HERD: Hierarchical Education-Guided Reading Distillation

Anonymous ACL submission

Abstract

Distilling large language models into compact students requires more than transferring final answers: the student must acquire reading abilities that support local linguistic analysis, cross-sentence understanding, and document-level reasoning. However, existing distillation methods typically emphasize outputs, task answers, or intermediate traces, and provide less guidance on how reading materials should be organized and scheduled across prerequisite skills. We propose HERD (**H**ierarchical **E**ducation-Guided **R**eading **D**istillation), a framework that turns teacher-generated reading supervision into a progressive and adaptive curriculum. HERD constructs reading materials at three levels, covering sentence-level linguistic analysis, paragraph-level relation modeling, and document-level evidence integration, and trains the student with a mastery-aware scheduler that adjusts the emphasis over reading levels. Across ten benchmarks, HERD improves the base student from 48.1 to 60.4 average accuracy, with especially clear gains on semantic judgment and reasoning tasks. It also provides a stronger starting point for downstream adaptation: after supervised fine-tuning, HERD reaches 67.4 average accuracy, suggesting that the curriculum improves both direct performance and transferability.

1 Introduction

Distillation for compact language models commonly transfers either general teacher behavior or task-specific supervision. Early knowledge distillation matches teacher distributions (Hinton et al., 2015), and later work extends the signal to internal representations and token-level generative behavior (Jiao et al., 2020; Gu et al., 2024). Task-oriented distillation can also use teacher-provided supervision for target generation tasks (Chen et al., 2020). Process-oriented methods add rationales or step-level feedback (Hsieh et al., 2023; Lightman et al., 2023), while explanation-trace distillation provides another form of intermediate supervision (Mukherjee et al., 2023). These studies show that intermediate supervision can help students learn beyond final answers, but reading-intensive understanding requires not only richer traces, but also a structured path for learning how to read.

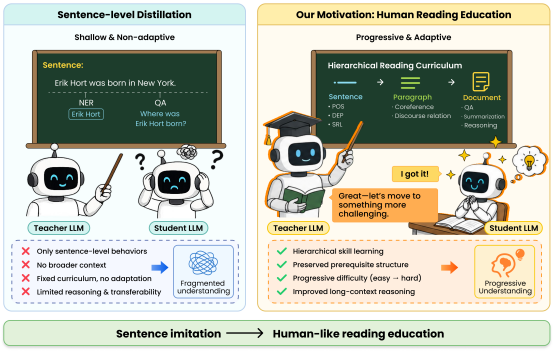

Figure 1: HERD organizes reading supervision into a hierarchical curriculum over sentence-, paragraph-, and document-level skills, and schedules training according to student mastery.

Reading ability is naturally hierarchical. Local analysis supports lexical, syntactic, and predicate-argument understanding; connected reading requires tracking entities, events, and discourse relations across sentences; and document understanding depends on salience, evidence aggregation, and global reasoning. Work on discourse parsing and implicit discourse relation recognition shows that contextual information matters for relation modeling (Atwell et al., 2022; Xiang et al., 2022). Long-document question answering and long-context studies further highlight the importance of organizing and using relevant evidence in extended texts (Nair et al., 2023; Liu et al., 2024). Transferable reading ability should therefore be learned through coordinated practice over sentence-, paragraph-, and document-level skills.

Existing reading-distillation pipelines do not

fully capture this learning structure. They may provide useful intermediate reading behaviors (Zhou et al., 2025). However, these signals are often introduced as separate sources or mixed under a preset schedule, without modeling how lower-level skills support higher-level comprehension. In a reading curriculum, materials should follow prerequisite relations and the training emphasis should change with the student's learning state. The key issue is thus not only generating reading supervision, but delivering it as a progressive and adaptive educational process.

We take inspiration from human reading education to address this issue. Reading instruction is progressive: students first learn local patterns, then connect sentences and paragraphs, and eventually handle global comprehension, synthesis, and reasoning. It is also adaptive: teachers revisit unstable prerequisite skills and move to harder materials when the learner is ready. This intuition is consistent with work showing that skill dependencies affect language-model training (Chen et al., 2023). Human-curriculum-inspired ordering and phased instruction tuning provide related evidence that training order can matter (Lee et al., 2024; Pang et al., 2024).

As illustrated in Figure 1, we propose HERD (Hierarchical Education-Guided Reading Distillation), a reading curriculum distillation framework for compact language models. HERD constructs teacher-generated reading materials at three levels: sentence-level linguistic analysis, paragraph-level relation modeling, and document-level evidence integration. It organizes these signals by prerequisite structure and trains the student with a mastery-aware scheduler that dynamically adjusts the emphasis over reading levels. In this way, HERD turns reading supervision from a flat collection of generated data into a structured and adaptive educational process.

Experiments show that HERD improves compact-model performance across language understanding, reasoning, cross-lingual transfer, and long-context comprehension tasks. Models initialized with HERD also benefit more from downstream task distillation and supervised fine-tuning, suggesting that hierarchical and adaptive reading education improves both direct task performance and transferability.

Our main contributions are as follows:

- We propose HERD, a reading curriculum distillation framework that organizes small-model distillation around prerequisite relations among reading skills.
- We construct hierarchical reading materials over sentences, paragraphs, and documents, covering local linguistic instruction, connected reading, and evidence-based reasoning.
- We deliver the hierarchical reading signals through a mastery-aware scheduler that adapts training emphasis across reading levels.

2 Related Work

2.1 Knowledge and Task Distillation

Knowledge distillation transfers teacher knowledge to smaller students by matching output distributions (Hinton et al., 2015). Subsequent methods match internal representations or attention patterns, as in TinyBERT and MiniLM (Jiao et al., 2020; Wang et al., 2020). Recent LLM distillation also transfers token-level generative behavior through reverse-KL or on-policy objectives (Gu et al., 2024; Agarwal et al., 2024). Other work develops general distillation frameworks for large pretrained models (Tan et al., 2023). Task-oriented distillation can also use teacher-provided supervision for target generation tasks (Chen et al., 2020). These methods are effective for compression and task transfer, but their main training signal is still teacher behavior, task answers, or distributional matching. HERD differs by treating distillation as a reading education process that explicitly trains transferable reading abilities from local linguistic analysis to document-level reasoning.

2.2 Curriculum Learning and Task Scheduling

Curriculum learning improves learning by arranging examples or tasks in a meaningful order (Bengio et al., 2009). Self-paced learning selects easier samples first and gradually expands the training set during optimization (Kumar et al., 2010). Automated curricula and competence-based curricula instead adjust exposure based on progress or difficulty (Graves et al., 2017; Platanios et al., 2019). Recent language-model training also shows that skill dependencies can influence learning outcomes (Chen et al., 2023). Human-curriculum-inspired and phased instruction-tuning studies provide further evidence that task order matters (Lee et al.,

2024; Pang et al., 2024). However, many curricula define difficulty through surface features, loss, confidence, or sample selection. Reading distillation requires a different axis: sentence-level linguistic skills support paragraph-level relation modeling, which in turn supports document-level reasoning. HERD therefore organizes the curriculum around prerequisite relations among reading abilities and adapts level sampling according to student mastery.

2.3 Reading-Inspired Distillation

Reading-inspired distillation reinterprets student training as an educational process. A representative direction trains compact models on sentence-level reading behaviors such as named entity recognition, question raising, and question answering (Zhou et al., 2025), showing that reading behaviors can provide more transferable supervision than task labels alone. Related work distills free-text explanations from LLMs (Zhang et al., 2024); another line distills structured conceptual knowledge for commonsense reasoning (Wang et al., 2024). These studies suggest that students can benefit from supervision beyond final answers when the teacher signal exposes useful intermediate structure. Yet these directions still leave the organization of reading supervision underexplored. Local reading behaviors, explanations, or structured knowledge alone do not specify how different reading abilities should be ordered during training. HERD addresses this gap by organizing teacher-generated reading supervision across sentence, paragraph, and document levels, and by adjusting level sampling according to the student's training state.

3 Method

As shown in Figure 2, HERD contains three stages. First, it constructs hierarchical reading supervision by prompting a teacher model on sentence, paragraph, and document units. Second, it converts the generated supervision into bundles organized by source document that preserve the prerequisite order among reading skills. Third, it trains the student with a mastery-aware scheduler that adapts both the reading level and the accessible material range during optimization.

3.1 Problem Formulation

Let $\mathcal{D} = \{d_m\}_{m=1}^{M}$ denote the source corpus. Each document d is segmented into sentence units $\mathcal{S}_d = \{s_i\}$, paragraph units $\mathcal{P}_d = \{p_j\}$, and the whole-document unit d. We organize reading supervision into three levels: sentence-level local analysis, paragraph-level relation modeling, and document-level evidence integration. For a text unit u and a reading task t, a task-specific prompt asks the teacher to generate a target reading behavior $y_{u,t}$. The student S_θ is trained to generate $y_{u,t}$ conditioned on the source text and task instruction. Thus, unlike plain language-model continuation, the objective of HERD is to learn an explicit mapping from source texts and task instructions to teacher-generated reading behaviors across multiple granularities.

3.2 Hierarchical Reading Supervision

Following prior reading distillation work, we use Common Crawl as the source of basic reading materials. We randomly sample 20M documents, filter them with QuRating (Wettig et al., 2024), and keep 100K relatively fluent, coherent, and information-rich documents. Each selected document is segmented into sentence, paragraph, and document units, and the teacher model is prompted to generate eight types of reading supervision over these units. As shown in Table 1, the tasks are organized according to the granularity of reading: sentence-level tasks provide local linguistic and semantic structures, paragraph-level tasks provide cross-sentence connections, and document-level tasks provide summarization, question answering, and reasoning over distributed information. In this way, the generated data forms a hierarchical supervision corpus rather than a flat collection of independent pseudo-labels.

3.3 Document-Centered Curriculum Construction

After teacher supervision is generated, HERD builds a curriculum along two axes: a coarse ordering of reading materials and prerequisite relations among reading skills. For each valid teacher output, we construct a task-conditioned training example

$$z_{u,t} = [\mathrm{INST}_t;\ x_u;\ \mathrm{SEP};\ y_{u,t}], \qquad (1)$$

where x_u is the source sentence, paragraph, or document, INST_t identifies the reading task, and SEP separates the source from the teacher-generated reading behavior. Outputs that are empty or violate the expected prompt format are removed. The resulting examples are grouped into sentence-, paragraph-, and document-level pools.

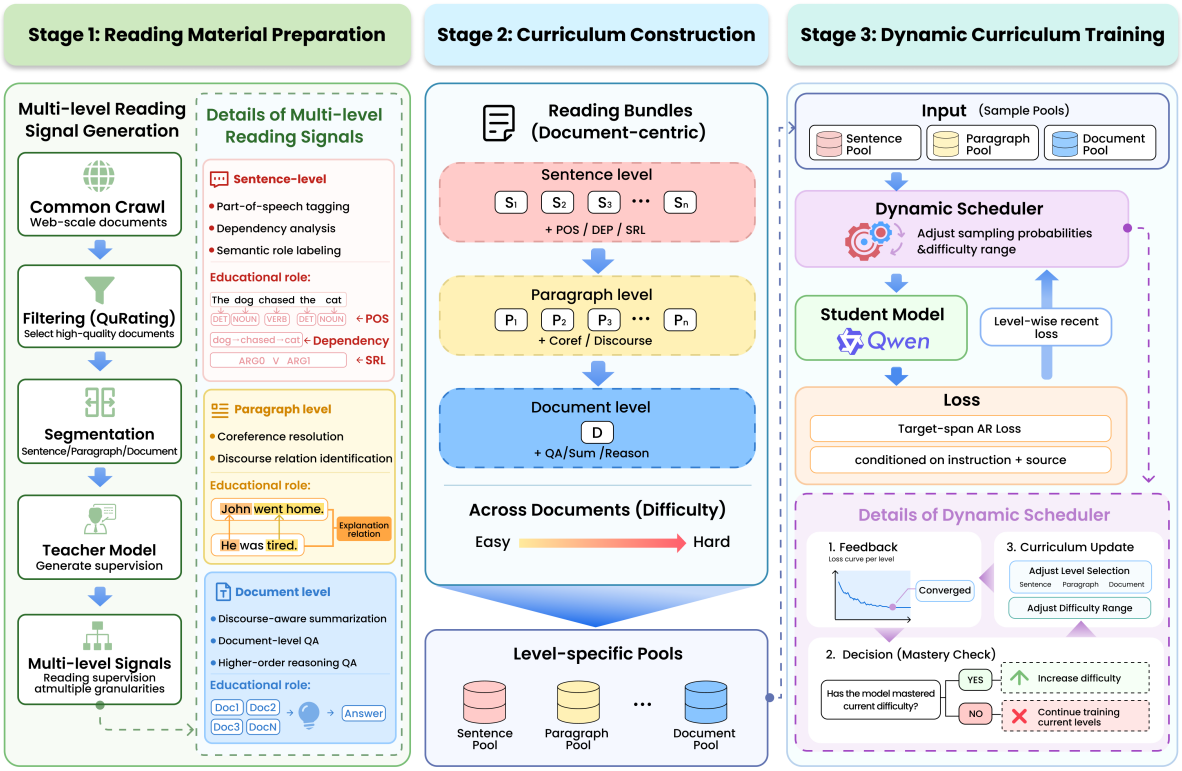

Figure 2: Overview of HERD. The framework prepares multi-level reading materials, constructs bundles organized by source document, and trains the student with dynamic curriculum scheduling.

To preserve document coherence, examples from the same document are first organized as a document-centered bundle: sentence-level annotations are placed before paragraph-level annotations, followed by document-level QA, summarization, and reasoning signals. This order follows a prerequisite structure in which local analysis precedes relation modeling, and relation modeling precedes document-level integration. The snippets are then greedily packed into autoregressive training chunks under the token limit. Bare sentences and paragraphs are not emitted, so the training signal remains source–annotation supervision rather than raw-text continuation.

Across documents, we keep the preprocessing order as a rough order for introducing documents into the sampling pool. This order only controls when more complex documents enter training; the main curriculum axis remains the reading-skill hierarchy and the student's recent loss tracked for each reading level.

3.4 Mastery-Aware Dynamic Scheduling

The static curriculum in Section 3.3 defines the textbook, while the scheduler decides what the student should practice at each step. For each reading level ℓ, we track an exponential moving average of recent training loss:

$$\bar{L}_{\ell}^{(t)} = (1-\rho)\bar{L}_{\ell}^{(t-1)} + \rho L_{\ell}^{(t)}, \tag{2}$$

where $L_{\ell}^{(t)}$ is the batch loss when level ℓ is sampled at step t. We use the recent level-wise loss as a proxy for student mastery. The scheduler uses this estimate of mastery at each level in two ways. First, it gradually increases the probability of paragraph- and document-level samples when their prerequisite levels become stable. Second, it keeps a replay path for lower levels whose loss rises during later training. The accessible document range is expanded together with this process, so training starts from simpler sentence-level examples and gradually exposes the student to longer and more integrative reading materials.

Given a sampled example $z_{u,t}$, the student is optimized with a conditional autoregressive loss on the teacher-generated reading behavior:

$$\mathcal{L}(z_{u,t}) = -\sum_{k=1}^{|y_{u,t}|} \log p_\theta(y_{u,t,k} \mid \mathrm{INST}_t, x_u, y_{u,t,<k}). \tag{3}$$

Level	Task	Form	Educational role
Sentence	Part-of-speech tagging	Analysis	Teaches lexical category recognition and basic word-level linguistic awareness.
Sentence	Dependency analysis	Analysis	Teaches syntactic relations among words, including subjects, objects, modifiers, and clauses.
Sentence	Semantic role labeling	Analysis	Teaches predicate-argument structure and event-level semantic roles.
Paragraph	Coreference resolution	QA	Teaches the student to track entities, events, and descriptions across sentences.
Paragraph	Discourse relation identification	Analysis	Teaches causal, contrastive, elaborative, and temporal relations between discourse units.
Document	Discourse-aware summarization	Generation	Teaches global condensation while preserving main ideas and discourse structure.
Document	Document-level question answering	QA with explanation	Teaches integration of non-adjacent evidence and explicit reasoning over long contexts.
Document	Higher-order reasoning QA	QA with explanation	Teaches synthesis, implication, argument evaluation, and holistic document-level reasoning.

Table 1: Reading supervision generated by the teacher model in HERD. The tasks are organized from sentence-level local analysis to paragraph-level connected reading and document-level reasoning.

In implementation, source tokens provide conditioning context and the loss is applied to the annotation span.

4 Experiments

We use Qwen3-8B as the teacher model due to its strong instruction-following ability and efficiency in generating large-scale reading supervision. We use Qwen3-0.6B, a smaller language model from the same decoder-only model family, to initialize the student model. To compare the student model with a larger model from the same family, we also include Qwen3-8B as an upper reference. In HERD, we randomly sample 20M documents from Common Crawl, filter them with QuRating, and keep 100K high-quality documents to collect hierarchical reading supervision generated by Qwen3-8B.

4.1 Baselines

We organize baselines by the claim they test. **Qwen3-0.6B** is the original student without reading distillation, and **Qwen3-8B** serves as an upper reference. To compare with conventional distillation, we include **SKD** and **MiniLLM**, a reverse-KL language-model distillation method. To test whether sentence-level reading education is sufficient, we compare with **BRD**, which trains on local reading behaviors. To evaluate transferability after reading education, we compare downstream adaptation from BRD or HERD using teacher-generated task answers (**BRD + Task Distill**, **HERD + Task Distill**) or gold labels (**BRD + SFT**, **HERD + SFT**). **Task Distillation** and **SFT** from the original student provide corresponding non-reading initialization baselines. All student-side reading and distillation baselines are trained from the same Qwen3-0.6B initialization under the matched optimization budget described in Section 5.1.

4.2 Evaluation

We adopt a spectrum of downstream benchmarks to evaluate different reading abilities. For local semantic understanding, we use CB (de Marneffe et al., 2019), PAWS-X (Yang et al., 2019), RTE from GLUE (Wang et al., 2018), SST-2 (Socher et al., 2013), and XNLI (Conneau et al., 2018). For contextual reasoning, we use BoolQ (Clark et al., 2019), LogicBench (Parmar et al., 2024), and LogiQA2.0 (Liu et al., 2023). For mixed comprehensive evaluation, following BRD (Zhou et al., 2025), we evaluate on the same 73 multiple-choice tasks from BigBench (Srivastava et al., 2023). We further use LongBench v2 (Bai et al., 2025) to test long-context document understanding. Unless otherwise specified, all results report accuracy.

4.3 Results

Table 2 evaluates whether hierarchical reading education improves both direct evaluation and downstream adaptation. The first block compares HERD with conventional distillation and sentence-level reading distillation, while the latter two blocks test whether the resulting initialization benefits more from task-level teacher supervision or gold labels.

The following ablations further disentangle the roles of hierarchical reading signals and mastery-aware curriculum scheduling under the matched training budget.

General distillation and reading education. HERD improves the average score of Qwen3-0.6B from 48.1 to 60.4, achieving the best overall average among the student-side general distillation and reading-education methods. Relative to the original student, the gains are especially clear on semantic judgment and reasoning-oriented benchmarks such as CB, RTE, XNLI, LogicBench, and LogiQA2.0, showing that hierarchical reading supervision is most beneficial when tasks require inference beyond isolated sentences.

Compared with traditional distillation methods, HERD yields a stronger overall profile: it brings large gains on several understanding benchmarks and improves reasoning-oriented evaluation, while remaining competitive on mixed and long-context evaluation. SKD and MiniLLM mainly transfer teacher behavior through distribution matching or KL objectives and perform well on some logical tasks, but they do not improve CB, SST-2, LogiQA2.0, and LongBench as consistently as HERD. The gain over BRD further suggests that extending reading supervision beyond sentence-level behaviors to paragraph- and document-level signals improves transferable reading ability.

Results without downstream task supervision. When downstream inputs are available but gold labels are not, task distillation uses teacher-generated answers as supervision. Under this setting, HERD provides a consistently stronger initialization than sentence-level reading distillation: HERD + Task Distill improves every reported benchmark and raises the average from 60.0 to 64.0. This indicates that hierarchical reading education is complementary to task-level teacher supervision and leads to better downstream adaptation from generated task answers.

Results with downstream task supervision. In this scenario, gold downstream task labels are available for supervised fine-tuning. HERD + SFT improves over SFT from 60.2 to 67.4 and over BRD + SFT from 63.8 to 67.4. The transfer benefit is particularly clear on LongBench v2, where HERD + SFT reaches 35.6 compared with 25.6 for BRD + SFT. These results indicate that hierarchical reading education not only improves direct evaluation performance, but also prepares the student model to benefit more from labeled downstream supervision.

5 Analysis

5.1 Training Budget and Ablation Setup

To control training computation, all student-side methods are initialized from Qwen3-0.6B and trained under the same optimization budget unless otherwise specified. We use the same global batch size, maximum sequence length, optimizer, learning-rate schedule, and train all reading or distillation methods for 12000 optimization steps. This matched setup makes the comparison focus on supervision format and curriculum organization rather than differences in training steps or optimizer settings.

5.2 Main Component Ablation

Table 3 ablates the two main components in HERD: hierarchical reading signals and mastery-aware curriculum scheduling.

Student-Further tests whether the gains come merely from additional text training, while Static Curriculum tests whether adaptive scheduling is necessary beyond hierarchical supervision. This separates the effect of hierarchical reading signals from that of additional corpus exposure. Table 3 shows that raw-text continuation yields only marginal average improvement over the original student: Student-Further slightly improves LogicBench and BigBench but decreases LongBench. Adding hierarchical reading signals with a static curriculum improves the average to 43.8, and full HERD further improves it to 45.9 with gains on all four benchmarks. These results indicate that both hierarchical reading supervision and mastery-aware curriculum organization contribute to the observed gains.

5.3 Reading Signal Levels

We next analyze the contributions of different levels of reading signals. The reading supervision in HERD consists of three categories: sentence-level local parsing, paragraph-level relation modeling, and document-level comprehensive reasoning. To test the necessity of each category, we remove one type of signal at a time while keeping the remaining training settings unchanged.

Table 4 shows that no single level fully explains the performance of HERD. Among the single-level variants, document-level supervision gives

Model	Basic Understanding						Reasoning		Mixed	Long-context	Avg.
	BoolQ	CB	PAWS-X	RTE	SST-2	XNLI	LogicBench	LogiQA2.0	BigBench	LongBench	Avg.
General distillation and reading education											
Qwen3-8B	86.4	80.4	79.4	85.9	94.6	77.6	77.6	63.2	49.0	42.2	73.6
Qwen3-0.6B	58.8	41.1	57.0	57.4	56.1	37.7	62.0	44.3	37.9	28.3	48.1
BRD	**73.9**	60.7	50.3	61.0	82.3	47.8	62.4	46.0	**38.6**	26.7	55.0
SKD	70.5	50.0	56.9	61.0	51.4	45.3	**68.9**	43.7	38.3	25.6	51.2
MiniLLM	68.2	53.6	56.8	66.4	66.5	45.7	67.8	44.9	37.8	25.0	53.3
HERD	70.3	**82.1**	**57.3**	**69.0**	**86.9**	**55.1**	68.4	**47.8**	38.4	**28.9**	**60.4**
Without downstream task supervision											
Task Distillation	62.5	66.1	64.2	63.5	82.6	47.7	64.9	45.2	37.9	27.2	56.2
BRD + Task Distill	65.6	75.0	66.8	67.9	90.1	57.5	67.7	47.2	38.1	24.4	60.0
HERD + Task Distill	**72.0**	**83.9**	**69.0**	**72.9**	**91.7**	**63.1**	**71.0**	**48.0**	**38.6**	**29.4**	**64.0**
With downstream task supervision											
SFT	65.6	73.2	72.6	66.8	83.6	54.1	71.9	46.9	38.3	28.9	60.2
BRD + SFT	72.2	78.6	72.8	72.2	90.5	67.0	72.5	48.3	38.6	25.6	63.8
HERD + SFT	**74.9**	**89.3**	**80.5**	**73.6**	**90.6**	**67.5**	**74.5**	**48.6**	**38.9**	**35.6**	**67.4**

Table 2: Main results. The first block reports original models and general distillation or reading-education methods; the second and third blocks report downstream adaptation without and with gold task supervision, respectively. Avg. denotes the average over the ten listed benchmarks. Bold indicates the best student-side result and underline indicates the second-best result within each block, excluding the Qwen3-8B teacher reference.

Method	Signals	Adaptive Curriculum	LogicBench	LogiQA2.0	BigBench	LongBench	Avg.
Student	✗	✗	62.0	44.3	37.9	28.3	43.1
Student-Further	✗	✗	63.9	44.1	38.0	26.7	43.2
HERD w/ Static Curriculum	✓	✗	64.9	44.3	38.0	27.8	43.8
HERD	✓	✓	**68.4**	**47.8**	**38.4**	**28.9**	**45.9**

Table 3: Main component ablation. Avg. is over LogicBench, LogiQA2.0, BigBench, and LongBench. Student-Further trains on raw paragraphs without teacher annotations; Static Curriculum uses the same reading signals as HERD with a fixed sentence-to-paragraph-to-document schedule. Bold indicates the best result in each metric column.

the strongest average performance, especially on LogicBench and LogiQA2.0, suggesting that QA, summarization, and reasoning signals are particularly useful for higher-level reasoning benchmarks. However, the full curriculum still achieves the best average score, showing that lower-level linguistic and paragraph-level relation signals provide complementary support rather than merely adding isolated tasks.

The removal results further support this conclusion. Removing sentence-level signals causes only a small average drop, suggesting that these signals play a complementary role when combined with stronger paragraph- and document-level supervision. Removing paragraph-level signals causes the largest average drop, with clear degradation on LogiQA2.0 and LongBench; removing document-level signals also hurts the average, especially LongBench. This suggests that paragraph-level coreference and discourse relations help bridge local linguistic analysis and document-level reasoning, while document-level supervision is important for global evidence integration. Overall, the best performance is obtained when sentence-, paragraph-, and document-level signals are combined. This finding supports the central intuition of HERD: a student model benefits from a curriculum that combines local linguistic supervision, cross-sentence relation modeling, and document-level reasoning.

5.4 Curriculum Strategies

To validate the curriculum learning mechanism, we further compare different training scheduling strategies. All methods use the same multi-level reading data and differ only in the order or sampling strategy by which samples enter training.

Table 5 compares different scheduling strategies using the same reading supervision. Random Mixing performs well on LogicBench but poorly on LogiQA2.0, suggesting that unstructured mixing can help some reasoning cases but yields less reliable gains across benchmarks. Reverse Curriculum further hurts LongBench, consistent with the view that emphasizing high-level document reasoning before prerequisite practice is less effective. Static Curriculum improves LogiQA2.0 over random and reverse schedules, but still lags behind HERD. In

Method	Sentence	Paragraph	Document	LogicBench	LogiQA2.0	BigBench	LongBench	Avg.
HERD Sentence-only	✓	✗	✗	66.6	40.3	37.8	23.3	42.0
HERD Paragraph-only	✗	✓	✗	66.2	44.4	38.3	28.3	44.3
HERD Document-only	✗	✗	✓	67.4	**47.8**	38.2	27.8	45.3
HERD w/o Sentence-level	✗	✓	✓	68.3	47.7	38.3	28.3	45.7
HERD w/o Paragraph-level	✓	✗	✓	67.2	46.3	38.0	26.7	44.6
HERD w/o Document-level	✓	✓	✗	67.6	47.3	38.0	27.2	45.0
HERD	✓	✓	✓	**68.4**	**47.8**	**38.4**	**28.9**	**45.9**

Table 4: Ablation results for reading-signal levels. Sentence-level signals include POS, dependency parsing, and SRL; Paragraph-level signals include coreference and discourse relations; document-level signals include QA, summarization, and reasoning. Bold indicates the best result in each metric column; ties are all bolded.

Method	Logic-Bench	LogiQA 2.0	Big-Bench	Long-Bench	Avg.
Random Mixing	67.4	39.9	38.3	27.8	43.4
Reverse Curriculum	66.6	40.3	38.0	26.1	42.7
Static Curriculum	64.9	44.3	38.0	27.8	43.8
HERD	**68.4**	**47.8**	**38.4**	**28.9**	**45.9**

Table 5: Ablation results for curriculum strategies. Random Mixing randomly mixes all multi-level samples; Static Curriculum follows sentence-level → paragraph-level → document; Reverse Curriculum uses the opposite order; HERD adjusts level sampling using recent autoregressive loss for each reading level. Bold indicates the best result in each metric column.

this curriculum comparison, full HERD achieves the best results on all four reported benchmarks, showing that dynamically adjusting the curriculum according to recent loss at each reading level is more effective than using a fixed order. This supports the scheduler's role in balancing progression to higher-level reading tasks with replay of lower-level skills when they become unstable.

5.5 Discussion

The ablation analysis leads to three conclusions. First, raw-text continuation under the same optimization budget does not explain the gains observed for HERD; hierarchical reading supervision provides more effective training signals. Second, the three reading levels are complementary. Document-level supervision is strong by itself, but full HERD performs best when sentence-, paragraph-, and document-level signals are combined. Third, curriculum organization matters: in this ablation, dynamic scheduling outperforms random, reverse, and static curricula on all reported metrics, indicating that hierarchical reading signals are most effective when delivered according to the student's recent performance at each reading level.

6 Conclusion

We presented HERD, a hierarchical education-guided reading distillation framework that extends isolated sentence-level supervision into a multi-level curriculum over sentence-, paragraph-, and document-level reading abilities. It constructs teacher-generated reading signals for local linguistic analysis, cross-sentence relation modeling, and document-level reasoning, and organizes them through a dynamic curriculum that adapts the training emphasis to the student's learning state. Experiments on Qwen3 small models show that HERD improves the overall direct evaluation average, with the clearest gains on semantic judgment and reasoning benchmarks. It also provides a stronger initialization for teacher-generated task distillation and supervised fine-tuning, with consistent gains over BRD-initialized models and a particularly clear LongBench v2 improvement. Ablation studies support that these improvements are associated with both hierarchical reading supervision and mastery-aware curriculum organization.

Limitations

This work is currently evaluated mainly within one model family, using Qwen3-8B as the teacher model and Qwen3-0.6B as the primary student model. Although HERD improves overall performance and provides stronger initialization for downstream adaptation, its direct gains on some long-context benchmarks remain moderate at the 0.6B scale. Future work should further examine whether the same hierarchical reading curriculum remains effective across different model families, teacher models, student scales, and multilingual corpora.

Ethical considerations

HERD relies on teacher-generated supervision derived from filtered Common Crawl documents. Although the source documents are filtered for quality, web text may still contain social biases, harmful content, private information, or copyrighted material, and teacher-generated annotations may reproduce or amplify such issues. The resulting student model should therefore be evaluated for bias and safety before deployment, and the generated reading supervision should not be treated as human-verified ground truth. We use the data for research on model training and evaluation, and we do not release personally identifying examples.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*.

Katherine Atwell, Remi Choi, Junyi Jessy Li, and Malihe Alikhani. 2022. The role of context and uncertainty in shallow discourse parsing. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 797–811, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2025. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3639–3664, Vienna, Austria. Association for Computational Linguistics.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48.

Mayee F. Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. 2023. Skill-it! a data-driven skills framework for understanding and training language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 36000–36040.

Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jingjing Liu. 2020. Distilling knowledge learned in BERT for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7893–7905, Online. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The CommitmentBank: Investigating projection in naturally occurring discourse. *Proceedings of Sinn und Bedeutung*, 23(2):107–124.

Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1311–1320. PMLR.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

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

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

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23.

Bruce W. Lee, Hyunsoo Cho, and Kang Min Yoo. 2024. Instruction tuning with human curriculum. In *Findings of the Association for Computational Linguis-*

tics: NAACL 2024, pages 1281–1309, Mexico City, Mexico. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. 2023. LogiQA 2.0–an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of GPT-4. *arXiv preprint arXiv:2306.02707*.

Inderjeet Nair, Shwetha Somasundaram, Apoorv Saxena, and Koustava Goswami. 2023. Drilling down into the discourse structure with LLMs for long document question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14593–14606, Singapore. Association for Computational Linguistics.

Wei Pang, Chuan Zhou, Xiao-Hua Zhou, and Xiaojie Wang. 2024. Phased instruction fine-tuning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5735–5748, Bangkok, Thailand. Association for Computational Linguistics.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. LogicBench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13679–13707, Bangkok, Thailand. Association for Computational Linguistics.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, and 426 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Shicheng Tan, Weng Lam Tam, Yuanchun Wang, Wenwen Gong, Shu Zhao, Peng Zhang, and Jie Tang. 2023. GKD: A general knowledge distillation framework for large-scale pre-trained language model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 134–148, Toronto, Canada. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Weiqi Wang, Tianqing Fang, Chunyang Li, Haochen Shi, Wenxuan Ding, Baixuan Xu, Zhaowei Wang, Jiaxin Bai, Xin Liu, Cheng Jiayang, Chunkit Chan, and Yangqiu Song. 2024. CANDLE: Iterative conceptualization and instantiation distillation from large language models for commonsense reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2351–2374, Bangkok, Thailand. Association for Computational Linguistics.

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

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. QuRating: Selecting high-quality data for training language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 52915–52971. PMLR.

Wei Xiang, Bang Wang, Lu Dai, and Yijun Mo. 2022. Encoding and fusing semantic connection and linguistic evidence for implicit discourse relation recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3247–3257,

Dublin, Ireland. Association for Computational Linguistics.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Hanyu Zhang, Xiting Wang, Xiang Ao, and Qing He. 2024. Distillation with explanations from large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pages 5018–5028, Torino, Italia. ELRA and ICCL.

Zhi Zhou, Sirui Miao, Xiangyu Duan, Hao Yang, and Min Zhang. 2025. Basic reading distillation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30489–30502, Vienna, Austria. Association for Computational Linguistics.

A Data Construction Details

Table 6 summarizes the main data-construction settings used by HERD. We sample documents from Common Crawl, filter them with QuRating, and keep 100K high-quality documents as the source corpus for teacher-supervision generation. The teacher outputs are merged into document-level records containing sentence-, paragraph-, and document-level supervision. Documents are then arranged with the preprocessing order and used to build level-specific curriculum pools.

Item	Setting
Raw source	Common Crawl documents
Filtered corpus	100K high-quality documents selected by QuRating
Teacher model	Qwen3-8B
Student model	Qwen3-0.6B
Reading levels	Sentence, paragraph, and document
Training records	596,878 packed curriculum records
Packing limit	2000 tokens per record
Separator	Task delimiter between source text and teacher annotation
Control data	Same document source with teacher annotations removed

Table 6: Summary of data construction settings.

For curriculum construction, each document is expanded into task-conditioned source–annotation snippets. Sentence-level snippets contain POS analysis, dependency analysis, and semantic role labeling outputs for each sentence. Paragraph-level snippets contain coreference QA and discourse relation analysis for each paragraph. Document-level snippets contain document-level reasoning QA, discourse-aware summarization, and higher-order reasoning QA for the whole document. Bare sentence and paragraph examples are removed to avoid plain language-modeling shortcuts. Snippets are greedily packed into training records under the token limit using the Qwen3-0.6B tokenizer.

B Training and Downstream Adaptation Details

Unless otherwise specified, all student-side methods use the training budget described in Section 5.1. The student model is initialized from Qwen3-0.6B, and Qwen3-8B is used as the teacher model for generating reading supervision. We train the student with full-parameter training in four A100 GPUs using bf16 precision. For reading distillation, the per-device batch size is 8 with gradient accumulation 1, giving an effective batch size of 32; the maximum sequence length is 2048 tokens. All reading or distillation methods are trained for 12000 optimization steps with AdamW, a learning rate of 1×10^{-5}, and a cosine learning-rate schedule. We keep these core settings fixed across compared methods so that the results mainly reflect differences in supervision and curriculum organization under a matched optimization budget.

For downstream adaptation, each task is converted into the same instruction-input-output format with task-specific label instructions. For the six classification tasks, SST-2, PAWS-X, XNLI, BoolQ, RTE, and CB, training subsets are stratified by label with seed 42 and capped at 500 examples per task, except CB, whose full 250-example training set is used. For the additional reasoning and long-context tasks, we construct analogous task-specific adaptation sets: LogicBench and LogiQA2.0 use 500 training instances each, BigBench uses 500 training instances sampled from the selected multiple-choice tasks, and the full LongBench v2 adaptation setting uses 180 training and 180 validation instances. Task Distillation uses teacher-generated answers for the same task inputs, while SFT uses the gold labels or answers. The downstream adaptation runs use full-parameter training in LLaMA-Factory with the Qwen3 template and thinking disabled. For the six classification tasks, we use one GPU per job, per-device batch size 8, gradient accumulation 1, cutoff length 2048, one epoch, learning rate 1×10^{-5}, AdamW,

cosine scheduling with warmup ratio 0.03, and bf16 precision. For the reasoning and long-context tasks, we keep the same optimizer, learning rate, epoch count, and precision, but use cutoff length 4096 with per-device batch size 2 and gradient accumulation 4 to accommodate longer inputs.

C Benchmark Descriptions

Table 7 summarizes the evaluation benchmarks used in this paper.

D Prompt Templates

We list the full prompt templates used for teacher-supervision generation below. The templates are directly extracted from the active `base_prompt_template` or `base_prompt` variables in the data-generation scripts.

D.1 Part-of-Speech Tagging

```
Task:
Perform selective part-of-speech tagging on consecutive
↪  sentences from a unified narrative, identifying 3-5
↪  linguistically significant words per sentence (excluding
↪  basic function words).Use ONLY these 12 categories:
noun, verb, adjective, adverb, pronoun, preposition,
↪  conjunction, interjection, numeral, article, quantifiers,
↪  onomatopoeic.

Prioritize:
Low-frequency lexical items
Words with multiple POS potential
Domain-specific terminology

Example:
Enter a sentence: Beneath the archaeologist's trembling
↪  fingers, the bronze astrolabe revealed intricate patina
↪  patterns, its engravings whispering of celestial
↪  navigation.
Output result: This sentence contains these key tags:
↪  "archaeologist's" as a noun, "trembling" functioning as an
↪  adjective, and "whispering" marked as a verb.
Enter a sentence: "Astounding!" murmured Dr. Liang, carefully
↪  rotating the artifact until its azimuthal markings aligned
↪  with the noon sun.
Output result: We identify "Astounding" as an interjection,
↪  "murmured" is clearly a verb, "carefully" serves as an
↪  adverb, and "azimuthal" is categorized as an adjective.
Enter a sentence: Twelve graduate students clustered around the
↪  excavation pit, their calipers and brushes poised to
↪  document every millimeter.
Output result: The word "Twelve" is a numeral here, "clustered"
↪  acts as the main verb, while "millimeter" is identified as
↪  a noun.
Enter a sentence: The relic emitted a faint metallic ping when
↪  touched, suggesting remarkably preserved internal
↪  mechanisms.
Output result: In this case, "emitted" is tagged as a verb,
↪  "metallic" as an adjective, "ping" as an onomatopoeic, and
↪  "remarkably" as an adverb.
Enter a sentence: After recording three dozen measurements, the
↪  team concluded that this discovery would rewrite maritime
↪  history textbooks.
Output result: Here we have "dozen" as a quantifier,
↪  "concluded" and "rewrite" both functioning as verbs, with
↪  "maritime" classified as an adjective.

Input
Enter a sentence: {sentence}
Output result:
```

D.2 Dependency Analysis

```
You are a linguist expert proficient in syntax and sentence
↪  structure.

Task:
Conduct a syntactic dependency analysis for the given sentence
↪  and translate it into plain English. First, identify the
↪  root verb of the sentence. Then, describe the syntactic
↪  relationships of key words (especially governors) in a
↪  single, fluent natural language sentence.

Core requirements:
Focus on relationships using natural language: Your analysis
↪  must concentrate on the functional relationships between
↪  words. Describe these connections using plain English terms
↪  (e.g., "grammatical subject", "direct object", "adjective
↪  describing", "adverbial clause") and strictly avoid using
↪  technical dependency abbreviations or symbols (such as
↪  nsubj, obj, acl, advcl).
Imitate the style of the examples: Strictly follow the
↪  narrative, fluent output format and language style of the
↪  provided examples.

Examples:
Enter a sentence: Beneath the archaeologist's trembling
↪  fingers, the bronze astrolabe revealed intricate patina
↪  patterns, its engravings whispering of celestial
↪  navigation.
Output result: The syntactic structure centers on the verb
↪  "revealed", where "astrolabe" acts as the grammatical
↪  subject performing the action upon the direct object
↪  "patterns", with "intricate" serving as an adjective
↪  describing the patterns and "bronze" acting as an adjective
↪  characterizing the astrolabe. The prepositional phrase
↪  "Beneath... fingers" provides spatial context through its
↪  preposition, while the participial phrase "whispering..."
↪  functions as an adverbial clause with "engravings" as its
↪  subject, linked to "navigation" which acts as a noun
↪  modifier.

Enter a sentence: "Astounding!" murmured Dr. Liang, carefully
↪  rotating the artifact until its azimuthal markings aligned
↪  with the noon sun.
Output result: In this construction, "murmured" serves as the
↪  main predicate with "Dr. Liang" as the subject carrying out
↪  the action, preceded by the exclamative "Astounding!"
↪  which acts as a discourse marker. The participial phrase
↪  "rotating..." acts as an adverbial clause described by the
↪  adverb "carefully", while "until" functions as a linking
↪  word that introduces a temporal clause. In this clause,
↪  "aligned" is the verb, taking "markings" as its
↪  subject—which is specified by the adjective
↪  "azimuthal"—and relates spatially through the preposition
↪  "with", while "its" indicates possession.

Enter a sentence: Twelve graduate students clustered around the
↪  excavation pit, their calipers and brushes poised to
↪  document every millimeter.
Output result: The clause structure shows "clustered" as the
↪  central action performed by the subject "students",
↪  quantified by the number "Twelve" and qualified by the
↪  adjective "graduate". The preposition "around" establishes
↪  a spatial relationship with the main verb, while the
↪  absolute construction "their calipers... poised" functions
↪  as an adverbial clause wherein "poised" is the predicate.
↪  "Calipers" and "brushes" serve as its coordinated subject
↪  linked by the conjunction "and", which then takes an
↪  infinitival purpose clause "to document", with "every"
↪  serving as a determiner specifying the word "millimeter".

Enter a sentence: The relic emitted a faint metallic ping when
↪  touched, suggesting remarkably preserved internal
↪  mechanisms.
Output result: This sentence features "emitted" as the core
↪  verb with "relic" as the main subject, producing the sound
↪  "ping" as its direct object, which is characterized by the
↪  adjectives "faint" and "metallic". The temporal clause
↪  "when touched" serves as an adverbial phrase modifying the
↪  main action, while the participial "suggesting" introduces
↪  a secondary adverbial clause about "mechanisms". This is
↪  intensified by the adverb "remarkably" and modified by the
↪  descriptive participle "preserved", with "internal" acting
↪  as an adjective describing the mechanisms.

Enter a sentence: After recording three dozen measurements, the
↪  team concluded that this discovery would rewrite maritime
↪  history textbooks.
```

Benchmark	Evaluation focus
BoolQ	Yes/no question answering over naturally occurring passages, testing evidence localization and reading comprehension.
CB	Textual entailment under naturally occurring discourse, especially inference over embedded clauses and speaker commitment.
PAWS-X	Cross-lingual paraphrase identification with adversarial sentence pairs that have high lexical overlap but different word order or meaning.
RTE	Textual entailment, testing whether a hypothesis can be inferred from a premise.
SST-2	Binary sentiment classification, testing sentence-level semantic composition and polarity.
XNLI	Cross-lingual natural language inference over entailment, contradiction, and neutral relations.
LogicBench	Systematic logical reasoning across formal and natural-language reasoning patterns.
LogiQA2.0	Multiple-choice logical reasoning in natural language understanding, requiring inference over arguments and conditions.
BigBench	Mixed comprehensive evaluation over 73 multiple-choice tasks following BRD.
LongBench v2	Long-context understanding and reasoning over realistic multitask inputs requiring evidence integration across extended documents.

Table 7: Descriptions of the downstream evaluation benchmarks.

```
Output result: The sentence structure positions "concluded" as
↪  the main verb with "team" as its subject. The introductory
↪  clause "After recording..." functions as an adverbial
↪  phrase where "measurements" is quantified by the numeric
↪  modifier "three dozen". The verb "concluded" takes a
↪  dependent clause as its object, introduced by "that", where
↪  "discovery" serves as the subject for the verb phrase
↪  "would rewrite" (with "would" acting as a helping verb).
↪  This verb phrase acts upon the complex noun phrase
↪  "maritime history textbooks", where "maritime" acts as an
↪  adjective modifying "history" within this compound direct
↪  object.

Enter a sentence: "Colleagues," announced Dr. Chen excitedly,
↪  "these fragments here - the ones we thought were
↪  unimportant - they actually prove the astrolabe's use in
↪  Polynesian navigation!"
Output result: This passage presents "announced" as the main
↪  verb with "Dr. Chen" as the subject, modified by the adverb
↪  "excitedly" and taking the direct speech as its dependent
↪  clause object. The word "Colleagues" is a direct address to
↪  the audience, while the direct speech features the verb
↪  "prove". The subject of "prove" is "they", which acts as a
↪  pronoun referring back to the emphasized, shifted noun
↪  phrase "these fragments here". The verb "prove" takes "use"
↪  as its direct object, which is modified by the possessive
↪  word "astrolabe's" and the prepositional phrase "in...
↪  navigation", while "here" functions as an adverb
↪  indicating location.

Input
Enter a sentence: {sentence}
Output result:
```

D.3 Semantic Role Labeling

```
You are a linguist expert proficient in Semantic Role Labeling
↪  (SRL).
Task:
Conduct a Semantic Role Labeling analysis for the given
↪  sentence. First, identify the core predicate(s), then
↪  describe the semantic relationships between the
↪  predicate(s) and their arguments in a single, fluent
↪  natural language sentence.

Core requirements:
Focus on semantic functions: Your analysis should concentrate
↪  on the semantic relationships, describing the role of each
↪  phrase in natural language (e.g., the performer of an
↪  action, the entity being affected, the location, time, or
↪  manner of the event).
Identify the core event(s): Accurately identify the main
↪  predicate(s) (typically verbs) that describe the core
↪  events or states.
Describe all arguments: For each event, identify and describe
↪  all related phrases, explaining their semantic
↪  contribution, such as who or what is involved, as well as
↪  the circumstances of the event (location, time, manner,
↪  purpose, etc.).
Imitate the style of the examples: Strictly follow the output
↪  format and language style of the following examples,
↪  integrating the full analysis into a single sentence.

Examples:
Enter a sentence: Beneath the archaeologist's trembling
↪  fingers, the bronze astrolabe revealed intricate patina
↪  patterns, its engravings whispering of celestial
↪  navigation.
Output result: The analysis identifies 'revealed' as the main
↪  predicate, where 'the bronze astrolabe' is the entity
↪  performing the action and 'intricate patina patterns' is
↪  the entity being revealed, with the event occurring at the
↪  location 'Beneath the archaeologist's trembling fingers';
↪  a secondary event is introduced by the predicate
↪  'whispering', where 'its engravings' acts as the performer
↪  and 'of celestial navigation' is the subject matter.
Enter a sentence: "Astounding!" murmured Dr. Liang, carefully
↪  rotating the artifact until its azimuthal markings aligned
↪  with the noon sun.
Output result: In this sentence, the primary event is
↪  'murmured', with 'Dr. Liang' identified as the speaker and
↪  '"Astounding!"' as the content of the speech; this action
↪  is described with a co-occurring event, 'rotating', which
↪  affects 'the artifact' and is modified by 'carefully' to
↪  indicate the manner, leading to the final predicate
↪  'aligned', in which 'its azimuthal markings' is the entity
↪  that aligns.
Enter a sentence: Twelve graduate students clustered around the
↪  excavation pit, their calipers and brushes poised to
↪  document every millimeter.
Output result: This sentence centers on the predicate
↪  'clustered', where 'Twelve graduate students' are the ones
↪  performing the action and 'around the excavation pit'
↪  specifies the location; this is followed by a state
↪  described by the predicate 'poised', where 'their calipers
↪  and brushes' is the entity in that state, positioned with
↪  the purpose 'to document' an action that would affect
↪  'every millimeter'.
Enter a sentence: The relic emitted a faint metallic ping when
↪  touched, suggesting remarkably preserved internal
↪  mechanisms.
Output result: The core event here is 'emitted', with 'The
↪  relic' being the source of the emission and 'a faint
↪  metallic ping' being what is emitted, occurring at the time
↪  defined by 'when touched'; this main event has a
↪  consequence expressed by the secondary predicate
↪  'suggesting', which presents 'remarkably preserved
↪  internal mechanisms' as the content of the suggestion.
Enter a sentence: After recording three dozen measurements, the
↪  team concluded that this discovery would rewrite maritime
↪  history textbooks.
Output result: The main predicate is 'concluded', where 'the
↪  team' is the entity that concluded something and the
↪  content of their conclusion is the entire clause 'that this
↪  discovery would rewrite maritime history textbooks'; this
↪  action occurs at a time following the event 'After
↪  recording three dozen measurements', where 'recording' is
↪  a secondary predicate and 'three dozen measurements' is
↪  what was recorded.
Enter a sentence: "Colleagues," announced Dr. Chen excitedly,
↪  "these fragments here - the ones we thought were
↪  unimportant - they actually prove the astrolabe's use in
↪  Polynesian navigation!"
```

```
Output result: This construction features 'announced' as the
↪  main predicate, with 'Dr. Chen' as the speaker, 'excitedly'
↪  describing the manner of the announcement, and the direct
↪  speech being the content of the announcement; within the
↪  utterance, the core event is 'prove', where 'they'
↪  (referring to 'these fragments') is the agent performing
↪  the action and 'the astrolabe's use in Polynesian
↪  navigation' is the concept being proven.

Input
Enter a sentence: {sentence}
Output result:
```

D.4 Coreference Resolution

```
Task:
Read the provided paragraph and identify a meaningful
↪  coreferential link between two or more words or phrases.
Formulate a clear question that tests the understanding of this
↪  link that go beyond simple pronouns (like "he," "it,"
↪  "they"), asking what a specific word or phrase refers to.
The answer must be a phrase copied directly from the paragraph,
↪  not a paraphrase.

Output Format:
Question: [Your question about the coreferential link]
Answer: [The corresponding mention from the text]

Examples:
Example 1 (Entity & Description):
Input Text: "Project Phoenix was a secret government initiative.
↪  This clandestine operation involved three agents."
Output:
Question: What does the phrase "This clandestine operation"
↪  refer to?
Answer: Project Phoenix

Example 2 (Event & Reference):
Input Text: "The team completed the final phase of their
↪  project. This major milestone was celebrated by everyone."
Output:
Question: What does the phrase "This major milestone" refer to?
Answer: The team completed the final phase of their project

Input Text: "{sentence}"
Output:
```

D.5 Discourse Relation Identification

```
You are a linguist expert proficient in Discourse Relation
↪  Identification (DRI).
Task:
Conduct a discourse relation analysis for the given paragraph.
↪  First, segment the paragraph into sentences or clauses that
↪  function as discourse units, then identify the discourse
↪  connective(s) or implicit relation(s) that link them.
↪  Finally, describe the discourse relation(s) between the
↪  connected units in a single, fluent natural language
↪  sentence.

Core requirements:
Focus on discourse functions: Your analysis should explain the
↪  logical or rhetorical relation (e.g., cause, contrast,
↪  elaboration, temporal sequence, condition, concession)
↪  between discourse segments.
Identify all relations: Accurately identify both explicit and
↪  implicit discourse relations within the paragraph.
Describe the linked segments: For each relation, specify which
↪  part serves as premise, cause, background, elaboration, or
↪  result, etc.
Output style: Imitate the examples below by integrating the
↪  full analysis into one fluent natural language sentence,
↪  not as a bullet list.

Examples:
Enter a paragraph: The storm intensified overnight. As a
↪  result, several flights were canceled, and passengers were
↪  forced to stay in the terminal. Some travelers, however,
↪  managed to rebook quickly and left the airport by dawn.
Output result: This paragraph contains a causal relation marked
↪  by 'As a result', where 'The storm intensified overnight'
↪  provides the cause and 'several flights were canceled, and
↪  passengers were forced to stay in the terminal' presents
↪  the consequence; additionally, a contrastive relation is
↪  introduced by 'however', where 'Some travelers managed to
↪  rebook quickly and left the airport by dawn' contrasts with
↪  the situation of passengers stranded in the terminal.
Enter a paragraph: The researchers first examined the artifact
↪  under natural light. Then they applied infrared scanning to
↪  reveal hidden engravings. Finally, they compared the
↪  results with historical records to confirm its
↪  authenticity.
Output result: This paragraph illustrates a temporal and
↪  procedural relation, where 'examined under natural light'
↪  occurs first, 'applied infrared scanning' follows as the
↪  next step, and 'compared with historical records' is the
↪  final step, showing a chronological progression of
↪  research actions.
Enter a paragraph: The project was a resounding success.  The
↪  team worked tirelessly for months, overcoming every
↪  obstacle they faced.
Output result: This paragraph demonstrates an implicit
↪  relation of explanation, where the initial assertion 'The
↪  project was a resounding success' serves as a conclusion,
↪  and the subsequent statement 'The team worked tirelessly
↪  for months... ' provides the underlying justification for
↪  this outcome;  the strong causal connection between the two
↪  is inferred by the reader without being marked by an
↪  explicit connector like 'because' or 'since'.
Enter a paragraph: Scientists discovered a new fossil in the
↪  desert, which suggests that mammals evolved earlier than
↪  previously thought.   For example, the fossilized jawbone
↪  contains teeth adapted for chewing plants.   However, some
↪  researchers argue that the dating of the rock layers may be
↪  unreliable, which could undermine the claim.
Output result: This paragraph expresses multiple discourse
↪  relations: a contingency relation is signaled by 'which
↪  suggests', where 'Scientists discovered a new fossil in the
↪  desert' provides the cause and 'mammals evolved earlier
↪  than previously thought' is the inferred result;   an
↪  expansion relation of exemplification is introduced by 'For
↪  example', where 'the fossilized jawbone contains teeth
↪  adapted for chewing plants' serves as specific supporting
↪  evidence;   finally, a comparison relation of concession
↪  is marked by 'however', where 'some researchers argue that
↪  the dating of the rock layers may be unreliable' contrasts
↪  with and challenges the earlier claim.

Input:
Enter a paragraph: {sentence}
Output result:
```

D.6 Discourse-Aware Summarization

```
You are a linguist expert proficient in Discourse-Aware
↪  Summarization.
Task:
Read the given passage carefully and produce a concise summary
↪  that captures the main ideas and logical flow of the text.
↪  Your summary must preserve not only the factual content but
↪  also the discourse structure, such as causal relations,
↪  contrasts, elaborations, and temporal sequences, so that
↪  the rhetorical intent of the original passage is
↪  maintained.

Core requirements:
Faithful condensation: Retain key information and main claims
↪  while omitting details that are minor or redundant.
Discourse sensitivity: Reflect the logical and rhetorical
↪  relations (e.g., cause-effect, contrast, elaboration,
↪  concession, sequence) among ideas when rewriting.
Fluent output: Write in natural, coherent sentences, not bullet
↪  points, ensuring the summary reads like a connected short
↪  passage rather than a list.
Length control: Aim for a summary that is significantly shorter
↪  than the original passage (about 25-35% of the original
↪  length).

Examples:
Input passage: "The storm intensified overnight.   As a result,
↪  several flights were canceled, and passengers were forced
↪  to stay in the terminal.   Some travelers, however, managed
↪  to rebook quickly and left the airport by dawn."
Output summary: "As the storm grew stronger overnight, flights
↪  were canceled and many passengers were stranded, though a
↪  few managed to rebook and depart by dawn."
Input passage: "The researchers first examined the artifact
↪  under natural light.   Then they applied infrared scanning
↪  to reveal hidden engravings.   Finally, they compared the
↪  results with historical records to confirm its
↪  authenticity."
Output summary: "The researchers analyzed the artifact step by
↪  step—using natural light, infrared scanning, and finally
↪  historical comparison—to establish its authenticity."
```

```
Input passage: {sentence}
Output summary:
```

D.7 Document-Level Question Answering

```
Task:
Carefully read the provided chapter (Input Text: ). Generate 3
↪  to 4 reasoning-based question-answer (QA) pairs that
↪  assess a chapter-level understanding of the text. Each QA
↪  must require connecting information and include a short
↪  explanation.

Requirements:
1.Generate 3-4 QA pairs (if the text is too short, at least 2,
↪  but aim for 3).
2.Each QA must represent a different reasoning type from the
↪  following list, focusing on connecting stated information:
Integration of Key Details: Combine distinct facts/details
↪  from different parts of the text to answer a "how" or
↪  "what" question.
Direct Cause-Effect Chain: Trace a cause-and-effect
↪  relationship that may span across multiple sentences or
↪  paragraphs.
Comparison of Concepts: Compare two concepts, methods, or ideas
↪  explicitly mentioned in the text.
3.Crucially, each question must require information from at
↪  least two different sentences that are not directly
↪  adjacent. This ensures a true chapter-level focus.
4.Questions must not be yes/no and must not ask for simple
↪  restatements. Prefer "why/how/in what way/compare/what if"
↪  formulations.
5.Answers should be concise (15-40 words).
6.Each QA must include an ""Explanation"" field (1-2 sentences)
↪  describing the reasoning steps.
7.Follow the exact output format below (separate entries with
):

Output Format :
ID: <1..N>
Reasoning Type: <one of the seven types above>
Question: <reasoning question>
Answer: <15-40 words>
Explanation: <1-2 sentences explaining reasoning>

Input Text: "{sentence}"
Output:
```

D.8 High-Order Reasoning Question Answering

```
Task:
Carefully read the provided chapter (Input Text: ). Generate 3
↪  to 4 advanced reasoning-based question-answer (QA) pairs
↪  that require a deep, holistic understanding of the
↪  chapter's arguments and implications. Each QA must include
↪  a short explanation.

Requirements:
1.Generate 3-4 QA pairs (if the text is too short, at least 2,
↪  but aim for 3).
2.Each QA must represent a different high-order reasoning type
↪  from the following list:
Synthesis of Concepts: Synthesize information from across the
↪  entire text to explain a core theme, tension, or
↪  overarching conclusion.
Evaluation of Argument/Claim: Question the strength,
↪  implication, or underlying assumption of a key argument
↪  presented by the author.
Implicit Inference: Infer a logical consequence, author's
↪  purpose, or the significance of a finding that is implied
↪  but not explicitly stated.
3.Crucially, questions must probe the "why" and "so what"
↪  behind the text's information. They should focus on
↪  implications, assumptions, or the synthesis of a novel idea
↪  not directly stated in any single sentence.
4.Questions must not be yes/no. Prefer "What is the
↪  significance of.../How does the author challenge.../What
↪  is the underlying assumption behind..." formulations.
5.Answers should be concise (15-40 words) and reflect deep
↪  inference.
6.Each QA must include an "Explanation" field (1-2 sentences)
↪  describing the inferential leap or synthesis process.
7.Follow the exact output format below (separate entries with
):

Output Format :
ID: <1..N>
Reasoning Type: <one of the seven types above>
Question: <reasoning question>
Answer: <15-40 words>
Explanation: <1-2 sentences explaining reasoning>

Input Text: "{sentence}"
Output:
```