Scaling LLM Knowledge Boundaries via Distribution-Optimized Synthesis

Songze Li[1,3], Yarong Lan[1,3], Zhongpu Bo[2], Zhaoyang Wang[2],
Zhiqiang Liu[1], Yuan Yuan[1], Chengtao Gan[1], Menghao Qian[1], Enpei Niu[1],
Xiaoke Guo[1], Yuanxiang Liu[1], Zhaoyan Gong[1,3], Xiangjin Hu[1,3], Liangyurui Liu[1], Jingdian Lu[1,3],
Lei Liang[2], Jun Zhou[2], Huajun Chen[1], Wen Zhang[1,3]*
[1]Zhejiang University, [2]Ant Group, [3]ZJU-Ant Group Joint Lab of Knowledge Graph
{li.songze,zhang.wen}@zju.edu.cn

Abstract

Knowledge injection via synthetic data is crucial for enhancing Large Language Models (LLMs). However, current synthesis methods simply stop at preset token counts or fixed data ratios, lacking awareness of knowledge distribution. This results in some domains being sparse while others are redundant, limiting LLM knowledge boundaries. We revisit knowledge injection from a distribution perspective and hypothesize that an optimal knowledge distribution exists to maximize knowledge boundary expansion. We propose **KDoS** (**K**nowledge **D**istribution-**o**ptimized **S**ynthesis), a framework that introduces knowledge density to drive synthesis through a three-stage feedback mechanism, shifting from blind generation to distribution-optimized synthesis. We construct Wikipedia-based synthetic data with varying knowledge distributions and conduct experiments on models from 0.6B to 16B (Qwen, Ling, LLaMA) and data scales from 1B to 5B tokens. Our key findings are: (1) an optimal knowledge distribution consistently maximizes boundary expansion; (2) this distribution is stable across backbones and scales; (3) KDoS outperforms baselines across six knowledge benchmarks. Our work offers a new perspective and practical framework for synthetic data-driven knowledge injection.

1 Introduction

The knowledge boundary of an LLM defines the scope of knowledge it can reliably handle (Li et al., 2025), serving as a core dimension for assessing model capabilities (Zhao et al., 2025; Yin et al., 2023). However, even state-of-the-art LLMs exhibit knowledge boundary limitations—particularly on long-tail knowledge (Kandpal et al., 2023) such as infrequent Wikipedia facts—where low-frequency but factually grounded questions cannot be reliably answered (Sun et al., 2024; Mallen et al., 2023). Scaling up LLM knowledge boundaries through efficient knowledge injection has thus become an important research challenge (Guu et al., 2020; Allen-Zhu and Li, 2024).

Synthetic data offers a flexible, scalable, and cost-effective way to target specific knowledge domains (Ke et al., 2023), alleviating long-tail coverage gaps in real data, and has been widely adopted in continual pre-training, instruction tuning, and other knowledge injection settings (Sun et al., 2023; Yang et al., 2024). However, existing methods share a fundamental limitation: they simply stop at preset token counts or fixed data ratios, with no awareness or control over knowledge distribution (Azerbayev et al., 2024; Ren et al., 2025). This constitutes a blind-synthesis paradigm: methods neither perceive the current knowledge distribution nor understand which distribution optimizes injection efficiency. As a result, some knowledge points are redundantly repeated while others remain critically sparse (Havrilla et al., 2024), directly constraining the model's knowledge boundary (Xie et al., 2023). As illustrated in Fig. 1, training the same model with and equal number of tokens but different knowledge distributions leads to a loss difference of up to 24.6% on knowledge QA benchmarks, confirming that knowledge distribution is a key factor in injection effectiveness (Penedo et al., 2024).

To address this limitation, we revisit LLM knowledge injection from a distribution perspective and propose a core hypothesis: *there exists at least one optimal knowledge distribution that maximizes knowledge boundary expansion*. Based on this, we propose **KDoS** (**K**nowledge **D**istribution-**o**ptimized **S**ynthesis), which shifts synthetic data generation from blind synthesis to targeted distribution shaping. KDoS introduces knowledge density as a controllable proxy for knowledge distribution, operating through three iterative stages: (1) extracting and extending knowledge points from

* Corresponding authors.

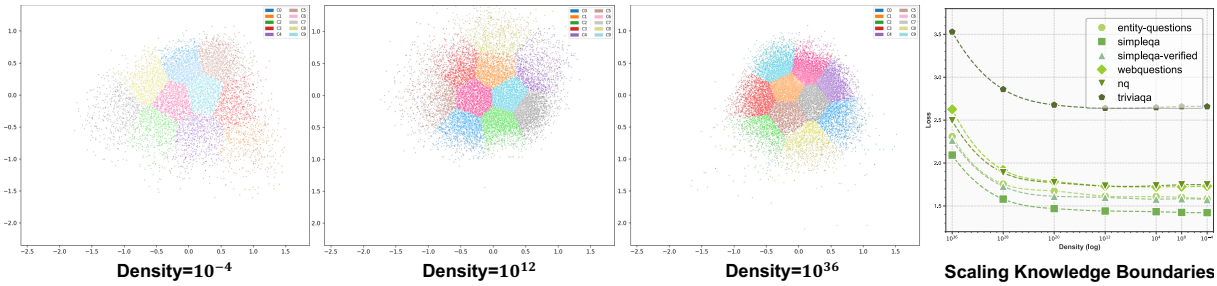


Figure 1: Scaling LLM Knowledge Boundaries with Different Distributions. We present scatter plots of three different knowledge densities (10^{-4}, 10^{12}, and 10^{36}) (computed via Eq. 1), along with eval loss scaling curves on six knowledge benchmarks. The loss gap between 10^{-4} and 10^{36} reaches up to 24.6%.

seed knowledge, organizing semantically related samples into knowledge groups, and generating new questions from knowledge point combinations within each group; (2) quality filtering of candidate questions; (3) rejection sampling over candidates based on a target knowledge distribution. KDoS continuously monitors the knowledge distribution of the data pool and dynamically adjusts the acceptance strategy, iteratively driving the distribution toward the preset target.

Our main contributions are as follows:

- **Problem Perspective.** We reframe knowledge injection as a knowledge distribution control problem, identifying the lack of distribution awareness as the fundamental limitation of existing methods, and offering a new research perspective for the field.

- **Methodological Innovation.** We propose KDoS, which introduces knowledge density as a controllable variable for knowledge distribution, and employs a dynamic feedback mechanism to precisely shape the distribution of synthetic data, enabling continuous scaling of LLM knowledge boundaries.

- **Experimental Insights and Validation.** Through systematic experiments across LLMs from 0.6B to 16B and varying data scales, we confirm our hypothesis and find that the optimal knowledge density consistently exists across different backbones and data scales, manifesting as 10^{-4}~10^{4} on our data and revealing a general principle of knowledge injection. Extensive experiments further validate that KDoS consistently outperforms existing baselines across six knowledge benchmarks.

2 Related Work

LLM Knowledge Injection. Scaling up the parametric knowledge boundaries of LLMs is a core challenge for improving their fundamental capabilities. Knowledge injection approaches include continual pre-training, supervised fine-tuning (SFT), and retrieval-augmented generation. ADEPT (Zhang et al., 2025) and LLaMA-Pro (Wu et al., 2024) extend model architecture for continual pre-training, but (Lv et al., 2025) identify a "memory collapse" threshold in knowledge injection, revealing inherent limitations of pre-training approaches. (Ovadia et al., 2024) show that retrieval-augmented methods can outperform fine-tuning in certain scenarios without training, while knowledge editing methods such as ROME (Meng et al., 2022a) and MEMIT (Meng et al., 2022b) face scalability bottlenecks under large-scale updates. As for LLM knowledge evaluation, knowledge-intensive benchmarks such as TriviaQA (Joshi et al., 2017), Natural Questions (Kwiatkowski et al., 2019), and WebQuestions (Talmor and Berant, 2018) primarily assess common factual knowledge. SimpleQA (Wei et al., 2024), SimpleQA-Verified (Haas et al., 2025), and Entity Questions (Sciavolino et al., 2021) target long-tail knowledge, where even state-of-the-art models perform poorly. (Kandpal et al., 2023) further show that QA performance strongly correlates with document frequency in pre-training data, confirming that LLMs systematically struggle with long-tail knowledge. These studies highlight that existing knowledge injection methods leave significant knowledge gaps in long-tail settings such as Wikipedia, and efficiently expanding LLM knowledge boundaries remains an open problem.

Data Synthesis. For general-purpose data synthesis, Self-Instruct (Wang et al., 2023), Evol-Instruct (Xu et al., 2023), and Magpie (Xu et al., 2025)

establish foundational paradigms for instruction synthesis, but all terminate synthesis at a human-specified token count without considering data distribution. For quality and distribution control, STaR (Zelikman et al., 2022), RFT (Yuan et al., 2023), DART-Math (Tong et al., 2024), DEITA (Liu et al., 2024), and TreeSynth (Wang et al., 2025) improve synthesis from the perspectives of reasoning, sampling, and diversity, yet still lack explicit modeling of knowledge distribution. For knowledge-aware synthesis, GraphGen (Chen et al., 2025b) and CodecLM (Wang et al., 2024) incorporate knowledge graphs and metadata to guide generation, but neither actively controls knowledge distribution during synthesis. (Qin et al., 2025) identify performance bottlenecks in synthetic data but offer no mechanism to dynamically adjust knowledge distribution. In summary, existing data synthesis methods remain fundamentally blind to knowledge distribution. We therefore propose KDoS, which scales LLM knowledge boundaries by optimizing the knowledge distribution of synthetic data.

3 Methods

3.1 Preliminary

Knowledge Density Definition. We denote the data pool as $\mathcal{S} = (T, \rho)$, where T and ρ represent the token count and knowledge density of $\mathcal{S}$, respectively. Following (Chen et al., 2025a), we define ρ as:

$$\rho = \frac{T}{V} = \frac{T \cdot \Gamma(n/2+1)}{\pi^{n/2} \cdot r^n}, \tag{1}$$

where V is the volume of the n-dimensional hypersphere formed by $\mathcal{S}$ in semantic space, and r is the average radius, i.e., the mean distance from all samples to the centroid. A higher ρ indicates that more knowledge is concentrated in a smaller semantic region, while a lower ρ indicates sparser coverage over a broader semantic space.

Problem Definition. Given a seed question pool $\mathcal{S}^{\text{seed}}$, a synthesis method $\mathcal{A}$ produces synthetic data $\mathcal{S}^{\text{syn}}$ (i.e., $\mathcal{A}\left(\mathcal{S}^{seed}, \mathcal{D}^{target}\right) \rightarrow \mathcal{S}^{syn}$), which is used to train LLM $\mathcal{M}$ and evaluated on test set. The goal is to maximize the test accuracy $\text{Acc}(\mathcal{M}, \mathcal{S}^{\text{syn}})$ to achieve optimal knowledge injection and expand the knowledge boundary of $\mathcal{M}$. Formally:

$$\mathcal{D}^* \sim \underset{\mathcal{D}}{argmax}\mathbb{E}_{P(\mathcal{S}^{syn})\sim\mathcal{D}}\left[Acc\left(\mathcal{M}, \mathcal{S}^{syn}\right)\right], \tag{2}$$

where $\mathcal{D}^*$ is the optimal knowledge distribution. Given a target token count T^{target} and target density ρ^{target}, KDoS iteratively drives the data pool to converge to the target distribution $\mathcal{D}^{\text{target}}$ over k iterations:

$$\begin{aligned} \mathcal{D}_{t+1} \leftarrow \text{Update}\left[\mathcal{D}_t, \mathcal{A}_t\left(\mathcal{S}_t, \mathcal{D}^{\text{target}}\right)\right], \\ \text{s.t.} \quad \lim_{t\to k} \mathcal{D}_t = \mathcal{D}^{\text{target}} \end{aligned} \tag{3}$$

3.2 Overview of Data Synthesis and Verification

Our work focuses on scaling LLM knowledge boundaries and consists of three stages: **Stage 1: Seed Pool Preparation**, which collects and processes raw data; **Stage 2: Knowledge Distribution-Optimized Synthesis (KDoS)**, our proposed framework for distribution-optimized synthesis; **Stage 3: Experimental Verification**, which evaluates the performance of different knowledge distributions $\mathcal{D}$ on LLM knowledge injection.

Seed Pool Preparation. To support large-scale data synthesis, we collect raw documents from Wikipedia. After data cleaning, we summarize knowledge points from each document and synthesize seed questions whose answers are directly grounded in the source documents. This step yields approximately 14M seed QA pairs. Details are provided in Appendix E.1.

Knowledge Distribution-Optimized Synthesis. Based on the seed question pool from Stage 1, KDoS controls the synthesis process according to preset T^{target} and ρ^{target}, driving the final synthetic data $\mathcal{S}^{\text{syn}} = (T^{\text{target}}, \rho^{\text{target}})$ to conform to the target distribution $P(\mathcal{S}^{\text{syn}}) \sim \mathcal{D}^{\text{target}}$. This is described in detail in Section 3.3.

Experimental Verification. We evaluate synthetic data of varying knowledge distributions from Stage 2 on six knowledge benchmarks, examining how LLM knowledge boundaries scale with knowledge distribution, identifying the optimal knowledge density range, and exploring general principles of knowledge injection across multiple model sizes, data scales, and backbones.

3.3 KDoS Framework

KDoS operates through three iterative stages. First, KDoS extracts a knowledge point list and knowledge logic chain for each question, maps them to a semantic space to form knowledge groups based on semantic proximity, and synthesizes new candidate questions via knowledge combination. Second, KDoS applies quality filtering to remove low-quality samples. Third, KDoS uses rejection sam-

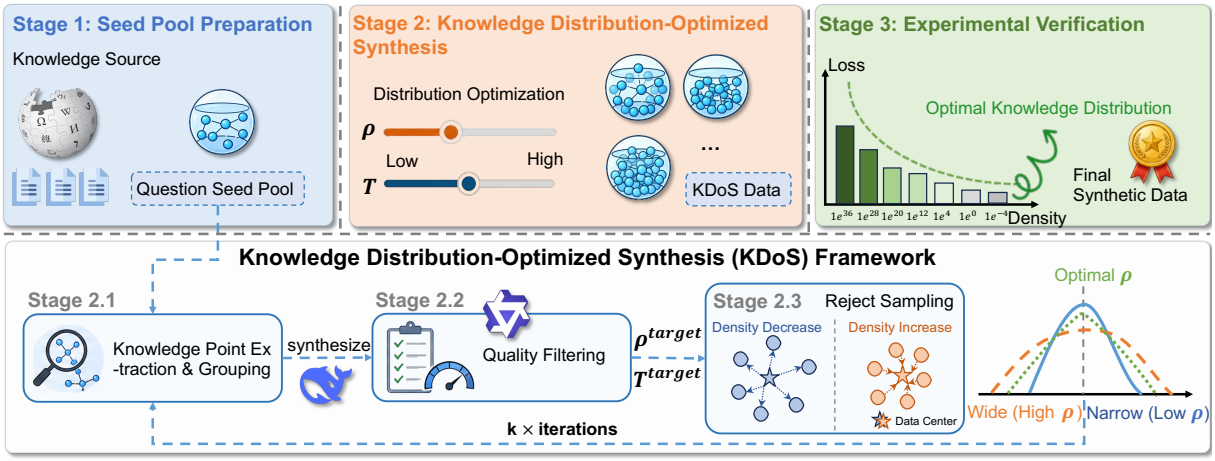


Figure 2: **Top:** Overview of our data synthesis and verification pipeline. **Bottom:** Overview of the **KDoS** (**K**nowledge **D**istribution-**o**ptimized **S**ynthesis) framework.

pling to preferentially select samples that drive the data pool toward the target distribution, iterating until convergence.

3.3.1 Knowledge Point Extraction & Grouping

For each question in the seed pool $\mathcal{S}^{\text{seed}}$, we use DeepSeek V3.2 (DeepSeek-AI, 2025) to extract a knowledge point list and a knowledge logic chain. The knowledge point list captures the relevant knowledge required to answer the question, while the knowledge logic chain provides an explicit representation of the logical relationships among knowledge points. Based on the knowledge point lists, we apply n-gram ($n = 5$) deduplication on knowledge point lists, retaining only one question among those with similar knowledge points. We also perform overlap detection between the current data pool and the test set to exclude samples that may cause test set leakage. We then map each knowledge point list to a semantic space using the embedding model `sentence-transformers/all-MiniLM-L6-v2` (Reimers and Gurevych, 2019), which enables grouping semantically similar samples into knowledge groups. Each knowledge group consists of a sample and its two nearest neighbors in semantic space, forming a group of 3 samples. For each group, we use DeepSeek V3.2 to semantically extend the knowledge points within the group (i.e., extending new related knowledge points from the model's parametric knowledge), and then combine knowledge points across the group to synthesize 5 new candidate questions. This promotes greater knowledge diversity, breaks the knowledge boundary of individual seed questions, and broadens knowledge coverage. Examples of knowledge point lists, logic chains, and knowledge groups are provided in Appendix E.2; prompts for knowledge extraction, semantic extension, and question synthesis are in Appendix B.1, B.2.

3.3.2 Quality Filtering

Synthesized candidate questions suffer from issues like ambiguous intent, meaningless content, hallucinated answers, incorrect knowledge points, etc. To address this, we use DeepSeek V3.2 and Qwen3.5-397B-A17B (Qwen-Team, 2026) as LLM judges in a two-step evaluation.

Preliminary Check. Each sample undergoes three binary tests: **Answer Independence** verifies that the answer is not directly inferable from the question itself; **Answer Verifiability** requires objective and verifiable answers; **Answer Correctness** filters out factual or common-sense errors. Samples failing any criterion are discarded immediately.

Scoring. Samples passing the preliminary check are scored across five dimensions (total: 12 points): **Educational Significance (0~4)** penalizes trivial or content-free questions; **Specificity and Concreteness (0~2)** encourages instance-level questions over abstract ones; **Internal Question Logic (0~2)** checks coherence of the question itself; **Question-Answer Logic (0~2)** ensures the answer logically follows from the question; **Knowledge-Point Relevance & Logic Completeness (0~2)** verifies that the associated knowledge points are relevant and form a complete reasoning chain. Samples scoring

zero on any dimension are excluded, and only those with an average score ≥ 8 from both judges are retained. The LLM judge prompt is in Appendix B.3.

3.3.3 Reject Sampling

In this stage, KDoS applies rejection sampling to iteratively drive the current distribution $\mathcal{D}_t$ toward $\mathcal{D}^{\text{target}}$. The process consists of two phases: **Cold-start phase** ($T < T^{\text{target}}$): The goal is to accumulate data volume. Samples passing quality filtering are directly added to the data pool $\mathcal{D}_t$ without any density constraint. **Density fine-tuning phase** ($T \to T^{\text{target}}$): As the token count approaches T^{target}, the process transitions to density fine-tuning, with knowledge density as the control target. The acceptance strategy is as follows: using the density formula in Sec. 3.1, we back-calculate r^{target} from T^{target} and ρ^{target}. If the current $r < r^{\text{target}}$ (density too high), we preferentially accept questions far from the centroid to increase r; if $r > r^{\text{target}}$ (density too low), we preferentially accept questions close to the centroid to decrease r. Stages 2.1~2.3 iterate until the data pool satisfies the convergence condition (We set the maximum number of iterations to k.):

$$\left|T - T^{target}\right| < \epsilon^{T}, \left|\rho - \rho^{target}\right| < \epsilon^{\rho} \qquad (4)$$

Finally, we obtain the data pool conforming to the target distribution $P(\mathcal{S}^{\text{syn}}) \sim \mathcal{D}^{\text{target}}$. The seed question pool of 14M is expanded to 71M synthetic samples. The detailed rejection sampling algorithm is provided in Appendix E.3.

4 Experiment

4.1 Experimental Settings

Datasets and Tasks. We evaluate on six knowledge benchmarks, divided into knowledge-intensive sets: **Web Questions (WebQ)**, **Natural Questions (NQ)**, and **TriviaQA (TriQA)**; and long-tail knowledge sets: **SimpleQA (Sim)**, **SimpleQA-Verified (Sim-V)**, and **EntityQuestions (EQ)**. Details of benchmarks are provided in Appendix A.

Baselines. We compare four synthesis strategies: **Rand.** (Random) applies no control and stops at target tokens; **Uni.** (Uniform) enforces equal ratios across domains; **Diff.** (Difficulty-weighted Importance Synthesis) prioritizes high-PPL samples to learn harder knowledge first; **Qual.** (Quality-filtered Rejection Synthesis) prioritizes high-quality samples from LLM judges.

Evaluation Metrics. We use accuracy and cross-entropy loss as evaluation metrics. For accuracy, we adopt an LLM-as-judge approach, using Qwen3.5-397B-A17B to classify each model prediction as Correct, Incorrect, or Not Attempted; accuracy is defined as the fraction of Correct samples. Cross-entropy loss measures the model's tendency to generate the correct answer, computed over the gold answer tokens.

Implementation Details. We collect approximately 14M seed QA pairs (1.73B tokens) from Wikipedia, which KDoS expands to 71M samples (9.28B tokens), with a maximum iteration count of $k = 200$. Data synthesis and quality filtering are conducted on an NVIDIA H20-3E cluster with 128 nodes $\times$ 8 GPUs (1024 H20-3E GPUs in total). Knowledge injection experiments are conducted on models including Qwen3.0-base (Qwen-Team, 2025), Ling-mini-2.0-base (Team et al., 2025), and LLaMA-3.2-base (Grattafiori et al., 2024), with Qwen3-4B-Base as the default backbone, using an NVIDIA H800 cluster with 8 nodes $\times$ 8 GPUs (64 H800 GPUs in total). More details are provided in Appendix C.

4.2 Main Result

Method	WebQ	TriQA	NQ	Sim	Sim-V	EQ	Avg.
Base	18.7	32.9	11.9	3.5	3.6	10.2	16.1
SP	23.0	33.4	13.3	5.8	6.8	11.1	17.4
Synthesis Strategy							
Rand.	26.1	32.8	18.0	6.0	6.7	12.1	18.3
Uni.	<u>27.9</u>	37.3	19.7	6.6	7.9	14.6	20.9
Qual.	26.9	36.3	19.8	6.3	6.8	14.8	20.6
Diff.	26.0	<u>38.5</u>	<u>19.9</u>	<u>7.2</u>	**8.2**	<u>15.2</u>	<u>21.5</u>
KDoS	**31.8**	**39.3**	**21.6**	**8.0**	<u>8.0</u>	**16.4**	**22.8**

Table 1: Performance comparison of KDoS and other synthesis strategies on six knowledge benchmarks, using Qwen3-4B-Base as the backbone. SP denotes Seed Pool. **Bold** and <u>underline</u> indicate the best and second-best results, respectively.

As shown in Tab. 1, the base model achieves an average score of 16.1 across six benchmarks. Training with the seed pool (SP) increases the average by 1.3 points. Among synthesis methods, Rand. achieves 18.3 but underperforms SP on some benchmarks, indicating that blind synthesis leads to uncontrolled distribution where some knowledge points are redundantly repeated

while others remain critically sparse, limiting LLM knowledge boundaries. Uni. improves to 20.9 by balancing domain ratios. Qual. achieves 20.6 and Diff. achieves 21.5, ranking second overall. In contrast, KDoS converges to the optimal knowledge distribution, achieving the best average of 22.8—1.3 points above Diff. and 1.9 points above Uni.—demonstrating the effectiveness of distribution-optimized synthesis.

4.3 Ablation Study

Benchmark	Module			
	KDoS	w/o F	w/o F & D	SP
EQ	16.4	15.5	11.5	11.1
Sim	8.0	7.0	5.8	5.8
Simp-V	8.0	7.2	6.6	6.8
WebQ	31.8	29.0	23.4	23.0
NQ	21.6	20.1	13.8	13.3
TriQA	39.3	38.4	33.5	33.4
Average	**22.8**	21.7	17.6	17.4

Table 2: Ablation study of KDoS modules.

We evaluate the effectiveness of the Quality Filtering (F) and Density-aware Rejection Sampling (D) modules. We compare four configurations: (1) full KDoS, (2) w/o F, (3) w/o F & D (i.e., only deduplication on the seed pool), and (4) the seed pool (SP) baseline. As shown in Tab. 2, removing F leads to a 1.07% drop in average score, while removing both F and D results in a 5.19% drop, validating the effectiveness of both modules. The results also indicate that distribution optimization plays the dominant role in performance improvement.

4.4 Scaling with Model and Data Size

We investigate the effect of knowledge density on knowledge injection across Qwen3-base models of varying sizes (0.6B∼14B) and data volumes (1B, 3B, 5B tokens). As shown in Fig. 3 and Fig. 4, all settings exhibit stable and consistent scaling curves, with the lowest eval total loss consistently achieved in the density range of 10^{-4}~10^{4}. As model size or data volume increases, the curves shift downward consistently. These results demonstrate that an optimal knowledge density range exists across all model and data scales, maximizing knowledge boundary scaling efficiency. All settings also exhibit a *Knowledge Collapse Region* where loss increases sharply at excessively high densities. Notably, larger models reach the *Knowledge Collapse Point* at lower densities (e.g., 10^{28} for 0.6B vs. 10^{12} for 14B), suggesting that larger models require finer-grained distribution control. Meanwhile, larger data volumes reach the *Knowledge Collapse Point* at higher densities (e.g., 10^{12} for 1B tokens vs. 10^{28} for 5B tokens), suggesting that different data scales require different levels of distribution control granularity. This may further indicate that the optimal density range varies across training stages: pre-training with larger data volumes may tolerate a wider optimal density range than post-training, likely because more data increases the absolute count of knowledge points in high-density regions, enabling the model to maintain learning efficiency in denser knowledge environments.

4.5 Scaling with LLM Backbones

We investigate the effect of knowledge density on knowledge injection across different backbone LLMs, including Qwen3-4B-Base, Ling-mini-2.0-16B-A3B, and LLaMA-3.2-3B. As shown in Fig. 5, regardless of whether the model is a dense or Mixture-of-Experts (MoE) architecture, all backbones exhibit stable and consistent scaling curves, with the lowest eval total loss consistently achieved in the density range of 10^{-4}~10^{4}. This is consistent with the findings in Sec. 4.4.

Scaling Principle of Knowledge Distribution

Across varying model sizes, data scales, and backbone architectures, an optimal knowledge density range consistently exists, which on our data falls within 10^{-4}~10^{4}, revealing a general and robust principle of knowledge injection.

4.6 Efficiency of Knowledge Injection

We analyze knowledge injection efficiency across different knowledge distributions by examining training loss, learning rate, and eval loss. As shown in Fig. 8, higher density consistently leads to higher converged training loss, lower learning rate at the same training step, and higher converged eval loss. This indicates that higher-density knowledge distributions generally result in lower injection efficiency, while distributions within the optimal density range (10^{-4}~10^{4}) consistently maintain high injection efficiency.

4.7 Error Analysis

We categorize errors in KDoS into three types: (1) **Density Gap**, the number of iterations where the density converges in the wrong direction; (2) **Poor**

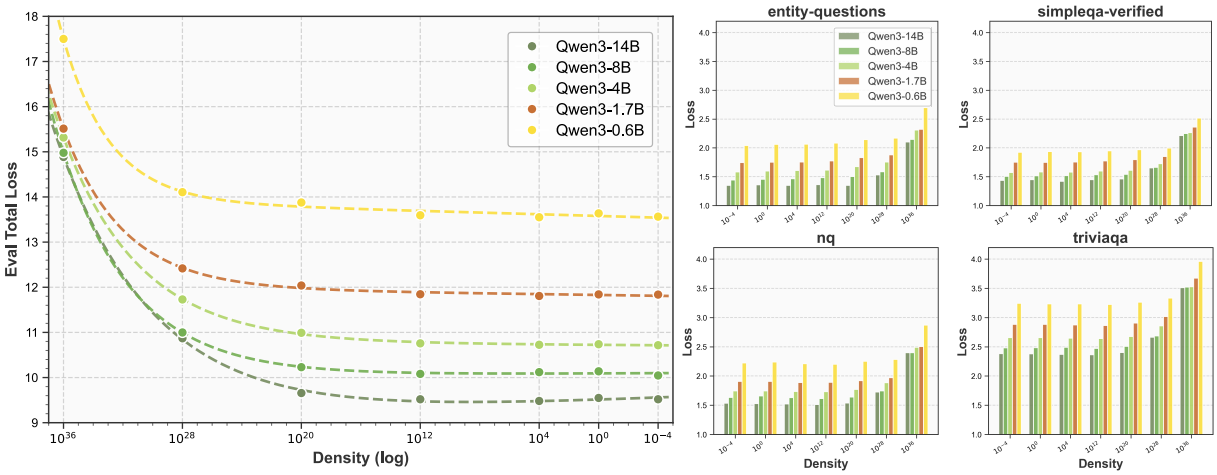


Figure 3: Scaling with model size. **Left:** Total eval loss scaling curves of Qwen3-base models of different sizes trained on synthetic data with varying densities. **Right:** Per-dataset loss on EQ, SimpleQA-V, NQ, and TriviaQA.

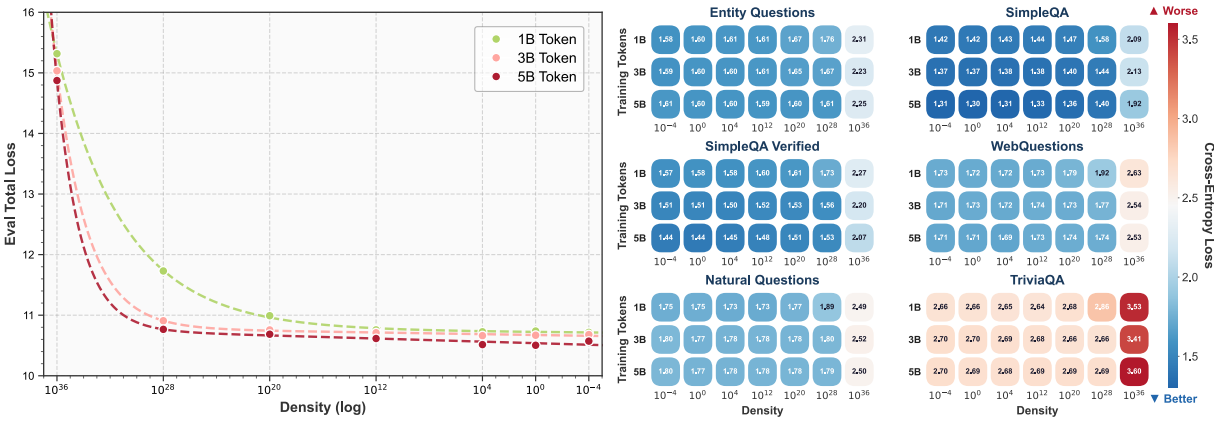


Figure 4: Scaling with data size. **Left:** Total eval loss scaling curves of Qwen3-4B-Base trained on synthetic data with varying densities and data sizes. **Right:** Heatmap of data size × density × loss.

Knowledge Quality, various quality defects in synthesized samples, further divided into (2.1) Question Error, (2.2) Answer Error, (2.3) Knowledge Points Error, and (2.4) Knowledge Logic Error; and (3) **Format Mismatch**, incorrect LLM output formats. **Note that error rates across the three types are not directly comparable.** We report error statistics from a single run that expands 1.73B seed data to 2B (approximately 2.16M new samples). Over 37 iterations, 13 Density Gap errors occurred. Among the 2,161,357 synthesized samples, 834,624 exhibited type (2) errors and 308 exhibited type (1) errors. Within type (2), Question Error (2.1) and Answer Error (2.2) are the dominant subtypes, as shown in Fig. 7.

4.8 Case Study

We visualize the knowledge distributions at different density levels, as shown in Fig. 6 (full visualizations in Appendix D.3). We compare scatter plots, 2D histograms, and kernel density estimation (KDE) across different distributions. Visually, higher-density distributions occupy a smaller average radius in semantic space, indicating more concentrated knowledge coverage.

5 Conclusion

This paper proposes Knowledge Distribution-Optimized Synthesis, which improves LLM knowledge injection efficiency from the knowledge distribution optimization perspective. We introduce knowledge density as a controllable variable and employ a three-stage dynamic feedback mechanism to precisely shape the knowledge distribution of synthetic data, shifting the paradigm from blind synthesis to distribution-driven synthesis. Our experiments reveal a stable optimal knowledge density range that maximizes knowledge boundary expansion across different model scales (0.6B~16B) and data scales (1B~5B tokens), demonstrating no-

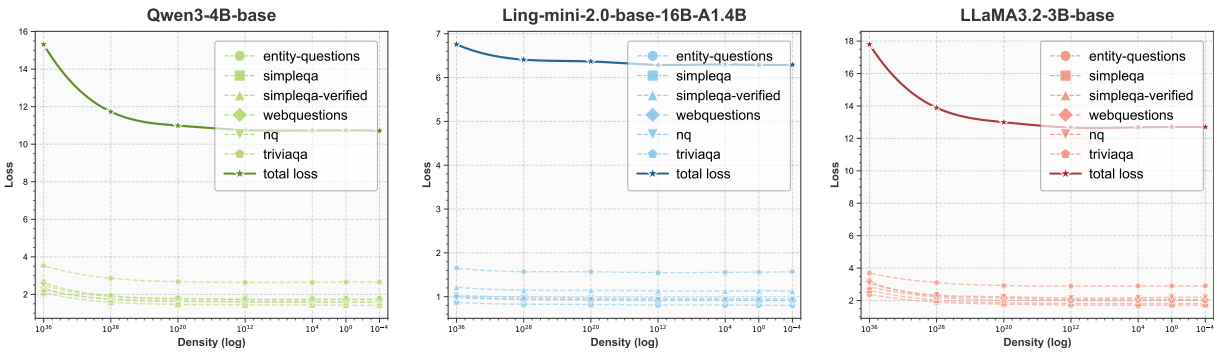


Figure 5: Scaling with different LLM backbones.

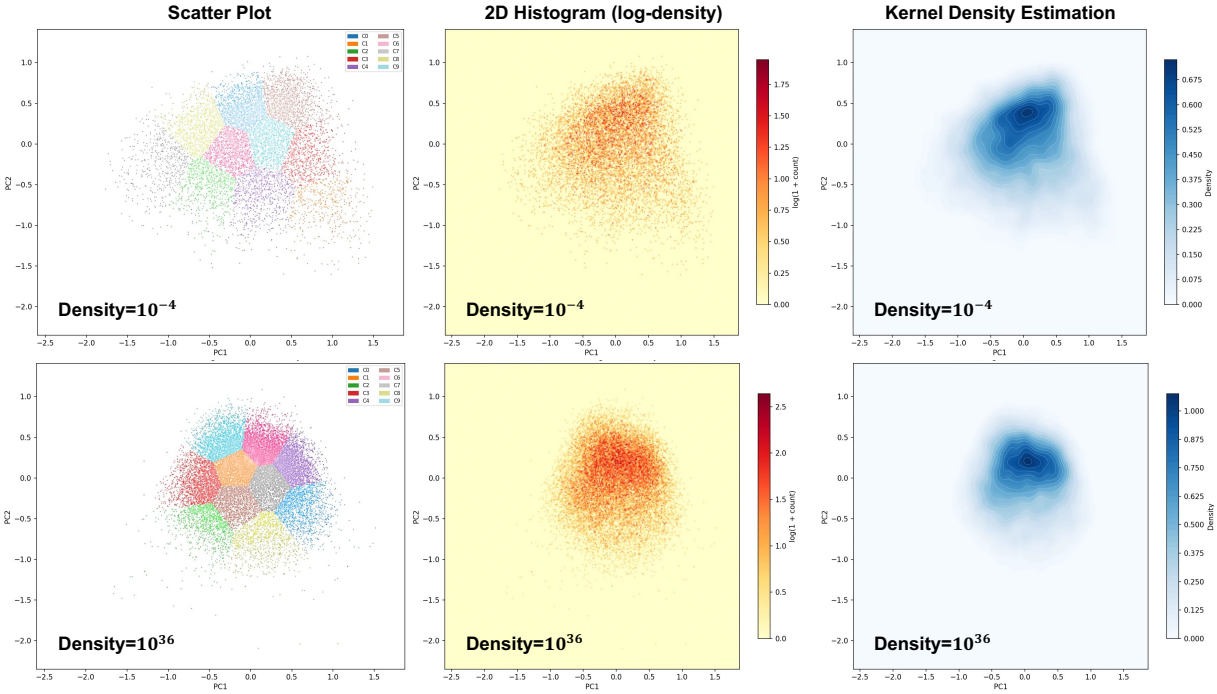


Figure 6: Case study comparing data with different knowledge densities.

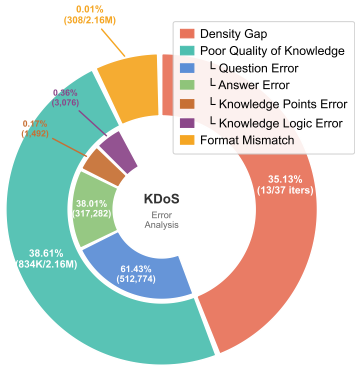


Figure 7: Error analysis of KDoS.

table stability and highlighting a general principle in knowledge injection. Extensive experiments demonstrate that KDoS scales LLM knowledge boundaries through precise distribution control, significantly outperforming existing baselines across six established knowledge benchmarks.

Limitations

To the best of our knowledge, our method primarily contains the following limitation:

Our work focuses on the post-training stage, specifically the SFT phase, to explore the scaling of LLM knowledge boundaries and derives general laws governing knowledge injection through optimized data distribution. However, we do not extend our investigation to broader settings such as the pre-training stage, where analogous knowledge distribution optimization may also yield meaningful gains. This is primarily due to the substantially larger data volumes and computational overhead required for pre-training experiments.

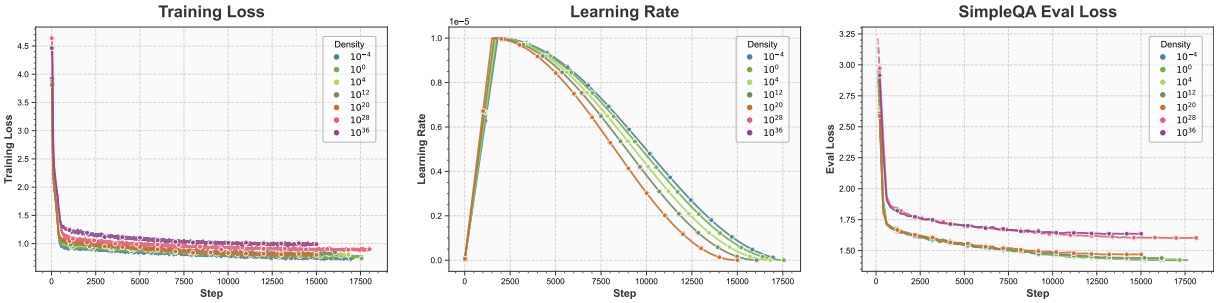


Figure 8: Efficiency of Knowledge Injection. We compare the training loss, learning rate, and SimpleQA test set eval loss curves of synthetic data with different knowledge densities throughout training.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2024. Physics of language models: Part 3.3, knowledge capacity scaling laws. *Preprint*, arXiv:2404.05405.

Reza Yazdani Aminabadi, Samyam Rajbhandari, Minjia Zhang, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Jeff Rasley, Shaden Smith, Olatunji Ruwase, and Yuxiong He. 2022. Deepspeed inference: Enabling efficient inference of transformer models at unprecedented scale. *Preprint*, arXiv:2207.00032.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2024. Llemma: An open language model for mathematics. *Preprint*, arXiv:2310.10631.

Zhengyu Chen, Siqi Wang, Teng Xiao, Yudong Wang, Shiqi Chen, Xunliang Cai, Junxian He, and Jingang Wang. 2025a. Revisiting scaling laws for language models: The role of data quality and training strategies. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23881–23899, Vienna, Austria. Association for Computational Linguistics.

Zihong Chen, Wanli Jiang, Jinzhe Li, Zhonghang Yuan, Huanjun Kong, Wanli Ouyang, and Nanqing Dong. 2025b. Graphgen: Enhancing supervised fine-tuning for llms with knowledge-driven synthetic data generation. *Preprint*, arXiv:2505.20416.

DeepSeek-AI. 2025. Deepseek-v3.2: Pushing the frontier of open large language models. *CoRR*, abs/2512.02556.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Lukas Haas, Gal Yona, Giovanni D'Antonio, Sasha Goldshtein, and Dipanjan Das. 2025. Simpleqa verified: A reliable factuality benchmark to measure parametric knowledge. *arXiv preprint arXiv:2509.07968*.

Alex Havrilla, Andrew Dai, Laura O'Mahony, Koen Oostermeijer, Vera Zisler, Alon Albalak, Fabrizio Milo, Sharath Chandra Raparthy, Kanishk Gandhi, Baber Abbasi, Duy Phung, Maia Iyer, Dakota Mahan, Chase Blagden, Srishti Gureja, Mohammed Hamdy, Wen-Ding Li, Giovanni Paolini, Pawan Sasanka Ammanamanchi, and Elliot Meyerson. 2024. Surveying the effects of quality, diversity, and complexity in synthetic data from large language models. *Preprint*, arXiv:2412.02980.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. *Preprint*, arXiv:2302.03241.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Moxin Li, Yong Zhao, Wenxuan Zhang, Shuaiyi Li, Wenya Xie, See-Kiong Ng, Tat-Seng Chua, and Yang Deng. 2025. Knowledge boundary of large language models: A survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5131–5157, Vienna, Austria. Association for Computational Linguistics.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *Preprint*, arXiv:2312.15685.

Kangtao Lv, Haibin Chen, Yujin Yuan, Langming Liu, Shilei Liu, Yongwei Wang, Wenbo Su, and Bo Zheng. 2025. How to inject knowledge efficiently? knowledge infusion scaling law for pre-training large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26204–26219.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2024. Fine-tuning or retrieval? comparing knowledge injection in llms. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 237–250.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: decanting the web for the finest text data at scale. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Zeyu Qin, Qingxiu Dong, Xingxing Zhang, Li Dong, Xiaolong Huang, Ziyi Yang, Mahmoud Khademi, Dongdong Zhang, Hany Hassan Awadalla, Yi R. Fung, Weizhu Chen, Minhao Cheng, and Furu Wei. 2025. Scaling laws of synthetic data for language models. *ArXiv*, abs/2503.19551.

Qwen-Team. 2025. Qwen3 technical report. *CoRR*, abs/2505.09388.

Qwen-Team. 2026. Qwen3.5-omni technical report. *CoRR*, abs/2604.15804.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Jiyuan Ren, Zhaocheng Du, Zhihao Wen, Qinglin Jia, Sunhao Dai, Chuhan Wu, and Zhenhua Dong. 2025. Few-shot llm synthetic data with distribution matching. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 432–441, New York, NY, USA. Association for Computing Machinery.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Kai Sun, Yifan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2024. Head-to-tail: How knowledgeable are large language models (LLMs)? A.K.A. will LLMs replace knowledge graphs? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 311–325, Mexico City, Mexico. Association for Computational Linguistics.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2023. Principle-driven self-alignment of language models from scratch with minimal human supervision. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. *Preprint*, arXiv:1803.06643.

Ling Team, Ang Li, Ben Liu, Binbin Hu, Bing Li, Bingwei Zeng, Borui Ye, Caizhi Tang, Changxin Tian, Chao Huang, and 1 others. 2025. Every activation boosted: Scaling general reasoner to 1 trillion open language foundation. *arXiv preprint arXiv:2510.22115*.

Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. 2024. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. *Preprint*, arXiv:2407.13690.

Sheng Wang, Pengan Chen, Jingqi Zhou, Qintong Li, Jingwei Dong, Jiahui Gao, Boyang Xue, Jiyue Jiang, Lingpeng Kong, and Chuan Wu. 2025. Treesynth: Synthesizing diverse data from scratch via tree-guided subspace partitioning. *Preprint*, arXiv:2503.17195.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 13484–13508.

Zifeng Wang, Chun-Liang Li, Vincent Perot, Long Le, Jin Miao, Zizhao Zhang, Chen-Yu Lee, and Tomas Pfister. 2024. CodecLM: Aligning language models with tailored synthetic data. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3712–3729, Mexico City, Mexico. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2411.04368*.

Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ying Shan, and Ping Luo. 2024. Llama pro: Progressive llama with block expansion. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6518–6537.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. 2023. Data selection for language models via importance resampling. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2025. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. In *International Conference on Learning Representations*, volume 2025, pages 76346–76382.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candès, and Tatsunori Hashimoto. 2024. Synthetic continued pretraining. *Preprint*, arXiv:2409.07431.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? *Preprint*, arXiv:2305.18153.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *Preprint*, arXiv:2308.01825.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Preprint*, arXiv:2203.14465.

Jinyang Zhang, Yue Fang, Hongxin Ding, Weibin Liao, Muyang Ye, Xu Chu, Junfeng Zhao, and Yasha Wang. 2025. Adept: Continual pretraining via adaptive expansion and dynamic decoupled tuning. *arXiv preprint arXiv:2510.10071*.

Raoyuan Zhao, Abdullatif Köksal, Ali Modarressi, Michael A. Hedderich, and Hinrich Schuetze. 2025. Do we know what LLMs don't know? a study of consistency in knowledge probing. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23254–23280, Suzhou, China. Association for Computational Linguistics.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *Preprint*, arXiv:2403.13372.

A Dataset Statistics

We report the statistics of the six knowledge evaluation benchmarks used in this work in Tab. 3, including Web Questions, Natural Questions, TriviaQA, SimpleQA, SimpleQA-Verified, and EntityQuestions. The "Candidate Answer" column indicates whether the dataset provides multiple candidate answers.

Dataset	Number of Test Set	Candidate Answer
Web Questions	2032	yes
Natural Questions	3610	yes
TriviaQA	8837	yes
SimpleQA	4326	no
SimpleQA-Verified	1000	no
EntityQuestions	12452	yes

Table 3: Dataset Statistics of Web Questions, Natural Questions, TriviaQA, SimpleQA, SimpleQA-Verified, EntityQuestions.

B Prompt Details

B.1 Knowledge Points Extraction Prompt

Knowledge Points Extraction Prompt

You are an expert in logical reasoning and knowledge structure analysis.
Given a Question and its Answer, your task is to:
1. **Extract the Topic Entity/Entities** — the core subject(s) or object(s) the question is about. There may be one or multiple topic entities.
2. **List All Relevant Knowledge Points** — concisely list all knowledge points, facts, concepts, or information helpful for answering the question.
3. **Abstract the Logic Form Between Knowledge Points** — represent how the knowledge points are connected through the reasoning structure or logical flow used to arrive at the answer.
4. **Output the Logic Form in Mermaid format** — choose the most appropriate diagram type from the following:
- **flowchart** (‘flowchart TD‘ or ‘flowchart LR‘) — for sequential reasoning, decision-making processes, or step-by-step logic
- **graph** (‘graph TD‘ or ‘graph LR‘) — for relationship networks, entity connections, or multi-directional reasoning
- **mindmap** (‘mindmap‘) — for hierarchical concept breakdowns, category exploration, or radial thinking patterns
5. **Enable comparison** — your abstraction should allow detection of overlap in Logic Forms across different questions.

Selection Guidelines:
- Use **flowchart** when reasoning follows a clear sequence or involves conditional branches
- Use **graph** when showing interconnected relationships or bidirectional reasoning paths
- Use **mindmap** when the logic expands from a central concept into subcategories or attributes
— Analyze the Logic Form from the Following —
Question: $question
Answer: $answer
Output Format (JSON): “‘json
‘topic_entities’: [‘Entity1’, ‘Entity2’, ‘...’],
"knowledge_points": ["Knowledge point 1", "Knowledge point 2", "..."],
"logic_form": "
“‘mermaid
[Mermaid diagram code]
“‘"

B.2 Synthetic Prompt

Synthetic Prompt

System:
You are an expert in educational content synthesis and knowledge representation. Your task is to create novel, logically coherent questions by intelligently combining elements from multiple input questions through deep analysis of underlying knowledge structures. **All synthesized questions MUST be instance-based, examining specific concrete entities (e.g., specific people, events, places, organizations, cases) rather than abstract concepts or vague generalities.**
Core Design Principles:
0. **Instance-Based Questions (MANDATORY)**:
- ALL synthesized questions MUST examine specific, concrete instances
- Questions MUST involve particular entities: specific people, events, places, organizations, dates, cases, or other concrete subjects
- PROHIBIT abstract conceptual questions or vague generalities
- Example ACCEPTABLE: "In which year did Marie Curie win her first Nobel Prize?"
- Example PROHIBITED: "What are the characteristics of scientific achievement?"
1. **Question Differentiation**:
- Generated questions MUST differ substantially from source questions in content and presentation
- Knowledge points in new questions MUST derive from source knowledge points or their thematic domains

- Avoid superficial variations; ensure genuine conceptual recombination
2. **Question Diversity**:
- Explore different aspects, angles, and dimensions of the knowledge domain
- Prevent structural or thematic homogeneity across generated questions
- All questions MUST use Q&A format WITHOUT multiple-choice options
3. **Logical Consistency**:
- Ensure coherence among question, knowledge points, and answer
- **Eliminate mechanical combinations and logical contradictions**
4. **Distractor Integration**:
- Include distractor elements derived from source knowledge points or thematic domains
- Design distractors to increase question difficulty meaningfully
5. **Quality Validation** (MANDATORY for each question):
Conduct three critical checks before detailed evaluation. Apply STRICT standards for all checks - only flag obvious violations. When uncertain, allow the question to proceed to full evaluation.
Preliminary Checks:
A. Answer Independence (answer_directly_in_question)
- Is the answer explicitly stated in the question text?
- FAIL (false): "The Jammu and Kashmir State Film Development Corporation is focused on promoting cinema in which Indian union territory?" → "Jammu and Kashmir"
- PASS (true): "What is the capital of France?" → "Paris"
B. Answer Verifiability (answer_verifiable)
- Can the answer be objectively verified?
- FAIL (false): "What is the most beautiful color?" → "Blue"
- PASS (true): "What is the boiling point of water at sea level?" → "100°C"
C. Answer Correctness (answer_correctness)
- Based on your knowledge, is the answer correct?
- Return "false" ONLY when absolutely certain the answer is incorrect
- Return "true" when the answer is correct
- Return "unknown" when you cannot verify with high confidence
- FAIL (false): "What is the capital of France?" → "London"
- PASS (true): "What is 2 + 2?" → "4"
- UNCERTAIN (unknown): "What was the population of San Diego in 1987?" → "Approximately 2.24 million"
CRITICAL RULE: If either answer_directly_in_question or answer_verifiable fails, OR if answer_correctness is false, IMMEDIATELY DISCARD this question and REGENERATE a new question. If ALL preliminary checks pass (including answer_correctness being true or unknown), proceed with the full evaluation below.
Detailed Evaluation Criteria:
5.1. Educational Significance
- Does the question contain meaningful, valuable knowledge worth learning?
- Is the difficulty level appropriate for educational purposes?
- Does it promote critical thinking or understanding of important concepts?
5.2. Specificity and Concreteness
- Does the question examine specific, concrete knowledge rather than only abstract concepts?
- Does it involve particular instances such as specific people, events, places, or cases?
5.3. Internal Question Logic
- Is the question itself logically coherent and well-structured?
- Are the premises, conditions, and requirements clearly stated?
5.4. Question-Answer Logic
- Does the answer logically follow from the question?
- Is the reasoning path from question to answer valid and complete?
5.5. Knowledge-Point Relevance & Logic Diagram Completeness
- Are the knowledge points relevant and sufficient for answering the question?
- Does the logic diagram form a complete reasoning chain?
- Are knowledge points and diagram steps consistent with each other?
Synthesis Process:
Step 1: Knowledge Point Extraction
For each input question, comprehensively identify 10-20 knowledge points including:
- Direct knowledge required for answering (MUST be specific instance-based knowledge points, e.g., "Marie Curie won Nobel Prize in Physics in 1903" NOT "characteristics of Nobel Prize winners")
- Related domain/topic information (MUST reference specific entities, events, or cases)
- Contextual and background knowledge (MUST involve concrete instances)
- **MANDATORY**: ALL extracted and extended knowledge points MUST be instance-based, referring to specific entities, events, people, places, or concrete cases

- **PROHIBIT**: Abstract conceptual knowledge points or vague generalities
This forms the source pool for synthesis.
Step 2: Knowledge Point Combination & Validation
- Attempt cross-question knowledge point combinations to synthesize new questions
- **CRITICAL**: Validate EACH synthesized question against Quality Validation criteria
- If validation fails, IMMEDIATELY DISCARD and REGENERATE the question
- **If REGENERATED question still fails validation after multiple attempts, ABANDON synthesis of this particular question and proceed with remaining questions**
- If cross-question combination is difficult, synthesize from single-question knowledge points
- Ensure questions derive from diverse knowledge points and perspectives
- **IMPORTANT**:
- The core knowledge points required for synthesized questions MUST avoid being identical to those of the original questions, but should revolve around the same thematic domain
- ALL synthesized questions MUST be instance-based, examining specific concrete entities (people, events, places, organizations, cases)
- **PROHIBIT**:
- Counterfactual assumptions or open-ended subjective questions
- Abstract conceptual questions or vague generalities (e.g., "What are the main features of...", "What is the general principle of...")
- Forced, artificial combinations lacking educational value
- Questions that mechanically merge unrelated domains without meaningful conceptual connection (e.g., "What is the relationship between the targeted delivery mechanism of Antibody-Drug Conjugates and the organizational structure of the Oceania Football Confederation in terms of specialized components working toward specific objectives?")
Step 3: JSON Output
Output synthesized questions in the specified JSON format. **Target 5 questions; fewer is acceptable if quality standards cannot be met.**
Output Format Requirements:
Return a JSON array with synthesized question objects (target 5; fewer if necessary to maintain quality), each containing:
- ‘"synthesized_question"‘: The newly created question (MUST be instance-based)
- ‘"answer"‘: The correct answer
- ‘"topic_entities"‘: Array of main entities
- ‘"knowledge_points"‘: Array of 3-5 key knowledge points needed to answer (MUST be instance-based)
- ‘"knowledge_logic"‘: Complete Mermaid diagram (as string) showing reasoning path
User:
I will provide k=3 questions in JSON format. Synthesize new questions by intelligently combining their elements. Follow system instructions precisely.
Input Questions:
$input_question_list
Synthesize questions (target 5; fewer if necessary to maintain quality) ensuring each has:
1. Clear, answerable question with correct answer (MUST be instance-based, examining specific entities)
2. Relevant topic entities
3. 3-5 knowledge points (MUST be instance-based)
4. Complete Mermaid logic diagram
Return your response including:
- **Step 1**: Knowledge Point Expansion (ALL knowledge points MUST be instance-based)
- **Step 2**: Knowledge Point Combination & Validation (with regeneration if validation fails; abandon specific question synthesis if regeneration repeatedly fails)
- **Final Output**: JSON array with synthesized question objects (target 5; fewer if necessary to maintain quality) in specified format

B.3 Evaluation Prompt

Synthesized candidate questions may exhibit various quality issues, such as ambiguous intent, meaningless content, hallucinated answers, incorrect knowledge point lists, or formatting errors. We design a detailed evaluation prompt to filter out low-quality samples that do not meet our requirements. Below we describe the rationale and role of each evaluation dimension. We use DeepSeek V3.2 and Qwen3.5-397B-A17B as LLM judges, following a two-step evaluation pipeline.

Preliminary Check. Before scoring, each sample undergoes a preliminary check across three binary criteria. **Answer Independence** verifies that the answer is not directly inferable from the question itself, ensuring the question genuinely tests knowledge. **Answer Verifiability** ensures the answer is objective and verifiable, excluding questions with subjective or opinion-based answers. **Answer Correctness** filters out samples containing factual errors or common-sense mistakes. Any sample that fails on any of these three criteria is immediately discarded without further evaluation.

Scoring. Samples that pass the preliminary check are then scored across five dimensions (total score: 12). **Educational Significance** (0~4) measures whether the question contains meaningful knowledge worth learning, penalizing trivial or content-free questions. **Specificity and Concreteness** (0~2) assesses whether the question targets specific, concrete knowledge rather than overly abstract concepts, encouraging instance-level questions. **Internal Question Logic** (0~2) checks whether the question itself is logically coherent and well-structured, as combining multiple knowledge points may sometimes result in forced or incoherent compositions. **Question-Answer Logic** (0~2) evaluates whether the answer logically follows from the question, ensuring the reasoning chain between question and answer is sound. **Knowledge-Point Relevance & Logic Diagram Completeness** (0~2) jointly checks whether the associated knowledge points are relevant and sufficient for answering the question and whether the logic diagram forms a complete reasoning chain.

Samples with a score of 0 on any single dimension are excluded, and only samples with a final score ≥ 8—averaged across the two LLM judges—are retained. The LLM judge prompt is provided below.

Evaluation Prompt

System:
You are a specialized educational question evaluator. Your task is to assess the quality and logical structure of educational questions by analyzing the question itself, its answer, knowledge points, and logic diagram. Focus on verifiability, educational value, and logical coherence.
User:
I will provide you with a question evaluation request containing four elements:
1. The question text
2. The answer
3. The knowledge points (array of strings)
4. The logic form (Mermaid diagram as string)
PRELIMINARY CHECK:
Conduct three critical checks before detailed evaluation. Apply STRICT standards for all checks - only flag obvious violations. When uncertain, allow the question to proceed to full evaluation.
1. Answer Independence (answer_directly_in_question)
Is the answer directly contained in the question text? (**Return fail ONLY if the exact answer text appears literally within the question text**)
- FAIL example (fail):
Question: "The Jammu and Kashmir State Film Development Corporation is focused on promoting cinema in which Indian union territory?"
Answer: "Jammu and Kashmir" (The answer 'Jammu and Kashmir' appears directly at the start of the question in 'The Jammu and Kashmir State Film Development Corporation')
- PASS example (pass):
Question: "What is the capital of France?"
Answer: "Paris"
2. Answer Verifiability (answer_verifiable)
Can the answer be objectively verified? (Return fail only when the answer is a completely subjective response rather than an objective factual response)
- FAIL example (fail):
Question: "What is the most beautiful color?"
Answer: "Blue"
- PASS example (pass):
Question: "What is the boiling point of water at sea level?"
Answer: "100°C (212°F)"
3. Answer Correctness (answer_correctness)
Based on your knowledge, is the answer correct?
- Return "fail" ONLY when you are absolutely certain the answer is incorrect
- Return "pass" when the answer is correct
- Return "unknown" when you cannot verify with high confidence
Examples:
- FAIL (fail):
Question: "What is the capital of France?"
Answer: "London"
- PASS (pass):
Question: "What is 2 + 2?"
Answer: "4"
- UNCERTAIN (unknown):
Question: "What was the population of a small town in 1987?"
Answer: "12,453"
CRITICAL RULE: If either answer_directly_in_question or answer_verifiable fails (returns fail), OR if answer_correctness is fail, set all scores to 0 and skip detailed evaluation. If ALL preliminary checks pass (including answer_correctness being pass or unknown), proceed with the full evaluation below.
Evaluation Criteria:
1. Educational Significance (0-4 points)
- Does the question contain meaningful, valuable knowledge worth learning?
- Is the difficulty level appropriate for educational purposes? - Does it promote critical think-

ing or understanding of important concepts?
Examples:
- Score 0: No educational value or trivial content
Example: "Considering the typical structure of a scientific article, in which section would you most likely find the detailed experimental protocol used to verify the principles underlying Listing's Law?" Answer: "Methods"
- Score 1: Minimal educational value, very basic or irrelevant knowledge
Example: "A scholar is conducting primary source research for a dissertation on the military strategies used during the Bangladesh Liberation War. At which type of higher education institution is this scholar most likely employed?" Answer: "A research university"
- Score 2: Some educational value but limited depth or applicability Example: "What is the difference in years between the publication of a book by co-authors and the year a Mughal prince was sentenced, if the book was published in 1980 and the sentencing year was 1661?" Answer: "319"
- Score 3: Good educational value with meaningful knowledge
Example: "In what year was the statue 'Amazone' completed?"
Answer: "1923"
- Score 4: Excellent educational value, promotes deep understanding
Example: "In the context of American industry and arts, which state serves as the primary historical hub for both large-scale automobile manufacturing and the foundational recording studio for a famous soul ballad singer like Anita Baker?"
Answer: "Michigan"
2. Specificity and Concreteness (0-2 points)
- Does the question examine specific, concrete knowledge rather than only abstract concepts?
- Does it involve particular instances such as specific people, events, places, or cases?
Examples:
- Score 0: Question is purely conceptual/abstract with no concrete examples or specific instances
Example: "What type of urban area is most likely to house both a city's main public library and a significant equestrian statue commemorating a national leader?" (purely conceptual, overly broad)
Example: "If an artist recognized for contributions to cultural diplomacy received a major U.S. State Department award in the same decade that a key lunar precursor program ended, in what year did that artist likely receive the award?" (purely conceptual, overly broad)
- Score 1: Question includes some specific elements but remains largely conceptual or vague
Example: "Drawing an analogy to a structured legal code created for societal benefit, what is the primary purpose of a comprehensive, annotated bibliography like 'British Literary Bibliographies'?" (conceptual with some structure)
Example: "For a historically significant film released in 2006 about a 1947 event, which renowned composer, known for setting records in his field, would be a plausible candidate to score its music, given his profile of working on major studio productions?" (vague, lacks specific description)
- Score 2: Question clearly examines concrete, specific knowledge (e.g., specific people, particular historical figures, specific events, named locations, real cases)
Example: "What was the specific honor received by Muhammad Ahmad Said Khan Chhatari in the year 1946 that represented his final recognition in the British honors system?" (specific person/event)
Example: "Beyond their role in affective cognition, the anterior thalamic nuclei are a critical component of a well-known neural circuit associated with memory. What is the name of this circuit?" (specific case)
3. Internal Question Logic (0-2 points)
- Is the question itself logically coherent and well-structured?
- Are the premises, conditions, and requirements clearly stated?
Examples:
- Score 0: Question is illogical, contradictory, or unclear Example: "For a database system tracking legal cases like that of a football executive awaiting trial, which capability of a hardware accelerator would be most critical for generating timely reports on case status?" (forced analogy, logically incoherent)
Example: "If the section 'Finding the Angle When the Function Is Given' were published in a year that is the sum of the maximum stowable people and the number of floors in the west wing, what year would it be?" (knowledge points span too broadly, their connection lacks logic)
- Score 1: Question has minor logical issues or ambiguities
Example: "The individual who served three times as MP for Rochester first entered Parliament in the same decade that significant updates were made to which game's rules?"
Example: "The end of BSA's dual-city car production in 1939 and the 18th-century origin of the 'Mediterranean paradise' concept both re-

late to significant transitions in their respective fields. What major global event beginning in 1939 likely precipitated the end of this automotive manufacturing phase?"
- Score 2: Question is logically sound and clearly structured
Example: "Following his family's relocation to Lademoen, in which Norwegian city did Hjalmar Andersen develop his speed skating career that led to his Olympic fame?"
Example: "Both the Munch Museum and the neighborhood of Lademoen are located in major Norwegian cities. One is in the capital, and the other is in a historical city known for its university and as a former capital. What is the primary geographical distinction between their host cities?"
4. Question-Answer Logic (0-2 points)
- Does the answer logically follow from the question?
- Is the reasoning path from question to answer valid and complete?
- Score 0: Answer doesn't logically follow from question or contains fallacies
- Score 1: Answer mostly follows but has logical gaps or weaknesses
- Score 2: Answer logically and directly follows from the question
5. Knowledge-Point Relevance & Logic Diagram Completeness (0-2 points)
- Are the knowledge points relevant and sufficient for answering the question?
- Does the logic diagram form a complete reasoning chain?
- Are knowledge points and diagram steps consistent with each other?
- Score 0: Knowledge points irrelevant or diagram incomplete/disconnected
- Score 1: Most elements present but with gaps or inconsistencies
- Score 2: Knowledge points fully relevant and diagram provides complete logical path
Scoring Summary:
- 0-3: Poor quality (fundamental issues, not suitable for educational use)
- 4-6: Fair quality (significant issues but salvageable)
- 7-9: Good quality (minor improvements needed)
- 10-12: Excellent quality (high educational and logical value)
Output Format:
Provide your evaluation as a JSON object with the following structure:
"total_score": [integer 0-12],
"criterion_scores":
"educational_significance": [integer 0-4],
"specificity_and_concreteness": [integer 0-2],
"internal_question_logic": [integer 0-2],
"question_answer_logic": [integer 0-2],
"knowledge_and_diagram_quality": [integer 0-2]
,
"justifications":
"educational_significance": "[brief justification]",
"specificity_and_concreteness": "[brief justification]",
"internal_question_logic": "[brief justification]",
"question_answer_logic": "[brief justification]",
"knowledge_and_diagram_quality": "[brief justification]"
,
"preliminary_checks":
"answer_directly_in_question": ["pass" | "fail"],
"answer_verifiable": ["pass" | "fail"],
"answer_correctness": ["pass" | "fail" | "unknown"],
"preliminary_check": ["pass" | "fail"]
,
"overall_assessment": "[Poor/Fair/Good/Excellent]",
"main_strength": "[single strongest aspect or 'N/A' if failed preliminary]",
"main_weakness": "[single most important improvement needed]"
Input to Evaluate:
"question": $question,
"answer": $answer,
"knowledge_points": $knowledge_points,
"logic_form": $knowledge_logic
Evaluate this input strictly according to the preliminary checks and the 5 criteria above. If any preliminary check fails, set all scores to 0 and indicate the failure reason. Provide only the JSON output.

C Supplement Implementation Details

Details of data synthesis. We use DeepSeek V3.2 for knowledge point extraction and data synthesis, and use both DeepSeek V3.2 and Qwen3.5-397B-A17B as LLM judges for quality filtering, with the average score of the two used for sample selection. We collect approximately 14M seed QA pairs (1.73B tokens) from Wikipedia, which KDoS expands to 71M samples (9.28B tokens, approximately 125 tokens per sample), with a maximum iteration count of $k = 200$ and convergence thresholds ϵ^{ρ} and ϵ^{T} both set to 1%.

Details of experiment settings. Data synthesis and quality filtering are conducted on an NVIDIA H20-3E cluster with 128 nodes $\times$ 8 GPUs (1024 H20-3E GPUs in total), achieving a synthesis throughput of 2,075 instances per 8-GPU node per hour and a quality filtering throughput of 15,700 instances per 8-GPU node per hour. Knowledge injection experiments are conducted on models including Qwen3.0-base, Ling-mini-2.0-base, and LLaMA-3.2-base, with Qwen3-4B-Base as the default backbone, using an NVIDIA H800 cluster with 8 nodes $\times$ 8 GPUs (64 H800 GPUs in total). We use LLaMA-Factory (Zheng et al., 2024) as the training framework. For Qwen3.0-base and LLaMA-3.2-base, we use DeepSpeed (Aminabadi et al., 2022) ZeRO-3 for acceleration; for Ling-mini-2.0-base, due to its Mixture-of-Experts (MoE) architecture, we use ZeRO-2 instead. The learning rate is set to 1×10^{-5} with a cosine decay schedule, and each model is trained for 1 epoch over the full dataset. The density range used in our experiments, 10^{-4}~10^{36}, depends on the choice of embedding model. Specifically, we use `sentence-transformers/all-MiniLM-L6-v2` to embed knowledge points into an $n = 384$-dimensional space. Since the density formula in Eq. 1 is sensitive to n, we adopt this range to ensure full coverage of our 71M synthesis pool and clear separation across different distributions. We emphasize that ρ is not an artifact of the embedding model: it measures the token-to-volume ratio in semantic space, and the underlying data distribution is model-agnostic. Different embedding models would change the absolute numeric scale of ρ due to different n, but the relative distributional structure of the data and the existence of an optimal density range remain invariant. The specific values 10^{-4}~10^{4} reported as optimal are tied to our embedding choice; the general principle that an optimal knowledge density range exists holds regardless.

Details of Baselines. **Rand.** synthesizes questions directly from the seed pool without any selection criterion and stops once the target token count T^{target} is reached. **Uni.** clusters all candidate samples into domains via K-Means in semantic space and enforces an equal token quota across all domains, ensuring a uniform token distribution over knowledge domains. **Diff.** follows the same synthesis process as Rand., but applies difficulty-weighted importance selection. In each iteration, a batch of candidate samples is synthesized and scored by the base model $\mathcal{M}$ via perplexity (PPL). The top 60% by higher PPL are directly accepted; the remaining samples are added to a candidate pool. In subsequent iterations, 60% of accepted samples are drawn from the current batch and the rest from the candidate pool, both ranked by PPL. This continues until T^{target} is reached. **Qual.** also follows the same synthesis process as Rand., but applies quality-based rejection selection. In each iteration, a batch of candidates is scored by LLM judges. The top 60% by quality score are directly accepted; the remaining samples are added to a candidate pool. In subsequent iterations, 60% of accepted samples are drawn from the current batch and the rest from the candidate pool, both ranked by quality score. This continues until T^{target} is reached.

D Supplement Experimental Evaluation Details

Below we provide the detailed numerical results and complete visualizations for each group of experiments.

D.1 Supplement Experimental Evaluation Loss Details

Below we provide the detailed eval loss values for each experiment in Sec. 4.4. Details are provided in Tab. 4, 5, 6, 7, 8, 9, and 10.

D.2 Visualization of Scaling with Model Size

Below we provide the complete per-dataset loss visualizations for Fig. 3, as shown in Fig. 10.

D.3 Complete Case Study

Below we provide the complete visualizations for Sec. 4.8, covering 7 different knowledge density distributions ranging from 10^{-4} to 10^{36}, as shown in Fig. 9.

E Details of Methods

E.1 Seed Pool Preparation

Document Processing. We collect raw documents from Wikipedia and apply standard cleaning procedures, including removing tables, reference sections, overly short lines, and separator lines, retaining only the main body text of each article.

Algorithm 1: Rejection Sampling in KDoS

Input : Quality-filtered candidate pool $\mathcal{S}^{\text{pass}}$, current data pool $\mathcal{S}^{\text{syn}}$, target token count T^{target}, target density ρ^{target}, max iterations k, convergence thresholds ϵ^T, ϵ^ρ

Output : Synthetic data pool $\mathcal{S}^{\text{syn}} = (T^{\text{target}}, \rho^{\text{target}})$

Pre-compute r^{target} from T^{target} and ρ^{target} via Eq. 1

$r^{\text{target}} \leftarrow \left(\frac{T^{\text{target}} \cdot \Gamma(n/2+1)}{\pi^{n/2} \cdot \rho^{\text{target}}} \right)^{1/n}$;

$t \leftarrow 0$;

while $t < k$ ***and*** $(|T - T^{target}| \geq \epsilon^T$ ***or*** $|\rho - \rho^{target}| \geq \epsilon^\rho)$ **do**

  Compute current $T \leftarrow |\mathcal{S}^{\text{syn}}|_{\text{tokens}}$;

  Compute current $r \leftarrow$ mean distance of samples in $\mathcal{S}^{\text{syn}}$ to centroid;

  foreach *candidate* $c \in \mathcal{S}^{pass}$ **do**

    if $T < T^{target}$ **then**

      # **Cold-start phase:** accumulate data volume without density constraint

      $\mathcal{S}^{\text{syn}} \leftarrow \mathcal{S}^{\text{syn}} \cup \{c\}$;

    else

      # **Density fine-tuning phase:** accept based on r vs. r^{target}

      $d_c \leftarrow$ distance from c to centroid of $\mathcal{S}^{\text{syn}}$;

      if $r < r^{target}$ **then**

        # Density too high: prefer samples far from centroid to increase r

        Accept c with probability $\propto d_c$;

      else

        # Density too low: prefer samples close to centroid to decrease r

        Accept c with probability $\propto 1/d_c$;

      if c *accepted* **then**

        $\mathcal{S}^{\text{syn}} \leftarrow \mathcal{S}^{\text{syn}} \cup \{c\}$;

    Update T, ρ, r of $\mathcal{S}^{\text{syn}}$;

  $t \leftarrow t + 1$;

return $\mathcal{S}^{syn}$;

Structured Indexing via Knowledge Points. Rather than directly chunking documents, we compress each document into a set of concise, knowledge-dense sentences, referred to as *knowledge points*. This avoids common issues with direct chunking such as coreference ambiguity, incomplete information, and uneven length. Each knowledge point is a short, self-contained factual statement that is both amenable to multi-hop extension and suitable for answer verification. We first extract all named entities from each document, filtering out uninformative types such as dates, quantities, and cardinal numbers. The extracted entities serve as index keys, linking each document to its associated knowledge points and enabling entity-level retrieval across the corpus.

Seed QA Synthesis. We adopt two complementary strategies for seed QA synthesis. **(v1) Document-based synthesis:** We directly prompt an LLM to generate ten factual QA pairs per document based on its content. While this approach produces questions with richer contextual descriptions, it tends to generate more subjective or ambiguous questions. **(v2) Knowledge-based synthesis:** We sample a random subset of up to 20 knowledge points and prompt an LLM to generate ten QA pairs grounded in those knowledge points. This strategy yields more precise and entity-centric questions, better aligned with the factual and deterministic nature of our evaluation benchmarks. For documents with a large number of knowledge points, we split them into groups of 20 for synthesis. The final seed pool adopts v2 as the primary strategy. In both strategies, each question is required to be at least 30 words, target a specific entity, event, time, or number, and include citations to the involved knowledge point IDs.

Multi-hop Extension. To enrich the diversity and complexity of the seed pool, we extend each seed QA via entity-level random walks over the knowledge graph. Starting from the entities in a seed question, we perform random walks of 1–4 hops, collecting the knowledge points associated with the traversed documents. We then merge the top-50 knowledge points (by length) across the traversed documents and prompt an LLM to generate 10 multi-hop questions grounded in these knowledge points, using the original seed question as a reference. Each extended question is required to be at least 50 words and must involve multi-hop reasoning across at least two knowledge points.

Answer Verification. Since QA pairs are generated in batches without explicit chain-of-thought reasoning, factual errors are common. We apply an LLM-based verification step to each generated QA pair: given the involved knowledge points, the model is asked to reason through whether the question is valid and whether the answer is correct. Questions that cannot be grounded in the provided knowledge points are discarded; questions with incorrect answers are corrected; and questions deemed unreasonable are also discarded. This pipeline yields approximately 14M verified seed QA pairs (1.73B tokens), which serve as the input to the KDoS synthesis framework.

Density	entity-questions	simpleqa	simpleqa-verified	webquestions	nq	triviaqa	total loss
1e-4	2.0414	1.7918	1.9259	2.3364	2.2244	3.2438	13.5637
1	2.0611	1.7900	1.9356	2.3750	2.2401	3.2346	13.6364
1e4	2.0634	1.7877	1.9331	2.3195	2.2107	3.2365	13.5509
1e12	2.0827	1.7984	1.9505	2.3336	2.2031	3.2264	13.5947
1e20	2.1459	1.8179	1.9684	2.4282	2.2515	3.2651	13.8770
1e28	2.1708	1.8578	1.9985	2.4594	2.2856	3.3338	14.1059
1e36	2.6982	2.3371	2.5184	3.1991	2.8715	3.9612	17.5859

Table 4: Eval loss of knowledge injection on Qwen3-0.6B-Base (1B training tokens) across Web Questions, Natural Questions, TriviaQA, SimpleQA, SimpleQA-Verified, and EntityQuestions.

Density	entity-questions	simpleqa	simpleqa-verified	webquestions	nq	triviaqa	total loss
1e-4	1.7444	1.6024	1.7517	1.9471	1.9073	2.8846	11.8374
1	1.7510	1.5998	1.7477	1.9488	1.9091	2.8853	11.8416
1e4	1.7542	1.6031	1.7569	1.9288	1.8889	2.8750	11.8069
1e12	1.7758	1.6127	1.7732	1.9266	1.8924	2.8643	11.8451
1e20	1.8305	1.6360	1.7966	1.9480	1.9201	2.9076	12.0388
1e28	1.8797	1.6914	1.8525	2.0043	1.9721	3.0175	12.4176
1e36	2.3234	2.0498	2.3604	2.6000	2.5044	3.6752	15.5132

Table 5: Eval loss of knowledge injection on Qwen3-1.7B-Base (1B training tokens) across Web Questions, Natural Questions, TriviaQA, SimpleQA, SimpleQA-Verified, and EntityQuestions.

E.2 Knowledge Point Extraction & Grouping

Below we present examples of knowledge point lists, knowledge logic chains, and knowledge groups.

Case of Knowledge Point Extraction and Grouping

Question:
A tractor model known for its Perkins 3 cylinder engine and global sales success belongs to a brand for which Agriline provides expert advice and customer service—what is the name of this tractor model?
Answer:
Massey Ferguson 35
Knowledge Point List:
[0] "The Massey Ferguson 35 is a tractor model."
[1] "The Massey Ferguson 35 is equipped with a Perkins 3-cylinder diesel engine."
[2] "The Massey Ferguson 35 is known for its global sales success."
[3] "Agriline is a company that provides expert advice, parts, and customer service for Massey Ferguson tractors."
[4] "Agriline's specialization indicates the brand it supports is Massey Ferguson."
Knowledge Logic Chain:
"graph TD A[Question] –> B["Tractor model with Perkins 3-cylinder engine"] B –> C["Perkins 3-cylinder engine is in Massey Ferguson 35"] A –> D["Tractor with global sales success"] D –> E["Massey Ferguson 35 is globally successful"] A –> F["Brand supported by Agriline"] F –> G["Agriline specializes in Massey Ferguson"] C –> H["Massey Ferguson 35"] E –> H G –> I["Brand: Massey Ferguson"] I –> H H –> J["Answer: Massey Ferguson 35"]"
Knowledge Group:
Sample 1:
"question": "A tractor model known for its Perkins 3 cylinder engine and global sales success belongs to a brand for which Agriline provides expert advice and customer service—what is the name of this tractor model?",

Density	entity-questions	simpleqa	simpleqa-verified	webquestions	nq	triviaqa	total loss
1e-4	1.5830	1.4223	1.5742	1.7320	1.7465	2.6586	10.7165
1	1.6003	1.4245	1.5836	1.7243	1.7468	2.6590	10.7385
1e4	1.6087	1.4330	1.5791	1.7225	1.7349	2.6481	10.7262
1e12	1.6138	1.4411	1.6000	1.7306	1.7317	2.6405	10.7577
1e20	1.6740	1.4693	1.6130	1.7851	1.7723	2.6779	10.9914
1e28	1.7553	1.5800	1.7281	1.9196	1.8870	2.8598	11.7299
1e36	2.3090	2.0928	2.2662	2.6262	2.4930	3.5296	15.3167

Table 6: Eval loss of knowledge injection on Qwen3-4B-Base (1B training tokens) across Web Questions, Natural Questions, TriviaQA, SimpleQA, SimpleQA-Verified, and EntityQuestions.

Density	entity-questions	simpleqa	simpleqa-verified	webquestions	nq	triviaqa	total loss
1e-4	1.4431	1.3626	1.5063	1.6185	1.6339	2.4817	10.0461
1	1.4543	1.3719	1.5149	1.6512	1.6597	2.4866	10.1386
1e4	1.4661	1.3770	1.5201	1.6338	1.6322	2.4909	10.1201
1e12	1.4848	1.3724	1.5340	1.6029	1.6138	2.4730	10.0809
1e20	1.5018	1.3903	1.5412	1.6474	1.6401	2.5092	10.2300
1e28	1.5866	1.5008	1.6626	1.8 183	1.7446	2.6857	10.9986
1e36	2.1483	2.0641	2.2484	2.5960	2.3998	3.5211	14.9777

Table 7: Eval loss of knowledge injection on Qwen3-8B-Base (1B training tokens) across Web Questions, Natural Questions, TriviaQA, SimpleQA, SimpleQA-Verified, and EntityQuestions.

"answer": "Massey Ferguson 35",
"knowledge_points": ["The Massey Ferguson 35 is a tractor model.", "The Massey Ferguson 35 is equipped with a Perkins 3-cylinder diesel engine.", "The Massey Ferguson 35 is known for its global sales success.", "Agriline is a company that provides expert advice, parts, and customer service for Massey Ferguson tractors.", "Agriline's specialization indicates the brand it supports is Massey Ferguson."],
"knowledge_logic": "graph TD A[Question] –> B[T̈ractor model with Perkins 3-cylinder engine] B –> C[P̈erkins 3-cylinder engine is in Massey Ferguson 35] A –> D[T̈ractor with global sales success] D –> E[M̈assey Ferguson 35 is globally successful] A –> F[B̈rand supported by Agriline] F –> G[Ägriline specializes in Massey Ferguson] C –> H[M̈assey Ferguson 35] E –> H G –> I[B̈rand: Massey Ferguson] I –> H H –> J[Änswer: Massey Ferguson 35]"
Sample 2:

"question": "The Massey Ferguson 300 series was known for providing a certain type of power and featured a versatile gearbox; which company provides expert advice and customer service on Massey Ferguson parts, including those for this series?",
"answer": "Agriline",
"knowledge_points": ["The Massey Ferguson 300 series is a line of agricultural tractors known for robust power and a versatile gearbox.", "Massey Ferguson is a brand of agricultural machinery, and its parts often require specialized suppliers for maintenance.", "Agriline is a company specializing in providing parts, expert advice, and customer service for Massey Ferguson machinery.", "The question asks for a company that offers expert advice and customer service on Massey Ferguson parts, specifically referencing the 300 series.", "Agriline is identified as the provider meeting these criteria, based on its known role in the agricultural parts market."],

Density	entity-questions	simpleqa	simpleqa-verified	webquestions	nq	triviaqa	total loss
1e-4	1.3513	1.2646	1.4374	1.5477	1.5357	2.3828	9.5195
1	1.3583	1.2637	1.4505	1.5666	1.5275	2.3818	9.5484
1e4	1.3505	1.2651	1.4214	1.5523	1.5214	2.3708	9.4815
1e12	1.3635	1.2796	1.4479	1.5490	1.5141	2.3650	9.5191
1e20	1.3513	1.3088	1.4603	1.5944	1.5399	2.4023	9.6570
1e28	1.5318	1.4856	1.6531	1.8088	1.7258	2.6651	10.8702
1e36	2.1028	2.0886	2.2147	2.5668	2.3988	3.5138	14.8855

Table 8: Eval loss of knowledge injection on Qwen3-14B-Base (1B training tokens) across Web Questions, Natural Questions, TriviaQA, SimpleQA, SimpleQA-Verified, and EntityQuestions.

Density	entity-questions	simpleqa	simpleqa-verified	webquestions	nq	triviaqa	total loss
1e-4	1.5883	1.3732	1.5061	1.7098	1.8029	2.6993	10.6797
1	1.5975	1.3723	1.5051	1.7287	1.7687	2.6966	10.6689
1e4	1.6042	1.3750	1.5023	1.7192	1.7758	2.6859	10.6624
1e12	1.6126	1.3814	1.5213	1.7401	1.7792	2.6754	10.7101
1e20	1.6500	1.4005	1.5270	1.7343	1.7801	2.6647	10.7566
1e28	1.6693	1.4425	1.5595	1.7739	1.7983	2.6648	10.9083
1e36	2.2294	2.1333	2.1951	2.5390	2.5247	3.4122	15.0337

Table 9: Eval loss of knowledge injection on Qwen3-4B-Base (3B training tokens) across Web Questions, Natural Questions, TriviaQA, SimpleQA, SimpleQA-Verified, and EntityQuestions.

"knowledge_logic": "graph TD A[Massey Ferguson 300 series] –> BRequires parts/service B –> C[Specialized supplier needed] C –> D[Agriline: expert advice/customer service] D –> EMatches question criteria? E –> F[Yes: Agriline is answer]"

Sample 3:

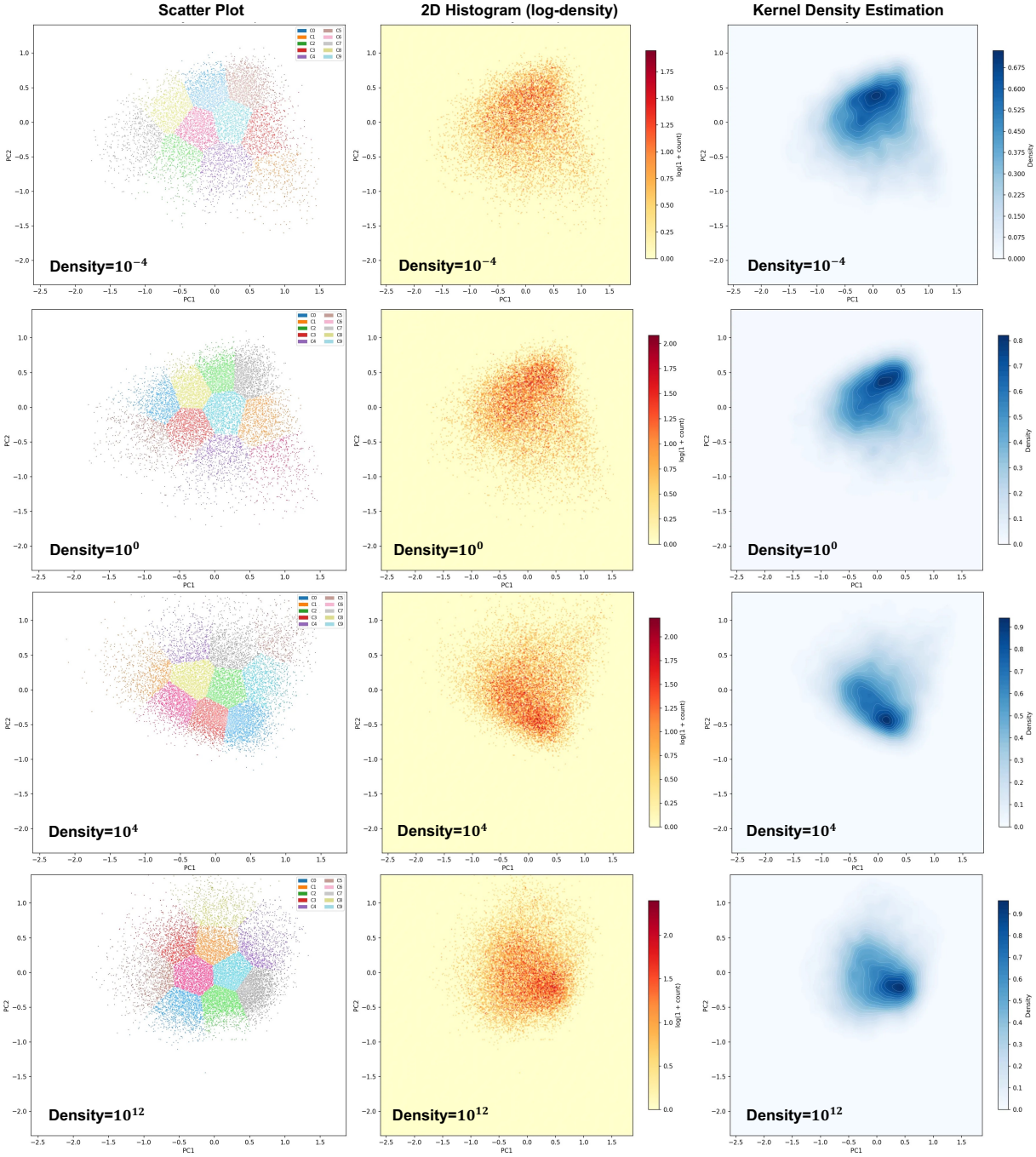
"question": "What organization is mentioned as providing expert advice and customer service on Massey Ferguson parts?",
"answer": "Agriline",
"knowledge_points": ["Massey Ferguson is a brand of agricultural machinery and equipment.", "Massey Ferguson parts are components used for repairing or maintaining this machinery.", "Organizations may provide services such as expert advice and customer support related to these parts.", "Agriline is a known supplier or service provider specializing in Massey Ferguson parts.", "The question asks for the organization associated with providing expert advice and customer service on these parts."],
"knowledge_logic": "graph TD A[Massey Ferguson parts] –> BWhich organization provides
expert advice & customer service?; B –> C[Agriline]; C –> D((Answer: Agriline));"

E.3 Algorithm Details

We detail the rejection sampling procedure from Sec. 3.3.3 in Algorithm 1 below.

Density	entity-questions	simpleqa	simpleqa-verified	webquestions	nq	triviaqa	total loss
1	1.6129	1.3096	1.4442	1.7082	1.7958	2.7011	10.5718
1e4	1.5996	1.3012	1.4351	1.7070	1.7738	2.6869	10.5036
1e12	1.5988	1.3091	1.4544	1.6942	1.7752	2.6829	10.5146
1e20	1.5937	1.3318	1.4848	1.7315	1.7803	2.6931	10.6152
1e28	1.6043	1.3581	1.5062	1.7393	1.7841	2.6939	10.6859
1e36	1.6143	1.3976	1.5341	1.7370	1.7885	2.6947	10.7662
1e44	2.2516	1.9151	2.0735	2.5331	2.5012	3.5991	14.8736

Table 10: Eval loss of knowledge injection on Qwen3-4B-Base (5B training tokens) across Web Questions, Natural Questions, TriviaQA, SimpleQA, SimpleQA-Verified, and EntityQuestions.



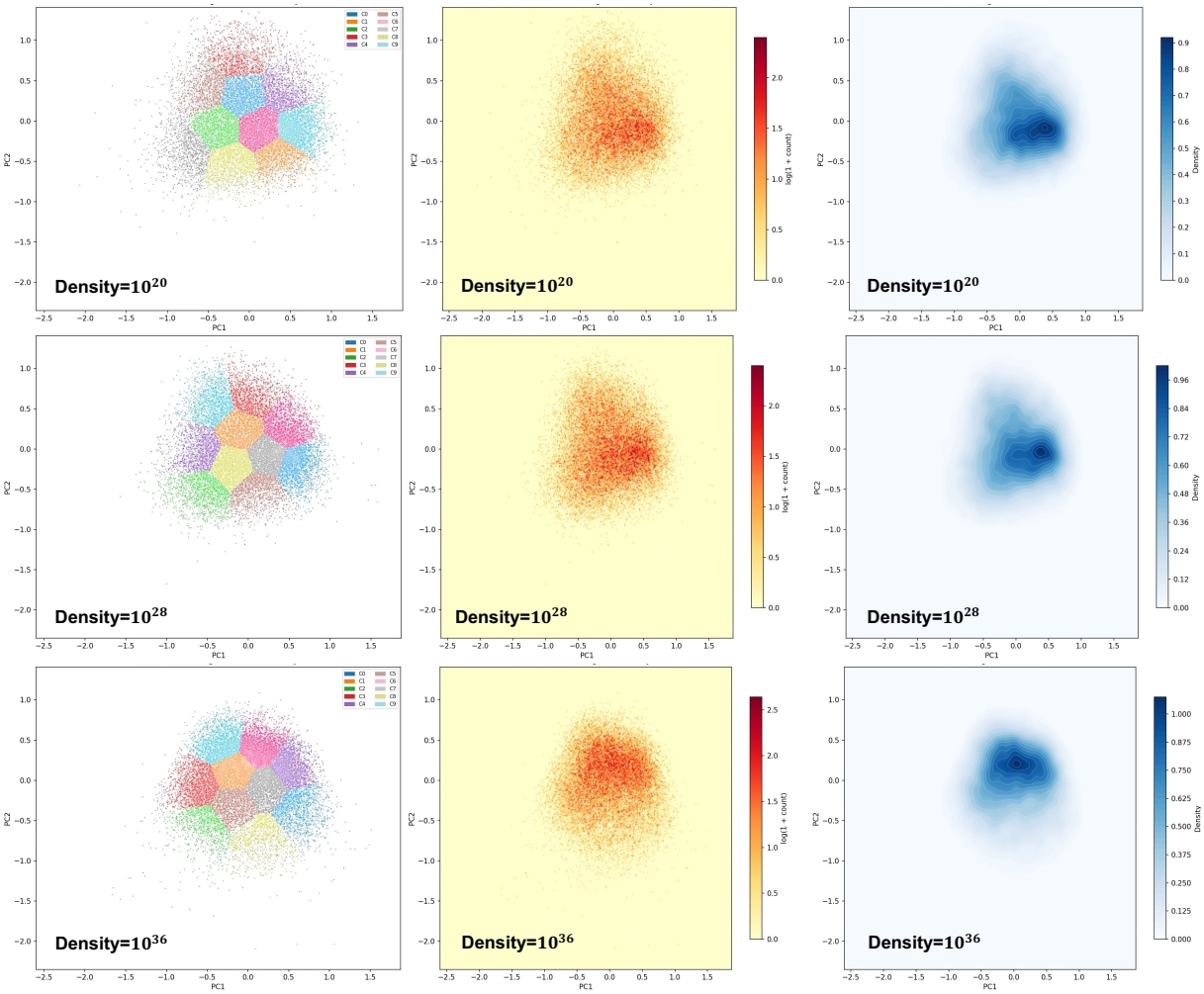


Figure 9: Case study: Visualization of data with different knowledge density distributions.

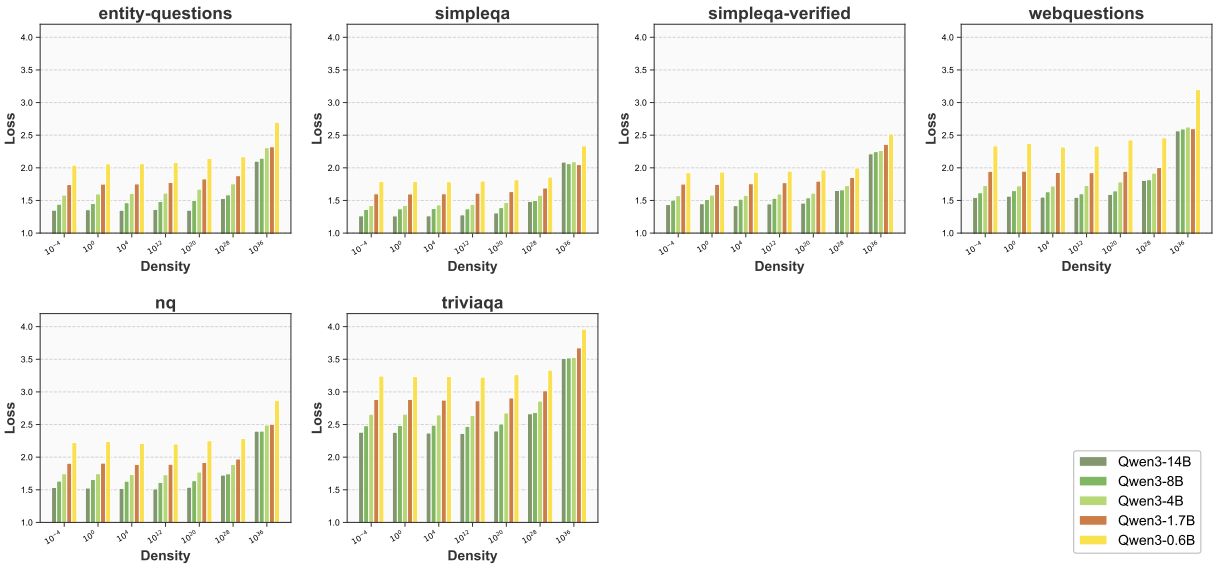


Figure 10: Eval loss of Qwen3-base models of different sizes on Web Questions, Natural Questions, TriviaQA, SimpleQA, SimpleQA-Verified, and EntityQuestions, trained with synthetic data of varying densities.