Effective Domain Adaptation of Instruction-Tuned LLMs for Knowledge-Intensive Tasks

Tianyi Zhang* **and Florian Mai*** **and Lucie Flek**
University of Bonn
tianyiz0423@gmail.com
fmai@uni-bonn.de
flek@bit.uni-bonn.de

Abstract

Continual pretraining promises to adapt large language models (LLMs) to new test domains using only unlabeled data, but naively applying common self-supervised objectives is known to degrade instruction-following performance. Existing fixes assume access to the original *base* model - a realistic barrier in settings where the base model weights are withheld for safety reasons. In this work, we propose **Instruction-Style Continual Adaptation (ISCA)**, a simple procedure that reformulates self-supervised objectives in the *format of an instruction-response dialogue*. Apart from a straight-forward adaptation of the masked-language modeling objective, we devise multiple self-supervised approaches that specifically designed for knowledge-intensive downstream tasks. Particularly, in the Masked Phrase Prediction (MPP), we mask out meaningful phrases. In the NL-KG Loop Prediction (NL-KG Loop), the model is trained to perform a bidirectional transformation between natural language and knowledge tuple representations. Benchmarking our ISCA objectives on knowledge-intensive downstream tasks, the results demonstrate the feasibility of domain-specific adaptation that preserves instruction-following ability without the need for access to a potentially dangerous base model.

1 Introduction

Large language models (LLMs) that have undergone *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, e.g. 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. The widely used instruction-tuned models, however, can lose their instruction-following ability when continually pretrained naively with a masked-language-model (MLM) objective (Jindal et al., 2024). 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.

To address this scenario, we introduce **Instruction-Style Continual Adaptation** (ISCA), a procedure that rewrites self-supervised objectives in the format of an instruction–response dialogue to encourage the model to keep its instruction following ability. Aiming at knowledge-intensive downstream tasks in particular, we propose two variants of this objective that are inspired by insights from human information processing. In the Masked Phrase Prediction task, the model is trained to predict phrases of entities, as identified by constituency parsing, rather than randomly selected tokens. In the NL-KG Loop Prediction task, the model is guided to perform bidirectional translation between natural language and knowledge graphs extracted from dependency parsing, thereby encouraging a deeper integration of entity-relation structure into the model representations. Our experimental evaluation on four knowledge-intensive downstream tasks demonstrates the effectiveness of our approach.

2 Related Work

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

*Equal contribution.

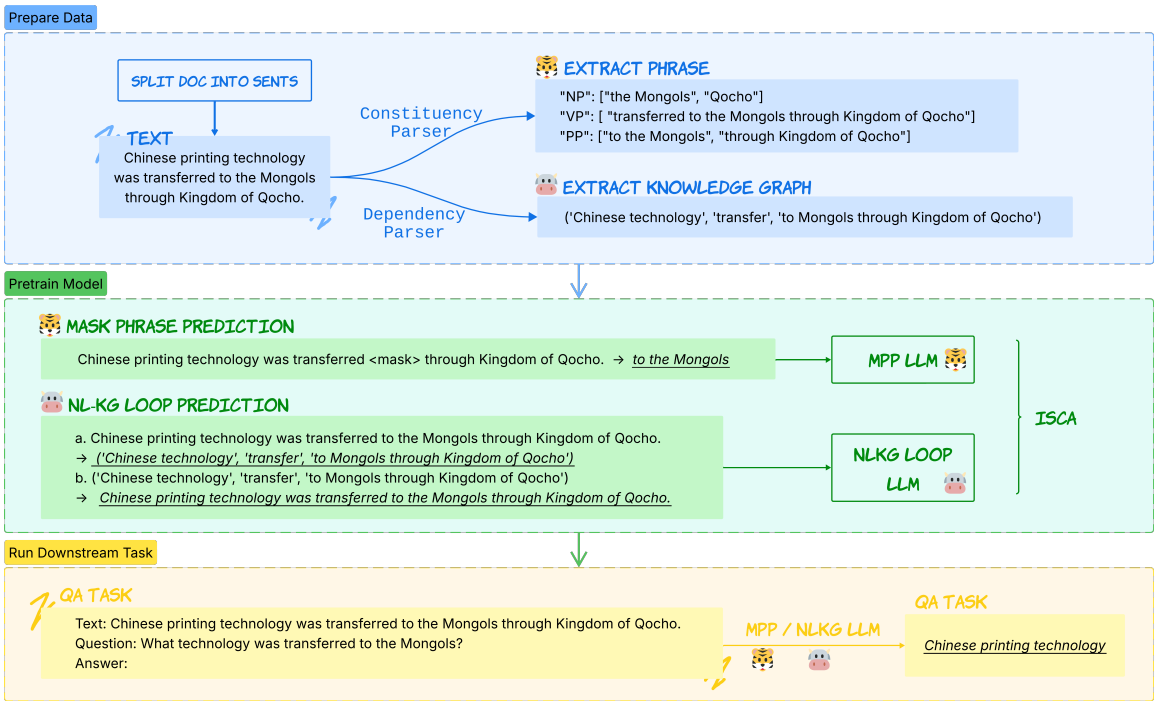

Figure 1: Overview of our framework, consisting of three stages: **(I)** data preparation, **(II)** model pretraining, and **(III)** downstream evaluation. **(I)** In the data preparation stage, we leverage off-the-shelf syntactic parsers to extract structural information from text. Specifically, we 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 (a) and a backward pass (b). **(III)** Finally, we evaluate the resulting models—**MPP-LLM** and **NLKG-Loop-LLM**—on knowledge-intensive downstream tasks such as question answering and abstractive summarization, demonstrating the effectiveness of structure and semantic informed pretraining. All our tasks is 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., 2021). 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. Masking strategies that explicitly target *knowledge* tokens yield larger downstream QA gains than vanilla MLM. 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. These studies motivate us to explore knowledge-intensive pretraining tasks in ISCA.

3 Method

3.1 Problem Setup

We assume a standard extractive QA benchmark such as SQuAD, 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 freshly 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-Style Continual Adaptation (ISCA)

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

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

where <system_prompt> is a system-wide instruction that the model should always follow, e.g. "You are a helpful assistant", 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 Loop transformation.

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

<user_query> = Please predict the missing token in the following sentence: $s_1 \dots s_{i-1}$<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> = Please predict the missing phrase in the following sentence: $s_1 \dots s_{k-1}$<mask>$s_{k+l+1} \dots s_n$
<response> = $s_k \dots s_{k+l}$

NL-KG Loop To emulate the human learning process—where input is encoded into structured knowledge and output is decoded from it—we propose a framework that explicitly constructs a knowledge graph from a natural language sentence (s), referred to as NL2KG, and reverses the process to generate text from the knowledge graph, referred to as KG2NL. Formally, we introduce two continuous tasks. The first task involves extracting a set of knowledge graph tuples from a natural language sentence.

```
<user_query> = Please extract
knowledge graph tuples from the
```

following sentence: s
<response> = KG_s

The second task asks for the reverse.

4 Experiments

4.1 Experimental Setup

Datasets To demonstrate the effectiveness of our approach on knowledge-intensive tasks, we evaluate it on question answering and summarization tasks. We conduct experiments on a diverse set of recently published datasets. HaluEval (Li et al., 2023) consists of documents and summaries collected from multiple sources, with hallucinated summaries generated by manually crafted model outputs. HaluSum and HaluNLI are modifications of this dataset by Anonymous (2025). Given the original dataset containing triplets of *(document, reference summary, hallucinated summary)*, we construct two tasks: For the HaluSum task, we use the *(document, reference summary)* pairs to assess the summarization. For the HaluNLI task, we frame *document* as the premise and *reference summary or hallucinated summary* as the hypothesis to formulate an entailment vs. contradiction classification problem. RepLiQA (Monteiro et al., 2024) includes human-curated news articles paired with automatically generated question-answer pairs. SQuAD (Rajpurkar et al., 2016) is included as a comparison with these recent datasets. 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 HaluSum, and RepliQA, which contain long-form answers, we report ROUGE-L-F1. On SQuAD, where the gold answers are typically short, we use exact match. On HaluNLI we use accuracy.

Hyperparameters We perform all experiments with Llama-3.2-3B-Instruct in a full fine-tuning manner. We perform a single run for each experiment. The full description of the experimental setup can be found in Appendix B.

4.2 Results

Results are presented in Table 1. Our method improves over the baseline on all four benchmarks. While the improvement is very small for SQuAD and HaluSum, it is substantial for HaluNLI and RepliQA, improving by 3.7 and 2.1 points, respectively. While MTP already yields some improvement, MPP and NLKG yield larger improvements, with no clear winner among them.

	SQuAD	HaluSum	HaluNLI	RepliQA
Baseline				
Llama-3.2-3B-Instruct	76.52	22.98	46.9	34.71
ISCA (ours)				
MTP	40.44	22.43	49.34	35.47
MPP	**76.56**	23.02	**50.62**	35.58
NLKG	74.66	**23.11**	50.40	**36.83**

Table 1: Our main results on various knowldege intensive downstream tasks such as QA and summarization.

5 Discussion

Our results demonstrate that our approach is working: 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. While our improvement on HaluNLI and RepliQA is substantial, the improvement on SQuAD is negligibly small. We explain this discrepancy with the fact that SQuAD is an old dataset from 2016. It is therefore quite likely that the contexts of SQuAD, which were synthesized from Wikipedia, have already been included in the pretraining of LLaMA-3, rendering continual pretraining on this dataset useless.

The fact that our proposed continual pretraining approaches, MPP and NLKG, perform substantially better than MTP demonstrates the value of carefully designing the pretraining objective can benefit knowledge-intensive tasks. This confirms the finding from previous studies (Joshi et al., 2020; Golchin et al., 2023; Kohli et al., 2025) and extends it to instruction-tuned models, which is enabled by our approach to frame tasks in an instruction style.

6 Conclusion

Addressing the issue of deminishing instruction-following ability during continual pretraining, we introduced a recipe that formulates self-supervised losses inspired by the human learning process in the instruction-response template. We propose two novel training objectives, MPP and NL-KG Loop Prediction, that encourage knowledge acquisition. Our experiments indicate promising accuracy gains with these objectives on question answering and summarization 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 dataset. Future work will explore model performance when trained on smaller subsets or single-sample scenarios that are typical in test time training.

Second, our experiments are conducted using a single model configuration—LLaMA-3.2-3B-Instruct, a 3-billion parameter decoder-only language model—which we selected to balance training cost and performance. However, this limited scope may not fully capture the scalability or adaptability of our methods. We plan to conduct a more extensive evaluation in future work, including both smaller models and larger models with parameter-efficient tuning strategies such as LoRA (Low-Rank Adaptation), on models like DeepSeek and other recent LLMs.

Ethical Considerations

Our experiments primarily focus on high-resource languages, specifically English, with supplementary attention to French and German. 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 ISCA 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*.

Anonymous. 2025. Anonymous. Under review.

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*.

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.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. *arXiv preprint arXiv:2305.11747*.

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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*.

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

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*.

A Dataset statistics

The statistics of dataset is shown in table 2.

	split	# doc	# sentence	# QA pairs
SQuAD	validation	2k	10k	10k
HaluEval	summary	10k	340k	-
RepliQA	repliqa_0	3k	160k	18k
SciQAG	select_50	50	7k	500
MultiOCR-QA	en	6k	61k	10k
MultiOCR-QA	fr	1k	14k	10k
MultiOCR-QA	de	7k	88k	39k

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 following pretrained models:
  - English: `en_core_web_trf`
  - French: `fr_core_news_lg`
  - German: `de_core_news_lg`
- **Constituency Parsing**: We integrate benepar with spaCy for constituency parsing, using the following models:
  - English: `benepar_en3`
  - French: `benepar_fr2`
  - German: `benepar_de2`

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.

model name	Llama-3.2-3B-Instruct
num_epochs	1
batch_size	4-10
accumulate_grad_batch	5
weight_decay	0
warmup_steps	0
precision	bf16
strategy	ddp
optimizer	AdamW
learning_rate	1e-6
scheduler	constant
device	1*NVIDIA A100-SXM4-80GB

Table 3: Hyper-parmeters for continual pretraining.

Training Configuration

- **Batch Size**: We use the maximum batch size that can fit into a single GPU.
- **Learning Rate**: We search over the range $[5 \times 10^{-5}, 1 \times 10^{-7}]$ and select 1×10^{-6} to balance learning effectiveness and retention of prior knowledge.
- **Epochs**: We compare training with 1 and 3 epochs and choose **1 epoch** for better stability and generalization.
- **Run Count**: Each experiment (training and testing) is conducted **once**, in accordance with prior work under limited resource conditions.

Our final hyper-parameter setting is detailed in table 3.

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.