Synthetic Mixed Training: Scaling Parametric Knowledge Acquisition Beyond RAG

Anonymous ACL submission

Abstract

Synthetic data augmentation helps language models learn new knowledge in data-constrained domains. However, naively scaling existing synthetic data methods by training on more synthetic tokens or using stronger generators yields diminishing returns below the performance of RAG. To break the RAG ceiling, we introduce Synthetic Mixed Training, which combines synthetic QAs and synthetic documents. This leverages their complementary training signals, and enables log-linear improvements as both synthetic data volume and generator strength increase. This allows the model to outperform RAG by a 2.6% relative gain on QuaLITY, a long-document reading comprehension benchmark. In addition, we introduce Focal Rewriting, a simple technique for synthetic document generation that explicitly conditions document generation on specific questions, improving the diversity of synthetic documents and yielding a steeper log-linear scaling curve. On QuaLITY, our final recipe trains a Llama 8B model that outperforms RAG by 4.4% relatively. Across models and benchmarks (QuaLITY, LongHealth, FinanceBench), our training enables models to beat RAG in five of six settings, outperforms by 2.6%, and achieves a 9.1% gain when combined with RAG.

1 Introduction

Language models fail to internalize all knowledge during pretraining, so recent studies have investigated whether domain-specific fine-tuning can improve knowledge learning. They report that retrieval-augmented generation (RAG)—the de facto approach for data-constrained domains—sets a strong upper bound that is difficult to surpass (Ovadia et al., 2024; Soudani et al., 2024). This is because incorporating new knowledge into language model parameters is challenging in data-constrained settings. One common approach is to perform continued pretraining using synthetic data generated from domain-specific source documents. However, the vast majority of studies have found only limited success (Yang et al., 2025b; Lin et al., 2025a; Caccia et al., 2025; Eyuboglu et al., 2025; Lampinen et al., 2025). While it is natural to attribute this failure to the quality of the synthetic data generators, prior work has observed that stronger generators yield diminishing returns (Lin et al., 2025a; Maini et al., 2025; Kang et al., 2025; Maini et al., 2024; Niklaus et al., 2026). In this work, we address these issues by answering the question:

How can we design synthetic data recipes for knowledge learning that scales better with number of synthetic tokens data and stronger generator?

First, we investigate existing data generation algorithms and find that they do not scale well. We experiment with four existing data generation algorithms—one for generating synthetic QA pairs and three for generating synthetic documents—and use Llama 3.1 8B and 70B models to generate up to 700M synthetic tokens for training an 8B model. We find that, on a reading comprehension benchmark requiring the acquisition of new knowledge (QuaLITY; Pang et al. (2022)), **existing training recipes based on synthetic data are insufficient to train a model that outperforms RAG, even when the data is generated by a 70B model that is much stronger than the 8B model being trained**. In particular, we find that training on synthetic QAs performs better than training on synthetic documents for a fixed generator. However, the gains from scaling the generator depend on the choice of data generation algorithm—document generation benefits more from generator scaling. Still, all methods exhibit diminishing returns as the number of synthetic tokens increases and remain 4.6% behind RAG in relative accuracy.

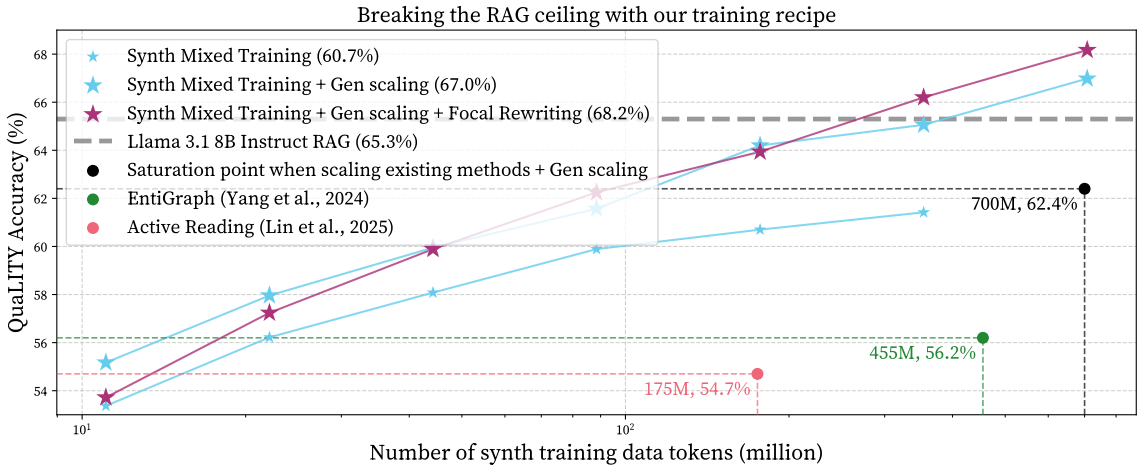

Figure 1: **Naively scaling synthetic data plateaus, but our simple methods allows effective scaling and surpass RAG.** We evaluate four synthetic data generation strategies using both 8B and 70B generators, scaling training data up to 700M tokens. Across all four baselines, performance saturates and remains below RAG, showing that simply increasing synthetic data or compute is insufficient. In contrast, our two simple techniques—**Synthetic Mixed Training** and **Focal Rewriting**—exhibit clear log-linear scaling with both more data and a stronger generator, ultimately surpassing RAG.

Building on the observations that synthetic QAs and documents have different scaling properties with respect to data and generator strength, we hypothesize that QA and document data each provide unique benefits during training. To achieve the best of both worlds, we propose **Synthetic Mixed Training**, which combines synthetic QAs with synthetic documents during training. This substantially improves synthetic token efficiency, exhibits clear log-linear scaling behavior up to 700M synthetic tokens, and enables the model to surpass RAG by a 2.6% relative gain. In addition, we introduce **Focal Rewriting**, a simple technique for diversifying the topics covered by synthetic documents by explicitly conditioning document generation on synthetic questions about the source document. This increases the lexical and semantic diversity of the synthetic documents and further improves model performance, yielding a 4.4% relative accuracy gain over RAG.

We show that our recipe generalizes well across different base models, including Qwen3 models ranging from 1.7B to 14B parameters, and across three benchmarks: QuaLITY, LongHealth, and FinanceBench. In particular, our recipe enables models to outperform RAG in five of six setups, yielding an average relative accuracy gain of 2.6% over vanilla RAG. Our approach is also complementary to RAG, providing a 9.1% relative accuracy improvement over vanilla RAG. These results suggest an untapped potential of synthetic data in enhancing internalization of new knowledge for language models.

2 Existing synthetic data recipes plateau when scaled

We present experimental results for training an 8B model on variants of synthetic data derived from documents in the QuaLITY benchmark (Pang et al., 2022). To study how performance changes with data scale, we vary the number of synthetic tokens across runs, scaling up to 700M tokens. As Yang et al. (2025b) demonstrate that training with only real data does not improve performance, we only consider options that train only with synthetic data.

Setup. We train the Llama 3.1 8B Instruct model (Grattafiori et al., 2024) on the synthetic data using a fixed set of hyperparameters (except for learning rate; we search across three LRs—5e-6, 1e-6 and 5e-5—and report the best accuracy), varying only the method used to generate the synthetic data. Similar to Yang et al. (2025b), we use FineWeb (Penedo et al., 2024) as a replay data, with a mixing ratio of 10%. For evaluation, we use the QuaLITY multiple-choice QA set with a zero-shot instruction that asks the LM to provide a short explanation followed by the final answer. We run evaluations for each setup eight times and report the average results for more reliable results. See Appendix B and C for more details.

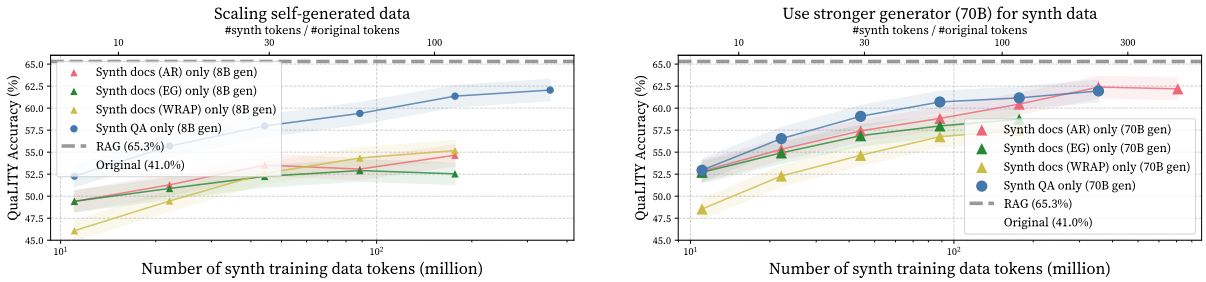

Figure 2: (Left) **Comparing the data scaling of existing methods: self-generated (8B) synthetic QAs and synthetic documents.** This shows QuaLITY accuracy as a function of the number of synthetic training tokens; shaded areas indicate the 95% confidence interval, estimated from $n = 8$ inference runs. We use Llama 3.1 8B Inst for both data generation and model training. AR indicates Active Reading (Lin et al., 2025a), EG indicates EntiGraph (Yang et al., 2025b), and WRAP indicates rephrasing (Maini et al., 2024). On QuALITY, synth QA is substantially more efficient than all existing methods which generate synthetic documents. (Right) **Scaling the generator to improve synthetic token efficiency.** (1) Scaling the generator to 70B does not improve synth token efficiency for QA, only 0.1% gain over 8B generator at 88M tokens. (2) In contrast, document-based methods do benefit from scaling the generator, achieving 4.5% gain on average. (3) For all methods, even with a stronger generator, data scaling plateaus.

Synthetic document generation. We test four data generation methods. All of these methods take original source documents as input and generate data grounded in them, and we do not consider approaches that require seed examples, such as (Divekar and Durrett, 2024). For synthetic document generation, we test three algorithms. The first is rephrasing (WRAP; Maini et al. (2024)), which rephrases documents using an LM. The second is EntiGraph (EG; Yang et al. (2025b)), which uses a two-stage approach: it first extracts core entities from the document using an LM, and then constructs documents describing the relationships between those entities using an LM again. The last is Active Reading (AR; Lin et al. (2025a)), which also uses a two-stage approach–first, an LM generates strategies to rephrase the document, which are then incorporated into the instruction to guide the LM in rewriting the document.

Synthetic QA generation. For synthetic QA generation, we first generate diverse QA pairs using an LM with a simple instruction, and then use an LM again to produce a response with a short explanation for each question. This is similar to the QA generation approaches of Lin et al. (2025a) and Yang et al. (2025b), except that we explicitly instruct the LMs to generate explanations in the responses. We generate open-ended QA pairs rather than multiple-choice QA (MCQA) pairs, since generating difficult yet faithful distractors for MCQA is nontrivial. See Appendix D for more details.

2.1 Scaling self-generated synthetic data has diminishing returns

Figure 2 (left) shows the evaluation results on QuaLITY when we train the model on synthetic data generated by the same 8B model (i.e., self-generated data). Training on synthetic QAs is more synthetic token efficient than training on any of the three synthetic document variants, while scaling synthetic documents begins to plateau. This was unexpected, as prior work found a different empirical result when training an 8B model with 8B-generated data (Lin et al., 2025a). We speculate that the difference arises from their synthetic QA pairs containing only short answers without explanations, as we find that training the model on responses containing only the answer, without an explanation (i.e., using the answers from the initially generated QA pairs), leads to poor accuracy. However, when we specifically scale synthetic QAs up to 350M tokens, it also begins to show signs of plateauing.

2.2 Strategy matters when scaling generator to improve synthetic token efficiency

We further explore a natural direction for improving synthetic token efficiency: scaling the generator model (use Llama 3.1 70B Instruct) to produce higher-quality synthetic data. Figure 2 (right) shows the results of generator scaling for synthetic data generation. For synthetic QA generation, a stronger generator does not yield meaningful im-

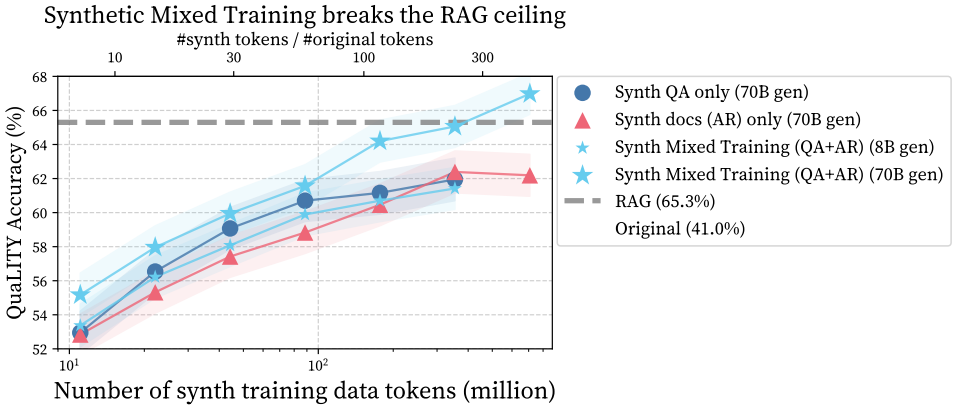

Figure 3: **Synthetic Mixed Training breaks the RAG ceiling.** We combine 70B-generated synthetic QAs and AR documents at a 1:1 ratio, attempting to achieve the best of both worlds. This (skyblue line) yields performance comparable to RAG at 350M synthetic training tokens and ultimately surpasses RAG when scaled to 700M tokens.

provements in synthetic token efficiency, and performance also saturates as we scale the number of synthetic QA tokens. In contrast, for document generation, a stronger generator improves synthetic token efficiency: it closes the gap between training on synthetic QAs and training on synthetic documents. The choice of data generation algorithm also matters greatly: AR shows the best synthetic token efficiency among the variants when using the 70B generator, whereas WRAP shows the worst synthetic token efficiency relative to its performance with the 8B generator. However, 70B-generated AR documents also shows saturating performance when scaled up to 700M tokens.

The results suggest that **we should think carefully about which generation procedures are likely to improve with more capable generators**, as not every data generation algorithm benefits from a stronger generator model. Prior synthetic data works report related empirical findings that stronger generators do not always provide better synthetic data (without clear explanation): Lin et al. (2025a) shows that using 70B-generated data to train the 8B model underperforms compared to using 8B generated data; Guha et al. (2025) shows that a weaker, smaller generator (QwQ-32B) can produce better synthetic data for math and code than a stronger, larger generator (DeepSeek-R1); and Maini et al. (2025) observes saturating returns when scaling the generator from a 3B to an 8B model for document rephrasing in pretraining setups. Kang et al. (2025) shows that using a 70B model does not yield better performance when rephrasing pre-training data, and more recently, Niklaus et al. (2026) shows that a 1B generator is sufficient for rephrasing, supported by extensive experimental results.

3 Unlocking synthetic data scaling

Building on our empirical findings, we present two simple methods for improving the efficiency of synthetic data and overcoming the limitations of existing synthetic data scaling. Our methods enable clear log-linear scaling up to 700M training tokens, and our trained 8B model substantially outperforms RAG on QuaLITY, achieving a 4.4% relative accuracy gain. In addition, we find that these methods also work well with different base models (Qwen3 1.7B–14B) and additional benchmarks (LongHealth, FinanceBench).

3.1 Synthetic Mixed Training: Mixing synthetic QAs and synthetic documents

Since we have shown synthetic data strategies have different scaling properties, we investigate whether we can design a synthetic data recipe that achieves the best of all worlds.

We hypothesize that synthetic QAs and synthetic documents play different roles: synthetic QAs primarily teach *behavioral knowledge* (e.g., how to recall facts through chain-of-thought reasoning), which can transfer across domains, whereas synthetic documents primarily teach *factual knowledge*, which is more domain-specific. To validate this hypothesis, we test mixing all three types of synthetic documents during training, and Figure 4 shows that the synthetic document types are similar and provide minimal improvement when mixed. Additionally, we measure the similarity between data points in gradient space (Jung et al., 2025) to

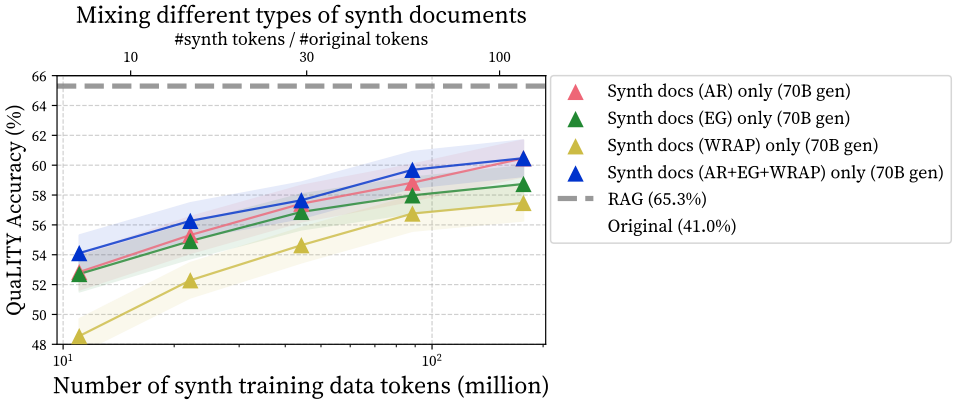

Figure 4: **Mixing synthetic documents does not help.** Mixing different kinds of synthetic documents (blue line) provides a minimal gain over just using AR documents (pink line).

quantify how synthetic QAs and documents from different domains are similar, see Appendix E for the analysis on this.

Based on this hypothesis, we explore training models with a mixture of synthetic QAs and synthetic documents. We call this approach **Synthetic Mixed Training**.[1] We choose AR for document generation because it benefits the most from generator scaling, and we mix QAs and AR documents at a 1:1 ratio. Figure 3 shows the results of Synthetic Mixed Training using 70B-generated QA and AR documents, compared with training only on synthetic QAs or only on AR documents. The scaling curve exhibits persistent log-linear behavior as the number of training tokens increases up to 700M, eventually surpassing RAG with 67.0% accuracy (+2.6% relative gain). Moreover, when comparing results using 70B- and 8B-generated data for Synthetic Mixed Training, we find that a stronger generator is substantially more helpful. This suggests that QA simply dominates AR at 8B generated scale, highlighting AR's unique benefit from generator scaling.

We further test whether target-domain synthetic QAs are important by mixing synthetic QAs from a different dataset (LongHealth, an unrelated domain) with synthetic AR documents from QuaLITY (the target domain). Figure 5 shows two findings. (1) Synthetic QAs can teach domain-agnostic behavior that is difficult to learn from AR documents alone: mixing LongHealth synthetic QAs with QuaLITY synthetic AR documents improves synthetic token efficiency on QuaLITY compared to using only QuaLITY synthetic AR documents. This supports our hypothesis and helps explaining the synergy between synthetic QAs and AR documents. (2) Synthetic QAs also teach domain-specific factual knowledge: the best-performing recipe mixes both synthetic QAs and synthetic AR documents from QuaLITY, outperforming the mixture of LongHealth synthetic QAs and QuaLITY synthetic AR documents. This suggests that target-domain synthetic QAs provide not only transferable behavior but also domain-specific knowledge.

3.2 Focal Rewriting: Improving the topic diversity of synthetic documents

In Section 2, we show that scaling synthetic documents yields diminishing returns, which we hypothesize is due to limited diversity in the generated data. In particular, WRAP and AR documents produce *stylistically* diverse documents, but the generated documents often cover highly similar *topics*. This is because these approaches do not explicitly condition the LM on specific topics; instead, the model implicitly decides what to focus on, leading to repeated or overlapping topics across generations. In contrast, EG generates documents covering more diverse *topics* by explicitly conditioning on different entities, but the resulting documents tend to be *stylistically* similar. We suspect that this limited diversity degrades performance when synthetic data is scaled extensively, for example, when the number of synthetic tokens exceeds that of the original documents by more than 100 times.

Based on this observation, we introduce **Focal Rewriting** which can diversify both the content and the style of generated documents. When rewriting documents with AR (or WRAP), we explicitly con-

[1]This name is inspired by Mixed Training, introduced by Allen-Zhu and Li (2024). Their approach is designed for pretraining from scratch and does not consider synthetic documents when mixing.

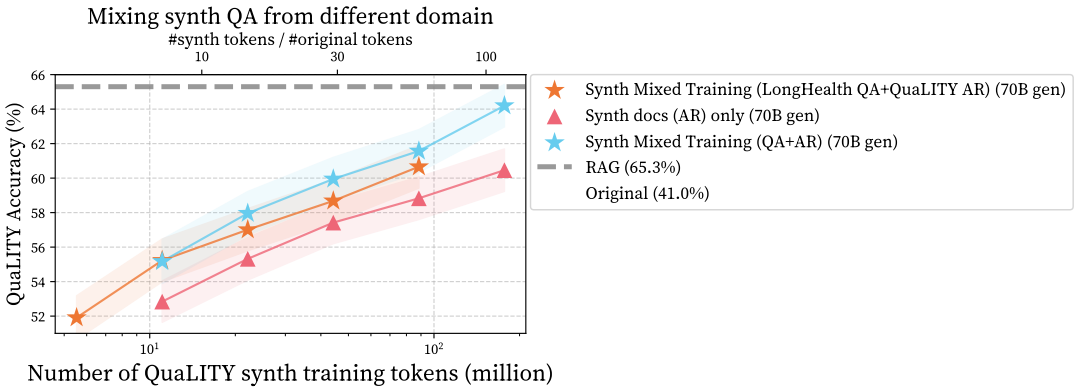

Figure 5: **Synthetic Mixed Training with mixture of domains.** Here, the x-axis denotes the number of synthetic tokens grounded in the QuaLITY dataset. (1) Mixing 50% synthetic QAs grounded in a different domain with 50% synthetic documents grounded in the target domain yields a better scaling curve than training solely on target-domain synthetic documents. (2) The best performance comes from mixing target-domain synthetic QAs with target-domain synthetic documents, suggesting that synthetic QAs not only teaches recall behavior but also provides domain-specific knowledge.

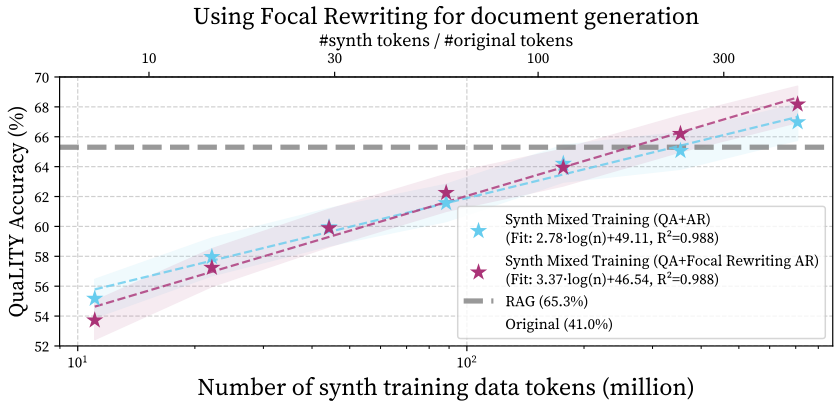

Figure 6: **Scaling synthetic document generation with Focal Rewriting.** We apply Focal Rewriting to AR when generating synthetic documents. Synthetic documents generated with Focal Rewriting (purple line) exhibit better scaling behavior than those generated without it (skyblue line), as shown by the steeper slope of the fitted scaling curve. Data are all generated using 70B model.

dition generation on a specific question, asking for a document that would be useful for answering that query. This technique can be implemented by simply adding the clause "`Focus on the question {{query}}`" in the generation instruction (see Appendix D for the prompts). We use the questions generated by LM as in Section 2. As shown in Appendix F, when we apply Focal Rewriting to AR, this produces documents with greater lexical and semantic diversity.

Figure 6 shows the results of scaling synthetic data using AR with Focal Rewriting. We find that when doing Synthetic Mixed Training, accuracy follows a log-linear relationship with the number of synthetic tokens. Accordingly, we fit a log-linear curve and plot it alongside the empirical results. The results show that Focal Rewriting yields a steeper log-linear scaling curve and achieves higher accuracy when data is scaled extensively (beyond 175M tokens; 100× more tokens than the original data).

3.3 Testing on different model and benchmarks

We additionally verify our recipe on another base model (Qwen3 8B; Yang et al. (2025a)) and two additional benchmarks (LongHealth; Adams et al. (2025) and FinanceBench; Islam et al. (2023)) that require learning new knowledge. Table 1 shows the key statistics of the datasets. In particular, FinanceBench provides source documents in PDF format, so we use `olmOCR-2-7B-1025` (Poznanski

Dataset	#docs	#avg tokens/docs	#eval	Eval type	Domain
QuaLITY	265	6K	4609	MCQA	Fictional stories
LongHealth	400	12K	400	MCQA	Medical
FinanceBench	1367	16K	150	Free-form	Finance

Table 1: **Dataset statistics.** `#docs` indicates the number of source documents used for data generation, and `#avg tokens/docs` indicates the average number of tokens per source document. `#eval` indicates the number of QA sets used in evaluation. For FinanceBench, we use Qwen3-14B to judge the model-generated answer using gold answer as a reference.

et al., 2025) to preprocess them into Markdown and use them as source documents.

Figure 8 shows the results. On QuaLITY and LongHealth, our training recipe enables the models to outperform RAG, and on FinanceBench, our methods allow the Llama 8B model to outperform RAG. On average, our method gives 2.6% relative accuracy gain compared to RAG. Our method also outperforms training recipes that scale only synthetic QAs or synthetic AR documents.

3.4 Training larger models is more synthetic token efficient

We train four Qwen3 models of different sizes (1.7B, 4B, 8B, 14B) to study how model size affects scaling behavior. We keep all configurations (including the learning rate) the same from the previous experiments. Figure 7 shows that these models all show log-linear scaling when tokens are scaled up to 88M, and larger models are more synthetic token efficient, achieving RAG-level performance with fewer synthetic tokens according to the fitted log-linear curve. For instance, the 14B model requires 102× more synthetic tokens than the original data, whereas the 1.7B model requires 813× more. This result is intuitive, as larger models have greater capacity to store knowledge (Morris et al., 2025).

3.5 Our training recipe enhances RAG

We test whether our trained model can be improved further when paired with retrieval augmentation. As shown in Table 2, this shows clear improvement when using RAG with our trained model and shows that it significantly outperforms the vanilla RAG baselines. On average, our trained models combined with RAG provides a 9.1% relative gain when compared to RAG. This suggests that domain-specific training with synthetic data augmentation can be helpful even when RAG is used.

4 Related Work

Training with synthetic data. Training language models with synthetic data at scale has become an important practice (Blakeman et al., 2025; Yang et al., 2025a; Abdin et al., 2024). At the pre-training stage, language models are often used to rewrite original data, such as web text. For example, Maini et al. (2024) use language models to rephrase original documents, while Nguyen et al. (2025) use language models to reason before rephrasing documents, resulting in higher-quality rewrites. More recently, Yang et al. (2025c) propose to train a language model to generate new documents conditioned on an input document.

In continued pre-training settings, where language models are further trained on domain-specific data after pre-training (Gururangan et al., 2020), more aggressive forms of data augmentation are often used because these settings are typically data-constrained and do not provide enough data for scaling (Muennighoff et al., 2023; Kim et al., 2025). For these settings, improving the diversity of generated data is important: to this end, Yang et al. (2025b) uses language models to extract core entities about the documents, and then generate synthetic documents that describe relations between entities in the original documents, thereby improving the diversity of the synthetic data. They empirically show that accuracy improves in a log-linear trend as the number of synthetic data tokens increases. Similarly, Lin et al. (2025a) use language models to diversify rewriting strategies. In a more domain-specific direction, Ruan et al. (2025) use reasoning traces produced by language models to capture the underlying thought processes related to the original documents, and show that these traces are helpful for continued training in math.

Analyzing knowledge of language models. How language models acquire knowledge during training and use that knowledge when performing downstream tasks is still not fully understood. To

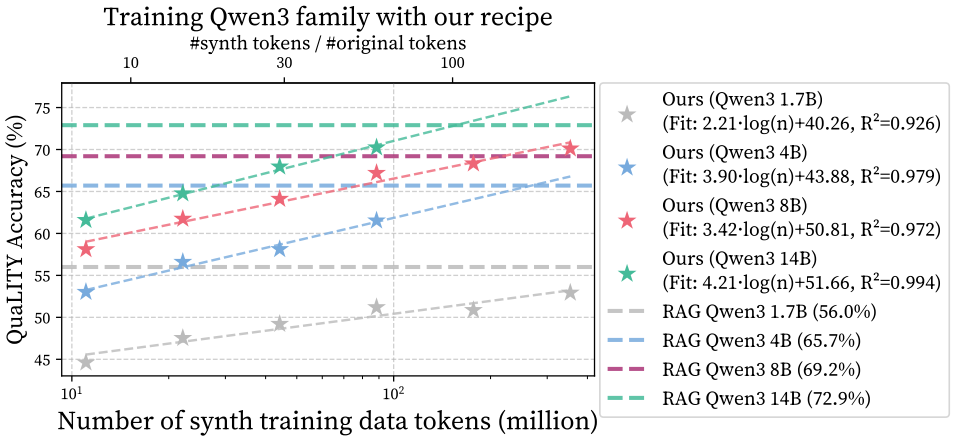

Figure 7: **Training different-sized models from the Qwen3 family on the synthetic QuaLITY dataset.** We train using our best recipe: Mixed Synthetic Training with synthetic QAs and Focal Rewriting AR documents, using 70B generator. All models exhibit log-linear scaling behavior, and larger models can match RAG performance with fewer synthetic tokens. Based on the fitted curves, we observe that the 14B model requires 102×, the 8B model 142×, the 4B model 177×, and the 1.7B model 813× more synthetic tokens than the original token count (shown as `#synth tokens/#original tokens`).

Benchmark	Model	Ours	Ours + RAG	Vanilla RAG	Δ
QuaLITY	Llama	68.2%	69.7%	65.3%	+4.4
	Qwen	70.1%	73.6%	69.2%	+4.4
LongHealth	Llama	71.2%	80.3%	70.0%	+10.3
	Qwen	76.7%	82.3%	73.7%	+8.6
FinanceBench	Llama	52.6%	54.3%	48.0%	+6.3
	Qwen	55.5%	60.2%	58.8%	+1.4

Table 2: **Our training complements retrieval augmentation.** "Ours" denotes the trained model without RAG, "Ours + RAG" denotes the same trained model with RAG, and "Vanilla RAG" denotes the RAG baseline. Δ denotes the absolute improvement of "Ours + RAG" over "Vanilla RAG". Averaged across all settings, our training improves over Vanilla RAG by 5.9 points.

help understand this, Allen-Zhu and Li (2024) and Allen-Zhu and Li (2025) conduct systematic studies on small language models and show that both storing knowledge in the parameters and learning how to use that knowledge are important. More recently, Calderon et al. (2026) argue that even frontier models are bottlenecked more by knowledge recall than by knowledge storage, and Gekhman et al. (2026); Ma and Hewitt (2026) show that reasoning can improve fact recall, indicating that teaching language models how to use the facts learned during training is important. We believe that the success of Synthetic Mixed Training aligns well with these findings, as synthetic QAs may help models learn how to use knowledge, highlighting the importance of knowledge use beyond mere storage in language models.

5 Conclusion

We study how to make synthetic data scale more effectively for knowledge learning in data-constrained domains. Our results show that simply increasing the amount of synthetic data or using a stronger generator is not sufficient: existing methods exhibit diminishing returns and still underperform RAG. Based on the observation that synthetic QAs and documents have different scaling properties, we introduce Synthetic Mixed Training, which combines synthetic QAs and documents to leverage their complementary training signals and achieve the best of both worlds. We further introduce Focal Rewriting, which improves the diversity of generated documents and leads to an even steeper scaling trend. Our methods generalize well across a range of settings, and the trained models are also complementary to RAG.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. arXiv preprint arXiv:2412.08905.

Lisa Adams, Felix Busch, Tianyu Han, Jean-Baptiste Excoffier, Matthieu Ortala, Alexander Löser, Hugo JWL Aerts, Jakob Nikolas Kather, Daniel Truhn, and Keno Bressem. 2025. Longhealth: A question answering benchmark with long clinical documents. Journal of Healthcare Informatics Research, 9(3):280–296.

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.

Zeyuan Allen-Zhu and Yuanzhi Li. 2024. Physics of language models: Part 3.1, knowledge storage and extraction. In International Conference on Machine Learning, pages 1067–1077. PMLR.

Zeyuan Allen-Zhu and Yuanzhi Li. 2025. Physics of language models: Part 3.2, knowledge manipulation. In International Conference on Learning Representations.

Vincent-Pierre Berges, Barlas Oğuz, Daniel Haziza, Wen-tau Yih, Luke Zettlemoyer, and Gargi Ghosh. 2024. Memory layers at scale. arXiv preprint arXiv:2412.09764.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, and 1 others. 2025. Lora learns less and forgets less. Transactions on Machine Learning Research.

Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, and 1 others. 2025. Nvidia nemotron 3: Efficient and open intelligence. arXiv preprint arXiv:2512.20856.

Lucas Caccia, Alan Ansell, Edoardo Ponti, Ivan Vulić, and Alessandro Sordoni. 2025. Training plug-and-play knowledge modules with deep context distillation. In Second Conference on Language Modeling.

Nitay Calderon, Eyal Ben-David, Zorik Gekhman, Eran Ofek, and Gal Yona. 2026. Empty shelves or lost keys? recall is the bottleneck for parametric factuality. arXiv preprint arXiv:2602.14080.

Abhishek Divekar and Greg Durrett. 2024. Synthesizrr: Generating diverse datasets with retrieval augmentation. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 19200–19227.

Sabri Eyuboglu, Ryan Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Liu, Will Tennien, Atri Rudra, James Zou, Azalia Mirhoseini, and 1 others. 2025. Cartridges: Lightweight and general-purpose long context representations via self-study. arXiv preprint arXiv:2506.06266.

Dan Friedman and Adji Bousso Dieng. 2023. The vendi score: A diversity evaluation metric for machine learning. Transactions on Machine Learning Research.

Tianyu Gao, Alexander Wettig, Luxi He, Yihe Dong, Sadhika Malladi, and Danqi Chen. 2025. Metadata conditioning accelerates language model pre-training. arXiv preprint arXiv:2501.01956.

Zorik Gekhman, Roee Aharoni, Eran Ofek, Mor Geva, Roi Reichart, and Jonathan Herzig. 2026. Thinking to recall: How reasoning unlocks parametric knowledge in llms. Preprint, arXiv:2603.09906.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 5484–5495.

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.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, and 1 others. 2025. Openthoughts: Data recipes for reasoning models. arXiv preprint arXiv:2506.04178.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In Proceedings of the 58th annual meeting of the association for computational linguistics, pages 8342–8360.

Xu Owen He. 2024. Mixture of a million experts. arXiv preprint arXiv:2407.04153.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. In International Conference on Learning Representations.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. 2023. Financebench: A new benchmark for financial question answering. arXiv preprint arXiv:2311.11944.

William B Johnson, Joram Lindenstrauss, and 1 others. 1984. Extensions of lipschitz mappings into a hilbert space. Contemporary mathematics, 26(189-206):1.

Jaehun Jung, Seungju Han, Ximing Lu, Skyler Hallinan, David Acuna, Shrimai Prabhumoye, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, and Yejin Choi. 2025. Prismatic synthesis: Gradient-based data diversification boosts generalization in llm reasoning. In The Thirty-ninth Annual Conference on Neural Information Processing Systems.

Feiyang Kang, Newsha Ardalani, Michael Kuchnik, Youssef Emad, Mostafa Elhoushi, Shubhabrata Sengupta, Shang-Wen Li, Ramya Raghavendra, Ruoxi Jia, and Carole-Jean Wu. 2025. Demystifying synthetic data in llm pre-training: A systematic study of scaling laws, benefits, and pitfalls. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 10750–10769.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. arXiv preprint arXiv:2001.08361.

Konwoo Kim, Suhas Kotha, Percy Liang, and Tatsunori Hashimoto. 2025. Pre-training under infinite compute. arXiv preprint arXiv:2509.14786.

Suhas Kotha and Percy Liang. 2026. Replaying pre-training data improves fine-tuning. arXiv preprint arXiv:2603.04964.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In Proceedings of the 29th symposium on operating systems principles, pages 611–626.

Andrew Kyle Lampinen, Martin Engelcke, Yuxuan Li, Arslan Chaudhry, and James L McClelland. 2025. Latent learning: episodic memory complements parametric learning by enabling flexible reuse of experiences. arXiv preprint arXiv:2509.16189.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in neural information processing systems, 33:9459–9474.

Jessy Lin, Vincent-Pierre Berges, Xilun Chen, Wen-Tau Yih, Gargi Ghosh, and Barlas Oğuz. 2025a. Learning facts at scale with active reading. arXiv preprint arXiv:2508.09494.

Jessy Lin, Luke Zettlemoyer, Gargi Ghosh, Wen-Tau Yih, Aram Markosyan, Vincent-Pierre Berges, and Barlas Oğuz. 2025b. Continual learning via sparse memory finetuning. arXiv preprint arXiv:2510.15103.

Emmy Liu, Graham Neubig, and Chenyan Xiong. 2025. Midtraining bridges pretraining and posttraining distributions. arXiv preprint arXiv:2510.14865.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

Kevin Lu and Thinking Machines Lab. 2025. On-policy distillation. Thinking Machines Lab: Connectionism. Https://thinkingmachines.ai/blog/on-policy-distillation.

Melody Ma and John Hewitt. 2026. Improving parametric knowledge access in reasoning language models. arXiv preprint arXiv:2602.22193.

Pratyush Maini, Vineeth Dorna, Parth Doshi, Aldo Carranza, Fan Pan, Jack Urbanek, Paul Burstein, Alex Fang, Alvin Deng, Amro Abbas, and 1 others. 2025. Beyondweb: Lessons from scaling synthetic data for trillion-scale pretraining. arXiv preprint arXiv:2508.10975.

Pratyush Maini, Skyler Seto, Richard Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. 2024. Rephrasing the web: A recipe for compute and data-efficient language modeling. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14044–14072.

John X Morris, Chawin Sitawarin, Chuan Guo, Narine Kokhlikyan, G Edward Suh, Alexander M Rush, Kamalika Chaudhuri, and Saeed Mahloujifar. 2025. How much do language models memorize? arXiv preprint arXiv:2505.24832.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. 2023. Scaling data-constrained language models. Advances in Neural Information Processing Systems, 36:50358–50376.

Thao Nguyen, Yang Li, Olga Golovneva, Luke Zettlemoyer, Sewoong Oh, Ludwig Schmidt, and Xian Li. 2025. Recycling the web: A method to enhance pre-training data quality and quantity for language models. In Second Conference on Language Modeling.

Joel Niklaus, Guilherme Penedo, Hynek Kydlicek, Elie Bakouch, Lewis Tunstall, Ed Beeching, Thibaud Frere, Colin Raffel, Leandro von Werra, and Thomas Wolf. 2026. The synthetic data playbook: Generating trillions of the finest tokens.

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.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question answering with long input texts, yes! In Proceedings of the 2022 Conference of the North American Chapter

of the Association for Computational Linguistics: Human Language Technologies, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, Thomas Wolf, and 1 others. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. Advances in Neural Information Processing Systems, 37:30811–30849.

Jake Poznanski, Luca Soldaini, and Kyle Lo. 2025. olmocr 2: Unit test rewards for document ocr. arXiv preprint arXiv:2510.19817.

Yangjun Ruan, Neil Band, Chris J Maddison, and Tatsunori Hashimoto. 2025. Reasoning to learn from latent thoughts. arXiv preprint arXiv:2503.18866.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538.

Charlie Snell, Dan Klein, and Ruiqi Zhong. 2022. Learning by distilling context. arXiv preprint arXiv:2209.15189.

Heydar Soudani, Evangelos Kanoulas, and Faegheh Hasibi. 2024. Fine tuning vs. retrieval augmented generation for less popular knowledge. In Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, pages 12–22.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. arXiv preprint arXiv:2505.09388.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candes, and Tatsunori Hashimoto. 2025b. Synthetic continued pretraining. In The Thirteenth International Conference on Learning Representations.

Zitong Yang, Aonan Zhang, Hong Liu, Tatsunori Hashimoto, Emmanuel Candès, Chong Wang, and Ruoming Pang. 2025c. Synthetic bootstrapped pretraining. arXiv preprint arXiv:2509.15248.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, and 1 others. 2023. Pytorch fsdp: experiences on scaling fully sharded data parallel. arXiv preprint arXiv:2304.11277.

Adam Zweiger, Xinghong Fu, Han Guo, and Yoon Kim. 2026. Fast kv compaction via attention matching. arXiv preprint arXiv:2602.16284.

A Limitations

Due to compute limitations, our study focuses on small-scale models up to 8B parameters, which stand to benefit the most from novel methods for knowledge learning. In addition, although we focus on learning new knowledge, mitigating the forgetting of existing knowledge is also an important problem. We use pretraining data replay (Yang et al., 2025b; Kotha and Liang, 2026) to mitigate this forgetting issue, and we view a deeper treatment of forgetting alongside knowledge acquisition as a promising direction for future work.

B Training details

Hyperparameters. We train all models on the synthetic data using a fixed set of hyperparameters: a batch size of 16, a sequence length of 2048, two training epochs, and a fixed replay rate of 0.1 from the FineWeb dataset (Penedo et al., 2024). We use cosine learning rate schedule with warm up ratio of 0.05, use AdamW optimizer (Loshchilov and Hutter, 2017) with weight decay of 0.01, beta1 of 0.9, and beta2 of 0.999. We also use gradient clipping of threshold 1.0 and use FSDP2 (Zhao et al., 2023) for model training (used 4 GPUs for training 8B models). The only variation is the method used to generate the synthetic data. For Llama 3.1 8B Instruct, we train models with two learning rates (5e-6 and 1e-5) and report the best result for each configuration. For Qwen3 models (1.7B, 4B, 8B, and 14B), we use two learning rates (1e-5 and 5e-5) and again report the best result for each configuration. To use training compute efficiently, as in pretraining, we pack randomly shuffled data instances into each sequence, separated by the EOD delimiter.

Data formatting: inclusion of metadata is important. After generating synthetic data, we emphasize that data formatting is also important: metadata about the data should be included (e.g., company name, story title and author name, etc). For example, if the generated data (using Active Reading) is based on FinanceBench, it could follow this format:

In our preliminary experiments, we found that omitting metadata leads to lower accuracy after training. This is reasonable, as without metadata, the model struggles to associate knowledge with the correct source. This observation is also consistent with the findings of Allen-Zhu and Li (2025)

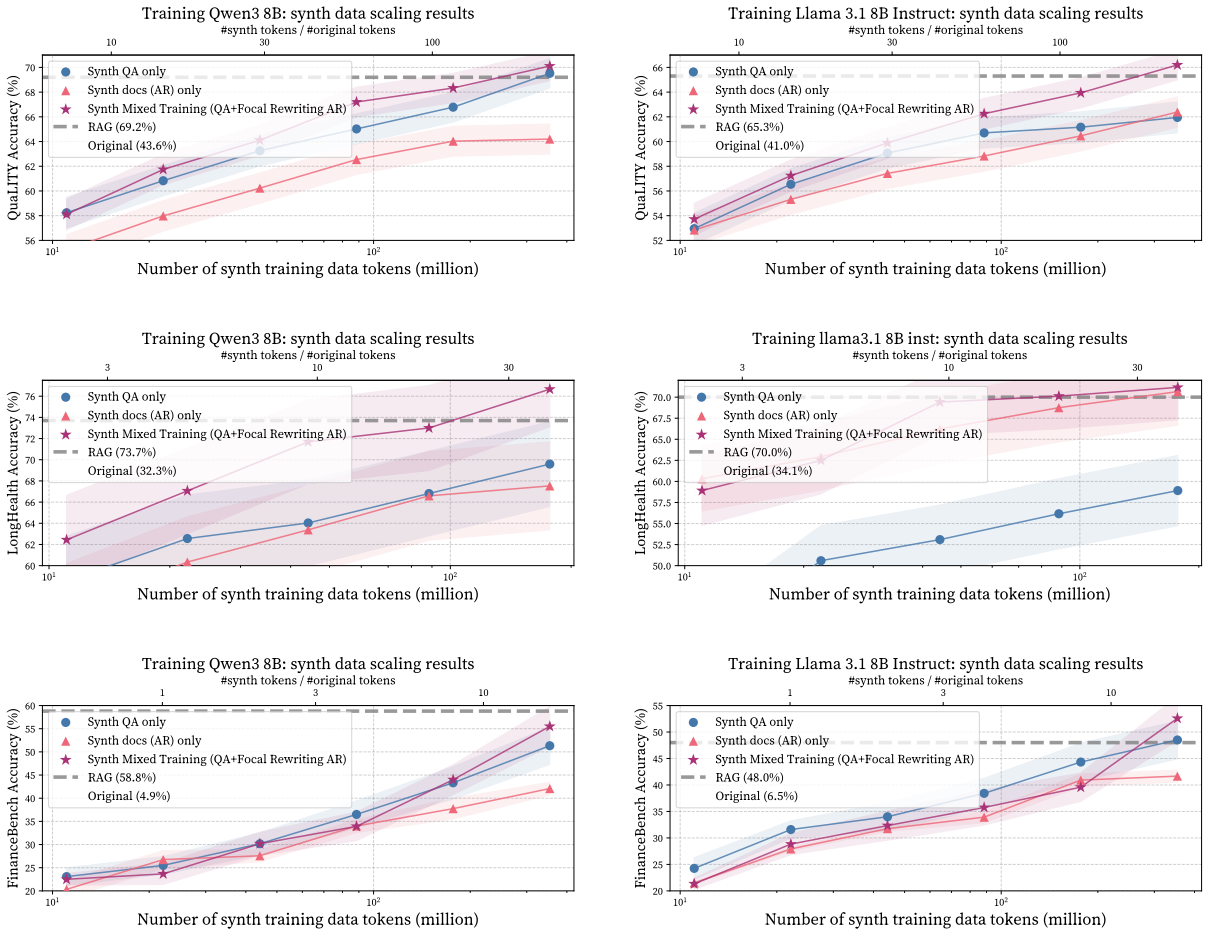

Figure 8: Training Qwen3 8B and Llama 3.1 8B Instruct using synthetic data (generated with 70B model) across three benchmarks (QuaLITY, LongHealth, FinanceBench). Using our recipe (purple line) enables beating RAG on 5/6 of the setups.

and Gao et al. (2025).

Estimating compute for synthetic training. We provide a crude estimate of the amount of compute required for synthetic training. We follow the common approximation for FLOP calculation from Kaplan et al. (2020): $2ND$ for the forward pass and $4ND$ for the backward pass, where N is the number of model parameters and D is the number of data tokens. Under this approximation, when we train a model with N parameters on D synthetic tokens generated by a model with M parameters, the total compute required for synthetic training can be written as $C \approx 2MD + 6ND$. Using this formula, we estimate the computational cost of our most expensive training run: training an 8B model on 700M tokens generated by a 70B model requires approximately 1.316×10^{20} FLOPs. Assuming using H100 (1979 TFLOPS) for synthetic training without any other overhead, it requires 18.5 H100 hours to generate synthetic data and train the model on it.

C Evaluation details

RAG implementation. We compare the trained model against a RAG (Lewis et al., 2020) that uses the model before synthetic data training. To implement RAG, we use Qwen3-Embedding-8B as the retriever to fetch the top-128 document chunks most relevant to the query. We then apply Qwen3-Reranker-8B to rerank these chunks and select the top-8 as context. Our Llama 3.1 8B RAG model achieves 65.3% performance on QuaLITY, compared with the 60.4% reported by Yang et al. (2025b), suggesting that our RAG configuration is reasonably strong.

Evaluation hyperparameters. During evaluation, we use temperature=0.1, top-p=0.95, and a maximum length of 512. We generate eight re-

```
Here's a learning strategy.
{strategy}

Apply this strategy to the document "{doc_name}" of {company}.

Output: {generated_text}
```

Table 3: Example data format for synthetic training data with metadata. This is an example of using FinanceBench matadata.

sponses per question (n=8) and report the average accuracy for a more robust evaluation. For model evaluations on the MCQA benchmarks (QuaLITY, LongHealth), we use the prompt in Table 4, and for the open-ended generation task (FinanceBench), we use the prompt in Table 5.

D Synthetic Data Generation Details

We use vLLM (Kwon et al., 2023) for efficient LLM inference during data generation. For QA pair generation, we use the prompt in Table 6. For document generation, we use the prompts in Table 7, Table 8, Table 9, Table 10, and Table 11. For QA generation, we use a temperature of 1.0, a top-p of 1.0, and a maximum length of 2048. For document generation, we use a temperature of 0.7, a top-p of 0.95, and a maximum length of 4096. We use Llama 3.1 8B Instruct for QA generation in all experiments, including the generation of questions for Focal Rewriting. For the experiments in Section 3.3, we use Llama 3.1 70B Instruct for response and document generation, except on FinanceBench, where we use Qwen3 30B A3B Instruct for both response and document generation.

E Measuring Similarity of the Synthetic Data in Gradient Space

To analyze how synthetic QA and synthetic documents are similar to each other, we compute gradient embeddings for synthetic QAs and synthetic documents (AR) generated from two datasets (QuaLITY, LongHealth (Adams et al., 2025)), and measure both intra- and inter-set gradient-embedding similarity. All synthetic datasets are generated using 70B generator.

Specifically, we follow Jung et al. (2025) when computing the gradient embeddings: we use next-token prediction loss, Qwen3 0.6B as the model, and a Johnson-Lindenstrauss transform (Johnson et al., 1984) to reduce the gradient dimensionality. We sample 16 batches with the sequence length of 2048 from each data types.

Figure 9 shows the results. QA datasets exhibit high gradient similarity across different domains, whereas AR datasets show lower gradient similarity. However, QA and AR data from the same domain exhibit the lowest gradient similarity, suggesting that data type is an important factor in shaping training signals.

F More results

F.1 Measuring diversity of generated documents

We measure diversity from two perspectives: semantic diversity and lexical diversity. For semantic diversity, we use the Vendi Score (Friedman and Dieng, 2023), which quantifies how varied the data instances are. Specifically, we compute an embedding for each instance using Qwen3-Embedding-8B and then use the cosine similarity between embeddings to construct the pairwise similarity matrix required for the Vendi Score. For lexical diversity, we report the unique 4-gram ratio, which captures the extent of surface-form variation in the text.

In Figure 10, we show how the diversity of AR documents and Focal Rewriting AR documents changes across different data sizes on QuaLITY. We plot the results this way to compare how diversity changes under different data budgets. The figure shows that Focal Rewriting yields higher lexical and semantic diversity. Interestingly, using a stronger model does not lead to higher diversity.

F.2 Different mixing ratio for synth mixed training

We experiment with different mixing ratios of synthetic QA data and documents for mixed training. Figure 11 shows the results of training Llama 3.1 8B on data generated by the 70B model. Among the tested variants, a 1:1 mixing ratio yields the best performance, and we use this mixing ratio for

```
### Question
{question}

### Choices
{options}

Choose the best answer from the following options after thinking step by step.
There is only one correct choice.
Your answer format should be like this:
Explanation: [your explanation]
Answer: [your answer (only one letter, A, B, C, D, or E)]
```

Table 4: Prompt template used for multiple-choice QA evaluation. This is used for QuaLITY and LongHealth.

```
### Question
{question}

Answer the question using the document above.
Your answer format should be like this:
Explanation: [your explanation]
Answer: [your answer]
```

Table 5: Prompt template used for open-ended QA evaluation. This is used for FinanceBench.

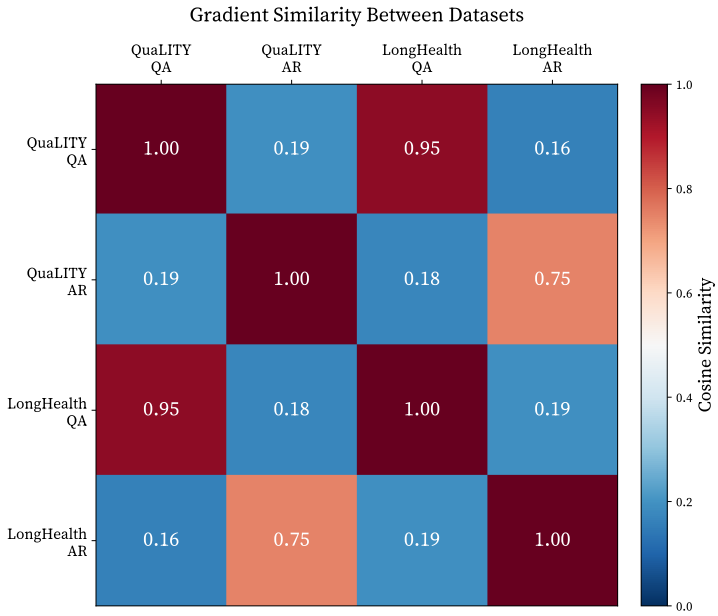

Figure 9: **Average gradient similarity between datasets.** We compute gradient embeddings for each data point and use cosine similarity between embeddings to measure the similarity of gradients across datasets. (1) QA examples from different datasets exhibit high gradient similarity (≥ 0.94), whereas QA examples and documents (AR) from different datasets show low gradient similarity (≤ 0.25). (2) Even QA and documents from the same dataset do not exhibit high gradient similarity (≤ 0.26).

the other experiments consistently. The optimal mixing ratio may differ across setups, and identifying it could further improve the effectiveness of our method.

G Additional Related Works

Parameter-efficient training for new knowledge. Parameter-efficient adaptation is a promising direction for teaching models new knowledge. LoRA (Hu et al., 2022) has been widely

```
Generate question-answer pairs from the following article.

Article:
{article}

ONLY 'Question: ...' and 'Answer: ...' tags are allowed. DO NOT include any
other text.
```

Table 6: Prompt template used for question-answer pair generation from an article.

```
Rewrite the following document to help the user understand the document better.

<document>
{document}
</document>
```

Table 7: Prompt template used for document rephrasing (Maini et al., 2024).

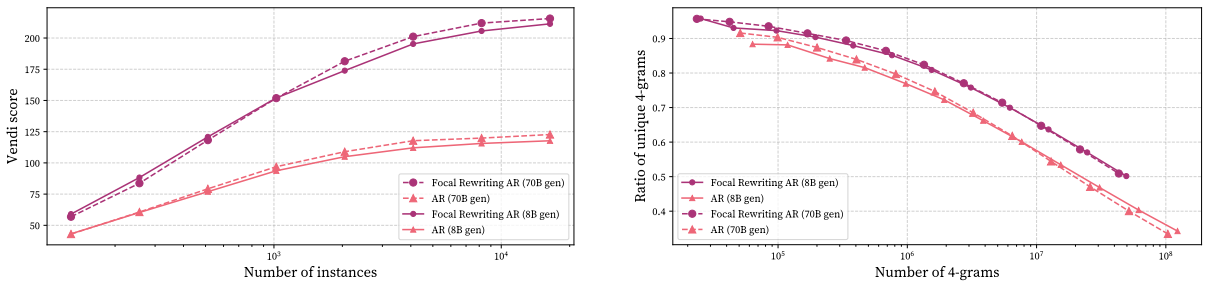

Figure 10: **How data diversity changes with different synthetic document generation methods.** (Left) Semantic diversity of synthetic documents, measured by the Vendi score (Friedman and Dieng, 2023) using embedding-based similarity to compute distances between data points. (Right) Lexical diversity of synthetic documents, measured as the ratio of unique 4-grams in the data. For both metrics, higher values indicate greater diversity. (1) Focal Rewriting increases both semantic and lexical diversity at all dataset sizes, and (2) scaling the generator does not significantly affect diversity.

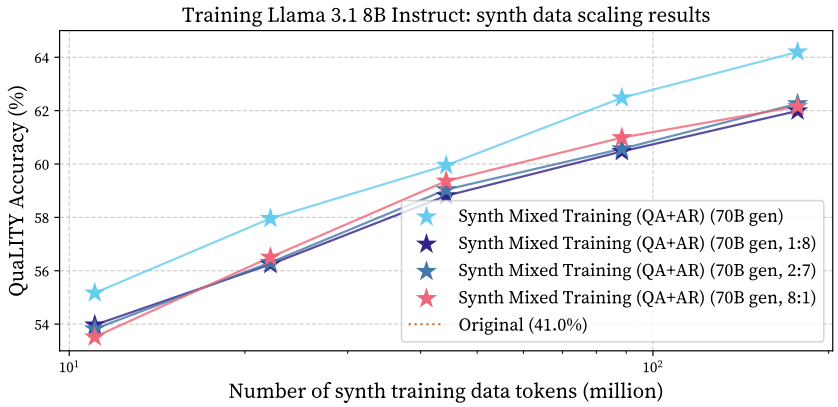

Figure 11: **Synthetic mixed training with different synthetic QA-document mixing ratios.** We test four mixing ratios of QA and AR: (1:1), (1:8), (2:7), and (8:1). The remaining 10% is used for replay with FineWeb. Using 1:1 mixing gives the best result.

used to adapt models through low-rank updates to their weights, and, combined with context distillation (Snell et al., 2022), Caccia et al. (2025) propose training LoRA layers to acquire new knowledge. However, their Llama 8B model trained on QuaLITY achieves 59.3% accuracy, which remains

```
As a knowledge analyzer, your task is to dissect and understand an article
provided by the user.
You are required to perform the following steps:

1. Summarize the Article: Provide a concise summary of the entire article,
capturing the main points and themes.
2. Extract Entities: Identify and list all significant "nouns" or entities
mentioned within the article. These entities should include but not limited to:
* People: Any individuals mentioned in the article, using the names or references
provided.
* Places: Both specific locations and abstract spaces relevant to the content.
* Object: Any concrete object that is referenced by the provided content.
* Concepts: Any significant abstract ideas or themes that are central to the
article's discussion.

Try to exhaust as many entities as possible. Your response should be structured
in a JSON format to organize the information effectively.
Ensure that the summary is brief yet comprehensive, and the list of entities is
detailed and accurate.

Here is the format you should use for your response:
{{
    "summary": "<A concise summary of the article>",
    "entities": ["entity1", "entity2", ...]
}}

Article:
{document}
```

Table 8: Prompt template used to extract entities for EntiGraph (Yang et al., 2025b).

substantially below our results. Biderman et al. (2025) also show that low-rank updates can limit the acquisition of new knowledge, highlighting a key limitation of LoRA for knowledge-intensive learning.

Motivated by the hypothesis that Transformer key-value (KV) caches function as a form of knowledge base (Geva et al., 2021), Eyuboglu et al. (2025) propose an end-to-end training approach that optimizes only the KV cache to store knowledge. More recently, Zweiger et al. (2026) introduce an optimization method that updates the KV cache to compress knowledge without requiring end-to-end training. Although we view these approaches as promising, we do not include them as baselines for two reasons. First, applying these compression-based methods to our setting is infeasible because concatenating all documents would require context lengths of 1.6M and 4.8M tokens, respectively, which are not supported by the base model we use. Second, even if the base model supported context lengths beyond 1M tokens, their performance degrades substantially at high compression ratios (i.e., when compressing by more than $20\times$), making them difficult to apply in our setting. Consistent with these limitations, Zweiger et al. (2026) evaluate on QuaLITY by compressing only a single document, while on LongHealth, Zweiger et al. (2026) and Eyuboglu et al. (2025) compress only five and ten documents, respectively. In contrast, we train models on all documents in each dataset: 265 documents from QuaLITY and 400 documents from LongHealth.

Alleviating forgetting. Continued training often leads to the forgetting of existing knowledge in language models. A common technique for mitigating this issue is replay–reusing pretraining data during the continued training stage (Lin et al., 2025a; Yang et al., 2025b; Kotha and Liang, 2026; Liu et al., 2025). In a symbolic distillation setup, Agarwal et al. (2024); Lu and Lab (2025) suggest using on-policy distillation: compared to training the model with the synthetic data generated by another model (e.g., stronger model), using the self-generated data

```
You will act as a knowledge analyzer tasked with dissecting an article provided
by the user. Your role involves two main objectives:
1. Rephrasing Content: The user will identify two specific entities mentioned in
the article. You are required to rephrase the content of the article twice:
    * Once, emphasizing the first entity.
    * Again, emphasizing the second entity.
2. Analyzing Interactions: Discuss how the two specified entities interact within
the context of the article.

Your responses should provide clear segregation between the rephrased content and
the interaction analysis. Ensure each section of the output include sufficient
context, ideally referencing the article's title to maintain clarity about the
discussion's focus.

Here is the format you should follow for your response:

### Discussion of <title> in relation to <entity1>
<Rephrased content focusing on the first entity>

### Discussion of <title> in relation to <entity2>
<Rephrased content focusing on the second entity>

### Discussion of Interaction between <entity1> and <entity2> in context of
<title>
<Discussion on how the two entities interact within the article>

### Document
{document}

### Entities:
- {entity1}
- {entity2}
```

Table 9: Prompt template used for entity linking for generating EntiGraph documents (Yang et al., 2025b).

for the points to compute the loss leads to less forgetting while learning new knowledge well.

There have also been attempts to address forgetting through improved language model architectures. For example, Lin et al. (2025b) suggest using memory layers (He, 2024; Berges et al., 2024), which are identical to Mixture-of-Experts (Shazeer et al., 2017) models but use a large number of experts in a specific layer. These layers are updated specifically for new knowledge, reducing interference with existing knowledge. The paper shows that there is a trade-off between learning new knowledge and forgetting existing knowledge, and that sparse model updates with memory layers provide a better Pareto frontier than full fine-tuning or LoRA (Biderman et al., 2025; Hu et al., 2022).

```
Consider the following document. What are some strategies specific to this
document that I can use to help me learn and remember all of the information
contained? Use markdown and prefix each strategy with ##.

<document>
{document}
</document>
```

Table 10: Prompt template used for generating active reading strategies (Lin et al., 2025a).

```
Here's a learning strategy.
{strategy}

Apply this strategy to the following document:

<document>
{document}
</document>
```

Table 11: Prompt template used for active reading document generation with a provided learning strategy (Lin et al., 2025a).

```
<document>
{document}
</document>

Here's a learning strategy.
{strategy}

Apply this strategy to the document above, with the focus on the question: {query}
```

Table 12: Prompt template used for Focal Rewriting active reading with a provided learning strategy.