

Cognitive Flow: An LLM-Automated Framework for Quantifying Reasoning Distillation

José Matos^{1,2} Catarina Silva^{1,2} Hugo Gonalo Oliveira^{1,2}

¹CISUC, LASI, Portugal

²DEI, University of Coimbra, Portugal

josematos@student.dei.uc.pt, {catarina,hroliv}@dei.uc.pt

Abstract

The ability of large language models (LLMs) to reason effectively is crucial for a wide range of applications, from complex decision-making to scientific research. However, it remains unclear how well reasoning capabilities are transferred or preserved when LLMs undergo Knowledge Distillation (KD), a process that typically reduces model size while attempting to retain performance. In this study, we explore the effects of model distillation on the reasoning abilities of various reasoning language models (RLMs). We introduce Cognitive Flow, a novel framework that systematically extracts meaning and map states in Chain-of-Thought (CoT) processes, offering new insights on model reasoning and enabling quantitative comparisons across RLMs. Using this framework, we investigate the impact of KD on CoTs produced by RLMs. We target DeepSeek-R1-671B and its distilled 70B, 32B and 14B versions, as well as QwenQwQ-32B from the Qwen series. We evaluate the models on three subsets of mathematical reasoning tasks with varying complexity from the MMLU benchmark. Our findings demonstrate that while distillation can effectively replicate a similar reasoning style under specific conditions, it struggles with simpler problems, revealing a significant divergence in the observable thought process and a potential limitation in the transfer of a robust and adaptable problem-solving capability.

1 Introduction

The rapid development and constant evolution of Large Language Models (LLMs) gave rise to a class of highly-capable Reasoning Language Models (RLMs). RLMs like OpenAI’s o1 (OpenAI et al., 2024) and o3¹, and DeepSeek-R1 (DeepSeek-AI et al., 2025), leverage reinforcement learning techniques to develop the ability to generate “thinking traces”, or Chain-of-Thought (CoT), in order

to solve complex tasks. To make this power accessible, reasoning from these large and heavy models (teacher models) is often distilled into smaller, more efficient LLMs (student models), leading to the transfer of capabilities that were once exclusive to high scale proprietary systems. This has led to the appearance and proliferation of open-source distilled reasoning models that achieve remarkable performance on reasoning benchmarks. This trend has also raised concerns about model homogenization, where different models tend to exhibit convergent behaviors and responses due to reliance on common distilled data sources (Lee et al., 2025).

However, the success achieved by distilled models in task performance remains somewhat uncertain. Are we truly distilling the “cognitive” properties of larger models, or merely replicating the superficial appearance of their reasoning? The CoT traces used for training reflect the behavior of teacher models, but they may not always accurately capture the underlying reasoning process (Agarwal et al., 2024; Turpin et al., 2023; Lindsey et al., 2025). This raises a critical issue that current evaluation methods fail to address: they rely on task accuracy, which is an inadequate and potentially misleading metric for assessing the quality of the transferred reasoning. This gap is a central and unsolved issue in the field. As stated in a recent comprehensive survey on LLM distillation, there is a critical need for “holistic evaluation protocols that measure nuanced LLM capabilities”, as current metrics “fail to evaluate whether distilled datasets preserve deeper reasoning abilities, such as chain-of-thought logic” (Fang et al., 2025). There is a need for tools to look deeper to assess quality, robustness and trustworthiness of the distilled thinking of these small models.

Looking to bridge this gap, we introduce a novel framework to extract meaning and identify transitions between the cognitive actions of a model’s CoT. As a case study, we apply it to quantify the

¹<https://openai.com/index/introducing-o3-and-o4-mini/>

fidelity of distilled reasoning. Our approach moves beyond solely structural analysis and comparison of final answers, modeling the observable reasoning process as a “Cognitive Flow”: a sequence of transitions between states of different cognitive activity. By systematically extracting these states (e.g. “Problem Decomposition”, “Calculation”, “Conclusion”) from a model’s reasoning traces and computing the probabilistic transitions between them, our technique allows for a quantitative comparison of reasoning patterns across different RLMs.

To demonstrate our framework, we conduct a novel comparative analysis of the DeepSeek-R1 family of RLMs (DeepSeek-AI et al., 2025) as a case study. These models are ideal for our analysis as they provide a clear teacher-student hierarchy, consisting of a 671B parameter teacher and its 70B, 32B and 14B student versions. By also including a control RLM from outside this model family, we can effectively isolate the impact of the distillation process itself. Our contributions are listed as follows:

- We propose a novel and scalable framework for automated extraction of the cognitive states of reasoning semantically, and the transitions between said states, allowing for analysis and comparison of reasoning flows in RLMs.
- We provide the first empirical measurement of how reasoning patterns are preserved or altered through distillation across different model sizes and task complexities.

2 Related Work

The appearance of **Chain-of-Thought (CoT)** marked a paradigm shift in the ability of LLMs to tackle complex reasoning tasks (Wei et al., 2022). It began with prompt engineering, but recently started being incorporated directly in the model’s Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) stages of the post-training process (OpenAI et al., 2024; DeepSeek-AI et al., 2025), giving way to modern RLMs, which treat the reasoning trace as an intrinsic part of their output. While CoT has shown to improve LLM performance by taking advantage of test-time scaling (Chen et al., 2025), one of its primary advantages lies in the promise of explainability. By expressing intermediate steps taken to solve a problem, a model reveals its internal thinking process.

However, this promise of transparent reasoning is shadowed by the challenge of faithfulness. A growing body of research demonstrates that while a model’s reasoning steps may be plausible and appear logical and coherent to a human observer, they are not guaranteed to be faithful to the actual internal process that produced the final answer (Agarwal et al., 2024; Turpin et al., 2023; Lindsey et al., 2025). Therefore, a model can arrive at a correct answer for incorrect reasons, meaning that “the end justifies the means”, and not the other way round.

In **Knowledge Distillation (KD)**, knowledge is transferred from a larger, more complex “teacher” model to a smaller, more efficient “student” model. In traditional KD, the student is trained on both the logits of the teacher model and a target dataset (Hinton et al., 2015; Polino et al., 2018). In the modern LLM paradigm, it involves performing instruction fine-tuning of smaller LLMs on a synthetic SFT dataset generated by larger ones. An emerging subject in this area is the transferring of reasoning and CoT capabilities from teacher RLMs to non-reasoning LLM students. Studies have shown that KD enables smaller LLMs to properly capture the reasoning capability and enable smaller models to achieve high performance on complex tasks, proving itself more effective than pure RL for such models (Shirgaonkar et al., 2024; DeepSeek-AI et al., 2025). The distillation process introduces another layer of uncertainty to the CoT faithfulness problem, as we are distilling a behavioral trace that may not be faithful to the teacher’s underlying cognitive process. The question shifts from if the student can learn to solve the task, to how it learns how to solve it. This concern is backed by findings suggesting that the structure of CoT demonstrations matter much more for successful learning than the correctness of the content itself (Li et al., 2025a). This implies that we can’t assess the broad performance of a distilled model solely based on its accuracy. Other research has focused on quantifying the broader effects of distillation (Lee et al., 2025), proposing a framework to measure the degree of “homogenization” across various LLMs by evaluating inconsistencies in “identity cognition” and similarity in final responses.

Given the challenges of faithfulness and the uncertainties of distillation, the need for methods

that **analyze and compare CoT** reasoning traces is crucial. While the broader field of **Explainable AI (XAI)** offers many techniques for assessing LLMs (Mumuni and Mumuni, 2025), for multi-step reasoning the most direct approach is to analyze the CoT itself. Some notable current approaches are Landscape of Thoughts (LoT) (Zhou et al., 2025) and ReasonGraph (Li et al., 2025b). LoT visualizes reasoning steps by mapping them as feature vectors in relation to answer choices. While useful for distinction between correct and incorrect paths, this approach is fundamentally tied to multiple-choice tasks and fails to capture the intrinsic semantic meaning of each reasoning step. ReasonGraph focuses on visualizing the structure of the reasoning path. It dynamically renders the CoT as a directed graph, and is able to represent sequential and tree based logic. However, it only focuses on the structure of the CoT, disregarding the semantic meaning of cognitive steps that make up the reasoning trace. Furthermore, both these tools are designed for single answer analysis, making it impossible to extract the common cognitive patterns that define a model’s characteristic reasoning style across many tasks of a similar type.

An evident gap arises in the current methodologies, which lack ways for quantitatively comparing the semantic and structural patterns of reasoning between teacher and student models. Existing tools tend to focus solely on the final answer, prioritize structural aspects excessively, or fall short in supporting a broader analysis. This work addresses that, by proposing a framework that enables visualization and quantitative comparison of a model’s “Cognitive Flow”, offering a method to evaluate the quality of reasoning distillation, extending beyond accuracy to focus on the reasoning itself.

3 Methodology

This section is divided into two main parts. First, we introduce the framework. Then, we present the experimental setup used to apply it to a family of teacher-student models, enabling a systematic comparison.

3.1 Extraction Framework

Here we present our primary contribution, a novel LLM-automated pipeline designed to turn raw, unstructured CoT text into a quantitative, structured

representation of a model’s reasoning style. Inspired by a state-of-the-art approach in dialogue analysis that models conversations as transitions between states in an unsupervised manner (Ferreira et al., 2024), our framework adapts this paradigm to model the observable reasoning process as a sequence of cognitive states. This enables the direct comparison of reasoning patterns between different models, and different tasks. The entire process, illustrated in Figure 1, consists of four main stages: (1) Step Segmentation, (2) Label Set Definition, (3) Step Classification, and (4) Flow Aggregation and Representation.

Step 1: Step Segmentation Raw outputs from the target LLM (i.e. the LLM being targeted for CoT analysis) are preprocessed. For each model completion, the reasoning trace is extracted from the text between the `<think>` and `</think>` tags, as this is the standard format used by all models in our study. The CoT is then split into a discrete sequence of reasoning steps, using the double-newline sequence (`\n\n`) as a delimiter. This results in an ordered list of strings, each representing a single reasoning step.

Step 2: Label Set Definition We then define a comprehensive set of cognitive state labels. This is handled by a dedicated Label Extractor LLM. To ensure that the resulting labels are representative and unbiased, we first create a diverse corpus by randomly sampling a large number of individual reasoning steps from all the outputs of the model under study. For this work, we sampled a total of 1000 steps. This volume was determined to be large enough to capture a wide range of reasoning behaviors, while being manageable within the context window of the Label Extractor LLM. This allowed the model to process the entire sample at once when defining the label set, ensuring the resulting labels were comprehensive. If LLMs with larger context windows are used as Label Extractors, the number of steps sampled can be further increased. This corpus of steps is handed over to the Label Extractor LLM, which is prompted with the task of generating a concise yet comprehensive set of labels that can categorize the underlying cognitive action of any given step. Along with the labels, the LLM is prompted to return clear definitions and illustrative examples for each one, using steps from the provided corpus. This set remains fixed and is used for all subsequent annotations. This step can be skipped if a predefined set of labels is provided.

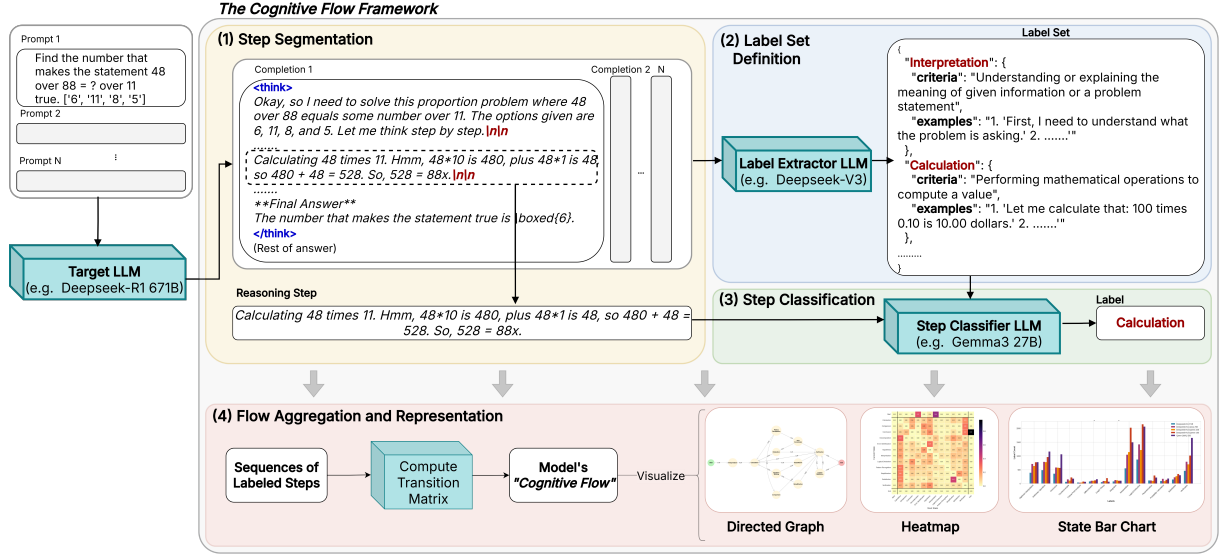


Figure 1: **Cognitive Flow Extraction Framework.** Our pipeline parses CoTs into sequential steps (1), defines a set of cognitive state labels (2), annotates each step (3), and aggregates the results into a “Cognitive Flow” (4), represented as a state transition matrix.

Step 3: Step Classification With a defined set of cognitive labels, the next stage is classifying every reasoning step for every CoT. This is performed by a second LLM-powered component, the Step Classifier LLM. This process is framed as a few-shot classification task. For each individual reasoning step, the step classifier model is provided with the step itself, and the complete set of cognitive state labels, along with definitions and examples for each one. The LLM is instructed to assign the single label that is the most appropriate, taking into account the underlying cognitive process present in the step. For every step in every completion, this process is repeated. Each CoT turns into a sequence of cognitive states that represent each step.

Step 4: Flow Aggregation and Representation The final stage of the pipeline transforms the individual reasoning sequences for all completions into what we refer to as the model’s Cognitive Flow. An aggregated, quantitative summary of a model’s overall reasoning style across all tasks presented. This aggregated behavior is represented by an $N \times N$ state transition matrix, M . Each element $M_{ij} \in M$ represents the conditional probability of the model transitioning from cognitive state L_i to state L_j . This probability is computed by counting all occurrences of that transition across the entire set of reasoning sequences, and then normalizing it by the total number of outgoing transitions from state L_i . The resulting matrix M serves as a finger-

print of the model’s observable reasoning process and enables direct, numerical comparisons with other RLMS. To capture only significant shifts in reasoning, the analysis exclusively counts transitions between different cognitive states. Transitions within the same state are ignored as they do not represent a progression in the model’s reasoning flow. In addition, we inserted two extra states into each sequence: *Start* and *End*. This is useful to analyze with which cognitive actions the model tends to start its reasoning, and how it tends to end it. Visualizing a model’s Cognitive Flow can lead to increased explainability, therefore we propose three different yet complementary ways to represent the resulting flows.

To provide an intuitive, high-level overview of a model’s most dominant reasoning pathways, we visualize the cognitive flow as a directed graph, where vertices correspond to reasoning states and the edges to one-way transitions between them. On the one hand, a graph containing all the states and transition probabilities would be cluttered and too hard to interpret. On the other, limiting transitions can ease the reading of the graph, at the cost of losing information on a few transitions. Here, as others have done for simplifying flow graphs (Ferreira et al., 2024), we used a threshold θ , where transitions with a probability below that threshold (e.g. $\theta = 0.10$) would not be included in the final graph. Some examples of this representation are

depicted in Appendix D.

For a complete, fine-grained view of the model’s reasoning dynamics, we directly visualize the state transition matrix M as a heatmap (example in Figure 2) where the y-axis and the x-axis represent the “Current” (L_i) and the “Next” (L_j) states, respectively. The color intensity of the cell at (i, j) corresponds to the transition probability $M_{i,j}$. This visualization is particularly effective for identifying subtle differences in reasoning patterns, finding a model’s most (and least) probable next steps from any given state, and discovering rare paths, omitted in threshold limited graphs.

Finally, to understand a model’s overall reasoning bias, irrespective of transitions, we visualize the stationary distribution cognitive states (example in Figure 3). A chart like this can help answer the question: “What cognitive operations does this model spend the most computational load on?”. We compute this by calculating the number of tokens in each step with a certain assigned cognitive label across the entire group of label sequences for a given model, and then normalize by the total number of tokens. This static view is complementary to the dynamic views of the graph and heatmap, revealing the model’s tendencies towards certain states of cognitive activity.

3.2 Experimental Setup

To empirically test the proposed framework, we designed an experiment to quantify the fidelity of reasoning distillation across a controller teacher-student model family.

3.2.1 Models

We selected a family of models centered around the DeepSeek-R1 series, forming a clear teacher-student group (DeepSeek-AI et al., 2025). A control model from a different family was included to establish a baseline for what a reasoning style not brought up by distillation from the teacher model looks like.

Teacher Model DeepSeek-R1-671B. This RLM will serve as the source of the teacher’s reasoning traces.

Student Models We selected three models distilled from DeepSeek-R1’s outputs (DeepSeek-AI et al., 2025), varying in size and base architecture: DeepSeek-R1-Distill-Llama-70B, DeepSeek-R1-Distill-Qwen-32B and DeepSeek-R1-Distill-Qwen-14B.

This selection allows the analysis of how reasoning is preserved across different parameter counts, and even foundational model families, through the distillation process.

Control Model QwenQwQ-32B, shares the same foundational architecture, parameter count, and inference conditions (4-bit quantization) as the DeepSeek-R1-Distill-Qwen-32B student model. This pairing allows us to isolate the post-training approach itself as the independent variable between them, thereby disentangling the results of distillation from the effects of model compression and quantization. It allows us to quantify the difference between a student inheriting its teacher’s patterns versus a model of the same size that developed its reasoning through reinforcement learning².

3.2.2 Dataset and Tasks

To assess the model’s reasoning, we used tasks from the Massive Multitask Language Understanding (MMLU) dataset (Hendrycks et al., 2020), focusing on three subsets in the Mathematics domain, with increasing levels of complexity: Elementary, High School and College (Examples in Appendix C). The combination of the three test sets from each subset amounts to a total of 748 unique problems. This selection allows us to study not only the effectiveness of reasoning distillation, but also how it changes as the complexity of the task varies. To effectively use the reasoning of DeepSeek-R1-671B as a baseline, all three label sets at different problem complexities were generated by the Label Extractor LLM based on its outputs. Each label set is then held fixed for all model’s annotations within its corresponding complexity level.

3.2.3 Framework LLMs

As part of our framework we used two different LLMs for all experiments. In the Label Extractor LLM role, we employed DeepSeek-V3 (Liu et al., 2024) via API. The decision to use this model stemmed from the need of a capable model that could withstand a large enough context, so we could provide it with a large sample of steps from the produced reasoning traces, and it could extract a meaningful set of labels that could represent the whole corpus. For the Step Classifier LLM, we used a 4-bit quantized aware trained (QAT) version of Gemma3-27B-IT³, an instruction-tuned version

²<https://qwenlm.github.io/blog/qwq-32b/>

³https://huggingface.co/google/gemma-3-27b-it-qat-q4_0-gguf

of Google’s Gemma3-27B (Gemma Team, 2025). We chose this model because this classification simpler and repetitive, but still requires some capability to understand the underlying cognitive behavior behind the step.

3.2.4 Implementation Details

Answers for the tasks were gathered from all models using default sampling parameters, except for temperature. A temperature of 0.0 was used for the DeepSeek family models, as recommended by official documentation for math related tasks⁴, and 0.6 for QwenQwQ-32B, also recommended⁵. DeepSeek-R1-671B was prompted via the official DeepSeek API, DeepSeek-R1-Distill-Llama-70B via the Groq API⁶, and the remaining models were inferred locally using quantized versions. For the 32B models, we used 4-bit quantization, and for the 14B model, 8-bit quantization. The choice to use models served via API was due to limits in computational resources, and the same reason applies to the choice of using quantized versions of the models.

A recurring issue during inference were reasoning loops, where a model would produce endless repetitions. To handle this systematically, if a model failed to produce a coherent output, exceeding its context window, it was resampled up to five times. If the looping behavior persisted after five attempts, the task was skipped for that specific model⁷.

3.2.5 Flow Evaluation Metrics

To quantitatively compare the Cognitive Flow between a teacher model and a student model, we employ two distinct similarity metrics. Together, they provide a comprehensive measure of flow alignment and allow us to assess the fidelity of the reasoning distillation process from different perspectives.

In order to assess the directional agreement in reasoning transitions, we compute the average Cosine Similarity (CS) between corresponding

rows of the two transition matrices. A score closer to 1 indicates that the student model’s relative probabilities of transitioning from a state better align with the teacher’s. In other words, it means a high degree of similarity in the direction of the cognitive flow.

The Kullback-Leibler Divergence (KLD) measures the “information loss” when approximating the teacher’s reasoning with the student’s. For each state i , the divergence from the student’s distribution (q_i) to the teacher’s (p_i) is:

$$D_{KL}(p_i||q_i) = \sum_j p_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right) \quad (1)$$

A lower KLD indicates the student model’s transition probabilities are a more faithful approximation of the teacher’s, reflecting a closer match in the magnitude and confidence of state transitions. In essence, CS measures the directional alignment of the cognitive flow, while KLD measures the alignment in probabilistic confidence. A high CS and low KLD score indicate strong flow alignment, whereas a divergence in either metric signals reasoning flow unalignment between the models.

4 Results & Discussion

Our analysis, conducted through the Cognitive Flow framework, provides a quantitative measurement of reasoning style similarity after distillation. Results are summarized in Table 1 and reveal a complex relationship between distillation, task complexity and model robustness. Our framework is designed to be agnostic of the correctness of the final answer, focusing instead on the semantic structure of the reasoning traces. This way we can assess *how* distilled models replicate their teacher’s cognitive patterns, leading to insights that go beyond accuracy-based evaluations.

To ensure increased confidence on these results, we conducted a human evaluation on 150 randomly sampled reasoning steps (Appendix A). The Step Classifier LLM achieved a Cohen’s Kappa of 0.836, a score indicating “almost perfect agreement” with the human annotator (Landis and Koch, 1977). This confirms that the Step Classifier LLM reliably captures the cognitive states of the observable reasoning process, validating the foundation for the upcoming analysis.

We first establish a baseline by evaluating models on tasks of moderate complexity (High School).

⁴https://api-docs.deepseek.com/quick_start/parameter_settings

⁵<https://huggingface.co/Qwen/QwQ-32B>

⁶<https://console.groq.com/docs/model/deepseek-r1-distill-llama-70b>

⁷The failed task counts are as follows: QwenQwQ-32B failed on 11/100 College and 28/270 High School tasks. DeepSeek-R1-Distill-Qwen-32B failed on 6/100 College tasks, and DeepSeek-R1-Distill-Llama-70B failed on 2/270 High School tasks.

Task Complexity	Metric	Distilled Models			Control(RL)
		L70B	Q32B	Q14B	QwQ32B
College (Highest)	KLD (↓)	0.423	0.783	0.293	0.260
	CS (↑)	0.954	0.944	0.928	0.956
High School (Medium)	KLD (↓)	0.065	0.080	0.109	0.074
	CS (↑)	0.959	0.958	0.942	0.958
Elementary (Lowest)	KLD (↓)	1.615	1.827	1.356	0.268
	CS (↑)	0.829	0.804	0.819	0.956

Table 1: Cognitive Flow similarity between the teacher model (DeepSeek-R1) and student models. Arrows indicate whether lower (↓) or higher (↑) scores are better. The best performance in each category is highlighted in bold.

In this setting, KD proves highly effective at transferring reasoning flows between teacher and student. As detailed in Table 1, all distilled models exhibit a high degree of flow alignment with the teacher, DeepSeek-R1-671B. The Llama-70B distilled model (L70B) achieves the highest fidelity, with a CS of 0.959 and a KLD of only 0.065. This demonstrates that distilled RLMs can, under the right conditions, faithfully inherit the teacher’s characteristic reasoning.

However, when models were tested on tasks of either higher or lower complexity, the high fidelity seen previously degrades significantly, revealing a critical weakness in generalization.

On highly complex problems (College), while the similarity in the direction of reasoning paths remains high (CS > 0.92 for all models), KLD scores increase. This indicates that, while the distilled models can still follow the teacher’s cognitive path, they do so with a different probabilistic confidence, revealing a divergence from the learned reasoning style under high cognitive requirements.

More surprisingly, we find a significant “failure” of distillation on the simpler Elementary tasks. For all distilled models, cognitive flows exhibit a large gap from the teacher, with KLD scores soaring to over 1.3 and CS scores lowering to around 0.8 (Table 1). The combination of a low CS and a high KLD suggests that distilled models are not only transitioning between states with different probabilistic confidence like before, they are now following a fundamentally different reasoning structure and flow. This structural divergence is visually apparent in the Cognitive Flow heatmaps in Figure 2, where the transition matrix of the distilled RLM shows clear differences from the teacher’s. It is important to note that the smaller models (32B and 14B) were evaluated using quantized versions. Quantization is an additional factor to consider, as compressing the models inevitably leads to some

information loss, potentially impacting accuracy.

Results obtained from the control model, QwenQwQ-32B, provide a counterpoint and help explain the findings so far. On the Elementary tasks where the distilled models heavily diverged from the teacher model, the control model actually resembled a very similar reasoning, achieving the highest CS (0.956) and the lowest KLD (0.268). This result is crucial. Firstly, it demonstrates that flow unalignment is not an inherent limitation posed by model size or architecture. Moreover, and because our control and 32B student models were both run under identical 4-bit quantization, we can also conclude that divergence is not merely an artifact of model compression. Secondly, after isolating the post-training methodology (distillation vs. reinforcement learning) as the key variable, this provides strong evidence that distillation, while effective for transferring specific skills, may be less effective at generating a robust and adaptable reasoning style than an independent learning process such as RL.

Our analysis of cognitive states distribution in Figure 3 provides an explanation for this difference. On Elementary tasks, the distilled models neglect the crucial *Verification* step, using about 20% less tokens towards it, compared to the teacher. This suggests a form of overfitting, where distilled models learn to replicate the teacher’s sophisticated problem-solving logic but fail to generalize to simpler problems where those advanced techniques are unnecessary, and fundamental steps like verification can play a huge part. Furthermore, across all three levels of complexity tested, the control model consistently allocates a larger portion of its computational effort to *Verification* (Figure 3).

Findings from this experiment highlight the profound impact of the underlying training objective. Distillation is a form of process supervision, implemented as SFT on the teachers’ outputs, mean-

Cognitive State Transitions in Elementary Maths Subset

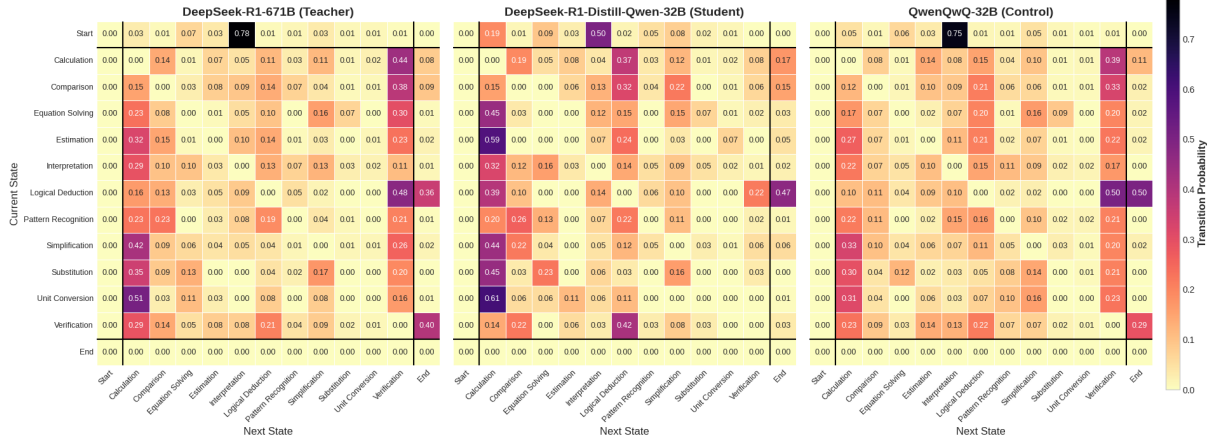


Figure 2: **Cognitive Flow Heatmaps for the Elementary Maths Subset.** Transitions of the distilled student (center) diverge significantly from its teacher (left). The control model (right), develops a flow remarkably similar to the teacher, isolating the divergence as a cause of the distillation process.



Figure 3: **Difference in Cognitive State Percentage from the Teacher.** On elementary tasks (bottom), distilled models exhibit a critical difference: a pronounced lack of the *Verification* step. In contrast the control model, trained via RL, consistently allocates more cognitive effort to verification.

ing that the students’ objective is to replicate the teacher’s reasoning path precisely. This effectively forces the student to adopt the teacher’s reasoning style but, as we show, can lead to fragile behavior and affect the generalization capability.

On the other hand, the control model’s RL training represents outcome supervision. Its policy is not optimized to follow a specific, defined path, but to maximize a reward signal based on the cor-

rectness of the outcome, encouraging emergent reasoning strategies. The model is free to discover that frequent self-verification is a highly-rewarding behavior because it often leads to correct answers, resulting in a robust, generalized reasoning ability developed through exploration and iteration.

5 Conclusion

We introduced Cognitive Flow, a novel framework for quantifying and comparing the semantic structure of an LLM’s reasoning. By modeling CoT as a sequence of transitions between cognitive states, we moved beyond accuracy-based metrics to assess the fidelity of reasoning distillation.

Our analysis of the DeepSeek-R1 family provided a key insight. Simply learning to replicate a teacher’s reasoning style through SFT is a fragile strategy when trying to distill reasoning flows between models. While highly effective for a set of tasks under the right conditions, the approach fails to generalize, leading to an increase of flow unalignment on simpler problems and the neglect of fundamental cognitive actions, like self-verification. In contrast, an independently RL trained model demonstrated a more robust and adaptable reasoning style.

These findings challenge the paradigm of pure KD as is, for creating smaller and efficient reasoning LLMs. They suggest that future work should focus on hybrid training approaches that combine the guidance of distillation with the exploratory benefits of RL to build models that can not only

achieve higher performance, but also higher robustness and alignment with their teachers.

The code used for our experimentation is available at <https://github.com/NLP-CISUC/Cognitive-Flow>.

Limitations

We identify the following limitations in our study:

First, using two core LLMs introduces potential variability. The labeling system itself can be seen as having further limitations: the set of cognitive labels is generated from a specific sample of steps and may not be representative, it potentially simplifies reasoning by assigning a single label to complex steps, and its static and “offline” nature could face problems being adapted for real-time explainability systems. Future work could address these, exploring the robustness of label generation, multi-label classification and dynamic set updating.

Secondly, our framework quantifies “reasoning style” as a sequence of cognitive state transitions. We acknowledge that this is a proxy for the underlying reasoning process.

Additionally, our empirical analysis is centered on the DeepSeek model family and mathematical reasoning tasks. Consequently, the conclusions about distillation’s fidelity may be specific to these models, and the identified cognitive patterns are influenced by the mathematical domain. To establish these findings as a general principle, future work should apply the framework to more diverse teacher-student pairs of models and across different reasoning domains, such as instruction following or commonsense reasoning.

Finally, due to computational constraints a mix of inference methods were used in our experimentation. The teacher model and one student were accessed via API, while others were run locally using quantized versions. Quantization can partially degrade a model’s performance and accuracy. While directly comparing the full-precision API-served teacher and the quantized students is a limitation, our core findings about distillation’s fragility are primarily drawn from a controlled comparison between the two 4-bit quantized 32B models. Albeit, future applications of the framework should, where possible, use uniform inference environments for all models to ensure a more direct and isolated comparison.

Acknowledgements

This work was partially supported by the AMALIA project, funded by FCT/IP in the context of measure RE-C05-i08 of the Portuguese Recovery and Resilience Program; by the Portuguese Recovery and Resilience Plan through project C645008882-00000055, Center for Responsible AI; and by national funds through FCT – Foundation for Science and Technology I.P., in the framework of the Project CISUC (UIDB/00326/2025 and UIDP/00326/2025).

References

- Chirag Agarwal, Sree Harsha Tanneru, and Himabindu Lakkaraju. 2024. [Faithfulness vs. Plausibility: On the \(Un\)Reliability of Explanations from Large Language Models](#). *arXiv preprint*. ArXiv:2402.04614 [cs].
- Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025. [Towards Reasoning Era: A Survey of Long Chain-of-Thought for Reasoning Large Language Models](#). *arXiv preprint*. ArXiv:2503.09567 [cs].
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. [DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning](#). *arXiv preprint*. ArXiv:2501.12948 [cs].
- Luyang Fang, Xiaowei Yu, Jiazhang Cai, Yongkai Chen, Shushan Wu, Zhengliang Liu, Zhenyuan Yang, Hao-ran Lu, Xilin Gong, Yufang Liu, Terry Ma, Wei Ruan, Ali Abbasi, Jing Zhang, Tao Wang, Ehsan Latif, Wei Liu, Wei Zhang, Soheil Kolouri, and 5 others. 2025. [Knowledge Distillation and Dataset Distillation of Large Language Models: Emerging Trends, Challenges, and Future Directions](#). *arXiv preprint*. ArXiv:2504.14772 [cs].
- Patrícia Ferreira, Daniel Martins, Ana Alves, Catarina Silva, and Hugo Gonalo Oliveira. 2024. [Unsupervised Flow Discovery from Task-oriented Dialogues](#). *arXiv preprint*. ArXiv:2405.01403 [cs].
- Gemma Team. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. [Distilling the Knowledge in a Neural Network](#). *arXiv preprint*. ArXiv:1503.02531 [stat].

- J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174.
- Sunbowen Lee, Juntong Zhou, Chang Ao, Kaige Li, Xinrun Du, Sirui He, Haihong Wu, Tianci Liu, Jiaheng Liu, Hamid Alinejad-Rokny, Min Yang, Yitao Liang, Zhoufutu Wen, and Shiwen Ni. 2025. [Quantification of Large Language Model Distillation](#). *arXiv preprint*. ArXiv:2501.12619 [cs] version: 3.
- Dacheng Li, Shiyi Cao, Tyler Griggs, Shu Liu, Xiangxi Mo, Eric Tang, Sumanth Hegde, Kourosh Hakhmaneshi, Shishir G. Patil, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. 2025a. [LLMs Can Easily Learn to Reason from Demonstrations Structure, not content, is what matters!](#) *arXiv preprint*. ArXiv:2502.07374 [cs].
- Zongqian Li, Ehsan Shareghi, and Nigel Collier. 2025b. [ReasonGraph: Visualisation of Reasoning Paths](#). *arXiv preprint*. ArXiv:2503.03979 [cs].
- Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, and 8 others. 2025. [On the biology of a large language model](#). *Transformer Circuits Thread*.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Fuseini Mumuni and Alhassan Mumuni. 2025. [Explainable artificial intelligence \(XAI\): from inherent explainability to large language models](#). *arXiv preprint*. ArXiv:2501.09967 [cs].
- OpenAI, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Ifitimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, and 243 others. 2024. [OpenAI o1 System Card](#). *arXiv preprint*. ArXiv:2412.16720 [cs].
- Antonio Polino, Razvan Pascanu, and Dan Alistarh. 2018. [Model compression via distillation and quantization](#). *arXiv preprint*. ArXiv:1802.05668 [cs].
- Anup Shirgaonkar, Nikhil Pandey, Nazmiye Ceren Abay, Tolga Aktas, and Vijay Aski. 2024. [Knowledge Distillation Using Frontier Open-source LLMs: Generalizability and the Role of Synthetic Data](#). *arXiv preprint*. ArXiv:2410.18588 [cs].
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. [Language Models Don’t Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 74952–74965. Curran Associates, Inc.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed 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*, volume 35, pages 24824–24837. Curran Associates, Inc.
- Zhanke Zhou, Zhaocheng Zhu, Xuan Li, Mikhail Galkin, Xiao Feng, Sanmi Koyejo, Jian Tang, and Bo Han. 2025. [Landscape of Thoughts: Visualizing the Reasoning Process of Large Language Models](#). *arXiv preprint*. ArXiv:2503.22165 [cs].

A Validation of the Step Classifier LLM

To ensure the validity of our framework’s automated classification, we measured the performance of the Step Classifier LLM, Gemma3-27B-IT-QAT⁸, against a human expert annotator.

Methodology We randomly sampled 150 steps, ten for each model, for each task complexity. One of the paper’s authors manually assigned a label to each step, while having access to the corresponding label set, provided by the Label Extractor LLM.

Results The classifier’s performance is depicted in Figure 4. Overall, the LLM annotator achieved a Cohen’s Kappa of 0.836, indicating “almost perfect agreement” with the human (Landis and Koch, 1977).

Discussion The confusion matrix visually confirms a high accuracy. It is to note that most errors are concentrated in semantically similar categories. A primary source of disagreement occurs between *Interpretation*, *Logical Deduction* and more specific actions like *Substitution*. This suggests that even when the LLM’s decision differs from the human’s, this disagreement can be attributed to semantic ambiguity within the steps, rather than classifier failure.

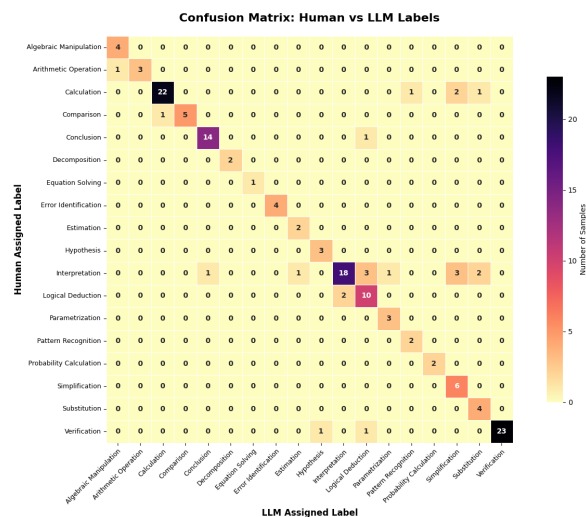


Figure 4: **Confusion matrix of the Step Classifier LLM vs. human labels.** The strong diagonal indicates a high number of equal classifications.

⁸https://huggingface.co/google/gemma-3-27b-it-qat-q4_0-gguf

B Framework Details

This appendix provides a more detailed look into the implementation of the framework, specifically design of the prompts used to guide the core LLM components, foundational to our analysis.

B.1 Label Extractor LLM Prompt

The Label Extractor LLM is tasked with defining a comprehensive and domain-agnostic set of cognitive labels. The system prompt for this model is detailed in Figure 5. This stage is unsupervised and foundational to the entire framework.

The prompt guides the LLM to produce a set of labels that are both meaningful and broadly usable. The input to this LLM is a diverse corpus of 1000 reasoning steps, randomly sampled from the outputs of the teacher model at each complexity, ensuring the resulting labels are representative of the model’s reasoning behavior. Key criteria are provided directly in the prompt: labels must be applicable across different domains, broad enough to be versatile but distinct enough to be meaningful, and oriented towards clustering similar reasoning patterns. By specifying an aim of 5-12 labels, we guide the model to produce a concise set that is comprehensive without being too specific.

B.2 Step Classifier LLM Prompt

The Step Classifier LLM is tasked with assigning a single cognitive label to each step of a model’s CoT. The prompt used to guide this model is depicted in Figure 6.

The prompt explicitly defines the model as a “Step Annotator” and frames the LLM’s task as a few-shot classification problem. For each step to be classified, the model is provided with both the reasoning step and the full set of cognitive state labels, along with definitions and examples for each one. The prompt then defines a strict set of rules both for classification criteria and format, minimizing variability and ensuring the model’s output is consistently usable to compute the transition matrix.

C Examples of CoT Annotations

Here we provide concrete examples of the annotated CoT outputs used in our analysis. Tables 2, 3 and 4 illustrate how the Cognitive Flow framework deconstructs a model’s raw reasoning trace into a sequence of discrete cognitive states. Each example presents a problem from one of the used MMLU subsets.

You are a Reasoning Pattern Analyzer specializing in extracting domain-agnostic labels from chain-of-thought reasoning steps. Your task is to analyze samples of reasoning steps and generate a concise set of labels (between 5-12) that represent fundamental thinking patterns that can be used for clustering similar reasoning approaches across domains.

INPUT: You will receive phrases representing individual steps in chain-of-thought reasoning processes from various problem-solving scenarios across different domains.

OBJECTIVE: Identify underlying cognitive patterns and create a set of broad, domain-agnostic labels that can effectively cluster similar thinking approaches, regardless of the specific content domain.

ANALYSIS APPROACH:

1. Examine each reasoning step independently, focusing on the cognitive operation being performed rather than the domain-specific content.
2. Identify the fundamental mental process occurring in each step (such as interpretation, conclusion formation, decomposition, etc.).
3. Prioritize breadth over depth - aim to cover diverse reasoning types rather than making fine distinctions between similar reasoning patterns.
4. Look for general patterns that would appear across completely different domains and problems.

LABEL CRITERIA:

1. Domain-agnostic: Labels should describe thinking patterns applicable across any domain (math, literature, business, science, etc.).
2. Broad yet meaningful: Each label should capture a distinct category of reasoning but be applicable across multiple contexts.
3. Clustering-oriented: Labels should effectively group similar reasoning steps together, even when the content differs completely.
4. Independent: Focus on labeling individual steps, not relationships between steps.
5. Comprehensive: The complete set should cover the major reasoning patterns present in the sample (aim for 5-12 total labels).

OUTPUT FORMAT: Provide your output in the following JSON-like structure and nothing else. Do not include any additional text or formatting in your response. Do NOT include special characters or line breaks in the label names:

```
{
  "LABEL1": {
    "criteria": "A clear definition of the cognitive operation it represents",
    "examples": "1-2 examples of how this pattern might appear across different domains" },
  "LABEL2": {
    "criteria": "A clear definition of the cognitive operation it represents",
    "examples": "1-2 examples of how this pattern might appear across different domains" },
  ...
}
```

Examples of possible general labels include "Interpretation," "Conclusion," "Breakdown," "Arithmetic Operation" - but you are not limited to these. Create labels that best represent the patterns you observe in the provided steps.

Remember, your primary goal is to create labels that will be effective for clustering similar reasoning steps across completely different domains and problems. Aim for 5-12 total labels.

Figure 5: System prompt used for the Label Extractor LLM.

You are a Step Annotator that analyzes reasoning steps and assigns the most appropriate label from a provided set. Your task is to categorize each step according to its dominant cognitive operation.

INPUT:

1. A set of labels with their criteria and examples (provided in the user message)
2. A reasoning step to analyze

TASK:

Assign EXACTLY ONE label from the provided set to the reasoning step. While steps may involve multiple cognitive processes, you must identify the MOST DOMINANT process and select only one label.

ANALYSIS APPROACH:

1. Examine the reasoning step, identifying the primary cognitive process present
2. Compare this process against the criteria for each provided label
3. Select the single most appropriate label that best captures the dominant reasoning pattern

IMPORTANT RULES:

- Select ONLY ONE label per step, even if multiple processes are present
- Prioritize the cognitive pattern that is most central or essential to the step
- Use exactly the label text as provided in the options, with the same spelling and capitalization
- Always select one label from the set, never leave a step unlabeled
- If the step does not fit any label totally, choose the one that is closest in meaning

OUTPUT FORMAT:

Return ONLY the chosen label name in plain text, with no additional text, explanation, or formatting.

Your entire response should be only the selected label text, nothing more.

Figure 6: **System prompt used for the Step Classifier LLM.**

Table 2: Annotated CoT of DeepSeek-R1-671B for an Elementary maths subset problem.

Problem: A worker on an assembly line takes 7 hours to produce 22 parts. At that rate how many parts can she produce in 35 hours? ['220 parts', '770 parts', '4 parts', '110 parts']	
LLM Reasoning Step	Cognitive Label
Okay, let me try to figure out this problem. So, the question is about a worker on an assembly line who takes 7 hours to produce 22 parts. We need to find out how many parts she can produce in 35 hours at the same rate. The options given are 220, 770, 4, or 110 parts. Hmm, let me think step by step.	Interpretation
First, I need to determine the rate at which the worker is producing parts. The rate is usually calculated as parts per hour. Since she makes 22 parts in 7 hours, her rate would be 22 divided by 7. Let me write that down: Rate = 22 parts / 7 hours. That simplifies to approximately 3.142857 parts per hour. But maybe I should keep it as a fraction to be exact. So, 22/7 parts per hour.	Calculation
Now, if we want to know how many parts she can produce in 35 hours, we can multiply her hourly rate by the number of hours. So, the total parts would be (22/7) parts/hour * 35 hours. Let me compute that. The hours unit cancels out, which is good. So, 22 divided by 7 multiplied by 35. Let me do the math here.	Calculation
First, 35 divided by 7 is 5. Then, 5 multiplied by 22 is 110. Wait, so that calculation gives 110 parts. Let me check that again. $22/7 * 35$. If I write 35 as $7*5$, then $22/7 * 7*5 = 22*5 = 110$. Yeah, that seems right. So, 110 parts in 35 hours.	Calculation
Looking at the options given, 110 parts is one of them. So, that should be the answer. But wait, let me make sure I didn't make a mistake in the calculation. Let me do it another way to cross-verify.	Comparison
Alternatively, we can set up a proportion. If 7 hours correspond to 22 parts, then 35 hours correspond to x parts. The proportion would be:	Equation Solving
$7 \text{ hours} / 35 \text{ hours} = 22 \text{ parts} / x \text{ parts}$	Equation Solving
But wait, is that the correct proportion? Hmm, actually, as the number of hours increases, the number of parts should increase as well. So, the ratio of hours should be equal to the ratio of parts. So, $7/35 = 22/x$. Solving for x, cross-multiplying: $7x = 35*22$. Then, $x = (35*22)/7$. Simplify that, 35 divided by 7 is 5, so $x = 5*22 = 110$. Yep, same answer. So that's consistent.	Equation Solving
Alternatively, another way: how many 7-hour periods are there in 35 hours? Since 35 divided by 7 is 5. So, 5 times 7 hours is 35 hours. Therefore, she can produce 5 times the number of parts she makes in 7 hours. So, 22 parts * 5 = 110 parts. That's another way to look at it. So, all methods lead to 110 parts. Therefore, the answer is 110 parts.	Calculation
Let me check the options again: 220, 770, 4, 110. 110 is there, so that's the correct choice. I think that's solid. I don't see any mistakes in the calculations. So, yeah, 110 parts is the answer.	Verification

Table 3: Annotated CoT of DeepSeek-R1-Distill-Llama-70B, for a High School maths subset problem.

Problem: The symbol $5!$ means $5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$. What is the greatest odd integer that is a factor of $5!$? ['5', '3', '15', '31']	
LLM Reasoning Step	Cognitive Label
Okay, so I have this problem here: The symbol $5!$ means $5 \times 4 \times 3 \times 2 \times 1$. I need to find the greatest odd integer that is a factor of $5!$. The options are '5', '3', '15', '31'. Hmm, let me think about how to approach this.	Interpretation
First, I remember that the factorial of a number, like $5!$, is the product of all positive integers up to that number. So, $5!$ is $5 \times 4 \times 3 \times 2 \times 1$. Let me calculate that first. 5×4 is 20, then 20×3 is 60, 60×2 is 120, and 120×1 is still 120. So, $5!$ equals 120.	Calculation
Now, the question is asking for the greatest odd integer that is a factor of 120. So, I need to find the largest odd number that divides 120 without leaving a remainder.	Interpretation
I know that factors of a number are the numbers that can divide it without leaving a remainder. So, to find the greatest odd factor, I need to look at the factors of 120 and pick the largest odd one.	Interpretation
But maybe there's a smarter way than listing all factors. Let me think. 120 is an even number because it ends with a 0, which is even. So, it's divisible by 2. But we want an odd factor, so I need to remove all the factors of 2 from 120.	Hypothesis
Wait, that might be a good approach. If I factor 120 into its prime factors, I can then remove the factors of 2 and see what's left. The remaining product will be the greatest odd factor.	Hypothesis
Let me try that. Prime factorization of 120: 120 can be divided by 2 first. $120 \div 2 = 60$. $60 \div 2 = 30$. $30 \div 2 = 15$. Now, 15 is not divisible by 2, so I switch to the next prime number, which is 3. $15 \div 3 = 5$. Then, 5 is a prime number itself. So, the prime factors of 120 are $2 \times 2 \times 2 \times 3 \times 5$.	Calculation
So, $120 = 2^3 \times 3 \times 5$. Now, if I remove all the factors of 2, which are 2^3 , I'm left with 3×5 . Multiplying those together gives 15. So, 15 is the greatest odd factor of 120.	Calculation
Alternatively, I can list all the factors of 120 and pick the largest odd one. Let's try that method to double-check.	Hypothesis
Factors of 120: 1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 20, 24, 30, 40, 60, 120.	Calculation
Now, the odd factors among these are 1, 3, 5, 15. So, the largest one is 15. That confirms my earlier conclusion.	Verification
So, the answer should be 15. Looking back at the options: '5', '3', '15', '31'. 15 is one of them, so that must be the correct choice.	Comparison
Wait, just to make sure I didn't make a mistake in my prime factorization. $2^3 \times 3 \times 5$ is $8 \times 3 \times 5$, which is 8×15 , which is 120. Correct. So, removing the 2s gives 15. So, yes, 15 is the greatest odd factor.	Verification
Another way to think about it is that any number can be expressed as $2^k \times \text{odd number}$. So, 120 is 8×15 , where 8 is 2^3 and 15 is the odd part. So, the greatest odd factor is 15.	Pattern Recognition
I think that's solid. So, I'm confident that 15 is the correct answer.	Conclusion

Table 4: Annotated CoT of QwenQwQ-32B for a College maths subset problem.

Problem: Statement 1 : Suppose s_n and t_n are sequences of complex numbers such that $\lim_{n \rightarrow \infty} s_n = s$ and $\lim_{n \rightarrow \infty} t_n = t$. $\lim_{n \rightarrow \infty} s_n t_n = st$. Statement 2 : There exists a bounded sequence of real numbers with no convergent subsequence. ['True, True', 'True, False', 'False, True', 'False, False']	
LLM Reasoning Step	Cognitive Label
Okay, so I have these two statements to evaluate. Let me start with the first one.	Interpretation
Statement 1 says that if you have two sequences of complex numbers, $\{s_n\}$ and $\{t_n\}$, and both converge to s and t respectively, then the limit of their product $s_n t_n$ as n approaches infinity should be $s \cdot t$. Hmm, I remember from real analysis that the limit of a product is the product of the limits, provided both limits exist. Since complex numbers are a field extension of the reals, I think the same property should hold here. Let me verify this.	Logical Deduction
Suppose $\lim s_n = s$ and $\lim t_n = t$. Then, for the product, we can use the property that the limit of the product is the product of the limits. So, yes, this should be true. The complex numbers, like the real numbers, are a topological ring, so multiplication is continuous. Therefore, the limit of the product should indeed be the product of the limits. So Statement 1 is True.	Logical Deduction
Now moving on to Statement 2: "There exists a bounded sequence of real numbers with no convergent subsequence." Hmm, this seems a bit tricky. Wait, in real analysis, there's the Bolzano-Weierstrass theorem, which states that every bounded sequence of real numbers has a convergent subsequence. So if that's the case, then Statement 2 would be False because it's claiming the opposite.	Logical Deduction
But let me think again. The Bolzano-Weierstrass theorem does state exactly that: every bounded sequence in $\mathbb{R}^n$ has a convergent subsequence. Since we're dealing with real numbers ($n = 1$ here), any bounded sequence must have a convergent subsequence. Therefore, the statement that there exists a bounded sequence with no convergent subsequence is false.	Logical Deduction
Wait, but could there be some exception? For example, if the sequence is constant? No, a constant sequence would converge to that constant, so it definitely has a convergent subsequence. What if the sequence oscillates but remains bounded? Like $\sin(n)$ or something? Wait, $\sin(n)$ is bounded between -1 and 1 . Does $\sin(n)$ have a convergent subsequence? Well, yes. Because even though the sequence itself might not converge, any bounded sequence in $\mathbb{R}$ has a convergent subsequence by Bolzano-Weierstrass. For example, $\sin(n)$ would have a subsequence converging to 1 , or -1 , or some value in between. So even if the whole sequence doesn't converge, there is a subsequence that does.	Logical Deduction
Therefore, Statement 2 is false.	Conclusion
So putting it together, Statement 1 is True, Statement 2 is False. Therefore the answer should be "True, False".	Conclusion
Wait, hold on. Let me check Statement 2 again. Maybe I misread it. It says "a bounded sequence of real numbers with no convergent subsequence". But according to Bolzano-Weierstrass, that can't exist. So yeah, Statement 2 is false.	Verification
Is there any way Statement 2 could be true? Maybe in some other space? But the question specifies real numbers. In real numbers, bounded sequences are precisely those that have convergent subsequences. So no. Therefore, Statement 2 must be false.	Logical Deduction
Therefore the correct pair is True, False. So the second option in the choices given.	Conclusion

D Directed Graphs

D.1 Importance of the θ parameter

In this section, we analyze the effect of the threshold parameter θ on the readability of the cognitive flow graphs. As shown in Figure 7, a lower value of θ includes too many transitions, resulting in a cluttered graph, while a higher value provides a clearer view of the main cognitive processes.

D.2 Examples of Cognitive Flow Graphs

In this section, in Figure 8, we display cognitive flow graphs for some models on the Elementary Maths subset.

E Cognitive State Distributions

Quantitative analysis of the distribution of cognitive states offers a complementary, static view of a model’s reasoning patterns. By aggregating state usage across all tasks, we can measure overall prevalence and computational effort put towards each cognitive activity. This allows us to move beyond the flow of reasoning and look at its composition.

E.1 Label Percentage

We first take a look at the frequency of each cognitive state by calculating the percentage of total reasoning steps assigned to each label, measuring which cognitive operations models utilize more often.

As shown in Figure 9, the distribution of cognitive actions varies significantly across models and complexities. For example, on college-level tasks, the teacher model allocates a substantial portion of its steps to *Logical Deduction*, whereas the student models show a higher frequency of *Calculation*. This suggests that different models may prioritize different cognitive actions when facing problems of varying difficulty, even when their final answers are the same.

E.2 Token Percentage

To measure the computational effort dedicated to each cognitive state, we analyze the distribution of generated tokens. Instead of frequency, this metric quantifies the verbosity and detail with which a model executes a particular action. A state may be used infrequently (low label percentage) but require significant effort when it appears (high token percentage). Tokenization for all text was done using the bert-base-uncased⁹ tokenizer.

Analysis of Figure 10 shows crucial differences in computational effort. A key finding is the consistent low allocation of tokens to *Verification* by the distilled models, especially on Elementary level tasks. The teacher and control models dedicate significant computational effort to verifying their steps, while the distilled students appear to neglect this critical action. This significant difference provides strong quantitative evidence that distillation may fail to transfer not just the reasoning path, but also the underlying cognitive discipline required for robust problem-solving.

⁹<https://huggingface.co/google-bert/bert-base-uncased>

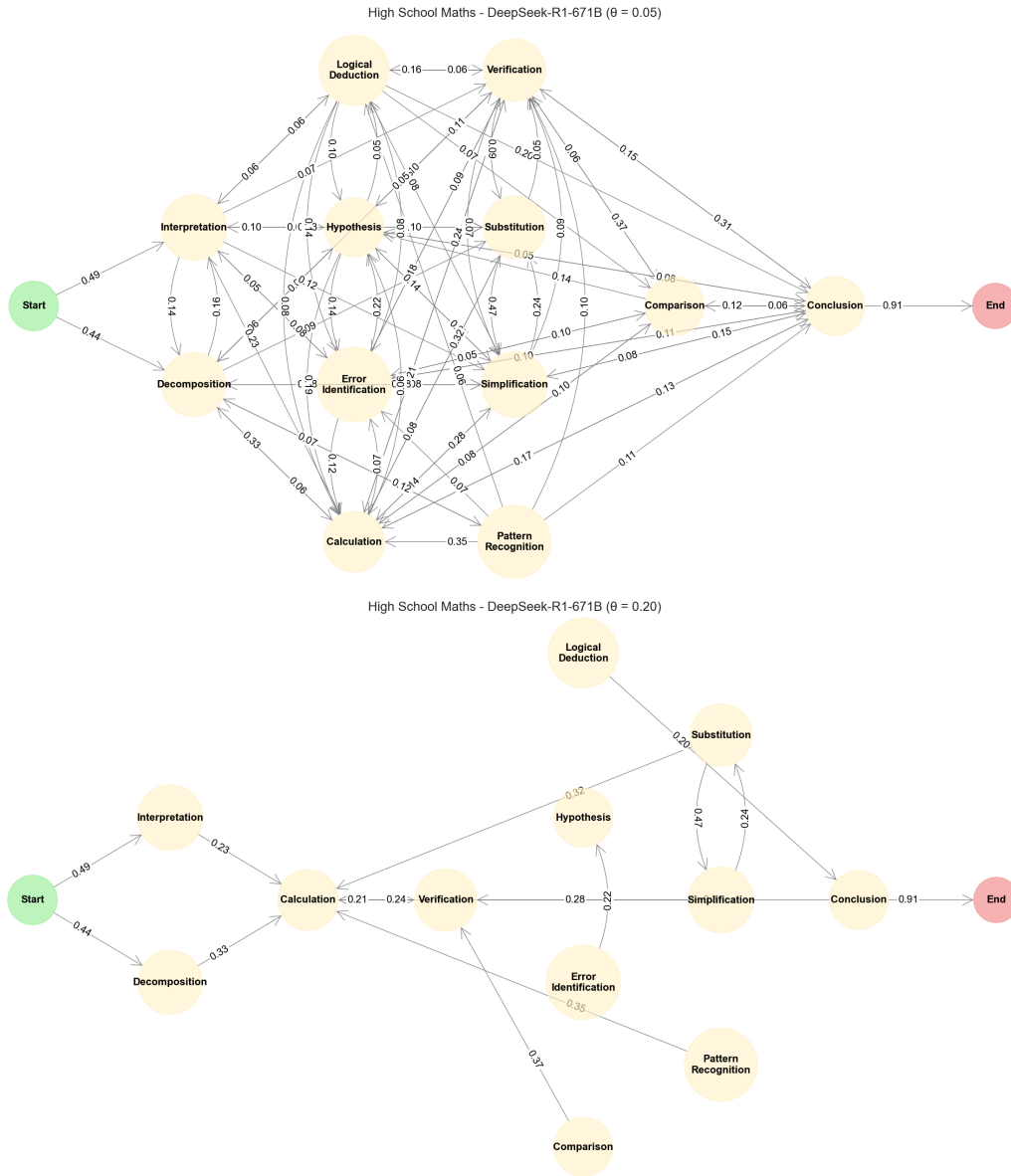


Figure 7: **Directed Graphs of the Cognitive Flow for DeepSeek-R1-671B on the High School Maths subset.** The top graph is constructed with a θ of 0.05, resulting in a cluttered visualization. The bottom graph has a θ of 0.2, providing a clear way to interpret the model's main cognitive processes.

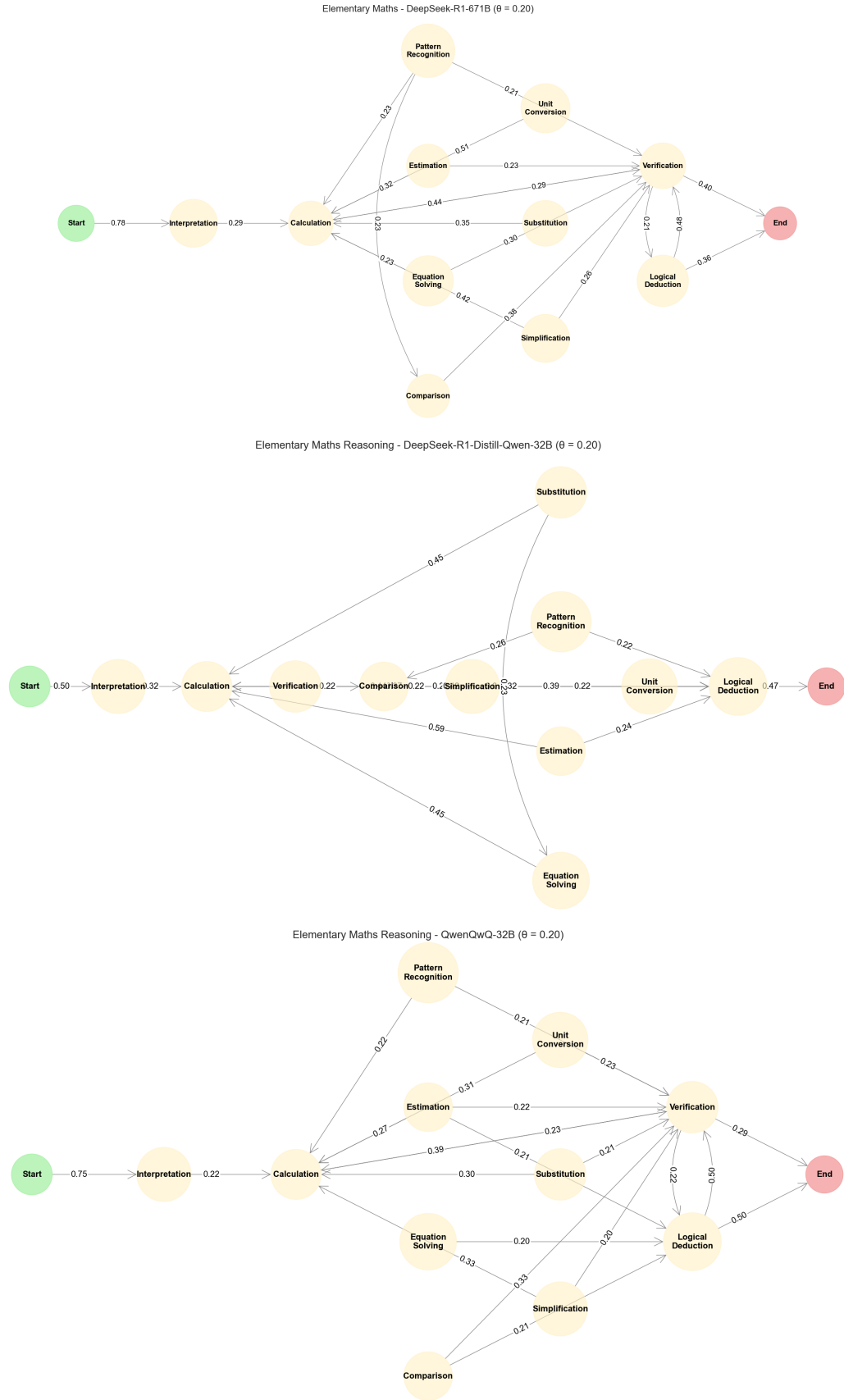


Figure 8: Directed Graphs of the Cognitive Flow for DeepSeek-R1-671B, its 32B Distilled model, and QwenQwQ-32B on the Elementary Maths subset.

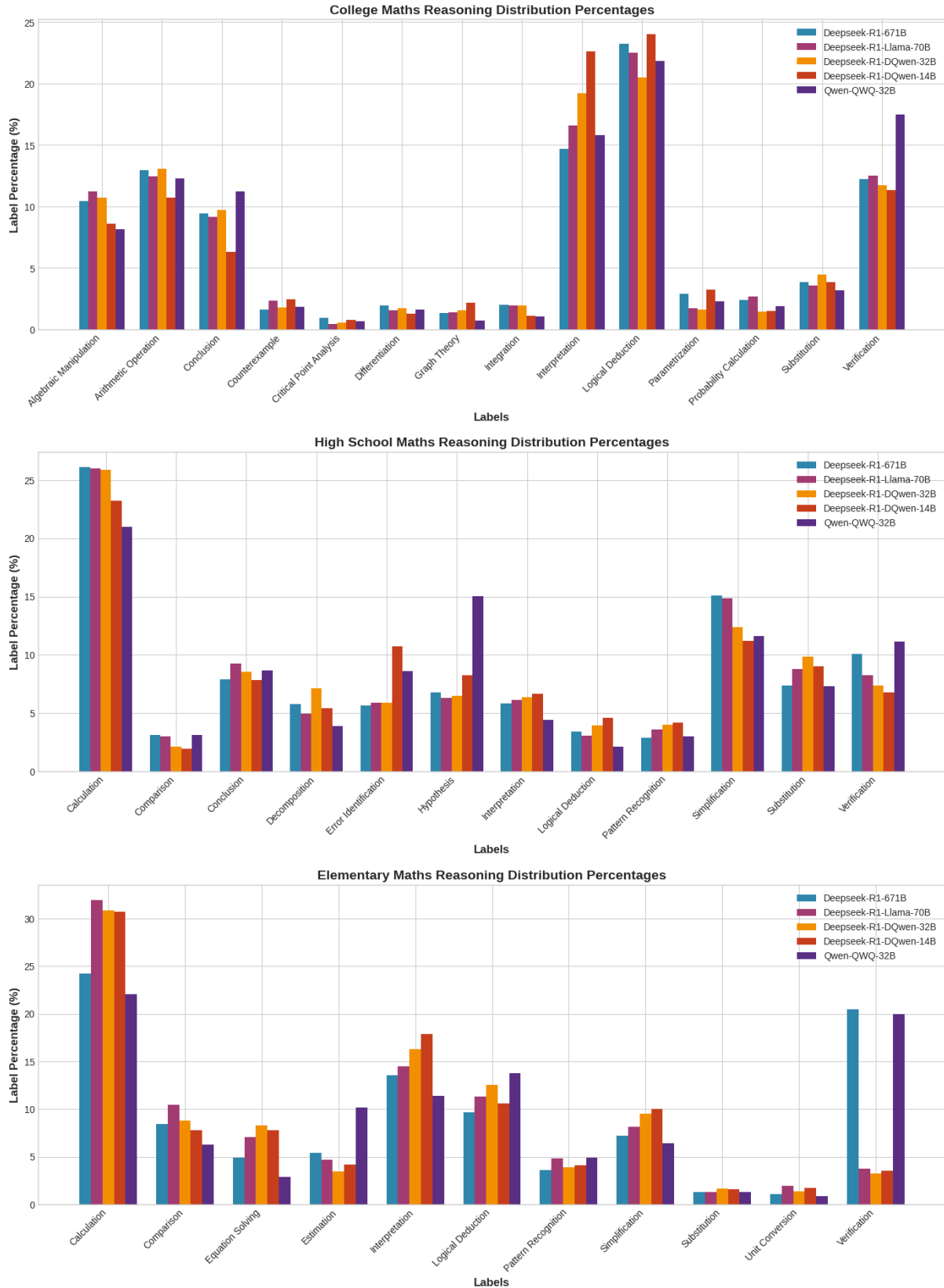


Figure 9: **Cognitive State Distribution by Frequency (Label Percentage).** Across the three complexities, each bar corresponds to a model’s percentage of reasoning steps assigned to each cognitive label. It indicates the frequency with which models utilize different cognitive actions in their reasoning.

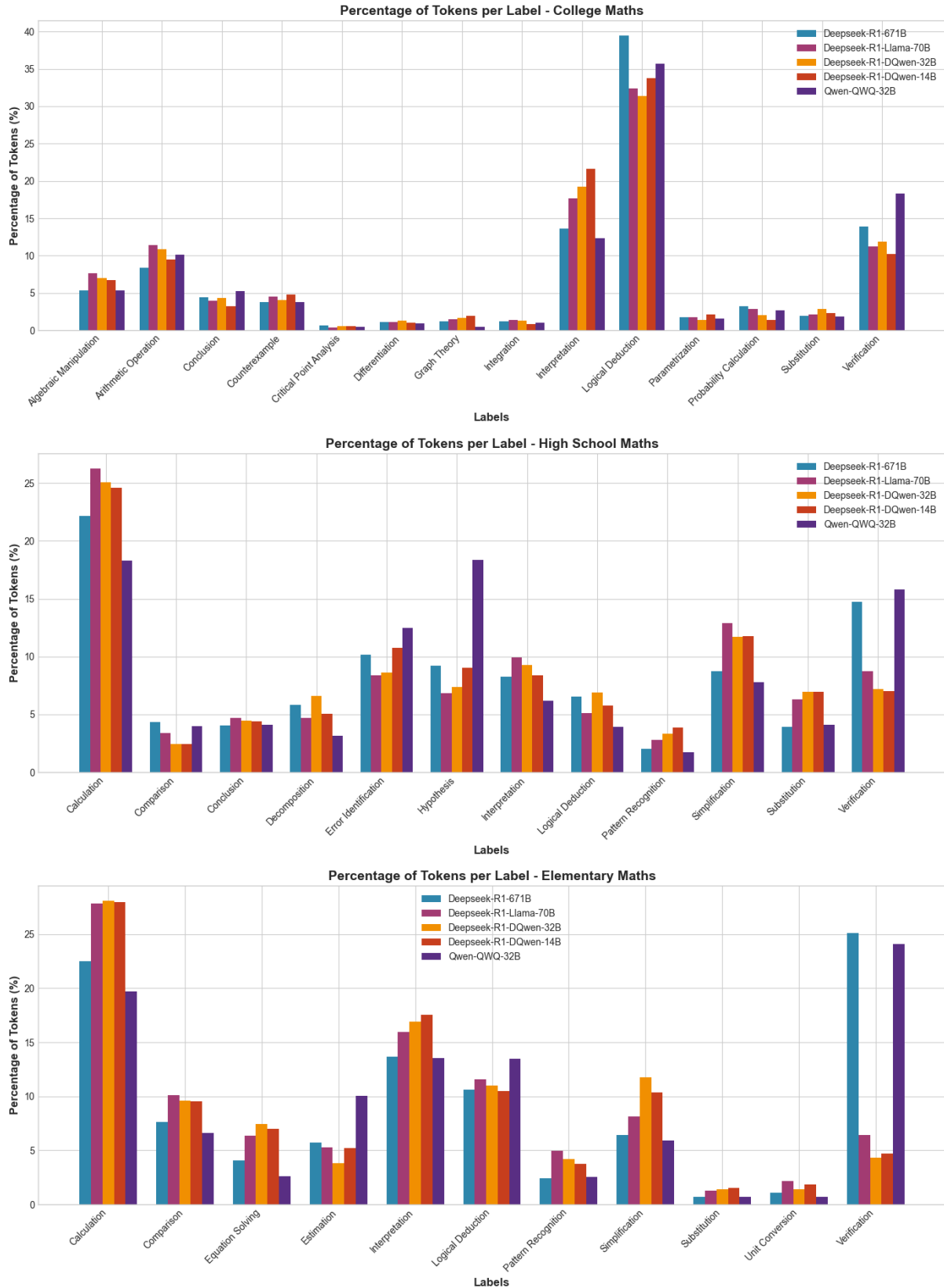


Figure 10: **Cognitive State Distribution by Computational Effort (Token Percentage).** Across the three complexities, each bar corresponds to the total number of tokens generated by a model for each cognitive state. It indicates the computational effort that the models put towards each cognitive action in their reasoning.