IKnow: Instruction-Knowledge-Aware Continual Pretraining for Effective Domain Adaptation

Anonymous ACL submission

Abstract

Continual pretraining promises to adapt large language models (LLMs) to new domains using only unlabeled test-time data, but naively applying standard self-supervised objectives to instruction-tuned models is known to degrade their instruction-following capability and semantic representations. Existing fixes assume access to the original *base* model or rely on knowledge from an *external* domain-specific database - both of which pose a realistic barrier in settings where the base model weights are withheld for safety reasons or reliable external corpora are unavailable. In this work, we propose **Instruction-Knowledge-Aware Continual Adaptation (IKnow)**, a simple and general framework that *formulates novel self-supervised objectives in the instruction-response dialogue format*. Rather than depending on external resources, IKnow leverages domain knowledge embedded within the text itself and learns to encode it at a deeper semantic level.

1 Introduction

Large language models (LLMs) that have undergone *scaled pretraining* and further *instruction tuning* (Wei et al., 2022) now underpin most modern NLP applications. Although these models achieve strong zero-shot performance, their accuracy degrades when the deployment domain drifts from the public web data on which they were pre-trained, i.e., when asked questions about world events that happened after the curation of the pretraining data. A classical remedy is *continued pre-training* on unlabelled in-domain text, first popularized by Gururangan et al. (2020) for BERT-style (Devlin et al., 2019) models. However, continued prerain models naively with a next-token-prediction objective can encounter undesired outcomes in instruction-tuned models, such as catastrophic forggetting (Jindal et al., 2024).

Firstly, the widely used instruction-tuned models can lose their instruction-following ability thereafter. Fleshman and Van Durme (2024) mitigate this effect by performing continual finetuning on the *base* model and adding the instruction-following ability back in by adding an instruction-task vector to the fine-tuned model weights. However, oftentimes LLM developers do not publish the base model to avoid safety risks that may emerge from models without post-training, e.g. Phi-4 (Abdin et al., 2024), making this approach unviable.

Secondly, knowledge-intensive downstream tasks, such as question answering require a deeper encoding of semantics rather than shallow sequence probabilities. To facilitate knowledge encoding, Zhang et al. (2019); Peters et al. (2019); Wang et al. (2021b) inject the knowledge embeddings acquired from external dataset into token embeddings in pre-training. However, this approach is unviable in continual pretraining phase due to the acquisition of compatible knowledge embeddings and modifications in the model architectures.

To address domain drift declination in an off-the-shelf instruction-tuned model scenario, we introduce **Instruction-Knowledge-Aware Continual Adaptation** (IKNOW), a light-weight continual pretraining framework that perform knowledge-aware self-supervised objectives in the format of an instruction–response dialogue to encourage the model to keep its instruction following ability. completion. Additionally, we propose two knowledge-aware pretraining tasks to enhance a model's ability to capture entity-relation knowledge by constructing relations from parse trees. Our experimental evaluation on two question answering tasks demonstrates the effectiveness of our approach.

2 Related Work

Continual pretraining. Gururangan et al. (2020) first showed that running extra unsupervised epochs

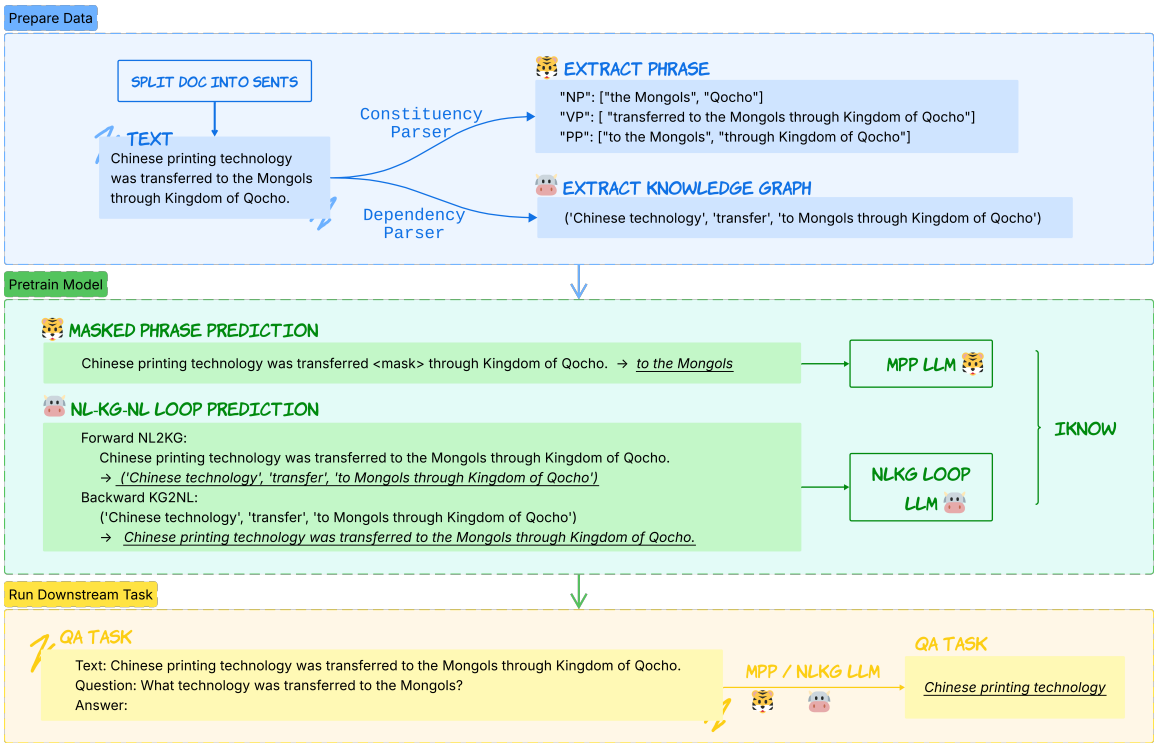

Figure 1: Overview of our framework, consisting of three stages: **(I)** data preparation, **(II)** model pretraining, and **(III)** downstream evaluation. **(I)** In the data preparation stage, we leverage off-the-shelf syntactic parsers to extract structural information from text. Specifically, we employ two complementary strategies: (1) extracting constituency to identify phrases, and (2) deriving knowledge graphs based on dependency structures. **(II)** In the pretraining stage, we introduce two distinct objectives: **Masked Phrase Prediction (MPP)**, which trains the model to reconstruct syntactically meaningful spans based on constituency structure, and the **NL-KG Loop** task, which encourages bidirectional reasoning between natural language and structured knowledge graphs with a forward pass NL2KG and a backward pass KG2NL. **(III)** Finally, we evaluate the resulting models—**MPP-LLM** and **NLKG-Loop-LLM**—on knowledge-intensive downstream tasks such as question answering, demonstrating the effectiveness of structure and semantic informed pretraining. All our tasks are knowledge-aware and formatted in the *instruction-response* template. The response is in *underlined italic format* following an arrow.

on task-specific corpora ("domain–adaptive" or "task–adaptive" pre-training) boosts downstream accuracy. Subsequent studies asked whether the same recipe helps *instruction-tuned* checkpoints: Jindal et al. (2024) report severe alignment loss, while Fleshman and Van Durme (2024) recover alignment by re-injecting the base–to–instruction weight delta—an option unavailable when the base model is unreleased. Our work stays in the continual-pretraining paradigm but replaces the vanilla MLM loss with an *instruction-style* variant that keeps the dialogue context intact.

Test-time training and adaptation. Test-Time Training (TTT) updates model parameters on *each individual* test example, usually with a self-supervised loss (Sun et al., 2020); Test-Time Adaptation extends this idea to streams of test batches (Wang et al., 2021a). Early NLP instances include T-SAS for QA (Jeong et al., 2023) and few-shot experiments by Akyürek et al. (2024). Although we pretrain on the entire test corpus *before* questions arrive—hence diverging from the per-sample TTT setting—the two lines of work are complementary, and we plan to explore a true TTT variant of our objective in future work.

Knowledge-intensive pretraining. Embedding strategies that explicitly target *knowledge* tokens yield larger downstream gains than vanilla MLM. Zhang et al. (2019) concatenate the entity embeddings with token embeddings. Peters et al. (2019) inject the external database, e.g. ConceptNet, knowledge embeddings into the original token embeddings. Wang et al. (2021b) combines entity de1scription embeddings with token embeddings. Another line of work encoding semantics through masked prediction on meaningful tokens. Joshi et al. (2020) first demonstrated that masking entire *spans* boosts performance on question answering

tasks. Building on that idea, Golchin et al. (2023) show that during continued pretraining it is more efficient to mask in-domain keywords, while Kohli et al. (2025) introduce a curriculum that gradually shifts the mask toward domain-specific concepts, cutting compute during biomedical adaptation by an order of magnitude. Our work aligns with the previous research on train the model with semantic knowledge, but we focus on the knowledge embedded in the text itself, not requiring any external dataset.

3 Method

3.1 Problem Setup

We assume a standard extractive QA benchmark, whose test portion is a collection of triples $\mathcal{T} = \{(C_i,\, Q_i,\, A_i)\}_{i=1}^{N}$, where each *context* C_i is a supportive context document, Q_i is a natural–language question with answer A_i. In many real deployments the *passages become available long before the questions are asked*. A concrete example is an enterprise assistant that must answer employee queries about a newly published internal manual: the manual (our C_i) can be processed offline, whereas the questions Q_i only arrive during usage. Hence, we assume access to the contexts of the test set.

Formally, at time $t{=}0$ we receive the unlabeled *context set* $\mathcal{C}_{\text{test}} = \{C_i\}_{i=1}^{N}$. We are given an LLM with parameters $\boldsymbol{\theta}_0$, but we *do not* have access to the underlying base checkpoint. Our goal is to adapt $\boldsymbol{\theta}_0$ using only $\mathcal{C}_{\text{test}}$, then answer all (Q_i) at $t{>}0$. Crucially, our LLM with parameter θ_0 is *instruction-tuned*, posing the challenge how to retain its instruction-following ability during continual pretraining.

3.2 Instruction-Knowledge-Aware Continual Adaptation (IKnow)

Instruction-tuned methods are trained on input data that usually have the following format:

`|user| <user_query>`
`|assistant| <response>`,

where `<user_query>` is the instruction to be followed, and `<response>` is the reply from the assistant. Note that the LLM is only trained on the response.

In order to retain the instruction-following ability of instruction-tuned LLMs during continual pretraining, our main idea is to transform the context set $\mathcal{C}_{\text{test}}$ into $(user_query, response)$ pairs. Figure 1 summarizes our approach.

Prepare data In order to make best use of the often limited data we have available, we first split the context c into sentences $S_c = \{s|s \text{ is sentence in } c\}$. Then we use a constituency parser to identify phrases $P_s = \{p_i|p_i \in const(s)\}$, and a dependency parser to identify the set of $(subject, root, object)$ relations in s, $KG_s = \{(s, r, v) \in dep(s)\}$.

After parsing the data, we generate instruction-tuning training examples in three different ways: Masked Token Prediction (MTP), Masked Phrase Prediction (MPP), and NL↔KG.

Masked Token Prediction Given a (tokenized) sentence $s = s_1 \dots s_n$, we randomly choose an index $1 \le i \le n$ to mask out. We then set

`<user_query>` = `Complete the masked token:` $s_1 \dots s_{i-1}$ `<mask>` $s_{i+1} \dots s_n$
`<response>` = s_i

MTP can be considered a straigth up adaptation of standard masked language modeling (Devlin et al., 2019) to the instruction-tuning case.

Masked Phrase Prediction Adaptation to new domains requires an understanding of the relevant entities and their relations in that domain, especially for knowledge intensive tasks. While MLM is a good option for teaching general language understanding ability, it chooses masked tokens randomly rather than focusing on entities as humans. To address this, we propose to mask out an entire phrase, e.g. noun phrase, verb phrase, or prepositional phrase, which correspond more to entities (NPs and PPs) or relations (VPs).

Analogous to before, we randomly select a phrase $p_j = s_k \dots s_{k+l}$ of length l that will be masked out:

`<user_query>` = `Complete the masked words:` $s_1 \dots s_{i-1}$ `<mask>` $s_{i+1} \dots s_n$
`<response>` = p_j

NL↔KG To emulate the human learning process—where input is encoded into structured knowledge and output is decoded from it (Atkinson and Shiffrin, 1968)—we propose a framework that explicitly constructs a knowledge graph from a natural language sentence (s), referred to as NL→KG. We formulate the task as follows:

```
<user_query> = Please extract
knowledge tuples (subject, verb,
object) from the text: s
<response> = KG_s
```

The KG→NL task asks for the reverse.

4 Experiments

4.1 Experimental Setup

Hypotheses and baselines Our experiments are designed to test two hypotheses: (**H1**) Instruction-style pretraining tasks improves over the naive next-token prediction (**NTP**) baseline. (**H2**) Pretraining tasks tailored toward knowledge-intensive tasks, namely Masked Phrase Prediction (**MPP**) and NLKG, improve performance compared to naive Masked Token Prediction (**MTP**).

Datasets To demonstrate the effectiveness of our approach on two recently published, knowledge-intensive question answering datasets. RepliQA (Monteiro et al., 2024) includes human-curated news articles paired with automatically generated question-answer pairs. SciQAG (Wan et al., 2024) focuses on scientific publications with corresponding generated QA pairs. Detailed statistics for each dataset are provided in Appendix A. We adopt different evaluation metrics as provided by Hugging Face Evaluation package based on the characteristics of the datasets. On RepliQA and SciQAG, which contain long-form answers, we report ROUGE-L-F1.

Hyperparameters To demonstrate the generalizability of our results, we perform our experiments with two models (Llama-3.2-3B-Instruct and Qwen3-1.7B) and two fine-tuning techniques, full-finetuning and LoRA (Hu et al., 2022). Learning rates are tuned per model and fine-tuning mode. The full description of the experimental setup can be found in Appendix B.

4.2 Results

Results are presented in Table 1. Across two tasks, two models, and two fine-tuning modes, instruction-style pretraining tasks outperform the naive NTP baseline 19 out of 24 times. The knowledge-intensive pretraining tasks outperform the MTP baseline in 7 out of 16 cases.

Model	RepliQA	SciQAG
Llama-3.2-3B-Instruct	34.71	39.52
Full Parameter		
NTP	0.2*	39.41
MTP	35.47	39.66
MPP	*35.58*	***40.11***
NLKG	***36.83***	*39.88*
LoRA		
NTP	37.19	39.89
MTP	28.13	39.24
MPP	***38.62***	39.12
NLKG	0.7*	*39.26*
Qwen3-1.7B	28.73	37.54
Full Parameter		
NTP	28.8	37.10
MTP	**31.71**	**38.55**
MPP	30.42	37.94
NLKG	29.4	38.30
LoRA		
NTP	29.31	37.78
MTP	**31.86**	38.56
MPP	31.39	***38.96***
NLKG	29.53	38.55

Table 1: Results on RepliQA and SciQAG. Underline indicates improvement of MTP/MPP/NLKG over NTP (**H1**). *Italics* indicates improvement of MPP/NLKG over MTP (**H2**). **Bold** indicates the overall best result per model and finetuning method. *: catast. forgetting.

5 Discussion

Our results demonstrate support for (**H1**): We can successfully improve the performance of instruction-tuned LLMs on a new domain through continual pretraining, a task where past approaches have failed without acccess to the base model.

The support for **(H2)** is mixed: Our knowledge-tailored pretraining tasks yield substantial performance gains—often several points—for Llama3.2-3B, but we observe no consistent improvement for Qwen3-1.7B. This discrepancy is difficult to pinpoint precisely; possible factors include Qwen3's stronger emphasis on reasoning and its smaller parameter count, which may limit its capacity to benefit from more sophisticated pretraining objectives targeted at knowledge acquisition.

6 Conclusion

Addressing the issue of diminishing instruction-following ability during continual pretraining, we introduced a recipe that formulates self-supervised losses in an instruction-response template. Our experiments indicate promising accuracy gains with these objectives for question answering tasks. Future work will explore combining multiple objectives.

Limitations

While our proposed framework demonstrates promising results, several limitations remain. First, our current evaluation setup involves training and testing on the full test dataset. Future work will explore model performance when trained on smaller subsets or single-sample scenarios that are typical in test-time training (Sun et al., 2020).

Second, our scope is limited to question answering tasks. We believe that it is likely that other tasks will also benefit from instruction-style continual pretraining, however, they may require a different set of novel pretraining tasks that are not tailored toward knowledge acquisition. Therefore, we leave a more extensive evaluation to future work.

Third, due to compute constraints we conducted our experiments on relatively small language models of up to 3 billion parameters. It is an open question to what extent our results generalize to larger-scale models.

Ethical Considerations

Our experiments focus on a high-resource language, specifically English. While these languages benefit from extensive NLP research and abundant annotated resources, our current evaluation does not extend to low-resource languages such as Faroese or Norwegian. This limitation highlights a potential bias in language coverage. We acknowledge that applying our IKnow approach to low-resource settings could have meaningful implications for linguistic inclusivity. Future work should prioritize expanding the evaluation to such languages to ensure broader applicability and equitable access to language technologies.

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*.

Elif Akyürek, Zhouhang Lin, Hengyuan Zhao, and Jacob Andreas. 2024. The surprising effectiveness of test-time training for few-shot learning. *arXiv preprint arXiv:2411.00000*.

R.C. Atkinson and R.M. Shiffrin. 1968. Human memory: A proposed system and its control processes. In Kenneth W. Spence and Janet Taylor Spence, editors, *The Psychology of Learning and Motivation: Advances in Research and Theory*, volume 2 of *Psychology of Learning and Motivation*, pages 89–195. Academic Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

William Fleshman and Benjamin Van Durme. 2024. Re-adapt: Recovering instruction-following ability after domain adaptation. *arXiv preprint arXiv:2409.00000*.

Shahriar Golchin, Mihai Surdeanu, Nazgol Tavabi, and Ata Kiapour. 2023. Do not mask randomly: Effective domain-adaptive pre-training by masking in-domain keywords. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 13–21.

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, Online. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, 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. *ICLR*, 1(2):3.

Mihee Jeong, Yongho Lee, Jimin Park, Soohyun Lim, and Minjoon Seo. 2023. Test-time self-adaptive small language models for question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Guneet Jindal, Yao Chen, Guangya Zhou, and Sebastian Gehrmann. 2024. Balancing continuous pretraining and instruction fine-tuning. *arXiv preprint arXiv:2410.00000*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Vanshpreet S. Kohli, Aaron Monis, and Radhika Mamidi. 2025. Choose your words wisely: Domain-adaptive masking makes language models learn faster. In *Proceedings of the 10th Workshop on Representation Learning for NLP (RepL4NLP 2025)*, pages 87–91.

Joao Monteiro, Pierre-Andre Noel, Etienne Marcotte, Sai Rajeswar Mudumba, Valentina Zantedeschi, David Vazquez, Nicolas Chapados, Chris Pal, and

Perouz Taslakian. 2024. Repliqa: A question-answering dataset for benchmarking llms on unseen reference content. *Advances in Neural Information Processing Systems*, 37:24242–24276.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. 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 43–54, Hong Kong, China. Association for Computational Linguistics.

Qianru Sun, Yaoyao Li, Hao Wang, and Luc Van Gool. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning*.

Yuwei Wan, Yixuan Liu, Aswathy Ajith, Clara Grazian, Bram Hoex, Wenjie Zhang, Chunyu Kit, Tong Xie, and Ian Foster. 2024. Sciqag: A framework for auto-generated science question answering dataset with fine-grained evaluation. *arXiv preprint arXiv:2405.09939*.

Dequan Wang, Yixin Wei, Jeff Bilmes, and Ali Farhadi. 2021a. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021b. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

A Dataset statistics

The statistics of dataset is shown in table 2.

	split	# doc	# sentence	# QA pairs
RepliQA	repliqa_0	3k	160k	18k
SciQAG	select_50	50	7k	500

Table 2: Statistics of dataset

B Experimental Setup

We provide the following implementation details to enable better reproducibility and to support the transparency and rigor of our experiments.

Parsing Tools

- **Dependency Parsing**: We use the spaCy library with the `en_core_web_trf` pretrained transformer-based parsing models.
- **Constituency Parsing**: We integrate benepar with spaCy for constituency parsing, using the pretrained `benepar_en3` models.

Experimental Framework

- **Frameworks**:
  - PyTorch Lightning is used for modular and scalable training routines.
  - Hugging Face Transformers is used for loading and fine-tuning pretrained language models.

Training Configuration

- **Batch Size**: We use the maximum batch size that can fit into a single GPU.
- **Learning Rate**: We perform experiments on Llama3 with learning rates ranging from 1e-6 to 1e-8 and on Qwen3 with learning rate ranging from 1e-7 to 1e-9 in full fine-tuning and LoRA settings. We search over the range $[5 \times 10^{-5}, 1 \times 10^{-9}]$ and select the learning rate casewise with decreasing loss (usually the average loss is from 10 to 1) to balance learning effectiveness and retention of prior knowledge.
- **Epochs**: We train 10 epochs and choose the **inflection point epoch** (usually 3-7 epochs depending on the dataset and pretraining tasks) for better stability and generalization.
- **Run Count**: Each experiment (training and testing) is conducted **once**, in accordance with prior work under limited resource conditions.
- **LoRA**: Each task is experimented with the LoRA configuration, see in table 3

Our final hyper-parameter setting is detailed in table 4.

rank	8
alpha	16
dropout	0.05
bias	none
modules	q,v, k, o gate, up, down

Table 3: LoRA Configuration

C Use of Generative AI

We used ChatGPT and Claude models to assist in writing small functions of our implementations. We also used them to assist in writing by fixing grammar, typos, and general style of our writing. However, we didn't use generative AI for generating ideas or other high-level aspects of our research.

model name	Llama-3.2-3B-Instruct	Qwen3-1.7B
num_epochs	10	10
batch_size	200	200
weight_decay	0	0
warmup_steps	0	0
precision	bf16	bf16
strategy	ddp	ddp
optimizer	AdamW	AdamW
learning_rate	1e-6 - 1e-8	1e-7 - 1e-9
scheduler	constant	constant
device	8*NVIDIA A100-SXM4-80GB	8*NVIDIA A100-SXM4-80GB

Table 4: Hyper-parmeters for continual pretraining.