Legal Compliance in AI Models: A Novel Framework for Analyzing Licensing Risks

Mohammed Abdul Nadeem
abdulnadeemms@gmail.com

***Abstract*—As AI model development becomes increasingly reliant on integrating datasets, pre-trained models, and open-source components, ensuring license compliance is critical. We present ModelGo, a novel framework designed to detect and resolve licensing conflicts across AI pipelines. Unlike traditional license analysis tools focused solely on software code, ModelGo incorporates a taxonomy tailored to machine learning workflows, enabling fine-grained evaluation of licensing compatibility for datasets, models, and composite artifacts. Our case studies, drawn from real-world scenarios, demonstrate the framework's effectiveness in identifying licensing risks and guiding developers toward compliant reuse. This work contributes a scalable and extensible approach to managing legal risks in AI systems.**

***Index Terms*—License analysis, AI licensing, model mining**

I. Introduction

The development of machine learning (ML) systems involves integrating multiple components—such as pretrained models, datasets, and libraries—each potentially governed by different licenses. These licenses vary in their legal implications and restrictions regarding redistribution, commercial use, and derivative creation. Traditional license compliance tools were designed for open-source software and lack support for non-code AI components. This mismatch creates ambiguity in legal responsibilities and exposes developers to risks of unintentional violations. This paper introduces **ModelGo**, a framework designed to assess license compatibility in AI workflows. Our key contributions are:

- A licensing taxonomy aligned with AI-specific operations.
- A dependency graph structure modeling legal and technical interactions.
- An automated compliance analysis engine integrated with development workflows.

ModelGo is introduced to address these challenges by offering an automated framework for license analysis tailored to ML development. The tool incorporates a taxonomy that aligns AI development activities with licensing terminology, enabling structured interpretation of rights and restrictions. Through a recursive dependency structure, the tool evaluates whether the licenses of reused components are compatible with the intended use and distribution of the final ML product. To validate the effectiveness of the proposed approach, five use cases involving real-world datasets and models are presented. These use cases demonstrate common patterns of license conflicts, highlight frequent non-compliance risks, and suggest practical strategies for resolving licensing ambiguities. The remainder of this paper presents related work in Section II, describes the methodology behind ModelGo in Section III, analyzes selected case studies in Section V, and concludes in Section **??**.

II. Related Work

License compliance has been a longstanding area of concern in software engineering, particularly in the realm of Open Source Software (OSS). Traditional efforts in this domain have primarily focused on automating the identification and classification of licenses embedded in source code. Early tools such as FOSSology and Ninka utilized pattern matching and heuristic-based analysis to detect license headers and textual similarities within software artifacts [1], [2]. These tools laid the foundation for scalable license auditing in large codebases.

As OSS ecosystems evolved, researchers turned their attention toward more complex challenges, such as license compatibility and conflict detection in dependency graphs. Tools like SPDX (Software Package Data Exchange) and Licensee adopted metadata-driven approaches to capture the hierarchical structure of software packages and their licensing constraints [3], [4]. Dependency-based modeling enabled the identification of transitive license violations, particularly in package managers and containerized applications.

However, traditional OSS compliance methodologies are largely insufficient for machine learning (ML) systems, which often integrate non-code assets such as datasets, pretrained models, and pipeline artifacts. Unlike code, these components are frequently governed by heterogeneous license types, including Creative Commons (CC), bespoke academic terms, and emergent AI-specific licenses. Recent studies have shown that dataset creators often fail to specify clear usage terms, resulting in ambiguities around redistribution, commercial use, and derivative creation [**?**], [5].

To address these gaps, initiatives like the Montreal Data License (MDL) introduced a modular taxonomy for dataset licensing in AI contexts [6]. MDL aims to clarify the permissible operations—such as training, benchmarking, and commercialization—across various license configurations. Despite its structured design, MDL has yet to see widespread adoption, and most ML practitioners continue to operate within a fragmented legal landscape.

Parallel legal scholarship has raised pressing concerns about copyright law's applicability to ML training. Several works have examined whether training on third-party content constitutes derivative work creation or violates fair use doctrines, especially when dealing with copyrighted datasets or scraped

web content [7], [8]. These debates are ongoing and have significant implications for both research and commercial deployment of ML models.

More recently, there has been a shift toward AI-specific license frameworks designed to capture the unique operational characteristics of ML models. Licenses like OpenRAIL and RAIL seek to impose behavioral constraints, restrict harmful use, and enforce transparency obligations in model deployment [9], [10]. These licenses mark a departure from traditional open-source principles by integrating ethical and social considerations directly into legal terms.

Despite these developments, few tools offer unified license analysis across code, data, and models. Existing OSS scanners fail to account for license propagation through composite artifacts or to track indirect reuse mechanisms such as model distillation and transfer learning. In this context, **ModelGo** fills a critical void by enabling fine-grained tracking of component-level dependencies and mapping them to domain-specific license obligations. By doing so, it supports compliance analysis across the full lifecycle of ML artifact creation, reuse, and redistribution.

III. Methodology

License compliance in machine learning (ML) projects presents a unique challenge, as traditional software compliance tools are ill-equipped to handle the multifaceted nature of ML assets. These assets—ranging from datasets and pretrained models to training scripts and pipelines—are governed by a diverse array of licenses, many of which extend beyond conventional software licensing paradigms. The **ModelGo** framework addresses this gap by offering an end-to-end methodology that integrates legal interpretation with technical dependency analysis. This section outlines the framework's core methodology, consisting of three primary components: an ML-specific license taxonomy, dependency graph construction, and automated license compliance analysis.

A. Taxonomy of AI Activities

A central innovation in ModelGo is its *taxonomy of AI activities*, which interprets license terms in the context of ML-specific operations. While traditional license analysis tools focus on actions like "modify" or "distribute," ML development requires a nuanced understanding of how data, models, and code are reused.

To achieve this, ModelGo introduces a set of clearly defined activity categories:

- **Embedding:** Incorporating models or data directly into an ML pipeline or final product.
- **Fine-tuning:** Adapting a pretrained model to a new dataset, often resulting in a derivative model.
- **Training:** Using datasets to generate new models from scratch.
- **Model Fusion:** Combining multiple models into a unified architecture (e.g., ensemble learning).
- **Synthetic Content Generation:** Using models to create new, often user-facing content (e.g., images, text).

Each of these activities is categorized under one of four broader licensing interpretations:

1) **Combination with strong separation** (e.g., SaaS deployment using external models).
2) **Combination with weak separation** (e.g., packaging datasets with model binaries).
3) **Derivatives from data** (e.g., outputs directly trained on or fine-tuned from licensed datasets).
4) **Derivatives from concepts** (e.g., inspiration-based learning or reverse-engineered architectures).

By aligning ambiguous licensing clauses—such as "remix," "reuse," or "create adaptations"—with these categories, ModelGo provides a systematic framework to evaluate the applicability and scope of each license term in ML contexts.

B. Dependency Graph Construction

ModelGo models ML projects as hierarchical, multi-modal dependency graphs. This structure captures both the technical lineage and the legal relationships among datasets, models, scripts, and downstream applications.

Each node in the graph represents an individual component, annotated with metadata including:

- Component type (e.g., dataset, model, library, API).
- License (standard SPDX identifier or custom).
- Format of use (e.g., raw files, binaries, API endpoints).
- Intended release context (e.g., internal R&D, public SaaS, open-source distribution).

Edges between nodes represent activity-based relationships and are typed according to the ML-specific taxonomy. These include:

- **Mix-works:** Direct integration into the final ML product (e.g., transfer learning where the pretrained model is shipped).
- **Sub-works:** Components used during development but not bundled in the final release (e.g., training data).
- **Aux-works:** Tools or libraries that aid development but do not affect final outputs (e.g., internal visualization tools).

This layered representation allows ModelGo to isolate and reason about license inheritance, scope of influence, and transitive risk exposure in a traceable and auditable manner.

C. License Compliance Analysis

With the dependency graph and taxonomy in place, ModelGo conducts a three-stage compliance analysis designed to detect violations, assess risks, and report actionable insights.

1) License Registration: Each component is registered with its associated license, either via standard SPDX identifiers or custom mappings derived from manual input or automated scraping of license files. The tool also captures metadata on:

- License version and modifications.
- Sublicensing permissions.
- Restrictions on commercial use, attribution, share-alike requirements, and revocation clauses.

2) License Propagation: In this phase, ModelGo simulates the flow of license obligations through the dependency graph. This includes:

- Propagation of copyleft constraints (e.g., GPL-style obligations).
- Evaluation of attribution requirements across derivative artifacts.
- Resolution of potential conflicts (e.g., combining a non-commercial dataset with a commercial product).
- Detection of revocable or time-limited rights that may affect model deployment.

ModelGo applies a rule engine to infer whether downstream use of a component triggers additional obligations due to license interdependencies.

3) Validation and Reporting: The final phase involves generating a comprehensive compliance report. The report categorizes detected issues into:

- **Errors:** Clear violations, such as combining incompatible licenses or distributing without mandatory attribution.
- **Warnings:** Potential issues needing legal review, such as vague terms of use or license expiration.
- **Recommendations:** Suggested remediation steps, such as substituting a component with a more permissive alternative.

Reports are exportable in human-readable (PDF/HTML) and machine-readable formats (JSON/XML), making them suitable for both legal audits and CI/CD integrations.

This methodology enables legal and engineering teams to proactively evaluate the compliance implications of complex ML workflows. By uniting formal license semantics with graph-based software modeling, ModelGo provides a scalable, extensible, and interpretable solution for managing the legal risks inherent in modern AI development.

IV. Comparative Study of Licensing Across Jurisdictions

The legal interpretation of AI licensing terms such as *fair use*, *derivative works*, and *redistribution rights* varies considerably across jurisdictions. This inconsistency presents significant challenges for machine learning (ML) developers and organizations operating in multiple countries. A license compliant in one region may inadvertently violate regulations in another, especially in cases of cross-border model deployment, dataset usage, or cloud-based AI services.

A. Variability in Legal Interpretation

Table I presents a comparative overview of how key jurisdictions interpret critical licensing concepts in the context of AI and ML systems.

B. Cross-Border ML Deployment Risks

Deploying machine learning models across jurisdictions entails unique legal uncertainties:

- **License Validity**: A dataset licensed under Creative Commons in the US may face usage restrictions in the EU due to stricter privacy or IP laws.
- **Copyleft Inheritance**: GPL-style licenses may compel public sharing of source code in one jurisdiction but not in another, creating compliance ambiguity.
- **Cloud-Based Deployment**: SaaS models using externally licensed components can trigger conflicting obligations when accessed internationally.
- **Ethical Licensing Terms**: Behavioral use clauses (e.g., OpenRAIL) lack global enforceability due to differences in contract and IP law.

C. Toward Harmonization

To enable scalable and legally robust AI development, there is a growing need for:

- International alignment of AI licensing standards (e.g., expansion of the Montreal Data License framework).
- Legal toolkits or APIs that translate license obligations across jurisdictions in real time.
- Policy dialogues that involve AI developers, lawyers, and regulators to co-develop compliant cross-border AI solutions.

In conclusion, developers and organizations must treat license compliance not as a static checklist, but as a dynamic and jurisdiction-sensitive process. The integration of tools like ModelGo into legal review workflows offers a promising path toward mitigating global licensing risks in ML.

V. Case Studies

To evaluate the legal implications and reuse feasibility of machine learning (ML) assets under diverse licensing regimes, we developed five representative case studies using the ModelGo tool. These case studies illustrate practical licensing challenges and demonstrate ModelGo's capability in identifying and analyzing conflicts in real-world scenarios.

A. Case I: Corpus Combination

This case explores the construction of a multilingual corpus by combining datasets and translation outputs sourced from arXiv, Stack Exchange, PubMed, Deep-Sequoia, and FreeLaw. The goal was to utilize this corpus in a commercial ML product.

Licensing Analysis:

- **Copyleft enforcement:** Licenses such as CC-BY-SA and LGPL-LR imposed conditions to retain original licensing terms, even when derivative or aggregated datasets were created.
- **Non-commercial and no-derivative clauses:** CC-BY-NC-ND and CC-BY-ND explicitly prohibited redistribution of modified or transformed outputs, curtailing the commercial use intent.
- **GPL-3.0 entanglement:** Inclusion of datasets under GPL-3.0 introduced mandatory disclosure and copyleft obligations on the entire derived corpus.

TABLE I
JURISDICTIONAL COMPARISON OF LICENSING INTERPRETATIONS IN AI CONTEXTS

Legal Concept	United States	European Union	Japan / India
Fair Use / Fair Dealing	*Flexible* – case-by-case; favors transformative use (e.g., training data)	*Limited* – narrower scope; often excludes ML training without consent	*Restrictive* – fair dealing more limited; lacks clarity for AI use
Derivative Works	Applies to model fine-tuning, transfer learning, and generation if content is similar to source	Broadly interpreted; derivative models may trigger additional obligations	Not well-defined for AI; derivative scope varies by content type
Redistribution of Outputs	Often allowed if output is transformative or disconnected from original content	May require license compliance even for outputs (especially under ShareAlike)	Outputs not clearly covered; high legal uncertainty, especially for synthetic data
Enforceability of Open Licenses	Strong precedent; courts uphold OSS terms like GPL	High compliance standards; licenses like CC are strictly interpreted	Enforceability varies; OSS principles gaining recognition but still evolving

This scenario demonstrates the legal entanglements that arise when combining data from multiple sources under restrictive licenses, making commercial deployment legally precarious.

B. Case II: Mixture of Experts (MoE)

This case evaluates a Mixture of Experts (MoE) architecture, where a gating model is trained to manage outputs from pretrained expert models, including BLOOM, BERT, and Baize. Two deployment strategies were analyzed: code sharing (open-source) and SaaS-based service delivery.

Key Observations:

- **License incompatibility:** Behavioral use restrictions from BLOOM-RAIL-1.0 clashed with the disclosure obligations mandated by GPL-3.0 when integrated into a composite MoE model.
- **GPL-triggered disclosures:** SaaS-based deployment using GPL-licensed models with MoE logic required source code release due to the Affero-like interpretation of GPL-3.0.
- **No redistribution permitted:** Use of expert models under CC-BY-NC-ND-4.0 prevented any redistribution or adaptation, including outputs from the ensemble.

This case underscores how restrictive licensing on individual models propagates to the combined framework, significantly limiting deployment flexibility.

C. Case III: Generation Pipeline

In this scenario, models such as Stable Diffusion and Whisper were used to generate cross-modal multimedia content intended for commercial use. Licensing constraints of generative models were analyzed with respect to downstream usage.

Compliance Outcomes:

- **OpenRAIL-M constraints:** These licenses imposed specific runtime and behavioral restrictions on content generation, especially for commercial and harmful use cases.
- **Lack of control in OSS licenses:** Traditional licenses like MIT and CC (non-RAIL) did not explicitly address the legal status or permissible use of generated content.

This case highlights a gap in traditional licensing models for generative AI and emphasizes the growing role of AI-specific licenses in governing content usage.

D. Case IV: Knowledge Transfer and Fusion

This case explores legal risks in model distillation and fusion, where student models are trained using teacher models (e.g., BERT, X-Clip), and then integrated through model averaging or neural fusion strategies.

Findings:

- **Redistribution limitations:** Licenses such as Llama2 included revocable rights and prohibited sublicensing, restricting post-processing or distribution of distilled models.
- **RAIL propagation:** Behavioral constraints from RAIL licenses are applied to the distilled or fused models, inheriting ethical usage conditions.
- **Commercial restrictions:** CC-BY-NC licensed components rendered the final model unusable for commercial purposes, even when original models were indirectly reused.

This case illustrates that derivative models created through knowledge transfer still inherit the obligations of original models, particularly under remix and behavioral clauses.

E. Case V: Remixing Data Across Modalities

This scenario investigates the remixing of multimodal data from various sources using augmentation methods like mixup, synthetic generation, and domain conversion. Datasets included StockSnap (CC0), Midjourney (CC-BY-NC), Thingiverse (CC-BY-NC-SA), and Wikimedia Commons (CC-BY-SA).

ModelGo Analysis:

- **License stacking and proliferation:** Combining data under ShareAlike (SA) and NoDerivs (ND) licenses triggered cascading incompatibilities.
- **Redistribution prohibitions:** CC-BY-NC-ND-4.0 datasets blocked redistribution of any transformed artifacts, regardless of technical remixing techniques.

- **Public domain as a safe harbor:** CC0 datasets avoided conflicts but offered limited diversity and scale for training.

This case reveals the complexity of remixing data from heterogeneously licensed sources and the practical limits imposed by restrictive clauses on transformative AI workflows.

Together, these case studies demonstrate that while model and data reuse is technically feasible, legal reuse is highly dependent on the compatibility and specificity of licenses. Tools like ModelGo are essential for navigating this intricate space and ensuring compliant, scalable AI development.

VI. Ethical and Operational Implications of AI Licensing in Practice

The intersection of AI licensing and ethical AI use is becoming increasingly important as the capabilities of generative and analytical models expand. Licensing terms can serve not only as legal tools but also as normative frameworks that enforce responsible behavior among developers and downstream users. For instance, licenses like OpenRAIL explicitly encode ethical principles—such as bias mitigation, non-discrimination, and content use limitations—into legally binding terms. These behavioral clauses ensure that AI artifacts are not merely evaluated by their functionality but also by their societal impact. This marks a shift from permissive redistribution norms toward proactive social accountability, where the license itself becomes a governance mechanism. Behavioral licensing frameworks, such as those introduced by the RAIL initiative, emphasize the need for value-driven deployment. These licenses restrict use in harmful contexts (e.g., surveillance, discrimination) and mandate transparent reporting of failure modes and model behavior. The implications are far-reaching: by embedding constraints at the licensing layer, these frameworks can influence design decisions, documentation practices, and release strategies. Moreover, they foster an ecosystem where downstream users inherit a shared ethical responsibility, enhancing collective accountability in AI deployment. The legal enforceability of these terms remains an open question, but their signaling function and deterrent value are significant.

From an operational standpoint, the integration of tools like ModelGo into DevOps and MLOps pipelines enables real-time license compliance checks within CI/CD environments. By treating licensing compliance as a continuous integration task rather than a post-development audit, organizations can mitigate risks early in the lifecycle. ModelGo's structured graph analysis allows license propagation to be validated dynamically, and non-compliance can trigger automated alerts or block release candidates. This automation minimizes manual review overhead while reinforcing policy adherence across rapidly evolving model iterations. Further, automated compliance reporting and risk flagging streamline legal audit workflows and support audit-readiness for enterprises operating at scale. Integrating these tools into model registries and artifact stores ensures that all dependencies—whether open-source, proprietary, or hybrid—are consistently monitored. This becomes particularly critical when proprietary AI APIs (e.g., OpenAI, Google Cloud) are used alongside open-source components. Closed-source licenses often carry usage restrictions or hidden costs that can conflict with obligations inherited from open licenses. Such combinations may inadvertently violate redistribution clauses or impose conflicting obligations, especially in composite ML systems. Therefore, navigating these intersections requires both technical tooling and nuanced legal interpretation, underscoring the necessity of frameworks like ModelGo to harmonize ethical mandates with legal and operational constraints.

VII. Discussion

The case studies demonstrate that ML licensing compliance involves complexities not addressed by traditional OSS tools. Incompatible license interactions, undefined rights propagation, and the lack of standardized license terms for ML-specific activities significantly contribute to legal uncertainty. ModelGo addresses these challenges through a structured taxonomy, dependency analysis, and license condition mapping tailored to machine learning workflows. Across all scenarios, copyleft-style and non-commercial licenses emerged as common sources of conflict. Ambiguity in the interpretation of terms such as "adapt," "reuse," and "distribution" was particularly problematic when applied to AI activities like fine-tuning and model fusion. Moreover, AI-specific licenses such as OpenRAIL introduced behavioral constraints that are rarely captured by existing compliance tools. These observations underscore the need for license frameworks that explicitly define reuse scenarios involving models and datasets. The effectiveness of ModelGo relies on accurate license attribution and structured project metadata. In real-world applications, missing or misreported licensing information remains a significant barrier to automated compliance assessment. Future extensions of ModelGo may integrate automatic license inference techniques or collaborative license verification platforms to enhance accuracy and coverage.

VIII. Conclusion

Licensing complexities in machine learning workflows extend beyond traditional software compliance practices. We introduced **ModelGo**, a framework tailored for evaluating and ensuring license compliance in AI development. By integrating a domain-specific license taxonomy, dependency modeling, and automated analysis, ModelGo bridges the gap between legal semantics and technical implementation. Our case studies underscore their practical utility in preventing licensing conflicts. Future work will focus on real-time integration into development pipelines, expanded support for proprietary licenses, and improved license inference for undocumented components.

References

[1] D. M. German, Y. Manabe, and K. Inoue, "A sentence-matching method for automatic license identification of source code files," in *ASE*, 2010, pp. 437–446.

[2] M. C. Jaeger, O. Fendt, R. Gobeille *et al.*, "The fossology project: 10 years of license scanning," *IFOSSLR*, vol. 9, p. 9, 2017.
[3] G. M. Kapitsaki, F. Kramer, and N. D. Tselikas, "Automating the license compatibility process in open source software with spdx," *Journal of Systems and Software*, vol. 131, pp. 386–401, 2017.
[4] X. Cui, J. Wu *et al.*, "An empirical study of license conflict in free and open source software," in *ICSE-SEIP*, 2023, pp. 495–505.
[5] P. Henderson, X. Li *et al.*, "Foundation models and fair use," *arXiv preprint arXiv:2303.15715*, 2023.
[6] M. Benjamin, P. Gagnon *et al.*, "Towards standardization of data licenses: The montreal data license," *arXiv preprint arXiv:1903.12262*, 2019.
[7] S. F. Hedrick, "I think, therefore i create: Claiming copyright in the outputs of algorithms," *NYU JIPEL*, vol. 8, no. 2, pp. 324–375, 2019.
[8] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *CVPR*, 2021, pp. 10 713–10 722.
[9] D. Contractor, D. McDuff *et al.*, "Behavioral use licensing for responsible ai," in *FAccT*, 2022, pp. 778–788.
[10] C. Commons, "Artificial intelligence and cc licenses," 2023, https://creativecommons.org/faq/#artificial-intelligence-and-cc-licenses.