International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 6, November - December 2024 |

DOI:10.15662/IJEETR.2024.0606030

Prompting, Retrieval, and Fine-Tuning: Foundations of Enterprise Language Model Adaptation

Karan Nisar

Independent Researcher, USA

ABSTRACT: General-purpose language models must be specialised to an organisation's documents, policies, vocabulary and output formats before they produce dependable value. Three mechanisms dominate practice: prompt engineering, which conditions a frozen model at inference time; retrieval-augmented generation (RAG), which attaches an external, non-parametric memory to a frozen model; and parameter-efficient fine-tuning, which modifies a small number of weights on curated task data. These are usually described separately, in the vocabulary of the research communities that produced them, which leaves enterprise architects without a common frame in which to compare them. This paper establishes that frame. We review the mechanism, enterprise strengths and enterprise limitations of each technique, together with the hybrid architectures that combine retrieval with weight modification, and we argue that the four resulting archetypes differ along a single structural axis: where the task knowledge required for a correct answer is stored, and which component is mutable at run time. Under prompting, knowledge is parametric and the input is mutable; under RAG, knowledge is external and an index is mutable; under fine-tuning, knowledge and behaviour are parametric and the weights are mutable; under hybrids, both are mutable on different cycles. We show that this one difference, rather than a list of independent trade-offs, accounts for the divergent update latency, citation capability, entitlement enforcement, erasure behaviour and maintenance profile of the four archetypes. The paper is deliberately descriptive: it does not rank the archetypes or prescribe a choice, but supplies the structural vocabulary on which such comparisons and decision procedures can be built.

KEYWORDS: Large language models; prompt engineering; retrieval-augmented generation; parameter-efficient fine-tuning; low-rank adaptation; enterprise architecture; model adaptation; in-context learning.

I. INTRODUCTION

The transformer architecture [1] and the finding that sufficiently large autoregressive models perform novel tasks from instructions and a few demonstrations [2] produced the class of general-purpose systems that Bommasani et al. termed foundation models [3]. These systems moved quickly from research artefacts into enterprise budgets, and with that move came a question the research literature does not answer directly: a pre-trained model is general, but the work an enterprise needs done is specific, so by what mechanism should the gap be closed?

Three mechanisms dominate. Prompt engineering conditions a frozen model through its input, exploiting in-context learning [2], [4]. RAG retrieves evidence from an external index at query time and supplies it alongside the question, attaching a non-parametric memory to an unchanged model [5]. Fine-tuning modifies parameters on task data, in enterprise settings almost always through parameter-efficient methods such as low-rank adaptation [6]. In practice the three compose, and hybrid designs that fine-tune a generator specifically to consume retrieved evidence are increasingly common.

Each mechanism arrived from a different research tradition and is usually explained in that tradition's terms: prompting in the vocabulary of few-shot learning, retrieval in that of open-domain question answering, fine-tuning in that of transfer learning. An architect comparing them must therefore reconcile three descriptions that share no frame of reference. The result, in practice, is a proliferation of attribute-by-attribute comparison tables in which cost, freshness, latency, auditability and maintenance appear as unrelated properties to be weighed one at a time.

This paper argues that they are not unrelated. Our claim is that the four adaptation archetypes differ along one structural axis, the location of task knowledge and the identity of the mutable component, and that most of the properties enterprises care about follow from where an archetype sits on that axis rather than being independent facts about it.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 6, November - December 2024 |

DOI:10.15662/IJEETR.2024.0606030

This is useful in two ways: it compresses a long comparison table into one distinction that can be reasoned about, and it separates properties that are architectural, and therefore stable, from properties contingent on current pricing or model capability.

The scope is deliberately limited. This paper describes mechanisms and their structural relationships. It does not quantify their relative cost or benchmark their accuracy, and it does not prescribe which to select for a given task; those are separate questions requiring evidence and methodology this paper does not supply. Section II covers the necessary background on foundation models and in-context learning. Sections III, IV and V treat prompting, retrieval and fine-tuning in turn. Section VI develops the structural argument and its consequences. Sections VII and VIII discuss implications and limitations.

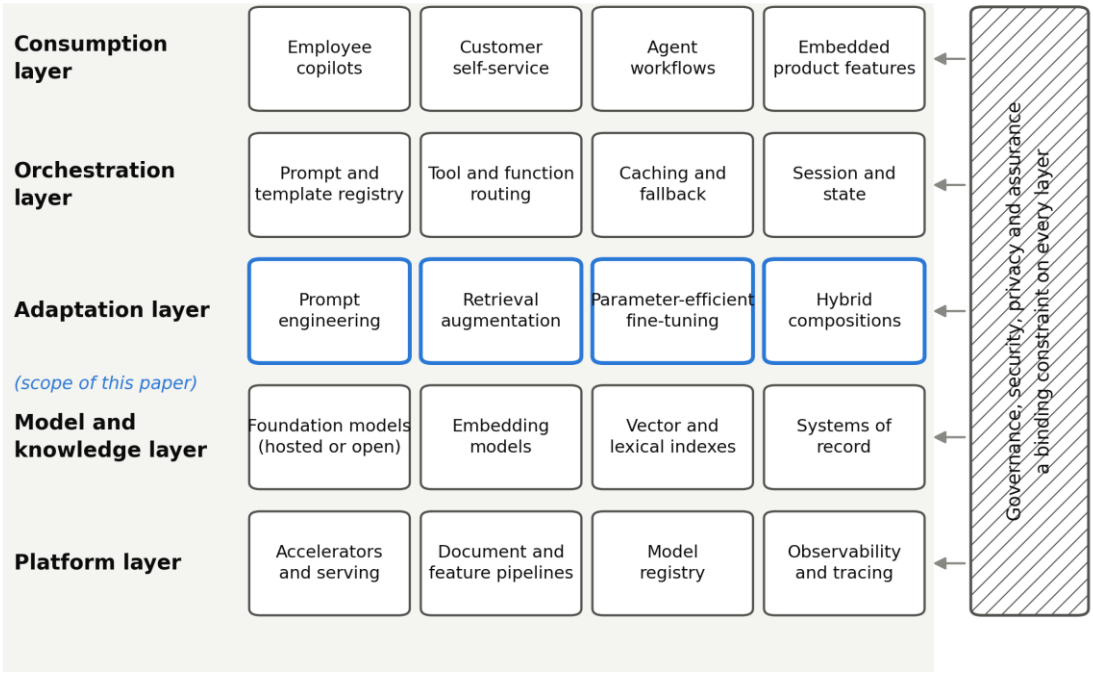

Figure.1. The enterprise language model stack, organised as five technical layers with governance as a cross-cutting constraint rather than a layer of its own. The adaptation layer, highlighted, is the subject of this paper.

II. BACKGROUND

A. Foundation models and the hosting bifurcation

Contemporary language models are decoder-only transformers trained with a next-token objective on large heterogeneous corpora [2], [7], and empirical scaling studies established that capability grows smoothly with parameters, data and compute [8], [9]. The enterprise consequence is a bifurcation that constrains every adaptation decision downstream. Frontier capability sits in a small number of very large models that cannot be reproduced in-house and are consumed through a hosted API. A competitive tier of openly distributed models [10], [11] is small enough to host, quantise and fine-tune within an ordinary infrastructure budget. This choice is logically prior to the adaptation choice, because several adaptation options are unavailable, or commercially unattractive, on the hosted side. An architect who has not settled the hosting question has not yet reached the adaptation question.

Alignment is the second relevant thread. Instruction tuning [12] and reinforcement learning from human feedback [13], later simplified by direct preference optimisation [14], produce models that follow instructions rather than merely continuing text. This matters twice over here: instruction-following is what makes prompting a viable adaptation mechanism at all, and it establishes the default behaviour against which any further behavioural adaptation is measured. Foundation models also carry properties that complicate enterprise use. They are opaque, their training-data provenance is often undisclosed, their factual knowledge is frozen at a training cut-off, and they produce fluent but unsupported

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 6, November - December 2024 |

DOI:10.15662/IJEETR.2024.0606030

assertions, a failure mode surveyed for generation generally by Ji et al. [15] and for language models specifically by Huang et al. [16].

B. In-context learning and the limits of long contexts

In-context learning lets a frozen model perform a task specified entirely in its input [2]. The mechanism is shallower than it first appears, in ways that bear directly on how far prompting can be pushed. Demonstrations function partly as specifications of format and label space rather than as vehicles that teach an input-output mapping [17]. Few-shot performance is sensitive to exemplar order and to majority-label and recency biases that calibration only partly corrects [18]. And output quality varies substantially under semantically equivalent changes to prompt formatting [19], which means a prompt is a fragile artefact in a way that source code is not.

Context windows have expanded rapidly across model generations [20], and long contexts are sometimes advanced as a general substitute for retrieval. Two considerations qualify this. Retrieval quality inside a long window degrades with the position of the relevant evidence, with models using information at the beginning and end of a window more reliably than material in the middle [21]. And the cost of a long prompt is incurred on every single request rather than once. Long contexts change the constants of the retrieval trade-off; they do not remove it.

III. PROMPT ENGINEERING

C. Mechanism

Prompt engineering conditions a frozen model through its input: a role and task specification, an output-format contract, constraints, and optionally a small number of exemplars. It is surveyed as a research area by Liu et al. [4] and, for the current model generation, by Schulhoff et al. [22]. The technique family that matters for enterprise work is narrow. Chain-of-thought prompting elicits intermediate reasoning steps and improves multi-step arithmetic and symbolic tasks at larger model scales [23]. ReAct interleaves reasoning with tool calls and is the pattern underlying most enterprise agent designs [24].

D. Enterprise strengths

Prompting is the only archetype with essentially zero build cost, no training-data requirement, and iteration measured in hours. It leaves no residue: a prompt can be withdrawn or rewritten without a training cycle, which suits volatile requirements and early-stage work where the specification is itself unstable. It is also the only archetype available under every combination of hosting model and data-residency constraint, because it modifies nothing and persists nothing.

E. Enterprise limitations

Three limitations are decisive. First, prompting cannot supply knowledge the model does not already hold and cannot see in its input; it can only reorganise what is in the weights or in the context window. Second, its behavioural reach is bounded. Instructions and a handful of exemplars change format and tone reliably, but reproducing a complex house style, a bespoke reasoning procedure or a rigid schema consistently across a long tail of inputs is precisely where prompt-only baselines plateau. Third, quality is fragile with respect to prompt surface form [17], [18], [19], which means prompt changes require regression testing with the discipline normally applied to code.

The recurring-cost consequence is structural rather than incidental: instructions and exemplars are re-transmitted and re-processed on every request, so a long prompt is a permanent per-request tax rather than an amortised investment. Prefix caching and paged key-value management reduce the marginal cost of long shared prefixes considerably [25] and should be assumed present in any serious comparison, but they change the constant, not the shape.

IV. RETRIEVAL-AUGMENTED GENERATION

F. Mechanism and pipeline

Retrieval augmentation predates instruction-following models. REALM introduced retrieval during pre-training [26], RAG established the generator-plus-retriever formulation and the parametric versus non-parametric memory distinction [5], dense passage retrieval made dense first-stage retrieval practical [27], and fusion-in-decoder demonstrated effective aggregation across many passages [28]. Gao et al. survey the design space for the current model generation [29]. More recent work makes retrieval adaptive rather than unconditional, either by training the model to decide when to retrieve and to critique what it retrieves [30], or by adapting the retriever to a black-box model [31].

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 6, November - December 2024 |

DOI:10.15662/IJEETR.2024.0606030

A production system comprises two paths (Figure 2). The offline ingestion path extracts and normalises documents, segments them into chunks, embeds the chunks, and stores vectors alongside lexical postings and access-control metadata. Embedding choice is consequential and can be grounded in multi-task benchmarks [32]; sentence-level bi-encoder training established the practical approach [33]. Retrieval at scale relies on approximate nearest-neighbour indexes, in practice graph-based HNSW [34]. The online path optionally rewrites the query, retrieves candidates using a hybrid of dense and sparse scoring, re-ranks with a cross-encoder [35], assembles surviving evidence into the context window, and generates an answer with citations.

G. Enterprise strengths

RAG's advantages follow from the location of its knowledge rather than from any property of the model. Because the corpus is external, an update is an indexing operation rather than a training operation, which makes retrieval the natural response to knowledge that changes. Because retrieved evidence is an identifiable object, the system can cite, which converts an assurance obligation from an interpretability problem into a provenance problem. Because access-control metadata travels with the chunk, entitlement can be enforced per user at retrieval time, which no weight-based mechanism can do. And because the base model is untouched, the corpus can be corrected, redacted or deleted, satisfying erasure requirements that a fine-tuned model cannot.

H. Enterprise limitations

RAG is frequently mis-sold as a cure for unsupported generation. Grounding reduces it but does not eliminate it: a model may ignore retrieved evidence, integrate conflicting passages incorrectly, or fail to abstain when the corpus does not contain the answer. Operationally, retrieval converts a model problem into a distributed-systems problem, and Barnett et al. document recurring failure points in exactly those seams, including missing content, ranking failures, consolidation loss and incorrect specificity [36]. The retrieval stack also adds per-request latency and cost and introduces a new attack surface, notably indirect prompt injection through retrieved content [37].

A subtler limitation is that retrieval addresses knowledge, not behaviour. It can place the right paragraph in the context window; it cannot make a model reliably emit a bespoke output schema, adopt a regulated register, or follow a proprietary reasoning procedure. That asymmetry is central to the structural argument in Section VI.

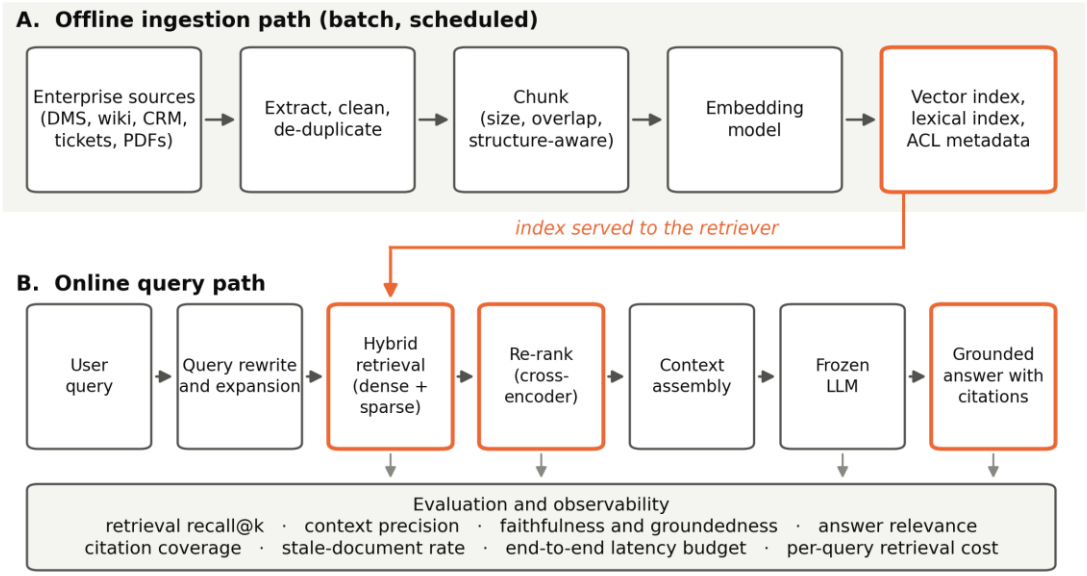

Figure.2. RAG architecture, separated into the offline ingestion path (A) and the per-request online path (B), with the decomposed evaluation metrics that make retrieval failures and generation failures separately diagnosable.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 6, November - December 2024 |

DOI:10.15662/IJEETR.2024.0606030

V. PARAMETER-EFFICIENT FINE-TUNING

I. Mechanism and workflow

Full fine-tuning is beyond most enterprise budgets and yields a full-size artefact per task. Parameter-efficient methods avoid both problems. Adapters insert small trainable modules between frozen layers [38]; low-rank adaptation constrains the update to a low-rank factorisation and adds no inference latency once merged [6]; and QLoRA combines low-rank adaptation with 4-bit quantisation of the frozen base, bringing single-GPU fine-tuning of large models within reach [39].

In an enterprise setting this means supervised fine-tuning with a parameter-efficient method, optionally followed by preference optimisation [13], [14] where subjective quality matters. The workflow (Figure 3) is dominated not by training but by data: behaviour specification, instance harvesting, curation, de-identification, licence clearance and construction of a temporally held-out test set. Evidence that small, carefully curated instruction sets suffice [40] should be read as an argument for curation effort rather than for volume. The binding constraint on enterprise fine-tuning is rarely GPU capacity; it is the availability, quality, licensing status and freshness of task data.

J. Enterprise strengths

Fine-tuning is the only archetype that changes default behaviour rather than instructing around it. Where the requirement is a consistent output schema, a domain register, a compressed proprietary decision procedure, or the removal of a systematic stylistic defect across a long tail of inputs, fine-tuning addresses it at the source. Because the learned behaviour no longer has to be described in the prompt, prompts shorten, which lowers per-request cost and prefill latency. Adapters are small, versionable and swappable, so one base deployment can serve many tasks.

K. Enterprise limitations

Fine-tuning is the weakest of the three mechanisms for the purpose enterprises most often reach for it, namely installing new facts. Because parametric knowledge is fixed at training time, any volatile fact embedded in weights begins decaying immediately, and correcting it requires a retraining cycle rather than an edit.

Adaptation also carries collateral cost. Continual fine-tuning induces catastrophic forgetting of general capability, increasingly so with scale [41]. Low-rank methods sit at a different point on that trade-off: Biderman et al. report that low-rank adaptation learns less than full fine-tuning on the target task but forgets less of the base model's general capability, acting as a regulariser [42]. The enterprise reading is that the standard recipe is comparatively conservative but also comparatively limited in how far it can move behaviour.

Finally, fine-tuning has the least accommodating governance profile of the three. Sensitive training data is embedded irreversibly and may be extractable [43], [44]; an erasure obligation cannot be discharged by deleting a source record; and the resulting artefact requires documented training-data provenance [45]. Its recurring cost is driven by how often retraining is triggered, not by model size.

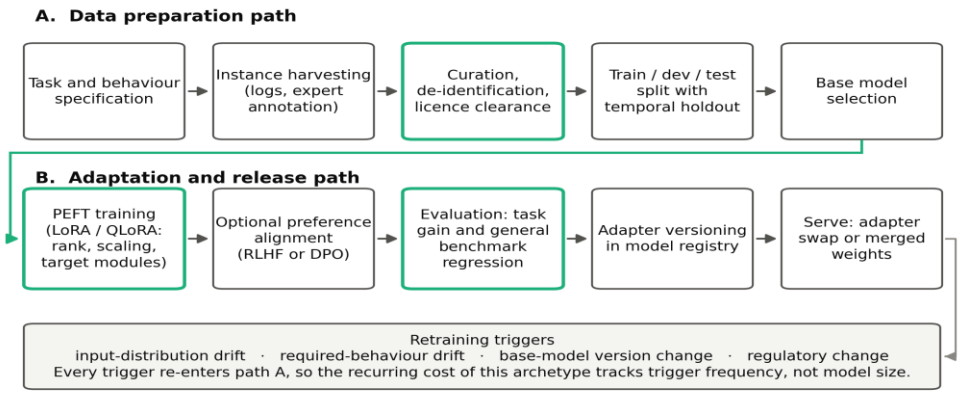

Figure.3. Parameter-efficient fine-tuning as an enterprise workflow, with the data preparation path (A) separated from the adaptation and release path (B). Every retraining trigger re-enters path A, which is where this archetype's recurring cost originates.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 6, November - December 2024 |

DOI:10.15662/IJEETR.2024.0606030

VI. THE STRUCTURAL DISTINCTION

L. One axis, not many

Comparisons of these techniques normally proceed attribute by attribute. We propose instead a single organising distinction, illustrated in Figure 4. The archetypes differ in where the task knowledge required for a correct answer is stored, and in which component is mutable at run time.

- Under **prompting**, knowledge is parametric and the mutable component is the input. Nothing persists between requests.
- Under **RAG**, knowledge is non-parametric and external, and the mutable component is an index. The model is frozen; the corpus is live.
- Under **fine-tuning**, knowledge and behaviour are both parametric, and the mutable component is the weights. Changes require a training cycle.
- Under **hybrid** architectures, behaviour is parametric and knowledge is external, so both components are mutable, but on different cycles.

M. What follows from it

Almost every operational contrast in Table 1 is a consequence of that placement rather than an independent property. Update latency for a corrected fact is minutes under retrieval, because correction is an index write; a full retraining cycle under fine-tuning, because correction means moving weights; and undefined under prompting, because the fact was never installed in the first place.

Citation, per-user entitlement filtering and erasure share a single precondition: they are possible only where knowledge lives outside the model as a discrete, addressable record carrying metadata. A fact diffused across weights has no address to cite, no tag to filter on, and no record to delete. This is why these three capabilities appear together in the RAG column and together absent from the fine-tuning column, and it is why they cannot be added to a fine-tuned model by engineering effort.

Per-request cost and prefill latency are dominated by prompt length, so the archetype that removes instructions from the prompt is cheapest per request and most expensive to build, while the archetype that adds retrieved evidence to the prompt is the reverse.

Maintenance attaches to whichever component is mutable. Prompting accrues prompt-regression obligations, retrieval accrues corpus and index hygiene obligations, and fine-tuning accrues retraining-trigger obligations. These fall on different teams with different skills, which is an under-recognised determinant of which archetype an organisation can actually operate.

Read as a list of independent trade-offs, Table 1 is a set of properties to be weighed one by one. Read as consequences of a single structural choice, it becomes tractable.

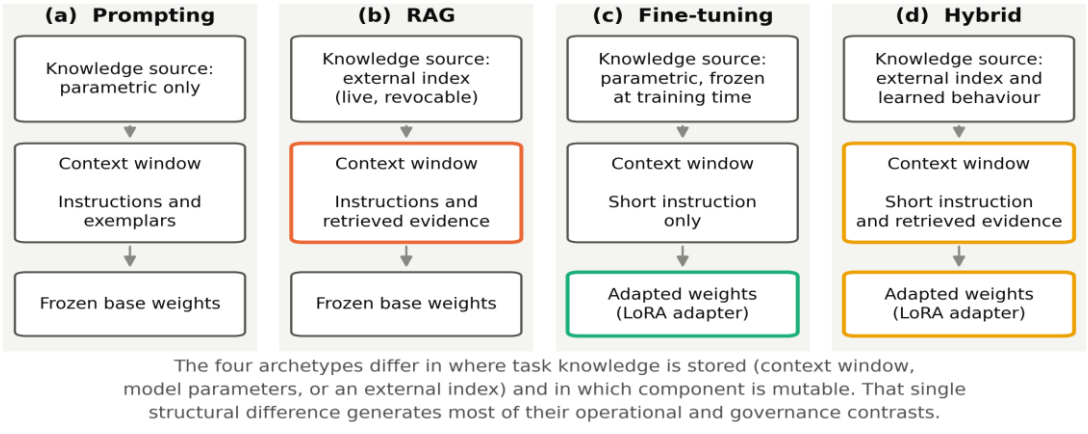

Figure.4. Comparative architecture of the four adaptation archetypes, showing the location of task knowledge and the identity of the mutable component in each

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 6, November - December 2024 |

DOI:10.15662/IJEETR.2024.0606030

TABLE.1. STRUCTURAL AND OPERATIONAL COMPARISON OF THE FOUR ADAPTATION ARCHETYPES ORDINAL ENTRIES ARE RELATIVE TO THE OTHER ARCHETYPES IN THE SAME ROW, NOT ABSOLUTE.

Attribute	Prompt engineering	RAG	Fine-tuning (PEFT)	Hybrid
Knowledge location	Parametric plus prompt	External index	Parametric	External index plus parametric behaviour
Mutable component	Input	Index	Weights	Index and weights
Training data required	None	None (corpus only)	Labelled instances	Labelled instances plus corpus
Update latency for a corrected fact	Not applicable	Minutes to hours	Retraining cycle	Minutes (facts), cycle (behaviour)
Knowledge freshness ceiling	Training cut-off	Index freshness	Training cut-off	Index freshness
Behavioural control	Weak to moderate	Weak	Strong	Strong
Native citation of sources	No	Yes	No	Yes
Per-user entitlement filtering	No	Yes	No	Yes
Data erasure and revocability	Not applicable	Yes	No	Partial
Recurring cost per request	High (long prompt)	High (retrieved context)	Low (short prompt)	Moderate
Maintenance driver	Prompt regression testing	Corpus and index hygiene	Retraining triggers	Both
Principal failure mode	Format and reasoning fragility	Retrieval miss, ignored evidence	Stale facts, forgetting	Compounded pipeline failure
Principal governance weakness	Weak auditability of reasoning	New attack surface	Irreversible data absorption	Largest attestation surface

N. Hybrids are not a midpoint

A common misreading treats hybrid architectures as a compromise between retrieval and fine-tuning. Structurally they are not: they are the union of both, with knowledge external and behaviour parametric. They therefore inherit both maintenance cadences, both failure surfaces and both governance obligations simultaneously. The literature supports the underlying idea, in particular training a generator on retrieval-augmented inputs including distractors so that it learns to use evidence rather than memorise it [46]. The structural point stands regardless: a hybrid is the most, not the least, demanding archetype to operate.

VII. DISCUSSION

Two implications follow for how enterprise teams reason about adaptation.

The first is that the knowledge-versus-behaviour distinction is the load-bearing one, and it is asymmetric. Externalising knowledge is cheap to start and cheap to reverse, because an index can be rebuilt or discarded. Modifying behaviour is expensive to start and difficult to reverse, because weights cannot be selectively unlearned. Treating a move along each axis as an equivalent step is a common planning error.

The second is that governance capability is architectural, not procedural. An organisation cannot add citation, entitlement filtering or erasure to a fine-tuned model through policy or process, because the precondition for all three is that the knowledge sits outside the weights as an addressable record. Where a regulatory or contractual obligation of that kind exists, it therefore constrains the architecture directly rather than being satisfiable downstream. Business cases regularly assume the opposite.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 6, November - December 2024 |

DOI:10.15662/IJEETR.2024.0606030

VIII. LIMITATIONS AND FUTURE WORK

This paper is descriptive and its scope is narrow by design. It establishes a structural vocabulary; it does not measure anything. We do not report comparative accuracy, cost, or latency for the four archetypes, and nothing here should be read as ranking them. Establishing those magnitudes requires a separate quantitative treatment.

Two classes of system fall outside the four-archetype taxonomy rather than merely stressing it. The first is architectures in which retrieval is iterative or agent-directed rather than a single pre-generation step, which the taxonomy does not represent. The second is systems that expend substantial computation at inference time to reason before answering, which breaks the assumption that per-request cost and latency track prompt length and output length. Both are increasingly present in enterprise deployments, and extending the structural account to cover them is the most useful next step.

Finally, the structural claims here are architectural and should be durable, but the surrounding constants are not: token prices, context-window sizes and serving efficiency all move.

IX. CONCLUSION

Prompting, retrieval-augmented generation and fine-tuning are usually introduced as three unrelated techniques with three separate vocabularies, and compared through long tables of apparently independent attributes. This paper has argued that they are better understood as three positions on one structural axis: knowledge may sit in the prompt, in an external index, or in the weights, and correspondingly the input, the index, or the weights is what an organisation is choosing to make mutable. Update latency, citation capability, entitlement enforcement, erasure, per-request cost and maintenance ownership all follow from that placement. Hybrids are the union of two positions rather than a midpoint between them, and inherit both sets of obligations. The distinction is offered not as a ranking but as the vocabulary in which a defensible ranking, and eventually a decision procedure, can be constructed.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems 30, 2017, pp. 5998–6008.
[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., “Language models are few-shot learners,” in Advances in Neural Information Processing Systems 33, 2020, pp. 1877–1901.
[3] R. Bommasani, D. A. Hudson, E. Adeli, et al., “On the opportunities and risks of foundation models,” arXiv preprint arXiv:2108.07258, Aug. 2021.
[4] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, “Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing,” ACM Computing Surveys, vol. 55, no. 9, pp. 1–35, Sep. 2023, doi: 10.1145/3560815.
[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in Advances in Neural Information Processing Systems 33, 2020, pp. 9459–9474.
[6] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-rank adaptation of large language models,” in Proc. Int. Conf. Learning Representations (ICLR), 2022.
[7] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI Tech. Rep., Feb. 2019.
[8] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," arXiv preprint arXiv:2001.08361, Jan. 2020.
[9] J. Hoffmann, S. Borgeaud, A. Mensch, et al., "Training compute-optimal large language models," in Advances in Neural Information Processing Systems 35, 2022, pp. 30016-30030.
[10] H. Touvron, L. Martin, K. Stone, et al., “Llama 2: Open foundation and fine-tuned chat models,” arXiv preprint arXiv:2307.09288, Jul. 2023.
[11] A. Q. Jiang, A. Sablayrolles, A. Mensch, et al., “Mistral 7B,” arXiv preprint arXiv:2310.06825, Oct. 2023.
[12] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, “Finetuned language models are zero-shot learners,” in Proc. Int. Conf. Learning Representations (ICLR), 2022.
[13] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 6, November - December 2024 |

DOI:10.15662/IJEETR.2024.0606030

Lowe, "Training language models to follow instructions with human feedback," in Advances in Neural Information Processing Systems 35, 2022, pp. 27730–27744.

[14] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," in Advances in Neural Information Processing Systems 36, 2023, pp. 53728–53741.

[15] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," ACM Computing Surveys, vol. 55, no. 12, art. 248, pp. 1–38, Dec. 2023, doi: 10.1145/3571730.

[16] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," arXiv preprint arXiv:2311.05232, Nov. 2023.

[17] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer, "Rethinking the role of demonstrations: What makes in-context learning work?," in Proc. 2022 Conf. Empirical Methods in Natural Language Processing (EMNLP), 2022, pp. 11048–11064, doi: 10.18653/v1/2022.emnlp-main.759.

[18] Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate before use: Improving few-shot performance of language models," in Proc. 38th Int. Conf. Machine Learning (ICML), PMLR vol. 139, 2021, pp. 12697–12706.

[19] M. Sclar, Y. Choi, Y. Tsvetkov, and A. Suhr, "Quantifying language models' sensitivity to spurious features in prompt design, or: How I learned to start worrying about prompt formatting," in Proc. Int. Conf. Learning Representations (ICLR), 2024.

[20] S. Chen, S. Wong, L. Chen, and Y. Tian, "Extending context window of large language models via positional interpolation," arXiv preprint arXiv:2306.15595, Jun. 2023.

[21] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," Transactions of the Association for Computational Linguistics, vol. 12, pp. 157–173, 2024, doi: 10.1162/tacl_a_00638.

[22] S. Schulhoff, M. Ilie, N. Balepur, K. Kahadze, A. Liu, C. Si, Y. Li, A. Gupta, H. Han, et al., "The Prompt Report: A systematic survey of prompting techniques," arXiv preprint arXiv:2406.06608, Jun. 2024.

[23] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in Advances in Neural Information Processing Systems 35, 2022, pp. 24824–24837.

[24] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," in Proc. Int. Conf. Learning Representations (ICLR), 2023.

[25] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, "Efficient memory management for large language model serving with PagedAttention," in Proc. 29th ACM Symp. Operating Systems Principles (SOSP), 2023, pp. 611–626, doi: 10.1145/3600006.3613165.

[26] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "Retrieval augmented language model pre-training," in Proc. 37th Int. Conf. Machine Learning (ICML), PMLR vol. 119, 2020, pp. 3929–3938.

[27] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," in Proc. 2020 Conf. Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 6769–6781, doi: 10.18653/v1/2020.emnlp-main.550.

[28] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in Proc. 16th Conf. European Chapter of the Association for Computational Linguistics (EACL), 2021, pp. 874–880, doi: 10.18653/v1/2021.eacl-main.74.

[29] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," arXiv preprint arXiv:2312.10997, Dec. 2023.

[30] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to retrieve, generate, and critique through self-reflection," in Proc. Int. Conf. Learning Representations (ICLR), 2024.

[31] W. Shi, S. Min, M. Yasunaga, M. Seo, R. James, M. Lewis, L. Zettlemoyer, and W.-t. Yih, "REPLUG: Retrieval-augmented black-box language models," in Proc. 2024 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), vol. 1, 2024, pp. 8371–8384, doi: 10.18653/v1/2024.naacl-long.463.

[32] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, "MTEB: Massive text embedding benchmark," in Proc. 17th Conf. European Chapter of the Association for Computational Linguistics (EACL), 2023, pp. 2014–2037, doi: 10.18653/v1/2023.eacl-main.148.

[33] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. 2019 Conf. Empirical Methods in Natural Language Processing and 9th Int. Joint Conf. Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3982–3992, doi: 10.18653/v1/D19-1410.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 6, November - December 2024 |

DOI:10.15662/IJEETR.2024.0606030

[34] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 4, pp. 824–836, Apr. 2020, doi: 10.1109/TPAMI.2018.2889473.
[35] R. Nogueira and K. Cho, "Passage re-ranking with BERT," arXiv preprint arXiv:1901.04085, Jan. 2019.
[36] S. Barnett, S. Kurniawan, S. Thudumu, Z. Brannelly, and M. Abdelrazek, "Seven failure points when engineering a retrieval augmented generation system," in Proc. IEEE/ACM 3rd Int. Conf. AI Engineering: Software Engineering for AI (CAIN), 2024, pp. 194–199, doi: 10.1145/3644815.3644945.
[37] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," in Proc. 16th ACM Workshop on Artificial Intelligence and Security (AISec), 2023, pp. 79–90, doi: 10.1145/3605764.3623985.
[38] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in Proc. 36th Int. Conf. Machine Learning (ICML), PMLR vol. 97, 2019, pp. 2790–2799.
[39] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient finetuning of quantized LLMs," in Advances in Neural Information Processing Systems 36, 2023, pp. 10088–10115.
[40] C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, S. Zhang, G. Ghosh, M. Lewis, L. Zettlemoyer, and O. Levy, "LIMA: Less is more for alignment," in Advances in Neural Information Processing Systems 36, 2023, pp. 55006–55021.
[41] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, and Y. Zhang, "An empirical study of catastrophic forgetting in large language models during continual fine-tuning," arXiv preprint arXiv:2308.08747, Aug. 2023.
[42] D. Biderman, J. Portes, J. J. Gonzalez Ortiz, M. Paul, P. Greengard, C. Jennings, D. King, S. Havens, V. Chiley, J. Frankle, C. Blakeney, and J. P. Cunningham, "LoRA learns less and forgets less," arXiv preprint arXiv:2405.09673, May 2024.
[43] N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, Ú. Erlingsson, A. Oprea, and C. Raffel, "Extracting training data from large language models," in Proc. 30th USENIX Security Symposium, 2021, pp. 2633–2650.
[44] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramèr, and C. Zhang, "Quantifying memorization across neural language models," in Proc. Int. Conf. Learning Representations (ICLR), 2023.
[45] National Institute of Standards and Technology, Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1, Gaithersburg, MD, USA, Jan. 2023, doi: 10.6028/NIST.AI.100-1.
[46] T. Zhang, S. G. Patil, N. Jain, S. Shen, M. Zaharia, I. Stoica, and J. E. Gonzalez, "RAFT: Adapting language model to domain specific RAG," arXiv preprint arXiv:2403.10131, Mar. 2024.