <p>A: $\text{OxfordStreet}(\text{oxfordCircus}) \wedge \text{RegentStreet}(\text{oxfordCircus}), \forall x (\text{OxfordStreet}(x) \rightarrow \text{London}(x)), \exists x (\text{RegentStreet}(x) \wedge \text{Design}(\text{johnNash}, x)), \text{Design}(\text{johnNash}, \text{oxfordCircus}), \text{British}(\text{johnNash}) \wedge \text{Architect}(\text{johnNash}), \text{CentralLine}(\text{oxfordCircus})$. The statement is True.</p>
shot-3	<p>Context: Buisnesses are either sanctioned or unsanctioned. Sanctioned buisnesses are limited. Unsanctioned buisnesses are free. The Crude Oil Data Exchange is a buisness that isn't free.</p> <p>Question: The statement "Crude Oil Data Exchange is unsanctioned" is True, False or Unknown?</p> <p>A: $\forall x (\text{Buisness}(x) \rightarrow \text{Sanctioned}(x) \wedge \neg \text{Sanctioned}(x)), \forall x (\text{Buisness}(x) \wedge \text{Sanctioned}(x) \rightarrow \text{Limited}(x)), \forall x (\text{Buisness}(x) \wedge \neg \text{Sanctioned}(x) \rightarrow \text{Free}(x)), \text{Buisness}(\text{crudeoildataexchange}) \wedge \neg \text{Free}(\text{crudeoildataexchange})$. The statement is False.</p>

Table 13: CoT prompt for Folio.

Subtasks	LLaMA-7B	LLaMA3-8B	LLaMA-13B	LLaMA-30B	LLaMA-65B	Average
temporal_sequences	12.0 _{4.0}	40.0 _{0.0}	36.0 _{24.0}	20.0 _{4.0}	20.0 _{4.0}	25.6 _{7.2}
disambiguation_qa	28.0 _{-8.0}	24.0 _{-4.0}	28.0 _{-8.0}	32.0 _{4.0}	28.0 _{0.0}	28.0 _{-3.2}
date_understanding	88.0 _{20.0}	76.0 _{12.0}	80.0 _{8.0}	72.0 _{-8.0}	76.0 _{-4.0}	78.4 _{5.6}
tracking_shuffled_objects_three_objects	24.0 _{4.0}	40.0 _{0.0}	32.0 _{4.0}	40.0 _{8.0}	24.0 _{0.0}	32.0 _{3.2}
penguins_in_a_table	73.3 _{40.0}	100.0 _{67.0}	86.7 _{46.7}	80.0 _{60.0}	73.3 _{53.3}	82.6 _{53.4}
geometric_shapes	8.0 _{0.0}	12.0 _{4.0}	8.0 _{0.0}	8.0 _{0.0}	12.0 _{4.0}	9.6 _{1.6}
snarks	72.2 _{16.0}	72.2 _{11.0}	77.8 _{10.8}	72.2 _{11.2}	77.8 _{16.7}	74.4 _{13.1}
ruin_names	60.0 _{4.0}	76.0 _{12.0}	64.0 _{12.0}	80.0 _{16.0}	64.0 _{0.0}	68.8 _{8.8}
tracking_shuffled_objects_seven_objects	16.0 _{4.0}	16.0 _{0.0}	20.0 _{-8.0}	20.0 _{0.0}	20.0 _{-4.0}	18.4 _{-1.6}
tracking_shuffled_objects_five_objects	24.0 _{20.0}	16.0 _{-4.0}	8.0 _{0.0}	12.0 _{4.0}	14.0 _{-14.0}	14.8 _{1.2}
logical_deduction_three_objects	60.0 _{20.0}	64.0 _{0.0}	48.0 _{12.0}	52.0 _{0.0}	48.0 _{4.0}	54.4 _{7.2}
hyperbaton	72.0 _{-16.0}	96.0 _{24.0}	84.0 _{0.0}	80.0 _{-12.0}	84.0 _{0.0}	83.2 _{-0.8}
logical_deduction_five_objects	24.0 _{4.0}	32.0 _{4.0}	28.0 _{12.0}	36.0 _{12.0}	28.0 _{8.0}	29.6 _{8.0}
logical_deduction_seven_objects	8.0 _{0.0}	40.0 _{-4.0}	24.0 _{4.0}	44.0 _{8.0}	24.0 _{8.0}	28.0 _{3.2}
movie_recommendation	64.0 _{-8.0}	76.0 _{-8.0}	72.0 _{0.0}	56.0 _{-20.0}	72.0 _{0.0}	68.0 _{-7.2}
salient_translation_error_detection:	24.0 _{8.0}	32.0 _{0.0}	16.0 _{0.0}	16.0 _{0.0}	24.0 _{8.0}	22.4 _{3.2}
all	41.1 _{7.0}	50.8 _{7.1}	44.5 _{7.3}	45.0 _{5.5}	43.1 _{5.2}	44.9 _{6.4}

Table 14: The results of BBH subtasks. The subscript indicates the difference from the greedy decoding, where red represents a lower value than greedy decoding, and blue indicates a value that is not lower than greedy decoding.

Subtasks	LLaMA-7B	LLaMA3-8B	LLaMA-13B	LLaMA-30B	LLaMA-65B	Average
abstract_algebra	60.0 _{0.0}	50.0 _{-10.0}	50.0 _{-10.0}	80.0 _{30.0}	70.0 _{20.0}	62.0 _{6.0}
anatomy	42.9 _{0.0}	50.0 _{0.0}	57.1 _{0.0}	57.1 _{0.0}	50.0 _{-7.1}	51.4 _{-1.4}
astronomy	25.0 _{-6.2}	37.5 _{0.0}	31.2 _{0.0}	37.5 _{-6.2}	37.5 _{-6.2}	33.8 _{-3.8}
business_ethics	90.0 _{20.0}	60.0 _{0.0}	60.0 _{-10.0}	60.0 _{-10.0}	80.0 _{10.0}	70.0 _{2.0}
clinical_knowledge	51.9 _{11.1}	44.4 _{3.7}	48.1 _{7.4}	40.7 _{7.4}	48.1 _{3.7}	46.7 _{6.7}
college_biology	26.7 _{13.3}	40.0 _{6.7}	40.0 _{13.3}	33.3 _{0.0}	33.3 _{0.0}	34.7 _{6.7}
college_chemistry	40.0 _{0.0}	50.0 _{-10.0}	20.0 _{-20.0}	40.0 _{0.0}	10.0 _{-20.0}	32.0 _{-10.0}
college_computer_science	50.0 _{10.0}	50.0 _{10.0}	60.0 _{40.0}	50.0 _{0.0}	50.0 _{0.0}	52.0 _{12.0}
college_mathematics	30.0 _{20.0}	20.0 _{10.0}	40.0 _{20.0}	40.0 _{30.0}	60.0 _{30.0}	38.0 _{22.0}
college_medicine	33.3 _{0.0}	50.0 _{16.7}	50.0 _{11.1}	44.4 _{22.2}	38.9 _{16.7}	43.3 _{13.3}
college_physics	27.3 _{0.0}	36.4 _{9.1}	27.3 _{0.0}	36.4 _{9.1}	36.4 _{9.1}	32.7 _{5.5}
computer_security	40.0 _{0.0}	50.0 _{0.0}	40.0 _{-10.0}	50.0 _{0.0}	50.0 _{0.0}	46.0 _{-2.0}
conceptual_physics	62.5 _{4.2}	70.8 _{8.3}	41.7 _{-4.2}	54.2 _{-12.5}	66.7 _{-4.2}	59.2 _{-1.7}
econometrics	33.3 _{0.0}	33.3 _{-16.7}	25.0 _{0.0}	25.0 _{0.0}	25.0 _{0.0}	28.3 _{-3.3}
electrical_engineering	20.0 _{-13.3}	46.7 _{0.0}	33.3 _{13.3}	20.0 _{-6.7}	20.0 _{-6.7}	28.0 _{-2.7}
elementary_mathematics	44.7 _{15.8}	55.3 _{21.1}	28.9 _{-2.6}	34.2 _{0.0}	39.5 _{2.6}	40.5 _{7.4}
formal_logic	38.5 _{7.7}	46.1 _{7.7}	38.5 _{-7.7}	61.5 _{15.4}	46.1 _{7.7}	46.2 _{6.2}
global_facts	40.0 _{10.0}	70.0 _{20.0}	60.0 _{30.0}	60.0 _{20.0}	50.0 _{10.0}	56.0 _{18.0}
high_school_biology	32.3 _{19.4}	58.1 _{32.2}	32.3 _{3.2}	41.9 _{6.5}	35.5 _{-3.2}	40.0 _{11.6}
high_school_chemistry	42.9 _{19.1}	57.1 _{19.0}	38.1 _{0.0}	38.1 _{-4.8}	47.6 _{19.1}	44.8 _{10.5}
high_school_computer_science	60.0 _{30.0}	40.0 _{20.0}	50.0 _{30.0}	60.0 _{40.0}	60.0 _{30.0}	54.0 _{30.0}
high_school_european_history	41.2 _{-11.8}	29.4 _{-11.8}	29.4 _{-23.5}	29.4 _{0.0}	29.4 _{-17.6}	31.8 _{-12.9}
high_school_geography	60.0 _{5.0}	55.0 _{0.0}	55.0 _{0.0}	50.0 _{0.0}	50.0 _{-5.0}	54.0 _{0.0}
high_school_government_and_politics	65.0 _{0.0}	50.0 _{-5.0}	60.0 _{-10.0}	65.0 _{-10.0}	45.0 _{-35.0}	57.0 _{-12.0}
high_school_macro_economics	30.8 _{-5.1}	33.3 _{-2.6}	20.5 _{-12.8}	30.8 _{-2.6}	35.9 _{2.6}	30.3 _{-4.1}
high_school_mathematics	40.7 _{25.9}	66.7 _{55.6}	40.7 _{7.4}	37.0 _{7.4}	40.7 _{11.1}	45.2 _{21.5}
high_school_micro_economics	45.8 _{20.8}	50.0 _{8.3}	45.8 _{4.2}	37.5 _{-4.2}	50.0 _{8.3}	45.8 _{7.5}
high_school_physics	56.2 _{-6.2}	50.0 _{-12.5}	56.2 _{6.2}	68.8 _{18.8}	62.5 _{18.8}	58.8 _{5.0}
high_school_psychology	54.5 _{0.0}	67.3 _{10.9}	54.5 _{5.5}	54.5 _{-1.8}	54.5 _{-10.9}	57.1 _{0.7}
high_school_statistics	27.3 _{9.1}	40.9 _{18.2}	27.3 _{9.1}	40.9 _{22.7}	22.7 _{4.5}	31.8 _{12.7}
high_school_us_history	33.3 _{4.8}	42.9 _{28.6}	33.3 _{4.8}	38.1 _{19.1}	33.3 _{-4.8}	36.2 _{10.5}
high_school_world_history	20.8 _{4.2}	33.3 _{8.3}	20.8 _{0.0}	25.0 _{8.3}	33.3 _{16.7}	26.7 _{7.5}
human_aging	43.5 _{0.0}	47.8 _{13.1}	52.2 _{8.7}	65.2 _{13.1}	39.1 _{-13.0}	49.6 _{4.3}
human_sexuality	42.9 _{0.0}	42.9 _{7.2}	50.0 _{7.1}	35.7 _{0.0}	50.0 _{14.3}	44.3 _{5.7}
international_law	30.8 _{15.4}	23.1 _{7.7}	23.1 _{-7.7}	23.1 _{7.7}	38.5 _{15.4}	27.7 _{7.7}
jurisprudence	18.2 _{9.1}	45.5 _{9.1}	18.2 _{-9.1}	27.3 _{-9.1}	36.4 _{-18.2}	29.1 _{-3.6}
logical_fallacies	41.2 _{0.0}	35.3 _{-17.6}	35.3 _{-17.6}	23.5 _{-11.8}	41.2 _{-5.9}	35.3 _{-10.6}
machine_learning	41.7 _{0.0}	25.0 _{-33.3}	41.7 _{-8.3}	25.0 _{-25.0}	41.7 _{-8.3}	35.0 _{-15.0}
management	45.5 _{9.1}	63.6 _{27.3}	36.4 _{0.0}	45.5 _{0.0}	45.5 _{0.0}	47.3 _{7.3}
marketing	58.3 _{-16.7}	58.3 _{-12.5}	50.0 _{-12.5}	66.7 _{8.3}	58.3 _{0.0}	58.3 _{-6.7}
medical_genetics	80.0 _{0.0}	70.0 _{10.0}	90.0 _{30.0}	70.0 _{0.0}	90.0 _{20.0}	80.0 _{12.0}
miscellaneous	48.1 _{-7.6}	77.2 _{24.1}	46.8 _{-7.6}	54.4 _{-5.1}	57.0 _{-3.8}	56.7 _{0.0}
moral_disputes	34.3 _{0.0}	28.6 _{-5.7}	34.3 _{-2.9}	31.4 _{0.0}	42.9 _{5.7}	34.3 _{-0.6}
moral_scenarios	60.0 _{36.7}	23.3 _{0.0}	32.2 _{8.9}	34.4 _{11.1}	28.9 _{5.6}	35.8 _{12.4}
nutrition	22.6 _{6.5}	22.6 _{3.2}	22.6 _{6.5}	22.6 _{0.0}	22.6 _{0.0}	22.6 _{3.2}
philosophy	31.2 _{15.6}	25.0 _{0.0}	25.0 _{6.2}	31.2 _{3.1}	40.6 _{18.7}	30.6 _{8.8}
prehistory	45.5 _{0.0}	54.5 _{0.0}	54.5 _{3.0}	54.5 _{0.0}	54.5 _{3.0}	52.7 _{1.2}
professional_accounting	20.7 _{-6.9}	37.9 _{0.0}	34.5 _{6.9}	20.7 _{-10.3}	31.0 _{-17.2}	29.0 _{-5.5}
professional_law	22.7 _{-0.7}	29.2 _{3.2}	25.3 _{1.9}	23.4 _{-2.6}	26.0 _{1.3}	25.3 _{0.6}
professional_medicine	39.3 _{3.6}	42.9 _{7.2}	32.1 _{-3.6}	32.1 _{3.6}	35.7 _{-3.6}	36.4 _{1.4}
professional_psychology	45.2 _{1.6}	48.4 _{6.5}	46.8 _{9.7}	50.0 _{9.7}	51.6 _{3.2}	48.4 _{6.1}
public_relations	81.8 _{27.3}	72.7 _{9.1}	81.8 _{9.1}	81.8 _{9.1}	81.8 _{18.2}	80.0 _{14.5}
security_studies	32.0 _{8.0}	28.0 _{-4.0}	36.0 _{8.0}	36.0 _{0.0}	36.0 _{0.0}	33.6 _{2.4}
sociology	28.6 _{4.8}	47.6 _{23.8}	33.3 _{4.8}	33.3 _{4.8}	23.8 _{-9.5}	33.3 _{5.7}
us_foreign_policy	40.0 _{30.0}	20.0 _{10.0}	20.0 _{10.0}	30.0 _{10.0}	40.0 _{10.0}	30.0 _{14.0}
virology	29.4 _{0.0}	23.5 _{0.0}	17.6 _{-5.9}	41.2 _{0.0}	29.4 _{0.0}	28.2 _{-1.2}
world_religions	44.4 _{0.0}	44.4 _{5.5}	50.0 _{0.0}	55.6 _{0.0}	55.6 _{-27.8}	50.0 _{-4.4}
all	40.6 _{5.8}	44.2 _{6.9}	38.2 _{2.0}	40.4 _{2.6}	41.4 _{1.0}	41.0 _{3.7}

Table 15: The results of MMLU subtasks.

Question	James runs 12 miles a day for 5 days a week. If he runs 10 miles an hour how many hours does he run a week?
ALW (✓)	James runs 12 miles a day for 5 days a week. That means he runs $12 * 5 = 60$ miles a week. If he runs 10 miles an hour, it takes him 6 hours to run 60 miles. The answer is 6 hours.
Baseline (×)	James runs 12 miles a day for 5 days a week. 12 miles is $12 * 5 = 60$ miles. 10 miles an hour is $10 * 60 = 600$ miles. So he runs 600 miles a week. $600 / 60 = 10$ hours. The answer is 10 hours.

Table 16: Qualitative Study for LLaMA-7B with ALW and baselines on Gsm8k.

Question	If your skin was turning the color of a zombie, could it be because of nickel?
ALW (✓)	Nickel is a metal that causes allergic reactions in some people. A common symptom of nickel allergy is a greenish discoloration of the skin. Zombies are often depicted with green skin. Thus, if your skin was turning the color of a zombie, it could be because of nickel. So the answer is yes.
Baseline (×)	Nickel is a metal. Metals are not known to cause skin to turn green. Thus, your skin turning green is not likely to be caused by nickel. So the answer is no.

Table 17: Qualitative Study for LLaMA-30B with ALW and baselines on StrategyQA.

Question	Context: All professors are teachers. All researchers are professors. No teachers are students. All people who study are students. Leon is not studying or a researcher. If Leon is not studying, then Leon is a teacher. Question: The statement "Leon is neither a student nor a researcher." is True, False or Unknown?
ALW (✓)	$\forall x (\text{Professor}(x) \rightarrow \text{Teacher}(x)), \forall x (\text{Researcher}(x) \rightarrow \text{Professor}(x)), \forall x (\text{Teacher}(x) \rightarrow \text{Not}(\text{Student}(x))), \neg \text{Studying}(\text{leon}) \rightarrow \text{Teacher}(\text{leon}), \neg \text{Researcher}(\text{leon}) \rightarrow \text{Teacher}(\text{leon}), \text{Teacher}(\text{leon}) \rightarrow \text{Not}(\text{Student}(\text{leon}))$. The statement is True.
Baseline (×)	$\forall x (\text{Professor}(x) \rightarrow \text{Teacher}(x)), \forall x (\text{Researcher}(x) \rightarrow \text{Professor}(x)), \forall x (\text{Student}(x) \rightarrow \text{Teacher}(x)), \forall x (\text{Teacher}(x) \rightarrow \text{Not}(\text{Student}(x))), \neg \text{Student}(\text{leon}) \rightarrow \neg \text{Teacher}(\text{leon}), \neg \text{Researcher}(\text{leon}) \rightarrow \text{Teacher}(\text{leon})$. The statement is Unknown.

Table 18: Qualitative Study for LLaMA-13B with ALW and baselines on Folio.

Question	It takes Matthew 3 minutes to dig a small hole for shrubs and 10 minutes to dig a large hole for trees. How many hours will it take him to dig 30 small holes and 15 large holes?
ALW (✓)	Matthew can dig 30 small holes in 3 minutes. That means he can dig $30 * 3 = 90$ small holes in 1 hour. He can also dig 15 large holes in 10 minutes. That means he can dig $15 * 10 = 150$ large holes in 1 hour. So he can dig $90 + 150 = 240$ holes in 1 hour. $240 / 60 = 4$ hours. The answer is 4.
Baseline (×)	Matthew will dig 30 small holes in 30 minutes. 10 minutes to dig a large hole. 30 small holes + 10 large holes = 40 holes. 40 holes in 30 minutes + 10 holes in 10 minutes = 50 minutes. 50 minutes / 60 minutes = 0.833. 0.833 hours. The answer is 0.833 hours.
Question	Context: Some basketball players are not American. All basketball players are tall. If someone is tall, they wear large shoes. If someone wears large shoes, they wear large-sized clothes. Yuri does not wear large-size clothes. Question: The statement "Yuri is an American basketball player." is True, False or Unknown?
ALW (✓)	$\forall x (\text{BasketballPlayer}(x) \rightarrow \text{American}(x)), \forall x (\text{BasketballPlayer}(x) \rightarrow \text{Tall}(x)), \forall x (\text{Tall}(x) \rightarrow \text{WearLargeShoes}(x)), \forall x (\text{WearLargeShoes}(x) \rightarrow \text{WearLargeSizeClothes}(x)), \neg \text{WearLargeSizeClothes}(\text{yuri})$. The statement is False.
Baseline (×)	$\forall x (\text{BasketballPlayer}(x) \rightarrow \text{American}(x)), \forall x (\text{Tall}(x) \rightarrow \text{BasketballPlayer}(x)), \forall x (\text{LargeShoes}(x) \rightarrow \text{LargeClothes}(x)), \forall x (\text{LargeClothes}(x) \rightarrow \text{BasketballPlayer}(x)), \neg \text{LargeClothes}(\text{yuri}) \rightarrow \text{Yuri}$. The statement is True.

Table 19: Discussion on cases where the reasoning path is incorrect but the answer is correct.