International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)
https://ijcjournal.org/index.php/InternationalJournalOfComputer/index

Paradigms of ELT Data Pipeline Architectures for LLM Training

Olusesan Ogundulu*

Data Engineer at Alvarez & Marsal Holdings,Tampa, FL, United States
Email:oogundulu@alvarezandmarsal.com

Abstract

This article presents a systematic analysis of ELT pipeline architectures used in the training of large language models. The study is based on an interdisciplinary approach that integrates engineering principles of data infrastructure design, theoretical foundations of transformer architectures, and data flow automation practices under conditions of high source variability. Particular attention is given to the content analysis of scientific and applied publications addressing the role of LLMs in transformation loops, the implementation of agent-oriented solutions, and the support of multimodal adaptive pipelines. Key ELT architecture types are identified, including prompt-driven, agent-based, high-throughput, and cognitively enhanced solutions, reflecting varying levels of model involvement in data processing. The analysis shows that architectural shifts toward feedback integration and dynamic routing enable the creation of robust and adaptive solutions suited to contemporary training scenarios. Special emphasis is placed on issues related to data stream instability, the lack of benchmarks for agent-based systems, and insufficient integration of pipelines with model evaluation mechanisms. The paper proposes a conceptual classification of ELT paradigms and an outline of their adaptive evolution toward building scalable and logically coherent infrastructures. The article will be of interest to researchers in machine learning systems, LLM infrastructure developers, data platform architects, and professionals in digitalization and automation of AI training workflows.

Keywords: ELT pipeline; large language models; data architecture; data transformation; agent-oriented systems; LLM infrastructure; multimodal data; dynamic routing; cognitive processing; system scalability.

Received: *9/19/2025*
Accepted: *11/19/2025*
Published: *11/29/2025*

---

* *Corresponding author.*

1.Introduction

The development of machine-learning technologies and the widespread deployment of large language models are transforming architectural approaches to data-processing systems. Against a backdrop of rapidly growing information volumes, high source variability and the need for swift adaptation to changing conditions, traditional ETL (Extract–Transform–Load) architectures—which assume data cleansing and structuring occur prior to loading—are increasingly constrained in terms of flexibility, scalability and compatibility with transformer-based designs [2]. This has driven a shift toward ELT (Extract–Load–Transform) paradigms, in which transformation operations are deferred to the final stage and executed directly within the storage and analytics environment.

The increased computational load associated with training and fine-tuning LLMs imposes specific demands on pipeline architectures. These must ensure resilience to high-frequency requests, support multithreading, adapt to distributed environments and enable flexible data routing. In this context, ELT architectures offer significant advantages, including simplified integration with cloud platforms, accelerated preloading stages and enhanced transformation capabilities tailored to the model's tasks [5]. The move to such architectures is especially pertinent in high-tech and creative industries, where data flow from numerous heterogeneous sources—including APIs, cloud storage, NoSQL databases and streaming services.

Another factor underscoring their relevance is that modern language models require massive data volumes and can actively participate in preliminary processing—normalization, enrichment and filtering [1]. This has given rise to hybrid pipelines in which LLMs serve not merely as training targets but as integral components of the data-processing architecture. Consequently, there is a growing need for theoretical reassessment of classical notions of linear data flow and for developing new approaches to constructing scalable, adaptive ELT systems.

The objective of this study is to conduct a systematic theoretical analysis of ELT-architecture paradigms applied to large language model training, focusing on scalability, alignment with transformer requirements, transfer-learning capabilities and diversity of data sources. Special attention is given to the principles governing extraction, loading and transformation stages and their interrelation with the internal characteristics of the models.

2.Materials and Methods

The methodological foundation of this study is positioned at the nexus of data-pipeline engineering practice, architectural analysis of transformer models and theory-driven investigation of scalable machine-learning infrastructures. The interdisciplinary nature of this task arises from the need to integrate expertise in distributed computing, LLM architectures and intelligent automation practices for data processing amid high source variability and dynamic streaming.

The primary research instrument is a qualitative content analysis of scientific and applied literature addressing the design, classification and performance evaluation of ELT-pipeline architectures employed in large-language-model training. The investigation draws on both foundational reviews and applied studies that explore the integration of LLMs into existing architectural frameworks.

Barbon Junior [1] is examined for its proposal of LLMs as a novel interface layer interacting with data-processing pipelines, thereby reinterpreting the model's role within the transformation loop. Colombo [2] analyses LLM application in constructing a legal knowledge graph, where the model acts not merely as a consumer but as an active participant in the transformation stage. Garcia [3] informs the analysis of human–model interaction during the configuration of data workflows, a consideration critical to hybrid ELT scenarios. Infrastructure-focused investigations by He [4] and Li [6] enable assessment of the elasticity of PipeTransformer and Chimera pipelines under billion-parameter transformer training. Jin's ELT-Bench benchmark [5] for evaluating AI agents is reviewed for its relevance to agent-oriented pipelines. Naveed [7] provides a comprehensive overview of language-model evolution, detailing key architectural transitions that shape data-processing structures. Raza [8] contributes an applied industry perspective that illuminates practical applicability considerations. Xue [9] expands the methodological scope by presenting dynamically interleaved architectures within a multimodal environment, highlighting approaches for unstable and adaptive sources.

Thus, the methodological strategy is grounded in a comparative theoretical analysis of architectural models, ELT-pipeline design criteria, LLM usage scenarios and scalability requirements. This approach has revealed structural distinctions between classical and adaptive ELT paradigms and laid a reasoned groundwork for further classification.

Previous research provides valuable but fragmented insights into ELT-related processes. He and Li focus mainly on throughput optimisation for large-scale transformer training, offering elastic and bidirectional pipelines but without addressing cognitive integration or feedback-driven adaptation. Colombo proposes an LLM-assisted ETL approach restricted to constructing a legal knowledge graph, demonstrating model participation but within a narrowly defined scenario. Barbon Junior and Garcia discuss human–model interaction in post-load transformation, yet these works do not attempt to unify architectural principles across different ELT paradigms. Jin introduces an agent-based benchmark, but the testing scope remains limited to routing and strategy selection rather than complete pipeline coherence. Xue's multimodal dynamic-interleaving approach addresses stream instability but focuses primarily on multimodal environments. Compared to these studies, the present work contributes by offering a consolidated architectural classification that spans infrastructural, adaptive, agent-driven and cognitively enriched ELT pipelines, establishing a broader theoretical foundation for LLM-oriented data-transformation research.

Several limitations should be acknowledged. First, this research is based primarily on literature analysis, without empirical validation of architectural behaviours under real-world high-variance conditions. Second, the terminology and evaluation methods used in the reviewed studies are heterogeneous, which restricts direct comparability across pipelines and limits the precision of the proposed typology. Third, agent-driven and cognitively enhanced ELT architectures remain insufficiently formalised in current research, complicating the assessment of routing reliability, robustness and long-term scalability. Finally, the study does not include controlled performance measurements, leaving latency, resilience under multimodal bursts and synchronisation overhead outside the empirical scope. These constraints outline directions for future work focused on reproducible evaluation protocols and unified feedback-aware ELT frameworks.

3.Results

In the current landscape of large language model development, a systemic shift is occurring toward reimagining the data-pipeline architectures employed in training. A principal trend in this transformation is the move from classical linear ETL schemes to more flexible, adaptive and cognitively enriched ELT architectures. The variety of technological solutions and methodological approaches necessitates their typological systematization across a set of critically important parameters. Table 1 presents an author's classification of typical ELT-architecture implementations, derived from a thematic synthesis of existing scholarly and applied practices.

Table 1 : Architectural Parameters of Typical ELT Implementations for LLM Training (Compiled by the author based on sources: [2,4,5,8])

Architecture Type	**Pipeline Objective**	**LLM Involvement**	**Scalability**	**Dynamism**	**Data Source Types**
Unified Semantic	Consolidation and normalization of heterogeneous data	Semantic filtering and normalization	Low	Low	APIs, text logs, CSV, JSON
Post-Load Analytical	Storage and subsequent transformation	Enrichment and verification	Medium	Medium	Relational databases, regulatory repositories
Agent-Driven	Dynamic routing and transformation	Strategy selection and transformation logic	Medium	High	Streaming and event-based sources
Distributed Parallel	High-performance model training	Not involved	High	Medium	Tensors, batches, cloud-based datasets
Multimodal Adaptive	Processing of unstructured and unstable data	Direct participation in transformation	High	Very high	Video, audio, sensors, texts, APIs

As shown in Table 1, ELT architecture types differ in technological maturity and the distribution of responsibilities among components. Unified Semantic pipelines are generally static, focusing on the harmonization of diverse sources within a common storage schema. Post-Load Analytical solutions emphasize deferred transformation and integration with the LLM at the analysis stage. Agent-Driven and Multimodal Adaptive pipelines are the most flexible: here, the LLM not only consumes data but also actively participates in decisions about the nature,

sequence and necessity of transformations [8]. This trend underscores the model's emerging role as a cognitive node that engages in a feedback loop with the architecture. Systematizing these implementations establishes a foundation for evaluating architectural effectiveness relative to specific training scenarios. Furthermore, this classification highlights the imperative to design scalable, adaptive solutions tailored to the multimodal and dynamic data environments characteristic of modern LLM training processes.

Development of ELT-pipeline paradigms for training large language models is driven by the need to ensure stable, adaptive and logically coherent data processing from diverse sources. In this study, five key ELT approaches have been typologized according to the functional role of the LLM within the architecture and the nature of the model's interaction with transformation stages.

Prompt-oriented ELT is characterized by the large language model orchestrating data transformations via prompts integrated into API interfaces. Here, the LLM acts as both interpreter and router, actively shaping the logic of data flows. This approach is examined in Barbon Junior [1] and Garcia [3], where the model performs filtering, enrichment and unification of data in the post-load phase. LLM-assisted ETL represents an intermediary scenario in which the language model does not govern the entire pipeline but functions as an auxiliary component handling specialized tasks—such as entity matching or relation extraction. Colombo [2] analyzes this scenario in the context of constructing a legal-entity knowledge graph. LLM-as-Agent ELT employs the model as an active agent that decides on dynamic data routing, selects transformation functions and determines their execution order. Jin [5] demonstrates how such agent-based architectures adapt to varying processing scenarios and enhance consistency between extraction and transformation stages. High-Throughput ELT is an infrastructure-oriented approach focused on delivering high throughput and scalability during transformer training. Key mechanisms include parallel loading, distributed processing and multi-stage data preparation; this paradigm is detailed by He [4] and Li [6] in their descriptions of the PipeTransformer and Chimera pipelines, optimized for billion-parameter models. Multimodal Adaptive ELT represents the most flexible form, targeting unstable, evolving and multimodal data streams. In this case, the model controls both transformation logic and the structure of the stream itself, dynamically adapting to changes in input sources. Xue [9] presents this approach through the PipeWeaver architecture for multimodal training.

This classification highlights the core structural principles that govern ELT-system behavior and determine their suitability for different LLM-training scenarios. It provides a foundation for designing adaptive, scalable and logically coherent pipelines within transformer-based computations. A crucial component of any ELT architecture is its method of interacting with data sources. Variations in format, structure and update frequency impose distinct requirements on scalability, stability and flexibility. Table 2 categorizes the main source types encountered in contemporary ELT scenarios, along with typical applications.

Table 2 : Overview of common data source categories, representative sources, and their real-world applications (Source: [5])

Data Source Category	**Representative Sources**	**Applications in Practice**
APIs	REST API	Web services, third-party platforms, real-time integrations
Cloud Services	Amazon S3	Big data platforms, scalable storage, analytics pipelines
Relational Databases	PostgreSQL	Traditional enterprise systems, transactional business logic
NoSQL Databases	MongoDB	Real-time apps, social platforms, adaptive data-driven systems
Flat Files	CSV, JSONL, Parquet	Offline analytics, backups, model retraining datasets

As shown in Table 2, the type of source directly influences ELT-pipeline behavior. For example, engaging with an API requires low-latency routing and dynamic serialization of data, whereas processing flat files is suited to batch-oriented operations. Working with cloud storage and NoSQL infrastructures demands support for asynchronous loading and flexible transformation logic, especially when handling unstable data streams characteristic of multimodal training.

4.Discussion

The diversity of existing ELT approaches for training large language models stems from a combination of architectural and functional considerations. The choice of a specific pipeline architecture depends on the model's characteristics, data-throughput requirements, source types and the infrastructure's ability to adapt to dynamic, high-load computational scenarios. A central concern is the alignment between the transformer's computational logic and the structure of the data stream ingested from external sources.

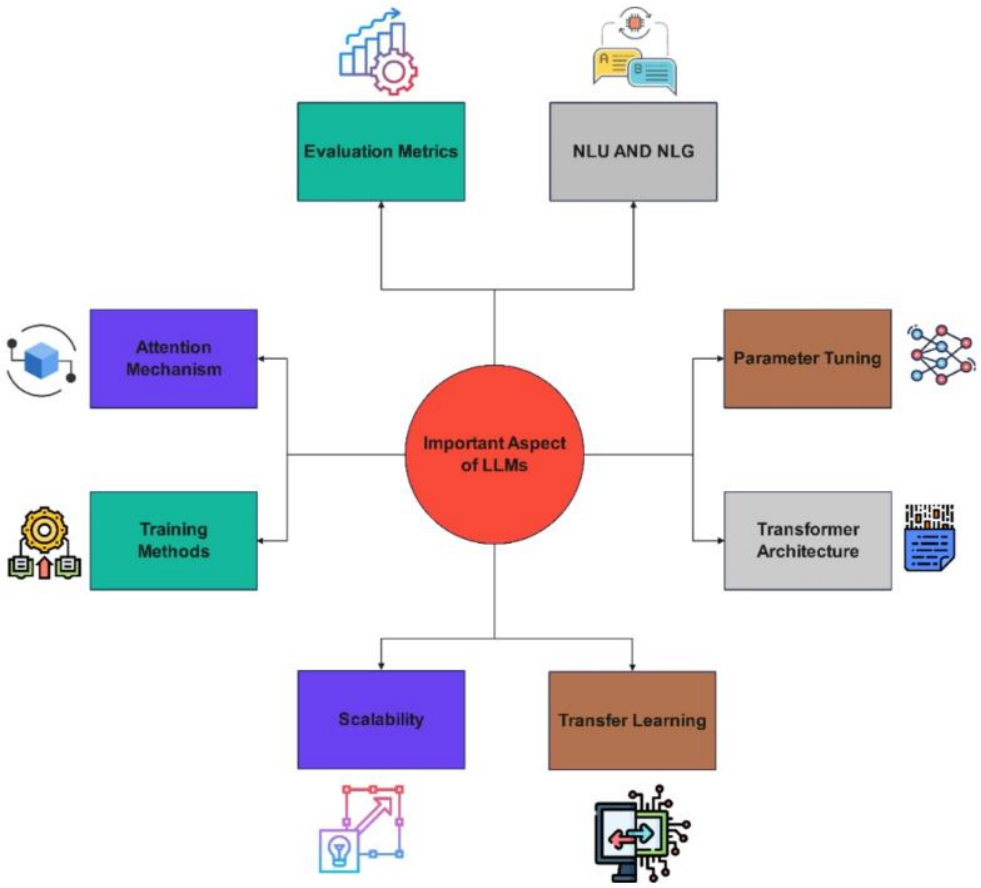

Figure 1 : Important Aspects of LLMs (Source: [8])

These aspects define the requirements to which an ELT pipeline—serving as the link between the model and its external data environment—must conform. One such requirement is a multi-stage data-preparation phase, during which data undergo cleansing, format alignment and structural unification before being fed into the model. In transformer architectures, this phase is especially critical, as attention mechanisms are sensitive to inconsistencies in input sequences. ELT approaches that omit formalized data preparation risk breaking the coherence between loading and transformation logic. Another key functional component is the data-prefetch mechanism—organizing pre-loading and buffering of data prior to model ingestion [7]. This mechanism helps prevent training stalls and reduces system sensitivity to unstable sources. Prefetching is most effective in architectures featuring parallel loading and distributed transformation.

Functional differences among ELT approaches also manifest in the degree of LLM involvement in data governance. In architectures where the model acts as an agent, it may initiate filtering, determine transformation routes and control the data-flow structure. In other cases, the LLM functions as a passive consumer, without influencing the preprocessing logic [4].

One of the key challenges documented in contemporary literature is the high volatility of data sources, which is especially characteristic of multimodal, streaming and event-driven systems. The unpredictability of structure, update frequency and input-stream formats undermines the integrity of the loading and transformation stages

within ELT architectures. Xue [9] proposes an approach that implements a mechanism of dynamically interleaved transformation, enabling adaptation to evolving input-data structures. However, despite its technical soundness, this mechanism has not yet seen widespread adoption, and its implementation is constrained by the complexity of integration with distributed architectures. The absence of a standard solution capable of ensuring resilience to such sources remains a critical limitation in designing ELT for LLMs.

Equally important is aligning the pipeline architecture with the feedback loops generated by the language model during training. At present, most ELT systems operate as isolated data-preparation layers, without interacting with the model's output-quality evaluation logic. This separation prevents the creation of closed training loops that can account for accuracy, stability and the behavioural specificity of generation. Raza [8] emphasises the need to move toward architectures in which the data-transformation pipeline adapts to metrics returned by the model itself—such as relevance, correctness and factual accuracy.

A further problem is the lack of formalised testing scenarios and quality-evaluation frameworks for agent-driven ELT architectures. Jin [5] introduces the ELT-Bench benchmark for evaluating systems with agent-managed transformation, yet even this effort reveals a shortage of comprehensive metrics that reflect performance, reliability, routing correctness and resilience to flawed strategies. As a result, rigorous comparison among different architectures remains difficult, slowing the uptake of agent-oriented ELT in practical settings.

Looking ahead, the greatest potential lies in architectures capable of automatically configuring their processing structure and logic according to model behaviour and input-stream characteristics. Theoretical foundations for this approach appear in Li [6], which examines a bidirectional pipeline organisation allowing adaptive reconfiguration of transformation modules. Promising directions also include distributed transformation (a federated approach) and the reuse of trained pipelines in transfer-learning tasks.

The comparative analysis of ELT paradigms indicates that their effectiveness is highly dependent on the degree of model integration into the transformation loop. Architectures with minimal LLM involvement demonstrate predictable and stable performance but lack adaptability when interacting with heterogeneous or volatile data streams. Conversely, pipelines that rely on agentic or cognitively enriched mechanisms introduce dynamic routing and modelling capabilities but simultaneously increase architectural complexity and sensitivity to feedback-loop instability.

Another important clarification concerns the trade-offs between throughput and flexibility. High-throughput and distributed pipelines excel in large-scale training scenarios but impose strict format constraints and limit the use of adaptive transformation components. In contrast, multimodal and agent-driven architectures support unstable or rapidly evolving streams but demand sophisticated synchronisation, monitoring and error-handling policies. These trade-offs define the practical boundaries for selecting an ELT paradigm for a specific LLM-training environment.

Finally, the findings highlight that ELT pipelines cannot be evaluated solely by throughput or latency—architectural coherence, cognitive alignment of transformation logic and support for model-driven feedback loops

become equally critical. This reinforces the need for integrated design methodologies that treat extraction, loading, transformation and evaluation as a single architectural continuum rather than isolated stages.

The analysis of these challenges confirms the need to shift from isolated solutions to new-generation architectures. Sustainable ELT development requires systems in which data preparation, model training and evaluation are unified within a coherent, dynamically configurable framework.

5.Conclusion

The study systematically examined architectural transformations in data preparation for training large language models, highlighting the paradigm shift from static ETL scenarios to adaptive ELT solutions that incorporate the model itself into the transformation loop. It was determined that involving the LLM in data routing and transformation enhances the pipeline's cognitive coherence, increases its resilience to unstable sources and enables the construction of dynamically configurable systems.

The identified architecture types—ranging from high-throughput infrastructure solutions to multimodal cognitive pipelines—demonstrate that ELT effectiveness is governed by stream characteristics, the model's agency, transformation logic and the ability to form closed-loop feedback. Of particular importance is the pipeline's capacity to adapt to model behavior, generation-quality metrics and training parameters, which calls for rethinking traditional boundaries between training, analysis and preprocessing stages.

The necessity of transitioning to new-generation architectures is substantiated, where transformation mechanisms adjust to incoming streams while simultaneously accounting for internal evaluation and tuning loops. Such systems can serve as the foundation for self-regulating LLM platforms capable of sustainable operation amid high data variability and computational constraints.

Thus, the architectural typology proposed in this work—together with principles of structural adaptability and functional model integration into the ELT loop—lays the groundwork for developing scalable, agent-oriented and intelligently managed data-preparation systems for transformer training. Promising future directions include the development of formalized feedback protocols, the implementation of evaluation-adaptive pipelines and the move toward distributed cognitive architectures operating in real time.

References

[1]. Barbon Junior, S., Ceravolo, P., Groppe, S., Jarrar, M., Maghool, S., Sèdes, F., Sahri, S., & van Keulen, M. (2024). Are large language models the new interface for data pipelines? (arXiv:2406.06596). arXiv. https://doi.org/10.48550/arXiv.2406.06596

[2]. Colombo, A., Bernasconi, A., & Ceri, S. (2025). An LLM-assisted ETL pipeline to build a high-quality knowledge graph of Italian legislation. Information Processing & Management, 62(4), 104082. https://doi.org/10.1016/j.ipm.2025.104082

[3]. Garcia, A. A., Candello, H., Badillo-Urquiola, K., & Wong-Villacres, M. (2025). Emerging data practices: Data work in the era of large language models. In CHI '25: Proceedings of the 2025 CHI

Conference on Human Factors in Computing Systems (Article 846, pp. 1–21). Association for Computing Machinery. https://doi.org/10.1145/3706598.3714069

[4]. He, C., Li, S., Soltanolkotabi, M., & Avestimehr, S. (2021). PipeTransformer: Automated elastic pipelining for distributed training of transformers (arXiv:2102.03161). arXiv. https://doi.org/10.48550/arXiv.2102.03161

[5]. Jin, T., Zhu, Y., & Kang, D. (2025). ELT-Bench: An end-to-end benchmark for evaluating AI agents on ELT pipelines (arXiv:2504.04808v2). arXiv. https://doi.org/10.48550/arXiv.2504.04808

[6]. Li, S., & Hoefler, T. (2021). Chimera: Efficiently training large-scale neural networks with bidirectional pipelines (arXiv:2107.06925). arXiv. https://doi.org/10.48550/arXiv.2107.06925

[7]. Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2024). A comprehensive overview of large language models (arXiv:2307.06435). arXiv. https://doi.org/10.48550/arXiv.2307.06435

[8]. Raza, M., Jahangir, Z., Riaz, M. B., & others. (2025). Industrial applications of large language models. Scientific Reports, 15, 13755. https://doi.org/10.1038/s41598-025-98483-1

[9]. Xue, Z., Hu, H., Chen, X., Jiang, Y., Song, Y., Mi, Z., Zhu, Y., Jiang, D., Xia, Y., & Chen, H. (2025). PipeWeaver: Addressing data dynamicity in large multimodal model training with dynamic interleaved pipeline (arXiv:2504.14145). arXiv. https://doi.org/10.48550/arXiv.2504.14145