# HuggingGraph: Understanding the Supply Chain of LLM Ecosystem

Mohammad Shahedur Rahman
University of Texas at Arlington

Peng Gao
Virginia Tech

Yuede Ji
University of Texas at Arlington

## Abstract

Large language models (LLMs) leverage deep learning architectures to process and predict sequences of words based on context, enabling them to perform a wide range of natural language processing tasks, such as translation, summarization, question answering, and content generation. However, the increasing size and complexity of developing, training, and deploying cutting-edge LLMs demand extensive computational resources and large-scale datasets. This creates a significant barrier for researchers and practitioners. Because of that, platforms that host models and datasets have gained widespread popularity. For example, on one of the most popular platforms, i.e., Hugging Face, there are more than 1.8 million models and more than 450K datasets by the end of Jun. 2025, and the trend does not show any slowdown.

As existing LLMs are often built from base models or other pre-trained models and using external datasets, they can inevitably inherit vulnerabilities, biases, or malicious components that exist in previous models or datasets. Therefore, it is critical to understand these components' origin and development process to detect potential risks better, improve model fairness, and ensure compliance with regulatory frameworks. Motivated by that, this project aims to study such relationships between models and datasets, which are the central parts of the *LLM supply chain*. First, we design a methodology to collect LLMs' supply chain information systematically. With the collected information, we design a new graph to model the relationships between models and datasets, which is a large directed heterogeneous graph, with *397,376 nodes* and *453,469 edges*.

Then, on top of this graph, we perform different types of analysis and make multiple interesting findings, such as (i) the LLM supply chain graph is large, sparse, and exhibits a power-law degree distribution; (ii) it features a densely connected core and a fragmented periphery; (iii) datasets are pivotal, playing critical roles in training; (iv) a strong interdependence exists between models and datasets; (v) the graph is dynamic, with daily updates capturing the ecosystem's continuous evolution.



*Conference'17, Washington, DC, USA*



ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

**ACM Reference Format:**
Mohammad Shahedur Rahman, Peng Gao, and Yuede Ji. 2025. HuggingGraph: Understanding the Supply Chain of LLM Ecosystem. In . ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 Introduction

Large language models (LLMs) are AI models designed to understand and generate human language by learning patterns and relationships within extensive datasets [30, 36], such as GPT (Generative Pre-trained Transformer) [48], BERT (Bidirectional Encoder Representations from Transformers) [21], and T5 (Text-To-Text Transfer Transformer) [5]. These models leverage deep learning architectures to process and predict sequences of words based on context, enabling them to perform a wide range of natural language processing tasks [6] such as translation [27], summarization [23], question-answering [3], and content generation [1]. LLMs usually have billions (or even trillions) of parameters [28], enabling them to generate high-quality text.

However, the increasing size and complexity of developing, training, and deploying cutting-edge LLMs demand extensive computational resources [46] and large-scale datasets [43]. This creates a significant barrier for researchers and practitioners, limiting their access to state-of-the-art models [31]. As the demand for democratizing access to such LLM models continues to rise, platforms that host models and datasets have gained widespread popularity. For example, Figure 1 shows the number of models and datasets (in a million scales) on Hugging Face, a popular AI model hosting platform [13], starting from July 31st, 2024, to Jun. 30th, 2025. By the end of June 2025, it has reached over 1.8M models and 450K datasets. In addition, the trend does not show any slowdown. Such LLM hosting platforms provide user-friendly interfaces, APIs, and cloud-based infrastructures that enable researchers and developers to easily share, fine-tune, and deploy models without requiring extensive computational resources. Moreover, they foster open collaboration, allowing the broader community to contribute to model improvements, benchmark performances, and enhance transparency in AI research.

On such platforms, different types of models can be classified into two categories based on their tasks, i.e., *base models* and *task-specific models*. (i) base models are large, pre-trained models that can be fine-tuned for specific downstream tasks [40]. They are usually trained on vast datasets and are general-purpose, such as GPT [48], BERT [21], and T5 [5]. (ii) Task-specific models are modified versions of base models for a

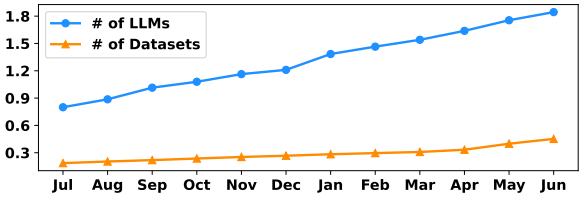


**Figure 1: The number of models and datasets (in millions) on Hugging Face from July 31, 2024, to June 30, 2025.**

specific task. Taking Hugging Face as an example, there are four types of such models. First, the *fine-tuned models* adapt base models for specific tasks by training on additional task-specific datasets [49]. Second, the *adapters models* add lightweight and modular layers to the pre-trained models for specific tasks [17]. Third, the *quantization models* trade off the precision in numerical computations for accelerating inference and reducing memory consumption (e.g., using less precise model parameters) [42]. Fourth, the *merge models* integrate multiple specialized or fine-tuned parameter sets into a single unified model by combining weights or configurations (e.g., via weighted averaging or parameter stitching), enabling support for multiple tasks or domains without separate deployments [2]. Besides models, such platforms also host many *datasets* used for training and fine-tuning the previously discussed models [35].

## 1.1 Motivation

As existing LLMs are often built from base models or other pre-trained models and using external datasets, they can inevitably inherit vulnerabilities, biases, or malicious components that exist in previous models or datasets. Therefore, understanding these components' origin and development process can help better detect potential risks, improve model fairness, and ensure compliance with regulatory frameworks.

Motivated by that, this project aims to study such relationships between models and datasets. They are the central parts of the *LLM supply chain* [45], which refers to the entire lifecycle of developing, training, and deploying LLMs, similar to a traditional supply chain in manufacturing or software development [4, 8, 39, 50]. The LLM supply chain can help to identify critical insights for both model evolution and dataset origin, as discussed below.

**Model evolution.** The study of the supply chain of LLMs gives a clear overview of how LLMs evolve from base models to fine-tuned variants, adapter integration, quantization models, and merged models that combine multiple specialized parameter sets into a single release. With that, one can easily keep track of them. For example, a use case is when a security or bias risk is found in one LLM, and we can quickly locate the potential models that might have the same problems.

**Dataset origin.** This supply chain can help to understand the datasets' origins used for training different models [36]. Dataset origin refers to the source from which the data is collected. For example, for a fine-tuned model, we not only care about which dataset is used for fine-tuning but also what other datasets are involved in training the previous model. Understanding such dataset origin helps to ensure that the dataset used is reliable, ethical, legally compliant, and relevant to the task.

## 1.2 Contribution

Our contributions are mainly threefold. First, we design a methodology to systematically collect the supply chain information of LLMs. In this paper, we mainly study the most popular ML platform, i.e., Hugging Face, but the same strategy also applies to other LLM platforms. In particular, we use the APIs from the ML platform to collect the metadata about the hosted model and dataset. To that end, we collected a large dataset, covering all the **models (1.8M)** and **datasets (450K)** as of *June 30, 2025*.

Second, with the collected metadata, we designed a new graph, named ***LLM supply chain graph***, to model the relationships between models and datasets. It is a directed heterogeneous graph where a node denotes different types of datasets and models (including base, fine-tune, adapter, quantization, and merge models). An edge denotes the dependency relationship between them, including dataset-dataset, and model-dataset relationships. Together, this graph is able to accurately capture the LLM supply chain information. To this end, we constructed a large graph with **397,376 nodes** and **453,469 edges**. An anonymized version of the complete graph is publicly available at GitHub[1].

Third, with this graph, we perform different types of analysis, including model-dataset relationship analysis. We answer five research questions, including (i) the properties of the LLM supply chain graph, (ii) the structural analysis, (iii) the supply chain relationships between datasets, (iv) the supply chain relationships between the models and datasets, and (v) $\Delta$-based temporal update and tracking.

## 2 Preliminary

**The LLM supply chain** encompasses the interconnected processes required for developing, deploying, and maintaining models [45]. This includes sourcing and preparing data to ensure high-quality and diverse datasets [45]. It also involves creating and training models [24]. Finally, it covers making trained models available through APIs [38]. In addition, LLMs can undergo adaptation, quantization, and fine-tuning, a process where they are tuned with domain-specific datasets to maximize performance on specific tasks [44], thus improving their accuracy and applicability.

Note that this study mainly focuses on the relationships between various types of models and datasets, which are the central parts of the whole LLM supply chain ecosystem. *We hope this study can not only provide insights on the relationships between LLM models and datasets but also raise awareness and future research interests in the LLM supply chain ecosystem.*

[1]Anonymized LLM supply chain graph: https://github.com/huggingface00/HuggingGraph

**Table 1: The APIs used to extract data from Hugging Face.**

| API Name | Description |
|---|---|
| Model hub API | Access the model hub to list, search, and download models and metadata. |
| Dataset API | Access the datasets for discovery, metadata retrieval, and downloading. |
| Metrics API | Access metrics for model evaluation, e.g., metric discovery, metadata retrieval, and calculation. |
| Search API | Search the name, tag, or other metadata for models and datasets. |

## 3 Methodology

This section discusses the main methodologies, including data collection, LLM supply chain graph construction, and analysis.

### 3.1 LLM Supply Chain Information Collection

To analyze the LLM supply chain ecosystem, we need a large dataset with such information. Fortunately, platforms like Hugging Face provide some APIs that allow us to access the model and dataset and collect their metadata information, which can be used to construct the supply chain ecosystem.

Table 1 summarizes four types of APIs we used. In particular, (i) *the model hub APIs* allow one to access the hub of existing models, including searching and downloading the model and its metadata. (ii) *The dataset APIs* allow one to access the datasets for discovery, metadata retrieval, and downloading. (iii) *The metrics APIs* allow one to access the metrics for model evaluation, including metric discovery, metadata retrieval, and calculation. (iv) In addition, one can use *search APIs* to search the name, tag, or other metadata for models and datasets.

With the help of these APIs, we are able to collect the metadata of the hosted models and datasets. In particular, we collected a large dataset, covering the models (330K) and datasets (66K) as of *June 30, 2025*.

**Handling missing information.** Accurate construction of the LLM supply chain graph critically depends on the quality of metadata from the LLM platforms (e.g., Hugging Face), which might suffer from missing or incomplete data. To address this limitation, we design the following two techniques, i.e., cross-reference links and textual pattern extraction.

*(i) Cross-reference links.* The model or dataset description could miss the supply chain data fields for API queries, which could be embedded within statically or dynamically rendered HTML pages. In this example URL https://huggingface.co/models?other=base_model:finetune:meta-llama/Meta-Llama-3-8B, one can tell the model "Meta-Llama-3-8B" is fine-tuned from the model in the previous webpage. To capture such information, we cross-reference the links of the filtered model listing webpages, extract model identifiers, enable the reconstruction of supply chain graph edges, and recursively trace model lineage from the leaf node. This scraping step complements API-based extraction and is only employed when reverse dependency data is otherwise inaccessible.

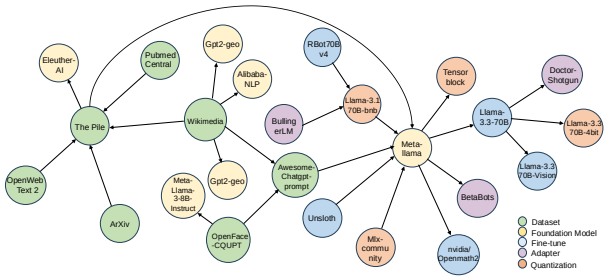


**Figure 2: An example subgraph centering on the base model "`Meta-llama`" from the complete LLM supply chain graph.**

*(ii) Textual pattern extraction.* When structured metadata is absent, the model and dataset cards might mention dependencies in unstructured text descriptions. To capture that, we employ a named entity recognition (NER) method [25, 41] to extract the dependency relationships from the text. For example, the textual phrase like "fine-tuned from Llama-2" contains the *fine-tuned* keyword, which implies from which model this model is actually fine-tuned. Similarly, we also look into other words, such as "train", and "adapt".

### 3.2 LLM Supply Chain Graph

With the collected metadata, we construct a directed heterogeneous graph to model the LLM supply chain. In this graph, a node denotes different types of datasets and models, including base, fine-tuned, adapter, and quantization models.

An edge denotes the dependency relationship between them, including model-model, dataset-dataset, and model-dataset relationships. Figure 2 shows a simplified supply chain subgraph centering on the base model *Meta-llama*.

*(i) Dataset-dataset relationship.* The datasets within the LLM supply chain might overlap, build upon, or extend from each other–e.g., a dataset may be a subset or modified version of another. To capture such information, we connect them with two types of edges. *(1) Subset relationship.* These arise when a dataset is explicitly documented as a subsample or partition of another. For instance, a dataset named "C4_200M" is described as a subset of "C4". *(2) Modified version.* These represent updates or enhanced variants of existing datasets. For example, "TruthfulQA_v2" incorporates factual corrections and improvements over an earlier version, "TruthfulQA_v1".

*(ii) Model-dataset relationship.* To capture the model and dataset dependency, we use the metadata information from both models and datasets. From the metadata of a model, it might include the datasets used to train it. However, not all the models disclose such information. To find more information, we observe the metadata of a dataset, including a data field on "trained_fine_tune_models" on this dataset. Thanks to that, we are able to capture the accurate model and dataset relationship. In Figure 2, the datasets *The Pile* and *Awesome-Chatgpt-prompt*

have directed edges to model *Meta-Llama*, meaning that both datasets are used to train the model.

### 3.3 Supply Chain Graph Analysis

This supply chain graph can help to understand the transformational processes of the models and datasets. In particular, we can understand how base models evolve into their variants, including fine-tuned, adaptive, and quantized models, and vice versa. Similar observations can be made for datasets. This would provide a clear view of how the base model (or dataset) is transformed for performing a particular task. In particular, we mainly perform two types of analysis, i.e., forward and backward analysis.

*Dataset analysis example.* For the dataset, our supply chain analysis shows how different datasets connect and form a new dataset and work together. This combination creates flexible resources that improve how models perform in various areas. In Figure 2, the dataset *The Pile* is composed of multiple subsets, including *Wikimedia*, *Arxiv*, *Openwebtext2*, and *Pubmed Central*. Together, these datasets form a unified corpus that serves as training data for models like *Meta-LlaMA*.

*Model and dataset analysis example.* In Figure 2, to analyze the backward supply chain of the model *RBot70Bv4*, we trace its lineage back through its development stages. This model is fine-tuned from *Unsloth*, which in turn originates from its base model, *Meta-Llama*, and the datasets used to train the base model are *The Pile* and *Awesome-Chatgpt-prompts*. Through this analysis, we establish the backward path, starting from the target model—*RBot70Bv4*—and tracing back to its base model, *Meta-Llama*, revealing the dependencies and transformations involved in its development. Similarly, we can analyze the datasets.

### 3.4 Temporal Update and Tracking

To ensure HuggingGraph remains an accurate and evolving representation of the LLM ecosystem, we have implemented a (i) Δ-based batch update mechanism and (ii) a versioned archival system.

**Δ-based graph update.** To ensure HuggingGraph reflects the evolving nature of the LLM ecosystem, we implement a Δ*-based batch update mechanism*. Instead of reconstructing the graph from scratch, we compute incremental updates by comparing successive versions. Let $G_t$ denote the graph at time $t$, and define the Δ as $\Delta_t = G_t - G_{t-1}$, where $\Delta_t$ captures all added or removed nodes and edges. This approach offers both computational scalability and storage efficiency. At scheduled intervals (e.g., daily), we collect metadata from Hugging Face and compute the differences. For instance, between June 25–July 15, 2025, the platform added 104,380 new models (~3,866/day), 20,034 new datasets (~742/day), and approximately 854 dependency edges/day.

**Versioning and longitudinal tracking.** To enable temporal and historical analyses, each update generates a versioned snapshot of the graph, archived with a timestamp and associated change log. This log captures additions, deletions, and dependency modifications, such as new edges from *TruthfulQA v2* to *TruthfulQA v1*. This ensures that HuggingGraph remains a living resource that adapts to ongoing developments in LLM research and deployment.

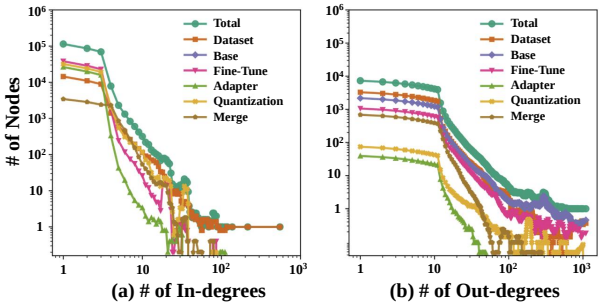


**Figure 3: (a) Indegree distribution and (b) Outdegree distribution. The X-axis represents the number of indegrees and outdegrees, respectively, and the Y-axis represents the number of vertices in a logarithmic scale.**

## 4 Experiment and Finding

To deeply understand the relationships between models and datasets, we aim to answer seven critical research questions (RQs). We believe they can offer valuable insights into the supply chain of the LLM ecosystem and potentially pave the way for future LLM development and deployment.

- **RQ #1**: What are the properties of the LLM supply chain graph?
- **RQ #2**: What structural patterns emerge from the LLM ecosystem?
- **RQ #3**: What are the supply chain relationships between datasets?
- **RQ #4**: What are the relationships between models and datasets?
- **RQ #5**: What insights can be gained as the graph temporally updating?

### 4.1 RQ #1: Supply Chain Graph Properties

This research question aims to understand the critical properties of the LLM supply chain graph, i.e., graph basics and degree distribution.

**Graph basics.** The collected supply chain graph is a large directed heterogeneous graph with **397,376 nodes** and **453,469 edges** as of *June 30th, 2025*. In particular, there are six different types of nodes, including 25,717 base models, 115,103 fine-tuned, 79,240 adapters, 98,125 quantization models, 13,027 merges and 66,164 datasets. The average degree is about 1.14, denoting that it is a very sparse graph.

**The degree distribution** of a graph describes how node degrees (the number of edges connected to a node) are distributed across the graph. Figure 3(a) and (b) illustrate the indegree and outdegree distribution of the graph, respectively. We show not only the total distribution but also the distribution of five types

of nodes, including base models, fine-tuned models, adapter models, quantization models, and datasets. We made two interesting observations.

(i) This degree distribution in our supply chain graph shows a heavy-tailed behavior. In particular, the indegree distribution shows a large spread across different categories. The outdegree distribution follows a similar pattern but may differ in specific cases (e.g., adapters seem to have a more restricted degree distribution). The **power-law behavior** suggests that most nodes have low degrees, while a few central nodes (hubs) dominate the graph. In particular, the model *images* from macrocosmos has the highest indegree value of 550, and *Mistral-7B-v0.1* from mistralai has the highest outdegree value of 1,093. Specifically, the base models act as high-degree hub nodes as they are heavily used by other task-specific models. To this end, we can conclude that this graph is a **power-law graph**, also known as a scale-free graph. That is, a graph whose degree distribution follows a power law distribution. In other words, a small number of nodes have many connections, while most nodes have a few connections.

(ii) We observe that all the base models have at least one outdegree as they are used by the task-specific models. For the dataset nodes, there are 20,093 datasets with zero indegrees, meaning they are the minimum datasets that can no longer be partitioned.

**Finding #1: The LLM supply chain graph is a large, sparse graph that exhibits a power-law degree distribution**, where a small number of high-degree base models and datasets serve as central hubs.

## 4.2 RQ #2: Supply Chain Structural Analysis

This research question aims to understand the topology and evolution of the LLM supply chain. To achieve that, we analyze the structural properties with connectivity and community analysis.

**Connectivity analysis** in a graph is crucial for understanding how well different nodes are linked and helps identify critical nodes and edges that maintain overall connectivity. This graph is a directed acyclic graph (DAG) because there are no dependencies in a cycle form, which also means there are no strongly connected components. Because of that, we only perform weakly connected components (WCC) analysis, and the total number of WCCs in our supply chain graph is 44,417.

Figure 4 shows the cumulative distribution function (CDF) of the WCC distribution. We made two interesting observations. (i) The largest WCC covers 244,063 nodes, accounting for 61.4% of all the nodes. It reflects the dense interconnections that pervade the ecosystem. This vital element is essential for effective information sharing, resource allocation, and structural support and is the base of the ecosystem. In the largest WCC, major models are included, such as Gemma-2B, DistilBERT, and GPT-2.

(ii) In contrast, the remaining 153,313 WCCs collectively hold 31.4% of the nodes, with most having 1, 2, to 3 nodes.

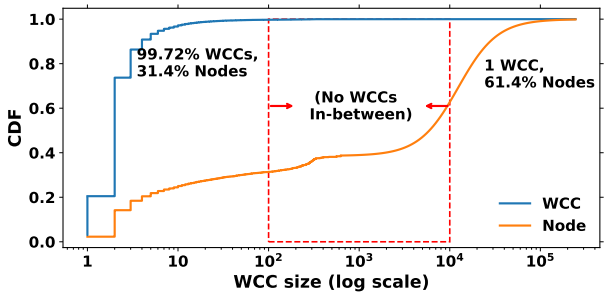


**Figure 4: Cumulative distribution function (CDF) of WCC size.**

This indicates a fragmented outer edge characterized by specialized models, less frequent datasets, or active experimental projects. The prevalence of these small, isolated pieces suggests niche attempts that lack integration with the overall system. For example, tblard/tf-allocine (French sentiment), distilbert-base-uncased-finetuned-sst-2-english (sentiment analysis), and PolyAI/minds14 (banking queries) remain disconnected due to limited reuse or insufficient metadata.

**Community analysis** is the process of identifying groups or clusters of nodes within a graph that are more densely connected internally than with the rest of the graph. In the context of the LLM supply chain, it reveals semantically or functionally aligned subgraphs that reflect the patterns of reuse and task-specific models or datasets. We use the Louvain method [10], which is a greedy optimization method for maximizing modularity. Modularity is a measure of how well a graph is divided into communities, where a high modularity score indicates that most edges fall within communities rather than between them. It compares the actual density of edges inside communities to the expected density if edges were placed at random.

Table 2 summarizes the top-10 communities detected using the Louvain method. Our analysis reveals an exceptionally modular structure, with each of the top communities achieving a high modularity score of 0.97, indicative of strong intra-community connectivity. Collectively, these communities span a wide range of functional domains, underscoring the presence of well-defined, task-aligned clusters within the ecosystem.

We made two interesting observations. (i) The largest community (ID 1) consists of 9,158 nodes and attains a modularity score of 0.97, indicating extremely cohesive internal structure. It revolves around base models such as OLMoE and CausalLM, and general-purpose datasets like prompt-perfect. This suggests a densely connected nucleus facilitating widespread reuse and fine-tuning—effectively functioning as the backbone of the LLM ecosystem.

(ii) Several other communities reflect clear task-based segmentation. For instance, Community 3 ( 7,058 nodes, modularity 0.97) focuses on benchmarking, with models like Wanxiang and datasets such as marco-o1. Similarly, Community 6 (5,030 nodes, modularity 0.97) centers on instruction-tuning, connecting models like MedLlama-3-8B with curated datasets like dpo-mix.

**Table 2: Top-10 Louvain communities in the HuggingGraph.**

| ID | Size | E.g. models | E.g. datasets | Modularity |
|---|---|---|---|---|
| 1 | 9158 | OLMoE, CausalLM | prompt-perfect | 0.97 |
| 2 | 7058 | qwen2.5_math | Marco-o1 | 0.97 |
| 3 | 6987 | Wanxiang | smartllama3.1 | 0.97 |
| 4 | 5696 | tinyllama | Llama-1B | 0.97 |
| 5 | 5355 | Qwen2.5-32B | Matter-0.2 | 0.97 |
| 6 | 5030 | MedLlama-3-8B | dpo-mix | 0.97 |
| 7 | 4297 | Electra, ArliAI | MixEval | 0.97 |
| 8 | 4133 | aesqwen1.5b | llava | 0.97 |
| 9 | 4067 | bert | TORGO | 0.97 |
| 10 | 3932 | Mistral | vicuna_format | 0.97 |

**Finding #2: The LLM supply chain graph reveals a structure with a densely connected core and a fragmented periphery.** A single dominant component encompasses the majority of nodes, supporting efficient model reuse and dependency tracing, while many smaller, isolated components reflect specialized or under-integrated efforts. Community analysis shows strong modularity, with nodes clustering around shared tasks and functional roles.

## 4.3 RQ #3: Dataset Supply Chain Analysis

This research question aims to provide comprehensive insights into the different usages of the datasets within the LLM supply chain graph. There are two relationships between datasets. On one hand, one dataset can be created by combining a variety of other datasets from disparate sources. On the other hand, one dataset can serve as a building block for new datasets, which enables continuous improvement in machine learning assets. In the following, we discuss the impacts on both training and benchmarking datasets.

**Training dataset impact.** In the LLM supply chain graph, 66,164 datasets act as training datasets that are used for model training or generating new datasets. Table 3 presents the top 10 datasets ranked by the highest number of included and derived datasets.

(i) We observe that a training dataset can include many small datasets. Sitting at the top is *macrocosm-os/images* [22] from macrocosm. It includes 550 sub-datasets and is a general-purpose image modification corpus used in advance multimodal AI research and large-scale vision-language model training. Moreover, *bespokelabs/Bespoke* [12] from bespokelabs includes 215 sub-datasets, is a high-quality synthetic dataset designed to enhance multimodal AI research.

(ii) The other observation is that a single dataset can be included in many larger datasets, showing the huge overlaps between the datasets and also the models trained with them. The right two columns of Table 3 show the top 10 datasets sorted by the maximum number of derived datasets they have been included. The leading contributor, *HuggingFaceH4* from HuggingFace has been involved in 989 derived datasets. It is primarily used for natural language understanding and generation tasks, including chatbots, text generation, code completion,

**Table 3: Top-10 datasets sorted by the # of included and # of derived datasets.**

| Dataset | # of included dataset | Dataset | # of derived datasets |
|---|---|---|---|
| macrocosm-os/images | 550 | HuggingFaceH4 | 989 |
| bespokelabs/Bespoke | 215 | CodeFeedback | 857 |
| databricks/databricks | 137 | MADLAD | 850 |
| Open-Orca/OpenOrca | 119 | Capybara | 823 |
| nguha/legalbench | 117 | Glot500 | 804 |
| OpenAssistant/oasst1 | 111 | dolphin-coder | 756 |
| LDJnr/Capybara | 104 | SlimOrca | 726 |
| kvn420/Tenro_V4.1 | 101 | orca_dpo_pairs | 687 |
| teknium/OpenHermes | 98 | gutenberg | 675 |
| Anthropic/hh-rlhf | 96 | samantha | 658 |

reasoning, and multilingual processing, enabling the training and enhancement of models specialized in advanced conversational and instruction-following capabilities [11]. Similarly, the *CodeFeedback* dataset from m-a-p has facilitated the creation of 857 derived datasets, is a collection of high-quality code instruction queries designed to enhance the training of large language models (LLMs) for code generation, debugging, and explanation tasks, enabling improved performance in complex programming scenarios[29].

**Finding #4: The datasets play critical roles in model training** with 66,164 datasets are contributing to model development through inclusion and derivation, as seen in *macrocosm-os/images* and *HuggingFaceH4*.

## 4.4 RQ #4: Supply Chain between LLM Models and Datasets

This research question aims to explore the interconnections between models and datasets within the supply chain graph. It provides valuable insights from dual perspectives, as discussed below, including one dataset versus multiple models and one model versus multiple datasets.

**One dataset versus multiple models** refers to the case when a single dataset is used to train multiple models. Table 4 shows the top 10 datasets based on the number of models trained on them. In particular, *Mistral-7B-v0.1* takes the leading position and is a widely adopted open-source dataset known for its strong performance in general-purpose language understanding and generation tasks. It has been used to train 1,093 models, including 300 fine-tuned variants, 300 adapters, 193 quantized models, and 300 merged models, highlighting its broad adoption across diverse model derivation strategies.

In addition, the dataset *TinyLlama-1.1B-v1.0*, a compact and efficient model variant designed for low-resource deployment, is used to train 728 models, featuring 300 fine-tuned variants and 300 adapters. Similarly, *open_llama_3b*, an open-access dataset of LLaMA, supports 285 adapter-based models, indicating a preference for lightweight, modular adaptation. The dataset *Yarn-Mistral-7b-128k* also shows significant reuse,

**Table 4: Top-10 datasets sorted by # of models trained.**

| Dataset | Total | Fine-tune | Adapter | Quantization | Merges |
|---|---|---|---|---|---|
| Mistral-7B-v0.1 | 1093 | 300 | 300 | 193 | 300 |
| TinyLlama-1.1B-v1.0 | 728 | 300 | 300 | 100 | 28 |
| open_llama_3b | 304 | 15 | 285 | 4 | 0 |
| Yarn-Mistral-7b-128k | 301 | 8 | 279 | 14 | 0 |
| WizardVicuna-open-llama | 280 | 12 | 261 | 7 | 0 |
| TinyLlama-1.1B-v0.6 | 266 | 10 | 243 | 13 | 0 |
| Yarn-Mistral-7b-64k | 248 | 0 | 242 | 6 | 0 |
| Nous-Capybara-7B-V1 | 213 | 11 | 174 | 27 | 1 |
| MAmmoTH2-7B | 213 | 0 | 0 | 3 | 210 |
| Starling-LM-7B-alpha | 210 | 10 | 165 | 18 | 17 |

**Table 5: Top-10 models sorted by the # of datasets used for training.**

| Model | Model Type | # of dataset used-for training |
|---|---|---|
| DeBERTa-ST-AllLayers-v3.1 | Fine tune | 116 |
| DeBERTa-ST-AllLayers-v3.1bis | Adapters | 116 |
| static-similarity-mrl-mul-v1 | Fine tune | 108 |
| static-similarity-mrl-multilingual | Fine tune | 108 |
| ModernBERT-base-embed | Fine tune | 88 |
| Llama-3.2-3B-Instruct | Fine tune | 87 |
| Llama-3.2-3B-Instruct-GGUF | Quantization | 87 |
| DavidLanz-3.2-3B-Instruct | Fine tune | 87 |
| static-retrieval-mrl-en-v1 | Fine tune | 79 |
| XLMRoBERTaM3-CustomPoolin | Fine tune | 72 |

contributing to 279 adapters and 14 quantized models. Furthermore, *MAmmoTH2-7B* stands out with 210 merged models, showcasing its role in ensemble-style model fusion rather than traditional fine-tuning or adapter strategies. Lastly, the dataset *Starling-LM-7B-alpha*, known for alignment-focused training, contributes to a diverse range of downstream models, including 165 adapters and 18 quantized versions, illustrating its utility in fine-tuning and compression workflows.

**One model versus multiple datasets** refers to the case when an LLM model is trained with multiple datasets. Table 5 shows the top 10 models ranked by the number of datasets used for training. *DeBERTa-ST-AllLayers-v3.1*, a fine-tuned variant of the DeBERTa architecture, takes the top position, having been trained on 116 different datasets. Its adapter-based counterpart, *DeBERTa-ST-AllLayers-v3.1bis*, also leverages the same number of datasets via adapter-based training, emphasizing modular reuse across tasks.

In addition, models like *static-similarity-mrl-mul-v1* and *static-similarity-mrl-multilingual* are both fine-tuned on 108 datasets, indicating their role in multilingual and multi-task similarity-based retrieval applications. The *ModernBERT-base-embed* and *Llama-3.2-3B-Instruct* families show strong dataset diversity as well, with 88 and 87 datasets respectively, across fine-tuning and quantized variants (e.g., GGUF format). Interestingly, most of these models are fine-tuned—specifically, 8 out of the top 10—highlighting a trend where base models are transformed into task-specific variants through diverse training datasets. This pattern suggests that fine-tuning remains a dominant strategy for adapting base models to downstream tasks across heterogeneous data sources.

**Finding #5: There exists a strong interdependence between models and datasets**, where widely adopted datasets like *Mistral-7B-v0.1* and *TinyLlama-1.1B-v1.0* serve as the base for training hundreds of models across different adaptation techniques, while models such as *DeBERTa-ST-AllLayers-v3.1* leverage diverse datasets to enhance their versatility, highlighting the critical role of dataset-model interactions in advancing AI capabilities.

## 4.5 RQ #5: Temporal Update and Tracking

This research question aims to focus on the value of tracking fine-grained, time-based changes in the LLM supply chain. Our Δ-based update mechanism (discussed in Section 3.4), produces a daily *snapshot* of the HuggingGraph, capturing precisely how many nodes and edges are added each day, how that translates into net growth, and what the busiest single day looks like. To better understand the implications of this tracking, we adopt both qualitative and quantitative perspectives on the evolving LLM ecosystem.

**Fine-grained temporal insights into the LLM ecosystem.** We would like to understand how the LLM supply chain evolves on a day-to-day basis. Hence, we adopt this fine-grained temporal qualitative approach that highlights fluctuations in model and dataset activity. This strategy allows us to uncover ecosystem-level dynamics such as release bursts, contributor behavior, and category-specific trends over time.

Figure 5 visualizes the daily Δ-based growth of five key node categories (base, fine-tuned, adapters, quantized variants models, and datasets) from June 25 to July 15, 2025.

We observe that, (i) the fine-tuned models dominate the daily activity, averaging over 1,700 new entries per day, followed by consistent contributions from adapters (∼750/day) and datasets (∼700/day). Clear spikes on specific days—e.g., July 7 and July 9—coincide with major releases such as the *Mistral-Fusion-v3* fine-tuning wave and dataset refreshes like *HFTime2025-News*. (ii) Furthermore, adapter uploads peaked at 887 on June 28, while quantized variants reached 290 on July 9, driven by releases such as *QWin-GGUF-7B*. These patterns demonstrate HuggingGraph's ability to capture evolving supply chain dynamics at a fine-grained, time-sensitive resolution, revealing ecosystem bursts and contributor behaviors in real time.

**Quantifying daily incremental graph growth.** To complement our qualitative insights with measurable evidence, we adopt a quantitative approach that captures the scale and efficiency of daily changes to the LLM supply chain. This allows us to evaluate the performance of our Δ-based update mechanism, ensuring support for large-scale, time-sensitive graph

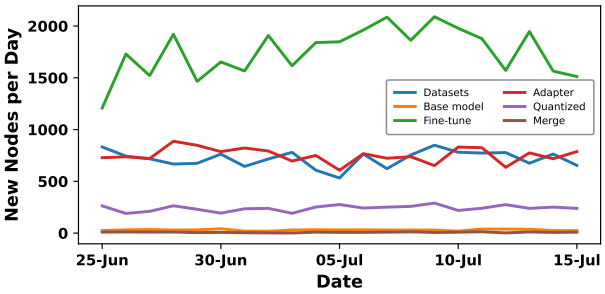


**Figure 5: The temporal update for the datasets and different models on Hugging Face from June 25 to July 15, 2025.**

maintenance in a computationally efficient and auditable manner.

Table 6 provides a detailed breakdown of daily additions, net changes, and peak updates across node and edge types from Jun. 25 to Jul. 15, 2025. (i) On average, Hugging Face added 126 new fine-tuned models per day (peaking at 2,090), followed by adapters and datasets with 41 and 39 daily insertions, respectively. Quantized and merged models contributed 16 and 1 additions per day, respectively. In total, 2,526 fine-tuned models and 785 datasets were added during this period. (ii) Each day also saw approximately 23 ± 6 new dependency edges—linking models to their parent checkpoints or training datasets—consistent. By comparing successive graph versions, our update mechanism applies only the incremental changes ($\Delta_t$) rather than rebuilding the full graph—keeping updates both scalable and auditable in practice.

**Finding #6: The Δ-based update mechanism offers clear visibility into the LLM ecosystem's rapid and continuous evolution**, with daily growth primarily driven by fine-tuned, adapter, and quantized models. It enables efficient, audit-friendly tracking by applying only incremental changes, eliminating the need to rebuild the entire graph. These capabilities ensure that HuggingGraph remains both scalable and precise in capturing the dynamic nature of the LLM supply chain.

## 5 Related Work

This section aims to provide in-depth coverage of the current research landscape concerning the opportunities within the LLM supply chain, mainly emphasizing the different dimensions of the relationship between the models and datasets, focusing on the model and dataset hosting platforms [14, 37].

**LLM supply chain perspectives in AI.** LLMs mark the revolutionary era in AI, with the edge passing through an exponential growth phase. A recent paper presents LLMs key components, including model infrastructure, lifecycle, and downstream applications [45]. Another work showed that reusing these models has become so widespread, which encourages the sharing and adaptation of foundation models on a larger scale [19]. An open-source AI ecosystem, such as Hugging Face, hosts a broad range of LLMs and datasets and, therefore, plays an important role in democratizing AI technologies [34].

**Table 6: Daily Δ statistics for HuggingGraph updates (June 25–July 15, 2025), showing average additions, net changes, and peak values by node and edge type.**

| Category | New- per day | Net Change- per day | Peak- in month |
|---|---|---|---|
| Datasets | 39 ± 62 | +785 | 849 |
| Base model | 2 ± 5 | +57 | 42 |
| Fine-tune | 126 ± 168 | +2526 | 2090 |
| Adapter | 41 ± 64 | +828 | 887 |
| Quantized | 16 ± 19 | +321 | 290 |
| Merge | 1 ± 2 | +34 | 13 |
| Dependency edges | 23 ± 6 | +189 | 289 |

The foundation models are key ingredients in such an ecosystem [46]. They can capture vast amounts of information from various datasets, allowing for effective task-specific variants like fine-tuning [16]. This trend underscores the significant impact of technological advancements on the democratization and innovation in AI [9].

**Relationship analysis between LLM models and datasets.** A recent study investigates the practical adaptation of foundation models to specific domains and tasks. Multitask fine-tuning has demonstrated the potential to enhance performance on target tasks with scarce labels [47]. In plant phenotyping, adapting vision foundation models by techniques like adapter tuning and decoder tuning has shown results comparable to those of leading task-specific models [7]. The Quadapter technique for language models tackles quantization difficulties by incorporating learnable parameters that scale activations channel-wise, mitigating overfitting during quantization-aware training [33]. The PathoTune framework in pathology imaging utilizes multimodal prompt tuning to reconcile domain discrepancies among foundation models, tasks, and specific instances, exhibiting enhanced performance compared to single-modality methods and facilitating the direct adaptation of natural visual foundation models to pathological tasks [26]. These findings underscore the efficacy of various adaptation approaches in improving foundation model performance across multiple areas.

From the above, we can see that models heavily depend on each other for fine-tuning, adaptation, or quantization. Building on these insights, two other works have indicated that serious security vulnerabilities may encumber the LLM supply chain to a few developers [20, 32]. These factors may affect the diversity of innovation. The authors relate several challenges arising during the software engineering, security, and privacy of different relations and components involved in model creation and deployment [18, 45]. Moreover, the vulnerabilities can be transferable from one model to another model during the fine-tuning process [15]. Therefore, serious security challenges are also posed because the vulnerabilities of the repositories threaten the whole ecosystem [20].

**Comparison with HuggingGraph.** Our work is different from prior works in three aspects. First, the motivation for this work stems from the transferable property [15] of one model

to others and during the training process with the datasets. Because of this, our analysis is mainly about the hierarchical relationship between the models, which model is used to fine-tune, adapt, or quantize other models, and which datasets are used to train the models. As it is challenging to get consistent hierarchical relational information, we made a supply chain graph from the metadata using APIs we got from the models and datasets. Second, as our analysis is one of the premier tasks in constructing a supply chain graph, for analyzing the relationship, we focus on the different graph properties like connectivity and WCC analysis. Third, we use forward and backward supply chain analysis to examine the built graph and learn more about the hierarchies of the models and datasets during fine-tuning, adaptation, quantization, and training. This is different from previous methods, which mostly focused on training the models. This would facilitate determining and identifying the relationship between the model and the dataset. More importantly, we conclude with multiple interesting findings, which we believe could provide important insights to the LLM community.

## 6 Discussion

HuggingGraph presents a technique to analyze the supply chain of the LLM ecosystem. The proposed graph can be used for various applications, e.g., auditing provenance, identifying biases, and revealing trends like quantized model scarcity, aiding researchers and platform maintainers. We discuss the following two use cases.

**Use case #1: Tracing lineage and dependencies in the LLM supply chain.** In the LLM ecosystem, models are frequently built upon others through fine-tuning, adapter training, or quantization, forming complex chains of dependencies. However, when these relationships are not explicitly visible, it becomes difficult to verify where a model comes from, whether it inherits bias from upstream datasets, or if it complies with licensing constraints. HuggingGraph can be used to address this challenge by constructing the supply chain of models and datasets, uncovering both direct and derived dependencies, even when they are not formally documented. For example, it can trace how the model Meta-llama indirectly relies on a dataset like Wikimedia via Awesome-Chatgpt-prompt (Figure 2). This transparency supports developers, auditors, and policymakers in validating provenance, detecting risks, and enabling responsible, trustworthy AI deployment.

**Use case #2: Identifying critical nodes and structural vulnerabilities.** In the LLM ecosystem, certain models (e.g., gemma-2b) and datasets (e.g., The Pile) are reused so frequently that they become critical structural hubs, where failure or removal of them could disrupt numerous downstream dependencies. These hidden single points of failure are difficult to detect without a comprehensive view of resource interconnections. HuggingGraph can be used to address this by modeling the supply chain as a graph and analyzing node connectivity to surface highly reused models and datasets with significant inbound or outbound links. This visibility enables maintainers to safeguard vital assets and helps developers mitigate the risk of overreliance on fragile or under-maintained components, fostering more robust and secure model development pipelines.

## 7 Conclusion

This project studies the relationships between models and datasets in the LLM ecosystem, which are the central parts of the *LLM supply chain*. First, we design a methodology to systematically collect the supply chain information of LLMs. With the collected information, we design a new graph to model their relationships, which is a large directed heterogeneous graph, having *397,376 nodes* and *453,469 edges*. From there, we perform different types of analysis and make multiple interesting findings.

## Acknowledgment

This work was supported in part by National Science Foundation grants 2331301, 2508118, 2516003, 2419843. The views, opinions, and/or findings expressed in this material are those of the authors and should not be interpreted as representing the official views of the National Science Foundation, or the U.S. Government.

## References

[1] Alexandre Agossah, Frédérique Krupa, Matthieu Perreira Da Silva, and Patrick Le Callet. 2023. Llm-based interaction for content generation: A case study on the perception of employees in an it department. In *Proceedings of the 2023 ACM International Conference on Interactive Media Experiences*. 237–241.

[2] Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. 2025. Evolutionary optimization of model merging recipes. *Nature Machine Intelligence* 7, 2 (2025), 195–204.

[3] Dean Allemang and Juan Sequeda. 2024. Increasing the LLM Accuracy for Question Answering: Ontologies to the Rescue! *arXiv preprint arXiv:2405.11706* (2024).

[4] Anonymous. 2022. Review of Supply Chain Management in Manufacturing Organizations. *ResearchGate* (2022). https://www.researchgate.net/publication/377659033_Review_of_supply_chain_management_in_manufacturing_organizations

[5] Mourad Bahani, Aziza El Ouaazizi, and Khalil Maalmi. 2023. The effectiveness of T5, GPT-2, and BERT on text-to-image generation task. *Pattern Recognition Letters* 173 (2023), 57–63.

[6] Euan Bonner, Ryan Lege, and Erin Frazier. 2023. Large Language Model-Based Artificial Intelligence in the Language Classroom: Practical Ideas for Teaching. *Teaching English with Technology* 23, 1 (2023), 23–41.

[7] Feng Chen, Mario Valerio Giuffrida, and Sotirios A Tsaftaris. 2023. Adapting vision foundation models for plant phenotyping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 604–613.

[8] M. C. Chou, H. Ye, X. M. Yuan, Y. N. Cheng, L. Chua, Y. Guan, S. E. Lee, and Y. C. Tay. 2006. Analysis of a Software-Focused Products and Service Supply Chain. *IEEE Transactions on Industrial Informatics* 2, 4 (2006), 295–303. doi:10.1109/TII.2006.884368

[9] Jelena Cupać, Hendrik Schopmans, and İrem Tuncer-Ebetürk. 2024. Democratization in the age of artificial intelligence: introduction to the special issue. 899–921 pages.

[10] Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. 2011. Generalized louvain method for community detection in large networks. In *2011 11th international conference on intelligent systems design and applications*. IEEE, 88–93.

[11] Jürgen Dietrich and André Hollstein. 2025. Performance and reproducibility of large language models in named entity recognition: Considerations for the use in controlled environments. *Drug Safety* 48, 3 (2025), 287–303.

[12] Mark Elliot, Claire Little, and Richard Allmendinger. 2024. The production of bespoke synthetic teaching datasets without access to the original data. In *International Conference on Privacy in Statistical Databases*. Springer, 144–157.

[13] Hugging Face. [n. d.]. Hugging Face – The AI community building the future. https://huggingface.co/

[14] Muhammad Usman Hadi, Qasem Al Tashi, Abbas Shah, Rizwan Qureshi, Amgad Muneer, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, et al. 2024. Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea Preprints* (2024).

[15] Hossein Hajipour, Keno Hassler, Thorsten Holz, Lea Schönherr, and Mario Fritz. 2024. CodeLMSec Benchmark: Systematically Evaluating and Finding Security Vulnerabilities in Black-Box Code Language Models. In *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 684–709.

[16] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. 2021. Pre-trained models: Past, present and future. *AI Open* 2 (2021), 225–250.

[17] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. 2023. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933* (2023).

[18] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* (2023).

[19] Wenxin Jiang, Nicholas Synovic, Matt Hyatt, Taylor R Schorlemmer, Rohan Sethi, Yung-Hsiang Lu, George K Thiruvathukal, and James C Davis. 2023. An empirical study of pre-trained model reuse in the hugging face deep learning model registry. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 2463–2475.

[20] Adhishree Kathikar, Aishwarya Nair, Ben Lazarine, Agrim Sachdeva, and Sagar Samtani. 2023. Assessing the vulnerabilities of the open-source artificial intelligence (AI) landscape: A large-scale analysis of the Hugging Face platform. In *2023 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 1–6.

[21] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, Vol. 1. Minneapolis, Minnesota, 2.

[22] Mantaro Kido. 1961. Origin of Japanese psychology and its development. *Psychologia* 4, 1 (1961), 1–10.

[23] Philippe Laban, Wojciech Kryściński, Divyansh Agarwal, Alexander Richard Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. 2023. SUMMEDITS: measuring LLM ability at factual reasoning through the lens of summarization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 9662–9676.

[24] Beibin Li, Konstantina Mellou, Bo Zhang, Jeevan Pathuri, and Ishai Menache. 2023. Large language models for supply chain optimization. *arXiv preprint arXiv:2307.03875* (2023).

[25] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering* 34, 1 (2020), 50–70.

[26] Jiaxuan Lu, Fang Yan, Xiaofan Zhang, Yue Gao, and Shaoting Zhang. 2024. Pathotune: Adapting visual foundation model to pathological specialists. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 395–406.

[27] Yinquan Lu, Wenhao Zhu, Lei Li, Yu Qiao, and Fei Yuan. 2024. Llamax: Scaling linguistic horizons of llm by enhancing translation capabilities beyond 100 languages. *arXiv preprint arXiv:2407.05975* (2024).

[28] Felix Merrick, Maria Radcliffe, and Rupert Hensley. 2024. Upscaling a smaller llm to more parameters via manual regressive distillation. (2024).

[29] Benjamin S Meyers, Nuthan Munaiah, Emily Prud'Hommeaux, Andrew Meneely, Cecilia Ovesdotter Alm, Josephine Wolff, and Pradeep K Murukannaiah. 2018. A dataset for identifying actionable feedback in collaborative software development. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 126–131.

[30] Suvir Mirchandani, Fei Xia, Pete Florence, Brian Ichter, Danny Driess, Montserrat Gonzalez Arenas, Kanishka Rao, Dorsa Sadigh, and Andy Zeng. 2023. Large language models as general pattern machines. *arXiv preprint arXiv:2307.04721* (2023).

[31] Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt llm evaluation. *Transactions of the Association for Computational Linguistics* 12 (2024), 933–949.

[32] Cailean Osborne, Jennifer Ding, and Hannah Rose Kirk. 2024. The AI community building the future? A quantitative analysis of development activity on Hugging Face Hub. *Journal of Computational Social Science* (2024), 1–39.

[33] Minseop Park, Jaeseong You, Markus Nagel, and Simyung Chang. 2022. Quadapter: Adapter for gpt-2 quantization. *arXiv preprint arXiv:2211.16912* (2022).

[34] Matteo Riva, Tommaso Lorenzo Parigi, Federica Ungaro, and Luca Massimino. 2024. HuggingFace's impact on medical applications of artificial intelligence. *Computational and Structural Biotechnology Reports* (2024), 100003.

[35] Paul Röttger, Fabio Pernisi, Bertie Vidgen, and Dirk Hovy. 2024. Safetyprompts: a systematic review of open datasets for evaluating and improving large language model safety. *arXiv preprint arXiv:2404.05399* (2024).

[36] Zhiqiang Shen, Tianhua Tao, Liqun Ma, Willie Neiswanger, Zhengzhong Liu, Hongyi Wang, Bowen Tan, Joel Hestness, Natalia Vassilieva, Daria Soboleva, et al. 2023. Slimpajama-dc: Understanding data combinations for llm training. *arXiv preprint arXiv:2309.10818* (2023).

[37] B Sindhu, RP Prathamesh, MB Sameera, and S KumaraSwamy. 2024. The evolution of large language model: Models, applications and challenges. In *2024 International Conference on Current Trends in Advanced Computing (ICCTAC)*. IEEE, 1–8.

[38] Tanmay Singla, Dharun Anandayuvaraj, Kelechi G Kalu, Taylor R Schorlemmer, and James C Davis. 2023. An empirical study on using large language models to analyze software supply chain security failures. In *Proceedings of the 2023 Workshop on Software Supply Chain Offensive Research and Ecosystem Defenses*. 5–15.

[39] Sonatype. 2015. *2015 State of the Software Supply Chain Report*. Technical Report. Sonatype. https://www.sonatype.com/hubfs/White_Papers/2015_State_of_the_Software_Supply_Chain_Report-.pdf

[40] Xin Tan, Taichuan Li, Ruohe Chen, Fang Liu, and Li Zhang. 2024. Challenges of Using Pre-trained Models: the Practitioners' Perspective. *arXiv preprint arXiv:2404.14710* (2024).

[41] Yuli Vasiliev. 2020. *Natural language processing with Python and spaCy: A practical introduction*. No Starch Press.

[42] Neelay Velingker, Jason Liu, Amish Sethi, William Dodds, Zhiqiu Xu, Saikat Dutta, Mayur Naik, and Eric Wong. [n. d.]. CLAM: Unifying Finetuning, Quantization, and Pruning by Chaining LLM Adapter Modules. In *Workshop on Efficient Systems for Foundation Models II@ ICML2024*.

[43] Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Will we run out of data? Limits of LLM scaling based on human-generated data. *arXiv preprint arXiv:2211.04325* (2024), 13–29.

[44] Kushala VM, Harikrishna Warrier, Yogesh Gupta, et al. 2024. Fine Tuning LLM for Enterprise: Practical Guidelines and Recommendations. *arXiv preprint arXiv:2404.10779* (2024).

[45] Shenao Wang, Yanjie Zhao, Xinyi Hou, and Haoyu Wang. 2024. Large language model supply chain: A research agenda. *ACM Transactions on Software Engineering and Methodology* (2024).

[46] Mengwei Xu, Wangsong Yin, Dongqi Cai, Rongjie Yi, Daliang Xu, Qipeng Wang, Bingyang Wu, Yihao Zhao, Chen Yang, Shihe Wang, et al. 2024. A survey of resource-efficient llm and multimodal foundation models. *arXiv preprint arXiv:2401.08092* (2024).

[47] Zhuoyan Xu, Zhenmei Shi, Junyi Wei, Fangzhou Mu, Yin Li, and Yingyu Liang. 2024. Towards Few-Shot Adaptation of Foundation Models via Multitask Finetuning. *arXiv preprint arXiv:2402.15017* (2024).

[48] Gokul Yenduri, M Ramalingam, G Chemmalar Selvi, Y Supriya, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, G Deepti Raj, Rutvij H Jhaveri, B Prabadevi, Weizheng Wang, et al. 2024. Gpt (generative pre-trained transformer)–a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE Access* (2024).

[49] Wentao Zou, Qi Li, Jidong Ge, Chuanyi Li, Xiaoyu Shen, Liguo Huang, and Bin Luo. 2023. A Comprehensive Evaluation of Parameter-Efficient Fine-Tuning on Software Engineering Tasks. *arXiv preprint arXiv:2312.15614* (2023).

[50] Ajdin Čolaković, Aleksandar Đorđević, Branislav Cvetić, Milan Danilović, and Dejan Vasiljević. 2021. Traditional vs Digital Supply Chains. *ResearchGate* (2021). https://www.researchgate.net/publication/381617842_Traditional_vs_Digital_Supply_Chains