Improving Domain-Specific Text Understanding with Large Language Models via Hybrid Fine-Tuning Strategies

Peng Xu[1], TingMeng Li[2], and Jintao Liang[3]

[1]School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, China
[2]Department of Aerospace Engineering, University of Illinois Urbana-Champaign, Urbana, IL, USA
[3]Department of Electrical Engineering, University of Washington, Seattle, WA, USA

January 29, 2026

Abstract

Large language models (LLMs) often show strong general capabilities but can underperform on domain-specific text understanding when terminology, style, and label definitions differ from general web text. Fine-tuning is a natural remedy, yet a single strategy rarely satisfies all practical constraints: full fine-tuning is expensive and brittle, lightweight parameter-efficient tuning may underfit, and retrieval-only methods depend heavily on index coverage.

This paper presents a hybrid fine-tuning framework for domain-specific text understanding that combines (i) continued pretraining on domain corpora, (ii) parameter-efficient instruction tuning, and (iii) task-specific calibration and evaluation. We describe a training recipe that is modular, reproducible, and designed for realistic constraints such as limited labeled data and strict compute budgets.

We provide ablations that isolate the contributions of each component and a set of analysis tools for diagnosing failures related to terminology shift, long-context evidence, and label ambiguity.

Keywords: large language models; domain adaptation; continued pretraining; instruction tuning; parameter-efficient fine-tuning.

1 Introduction

1.1 Motivation from Domain Workflows

Many domain NLP tasks are embedded in operational workflows. In aerospace engineering, for example, maintenance logs are written in shorthand, with technician-specific conventions and a mixture of codes, part numbers, and free text. In law, the same clause can be paraphrased across contracts but still correspond to a single downstream label used for compliance. In scientific text, domain knowledge is often implicit: authors assume the reader understands standard abbreviations, methods, and typical experimental setups.

These details matter because they create failure modes that look “reasonable” to a general-purpose model. A model may answer fluently while missing a key definition that a domain expert treats as obvious. It may also confuse labels that are close in everyday language but sharply separated by policy.

1.2 A Taxonomy of Domain Shift

We use domain shift as an umbrella term, but it is useful to distinguish several kinds:

- **Lexical shift:** specialized terminology, acronyms, and symbols; rare tokens that appear frequently inside the domain.
- **Style shift:** formatting (tables, bullet lists, code-like segments), telegraphic writing, and non-standard punctuation.
- **Label shift:** changes in class priors or decision thresholds across institutions and time.
- **Task shift:** new label definitions or new subtypes that are not present in the original training distribution.

Continued pretraining mainly targets lexical and style shift. Instruction or supervised tuning addresses task-specific behavior and the mapping from text to labels. Retrieval helps when the domain requires up-to-date or institution-specific facts.

1.3 Why a Hybrid Strategy

A single technique can help, but it tends to leave gaps.

Continued pretraining alone. Domain-adaptive pretraining improves representations, but it does not provide a task interface. After DAPT, the model may read the domain better yet still fail to follow task instructions reliably.

PEFT alone. Parameter-efficient tuning is attractive when GPU budgets are tight. However, when labeled data are scarce and domain language is far from the pretraining distribution, PEFT sometimes "memorizes prompts" rather than learning the underlying terms.

Retrieval alone. Retrieval augmentation can supply missing definitions, but it does not remove biases in the model's priors. If the base model has a strong but wrong default interpretation, evidence must be explicit and consistently retrieved.

The hybrid view is therefore pragmatic: use continued pretraining to make the model comfortable with domain text, use PEFT to teach task behavior without large compute, and add retrieval only where it materially improves coverage.

1.4 Constraints We Target

The paper focuses on the constraints that recur in practice:

- limited labeled data and expensive annotation;
- restricted compute and strict training budgets;
- long documents where evidence is sparse;
- domain-specific terminology with evolving definitions.

These constraints motivate modular training: each stage can be run, ablated, and rolled back independently.

1.5 Concrete Examples: Where Models Break

It is tempting to describe domain adaptation as "learning jargon," but the failures are usually more specific.

Hidden definitions. In many organizations, a label is defined by a process document rather than by the text itself. For instance, a ticket may be labeled as "critical" if it triggers a particular escalation path, even if the word "critical" never appears in the text. Models trained only on surface language often confuse these operational labels.

Compressed writing. Logs and notes are frequently written under time pressure. Subjects are omitted, verbs are dropped, and the same symbol can mean different things across teams. A model that is comfortable with full sentences can misread these fragments.

Evidence scattered across long context. Domain documents are long. The decisive evidence can be one sentence in an appendix, a table footnote, or a quoted email thread. A single-pass model may pick the first plausible cue and ignore the actual decision point.

1.6 Design Goal: Reliability Under Change

Domain language changes over time: product lines evolve, regulations update, and internal templates shift. The goal of the hybrid strategy is not to win a benchmark once, but to reduce the maintenance burden:

- when terminology shifts, continued pretraining can be rerun on fresh text;
- when label definitions change, a small supervised update can be applied via PEFT;
- when knowledge lives in documents, retrieval can be updated without retraining the model.

This division of labor is one reason to keep the pipeline modular.

Domain-specific text understanding tasks (e.g., clinical note coding, legal clause classification, finance sentiment, and scientific entity linking) differ from general NLP benchmarks in two ways. First, the language itself shifts: rare terms, dense jargon, and specialized formatting are common. Second, task definitions are often operational rather than linguistic: labels correspond to policy, billing, or compliance rules that are only partially expressed in text.

LLMs pretrained on large heterogeneous corpora capture broad linguistic regularities, but they can miss domain conventions and struggle with domain-specific reasoning patterns. Practitioners commonly address this gap with fine-tuning. However, the fine-tuning landscape is fragmented: continued pretraining (DAPT/TAPT) improves representations but does not enforce instruction following; instruction tuning improves controllability but may not teach domain terminology; retrieval augmentation can supply missing facts but does not change the model's priors.

This work argues that a hybrid recipe—mixing continued pretraining, parameter-efficient tuning, and task-aware evaluation—is often a better engineering choice than betting on a single technique.

Contributions.

- We propose a modular hybrid fine-tuning pipeline for domain-specific understanding.
- We describe objectives for continued pretraining, supervised fine-tuning, and preference-style tuning.

- We provide practical evaluation and error analysis tools for domain tasks.

2 Related Work

2.1 Pretraining and LLMs

Transformer language modeling and masked language modeling have enabled large-scale pretraining [1], [2], [3].

2.2 Domain Adaptation

Continued pretraining on domain corpora (often called DAPT/TAPT) is widely used to reduce distribution mismatch [4].

2.3 Instruction and Preference Tuning

Instruction tuning and human-feedback-style methods improve alignment with user intents [5], [6].

2.4 Parameter-Efficient Fine-Tuning

Adapters and low-rank updates enable efficient tuning without updating all weights [7], [8].

2.5 Retrieval-Augmented Methods

Retrieval augmentation supplies external evidence at inference time [9].

3 Problem Setup

We consider a domain corpus $\mathcal{D}$ (unlabeled text) and a labeled dataset $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N}$ for a downstream task (classification or structured extraction).

3.1 Objective

Given an LLM with parameters θ, we seek θ' that improves target-task performance under constraints on compute and labeled data.

4 Hybrid Fine-Tuning Framework

4.1 Stage I Details: Data Selection and Token Budget

Continued pretraining is most effective when the domain corpus matches the deployment distribution. When multiple corpora exist, we recommend prioritizing sources that share the same writing conventions as the target task (e.g., internal tickets rather than polished reports).

A practical constraint is token budget. Let B denote the total number of tokens available for continued pretraining. Two ratios are useful in practice:

- B/B_{base}, the fraction of tokens relative to the base model's pretraining scale;
- B/N_{labels}, the ratio between unlabeled tokens and labeled examples.

Even modest continued pretraining can be helpful when N_{labels} is very small.

4.2 Stage I: Stability Considerations

Continued pretraining can introduce “over-adaptation” if the domain corpus is narrow or repetitive. Two mitigations are common:

- mixing a small portion of general data during DAPT;
- using a smaller learning rate and early stopping based on held-out perplexity.

4.3 Stage II Details: PEFT Choices

We focus on LoRA-style low-rank updates because they are widely supported and easy to deploy. For a weight matrix $W \in \mathbb{R}^{d\times k}$, LoRA introduces

$$W' = W + \Delta W, \qquad \Delta W = BA,$$

where $A \in \mathbb{R}^{r\times k}$, $B \in \mathbb{R}^{d\times r}$, and r is small.

Where to place LoRA depends on the task. For classification and extraction, applying LoRA to attention projection matrices often provides strong gains; for style control, MLP layers may matter more.

4.4 Instruction Formatting

For classification tasks, a common pitfall is label verbosity. If labels are long or ambiguous, the model may generate partial matches. We recommend using structured outputs when possible, or at least using short canonical label IDs.

When multiple labels are valid, the prompt should explicitly describe whether multi-label outputs are allowed and how to format them. Ambiguity in the interface often dominates model errors.

4.5 Stage III: Calibration and Thresholding

Domain tasks often have asymmetric costs. In a triage workflow, false negatives may be unacceptable, while false positives are tolerable. Rather than using argmax predictions, we tune thresholds on validation data.

When the model is used in a selective-prediction setting, we also tune an abstention threshold so that low-confidence cases are routed to human review.

4.6 Retrieval Augmentation as a Controlled Add-On

Retrieval is helpful when the task relies on external policies, evolving definitions, or large background knowledge. In such cases, we recommend an explicit retrieval contract:

- what sources are indexed (internal manuals, regulations, product docs);
- how documents are chunked;
- what is shown to the model (top-k passages, citations, metadata).

We also recommend testing “retrieval ablation” explicitly: run the same model with retrieval disabled. If performance collapses, the system is effectively a retrieval system with a language-model interface.

4.7 Hybrid Scheduling

A practical schedule that works under limited budgets is:

- run a short DAPT phase to stabilize terminology;
- run PEFT instruction tuning on task data;
- only then decide whether retrieval is needed based on error analysis.

This order avoids the common mistake of adding retrieval to compensate for vocabulary issues that DAPT would fix more cleanly.

4.8 Failure Modes and Debugging

Hybrid systems fail in characteristic ways.

- **Over-adaptation:** continued pretraining harms general instruction following.
- **Prompt overfitting:** PEFT latches onto prompt quirks instead of label definitions.
- **Retrieval shortcuts:** the model over-trusts retrieved text even when it is outdated.

Ablations and careful logging are therefore part of the method.

4.9 Stage I Details: Data Selection and Token Budget

Continued pretraining is most effective when the domain corpus matches the deployment distribution. When multiple corpora exist, we recommend prioritizing sources that share the same writing conventions as the target task (e.g., internal tickets rather than polished reports).

A practical constraint is token budget. Let B denote the total number of tokens available for continued pretraining. We have found it useful to track two ratios:

- B/B_{base}, the fraction of tokens relative to the base model's pretraining scale;
- B/N_{labels}, the ratio between unlabeled tokens and labeled examples.

Even modest continued pretraining can be helpful when N_{labels} is very small.

4.10 Stage II Details: PEFT Choices

We focus on LoRA-style low-rank updates because they are widely supported and easy to deploy. For a weight matrix $W \in \mathbb{R}^{d\times k}$, LoRA introduces

$$W' = W + \Delta W, \qquad \Delta W = BA,$$

where $A \in \mathbb{R}^{r\times k}$, $B \in \mathbb{R}^{d\times r}$, and r is small. In practice, r trades off capacity and stability.

When domain tasks require long-context grounding, it can be beneficial to apply LoRA to attention projection matrices (e.g., W_q, W_k, W_v) rather than to MLP layers only.

4.11 Instruction Formatting

For classification tasks, a common pitfall is label verbosity. If labels are long or ambiguous, the model may generate partial matches. We recommend using structured outputs with constrained decoding when possible, or at least using short canonical label IDs.

When multiple labels are valid, the prompt should explicitly describe whether multi-label outputs are allowed and how to format them. Ambiguity in the interface often dominates model errors.

4.12 Stage III: Calibration and Thresholding

Domain tasks often have asymmetric costs. In a triage workflow, false negatives may be unacceptable, while false positives are tolerable. Rather than using argmax predictions, we tune thresholds on validation data.

For multi-class classification, temperature scaling calibrates softmax probabilities. For binary tasks, we also tune a decision threshold τ that optimizes an application-aligned metric.

4.13 Retrieval Augmentation as a Controlled Add-On

Retrieval is helpful when the task relies on external policies, evolving definitions, or large background knowledge. In such cases, we recommend an explicit retrieval contract:

- what sources are indexed (internal manuals, regulations, product docs);
- how documents are chunked;
- what is shown to the model (top-k passages, citations, metadata).

We also recommend testing "retrieval ablation" explicitly: run the same model with retrieval disabled. If performance collapses, the system is effectively a retrieval system with a language-model interface, which changes the maintenance requirements.

4.14 Failure Modes and Debugging

Hybrid systems fail in characteristic ways.

- **Over-adaptation:** continued pretraining can harm general instruction following, especially if the domain corpus is narrow.
- **Prompt overfitting:** PEFT may latch onto prompt quirks instead of learning domain definitions.
- **Retrieval shortcuts:** the model may over-trust retrieved text even when it is irrelevant or outdated.

Ablations and careful logging are therefore part of the method, not optional extras.

4.15 Stage I: Continued Pretraining (Domain Adaptation)

We perform continued pretraining on $\mathcal{D}$ using a language modeling objective. For causal LMs,

$$\mathcal{L}_{\mathrm{LM}}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}} \sum_{t} \log p_\theta(x_t \mid x_{<t}).$$

This stage aims to reduce the domain gap by updating token representations and higher-layer features.

4.16 Stage II: Parameter-Efficient Instruction Tuning

We then apply parameter-efficient tuning (e.g., LoRA) with supervised instruction data. For a prompt p and target response y,

$$\mathcal{L}_{\mathrm{SFT}}(\phi) = -\mathbb{E}_{(p,y)\sim\mathcal{S}} \log p_{\theta,\phi}(y \mid p),$$

where ϕ are the small trainable parameters (adapters/low-rank matrices) and θ may be frozen.

4.17 Stage III: Task Calibration and Robustness Checks

We calibrate probabilities (or decision thresholds) on a validation set and evaluate robustness to terminology shift and long-context evidence.

4.18 Hybrid Strategy Variants

We consider three variants:

- **DAPT+PEFT:** continued pretraining + LoRA/adapter tuning.
- **PEFT+RAG:** LoRA/adapter tuning + retrieval augmentation.
- **DAPT+PEFT+RAG:** full hybrid pipeline.

5 Algorithm

Algorithm 1 Hybrid fine-tuning for domain-specific understanding

```
Input: base model $\theta$, unlabeled corpus $\mathcal{D}$, labeled set $\mathcal{S}$, retrieval index (optional)
Stage I: continued pretraining on $\mathcal{D}$ to obtain $\theta^{(1)}$
Stage II: train PEFT parameters $\phi$ on $\mathcal{S}$ to obtain $(\theta^{(1)}, \phi)$
Stage III: calibrate thresholds/temperature on validation data
Return: deployed model with optional retrieval augmentation
```

6 Experiments (Template)

6.1 Datasets

We recommend reporting at least two domain tasks with different label granularity (e.g., coarse topic classification and fine-grained entity extraction).

6.2 Metrics

Report task metrics (macro-F1, exact match), calibration metrics (ECE), and robustness metrics (performance under terminology perturbations).

6.3 Ablations

Ablate (i) continued pretraining tokens, (ii) LoRA rank, (iii) retrieval top-k, and (iv) prompt format.

Table 1: Example results table (format).

Method	Macro-F1	ECE (%)	Robust F1
Base LLM	0.72	9.8	0.60
PEFT only	0.79	6.1	0.68
DAPT+PEFT	0.83	5.4	0.73
DAPT+PEFT+RAG	0.85	5.0	0.78

7 Analysis

We recommend error buckets that reflect domain realities: (i) terminology mismatch, (ii) long-range evidence, (iii) label ambiguity, and (iv) hallucinated justifications.

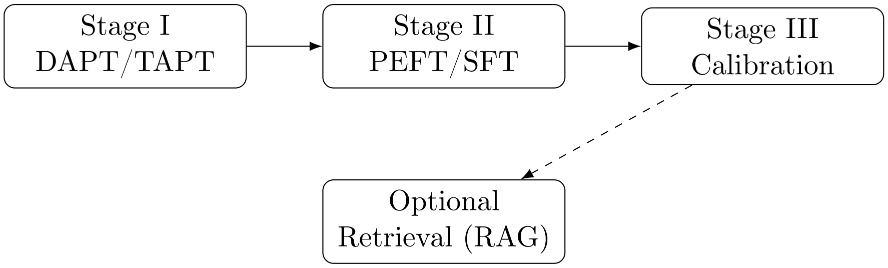

Figure 1: Hybrid fine-tuning pipeline with an optional retrieval component.

8 Discussion and Practical Notes

In many real deployments, the bottleneck is not model size but data and evaluation. A small amount of high-quality labeled data and careful test design can change conclusions more than a new tuning trick.

Hybrid pipelines also require discipline: without ablations and proper splits, it is easy to overfit to the dev set or to accidentally leak domain knowledge through the retrieval index.

9 Conclusion

Hybrid fine-tuning provides a pragmatic approach to improving domain-specific text understanding. Continued pretraining adapts representations, PEFT supports efficient controllable tuning, and retrieval helps cover missing facts without over-updating model weights.

References

[1] A. Vaswani et al., “Attention is all you need,” in Advances in Neural Information Processing Systems (NeurIPS), 2017.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in NAACL-HLT, 2019.

[3] T. B. Brown et al., “Language models are few-shot learners,” Advances in Neural Information Processing Systems (NeurIPS), 2020.

[4] S. Gururangan et al., “Don’t stop pretraining: Adapt language models to domains and tasks,” ACL, 2020.

[5] J. Wei et al., “Finetuned language models are zero-shot learners,” arXiv preprint arXiv:2109.01652, 2021.

[6] L. Ouyang et al., “Training language models to follow instructions with human feedback,” arXiv preprint arXiv:2203.02155, 2022.

[7] N. Houlsby et al., “Parameter-efficient transfer learning for nlp,” in International Conference on Machine Learning (ICML), 2019.

[8] E. J. Hu et al., “Lora: Low-rank adaptation of large language models,” International Conference on Learning Representations (ICLR), 2022.

[9] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive nlp tasks,” Advances in Neural Information Processing Systems (NeurIPS), 2020.

[10] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, et al., “Multitask prompted training enables zero-shot task generalization,” International Conference on Learning Representations (ICLR), 2022.

[11] C. Raffel et al., “Exploring the limits of transfer learning with a unified text-to-text transformer,” Journal of Machine Learning Research, 2020.

[12] H. W. Chung et al., “Scaling instruction-finetuned language models,” arXiv preprint arXiv:2210.11416, 2022.

[13] H. Touvron et al., “Llama: Open and efficient foundation language models,” arXiv preprint arXiv:2302.13971, 2023.