

DynaGuide: Efficient Reasoning Capability Distillation via Adaptive Data Curation and Overthinking Mitigation

Anonymous ACL submission

Abstract

The exponentially increasing computational demands of large language models (LLMs) facilitate the distillation to small models. Existing distillation attempts to transfer LLMs’ reasoning capabilities to compact models face critical limitations: expensive training or annotation cost, suboptimal data selection, and flawed synthetic data due to LLMs’ general overthinking behaviors. This paper introduces DynaGuide, a novel framework that optimizes the distillation process in both efficiency and performance. Our approach integrates (1) Dynamic Data Selection that adaptively performs fine-grained valuable data selection during the training process, and (2) Reasoning Pattern Guidance that mitigates the overthinking problem in synthetic data by incorporating specialized guidance during fine-tuning. Extensive experiments demonstrate that DynaGuide enables a 7B parameter model to achieve superior performance on knowledge reasoning question answering benchmarks, even achieving or exceeding its 32B counterpart. Our systematic ablation studies and analysis further reveal insights into distillation and reasoning.

1 Introduction

The rapid evolution of artificial intelligence has witnessed a dramatic surge in model complexity, progressing from early small models to today’s large language models (LLMs) that exhibit remarkable generative and reasoning capabilities. However, this advancement comes at an exponential increase in training costs, creating significant computational and financial barriers (Cottier et al., 2024). Although knowledge distillation is thought as a promising solution to this challenge by transferring LLMs’ excellent reasoning capabilities to more compact and efficient models using LLM-generated synthetic data (Xu et al., 2024b), distillation based on large datasets (DeepSeek-AI, 2025; Yu et al., 2025) remains computationally intensive

and time-consuming. Alternative approaches explore to use only a small amount of data for distillation, but introduce costly human experts annotations (Ye et al., 2025), or adopt coarse-grained data selection and ignore the adaptability to the model (Team, 2025; Muennighoff et al., 2025).

Moreover, recent studies have found that reasoning LLMs generally suffer from overthinking (Chen et al., 2024). Such models can get the correct answer at early reasoning stages (Fu et al., 2024), but continue the thinking process with much verification of previous steps or exploration of other unnecessary reasoning paths (Chen et al., 2025), generating redundant thinking tokens and reducing inference efficiency (Sui et al., 2025). Even worse, frequent verification and transition can disrupt reasoning continuity, degrade contextual coherence, reduce reasoning depth, and ultimately result in lower performance (Wang et al., 2025). When such flawed synthetic data is used for distillation, it can be more difficult for small models to acquire robust knowledge reasoning ability, thus more challenging to maintain efficiency and accuracy.

To address these limitations, we propose **DynaGuide**, a novel distillation framework that efficiently transfers the knowledge reasoning capability of LLMs to small models. DynaGuide includes two key components: **Dynamic Data Selection (DDS)** and **Reasoning Pattern Guidance (RPG)**. DDS performs adaptive data selection during the training process, similar to the idea of active learning (Cohn et al., 1996), where a small number of the most valuable samples are selected for training. Differently, we have access to the metadata (such as domains) of all data and the reasoning trace given by LLM (specifically DeepSeek-R1 (Guo et al., 2025) in our experiments), so we can leverage more comprehensive information for fine-grained data selection. RPG addresses the overthinking problem by incorporating additional guidance during distillation, derived from our systematic analysis

of reasoning patterns in knowledge QA tasks. Together, these two components enable more efficient and adaptive distillation while improving the distilled model’s reasoning capability.

In summary, our work makes the following contributions: (1) We propose Dynamic Data Selection during fine-tuning to better and more efficiently transfer the advanced reasoning ability of LLMs to small models through distillation. (2) We explore the reasoning patterns in knowledge QA and incorporate Reasoning Pattern Guidance into the fine-tuning process to mitigate overthinking and encourage the distilled model to think efficiently and correctly. (3) Comprehensive experiments demonstrate the effectiveness of our framework. Notably, our fine-tuned 7B model can achieve or even exceed the performance of its 32B counterpart. We further provide a systematic analysis of its generalization capability and extensive ablation studies.

2 Related Works

2.1 Distillation of Large Language Models

Knowledge Distillation has emerged as a promising approach to transfer the advanced capabilities of LLMs to compact open-source models (Xu et al., 2024b). Early exploration focused on learning specific knowledge from LLMs (Ding et al., 2023), while recent studies attempt to transfer the advanced reasoning capability to small models (Hsieh et al., 2023; Sun et al., 2025), particularly in mathematical and programming domains (Xu et al., 2025; Team, 2025; Labs, 2025). However, the distillation of reasoning-based knowledge QA tasks remains relatively underexplored. Current approaches also exhibit limitations in data curation, including dependence on large-scale datasets (DeepSeek-AI, 2025), reliance on coarse-grained data selection (Muennighoff et al., 2025), and the necessity for costly human expert annotations (Yu et al., 2025). Therefore, our work investigates data-efficient distillation through fine-grained data selection in knowledge QA tasks.

2.2 Knowledge QA

As LLM continues to evolve, performance on QA tasks gradually improves, but problems such as hallucinations still exist (Huang et al., 2023; Jiang et al., 2024; Luo et al., 2024). Retrieval-augmented generation (RAG) can be helpful by introducing external knowledge into the context or training objectives (Gao et al., 2023; Asai et al., 2023; Tu

Algorithm 1: Dynamic Data Selection during Fine-Tuning

Input : Training Data Pool \mathcal{D} , Model θ_0 ,
Amount of training data n ,
Amount of Warm-up Data n_w ,
Batch Size n_b

Output : Fine-Tuned model θ

```

1 Initialize Uniform distribution  $\mathcal{W}_0$  across
  all types of data
  // Warm-up
2  $\mathcal{D}_{\text{train}} \leftarrow \text{SAMPLE}(\mathcal{D}, \mathcal{W}_0, n_w)$ 
3  $\mathcal{D} \leftarrow \mathcal{D} \setminus \mathcal{D}_{\text{train}}$ 
4  $\theta_1, \text{losses} \leftarrow \text{TRAIN}(\theta_0, \mathcal{D}_{\text{train}})$ 
5  $\mathcal{W}_1 \leftarrow \text{UPDATEWEIGHTS}(\mathcal{W}_0, \mathcal{D}_{\text{train}}, \text{losses})$ 
  // Dynamic Data Selection
6  $i \leftarrow 1$ 
7 while  $|\mathcal{D}_{\text{train}}| < n$  do
8    $\mathcal{D}_{\text{batch}} \leftarrow \text{SAMPLE}(\mathcal{D}, \mathcal{W}_i, n_b)$ 
9    $\mathcal{D} \leftarrow \mathcal{D} \setminus \mathcal{D}_{\text{batch}}$ 
10   $\theta_{i+1}, \text{losses} \leftarrow \text{TRAIN}(\theta_i, \mathcal{D}_{\text{batch}})$ 
11   $\mathcal{W}_{i+1} \leftarrow \text{UPDATEWEIGHTS}(\mathcal{W}_i, \mathcal{D}_{\text{batch}}, \text{losses})$ 
12   $i \leftarrow i + 1$ 
13   $\mathcal{D}_{\text{train}} \leftarrow \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{batch}}$ 
14 return  $\theta_i$ 

```

et al., 2025). Knowledge-based QA is well suited for testing the model’s reasoning ability, as it is a challenging task to reconcile multiple knowledge and reason between input texts (Yang et al., 2018; Geva et al., 2021). Previous works have similarly proposed training to improve reasoning ability on knowledge QA tasks, but require large amounts of labeled or generated data (Xu et al., 2024a; Lyu et al., 2024). Our work focuses on data selection to achieve optimal results with a small amount of data and to maintain the model’s ability to generalize, improving model’s reasoning ability in both in-domain and out-of-domain knowledge QA tasks.

3 Dynamic Data Selection during Fine-Tuning

In this section, we present our dynamic data selection framework for fine-tuning. Our fundamental premise is that distinct data characteristics result in divergent learning dynamics during the fine-tuning process. Certain domains or complexity levels require much exposure for adaptation, while others stimulate the model’s capabilities through few ap-

pearances. In this paper, we characterize the data from two orthogonal dimensions: (i) domain specificity and (ii) task complexity.

Our dynamic data selection methodology, formalized in Algorithm 1, operates on the principle of continuous weight adaptation during fine-tuning. The framework starts with a warm-up phase and maintains a dynamic weight distribution across data feature classes, which it uses to probabilistically sample each subsequent training batch. This adaptive approach enables the model to automatically prioritize data features that require more attention while maintaining exposure to all the classes.

3.1 Warm-Up

The cold start problem poses a significant challenge that purely dynamic data selection may lead to insufficient model understanding of the overall data distribution. Without proper initialization, the weights assigned to initially selected data types could progressively increase, creating a self-reinforcing cycle where these data types continue to be preferentially selected. This phenomenon may result in the neglect of other data feature classes, ultimately reducing training data diversity and compromising the model’s generalization capability.

To address this issue, we introduce a warm-up phase prior to dynamic data selection. During this phase, we construct a balanced warm-up dataset by uniformly sampling equal amounts of data from all the classes. This warm-up dataset constitutes 4% of the total selected data, serving to establish a more representative initial data distribution before transitioning to dynamic selection.

3.2 Dynamic Selection

Since performing inference on the entire training data pool to identify samples with the highest model uncertainty is computationally prohibitive, our approach dynamically adjusts the weights of data feature classes that result in higher or lower loss in the currently observed batch. This strategy aims to prioritize the selection of such informative samples in subsequent training iterations.

Specifically, our approach calculates the ratio of each sample’s loss to the batch’s average loss during training. Subsequently, conditioned on this ratio, we implement weight adjustments: for samples with a ratio below a lower threshold t_l , we downweight the type to which the sample belongs; conversely, for samples with a ratio above an upper threshold t_u , we upweight the corresponding type.

These thresholds act as a margin to explicitly separate samples the model finds easy (low loss) from those it finds difficult (high loss), thereby stabilizing the weighting mechanism.

We compute a weight adjustment factor f based on the loss ratio. The underlying principle is to assign larger weight increments to types with higher loss ratio values and larger reductions to those with lower loss ratio values. To mitigate weight explosion or weight disappearance, we require the weight growth rate to be sublinear with respect to the loss ratio. Thus, we adopt a simple rational function with a lower limit for smoother scaling:

$$f = \max\left(\frac{1}{2}, \frac{2r}{r+1}\right), \text{ where } r = \frac{\ell_i}{\bar{\ell}_{\text{batch}}}. \quad (1)$$

Here, ℓ_i denotes the per-sample loss and $\bar{\ell}_{\text{batch}}$ represents the batch-averaged loss. Such f ensures monotonic yet controlled adjustments, approaching 2 for large r and 0.5 for small r , reducing the sensitivity to extreme values.

Upon selecting a predefined number of instances, we terminate the dynamic data selection process. To demonstrate the data efficiency of our method and facilitate a fair comparison with prior work (Muennighoff et al., 2025), we limit the total selected data to 1,000 samples. The fine-tuning procedure consists of 5 epochs, with the dynamic data selection performed exclusively during the first epoch. After that, we train the model on the selected subset for another 4 epochs. Such a procedure ensures consistent evaluation conditions and maintains computational efficiency.

4 Incorporate Control of thinking

To systematically analyze the reasoning patterns in knowledge-based question answering tasks, we follow the definition of Chen et al. (2025) to segment reasoning traces into discrete steps using double newline delimiters (`\n\n`) and categorize these steps into three distinct types: execution, reflection, and transition. Execution steps perform factual retrieval or concrete computation, reflection steps verify the previous steps, and transition steps bridge two different reasoning paths.

4.1 Reasoning Patterns Analysis

First, we explore the model’s reasoning patterns in knowledge QA tasks, with particular attention to the correlation between step-type frequencies and task performance metrics. Table 1 presents

Metric	Answer Type	
	Correct	Wrong
Average # Tokens	1804.54	1823.11
Execution Steps	73.10%	58.40%
Reflection Steps	17.13%	23.30%
Transition Steps	9.77%	18.30%

Table 1: Analysis of DeepSeek-R1’s reasoning patterns in knowledge question answering tasks.

an analysis of DeepSeek-R1’s chains of thought on knowledge question answering tasks (on strategyQA (Geva et al., 2021), hotpotQA (Yang et al., 2018) and superGPQA (Du et al., 2025) datasets). We systematically examined the model’s performance by quantifying the average token length of reasoning chains, and the relative frequency distribution of different reasoning step types across both correct and incorrect responses.

Different from the findings of previous work (Chen et al., 2025) in the field of mathematical tasks, we find that in the field of knowledge QA tasks, there is no significant difference in the number of model’s thinking tokens when answering correctly and incorrectly. However, our analysis reveals distinct patterns in reasoning step type distributions between correct and incorrect responses. For erroneous answers, we observe a statistically significant decrease in execution-type steps, accompanied by a marked increase in other step types, particularly transition steps. This inverse relationship suggests that excessive reflection and transition steps may disrupt the model’s reasoning process, potentially leading to performance degradation. Specifically, the disproportionate growth in meta-cognitive steps appears to compromise the model’s ability to maintain focused reasoning.

4.2 Reasoning Pattern Guidance

We further perform encoding on the training set to extract the hidden states of the tokens containing ‘\n\n’ as feature vectors representing subsequent thinking steps. Our analysis reveals that vectors from deeper model layers, especially the 20th layer, exhibit weak separability when projected into 2D space (detailed visualizations are provided in Appendix C). This observation suggests stronger separability in the original high-dimensional hidden space. This discovery is consistent with the conclusions of Chen et al. (2025) on mathematical tasks.

To guide the model’s reasoning process during fine-tuning and encourage more execution steps,

we propose adjusting the hidden states of tokens containing ‘\n\n’ toward the direction of the average execution-step feature vector, denoted as H_E . However, since H_E may evolve during training, we employ an iterative tuning approach to align the representations with the target reasoning trajectory.

Specifically, after epochs 1-4, we perform encoding on the selected training dataset and calculate the average execution-step feature vector H_E^i , where i denotes the epoch number. During epochs 2-5, we introduce a projection loss for all tokens containing ‘\n\n’. Let H_c^i represent the hidden state of such a token at epoch i , and let Sim represents cosine similarity. The loss encourages alignment between H_c^i and the average execution-step feature vector H_E^{i-1} from the previous epoch.

If the subsequent step is a reflection step or a transition step, we apply the projection loss:

$$\mathcal{L}_{\text{proj}} = \frac{(1 - \text{Sim}(H_c^i, H_E^{i-1}))}{2}. \quad (2)$$

The loss ‘pushes’ H_c^i toward H_E^{i-1} , thereby encouraging the model to generate more execution steps. If the subsequent step is already an execution step, we still apply the same projection loss to minimize semantic drift in the learned representations. This ensures consistent optimization across all reasoning step types, resulting in a unified projection loss.

We jointly optimize both the causal language modeling loss and the projection loss during training. The total loss function is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CLM}} + \alpha \cdot \mathcal{L}_{\text{proj}}, \quad (3)$$

where \mathcal{L}_{CLM} denotes the standard causal language modeling loss, $\mathcal{L}_{\text{proj}}$ is our proposed projection loss that enforces latent space alignment constraints, and α serves as a balancing hyperparameter. This configuration maintains equilibrium between the two objectives and prevents significant deviation from the original pre-trained model’s hidden space.

We apply the projection loss to the 20th layer of the model, as this layer demonstrates the strongest feature vector separability in our analysis. This choice aligns with the experimental setup of Chen et al. (2025). While our study employs Qwen2.5-7B-Instruct and theirs uses R1-Distill-Qwen-1.5B, both models share a similar architecture of 28 layers, ensuring comparable layer-wise behavior.

Model	Fine-Tuning Data	In-domain Datasets			Out-of-domain Datasets			
		StrategyQA test set	HotpotQA dev	HotpotQA dev ^c	FRAMES ^c	GPQA extended	SimpleQA	SimpleQA ^c
R1-Distill-Qwen-7B	800K	70.16	19.27	54.80	22.33	46.34*	2.17	46.44
R1-Distill-LLaMA-8B	800K	71.03	20.20	55.24	23.18	50.73*	2.36	48.94
Qwen2.5-7B-Instruct	-	68.41	15.87	46.91	20.63	31.87	2.47	47.55
s1-7B	1,000	74.09	20.77	57.92	23.54	39.74	3.05	46.51
s1-7B + BF	1,000	74.53	21.36	59.43	25.24	44.51*	3.33	49.40
DynaGuide-7B (ours)	1,000	77.73*	24.98	61.59	28.52*	47.07*	3.98	54.60*
Qwen2.5-32B-Instruct	-	75.98	27.94	65.47	27.55	42.31	4.90	53.42

Table 2: Model accuracy on various benchmarks. ^cQuestion answering tasks with retrieved context (i.e., RAG evaluation) requiring contextual reasoning. **Bold** formatting highlights the top-performing model within the 7B-8B parameter cohort. *Asterisked results demonstrate performance surpasses that of Qwen2.5-32B-Instruct, while utilizing approximately 75% fewer parameters.

5 Experiments

5.1 Setup

Datasets Our training dataset consists of the training set of StrategyQA (Geva et al., 2021), the training set of HotpotQA (Yang et al., 2018), and SuperGPQA (Du et al., 2025), aggregating to 118,579 samples that span diverse domains and complexity levels. Then, similar to s1 (Muennighoff et al., 2025), we apply a quality filter retaining only questions that neither Qwen2.5-7B-Instruct nor Qwen2.5-32B-Instruct (Team, 2024) can answer correctly. This filtering process results in a refined training data pool of 71,662 examples. Then we request the DeepSeek-R1’s inference API to generate reasoning traces and answers, which serve as pseudo-annotations for each question. This dataset forms the training data pool for our dynamic data selection framework.

Our methodology focuses specifically on transferring knowledge reasoning capabilities from Reasoning LLMs to small models, rather than context retrieval performance. To maintain this focus, we omit external context in our training examples, requiring the model to rely exclusively on its internal knowledge for reasoning. This design ensures the fine-tuning process specifically enhances the model’s inherent reasoning abilities without confounding factors from retrieval augmentation.

Implementation Details We conduct full-parameter supervised fine-tuning (SFT) of the Qwen2.5-7B-Instruct model (Team, 2024) using a two-phase training approach: (i) During the initial epoch, we perform dynamic data selection

described in Section 3 until accumulating a curated set of 1,000 training examples; (ii) For the subsequent four epochs, we train exclusively on the selected subset while incorporating our proposed reasoning pattern guidance (RPG) framework, described in Section 4.2. In our experiment, we set the thresholds $t_l = 0.9$ and $t_u = 1.1$, and the loss weighting hyperparameter $\alpha = 1.0$. More experimental details can be found in Appendix A.

Evaluation Our evaluation protocol includes both in-domain and out-of-domain datasets. For in-domain evaluation, we assess model performance on the StrategyQA test set and the HotpotQA development set (since the answers of the HotpotQA test set are unavailable). For out-of-domain evaluation, we test the model on some challenging benchmarks: FRAMES (Krishna et al., 2024), GPQA extended set (Rein et al., 2024), and SimpleQA (Wei et al., 2024). All results report the accuracy rate.

Baselines We compare our framework with: (1) **Qwen2.5-7B-Instruct** (Team, 2024), the foundation model prior to our fine-tuning; (2) **R1-Distill-Qwen-7B** and **R1-Distill-LLaMA-8B** (DeepSeek-AI, 2025): models distilled on 800K data from DeepSeek-R1 based on Qwen and Llama, released by DeepSeek-AI; (3) **s1-7B** and **s1-7B + BF**: a model fine-tuned from Qwen2.5-7B-Instruct following the framework of Muennighoff et al. (2025) but on knowledge reasoning tasks, where BF denotes their proposed test-time scaling technique Budget Forcing; and (4) **Qwen2.5-32B-Instruct** (Team, 2024): a model with approximately 4x parameters for cross-scale comparison.

Data Selection Method	In-domain Datasets			Out-of-domain Datasets			
	StrategyQA test set	HotpotQA dev	HotpotQA dev ^c	FRAMES ^c	GPQA extended	SimpleQA	SimpleQA ^c
Random	73.95	21.07	60.03	21.84	37.73	2.80	48.20
Longest	69.29	18.16	56.88	24.52	46.15	2.89	48.66
s1	74.09	20.77	57.92	23.54	39.74	3.05	46.51
w/o Warm-up	76.61 ± 1.40	22.75 ± 1.45	59.88 ± 1.89	23.14 ± 2.08	41.15 ± 2.49	2.96 ± 0.14	48.55 ± 0.47
DDS (ours)	77.29 ± 1.03	23.92 ± 0.70	60.18 ± 1.24	24.60 ± 0.97	43.28 ± 1.56	3.28 ± 0.17	49.47 ± 0.57

Table 3: Performance comparison of different data selection methods. ^cQuestion answering tasks with retrieved context (i.e., RAG evaluation) requiring contextual reasoning. **Bold** formatting highlights the top-performing model. All results report task accuracy. We also report the radius of the 95% confidence interval in the ablation study of the warm-up phase. All models are trained without Reasoning Pattern Guidance.

5.2 Main Results

The main experimental results are shown in Table 2, which demonstrates our framework’s superior performance across various benchmarks. Our proposed distillation framework significantly enhances Qwen2.5-7B-Instruct model’s performance, demonstrating notable gains across all benchmarks. Our model also outperforms the R1-Distill models of similar size on most of the benchmarks, which are fine-tuned on 800K data. Our framework also has advantages over the s1 framework with the same data efficiency. Moreover, our fine-tuned 7B model can approach or even exceed the performance of Qwen2.5-32B-Instruct, which has approximately 4 times the number of parameters.

6 Analysis

6.1 Generalization to RAG Tasks

Since our model is fine-tuned without external context and the RAG task represents an important knowledge-based question answering scenario, we additionally evaluate our framework’s generalization capability to the RAG task on hotpotQA, FRAMES, and SimpleQA datasets.

We organize the external context input to the model as follows: (i) For HotpotQA dataset, the context is contained in the dataset file, and we concatenate them into several pieces of text in their original order; (ii) For the FRAMES and SimpleQA datasets, the data files contain context URLs. We crawl all Wikipedia URLs and delete irrelevant content such as navigation bars, sidebars, and hyperlinks, retaining the title and body, and then connect the texts to form the context according to the order of the URLs in the data file. Since the context

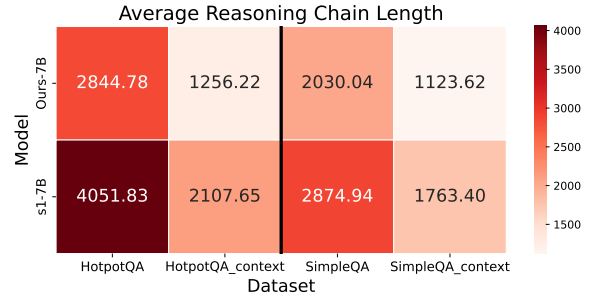


Figure 1: Comparison of reasoning chain length (number of tokens) between s1-7B and Ours-7B model under different input settings. “HotpotQA” and “SimpleQA” denote inputs without external context, while “HotpotQA_context” and “SimpleQA_context” represent RAG tasks with context.

length of the model is limited and some context windows need to be reserved for reasoning, we truncate all contexts and only keep the first 8K tokens as context input to the model.

The results in Table 2 shows that the model fine-tuned by our framework not only has a good ability to rely on internal knowledge for reasoning, but also has strong contextual reasoning capabilities.

Furthermore, we evaluate the models’ reasoning efficiency on RAG tasks. As illustrated in Figure 1, the introduction of external contextual information significantly reduces required reasoning steps by providing supplementary evidence. Compared to scenarios without external context, our 7B model reduces the average number of reasoning tokens by 50.25%, while while s1-7B achieves a reduction of only 43.32%, indicating that our model can perform contextual reasoning more efficiently.

Method	In-domain Datasets			Out-of-domain Datasets			
	StrategyQA test set	HotpotQA dev	HotpotQA dev ^c	FRAMES ^c	GPQA extended	SimpleQA	SimpleQA ^c
Vanilla SFT	77.29 (861.89)	23.92 (2639.62)	60.18 (1236.33)	24.60 (2036.92)	43.28 (23381.84)	3.28 (1644.97)	49.47 (1068.48)
Exe-only	75.25 (793.64)	22.32 (2341.80)	60.04 (1260.39)	24.03 (1779.49)	43.22 (20991.40)	2.96 (1722.02)	50.18 (763.99)
SEAL	76.86 (605.44)	24.51 (1067.05)	60.28 (730.00)	26.82 (1581.05)	45.79 (10652.20)	3.65 (1033.96)	51.69 (682.63)
RPG (ours)	77.73 (788.71)	24.98 (1991.35)	61.59 (990.71)	28.52 (1511.84)	47.07 (16723.81)	3.98 (1421.03)	54.60 (786.53)

Table 4: Performance and average thinking token number comparison of reasoning control methods on question-answering tasks. The average number of thinking tokens is shown in parentheses. ^cQuestion answering tasks with retrieved context (i.e., RAG evaluation) requiring contextual reasoning. **Bold** formatting highlights the top-performing model. All results report task accuracy. All models are trained with Dynamic Data Selection.

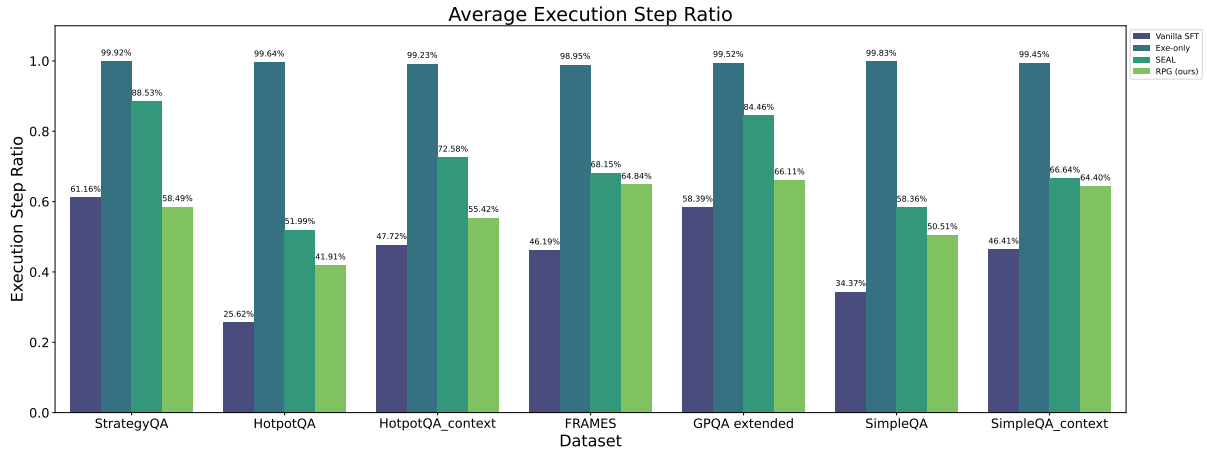


Figure 2: Comparison of the proportion of execution steps of different methods on various datasets. “HotpotQA” and “SimpleQA” denote inputs without external context, while “HotpotQA_context” and “SimpleQA_context” represent RAG tasks with external context input.

6.2 Ablation Study of Dynamic Data Selection

To evaluate the performance improvement achieved by our proposed dynamic data selection framework, we compared it with several baseline data selection methods: (1) **Random**: Uniform random sampling of 1,000 instances; (2) **Longest**: Selecting samples with the longest pseudo-annotation reasoning chains; (3) **s1**: the data selection method described by Muennighoff et al. (2025), which comprise two phases - initial uniform sampling of 300 instances, followed by repeatedly difficulty-weighted data selection in a randomly chosen domain, until reaching 1,000 instances. We also conduct an ablation study to show the necessity of the warm-up phase.

As shown in Table 3, our proposed data selection method achieves significant performance gains across multiple tasks, while baseline methods suf-

fer from uneven data distribution, especially the Longest baseline. Since most training data with the longest reasoning chain originates from reasoning traces on the SuperGPQA dataset, the most difficult training set, this baseline yields huge improvements on the similar GPQA Benchmark, but shows minimal gains on other datasets. Furthermore, incorporating the warm-up phase not only enhances overall performance, but also stabilizes training dynamics, reducing the result variance.

6.3 Ablation Study of Reasoning Pattern Guidance

We evaluate our proposed Reasoning Pattern Guidance framework against several baseline approaches that may facilitate the model’s execution step: (1) **Vanilla SFT**: The standard approach

Training Data Pool	Data Size	In-domain Datasets			Out-of-domain Datasets			
		StrategyQA test set	HotpotQA dev	HotpotQA dev ^c	FRAMES ^c	GPQA extended	SimpleQA	SimpleQA ^c
R1-correct	25725	78.31	23.59	62.04	25.36	45.24	3.77	51.62
R1-wrong	45937	75.40	23.36	59.22	23.30	38.28	2.61	48.75
All	71662	77.29	23.92	60.18	24.60	43.28	3.28	49.47

Table 5: Performance comparison of models trained on different training data pools. ^cQuestion answering tasks with retrieved context (i.e., RAG evaluation) requiring contextual reasoning. All models are trained with Dynamic Data Selection and without Reasoning Pattern Guidance.

where only supervised fine-tuning (SFT) is performed on the selected data, with no modifications during training or inference; (2) **Exe-only**: A simplified variant where we remove all reflection and transition steps from the training data, retaining only the execution steps; (3) **SEAL** (Chen et al., 2025): An approach that performs targeted modifications to the model’s hidden states during decoding to encourage more execution steps.

As demonstrated in Table 4, our method achieves the most significant performance gains over Vanilla SFT across all evaluated datasets. Notably, the Exe-only variant exhibits degraded performance, which we attribute to the removal of reflection and transition steps. This modification disrupts the coherence of reasoning chains, compromising both semantic integrity and contextual relevance.

We further analyzed the execution step ratios across different methods (Figure 2), revealing two key insights: (1) The Exe-only approach achieves near-complete execution dominance. However, this comes at the cost of semantic coherence, as the removal of reflection and transition steps leads to fragmented reasoning chains and compromised contextual relevance, ultimately impairing task performance. (2) While SEAL demonstrates higher execution rates than our RPG framework, this comes through forced conversion of reflection/transition steps into execution during decoding. In contrast, RPG maintains the model’s capacity for necessary reflection and transition while promoting execution during training, achieving superior overall performance through more balanced reasoning processes.

6.4 Impact of the Correctness of R1-Responses

Furthermore, we investigate a critical scientific question: *Does the small model primarily acquire factual knowledge through distillation, or does it mainly develop reasoning capabilities?* To exam-

ine this distinction, we conduct experiments using two distinct data pools respectively: (i) **R1-correct**: Samples where DeepSeek-R1 provides correct answers, containing accurate reasoning traces; (ii) **R1-wrong**: Samples where DeepSeek-R1 provides incorrect answers, representing cases where the reasoning traces contain erroneous knowledge. This design enables us to distinguish the model’s ability to learn reasoning patterns from its capacity to acquire factual knowledge through distillation.

As presented in Table 5, our experimental results reveal some key observations: First, models fine-tuned exclusively on incorrect reasoning chains (R1-wrong) still achieve competitive performance. Second, the performance gain from using exclusively correct chains (R1-correct) is marginal compared to training on the complete dataset. These experimental results strongly suggest that the model primarily acquires reasoning capabilities rather than merely memorizing factual knowledge during the distillation process.

7 Conclusion

In this work, we present DynaGuide, an innovative framework for efficiently distilling the reasoning capabilities of LLMs into more compact and deployable models. First, our proposed Dynamic Data Selection (DDS) provides better data curation than current distillation approaches. Second, the Reasoning Pattern Guidance (RPG) resolves the overthinking issue in LLM-generated synthetic data by optimizing the reasoning process during fine-tuning. Together, these components enable more data-efficient distillation while maintaining the reasoning quality of distilled models.

Furthermore, our extensive analysis of data selection and the model’s reasoning pattern provides valuable insights for future research, advancing the field of knowledge distillation of LLMs.

Limitations

There are some potential limitations to our current work, which we aim to overcome in our future work: (1) Our current framework for reasoning pattern analysis and guidance is developed based on the reasoning traces generated by DeepSeek-R1. Its generalizability to other teacher models remains an open question that requires further investigation. (2) Due to computational constraints, we do not evaluate our framework across different model scales, nor analyze data scaling effects. These further studies could yield valuable insights into the scaling law of distillation.

References

- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. [Self-rag: Learning to retrieve, generate, and critique through self-reflection](#). *ArXiv*, abs/2310.11511.
- Runjin Chen, Zhenyu Zhang, Junyuan Hong, Souvik Kundu, and Zhangyang Wang. 2025. Seal: Steerable reasoning calibration of large language models for free. *arXiv preprint arXiv:2504.07986*.
- Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024. Do not think that much for $2+3=?$ on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.
- David A Cohn, Zoubin Ghahramani, and Michael I Jordan. 1996. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145.
- Ben Cottier, Robi Rahman, Loredana Fattorini, Nestor Maslej, Tamay Besiroglu, and David Owen. 2024. The rising costs of training frontier ai models. *arXiv preprint arXiv:2405.21015*.
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.
- Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, and 1 others. 2025. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*.
- Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Aurick Qiao, and Hao Zhang. 2024. Efficiently serving llm reasoning programs with certindex. *arXiv preprint arXiv:2412.20993*.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. [Retrieval-augmented generation for large language models: A survey](#). *ArXiv*, abs/2312.10997.
- Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. [Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies](#). *Trans. Assoc. Comput. Linguistics*, 9:346–361.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander 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. *arXiv preprint arXiv:2305.02301*.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. [A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions](#). *ArXiv*, abs/2311.05232.
- Xuhui Jiang, Yuxing Tian, Fengrui Hua, Chengjin Xu, Yuanzhuo Wang, and Jian Guo. 2024. [A survey on large language model hallucination via a creativity perspective](#). *ArXiv*, abs/2402.06647.
- Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqi. 2024. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. *arXiv preprint arXiv:2409.12941*.
- Bespoke Labs. 2025. Bespoke-stratos: The unreasonable effectiveness of reasoning distillation. www.bespokelabs.ai/blog/bespoke-stratos-the-unreasonable-effectiveness-of-reasoning-distillation. Accessed: 2025-01-22.
- Wen Luo, Tianshu Shen, Wei Li, Guangyue Peng, Richeng Xuan, Houfeng Wang, and Xi Yang. 2024. [Halludial: A large-scale benchmark for automatic dialogue-level hallucination evaluation](#). *CoRR*, abs/2406.07070.
- Yongang Lyu, Lingyong Yan, Shuaiqiang Wang, Haibo Shi, Dawei Yin, Pengjie Ren, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. 2024. [Knowtuning: Knowledge-aware fine-tuning for large language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 14535–14556. Association for Computational Linguistics.

675	Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. <i>arXiv preprint arXiv:2501.19393</i> .	2024, Bangkok, Thailand, August 11-16, 2024, pages 133–145. Association for Computational Linguistics.	726 727
680	David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In <i>First Conference on Language Modeling</i> .	Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024b. A survey on knowledge distillation of large language models. <i>arXiv preprint arXiv:2402.13116</i> .	728 729 730 731 732
682	Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and 1 others. 2025. Stop overthinking: A survey on efficient reasoning for large language models. <i>arXiv preprint arXiv:2503.16419</i> .	Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In <i>Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018</i> , pages 2369–2380. Association for Computational Linguistics.	733 734 735 736 737 738 739 740
685	Lin Sun, Guangxiang Zhao, Xiaoqi Jian, Yuhang Wu, Weihong Lin, Yongfu Zhu, Linglin Zhang, Jinzhu Wu, Junfeng Ran, Sai-er Hu, and 1 others. 2025. Tinyr1-32b-preview: Boosting accuracy with branch-merge distillation. <i>arXiv preprint arXiv:2503.04872</i> .	Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. <i>arXiv preprint arXiv:2502.03387</i> .	741 742 743
691	NovaSky Team. 2025. Sky-t1: Train your own o1 preview model within 450. https://novasky-ai.github.io/posts/sky-t1 . Accessed: 2025-01-09.	Zhaojian Yu, Yinghao Wu, Yilun Zhao, Arman Cohan, and Xiao-Ping Zhang. 2025. Z1: Efficient test-time scaling with code. <i>arXiv preprint arXiv:2504.00810</i> .	744 745 746
696	Qwen Team. 2024. Qwen2.5 technical report. <i>arXiv preprint arXiv:2412.15115</i> .	A Experimental Details	747
697	Yiteng Tu, Weihang Su, Yujia Zhou, Yiqun Liu, and Qingyao Ai. 2025. Rbft: Robust fine-tuning for retrieval-augmented generation against retrieval defects. <i>CoRR</i> , abs/2501.18365.	A.1 Datasets	748
698	Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. <i>Journal of machine learning research</i> , 9(11).	We provide a brief description of the datasets used in this work. All these datasets are in English.	749 750
702	Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, and 1 others. 2025. Thoughts are all over the place: On the underthinking of o1-like llms. <i>arXiv preprint arXiv:2501.18585</i> .	Our training dataset consists of:	751
703	Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. <i>arXiv preprint arXiv:2411.04368</i> .	<ul style="list-style-type: none"> The training set of StrategyQA (Geva et al., 2021), a question answering (QA) benchmark that requires multiple reasoning steps for each question. The questions are short but span diverse topics. 	752 753 754 755 756
705	Haotian Xu, Xing Wu, Weinong Wang, Zhongzhi Li, Da Zheng, Boyuan Chen, Yi Hu, Shijia Kang, Jiaming Ji, Yingying Zhang, and 1 others. 2025. Redstar: Does scaling long-cot data unlock better slow-reasoning systems? <i>arXiv preprint arXiv:2501.11284</i> .	<ul style="list-style-type: none"> The training set of HotpotQA (Yang et al., 2018), a QA dataset that requires reasoning over multiple supporting documents, which include factual knowledge. 	757 758 759 760
710	Shicheng Xu, Liang Pang, Mo Yu, Fandong Meng, Huawei Shen, Xueqi Cheng, and Jie Zhou. 2024a. Unsupervised information refinement training of large language models for retrieval-augmented generation. In <i>Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , ACL	<ul style="list-style-type: none"> The SuperGPQA dataset (Du et al., 2025), a QA benchmark covering 285 subjects to test the model’s graduate-level knowledge and reasoning capabilities. 	761 762 763 764
715		Our in-domain evaluation benchmark includes:	765
716		<ul style="list-style-type: none"> The test set of StrategyQA (Geva et al., 2021). 	766
717		<ul style="list-style-type: none"> The development set of HotpotQA (Yang et al., 2018), since the answers of its test set are unavailable. 	767 768 769
720		And our out-of-domain evaluation benchmark includes:	770 771

- The FRAMES dataset (Krishna et al., 2024), a QA benchmark to test model performance in RAG scenarios. It requires multi-step reasoning over factual information from multiple sources.
- The GPQA extended set (Rein et al., 2024), a challenging graduate-level QA benchmark in biology, physics, and chemistry domains. We use the extended set in our evaluation.
- The SimpleQA (Wei et al., 2024) dataset, a QA benchmark including short questions that require factual retrieval and reasoning ability, covering a wide range of topics.

A.2 Training Details

We list the details of training hyperparameters here:

Batch size: 8,
Number of machines: 1,
Number of processes: 8.
Training epochs: 5,
Training steps: 625,
Learning rate: $5e-6$,
Optimizer: AdamW (with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay = 0.01),
Scheduler: cosine schedule (warmup steps: 62)
ZeRO optimization stage: 3,
Mixed precision: bf16,
 t_l : 0.9,
 t_u : 1.1,
 α : 1.0.

A.3 Prompts

We use the prompt shown in Table 6 to request DeepSeek-R1 API for its reasoning trace, and use the prompt shown in Table 7 to evaluate the models across all the datasets.

You are a helpful assistant. You will be given a question. You need to answer the question by reasoning step by step. In the end, output the final answer in a new line with the prefix "Final answer:". The final answer should be yes or no, a choice letter, or a short phrase, without further explanations.

Question: {*Question*}

Options: {*Options*} (if there are choices for the question)

Table 6: Prompt used to request DeepSeek-R1 API

You are Qwen, created by Alibaba Cloud. You are a helpful assistant. You will be given a question. You need to answer the question by reasoning step by step. In the end, output the final answer in a new line with the prefix "Final answer:". The final answer should be yes or no, a choice letter, or a short phrase, without further explanations.

Context: {*Context*} (if there exist context for the question)

Question: {*Question*}

Options: {*Options*} (if there are choices for the question)

Table 7: Prompt used to evaluate the models

B Case Study

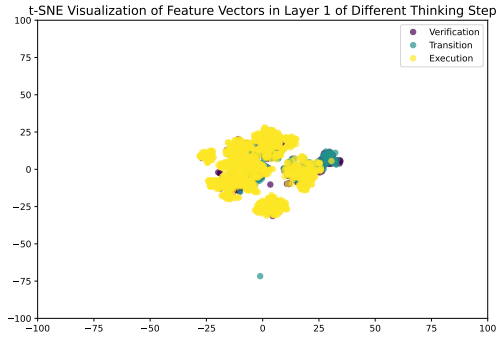
We show the output of original Qwen2.5-7B-Instruct and that of our fine-tuned model on a question from the GPQA dataset in Table 8. Qwen2.5-7B-Instruct lists all the systems and claims all of them can coexist in a multi-star system directly, thus giving a wrong answer, which is not even included in the choices. DynaGuide-7B gives its reasoning step by step (we omit many reasoning steps here) and finally outputs the correct answer. Furthermore, it thinks the possible systems that can coexist are the second, third, and fourth, which is completely correct according to the ground truth explanation in the dataset. This case study demonstrates the effectiveness of our framework.

C t-SNE Visualization of Feature Vectors of Different Thinking Step

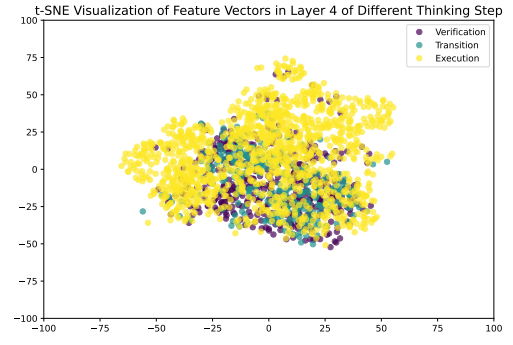
The t-SNE visualization (Van der Maaten and Hinton, 2008) of feature vectors of different thinking step types are shown in Figure 3. We observe that in deep layer of the model, especially the 20th layer, the vectors show weak separability after projection to 2D space, which implies they are more separable in the original high-dimensional hidden space.

<p>Question: The majority of stars in our Galaxy form and evolve in multi-stellar systems. Below are five potential multi-star systems that are presented. How many of these systems can coexist? W Virginis type star, G2V, M4V, RGB star(1.5Msun) WD (B5 when in the MS) and A0V G2V, K1V, M5V DA4, L4 WD (MS mass of 0.85Msun), K3V, A star with a mass of 0.9Msun in the MS. Options: A. 3 B. 4 C. 1 D. 2</p>	
<p>Answer: A. 3</p>	
Qwen2.5-7B-Instruct	DynaGuide-7B
<p>To determine how many of these systems can coexist, we need to understand what a multi-star system is and the characteristics of the stars listed.</p> <ol style="list-style-type: none"> 1. **W Virginis type star, G2V, M4V, RGB star(1.5Msun)**: This system includes a variable star, a main-sequence star, and a red giant. These can coexist in a multi-star system. 2. **WD (B5 when in the MS) and A0V**: This system includes a white dwarf and a main-sequence star. These can coexist in a multi-star system. 3. **G2V, K1V, M5V**: This system includes three main-sequence stars. These can coexist in a multi-star system. 4. **DA4, L4**: This system includes a white dwarf and a brown dwarf. These can coexist in a multi-star system. 5. **WD (MS mass of 0.85Msun), K3V, A star with a mass of 0.9Msun in the MS**: This system includes a white dwarf, a main-sequence star, and another main-sequence star. These can coexist in a multi-star system. <p>Each of the five systems listed can coexist in a multi-star system. Therefore, the answer is: Final answer: 5</p>	<p>think</p> <p>Okay, let's try to figure this out. The question is about how many of the five multi-star systems can coexist. The options are A, B, C, D, and the answer choices are numbers. Wait, but the user didn't provide options in the original question. Hmm, maybe the options are part of the question here. Let me check again.</p> <p>Wait, the user wrote: "The majority of stars in our Galaxy form and evolve in multi-stellar systems. Below are five potential multi-star systems that are presented. How many of these systems can coexist?"</p> <p>Then the five systems are listed. And the options are A.3 B.4 C.1 D.2. So the options are those four letters. So the answer is one of those. Let me read the systems again.</p> <p>The first system is W Virginis type star, G2V, M4V, RGB star(1.5Msun). I'm not sure what a W Virginis star is. Maybe a variable star? But I don't recall them forming multi-star systems. G2V and M4V are spectral types, which are single stars. So maybe this system can't coexist.</p> <p>The second system is WD (B5 when in the MS) and A0V. WD might be a white dwarf, but B5 and A0 are main sequence stars. Wait, white dwarfs form when main sequence stars die. So maybe a system where a white dwarf is part of a binary with another main sequence star. That could happen. So maybe this one can coexist.</p> <p>The third system is G2V, K1V, M5V. All main sequence stars. Maybe they form a triple system.</p> <p>...</p> <p>In summary, the possible systems that can coexist are the second, third, and fourth. So three. Therefore, the answer is A.3.</p> <p>answer</p> <p>Final answer: A</p>

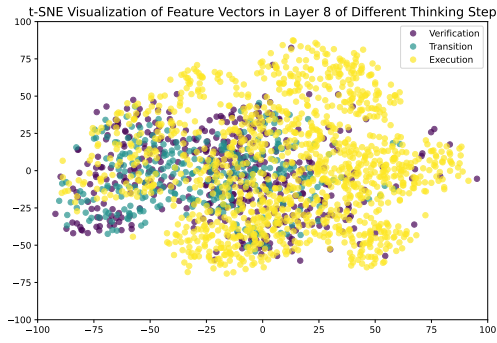
Table 8: Case Study



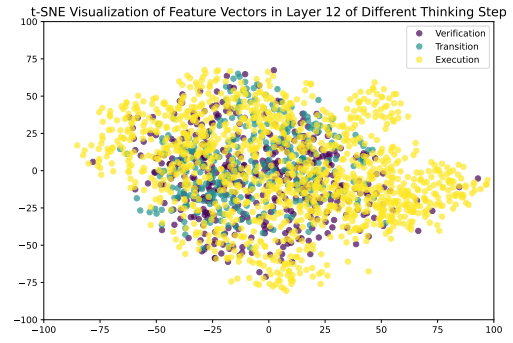
(a) Layer 1



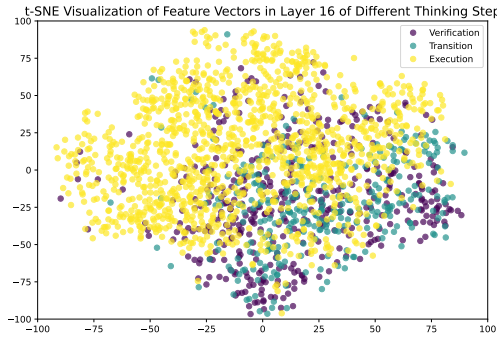
(b) Layer 4



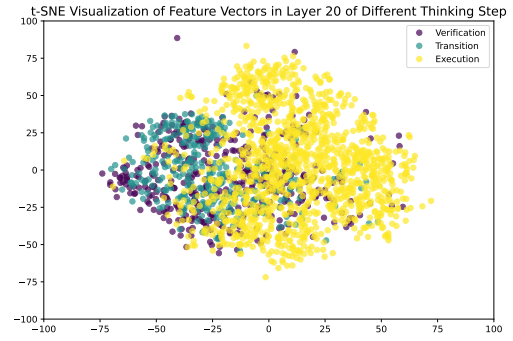
(c) Layer 8



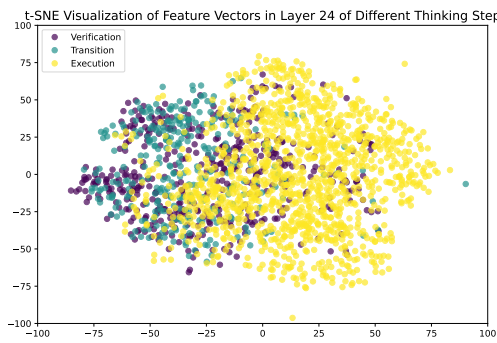
(d) Layer 12



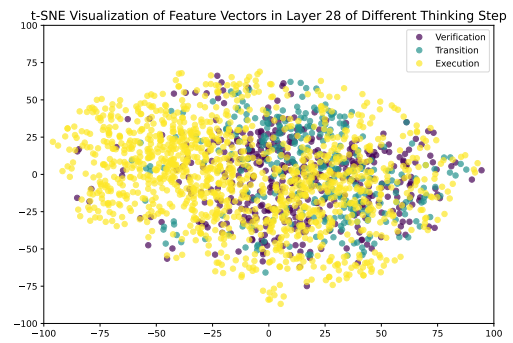
(e) Layer 16



(f) Layer 20



(g) Layer 24



(h) Layer 28

Figure 3: t-SNE Visualization of Feature Vectors of Different Thinking Step across Different Layers