Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 12, 2025

EXPLORING FINE-TUNING TECHNIQUES FOR LARGE LANGUAGE MODELS IN DOMAIN-SPECIFIC APPLICATIONS

[1]Huzaifah Bin Khawar, [2]Talha Nadeem, [3]Muhammad Zarar, [4] Hood Khizer

[1] AI Researcher, Research and Development Department, Trouve Labs, AHOY DMCC and The Superior University Lahore, Pakistan huzaifahkhawar.ieee@gmail.com

[2] Systems Architect AI/ML, Research and Development Department, Trouve Labs, AHOY DMCC and The Superior University Lahore, Pakistan talhanadeem.ieee@gmail.com

[3] AI Researcher, Research and Development Department, Trouve Labs, AHOY DMCC and The Superior University Lahore, Pakistan muhammadzarar.ieee@gmail.com

[4] CEO Trouve Labs, CTO AHOY DMCC, Research and Development Department, Trouve Labs, AHOY DMCC, Dubai hoodkhizer@gmail.com

DOI:

Keywords
Large language models, domain adaptation, parameter-efficient fine-tuning, LoRA, QLoRA, continued pre-training, evaluation, model governance.

Article History
Received: 04 Dec, 2025
Accepted: 14 Dec, 2025
Published: 31 Dec, 2025

Copyright @Author
Corresponding Author: * Talha Nadeem*

Abstract

Domain-specific deployment of large language models (LLMs) requires more than transferring a general-purpose checkpoint to a small labelled dataset. A defensible adaptation pipeline must align the model, data, objective, evaluation protocol, and operational controls with the target domain. This paper presents a structured technical synthesis of full-parameter fine-tuning, continued domain- and task-adaptive pre-training, adapters, prompt and prefix tuning, low-rank adaptation (LoRA), and quantized LoRA (QLoRA). It replaces unsupported cross-domain score comparisons with an evidence-bounded framework that separates established findings from implementation guidance. The methodology covers corpus governance, leakage-resistant data splitting, trainable-parameter selection, regularized optimization, early stopping, calibration, robustness, fairness, privacy, latency, and cost evaluation. Evidence from biomedical, scientific, legal, and financial language modelling shows that domain-relevant pre-training and carefully controlled adaptation can improve specialised tasks, while parameter-efficient methods reduce storage and memory requirements. However, no method is uniformly best: the appropriate choice depends on model access, data volume, domain shift, compute, safety requirements, and expected maintenance frequency. The paper concludes with a reproducible decision framework and monitored deployment architecture for selecting, validating, and maintaining domain-adapted LLMs.

1. INTRODUCTION

Based on transformers, language models are able to learning of transferable representations from enormous corpora of text and be extrapolated to the task of classification, extraction, question answering, summarisation and generation. Self-attention mechanism as a scalable alternative to recurrent sequence processing: Transformer [1]. Later, BERT introduced a pre-training/few-shot fine-tuning paradigm for bidirectional encoders [2] and autoregressive scaling showed promising performances in few-shot scenarios on large decoder-only models [3]. Such improvements provided a practical basis for the development of reusability for a generic system rather than starting from scratch on each specialized system.

The achievement of a general proficiency in the language is not a guarantee of reliable functioning in the domain. The special vocabulary, conventions, implicit assumptions and "risk-sensitive" outputs in biomedical notes, statutes, engineering reports, and scientific articles are outputs. A model may be able to use the full model accurately but miss the meaning of an abbreviation, use an old rule and/or give a correct, though possibly not supported, answer to a question. The issue of domain adaptation is then defined both in terms of the predicted usefulness and in terms of the operational reliability. It should be considered as a managed life cycle which includes data governance, rollback and periodically questioning, and monitoring.

However, a few different adaptation mechanisms exist and often fine-tuning is the term used for them collectively. Full fine tuning: Updates the entire parameter set. The initial objective of self-supervised learning [4] continues with domain-adaptive pre-training (DAPT) or task-adaptive pre-training models (TAPT) for domain- or task-specific text. Parameter-efficient fine-tuning (PEFT) freezes most of the base parameters and adds small task-specific specific components like adapters [6], continuous prefixes [7], soft prompts [8] and low rank updates [9]. For saving memory during adaptation, QLoRA introduces 4-bit quantisation of the frozen base model [10]. Every technique alters different component of the optimisation and deployment problem.

1.1 Research Problem and Contributions

The aim of the research problem is to find the best way of adapting an LLM to a specific domain where data sizes, computing power needed, ability to access models, model speed, and risk constraints are not the same. Based on the experimental study, the paper discusses three questions asked: (i) which families of adaptations are technically suitable depending on the given resources and domain-shift requirements; (ii) how to organize the data preparation, objectives and assessment for carrying out the reproducible comparison; and (iii) which controls are necessary to implement and sustain the adapted model responsibly.

The contribution does not consist of a new un-reported experiment but of a coordinated, evidence-based approach. It adds tried and tested adaptation techniques, provides a practical decision process, provides consistent notation and learnable mathematical formulas, removes some unnecessary examples and citations of primary LLM research, and disconnects some tasks from notions of utility, calibration, robustness, fairness, privacy, latency, and cost. This scope does not assume that the heterogeneous domain metrics can be replaced by

each other on a "one-to-one" basis, and does not obscure the places where an empirical study is still needed.

Figure 1 shows the key adaptation surfaces on a Transformer and the difference between the frozen base and parts used to train the model.

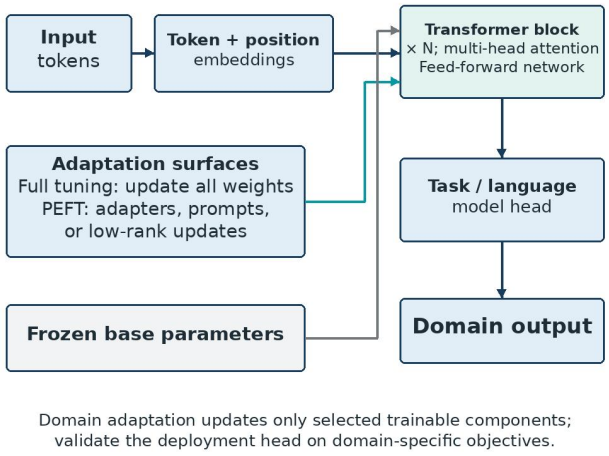


Figure 1. Rebuilt Transformer adaptation schematic showing frozen and trainable components.

2. RELATED WORK

2.1 Pre-training and Domain Adaptation

Pre-training focusses on building general representations (from a broad corpus) and adaptation focuses on refining those representations (by adapting them to the target distribution). In this work, ULMFiT showed that language-model pre-training, discriminative learning rates and gradual unfreezing help to transfer to text classification effectively [5]. Gururangan et al. subsequently demonstrated that further pre-training on the domain text and also pre-training on the unlabelled task data can further boost performance on biomedical, computer-science, news and review domains [4]. These findings reinforces a crucial and overall tenet that domain-relevant unlabelled text can be beneficial even in the absence of task labels.

Further evidence comes from domain-specific models. SciBERT adopted a scientific corpus and the vocabulary obtained from the scientific text to enhance scientific NLP tasks [11]. BioBERT further pre-trained BERT on biomedical corpora that showed improved performance on biomedical named entity recognition, biomedical relation extraction, and biomedical question answering [12]. ClinicalBERT variant focused on adapting representations to clinical notes [15] and LEGAL-BERT examined continued pre-training and training-from-scratch options for the legal domain [13]. FinBERT is an adaptation of BERT for financial sentiment analysis [14]. The above studies do not indicate universal numbers that were improved; gains will varies based on quality of corpora, task, baseline, and evaluative design.

2.2 Full-Parameter and Parameter-Efficient Tuning

Full fine-tuning provides the most options: All parameters can be modified. If the base model is fully accessible, there is sufficient representative data and the organisation can maintain and manage a full checkpoint of each release, then it is appropriate. It has drawbacks such as consuming a lot of memory and space, being prone to optimization factors, and overfitting or catastrophic forgetting if the target data is limited.

The model consists of a trainable module, called adapter, placed between the different layers of the frozen Transformer, which enables several tasks to use a single base model [6]. In prefix tuning, an attention modulatory continuous vector is learned on top of the original parameters of the model. Prompt tuning trains soft input embeddings, and is more competitive with larger models [8]. LoRA is used to update smaller matrices (low rank matrices)

to train and store fewer parameters than a full checkpoint [9]. QLoRA aims to achieve both of these benefits by freezing the respective processor in 4-bit quantization and using LoRA instead of fine-tuning the own trained model. Dettmers et al. showed, in the experimental setup, that a base model quantized to 4 bits with 48 GB of memory can still be adapted to a single 48 GB GPU [10]. This is a hardware- and implementations dependent result – it is not guaranteed for each model nor each duration of the sequence.

These methods are organised according to which component is being updated in Figure 2 and contrasted with respect to data, compute, storage and serving aspects in Table 1.

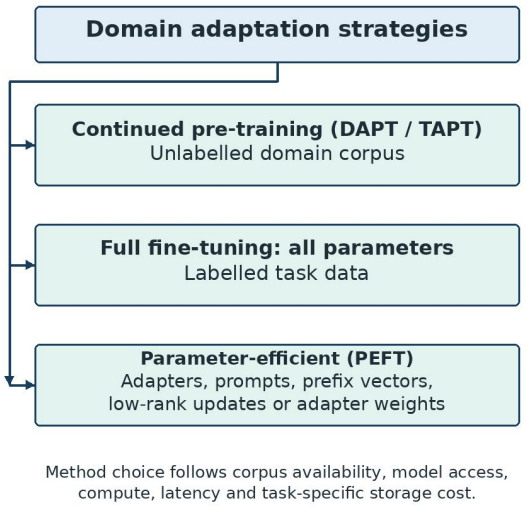


Figure 2. Taxonomy of domain-adaptation strategies and their trainable components.

Table 1. Comparison of principal adaptation methods.

Method	Updated component	Best suited to	Main limitation
DAPT/TAPT	Model weights under self-supervised loss	Large unlabelled domain corpus	Compute cost; forgetting; corpus bias
Full tuning	All parameters	Adequate labelled data and full model access	Memory, storage, and overfitting risk
Adapters	Inserted bottleneck modules	Many tasks sharing one base model	Architecture and serving complexity
Prompt/prefix	Continuous prompt vectors	Low-storage task adaptation	May trail full tuning on some small models
LoRA	Low-rank weight updates	Efficient supervised or instruction tuning	Rank and target-module selection
QLoRA	LoRA over a quantised frozen base	Limited accelerator memory	Quantisation and kernel constraints

3. METHODOLOGY

In this study structured technical-synthesis design type is employed. Nor does it suggest a new head-to-head test in the healthcare, law and finance fields, as none of the source manuscript datasets, model checkpoints, random seeds, full hyperparameters or auditable logs were submitted to facilitate a comparison. Rather, the approach is to structure the findings from first-hand studies and to rework them into a protocol for adapting and evaluating the findings, which can be reproduced. The results are heterogeneous and interpreted in the context of the specific task and the metric they come from, and not pooled as if the same thing as accuracy / F1 / AUC.

3.1 Evidence and Scope

The synthesis focuses on basic and primary research on Transformer pre-training, domain adaptive pretraining, domain-specific language models,

adapters, prompt-based tuning, LoRA, QLoRA, Evaluation, Documentation and Privacy. There needs to be a clear description of the method and identifiable publication history for studies to be included. Evidence is classified based on the trained adaptation component (reinforcement learning, distributed neural networks, or neural networks), the type of domain data needed (e.g., telemetry, satellite images), and the consequence of the adaptation during deployment (e.g. naval victory, economic loss, national security). Method selection support for this organisation is not confining themselves in the creation of a universal rank.

3.2 Data Governance and Preparation

The first step in domain adaptation is a specification of the data. Each corpus needs to be connected to its source, licence, or lawful basis, time of collection, language in which it was translated, intended purpose and exclusion criteria, and responsible owner. Access controls and, if applicable, de identification are required for sensitive data. Duplicate passages and "near" duplicate passages should be removed prior to splitting so as to prevent "optimistic evaluation," for the presence of duplicate and duplicate passages. If splitting a table randomly would send patient/client or contract data across partitions, or company data, author, or event across partitions, then the table needs to be split either in the temporal or entity dimension, whichever works best for the situation.

A training split is used to update the parameters, a validation split is used to choose checkpoints/hyper-parameters, and a held-out test split is used to estimate final performance. In order to solve this configuration, the test set has to be left untouched until the configuration is solved. Evaluation data should contain prompts that represent typical cases, challenging edge cases, refusal or abstention cases, adversarial inputs for generative tasks. Labels need to be well annotated in writing and then reviewed by appropriate and qualified domain experts where appropriate and for high stakes tasks.

Table 2. Minimum dataset-governance checklist.

Control	Required evidence	Failure prevented
Provenance	Source, owner, licence/lawful basis	Untraceable or unauthorised data
Scope	Domain, language, time period, exclusions	Population or temporal mismatch
Quality	Deduplication, normalisation, label audit	Noise and duplicate leakage
Privacy	Sensitivity review and access controls	Disclosure of protected information
Splitting	Entity/temporal rules and frozen test set	Target and evaluation leakage
Versioning	Dataset hash, code, and change log	Irreproducible releases

Figure 3 converts these controls into a reproducible workflow with explicit leakage and release gates.

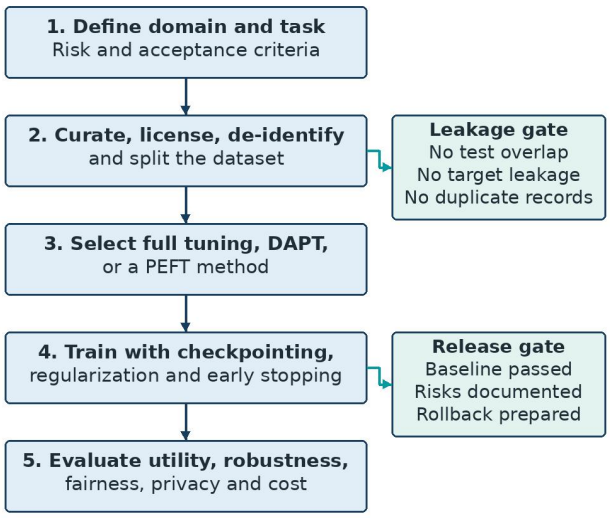


Figure 3. Reproducible fine-tuning workflow with leakage and release gates.

3.3 Adaptation Decision Process

The first question is whether there is a need to change the weights of the models? If knowledge is apt to change from time to time, sources must be cited at the time of inference, and/or if the model can only be accessed via an API, then prompt engineering or retrieval augmentation may be preferred. FINE-TUNING should be considered when stable adaptation in behavior, use of domain terminology, structure of output, boundaries of classification, or patterns of tool use will not be reliably achieved through prompting alone. If there is enough of the domain unlabelled domain text and it is distinctly different from the distribution in the base corpus, then the continued pre-training approach is used.

In the discussion of fine tuning, full tuning is a reference fine tuning parameter if there exists sufficient amount of data and compute. PEFT is ideal when the base checkpoint needs to be shared, multiple domain variants are required to be stored, and there is limited memory available for the accelerator. The following settings, which are unique to LORY, should be set: loRA rank, loRA scaling, loRA dropout, loRA target modules, learning rate, sequence length, batch construction, precision, checkpoint frequency. Compare against the same base model, tokenizer, training/example, stopping rule and evaluation harness.

3.4 Mathematical Formulation and Notation

Let $D = \{(x_i, y_i)\}$ for $i = 1,..., N$ denote the labelled domain dataset, $M(\theta)$ the pre-trained model, θ_0 its initial parameters, and θ_t the trainable subset. For C-class prediction, logits z are mapped to probabilities by the softmax function:

$$p_\theta(y = c|x) = \frac{\exp(z_c)}{\sum_{j=1}^{C} \exp(z_j)} \quad (1)$$

The mean cross-entropy objective is:

$$L_{CE}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log p_\theta(y_i|x_i) \quad (2)$$

A regularised objective can penalise the trainable update; λ controls the strength of the penalty:

$$L_{reg}(\theta) = L_{CE}(\theta) + \lambda \| \theta_t \|_2^2 \quad (3)$$

For LoRA, a frozen weight matrix W is augmented by a rank-r update. This scale α/r distinguishes between the size of the update and the rank:

$$W' = W + \frac{\alpha}{r} BA \quad (4)$$
$$B \in \mathbb{R}^{d_{out} \times r}, \quad A \in \mathbb{R}^{r \times d_{in}}$$

Equations (1)-(4) give the classification and adaptation aims. They can be inserted in a single article column and are still not displayed as a graphics image, but as a vector value word mathematics.

Table 3. Notation used in the adaptation framework.

Symbol	Meaning	Domain / shape
D	Domain dataset	$\{(x_i, y_i)\}$; $i = 1,..., N$
N	Number of labelled examples	Positive integer

Symbol	Meaning	Domain / shape
M(θ)	Model parameterised by θ	Model mapping
θ_0 , θ_t	Initial and trainable parameters	Subsets of $\mathbb{R}^p$
z, C	Logits and class count	$\mathbb{R}^c$; positive integer
λ	Regularisation coefficient	λ >= 0
r, α	LoRA rank and scale	r > 0; α > 0
TP, TN, FP, FN	Confusion-matrix counts	Non-negative integers

3.5 Training and Evaluation Protocol

All of the following information should be recorded while training: bases model identifier & revision, tokenizer, dataset version, trainable modules, optimiser, learning-rate schedule, weight decay, effective batch size, maximum sequence length, precision, gradient accumulation, gradient clipping, seed, epochs, evaluation frequency, early stopping rule. For implementations that need to be distributed, the versioned library of Transformers, ensuring the same exact software and model version, can be used [16]. Thus, the commonly used AdamW method is a method without a correlated updating of the gradient and weight decay [17]. For small datasets and/or when tuning is not stable, multiple seeds are recommended; only the best of the run(s) should be reported, making it look worse than it is [18].

There might be instances where classification evaluation is only meaningful when there is class balance, such as when the numbered areas are located around a tree surrounded by trees. Per-class precision, per-class recall and with an explicit macro, micro or weighted aggregation precision-recall should be reported. The usual binary definitions are:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

$$F_1 = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

When producing probability outputs, calibration is of concern as it can be optimistically ranked. Expected calibration error (ECE) divides predictions into M confidence areas B_m :

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)| \tag{8}$$

The use of generative evaluation should include task-specific automatic measures as well as human or expert (blinded) assessment. If the information is limited, the overlap in token or exact match can be helpful for extraction, but ROUGE or other measures of overlap can be helpful for summarisation but not for achieving factual correctness. There will therefore be weight given to faithfulness to source material, unsupported claims, harmful content, instructions followed, forbearance, robustness to paraphrase, responses made by subgroups, latency, throughput, peak memory, and cost. A model should be compared against the baseline model, and to simpler alternatives for adaption.

Algorithm1 formalizes the entire adaptation, validation, release and monitoring cycle.

Algorithm 1 Controlled Domain-Specific LLM Adaptation and Release

Require: approved base model M(theta_0), versioned domain dataset D, acceptance criteria A

Ensure: released model version v or documented no-release decision

1. Validate provenance, privacy, licences, labels, duplicates, and split boundaries.
2. Freeze the test set; establish an unfine-tuned baseline.
3. Select DAPT/TAPT, full tuning, or PEFT from data, compute, and serving constraints.
4. Fix the search space, seeds, stopping rule, and evaluation harness.
5. Train on D_train; select checkpoints only with D_val.
6. Evaluate utility, calibration, robustness, fairness, privacy, latency, and cost on D_test.
7. If every mandatory criterion in A passes, register weights/adapters, tokenizer, data version, code, and model card.
8. Deploy behind access controls, logging, fallback, and rollback.
9. Otherwise, document failure modes and do not promote the candidate.
10. Monitor drift and incidents; trigger review or retraining when thresholds are exceeded.

4. FINDINGS AND DISCUSSION

4.1 What the Evidence Supports

We do not conclude, and it does not have to be the case, that fine-tuning always provides a fixed percentage gain but that relevance from pre-training or adaptation distribution to the target distribution usually gives more benefit to specialised representations. In Gururangan et al. [4] DAPT and TAPT saw improvements in multiple tasks and resources. SciBERT [11] and BioBERT [12] demonstrate the same above for their respective domains and ClinicalBERT, LEGAL-BERT [13] and FinBERT [14] for their respective bases. As these studies are based on different sets of data, on different baselines, with different sizes of model, and with different metrics, it is important to interpret the values within each study and to not mix them all together into a cross domain score.

PEFT techniques shift efficiency frontier. Also adaptors are used to use a common frozen base for small-length task modules [6]. Instead of updating the complete network [7] and in the process of prompt tuning only the necessary part of the network is updated [8]. LoRA learns low-rank weight updates and infers efficiently using low-rank weights to deploy without extra latencies [9] (when weights can be merged, subject to implementation details). QLoRA quantises the frozen model parameters during training, thereby decreasing the memory used for training [10]. While these methods do bring down the cost of storing variants, they don't get rid of the need to have representative data, hyperparameter validation, risk testing, etc.

Table 4. Evidence-bounded synthesis of adaptation findings.

Finding	Supported interpretation	Boundary
Domain text helps	DAPT/TAPT and domain models often improve specialised tasks [4], [11]-[15]	Magnitude is task- and baseline-specific
PEFT saves trainable/storage parameters	Adapters, prompts, and LoRA share a frozen base [6]-[9]	Quality and serving trade-offs remain
Quantisation lowers training memory	QLoRA backpropagates through a frozen 4-bit base [10]	Hardware, kernels, and sequence length matter

Finding	Supported interpretation	Boundary
One metric is insufficient	Utility must be paired with calibration and risk measures [19], [20]	Metric choice depends on task and harm
Documentation is part of release	Model cards record intended use and limitations [21]	Documentation does not replace testing

4.2 Method Selection Trade-offs

The full fine-tuning method is the most flexible as it requires a full domain-specific checkpoint, but also comes with the largest training and storage overhead. DAPT works where there is a large amount of unlabelled domain text, but can be more costly than supervised PEFT and it can modify general capabilities. Modular and handy for many applications – but custom modules can add complexity to serving. Prompt and prefix methods require very little in terms of storage space, but are often strongly dependent on the model scale and the type of tasks. For PEFT, LoRA is a good general baselines with rank and target module tunability without base weights change. QLoRA is appealing when there is memory limitations, yet it is important to test the quantisation configuration with the target hardware.

The selection also relies on model possession. Weight fine-tuning could be unavailable or provider dependent in the case of an inference API only. Local training can lead to increased goals of reproducibility and governance when open weights are available while raising the expectations on operational responsibility. Retrieval can bring forward current knowledge more directly than sequentially embedding new facts in parameters in rapidly changing domains. A hybrid design might be fine-tuned in terms of behaviour and words; retrieval could provide current and attributable information.

4.3 Domain-Specific Considerations

To be helpful in healthcare adaptation, the privacy of data must be protected, a clinically meaningful evaluation must be performed, a subgroup analysis must be tried and reviewed by qualified professionals. Do not use unsupervised clinically if the result is a high aggregate score. Not only do legal systems need jurisdiction, date and authority, but they also need to be evaluated with reference to that, meaning that a "fluent" answer defining a precedent only, but not citing any existing law, is not acceptable. Temporal splitting is necessary in financial systems to prevent look ahead bias and systems should be test on market and reporting changes. In Scientific and Engineering applications, requirements on terminology coverage, unit consistency, source traceability and validation of documents apart from the ones used for training are required.

Tokenizer behaviour can also be a factor. Specialized words that split into many subwords will add length to a sequence and will have Domain specific vocabularies were used in models like SciBERT [11] but updating tokenizers was also a potential drawback when training checkpoints. Prior to replacing the tokenizer, practitioners would want to measure the fragmentation, use of out-of-vocabulary surrogates, maximum-length truncation, and benefit downstream.

4.4 Reliability, Bias and Privacy

Fine-tuning can consolidate errors/biases in a very small corpus. Class size imbalances, disagreements among annotators, historical discrimination and limited-stake-positional data may be transformed

into confident model behaviour. If sub-group analyses are lawful and meaningful, then they should be reported; errors to worse case scenarios should be considered; and a human reviewed path should be documented. A model-card approach [21] should be followed: Documentation should describe the intended uses, out-of-scope applications, characteristics of training data, evaluation conditions and limitations are known.

Even if direct identifiers are deleted, data endangers privacy. The memorised capability of large models to repeat rare sequences can reveal sensitive text in adversarial examples, as revealed by the research of training-data extraction [22]. Mitigation involves minimising sensitive data, deduplication, access control, canary / memorisation tests, output filtering and an incident process. These controls will minimise, but not remove, risks.

4.5 Deployment and Monitoring

registry of adapted model should be shared and connected to the base model, to the Weights or merged weight, to the tokenizer, to the dataset version, to the code revision, to the configuration, to the evaluation report, to the approvals and to the model card. A serving layer should require authentication, rate limiting, logging in accordance with privacy requirements, input/output protections and reversion / abstaining path. Task quality, refusal and escalation rates, drift, harmful outputs, latency, throughput, memory, cost should all be monitored. An automatic continual learning model should be discontinued and instead, actions should be taken to investigate, roll back, review data or retrain if the threshold is breached.

The resulting architecture of adaptation to deployment and its feedback loop is summarised in the figure below, 4.

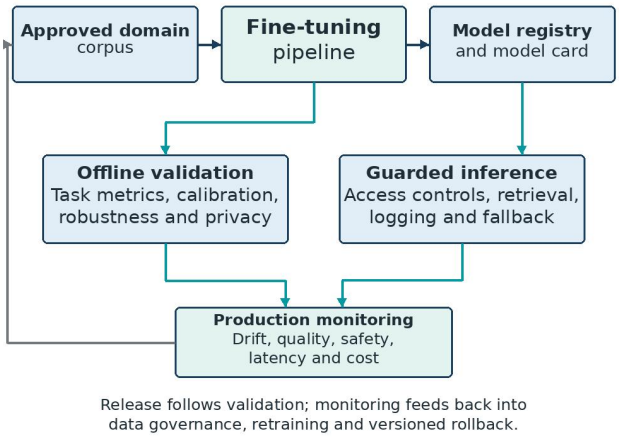


Figure 4. Rebuilt domain-specific adaptation, validation, deployment, and monitoring architecture.

4.6 Limitations

This paper is not a systematic meta analysis or a new multi domain experiment; rather, a technical synthesis. It consequently does not calculate a combined effect size. The literature is constantly evolving and efficiency is reported to be dependent on hardware/software kernels, precision, sequence length, batch construction, model architecture. English and relatively well resourced sectors tend to have domain-specific studies, as do English and mathematics. These constraints further underscore the need for the replication of the comparisons that might be conducted in the intended domain, model, infrastructure and risk profile.

5. CONCLUSION AND FUTURE WORK

LM adaptation for the specific domain is a multi-aspect problem (data-optimisation-evaluation-governance). The representations can be learned to agree with domain language through continued pre-training, whereas the full fine-tuning gives maximum flexibility, and PEFT methods such as LORA, QLoRA, Prompt or Prefix tuning consume

fewer parameters or memory resources during the process of representation learning for the adapted domain. There isn't any one that is always better. Defensible selection accounts for domain shift, labelled and unlabelled data, access to models, access to computing, model maintenance, model serving constraints, the cost of error.

The updated architecture has splits that support leakage and inputs that are versioned, baseline points that are fixed, equations that can be edited and consistently defined, multi-dimensional evaluation, release gates, monitoring, and rollback, among other features. Future research could include comparing adaptation methods under the two settings: matched data, and compute budgets; evaluating multilingual and low-resource domains; exploring continual adaptation without catastrophic forgetting; creating more robust privacy and memorisation tests; and investigate the interaction of fine-tuning with retrieval, tools and human oversight. Improvements claims should be grounded in specific datasets, baselines and metrics of uncertainty, and not rolled over from one context to another.

REFERENCES

1. A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems 30, pp. 5998-6008, 2017.
2. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, pp. 4171-4186, 2019, doi: 10.18653/v1/N19-1423.
3. T. B. Brown et al., "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems, vol. 33, pp. 1877-1901, 2020.
. S. Gururangan et al., "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks," in Proc. 58th Annual Meeting of the ACL, pp. 8342-8360, 2020, doi: 10.18653/v1/2020.acl-main.740.
. J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," in Proc. 56th Annual Meeting of the ACL, pp. 328-339, 2018, doi: 10.18653/v1/P18-1031.
. N. Houlsby et al., "Parameter-Efficient Transfer Learning for NLP," in Proc. 36th International Conference on Machine Learning, PMLR, vol. 97, pp. 2790-2799, 2019.
. X. L. Li and P. Liang, "Prefix-Tuning: Optimizing Continuous Prompts for Generation," in Proc. ACL-IJCNLP, pp. 4582-4597, 2021, doi: 10.18653/v1/2021.acl-long.353.
. B. Lester, R. Al-Rfou, and N. Constant, "The Power of Scale for Parameter-Efficient Prompt Tuning," in Proc. EMNLP, pp. 3045-3059, 2021, doi: 10.18653/v1/2021.emnlp-main.243.
. E. J. Hu et al., "LoRA: Low-Rank Adaptation of Large Language Models," in International Conference on Learning Representations, 2022, arXiv:2106.09685.
0. T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs," in Advances in Neural Information Processing Systems, vol. 36, 2023, arXiv:2305.14314.
1. I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A Pretrained Language Model for Scientific Text," in Proc. EMNLP-IJCNLP, pp. 3615-3620, 2019, doi: 10.18653/v1/D19-1371.
2. J. Lee et al., "BioBERT: A Pre-trained Biomedical Language Representation Model for Biomedical Text Mining," Bioinformatics, vol. 36, no. 4, pp.

1234-1240, 2020, doi: 10.1093/bioinformatics/btz682.
13. I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The Muppets Straight Out of Law School," in Findings of EMNLP, pp. 2898-2904, 2020, doi: 10.18653/v1/2020.findings-emnlp.261.
14. D. Araci, "FinBERT: Financial Sentiment Analysis with Pre-trained Language Models," arXiv:1908.10063, 2019.
15. E. Alsentzer et al., "Publicly Available Clinical BERT Embeddings," in Proc. 2nd Clinical Natural Language Processing Workshop, pp. 72-78, 2019, doi: 10.18653/v1/W19-1909.
16. T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," in Proc. EMNLP: System Demonstrations, pp. 38-45, 2020, doi: 10.18653/v1/2020.emnlp-demos.6.
17. I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in International Conference on Learning Representations, 2019, arXiv:1711.05101.
8. J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. Smith, "Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping," arXiv:2002.06305, 2020.
9. M. Sokolova and G. Lapalme, "A Systematic Analysis of Performance Measures for Classification Tasks," Information Processing & Management, vol. 45, no. 4, pp. 427-437, 2009, doi: 10.1016/j.ipm.2009.03.002.
0. P. Liang et al., "Holistic Evaluation of Language Models," Transactions on Machine Learning Research, 2023.
1. M. Mitchell et al., "Model Cards for Model Reporting," in Proc. Conference on Fairness, Accountability, and Transparency, pp. 220-229, 2019, doi: 10.1145/3287560.3287596.
2. N. Carlini et al., "Extracting Training Data from Large Language Models," in 30th USENIX Security Symposium, pp. 2633-2650, 2021.