Improving Reasoning Capabilities in Small Models through Mixture-of-Layers Distillation with Stepwise Attention on Key Information

Yao Chen[1,2], **Jiawei Sheng**[1], **Wenyuan Zhang**[1,2], **Tingwen Liu**[1,2*]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
{chenyao2023, shengjiawei, zhangwenyuan, liutingwen}@iie.ac.cn

Abstract

The significant computational demands of large language models have increased interest in distilling reasoning abilities into smaller models via Chain-of-Thought (CoT) distillation. Current CoT distillation methods mainly focus on transferring teacher-generated rationales for complex reasoning to student models. However, they do not adequately explore teachers' dynamic attention toward critical information during reasoning. We find that language models exhibit progressive attention shifts towards key information during reasoning, which implies essential clues for drawing conclusions. Building on this observation and analysis, we introduce a novel CoT distillation framework that transfers the teacher's stepwise attention on key information to the student model. This establishes structured guidance for the student's progressive concentration on key information during reasoning. More importantly, we develop a Mixture of Layers module enabling dynamic alignment that adapts to different layers between the teacher and student. Our method achieves consistent performance improvements across multiple mathematical and commonsense reasoning datasets. To our knowledge, it is the first method to leverage stepwise attention within CoT distillation to improve small model reasoning.

1 Introduction

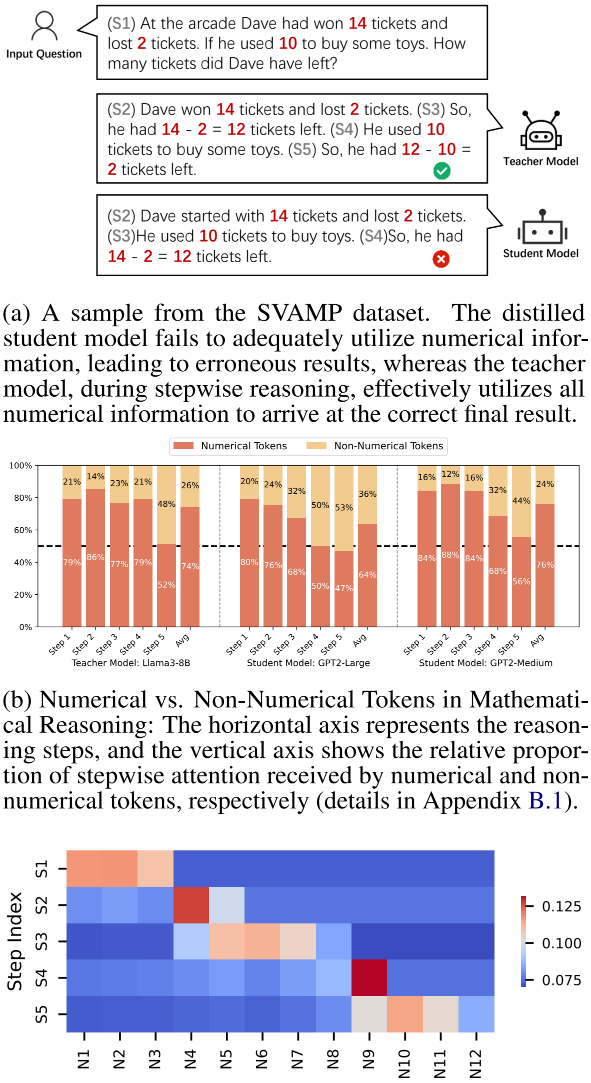
The ability of complex reasoning is a cornerstone of human intelligence, playing a crucial role in problem-solving, decision-making, and world understanding (Cobbe et al., 2021; Chu et al., 2024; Plaat et al., 2024). Recent advances have shown substantial improvements in the few-shot reasoning abilities of large language models. However, the immense scale of these models demands enormous memory and computational resources, making them prohibitively expensive to deploy on edge devices and impeding applications (Liu et al., 2024; Hu et al., 2024). To address this challenge, CoT distillation (Ho et al., 2023; Fu et al., 2023; Li et al., 2023; Hsieh et al., 2023) has emerged as a promising approach. In complex reasoning, CoT distillation methods typically transfer the step-by-step rationales generated by the teacher model to the student model, serving as an effective means of knowledge distillation.

Existing CoT distillation methods typically treat all tokens equally, often neglecting critical information for complex reasoning. We observe that the student models distilled via existing methods struggle to fully utilize key information across multi-step reasoning (Figure 1a). Notably, language models allocate more average attention to critical tokens during reasoning, implicitly encoding key clues for stepwise reasoning. For example, numerical tokens are intuitively crucial for mathematical reasoning, and our analysis results indicate that they indeed receive significantly more attention than non-numerical tokens during this process in both teacher and student models (Figure 1b). More importantly, we explore how the teacher model's attention to these critical tokens evolves during stepwise reasoning, and find that the attention distribution exhibits stepwise changes, with higher attention scores assigned to the critical tokens relevant to each reasoning step (Figure 1c & Figure 2). This highlights the teacher model's ability to progressively capture key information during reasoning. However, current CoT distillation methods directly provide the rationales generated by the teacher model to the student. This approach fails to fully exploit the aforementioned phenomena, leading to a failure in improving the student's ability to progressively capture and utilize key information.

Building on the above insights, we introduce **MoLSAKI**, a novel CoT distillation framework that captures and transfers the teacher model's <u>S</u>tepwise <u>A</u>ttention on <u>K</u>ey <u>I</u>nformation to enhance

[*]indicates corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 4953–4972
November 4-9, 2025 ©2025 Association for Computational Linguistics



(a) A sample from the SVAMP dataset. The distilled student model fails to adequately utilize numerical information, leading to erroneous results, whereas the teacher model, during stepwise reasoning, effectively utilizes all numerical information to arrive at the correct final result.

(b) Numerical vs. Non-Numerical Tokens in Mathematical Reasoning: The horizontal axis represents the reasoning steps, and the vertical axis shows the relative proportion of stepwise attention received by numerical and non-numerical tokens, respectively (details in Appendix B.1).

(c) Visualization of stepwise attention on numerical tokens from the 13th layer of the teacher model Llama3-8B for the sample in Figure 1a (details in Appendix D). The horizontal axis represents the indices of numerical tokens (the tokens highlighted in **red** in sample Figure 1a), and the vertical axis represents the indices of steps (the **grey Sx** labels in Figure 1a).

Figure 1: Stepwise attention on critical tokens implicitly encodes reasoning clues: A comprehensive analysis.

the student model's reasoning capabilities via a **M**ixture-**o**f-**L**ayers alignment strategy. Specifically, we define stepwise attention on critical tokens as the attention weights assigned to each critical token at each reasoning step. By concatenating these per-step distributions, we capture the model's evolving focus on key information throughout the entire reasoning process. Building on this concept, we then extract these stepwise attention maps from every layer of both the teacher and student models during the CoT distillation. For layer mapping in distillation, we design Mixture-of-Layers (MoL), drawing inspiration from Mixture-of-Experts (MoE) (Zhou et al., 2022; Jin et al., 2024). MoL facilitates adaptive weighted alignment between teacher and student layers, thereby overcoming the distillation challenge of mismatched layer counts. In summary, our contributions are as follows:

- We introduce a new perspective: during the reasoning process, large language models exhibit a progressive attention pattern towards certain critical tokens, a pattern that implicitly encodes valuable clues for stepwise reasoning.
- We propose a novel chain-of-thought distillation framework, MoLSAKI, which introduces the concept of stepwise attention on critical tokens and transfers the teacher model's progressive, dynamic focus on key information to the student model, thereby enhancing its capacity for effective reasoning.
- We design MoL to adaptively align layers between teacher and student models of different depths in a weighted and dynamic manner, thereby successfully overcoming the challenge of their mismatched layer counts.
- Our method yields performance gains in in-domain and out-of-domain settings across varying teacher-student model scales on mathematical and commonsense reasoning benchmarks.

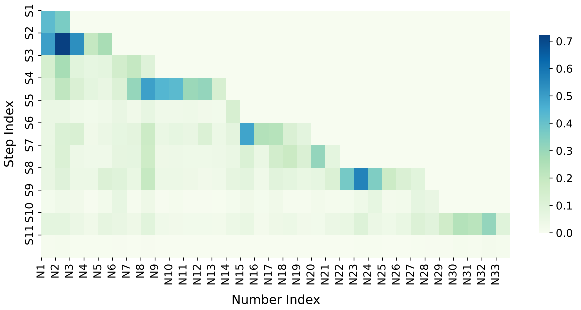
2 Related Work

2.1 Chain-of-Thought Distillation

Large language models (LLMs) demonstrate strong reasoning capabilities (Kojima et al., 2022; Wei et al., 2022), yet their massive scale hinders practical deployment. Recent work distills reasoning abilities into smaller models through CoT knowledge transfer (Ho et al., 2023; Hsieh et al., 2023; Fu et al., 2023; Li et al., 2023). Key approaches include Fine-tune-CoT's zero-shot rationale extraction (Ho et al., 2023) and DSS's multi-task separation of reasoning/answer prediction (Hsieh et al., 2023). Subsequent improvements introduce mutual information maximization (MMIloss (Chen et al., 2024)) and auxiliary model-based distillation (Mentor-KD (Lee et al., 2024)) (details in Appendix A.1). Existing methods neglect key information in reasoning and face structural constraints from logit distillation requirements (Lee et al., 2024; Zhang et al., 2024b). Our approach introduces stepwise attention on critical tokens distillation without requiring tokenizer alignment or projection layers.

Question: (S1)Where is a business restaurant likely to be located? (S2)Answer choices: (A) town, (B) at hotel, (C) mall, (D) business sector, (E) yellow pages.
Rationale: (S3)The question asks about the likely location of a business restaurant. (S4)A business restaurant is typically a place where business meetings or professional lunches take place. (S5)(A) town too broad, does not specify a business context. (S6)(B) at hotel hotels often have restaurants suitable for business meetings. (S7)(C) mall while malls have restaurants, they are more casual and not specifically for business. (S8)(D) business sector indicates a location focused on businesses, making it ideal for a business restaurant. (S9)(E) yellow pages a directory, not a physical location. (S10)Given the context of a business restaurant, both (B) and (D) are strong contenders, but (D) business sector is more specific to the business context. (S11)So, the answer is D.
Keywords: ["business", "restaurant", "location", "hotel", "sector", "professional", "meetings"]

(a) A sample from the CommonSenseQA dataset.



(b) Visualization of stepwise attention on critical tokens from the 32nd layer of the teacher model Qwen2.5-32B for the sample in Figure 2a. The horizontal axis represents the indices of critical tokens, and the vertical axis represents the indices of steps.

Figure 2: Progressive attention pattern on critical tokens (details in Appendix D).

2.2 Self-Attention Distillation

Prior methods transfer self-attention patterns via layer mapping: TinyBERT (Jiao et al., 2020) uses uniform mapping, MOBILEBERT (Sun et al., 2020) assumes identical layer counts, and MINILM (Wang et al., 2020) distills only final layers (details in Appendix A.2). These methods require matched attention dimensions and fixed layer correspondences. We overcome these limitations by 1) focusing distillation on critical tokens in reasoning steps instead of full attention matrices, and 2) using dynamic layer routing via MoL modules to automatically select optimal teacher-student layer pairs, outperforming rigid mapping approaches.

3 Methodology

MoLSAKI introduces a novel knowledge distillation framework that enhances the reasoning of the student model through synergistic integration of CoT distillation and stepwise attention guidance. Specifically, we first prepare CoT data annotated by the teacher model and conduct CoT distillation (§3.1), subsequently extract stepwise attention on critical tokens from the teacher and student models in the process of CoT distillation (§3.2), and finally implement adaptive MoL layer alignment (§3.3).

3.1 CoT Distillation

We obtain CoT data for each question-answer pair $\{q, \hat{a}\}$ in a raw dataset $\mathcal{D}$ by few-shot prompting the teacher model (details in Appendix F.3). The teacher's response to each question q is divided into two components: rationale r and answer a (see the sample in Figure 3). The labeled dataset $\{q, r, a \mid q \in \mathcal{D},\ a = \hat{a}\}$ will be used for the subsequent CoT distillation of the student model.

Following Hsieh et al. (2023), we perform CoT distillation comprising two tasks (*CoT Distillation* module in Figure 3): 1) final answer prediction a given a question q and 2) rationale r generation for the same input q. The respective loss functions are as follows:

$$\begin{aligned} \mathcal{L}_{\text{pre}} &= \mathbb{E}_{q \in \mathcal{D}} \left[\mathcal{L}_{\text{ce}}(f(q), a)\right], \\ \mathcal{L}_{\text{exp}} &= \mathbb{E}_{q \in \mathcal{D}} \left[\mathcal{L}_{ce}(f(q), r)\right], \end{aligned} \tag{1}$$

where f denotes the student model and $\mathcal{L}_{\text{ce}}$ denotes the cross-entropy loss between model predictions and target tokens.

3.2 Stepwise Attention on Critical Tokens

Believing that distilling the teacher's stepwise attention on critical tokens during reasoning is more impactful than simply transferring rationales, we introduce the loss L_{att} (in Eq.(6)) of stepwise attention on critical tokens during CoT distillation to guide the student's progressive focus on key information.

To compute the loss L_{att}, we first extract stepwise attention on critical tokens from both the teacher and student models (*Extract Stepwise Attention on Critical Tokens* module in Figure 3). In our design, *Stepwise* denotes reasoning steps incorporating the question. As shown in the example in Figure 3, we segment the input sequence composed of question and rationale into reasoning steps based on periods, resulting in 5 steps.

The teacher model's tokenizer converts the input sequence composed of question and rationale into a token sequence $\{x_1^t, x_2^t, ..., x_M^t\}$. $\mathcal{M}_1$ denotes the index set of all tokens partitioned by reasoning steps. Its element specifically denotes the index set of all tokens within a single reasoning step. Utilizing regular expression matching and the tokenizer's mapping, we obtain the index set of critical tokens from the token sequence, denoted as $\mathcal{M}_2$. Its element denotes the index set of critical tokens corresponding to a specific critical word in the original text after tokenization (details in Appendix C.1).

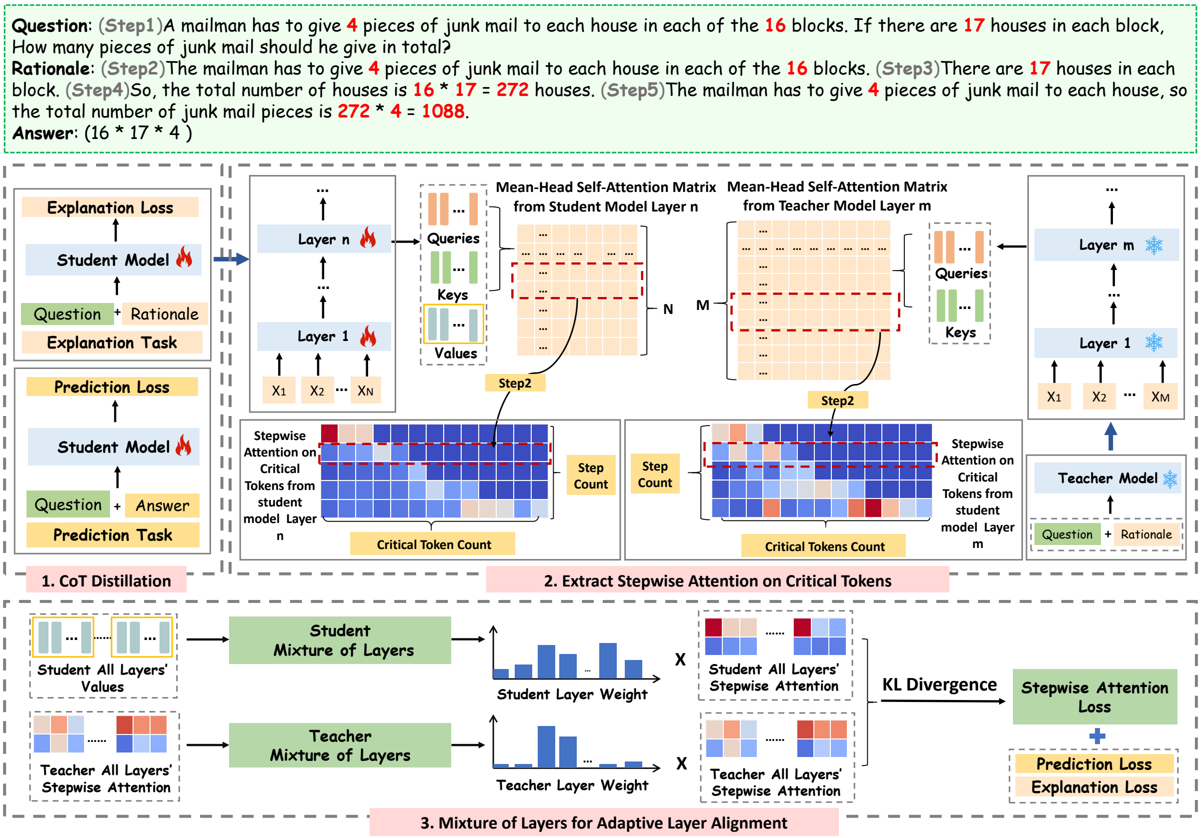

Question: (Step1)A mailman has to give 4 pieces of junk mail to each house in each of the 16 blocks. If there are 17 houses in each block, How many pieces of junk mail should he give in total?
Rationale: (Step2)The mailman has to give 4 pieces of junk mail to each house in each of the 16 blocks. (Step3)There are 17 houses in each block. (Step4)So, the total number of houses is 16 * 17 = 272 houses. (Step5)The mailman has to give 4 pieces of junk mail to each house, so the total number of junk mail pieces is 272 * 4 = 1088.
Answer: (16 * 17 * 4)
Explanation Loss
Student Model
Question + Rationale
Explanation Task
Prediction Loss
Student Model
Question + Answer
Prediction Task
1. CoT Distillation
Layer n
Layer 1
X1 X2 ... XN
Queries
Keys
Values
Mean-Head Self-Attention Matrix from Student Model Layer n
Mean-Head Self-Attention Matrix from Teacher Model Layer m
N
M
Queries
Keys
Layer m
Layer 1
X1 X2 ... XM
Teacher Model
Question + Rationale
Step2
Step2
Stepwise Attention on Critical Tokens from student model Layer n
Step Count
Critical Token Count
Step Count
Stepwise Attention on Critical Tokens from student model Layer m
Critical Tokens Count
2. Extract Stepwise Attention on Critical Tokens
Student All Layers' Values
Student Mixture of Layers
Student Layer Weight
X
Student All Layers' Stepwise Attention
Teacher All Layers' Stepwise Attention
Teacher Mixture of Layers
Teacher Layer Weight
X
Teacher All Layers' Stepwise Attention
KL Divergence
Stepwise Attention Loss
+
Prediction Loss
Explanation Loss
3. Mixture of Layers for Adaptive Layer Alignment

Figure 3: The MoLSAKI framework consists of three components. In the example, the question and rationale have 13 numerical tokens and 5 steps in total. Thus, the stepwise attention on numerical tokens in both teacher and student models is 5×13.

The l-th layer of the teacher model subsequently constructs the self-attention matrix $I_l^t \in \mathbb{R}^{M\times M}$. To compute stepwise attention on critical tokens, we first extract columns from I_l^t at the indices of critical tokens, where each column represents attention distribution from all tokens to a specific critical token. Based on this, we compute the aggregated stepwise attention on critical tokens by summing rows of the corresponding columns in I_l^t at each reasoning step, as follows:

$$\begin{aligned} \mathcal{M}_1 &:= \{\{0,1,2,...\},...,\{...,M-1\}\},\\ \mathcal{M}_2 &:= \{\{\mu_1,\mu_2,...\},...\},\ \mu\in\mathbb{N},\ \mu<M,\\ A_l^t &= \sum_{i\in\mathcal{K},\ j\in\mathcal{P}} I_l^t[i,j]\ (\mathcal{K}\in\mathcal{M}_1,\mathcal{P}\in\mathcal{M}_2), \end{aligned} \tag{2}$$

where $A_l^t \in \mathbb{R}^{|\mathcal{M}_1|\times|\mathcal{M}_2|}$ corresponds to the stepwise attention scores on critical tokens generated by the l-th layer of the teacher model.

The student model processes the token sequence $\{x_1^s, x_2^s, ..., x_N^s\}$ to generate self-attention matrix $I_l^s \in \mathbb{R}^{N\times N}$ of the l-th layer, from which we apply the identical extraction and aggregation mechanism to compute its stepwise attention on critical tokens:

$$\begin{aligned} \mathcal{N}_1 &:= \{\{0,1,2,...\},...,\{...,N-1\}\},\\ \mathcal{N}_2 &:= \{\{\lambda_1,\lambda_2,...\},...\},\ \lambda\in\mathbb{N},\ \lambda<N,\\ A_l^s &= \sum_{i\in\mathcal{K}\ j\in\mathcal{O}} I_l^s[i,j]\ (\mathcal{K}\in\mathcal{N}_1,\ \mathcal{O}\in\mathcal{N}_2), \end{aligned} \tag{3}$$

where $A_l^s \in \mathbb{R}^{|\mathcal{N}_1|\times|\mathcal{N}_2|}$ denotes the stepwise attention scores on critical tokens generated by the l-th layer of the student model with $|\mathcal{N}_1| = |\mathcal{M}_1|$ indicating the total count of reasoning steps and $|\mathcal{N}_2| = |\mathcal{M}_2|$ representing the total count of critical tokens (details in Appendix C.2). Our mechanism achieves functional compatibility between architecturally distinct models by aligning the stepwise attention dimensions on critical tokens across teacher and student models, thereby eliminating the requirement for shared tokenizers or vocabularies.

3.3 MoL for Adaptive Layer Alignment

Though distilling stepwise attention on critical tokens is feasible, determining optimal layer mapping in distillation presents a non-trivial challenge. This arises from architectural disparities between teacher and student models, which preclude complete layer-to-layer correspondence. Conventional rigid Single-Layer (SL) alignment approaches prove suboptimal due to their inflexibility. To overcome these limitations, we propose a Mixture-of-Layers (MoL) module that dynamically aggregates stepwise attention across all layers through trainable weighting parameters in the layer router.

Leveraging insights from the analysis of teacher and student model characteristics, distinct inputs are provided to the teacher and student MoL modules (*Mixture of Layers for Adaptive Layer Alignment* module in Figure 3). For the teacher model, the stepwise attention on critical tokens A_l^t varies across its different layers (Figure 13). By analysing the column gradients of A_l^t, we find that the most significant variation of stepwise attention on critical tokens occurs in the intermediate layers (Figure 4).

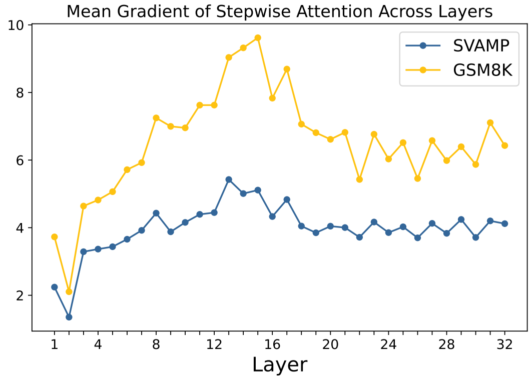


Figure 4: We analyse the average column gradient distribution of stepwise attention on critical tokens across the layers of Llama3-8B (details in Appendix E.1).

To effectively transfer the significant attention dynamics to the student model, we determine the teacher model's layer weights through temperature-controlled softmax normalisation applied to the gradients of A_l^t, as follows:

$$\begin{aligned} \mathrm{G}(A_l^t) &= \frac{\sum_{i=1}^{|\mathcal{M}_1|}\sum_{j=1}^{|\mathcal{M}_2|-1} \left|A_l^t[i,j+1]-A_l^t[i,j]\right|}{|\mathcal{M}_1|(|\mathcal{M}_2|-1)}, \\ p^t &= \mathrm{softmax}([\mathrm{G}(A_1^t),..,\mathrm{G}(A_{L_1}^t)],\tau_1) \in \mathbb{R}^{L_1}, \\ \mathrm{softmax}(z,\tau)_i &= \frac{e^{z_i/\tau}}{\sum_{j=1}^{K} e^{z_j/\tau}}, \end{aligned} \tag{4}$$

where $\mathrm{G}(A_l^t)$ denotes the mean gradient of A_l^t, L_1 indicates the count of layers in the teacher model, the $p^t \in \mathbb{R}^{L_1}$ denotes the layer weights obtained by the MoL of the teacher model, and τ_1 denotes the temperature parameter of the softmax function.

For the student model, we process value vectors from all layers through a learnable routing mechanism: First, we apply RMSNorm (Zhang and Sennrich, 2019) to stabilise the features. Second, we sum over the sequence dimensions to obtain compact layer embeddings. Third, we concatenate multi-layer representations. Finally, we generate adaptive layer weights via an affine transformation and a temperature-controlled softmax. The above procedure is formulated as:

$$\begin{aligned} \widetilde{V}_l &= \mathrm{RMSNorm}(V_l) \in \mathbb{R}^{N\times d}, \\ h_l &= \sum_{i=1}^{N}(\widetilde{V}_l[i,:]) \in \mathbb{R}^{d}, \\ H &= \mathrm{concat}(h_1,h_2,...,h_{L_2}) \in \mathbb{R}^{L_2\times d}, \\ p^s &= \mathrm{softmax}(HW+b,\tau_2) \in \mathbb{R}^{L_2}, \end{aligned} \tag{5}$$

where d denotes the dimension of the value vector, L_2 indicates the count of layers in the student model, the $p^s \in \mathbb{R}^{L_2}$ denotes the layer weights obtained by the MoL of the student model, and τ_2 denotes the temperature parameter of the softmax function. Building upon this foundation, we independently applied weighting to A_l^t and A_l^s respectively, followed by performing softmax normalisation along the temporal step dimension. Subsequently, the averaged Kullback-Leibler (KL) divergence across corresponding steps is calculated and designated as stepwise attention loss $\mathcal{L}_{att}$:

$$\begin{aligned} A^t &= \sum_{l=1}^{L_1}(p_l^t A_l^t) \in \mathbb{R}^{|\mathcal{N}_1|\times|\mathcal{N}_2|}, \\ \widetilde{A}^t[i,:] &= \mathrm{softmax}(A^t[i,:]), \\ A^s &= \sum_{l=1}^{L_2}(p_l^s A_l^s) \in \mathbb{R}^{|\mathcal{N}_1|\times|\mathcal{N}_2|}, \\ \widetilde{A}^s[i,:] &= \mathrm{softmax}(A^s[i,:]), \\ \mathcal{L}_{att} &= \frac{1}{|\mathcal{N}_1|}\sum_{i=1}^{|\mathcal{N}_1|}\mathrm{KL}(\widetilde{A}^t[i,:] \parallel \widetilde{A}^s[I,:]). \end{aligned} \tag{6}$$

Finally, we formulate the overall objective function $\mathcal{L}$ through a weighted combination as:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{pre}} + (1-\alpha)\mathcal{L}_{\text{exp}} + \beta\mathcal{L}_{\text{att}}. \tag{7}$$

where the prediction loss $\mathcal{L}_{\text{pre}}$ and the explanation loss $\mathcal{L}_{\text{exp}}$ are in Eq.(1), and $\mathcal{L}_{\text{att}}$ is the aforementioned stepwise attention loss.

4 Experiments

4.1 Setup

Datasets. In the experiment, five public reasoning datasets are utilized: SVAMP (Patel et al.,

	SVAMP	**SingleEq**	**AsDiv**	**GSM8K**	**CSQA**
In-Domain	✓	✗	✗	✗	✓
Teacher:Llama3-8B Student:GPT2-Large					
Vanilla Finetune	10	12.1	9.2	4.2	16.7
DSS	48.0	36.1	30.3	12.4	19.1
MMIloss	47.0	37.9	30.7	12.5	19.4
MoLSAKI(ours)	**49.5**	**39.8**	**32.2**	**15.1**	**21.0**
Teacher:Qwen2.5-32B Student:TinyLlama-1.1B					
Vanilla Finetune	14.5	21.4	14.3	6.7	17.8
DSS	59.5	48.1	33.5	13.8	28.9
MMIloss	64.5	48.1	42.6	14.0	25.8
MoLSAKI(ours)	**68.5**	**51.8**	**43.3**	**16.9**	**30.3**

Table 1: Accuracy(%) of different approaches.

2021), SingleEq (Koncel-Kedziorski et al., 2015), Asdiv (Miao et al., 2021), GSM8K (Cobbe et al., 2021), CommonSenseQA (CSQA) (Talmor et al., 2019). To assess the effect of our method on the generalization of the student model, we established an in-domain and out-of-domain evaluation setting using mathematical reasoning datasets. For mathematical reasoning, SVAMP is used as an in-domain test dataset, and SingleEq, Asdiv, and GSM8K serve as out-of-domain test datasets. For commonsense reasoning, CSQA is used as an in-domain test dataset. (details in Appendix F.1).

Baselines. We compare our proposed MoLSAKI framework with three established baseline methods: 1) *Vanilla Fine-Tuning* ($\mathcal{L} = \mathcal{L}_{\text{pre}}$) trains models exclusively on answer labels without CoT utilization; 2) *DSS* (Hsieh et al., 2023) ($\mathcal{L} = \alpha\mathcal{L}_{\text{pre}} + (1-\alpha)\mathcal{L}_{\text{exp}}$) conducts multi-task distillation that decouples rationale and answer optimisation; 3) *MMIloss* (Chen et al., 2024) ($\mathcal{L} = \alpha\mathcal{L}_{\text{pre}} + (1-\alpha)\mathcal{L}_{\text{exp}} + \beta\mathcal{L}_{\text{MMI}}$) extends DSS by incorporating cross-entropy loss between rationale generation and answer prediction as an auxiliary objective under the information bottleneck principle. In the experiments, we followed the default hyperparameter settings of these works, using $\alpha = 0.5, \beta = 0.1$.

Settings. In the main experiments, we employed two teacher-student model configurations: (1) Llama3-8B (Meta, 2024) as the teacher model and GPT-2 Large (774M) (Radford et al., 2019) as the student model; and (2) Qwen2.5-32B (Qwen et al., 2025) as the teacher model and TinyLlama-1.1B (Zhang et al., 2024a) as the student model. For the analysis experiment, the teacher model was Llama3-8B, and the student model was GPT-2 Medium (355M). For the main experiments, the weight hyperparameter β of the stepwise attention loss was set to 1.0, and the temperature hyperparameters for the MoL of the teacher and student models were set to $\tau_1 = 0.1$ and $\tau_2 = 0.5$, respectively (details in Appendix E).

4.2 Main Results

In this section, we thoroughly evaluate MoLSAKI through in-domain and out-of-domain tests. The results demonstrate that it works effectively and maintains consistent performance across different reasoning datasets, demonstrating its reliability.

1) *MoLSAKI substantially boosts student models' reasoning performance.* CoT distillation methods notably improve the performance of student models in reasoning tasks compared to the standard fine-tuning approach, as shown in Table 1. Our proposed MoLSAKI method achieves an average relative improvement of 7.5% (GPT2-Large) and 11.3% (TinyLlama) over the baselines. A detailed computational comparison of the methods and further case studies is provided in Appendix F.5 and Appendix F.6.

2) *MoLSAKI consistently achieves significant in-domain accuracy improvements across varying model scales.* Specifically, on the in-domain datasets, it outperforms both the DSS and MMIloss baselines for two distinct student model scales. This superior performance is attributed to MoLSAKI's ability to improve knowledge transfer by guiding the student's attention at each reasoning step through stepwise attention alignment.

3) *MoLSAKI demonstrates strong generalization capabilities on out-of-domain reasoning tasks.* Experimental results consistently show that our method outperforms baseline approaches across out-of-domain benchmarks for two different student models. This superior out-of-domain generalization underscores the importance of distilling the teacher model's stepwise attention focusing on critical tokens.

4.3 Hyperparameter Analysis

In this section, we systematically assess the impact of the weight hyperparameter β of stepwise attention loss $\mathcal{L}_{\text{att}}$, as well as the temperature parameters τ_1 and τ_2 modulating the layer weight of the teacher and student models respectively. We conducted experiments using the student model GPT2-Medium and the teacher model Llama3-8B on three mathematical reasoning datasets, where SVAMP serves

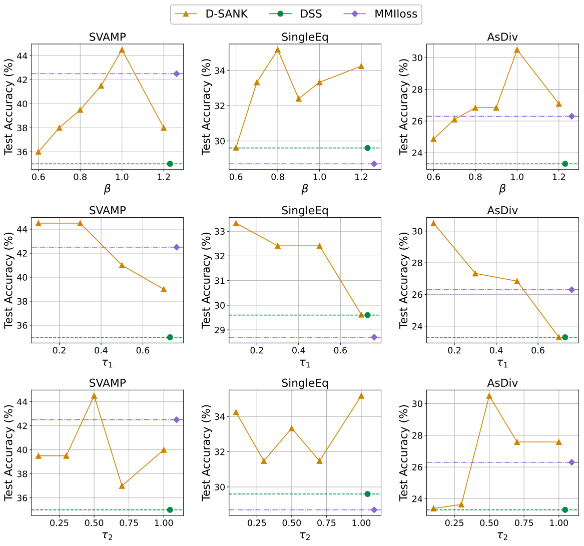


Figure 5: Hyperparameter analysis of β, τ_1, and τ_2 in MoLSAKI.

as the in-domain test dataset and SingleEq and AsDiv are out-of-domain test datasets.

Our β analysis reveals that MoLSAKI's SVAMP performance peaks when β is set to 1.0 before declining (Figure 5), surpassing MMIloss only at this optimal value. However, it maintains consistent advantages across out-of-domain datasets under most β settings, further demonstrating that MoLSAKI enhances the generalisation of the student model's mathematical reasoning ability.

For τ_1 of MoL in the teacher model, the performance of MoLSAKI gradually declines as τ_1 increases (Figure 5). This suggests that during the stepwise attention distillation process, the teacher model tends to prioritise layers with larger stepwise attention gradients, enabling these layers to contribute more prominently to the transfer of attention information in the distillation process. For τ_2 of MoL in the student model, the results on the SVAMP and AsDiv datasets reach their maximum values when τ_2 is set to 0.5 (Figure 5). Extreme τ_2 values degrade MoLSAKI performance, indicating that stepwise attention distillation necessitates balanced layer participation in the student model while preventing excessive uniformity. Temperature hyperparameters exhibit patterns comparable to β: while suboptimal configurations cause SVAMP performance to dip below MMIloss, most settings achieved superior generalisation.

4.4 Layer Weight Visualization

This section visualises layer weight distributions in MoLSAKI's MoL under three (τ_1, τ_2) configurations, comparing teacher (Llama3-8B) and student (GPT2-Medium) models in Figure 6.

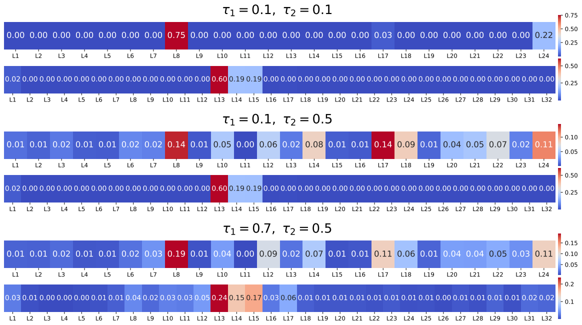


Figure 6: Comparative visualization of layer weight in Llama3-8B (32-layer) and GPT2-Medium (24-layer) under parameter configurations τ_1 and τ_2.

When setting temperature coefficients to $\tau_1 = 0.1$ and $\tau_2 = 0.5$, the teacher model exhibits significant layer-weight differentiation during stepwise attention distillation (details in Appendix E). Specifically, layers 13-15 demonstrate maximum weight values, while other layers show parameter attenuation approaching zero. In contrast, the student model maintains relatively balanced weight distribution throughout the distillation process: although layers 8, 17, and 24 attain comparatively higher weights, the remaining layers preserve non-negligible values, forming a distinct contrast with the near-zero weight pattern observed in most teacher model layers.

With temperature parameters $\tau_1 = 0.1$ and $\tau_2 = 0.1$, the student model demonstrates pronounced weight concentration during distillation: layer 8 emerges as the dominant contributor with maximum weight magnitude, followed distantly by layer 24, while all other layers' weights approach negligible values. In contrast, when configuring $\tau_1 = 0.7$ and $\tau_2 = 0.5$, the teacher model exhibits fundamental shifts in weight dynamics: the previously dominant layers 13-15 lose their absolute predominance, giving way to more balanced inter-layer weight allocation. Additionally, we visualise the layer weight for the teacher and student models under $\tau_1 = 0.1$ and $\tau_2 = 1.0$, with the results presented in Appendix F.4.

4.5 SL Alignment vs. MoL Alignment

To validate the effectiveness of the MoL module, we implement two fixed single-layer (SL) alignment strategies for comparison with our adaptive weighted layer alignment (MoL) method: 1) Based on the layer visualization results with $\tau_1 =$

Method	T → S	SVAMP	AsDiv	AVG
DSS	-	35.0	23.3	29.1
MMIloss	-	42.5	26.3	34.4
MoLSAKI	13 → 8	36.5	25.1	30.8
	13 → 17	38.5	26.6	32.5
	13 → 24	35.0	26.1	30.5
	14 → 8	36.0	26.6	31.3
	14 → 17	42.5	28.3	35.4
	14 → 24	43.0	28.1	35.5
	15 → 8	35.5	27.3	31.4
	15 → 17	36.5	26.6	31.5
	15 → 24	39.0	24.6	31.8
	32 → 24	43.0	28.1	35.8
	MoL	**44.5**	**30.5**	**37.5**

Table 2: Ablation comparison experiment results of adaptive weighted layer alignment (MoL) and fixed single-layer mapping (SL). Boldface denotes the best performance, while underlining denotes the second best.

0.1, $\tau_2 = 0.5$ (in Sec.4.4), we select the top three highest-weighted layers from both the teacher model Llama3-8B (layers 13, 14, 15) and the student model GPT2-Medium (layer 8, 17, 24). 2) Following MINILM (Wang et al., 2020), we separately select the last layer of the teacher model Llama3-8B (layers 32) and the student model GPT2-Medium (layer 24).

The experimental results demonstrate that MoL's adaptive weighted layer alignment mechanism outperforms conventional fixed single-layer alignment approaches (Table 2). Notably, even with the simplified single-layer alignment configuration, some of it still surpasses baseline methods. This finding suggests that even if only a specific layer of the student model learns to capture the teacher model's specific layer's attention on critical tokens at the step level, it still contributes to the student model's final reasoning.

4.6 Rationale and Stepwise Attention Derives from Different Models

To systematically evaluate the robustness of MoLSAKI across diverse teacher model configurations, we conduct comparative experiments under two distinct scenarios: 1) Unified Configuration employing Llama3-8B as the sole teacher for both rationale generation and Stepwise Attention on Numerical Tokens extraction, and 2) Hybrid Configuration combining PaLM-540B's rationale generation (following DSS) with Llama3-8B's numerical attention patterns.

Method	SVAMP	SingleEq	AsDiv	AVG
In-Domain	✓	✗	✗	-
Rationale ← Llama3-8B **Stepwise Attention ← Llama3-8B**				
DSS	48.0	36.1	30.3	38.1
MMIloss	47.0	37.9	30.7	38.5
MoLSAKI	**49.5**	**39.8**	**32.2**	**40.5**
Rationale ← PaLM-540B **Stepwise Attention ← Llama3-8B**				
DSS	43.0	33.3	33.0	36.4
MMIloss	42.0	29.6	**34.9**	35.5
MoLSAKI	**45.5**	**37.9**	34.5	**39.3**

Table 3: Performance comparison of MoLSAKI and baselines on reasoning datasets across different teacher model configurations.

Using GPT2-Large as the student model across both settings, our experimental results (Table 3) yield three principal observations: First, configuration analysis reveals that despite PaLM-540B's substantial parameter advantage (540B vs 8B), its inferior mathematical reasoning capability, as evidenced by official benchmark (Chowdhery et al., 2023; Grattafiori et al., 2024) comparisons, explains the performance gap between configurations. Second, MoLSAKI demonstrates consistent superiority over baseline methods in both configurations, achieving significant relative accuracy improvements of 5.1% (Unified) and 7.9% (Hybrid), thereby validating its teacher-agnostic knowledge integration capability. Third, the architecture's decoupled design enables practical deployment flexibility, allowing simultaneous utilization of black-box models (e.g., GPT-4.1, Gemini2.5) for high-quality rationale generation and white-box models (e.g., Llama3-8B) for numerical attention extraction - an innovative paradigm for heterogeneous knowledge distillation. These findings collectively substantiate MoLSAKI's effectiveness in cross-configuration applications while proposing a novel framework for optimally leveraging diverse model capabilities in knowledge transfer scenarios.

5 Conclusion

We contribute a new perspective to improving CoT distillation, positing that the stepwise attention on critical tokens implicitly encodes essential reasoning cues inherent in large models. Building upon this perspective, we propose MoLSAKI, a novel distillation framework aimed at resolving the issue of critical information underutilization in CoT distillation for reasoning. It facilitates the transfer of the teacher model's stepwise attention on critical tokens to the student model through a MoL strategy, which enables adaptive layer alignment.

Limitations

Limited computational resources constrained our exploration of diverse model sizes and architectures for both teachers and students. Nevertheless, we believe this work offers a valuable perspective on Chain-of-Thought distillation and large language model reasoning. Despite its current scope, this study establishes a foundation for future research to extend attention-based distillation across a wider range of model scales and architectures. In addition, our experiments focused on relatively simple reasoning tasks with short chains of thought. Future work should examine whether these findings generalize to more complex problems that demand longer and more intricate reasoning paths.

Ethics Statement

This research complies with established ethical standards. All datasets employed are publicly available and utilized solely for their intended research purposes. These datasets contain no personally identifiable information or sensitive content, thereby posing no risks to privacy or confidentiality. In addition, the study did not involve human subjects or annotators.

Acknowledgments

We thank all anonymous reviewers for their insightful comments and suggestions. This work was supported by the National Natural Science Foundation of China (Grant No. 62572465).

References

Federico Barbero, Alvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Michael Bronstein, Razvan Pascanu, et al. 2025. Why do llms attend to the first token? *arXiv preprint arXiv:2504.02732*.

Xin Chen, Hanxian Huang, Yanjun Gao, Yi Wang, Jishen Zhao, and Ke Ding. 2024. Learning to maximize mutual information for chain-of-thought distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6857–6868, Bangkok, Thailand. Association for Computational Linguistics.

Yilong Chen, Junyuan Shang, Zhenyu Zhang, Yanxi Xie, Jiawei Sheng, Tingwen Liu, Shuohuan Wang, Yu Sun, Hua Wu, and Haifeng Wang. 2025. Inner thinking transformer: Leveraging dynamic depth scaling to foster adaptive internal thinking. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28241–28259, Vienna, Austria. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. 2024. Navigate through enigmatic labyrinth a survey of chain of thought reasoning: Advances, frontiers and future. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1173–1203, Bangkok, Thailand. Association for Computational Linguistics.

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

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In *International Conference on Machine Learning*, pages 10421–10430. PMLR.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, Toronto, Canada. Association for Computational Linguistics.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174.

Peng Jin, Bo Zhu, Li Yuan, and Shuicheng Yan. 2024. Moh: Multi-head attention as mixture-of-head attention. *arXiv preprint arXiv:2410.11842*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597.

Hojae Lee, Junho Kim, and SangKeun Lee. 2024. Mentor-kd: Making small language models better multi-step reasoners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17643–17658.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2665–2679, Toronto, Canada. Association for Computational Linguistics.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, et al. 2025. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*.

Yantao Liu, Zhao Zhang, Zijun Yao, Shulin Cao, Lei Hou, and Juanzi Li. 2024. Aligning teacher with student preferences for tailored training data generation. *arXiv preprint arXiv:2406.19227*.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date. Blog. Accessed: 2024-01-24.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2021. A diverse corpus for evaluating and developing english math word problem solvers. *arXiv preprint arXiv:2106.15772*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Back. 2024. Reasoning with large language models, a survey. *arXiv preprint arXiv:2407.11511*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. In *OpenAI Blog*.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170.

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.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. 2024.

Knowledge circuits in pretrained transformers. *arXiv preprint arXiv:2405.17969*.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024a. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. 2024b. Dual-space knowledge distillation for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18164–18181, Miami, Florida, USA. Association for Computational Linguistics.

Wenyuan Zhang, Tianyun Liu, Mengxiao Song, Xiaodong Li, and Tingwen Liu. 2025a. SOTOPIA-Ω: Dynamic strategy injection learning and social instruction following evaluation for social agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24669–24697, Vienna, Austria. Association for Computational Linguistics.

Wenyuan Zhang, Shuaiyi Nie, Jiawei Sheng, Zefeng Zhang, Xinghua Zhang, Yongquan He, and Tingwen Liu. 2024c. Revealing and mitigating the challenge of detecting character knowledge errors in llm role-playing. *arXiv preprint arXiv:2409.11726*.

Wenyuan Zhang, Shuaiyi Nie, Xinghua Zhang, Zefeng Zhang, and Tingwen Liu. 2025b. S1-bench: A simple benchmark for evaluating system 1 thinking capability of large reasoning models. *arXiv preprint arXiv:2504.10368*.

Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Mingchuan Yang, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2024. Attention heads of large language models: A survey. *arXiv preprint arXiv:2409.03752*.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. 2022. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114.

A Related Work

A.1 Chain-of-Thought Distillation

In complex reasoning, CoT distillation methods typically transfer the step-by-step rationales generated by the teacher model to the student model to enhance its reasoning abilities (Ho et al., 2023; Fu et al., 2023; Li et al., 2023; Hsieh et al., 2023; Zhang et al., 2024c, 2025a). DSS (Hsieh et al., 2023) treats CoT distillation as a multitask learning problem, assigning two labels per query: the final answer and the rationale generated by the teacher model. Following this, several studies incorporate an auxiliary loss to further enhance the complex reasoning capabilities of small language models. Mentor-KD (Lee et al., 2024) introduces a mentor model situated between the student and teacher models with its logit output distribution that serves as an auxiliary soft label for distillation. MMIloss (Chen et al., 2024) introduces a maximum mutual information loss as an additional distilling objective, addressing DSS's oversight of the mutual information between the rationale and final answer.

However, existing CoT distillation primarily focuses on transferring the result of the teacher's reasoning (the rationales), rather than the process itself. We contend that the teacher's ability to progressively attend to critical tokens during reasoning is a more fundamental and valuable skill. Hence, our goal is to transfer this crucial progressive attention pattern to the student model. We achieve this by incorporating a stepwise attention on critical tokens distillation loss L_{att} (Eq.(7)), which encourages the student to learn this vital ability to focus on key information step-by-step.

A.2 Self-Attention Distillation

Existing self-attention distillation methods (Jiao et al., 2020; Sun et al., 2020; Wang et al., 2020) suffer from two main drawbacks. They are not designed for reasoning tasks, often neglecting reasoning-specific attention patterns and distilling the full self-attention matrix, which mandates identical teacher-student tokenizers. Moreover, they typically handle varying teacher-student layer counts with rigid single-layer alignment (SL), ignoring the functional diversity of layers (Geva et al., 2023; Yao et al., 2024). Conversely, we utilize a more flexible MoL layer alignment strategy, whose superiority over SL is demonstrated in our ablation studies (see in Sec.**4.5**).

B Critical Tokens in CoT

Chen et al. (2025) investigated critical tokens during the pre-training stage and allocated more computational resources to these tokens, whereas our work primarily focuses on critical tokens within the chain-of-thought (CoT) during the reasoning process. Effective reasoning relies on focusing attention on critical information, which provides essential clues for successful problem-solving, much like in human cognition. Driven by this understanding, we sought to analyse the attention distribution over critical tokens in LLMs' CoT. Although prior research (Xiao et al., 2023; Barbero et al., 2025) indicates that autoregressive LLMs often prioritize the initial token, our specific interest lies in the attention distribution across the remaining tokens within the reasoning steps. To highlight this, we developed methods tailored to mathematical and commonsense reasoning to identify these critical tokens and performed visualization analysis, omitting the initial token's attention.

B.1 Mathematical Reasoning

Recognizing the intuitive importance of numerical tokens in mathematical reasoning, we conducted a visualization analysis to confirm this. We conducted analysis experiments on the mathematical reasoning datasets GSM8K and SVAMP (100 randomly sampled instances from each, totaling 200 samples).

The student model was fine-tuned using the DSS (Hsieh et al., 2023) method on CoT data from GSM8K and SVAMP to endow it with mathematical reasoning capability. For each sample, given the question and the zero-shot prompt 'Let's think step by step,' the model was tasked with generating a reasoning process. Attention scores for numerical and non-numerical tokens were recorded during this process. To compute the step-by-step attention, we segmented the reasoning steps based on periods and applied an averaging strategy across layers and attention heads. Finally, we applied the softmax function to the average attention scores of numerical and non-numerical tokens, yielding the relative proportions of stepwise attention allocation (see Figure 1b).

The results demonstrate that both the teacher and student models allocate higher average attention weights to numerical tokens compared to non-numerical tokens in the question during mathematical reasoning. These findings collectively under-

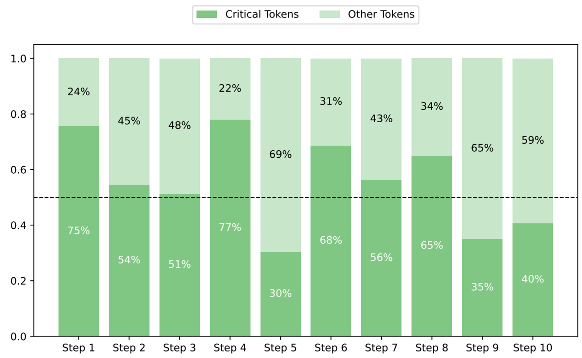


Figure 7: We randomly sampled 100 instances from CommonSenseQA to analyze the average attention allocated by Qwen2.5-32B to critical tokens corresponding to keywords relative to other tokens at each step.

score the critical role of numerical tokens in mathematical reasoning.

B.2 Commonsense Reasoning

For commonsense reasoning, we adopt a keyword extraction method. To obtain critical tokens using this method, we design a prompt during few-shot CoT generation by the teacher model, asking it to list 3-8 unique keywords deemed crucial for reasoning (in Figure 10). The visualization analysis method was the same as described in Appendix B.1. Our results indicate that, in most steps of the inference process, tokens corresponding to extracted keywords received significantly higher attention, highlighting their crucial role in facilitating inference. (see Figure 7).

C Extracting Stepwise Attention

C.1 Extracting Indices of Critical Tokens

Critical tokens' indices are automatically extracted from the token sequence, eliminating the need for manual annotation. We first identify critical words based on the task type: For commonsense reasoning, we prompt the teacher model during CoT generation to provide keywords it deemed important for reasoning. For mathematical reasoning, numerical words are considered critical based on prior analysis. This is achieved by first locating numerical words or teacher-provided keywords in the text sequence via regex matching.

Once these critical words are identified in the input text sequence, we automatically extract their corresponding indices ($\mathcal{M}_2$ in Eq.(2) & $\mathcal{N}_2$ in Eq.(3)) in the token sequence using regex matching and tokenizer mapping. It is noted that the elements within $\mathcal{M}_2$ and $\mathcal{N}_2$ are not simple integer indices, but rather are composed of smaller index sets. Because a single critical word sometimes corresponds to multiple tokens, we obtain a small index set $\mathcal{P} \in \mathcal{M}_2$ ($\mathcal{O} \in \mathcal{N}_2$) for each keyword. In our stepwise attention calculation, these critical tokens in $\mathcal{P}$ are treated as a whole, and the attention received by all member tokens is summed to represent the attention received by the critical word.

C.2 Identical Shape

Notably, during student model distillation, the input CoT text and the critical words within the CoT are all derived from the teacher model. Consequently, the identical input text and critical words used in CoT distillation (see Figure 3) ensure that the stepwise attention matrix shares the same shape ($|\mathcal{N}_1| = |\mathcal{M}_1|, |\mathcal{N}_2| = |\mathcal{M}_2|$) for teacher and student models, despite the significant difference between their tokenizers resulting in $\mathcal{N}_1 \neq \mathcal{M}_1$, $\mathcal{N}_2 \neq \mathcal{M}_2$.

D Progressive Attention Pattern

Analogous to human reasoning, where attention to different key information shifts dynamically as steps evolve, we observe a dynamic pattern in the teacher model's stepwise attention towards critical tokens. This attention pattern implicitly encodes the teacher model's capture and utilization of key information during the reasoning process. To illustrate this, we visualized the stepwise attention on critical tokens from a specific layer of the teacher model for a reasoning sample. The results revealed distinct step-by-step variations in the teacher model's attention to critical tokens during reasoning (see Figure 1c & Figure 2).

E Temperature Parameters

The temperature parameters τ_1, τ_2 control the sharpness of the adaptive layer weight distributions in the teacher and student models, respectively. The rationale for the MoL configuration and the specific values of hyperparameters $\tau_1 = 0.1$, $\tau_2 = 0.5$ is elaborated below.

E.1 Teacher: τ_1

We conduct a visual analysis of the stepwise attention of the teacher model from both qualitative and quantitative perspectives.

Qualitative: We visualized the stepwise attention on critical tokens for a selected sample across all 32 layers of the teacher Llama3-8B model. As shown in Figure 13, this attention exhibits clear variations across different layers.

Quantitative: We randomly selected 100 samples from each of the GSM8K and SVAMP datasets and input the corresponding questions and rationales into the teacher Llama3-8B. We then extracted stepwise attention on critical tokens from each layer and computed their column gradients. Importantly, these column gradients are distinct from backpropagation gradients. Calculated using Eq.(4), they evaluate the average magnitude of attention weight differences between adjacent critical tokens. The column gradients reflect the magnitude of stepwise change in attention on critical tokens within a given layer. The results highlight significant attention shifts in the intermediate layers (with the most notable changes occurring in layers 13–15 in Figure 4).

Through qualitative and quantitative analysis, we find that the most significant gradual change in attention to critical tokens takes place in the intermediate layers. This finding aligns with previous interpretability studies (Geva et al., 2023; Yao et al., 2024; Zheng et al., 2024), suggesting that the intermediate layers of large models are more strongly associated with reasoning than other layers. Aiming to ensure the student model prioritizes learning from these crucial intermediate layer patterns, we set τ_1 to an extremely small value, thereby allocating them greater weight.

E.2 Student: τ_2

Since the student model is small, we aim for all its layers to participate substantially yet non-uniformly in attention distillation. To achieve this, τ_2 is set to a moderate value, yielding a less peaked but non-uniform layer weight distribution.

F Details of Experiments

F.1 Datasets

The reasoning abilities of current large language models are generally categorized into two modes: System 1 thinking (Zhang et al., 2025b) and System 2 thinking (Li et al., 2025). In this work, we primarily focus on System 2 thinking. To comprehensively evaluate performance across varying difficulty levels, we conducted experiments on five benchmarks spanning commonsense and mathematical reasoning. Table 4 provides data statistics for these benchmarks.

The mathematical reasoning datasets, all human-authored, consist primarily of grade school math word problems. Among these, GSM8K represents a challenging problem domain. While Asdiv was originally a multiple-choice mathematical reasoning dataset, we modified it by removing the options and rephrasing the questions as open-ended. This change was implemented to enhance task difficulty and minimize potential interference from random guessing.

For the commonsense reasoning benchmarks, Commonsense QA assesses the ability to apply everyday knowledge and commonsense reasoning about the physical and social world to answer questions in practical scenarios.

Dataset	In-Domain	Train	Test
SVAMP	✓	800	200
SingleEq	✗	-	108
Asdiv	✗	-	406
GSM8K	✗	-	1318
CommonSenseQA	✓	9741	1221

Table 4: Dataset statistics used in our experiments.

F.2 Hyperparameter Settings

All experiments were performed on the NVIDIA A800 $\times 1$ GPU cloud environments. The GPT2-Medium and GPT2-Large models were trained with the following configurations: learning rate = 5×10^{-5}, batch size = 16, maximum training steps = 4,000. The TinyLllama model was trained with the following configurations: learning rate = 1×10^{-4}, batch size = 16, maximum training steps = 2,000. We report the average results over three random runs.

F.3 Prompts

To obtain more accurate CoT samples, we design a dual-phase CoT generation pipeline to handle complexity-stratified questions. In the first stage, we prompt the teacher model to generate r and a based on the question q, similar to previous works. This initial phase ensures accurate responses for most relatively simple questions. The second stage addresses incorrect samples with higher complexity levels by prompting the teacher model to regenerate rationale r and answer $\hat{a}$ under the guidance of both the question q and the ground-truth $\hat{a}$. This dual-phase pipeline enables scalable generation of high-quality CoT samples while maintaining rigorous quality control throughout the process (details in Figure 9a & Figure 9b).

F.4 Layer Weight Visualization

We visualize the layer weights for the teacher and student models under $\tau_1 = 0.1$ and $\tau_2 = 1.0$, as shown in Figure 8. Under this temperature parameter configuration, the weights of each layer in the student model are more evenly distributed.

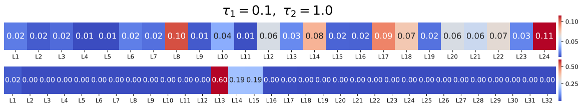


Figure 8: Layer weight visualization when $\tau_2 = 1.0,\ \tau_1 = 0.1$.

F.5 Computational Cost

Regarding the computational overhead of MoLSAKI, the attention matrix is directly utilized as an intermediate result from the standard forward pass, thus introducing no additional computation. The newly introduced MoL module contributes only a marginal computational cost, consisting of one linear layer and RMSNorm. Consequently, the total increase in FLOPs from integrating MoLSAKI is slight, yielding significant distillation performance gains. To quantify this, we compared the FLOPs of all evaluated methods on the two student models.

Student Model	Method	FLOPs ($\times 10^{11}$)
GPT2-Medium	Vanilla Finetune	0.947
	DSS	2.927
	MMIloss	2.9270010
	MoLSAKI	2.9270186
GPT2-Large	Vanilla Finetune	2.007
	DSS	6.081
	MMIloss	6.0810010
	MoLSAKI	6.0810273

Table 5: Computational cost (FLOPs).

F.6 Case Study

In this section, we select two samples each from SVAMP and GSM8K for case analysis (Figure 11). We compare the differences between the MoLSAKI method and the baseline method in terms of rationale generation. And to further compare their stepwise attention on numerical tokens, we visualize the stepwise attention of the student model GPT2-Medium (8th layer) distilled by different methods and the teacher model Llama3-8B (13th layer).

When reasoning the question presented in Figure 11b, the student model distilled by DSS mentions the condition "He used 10 tickets to buy toys" during the generation of rationales. However, it fails to utilize the number "10" in the subsequent reasoning process, leading to an incorrect result. When addressing the question in Figure 11d, MMIloss overlooks the condition "but he lost 2 of them" while generating rationales, which also results in an incorrect answer. In contrast, the MoLSAKI method makes full and effective use of all relevant numerical conditions.

By comparing the stepwise attention on numerical tokens during the generation of rationales for the questions in the above two examples between the teacher model and the student models distilled by different methods in Figure 12, it can be observed that, compared with the baseline methods, the student model distilled by D-SANK exhibits a high degree of similarity to the teacher model in terms of stepwise attention. This indicates that distilling the teacher model's stepwise attention on number tokens to the student model can enhance the student model's comprehensive attention and in-depth understanding of numerical conditions. Consequently, the mathematical reasoning ability of the student model is improved.

system content	Assume you are one of the greatest AI scientists, logicians, and mathematicians. Please answer the questions according to the following examples and requirement
user content	**[Examples]** Q: {Question} R: {Rationale} ## 1. Let's think through the problem step by step and provide the answer strictly in the R format as shown in the above example. 2. For percentages, to allow the eval() function to compute, express them as a division by 100. For example, “40%” should be written as (40 / 100). 3. Please ensure that the final answer ends with “The answer is (expression)”, where (expression) is enclosed in parentheses. 4. The (expression) should not contain any commas and should be the raw combined formula. Q: {Question}

(a) Generating CoT when given the question.

system content	Assume you are one of the greatest AI scientists, logicians, and mathematicians. Please answer the questions according to the following examples and requirement
user content	**[Examples]** Q: {Question} GT: The final answer to this question is {Ground Truth}. Based on this answer, please work through the problem step by step to deduce the question. R: {Rationale} ## 1. Let's think through the problem step by step and provide the answer strictly in the R format as shown in the above example. 2. For percentages, to allow the eval() function to compute, express them as a division by 100. For example, “40%” should be written as (40 / 100). 3. Please ensure that the final answer ends with “The answer is (expression)”, where (expression) is enclosed in parentheses. 4. The (expression) should not contain any commas and should be the raw combined formula. Q: {Question} GT: The final answer to this question is {Ground Truth}. Based on this answer, please work through the problem step by step to deduce the question.

(b) Generating CoT when given the question and ground truth.

Figure 9: Prompt template for generating CoT of the teacher model with dual-phase pipeline.

system content	Assume you are one of the greatest AI scientists, logicians, and mathematicians. Please answer the questions according to the following examples and requirement

user content	**[Examples]**
	Q: The sanctions against the school were a punishing blow, and they seemed to what the efforts the school had made to change? Answer choices: (A) ignore, (B) enforce, (C) authoritarian, (D) yell at, (E) avoid.
	T: Please work through the problem step by step to deduce the question. And give 3-8 keywords that you think are important in the process of reasoning. Each keyword must be an independent word. The answer is given in standard json string format.
	{
	"Rationale": "The question asks for a type of person who typically contracts illness—in other words, who is more likely to become ill. (A) hospital – This is a place, not a person. (B) head – This is a body part, not a person. (C) sick person – This is someone who already has an illness, not someone who typically contracts it. Illogical. (D) elderly person – Elderly people are generally more vulnerable to illness due to weaker immune systems, so they are more likely to contract illness. (E) doctor's office – Also a place, not a person. The only logical and grammatically correct answer that refers to a type of person who is more likely to contract illness is: (D) elderly person. So, the answer is D.",
	"Answer": "D",
	"Keywords": ["contract", "illness", "vulnerable", "elderly", "sick", "immune", "system", "likely"]
	}
	##
	Give the response in strict accordance with the json format of the sample shown above.
	Q: {question}
	T: Please work through the problem step by step to deduce the question. And give 3-8 keywords that you think are important in the process of reasoning. Each keyword must be an independent word. The answer is given in standard json string format.

Figure 10: Prompt template for generating keywords in the reasoning process of the teacher model.

Question	(S1) 6 red peaches, 90 yellow peaches and 16 green peaches are in the basket. How many red and green peaches are in the basket?
Llama3-8B	(S2) There are 6 red peaches and 16 green peaches in the basket. (S3) So, the total number of red and green peaches is 6 + 16 = 22 ✓
GPT2-Medium (DSS)	(S2) There are 6 red peaches and 90 yellow peaches in the basket. (S3) So, the total number of red and green peaches is 6 + 90 = 96. ✗
GPT2-Medium (D-SANK)	(S2) There are 6 red peaches and 16 green peaches in the basket. (S3) So, the total number of red and green peaches is 6 + 16 = 22. ✓

(a)

Question	(S1) At the arcade Dave had won 14 tickets and lost 2 tickets. If he used 10 to buy some toys. How many tickets did Dave have left?
Llama3-8B	(S2) Dave won 14 tickets and lost 2 tickets. (S3) So, he had 14 - 2 = 12 tickets left. (S4) He used 10 tickets to buy some toys. (S5) So, he had 12 - 10 = 2 tickets left. ✓
GPT2-Medium (DSS)	(S2) Dave started with 14 tickets and lost 2 tickets. (S3)He used 10 tickets to buy toys. (S4)So, he had 14 - 2 = 12 tickets left. ✗
GPT2-Medium (D-SANK)	(S2)Dave started with 14 tickets and lost 2 tickets. (S3)So, he had 14 - 2 = 12 tickets left. (S4)He used 10 tickets to buy toys. (S5)So, he had 12 - 10 = 2 tickets left. ✓

(b)

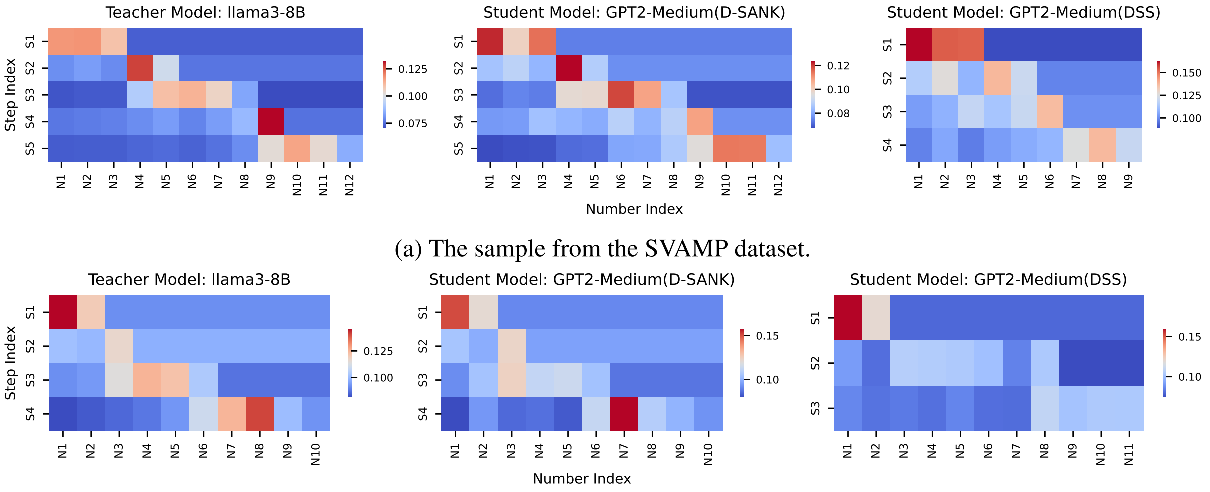
Question	(S1) 2 sports coaches went shopping together. The baseball coach bought 9 new baseballs for $3 each. The basketball coach bought 8 new basketballs for $14 each. How much more did the basketball coach spend than the baseball coach?
Llama3-8B	(S2) The baseball coach bought 9 new baseballs for 3 each, so the total cost for the baseballs is 9 * 3 = 27. (S3) The basketball coach bought 8 new basketballs for 14 each, so the total cost for the basketballs is 8 * 14 = 112. (S4) Now, we subtract the cost of the baseballs from the cost of the basketballs: 112 - 27 = 85. ✓
GPT2-Medium (MMIloss)	(S2) The baseball coach bought 9 new baseballs for 3 each, so he spent 9 * 3 = 27 dollars on baseballs. (S3) The basketball coach bought 8 new basketballs for 14 each, so he spent 8 * 14 = 112 dollars on basketballs. (S4) The difference in the amount spent by the basketball coach and the baseball coach is 112 - 27 = 85 dollars. ✗
GPT2-Medium (D-SANK)	(S2) The baseball coach bought 9 new baseballs for 3 each, so he spent (9 * 3) - (8 * 14) = 8 * 14 - 9 = 63. (S3) The basketball coach bought 8 new basketballs for 14 each, so he spent (8 * 14) = 112. (S4) The difference in the amount spent between the basketball coach and the baseball coach is 63 - 112 = 19. ✓

(c)

Question	(S1) Sarah has 9 books and Joseph had twice the number of Sarah's books, but he lost 2 of them. How many books does Joseph currently have?
Llama3-8B	(S2) Sarah has 9 books. (S3) Joseph had twice the number of Sarah's books, so he initially had 9 * 2 = 18 books. (S4) However, he lost 2 books, so he currently has 18 - 2 = 16 books. ✓
GPT2-Medium (MMIloss)	(S2) Sarah has 9 books, and Joseph's books are calculated as 9 times 2, 9 * 2 = 18 books. (S3) So Joseph now has 9 + 18 = 27 books. ✗
GPT2-Medium (D-SANK)	(S2) Sarah has 9 books. (S3) Joseph has twice the number of Sarah's books, so Joseph has 9 * 2 = 18 books. (S4) Joseph lost 2 of them, so he now has 18 - 2 = 16 books. ✓

(d)

Figure 11: Cases from SVAMP and GSM8K.



(a) The sample from the SVAMP dataset.

(b) The sample from GSM8K dataset.

Figure 12: We select one example each from SVAMP and GSM8K, visualizing stepwise attention on numerical tokens for student models distilled by DSS, MMIloss, and MoLSAKI, compared with the teacher model. Vertical and horizontal axes, respectively, denote the index of the step and the number.

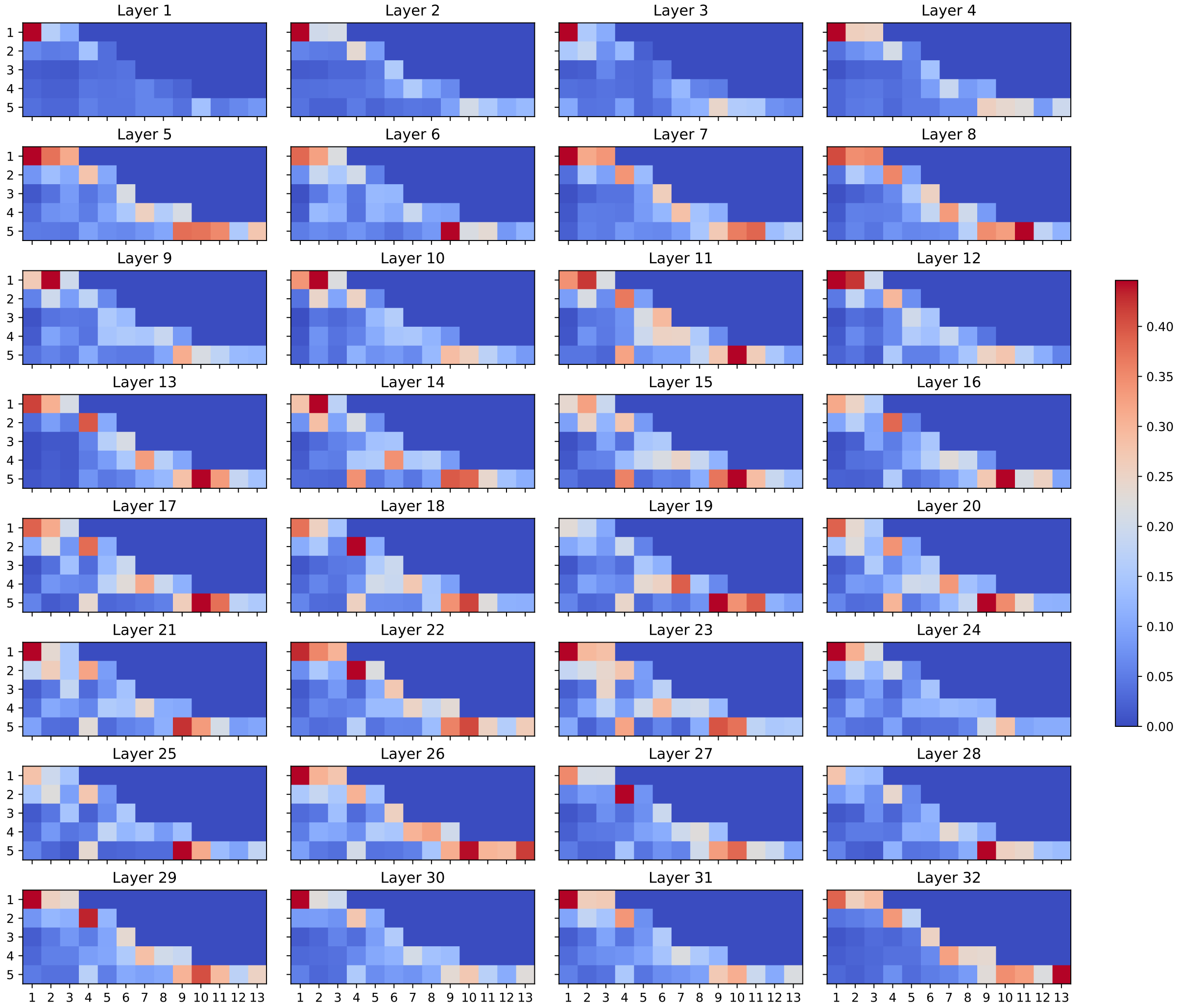


Figure 13: **Stepwise attention heatmap on critical tokens from the teacher model Llama3-8B, which consists of 32 layers for a specific example.** For each layer, the average attention is computed from all attention heads. The horizontal axis represents the order of critical tokens, while the vertical axis indicates the step number (counted from the beginning of the question). The example is as follows: "Question: A mailman is tasked with delivering 4 pieces of junk mail to each house in 16 blocks, with each block containing 17 houses. How many pieces of junk mail should he deliver in total? Rationale: The mailman delivers 4 pieces of junk mail to each house in 16 blocks, with each block containing 17 houses. Therefore, the total number of houses is 16 × 17 = 272. Since the mailman delivers 4 pieces of junk mail to each house, the total number of junk mail pieces is 272 × 4 = 1088."