Anatomy of a Machine Learning Ecosystem: 2 Million Models on Hugging Face

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Foundation models are resource-intensive but broadly capable. They become specialized for downstream tasks through transformations such as fine-tuning, adaptation, and quantization. While these processes are often examined through individual evaluations or case studies, little work has explored their collective dynamics and interactions at scale. This paper analyzes 1.86 million models on Hugging Face, a leading peer production platform for model development. Our study of model family trees—networks that connect fine-tuned models to their base or parent—reveals sprawling fine-tuning lineages that vary widely in size and structure. Using an evolutionary biology lens to study ML models, we use model metadata and model cards to measure the *genetic similarity* and *mutation of traits* over model families. We find that models tend to exhibit a family resemblance, meaning their genetic markers and traits exhibit more overlap when they belong to the same model family. However, these similarities depart in certain ways from standard models of asexual reproduction, because mutations are fast and directed, such that two 'sibling' models tend to exhibit more similarity than parent/child pairs. Further analysis of the directional drifts of these mutations reveals qualitative insights about the open machine learning ecosystem: insights potentially relevant for policymakers and regulators: Licenses counter-intuitively drift from restrictive, commercial licenses towards permissive or copyleft licenses, often in violation of upstream license's terms; models evolve from multi-lingual compatibility towards english-only compatibility; and model cards reduce in length and standardize by turning, more often, to templates and automatically generated text. This work shows how platform tools shape derivative development. The structured dataset, which traces model lineage at a fine-grained level, enables deeper analysis of how models emerge and interact, offering new leverage points for policy and oversight.

The rapid development of machine learning (ML) models is changing human behaviors and systems across domains such as education and medicine. As technologies enter these domains, there is limited institutional understanding of model attributes and internals despite the widespread awareness of their possible safety risks and social stakes. One reason for this limited understanding is that many models are closed-source, meaning that changes to their weights, training data, source code, configurations, training procedures, and other details are not publicly available through documentation Wu et al. [2025], Qiu et al. [2025], Bommasani et al. [2024]. Without the ability to access these artifacts, research has predominantly focused on model outputs, benchmark performance, or individual component architectures, rather than dissecting the upstream chains of development, diffusion, and evolution towards deployment Kim et al. [2025], Raji et al. [2021]. While analogies to ecosystems and biological evolution are frequently drawn Hopkins et al. [2025], Bommasani et al. [2023], we lack a large-scale, data-driven account of the mutations of model traits as they transfer from pre-trained language models to fine-tuned bespoke products.

Submitted to Workshop on Regulatable ML at the 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Emerging open-source ecosystems offer a valuable opportunity to study machine learning from an evolutionary perspective. Hugging Face, the largest repository of open-source machine learning models, hosts roughly two million machine learning models, trained by community members for diverse tasks. In addition to rich metadata and documentation via model cards Mitchell et al. [2019], the platform records links between models, so that people can see whether one model is a derivative of another. These links enable the construction of sprawling graphs representing the models' "family trees," which can be used to systematically track ancestry, variation, and the inheritance of traits over generations of models. Existing work has called for the systematic study of these emerging lineages Horwitz et al. [2025], and some have taken steps toward understanding these emerging communities from subsets of the vast set of available models Rahman et al. [2025], Choksi et al. [2025].

Here we analyze a dataset containing the comprehensive population of 1.86 million models accessible on Hugging Face.[1] We map their lineages and measure their genetic similarities, mutation rates, and the directions of drift in traits. We find a high rate of mutation that is strongly directed, such that siblings exhibit more similarities to each other than to their parents, on average. Individual traits exhibit characteristic drifts—for example, model licenses are observed to evolve from commercial or use-restricted varieties to permissive or copyleft varieties Heffan [1996]. Language compatibility drifts from general multilingual support towards specialized, single-language support, with an overwhelming trend towards English-only compatibility. Documentation practices evolve from wide detail and coverage to lean varieties, and distinctive markers emerge among derivative models that suggest documentation is automatically generated.

These drifts in traits are predominantly acyclic, suggestive of evolutionary processes with clear directional trends. These trends yield new hypotheses about the environmental pressures on machine learning development. For instance, the observation that licenses trend towards permissiveness and copyleft varieties suggests that preferences for openness outweigh existing regulatory pressures to comply with licenses Shanklin et al. [2025]. The drift towards English-speaking models suggests a formidable market for English-language products Nicholas and Bhatia [2023], Solatorio et al. [2024].

By introducing a new methodological lens for quantifying these changes over the population of models, and by making public the largest-to-date dataset of model linkages and documents, we intend this study as a first step towards understanding the forces shaping the development and diffusion of artificial intelligence (AI) and ML. We discuss open directions for empirical and theoretical work on the evolutionary biology of these systems. These perspectives also lay the foundation for governance and regulatory approaches in which decision makers use comparative inference to design policies for models and their families.

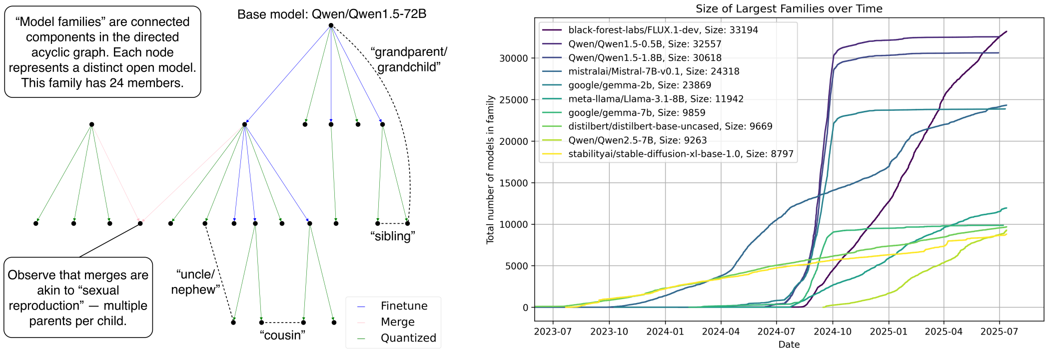
Snippets of text contain information about traits

With rich structured data about the relationships between AI models, there are a number of questions we can ask about the diffusion of model attributes. Inspired by ecological and genetic perspectives Hamilton [1964], Eberhard [1975] and existing work on network diffusion Ugander et al. [2012], we explore the relationship between *family structure* and *attribute similarity*. Our analysis centers around snippets of text for every model, known as the model's metadata and model card. Model metadata comes in a highly structured JSON format and is available for every model through Hugging Face's API. Segments of metadata are depicted in Figure 1c. The model cards, on the other hand, allow free-form text detailing model structure, performance, and other attributes, and are available for 1.25 million models.

Leveraging these snippets of text and the rich linkage structure between models, we can explore the relationship between family structure and the similarity in stored data. If finetuning family trees are akin to genetic family trees, we might expect two models finetuned from the same parent model ('siblings') to be more similar, on average, than any two models selected at random from our dataset. Taking the metaphor further, if we think of the encodings of model attributes—including licenses, tags, text data, and other metadata information—as akin to DNA in biological species, reproductive models would predict that parent-child pairs tend to be more genetically related than uncle-nephew pairs or grandparent-grandchild pairs.

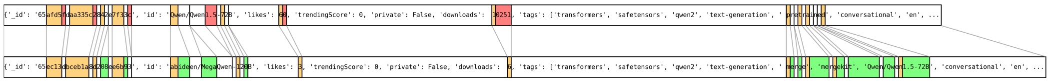
---

[1] Our dataset is publicly available at the following link: Hugging Face dataset. Our codebase is available at the following link: GitHub repository.



(a) Schematic of an example family tree from our dataset. If we compare model derivatives to biological reproduction, we might expect models that are topologically closer to exhibit similarities in traits. Our analysis aims to characterize trait inheritance and trait mutation over generations.

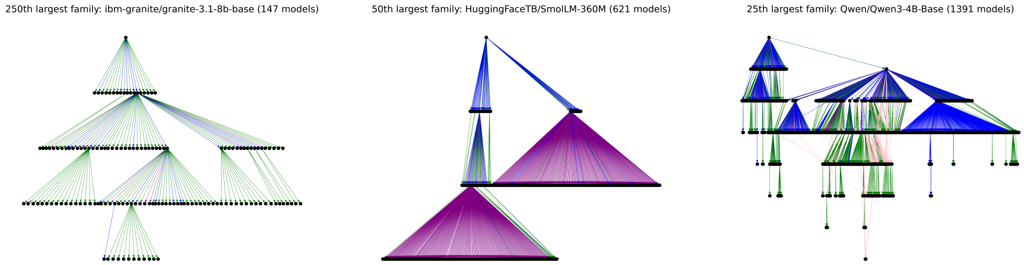
(b) The growth of the largest family trees over time using the `CreatedAt` field logged when a model is indexed on Hugging Face. The growth over time reveals "S-curve" adoption patterns Foster [1986], analogous to other domains with diffusion over a network. Merges are omitted to show growth from a single common ancestor.



```
{'_id': '65afd5fdaa335c2842e7f33c', 'id': 'Qwen/Qwen1.5-72B', 'likes': 60, 'trendingScore': 0, 'private': False, 'downloads': 10251, 'tags': ['transformers', 'safetensors', 'qwen2', 'text-generation', 'pretrained', 'conversational', 'en', ...

{'_id': '65ec13dbceb1a8d208ee6b93', 'id': 'abideen/MegaQwen-120B', 'likes': 3, 'trendingScore': 0, 'private': False, 'downloads': 6, 'tags': ['transformers', 'safetensors', 'qwen2', 'text-generation', 'merge', 'mergekit', 'Qwen/Qwen1.5-72B', 'conversational', 'en', ...
```

(c) The `diff` between two sequences of model metadata. We measure the overall mutation rate and genetic similarity by tracking rates of overlap and departure between these sequences. The top sequence is metadata from `Qwen/Qwen1.5-72B`, the base model depicted in 1a; the bottom sequence is metadata from one of its finetunes. Additions are shown in green, deletions in red, and substitutions in yellow. Here we depict character-level mutations corresponding most closely to the Levenshtein distance. We additionally measure and report similarity on term-level representations (using bag-of-words and TF-IDF), which we believe better captures categorical shifts in metadata.



(d) Examples of family trees from our dataset. Colored edges represent different forms of derivative models that are documented as having finetuned, quantized, adapter or merged pre-existing models. Diffusion patterns reveal large broadcasts and numerous generations of derivatives. All graphs are directed and acyclic. Those without merges have tree structures (left, center).

Figure 1: Model families.

In living organisms, genes are encoded in a semantic language through sequences of nucleotide bases—or "building blocks"—in DNA. One way of measuring genetic relation is by measuring the overlap or similarity in DNA sequences. AI models encode their own forms of semantically meaningful instruction sequences through their code bases, model cards, metadata, and model weights. Luckily, open models on Hugging Face make some of these resources publicly accessible, enabling formal approaches to reasoning about model similarity. Each of these artifacts is different in kind, and of course, none are perfect analogies for DNA. Here, we provide a method for measuring genetic similarity between models, inspired by the biological genetics. Our approach measures the semantic distance between the models' tokenized metadata. We propose measuring the frequency of different terms in the model metadata and tracking differences in these relative frequencies.

Measuring genetic similarity

Our aim is to measure the similarities between models residing within different 'immediate family' structures in our large tree graph. Our approach to calculating similarities borrows from classical contributions in natural language processing and formal language theory. Our broad approach uses differences in tokens between text snippets to determine the level of similarity between two such snippets. We replicate our analysis for three measures—the normalized Levenshtein Distance Yujian and Bo [2007], which directly computes character-level insertions and deletions as depicted in Figure 1c, the cosine similarity in term frequency (or "bag-of-words") embeddings, and the cosine similarity in term frequency-inverse document frequency ("TF-IDF") embeddings. We measure similarities across two different model artifacts—metadata JSONs and the text of model cards. Our results across these different metrics reveal the same pattern: that models of the same finetuning family tree are more genetically similar than randomly paired models, and that genetic similarity is negatively related to the generational divide and topological distance. Further information on these various metrics and approaches are provided in Appendix C. In the body of the text, we report the cosine similarity on TF-IDF embeddings derived from the metadata strings (Figure 2).

To understand the qualities of genetic similarity in model family trees, we first limit our analysis to a specific type of family relation—*finetuned models*—to control for specific similarity patterns within groups and because the other relations rarely produce offspring of their own. This also allows us to work with a tree graph, meaning we avoid cases where merged models have more than one parent and thus relate to other models in more than one way. Our analysis consists in enumerating the possible graph relationships between model pairs—parent/child, grandparent/grandchild, sibling, uncle/nephew, and so forth (illustrated in Figure 1b. To be comprehensive, we enumerate all possible local family structures of size one, two, three and four, and within these structures, we aim to measure the attribute similarity between every possible pair of nodes. A depiction of these possible structures is provided in Figures 4 and 2. The challenge with estimating quantities over these local structures is that they may appear combinatorically many times in a large graph, creating algorithmic challenges Clauset [2005], Kleinberg et al. [1999]. To illustrate what we mean by this, consider the set of all pairs of siblings in a tree graph. If one model has 500 children, the total number of pairs of siblings among them is $\binom{500}{2}$ or $124,750$. Therefore, estimating the typical similarity over all pairs of siblings quickly becomes computationally burdensome. To handle this challenge, we design an estimation procedure, where we draw a representative sample from the set of all pairs of models meeting the relationship condition. We implement our estimation procedure, sampling 5000 pairs of nodes for each distinct subtree topology and pair combination. We then construct 95% confidence intervals around our mean similarity estimates treating our graph-wide measurement as the population mean and drawing 10,000 bootstrap subsamples from our sample with replacement.

Family resemblance and diffusion characteristics

Our main results are depicted in Figure 2. The results suggest that models that are close in network topology have considerably more similarity than randomly selected pairs of nodes. This offers some evidence that model family trees truly do exhibit family resemblances. However, patterns of similarity over family trees are not cleanly predicted by typical models of genetic diffusion. For example, we find that siblings are significantly more similar to one another than either is to its parent, on average (depicted in the first subfigure labeled 'C'). This is counter to what an asexual model of genetic reproduction with mutation might predict. If we imagine each child model in a family inheriting

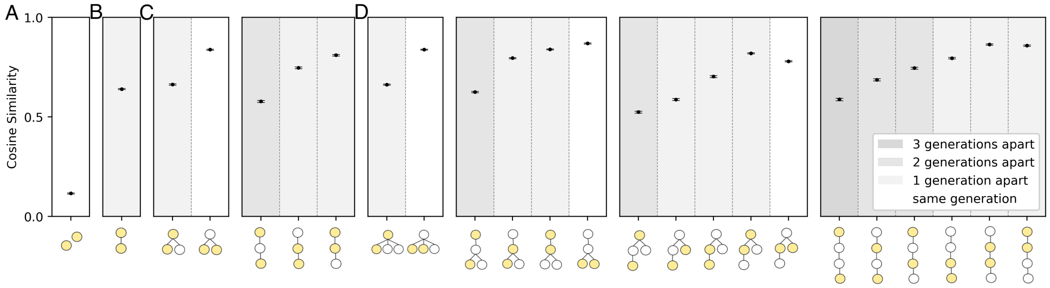


Figure 2: Cosine similarity between TF-IDF embedding vectors, trained on the model metadata strings for all models in our dataset. Here, we sample finetunes meeting specific family structures. We enumerate all possible sub-trees of size 2 (B), 3 (C), and 4 (D), and enumerate all possible pairs of nodes within each sub-tree. When we compare these genetic similarities to the baseline of the similarity between any two nodes in the graph (A), we find that all observed family ties strongly predict attribute similarity. Similarities between pairs of models suggest that models are more related when they reside at similar depths and when they are topologically close in distance. Mean estimates are from samples of 5000; confidence intervals are calculated over $10,000$ bootstrap draws from the sample.

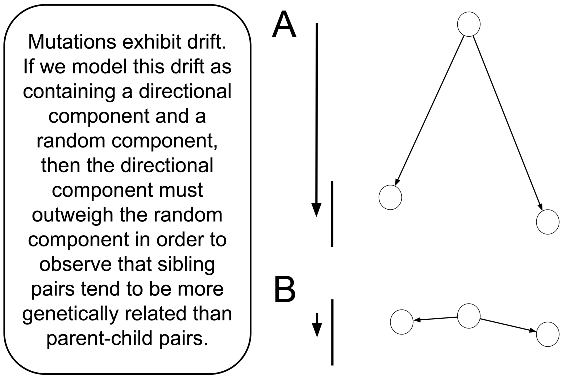


Figure 3: We observe that siblings exhibit greater similarity in traits than parent-child pairs. This implies not only that there is a high rate of mutation, but that mutations are sufficiently directed.

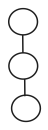
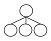
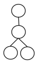
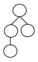
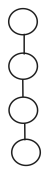
Topology	**Occurrences**
	3,470,193,356,870
	191,072
	119,795,843
	40,922
	193,010,561,824
	11,847,103
	19,932,645
	10,965

Figure 4: The graph contains many instances of some family subtrees. Pairwise similarities within subtrees are estimated via sampling.

the parent's genes subject to some rate of random mutation, siblings should be more related to their parent than each other, on average. We observe the opposite, suggesting that there is some directional effect of fine-tuning whereby all children tend to depart in attributes from their parents, on average, in characteristically similar ways (illustrated in Figure 3).

When we look at pairs of nodes in a variety of subgraphs, we see evidence of three major heuristics that seem to dictate the level of similarity between pairs of models. The first is being members of the *same family*. If models belong to the same family tree, they appear to exhibit significantly higher levels of similarity, compared to models paired at random over our dataset.

The second factor that appears relevant to trait similarity is *generational divide*. When we compare two models that are the same *generation* in their family tree (e.g., siblings or cousins), we find that this majorly increases the level of similarity between models. Models that are one generation apart (e.g., parent/child pairs or uncle/nephew pairs) tend to be significantly more similar, on average, than models that are two generations apart (e.g., grandparent/grandchild pairs). The same relationship holds when comparing grandparent/grandchild pairs to great-grandparent/great-grandchild pairs.

A third heuristic that seems to explain the observed similarities in model attributes is the *network distance*, that is, the total number of edges one would need to traverse to get from one node pair to the other. This is what a genetic model of mutation-based asexual reproduction would predict. This factor is supported by the fact that uncle/nephew pairs are observed to be less similar, on average, than parent/child pairs belonging to the same subgraph structures (depicted in the second and third columns of subfigure D3 in Figure 2). Though most measures suggest generational divide outweighs network distance in importance, there is one exception: In the last two similarity measures in D3, we observe a parent-child pair with network distance one exhibits higher similarity than a sibling pair with network distance two.

Evolution of traits

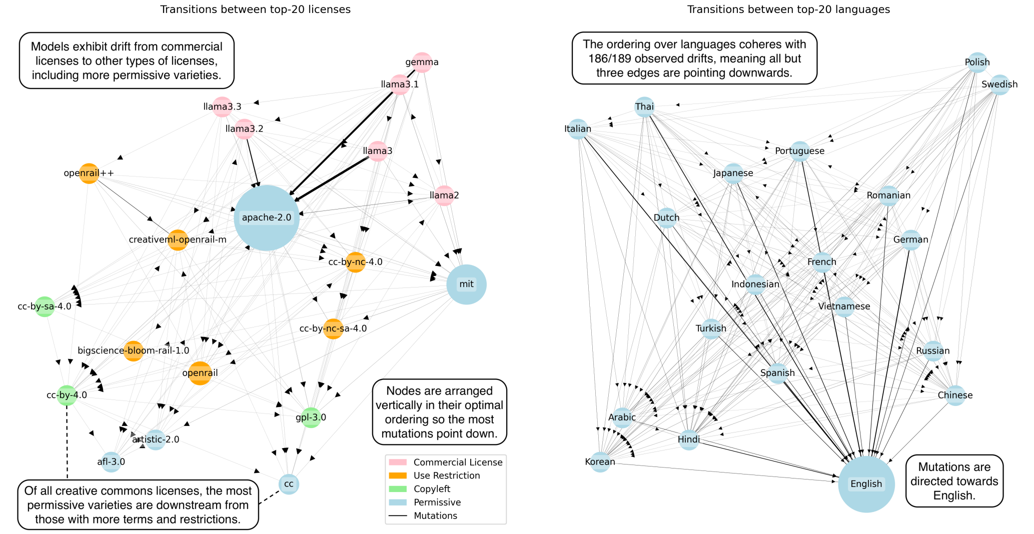
The previous sections examined overall similarities between models across their recorded features. We now turn our attention to *individual traits* arising from structured sequences of the model metadata. In many cases, traits remain the same between parent and child. However, if traits were *always* constant between parent and child, we'd observe far less heterogeneity in our data, and we'd find perfect similarity across all related model pairs in Figure 2. Because we do, in fact, observe feature diversity across models, here we focus on cases where model traits *change* between a parent and a child, that is, cases where the parent has trait i, the child holds trait j, and $i \neq j$. Further information about our formal way of defining the rate of mutation is provided in Appendix D. In observing these instances of mutation, we make a number of specific observations and findings pertaining to the individual traits in question, which we discuss in the proceeding sections.

At a general level, we make two empirical observations that hold descriptively, but are not necessary or obvious. First, we observe that mutations tend to be *directed*. Formally, for any two traits (i, j), it is most common that i mostly mutates to j or that j mostly mutates to i, rather than some balance of 'traffic' of mutations in both directions. We refer to any imbalance in the direction of mutation as a *drift*. Second, when we consider the orientations of all directed mutations, we find that these orientations are *ordered*. If we define the oriented graph of 'typical' transitions between traits, we are able to find orderings over these transitions that explain virtually all these orientations.

Notice that the first observation does not imply the second. It could be that i mostly mutates to j, which mostly mutates to k, which mostly mutates back to i. We do not observe this for the vast majority of drifts. Second, we note that the task of finding an ordering over a directed graph is an integer programming problem, NP-hard in the worst cases. Our implementations are able to find optimal orderings, not due to luck but due to the natural orderings that emerge from our oriented graphs.

Licenses drift from commercial to permissive and copyleft.

How do license assignments change and mutate across model lineages? Our analysis of the direction of evolution of licenses is summarized in Figures 5a. Figure 5a depicts the twenty most-common licenses and the 'drifts' between them—that is, the arrows point in the more frequent mutation direction over all observed mutations. The graph is an oriented directed graph of all 121 drifts between 20 traits, where edge weight depends on the total traffic of mutations. Using the graph, we can ask, what ordering over traits is most compatible with these drifts? If mutations were fully random, or if cycles were common, we would not be able to produce an ordering that captures more than approximately half of the observed mutation directions. However, we are able to produce an ordering accounting for 93% of all drift directions, and 85% of all mutations. This suggests a strong directedness in the evolution of licenses. Equipped with this ordering, we can begin to develop hypotheses about the environmental pressures leading to the observed evolution.



Models exhibit drift from commercial licenses to other types of licenses, including more permissive varieties.

Nodes are arranged vertically in their optimal ordering so the most mutations point down.

Of all creative commons licenses, the most permissive varieties are downstream from those with more terms and restrictions.

The ordering over languages coheres with 186/189 observed drifts, meaning all but three edges are pointing downwards.

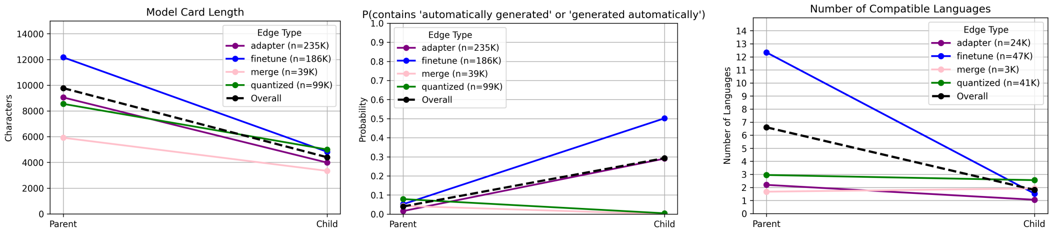
Mutations are directed towards English.

(a) Trait evolution for licenses over fine-tune family trees. Typical mutation directions suggest families often start with commercial licenses and fine-tune to others. Many instances show licenses getting less strict or dropping upstream terms.

(b) Trait evolution for language compatibilities over fine-tune family trees. The graph is fully connected and shows a marked drift toward English-only support.

Trait	Observed inheritances	Mutation rate	Drifts compatible with order	Mutations compatible with order
License	138,694	19.76%	113/121 (93.39%)	84.61%
Language	115,660	12.80%	186/189 (98.41%)	74.99%

(c) Summary statistics on the evolution of licenses and languages.



(d) Model card length decreases. (e) Evidence of automation increases. (f) Language support declines.

Figure 5: Typical mutation directions reveal emerging patterns in trait evolution across fine-tune family trees.

Perhaps surprisingly, we observe many instances in which the more restrictive, commercial licenses are *upstream* from the more permissive licenses.[2] Consider, as one example, the `gemma` license, which appears first in our observed ordering. The terms of this license include the following requirement: "You must provide all third party recipients of Gemma or Model Derivatives a copy of this Agreement." The license lists further restrictions, including on uses that "sexually explicit content, including content created for the purposes of pornography or sexual gratification (e.g. sexual chatbots)" Google LLC [2025]. This license mutates most frequently to `Apache-2.0` and `MIT` licenses, each which contain no such provisions. As a second example, we observe mutation drifts from `cc-by-nc-4.0`, a creative commons license that restricts derivatives from commercial uses, to `MIT`, which grants permissions "without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell"

---

[2] When we refer to the categories of permissive, restrictive, commercial, and copyleft, we are using categorizations from existing scholarship, including Longpre et al. [2024] in the context of data provenance and Choksi and Grimmelmann [2024] in the context of open-source software.

Open Source Initiative [1988]. The same non-commercial license also mutates to other licenses of the same variety (creative commons) but without the non-commercial agreement, which seems to be a strict relaxation of terms.

These instances of 'relaxations' appear to be the norm rather than the exception. Of the first five licenses in our ordering, all are commercial (`gemma` or `llama` varieties). Of the last five licenses in our ordering, the last three are permissive or public domain (`cc`, `afl-3.0`, `artistic-2.0`) and the remaining two are copyleft without any restrictions on use (`cc-by-*`, `gpl-3.0`). Looking exclusively at creative commons licenses, non-commercial restrictions and share-alike provisions lie upstream from versions without these previsions.

Why would licenses weaken and relax even when doing so might constitute a violation of upstream agreement terms? The observed mutation drift suggests market and behavioral pressures toward openness outweigh the specter of legal enforcement as a motivator for AI developers.

Documentation thins.

We now turn our attention to information from the model cards. We are interested in the effort and resources devoted to documentation and transparency for models of different generations in the open source ecosystem. One significant trend that we observe is that documentation thins. Markers of bespoke effort aimed at supporting users, communicating methods, and demonstrating capabilities seem to atrophy. Markers of leaner approaches and automation develop and multiply.

When we look at the state of model cards between parents and children in our family trees, we can make a few straight-forward observations. Model cards exist at a very high rate for models that belong to family trees. Missing model cards are far more frequent among models with no family ties. Among models with family ties, the model card is almost always available, even if it is only a few characters long. Among parent-child pairs with model cards, we observe that the length of these cards drops by $\approx 5,000$ characters. The parent's model card is roughly twice the size of the child's model card, on average. Even though the model cards get significantly shorter, we observe that they more frequently contain the terms that suggest automatic card generation. About 30% of derivative models contain the bigrams `automatically generated` or `generated automatically`. These results, depicted in Figures 5d and 5e, suggest pressures toward lean documentation and automation technologies that remove costs to document and explain models, their capabilities, their uses, and other information typically contained in the model card.

Languages specialize and drift toward English.

Language traits are different in kind from other categorical features because an individual model can be compatible with more than one language, meaning that partial mutations are possible. Consider a case where model i finetunes to model j. Model i has language group (A, B, C) and j has language group (B, C, D). We say that the overall mutation rate is the shared members of both groups divided by the union of both groups (i.e., in this example, the mutation rate would be $\frac{1}{2}$). Further, we log distinct directional mutations from every dropped language to every child language, and from every parent language to every added language. To continue our example, we'd log mutations from A to B, C, and D and from A, B and C to D. These enable us to produce similar drift diagrams and orderings to those produced for licenses. Our findings are summarized in Figures **??** and 5f.

The language traits show two trends: 1) specialization and 2) drift towards English. The first of these trends, specialization, refers to the significant reduction in language compatibility from base models to child models. Large base models supporting significant family trees tend to support many languages, whereas derivative models tend to list compatibility with one or a handful of languages. Therefore, we see a precipitous reduction in the language support between parents and finetuned children.

The second observation we can make about language traits is that they drift overwhelmingly from broad language support to English-language support. This drift suggests a considerable market pressure towards English-speaking products and compatibilities. This drift is not entirely surprising given Hugging Face is a United States-based company. However, an increasing number of Chinese models are being developed and hosted and we do not observe a commensurate drift towards Chinese compatibility.

Discussion

Limitations

Limitations to our findings include the fact that we only account for models that have logged fine-tuning relationships on Hugging Face. Many models may be related without having these relationships. For instance, models released with different numbers of paramaters are often each available as their own base model, so we do not consider `Qwen/Qwen1.5-0.5B` and `Qwen/Qwen1.5-1.8B` to be members of the same family. Though we use metadata and model card snippets as metaphors for DNA, there are other sources of semantic information we do not access. Future work may analyze model repositories' `config.JSON` files to extract architectural parameters, such as vocabulary size (inferring the training dataset size and costs), attention heads, and hidden dimensions, to reveal further attributes of models and trace how structural traits evolve across the ecosystem. Text from code repositories and even the model weights themselves could contain additional low-level semantic encodings of model properties and internals. Finally, the timescale of this analysis is limited to the lifespan of the Hugging Face platform. However, since open models predate Hugging Face, future work could extend this analysis by incorporating historical data from earlier model repositories and academic publications to capture the complete evolutionary trajectory of open source ML.

Changes to the Hub interface (e.g., available fields and auto-generated documentation) affect what developers report about their models. For example, the `CreatedAt` field was introduced in March 2022 and all existent models were back-filled values equal to the date of feature launch. This could possibly inflate the genetic similarity between otherwise unrelated models.

We see the present work as a first step towards a range of studies that this dataset and perspective could support. For example, though our data represent a snapshot in which models exhibit fixed qualities, there are a variety of attributes that may be time-dependent or trends that could be uncovered with time-series data. Further, where our approach focused on open source models, there is a huge industry of closed models and these ecosystems have interesting interactions.

Structural complexity arises not only from the number of descendants but also from the introduction of merges, which combine distinct lineages—essentially 'marrying families.' Mergers between models could be viewed as a form of 'sexual reproduction,' contrasting with the one-to-one parent–child mappings that this paper focuses on. As merges become more popular, the Hugging Face graph may undergo a phase transition in which nearly all nodes become connected in a single, massive connected component. Further analysis is needed to understand model merges and their effect on the ecosystem.

Future Work

This work opens several research trajectories. Future studies could extend the ecological framing to processes such as niche formation, cooperation, and succession, helping explain how model families grow, stabilize, or die out. Another direction is to investigate malignant behaviors in model lineages, such as catastrophic inheritance Chen et al. [2024] - and to investigate whether these can be inferred from higher-level properties of parent models, potentially offering a more resource-efficient alternative to assessing model capabilities than bottom-up approaches Sharma et al. [2025].

Governance Implications for ML Platforms

Our empirical findings highlight three governance challenges for ML platforms. First, as development becomes more decentralized, platforms must track model dependencies, document performance updates, and ensure backward compatibility, with emerging solutions including Model ChangeLists Eyuboglu et al. [2024], analogous to Software Bills of Materials (SBOMs) Cyb [2025]. Second, while few models release code, those that do often expose vulnerabilities such as leaked API keys and credentials, underscoring the need for stronger review practices. Third, documentation standards like Croissant, a metadata format for ML-ready datasets Akhtar et al. [2024] now required for NeurIPS, could consolidate practices across platforms, though trade-offs remain between transparency and developer burden.

References

Framing software component transparency, Feb 2025. URL `https://www.cisa.gov/resources-tools/resources/framing-software-component-transparency-2024`.

Mubashara Akhtar, Omar Benjelloun, Costanza Conforti, Pieter Gijsbers, Joan Giner-Miguelez, Nitisha Jain, Michael Kuchnik, Quentin Lhoest, Pierre Marcenac, Manil Maskey, Peter Mattson, Luis Oala, Pierre Ruyssen, Rajat Shinde, Elena Simperl, Goeffry Thomas, Slava Tykhonov, Joaquin Vanschoren, Jos van der Velde, Steffen Vogler, and Carole-Jean Wu. Croissant: A metadata format for ml-ready datasets. In *Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning*, SIGMOD/PODS '24, page 1–6. ACM, June 2024. doi: 10.1145/3650203.3663326. URL `http://dx.doi.org/10.1145/3650203.3663326`.

Rishi Bommasani, Dilara Soylu, Thomas I Liao, Kathleen A Creel, and Percy Liang. Ecosystem graphs: The social footprint of foundation models. *arXiv preprint arXiv:2303.15772*, 2023.

Rishi Bommasani, Sayash Kapoor, Kevin Klyman, Shayne Longpre, Ashwin Ramaswami, Daniel Zhang, Marietje Schaake, Daniel E Ho, Arvind Narayanan, and Percy Liang. Considerations for governing open foundation models. *Science*, 386(6718):151–153, 2024.

Joel Castaño, Silverio Martínez-Fernández, Xavier Franch, and Justus Bogner. Analyzing the evolution and maintenance of ml models on hugging face. In *Proceedings of the 21st International Conference on Mining Software Repositories*, pages 607–618, 2024.

Hao Chen, Bhiksha Raj, Xing Xie, and Jindong Wang. On catastrophic inheritance of large foundation models, 2024. URL `https://arxiv.org/abs/2402.01909`.

Madiha Zahrah Choksi and James Grimmelmann. How licenses learn. *Lewis & Clark L. Rev.*, 28: 249, 2024.

Madiha Zahrah Choksi, Ilan Mandel, and Sebastian Benthall. The brief and wondrous life of open models. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, pages 3224–3240, 2025.

Aaron Clauset. Finding local community structure in networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 72(2):026132, 2005.

Sarah Dean, Evan Dong, Meena Jagadeesan, and Liu Leqi. Accounting for ai and users shaping one another: The role of mathematical models. *arXiv preprint arXiv:2404.12366*, 2024.

Moming Duan, Mingzhe Du, Rui Zhao, Mengying Wang, Yinghui Wu, Nigel Shadbolt, and Bingsheng He. Position: Current model licensing practices are dragging us into a quagmire of legal noncompliance. In *Forty-second International Conference on Machine Learning Position Paper Track*.

Mary Jane West Eberhard. The evolution of social behavior by kin selection. *The Quarterly Review of Biology*, 50(1):1–33, 1975.

Sabri Eyuboglu, Karan Goel, Arjun Desai, Lingjiao Chen, Mathew Monfort, Chris Ré, and James Zou. Model changelists: Characterizing updates to ml models. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, pages 2432–2453, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704505. doi: 10.1145/3630106.3659047. URL `https://doi.org/10.1145/3630106.3659047`.

Hugging Face. Models - Hugging Face — huggingface.co. `https://huggingface.co/models?library=safetensors`, 2022. [Accessed 21-07-2025].

Richard N. Foster. Assessing technological threats. *Research Management*, 29(4):17–20, 1986. ISSN 00345334. URL `http://www.jstor.org/stable/24121836`.

David Gibson, Jon Kleinberg, and Prabhakar Raghavan. Inferring web communities from link topology. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia: links, objects, time and space—structure in hypermedia systems: links, objects, time and space—structure in hypermedia systems*, pages 225–234, 1998.

Sharad Goel, Duncan J Watts, and Daniel G Goldstein. The structure of online diffusion networks. In *Proceedings of the 13th ACM conference on electronic commerce*, pages 623–638, 2012.

Sharad Goel, Ashton Anderson, Jake Hofman, and Duncan J Watts. The structural virality of online diffusion. *Management science*, 62(1):180–196, 2016.

Claudia Goldin and Lawrence Katz. The origins of technology-skill complementarity. Technical report, National Bureau of Economic Research, Cambridge, MA, July 1996.

Google LLC. Gemma terms of use, March 2025. URL `https://ai.google.dev/gemma/terms`. Accessed: 2025-08-11.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

William D Hamilton. The genetical evolution of social behaviour. ii. *Journal of theoretical biology*, 7 (1):17–52, 1964.

William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

Ira V Heffan. Copyleft: licensing collaborative works in the digital age. *Stan. L. Rev.*, 49:1487, 1996.

Aspen Hopkins, Sarah H Cen, Andrew Ilyas, Isabella Struckman, Luis Videgaray, and Aleksander Madry. Ai supply chains: An emerging ecosystem of ai actors, products, and services. *arXiv preprint arXiv:2504.20185*, 2025.

Eliahu Horwitz, Nitzan Kurer, Jonathan Kahana, Liel Amar, and Yedid Hoshen. Charting and navigating hugging face's model atlas. *arXiv preprint arXiv:2503.10633*, 2025.

Meena Jagadeesan, Michael I Jordan, and Jacob Steinhardt. Safety vs. performance: How multi-objective learning reduces barriers to market entry. *arXiv preprint arXiv:2409.03734*, 2024.

Pratyusha Ria Kalluri, William Agnew, Myra Cheng, Kentrell Owens, Luca Soldaini, and Abeba Birhane. Computer-vision research powers surveillance technology. *Nature*, pages 1–7, 2025.

Elliot Kim, Avi Garg, Kenny Peng, and Nikhil Garg. Correlated errors in large language models. *arXiv preprint arXiv:2506.07962*, 2025.

Jon M Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S Tomkins. The web as a graph: Measurements, models, and methods. In *International Computing and Combinatorics Conference*, pages 1–17. Springer, 1999.

Gueorgi Kossinets and Duncan J Watts. Empirical analysis of an evolving social network. *science*, 311(5757):88–90, 2006.

Benjamin Laufer, Jon Kleinberg, and Hoda Heidari. Fine-tuning games: Bargaining and adaptation for general-purpose models. In *Proceedings of the ACM Web Conference 2024*, pages 66–76, 2024.

Benjamin Laufer, Jon Kleinberg, and Hoda Heidari. The backfiring effect of weak ai safety regulation. *arXiv preprint arXiv:2503.20848*, 2025.

Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650, 2010.

Simon A Levin. Ecosystems and the biosphere as complex adaptive systems. *Ecosystems*, 1(5): 431–436, 1998.

David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559, 2003.

Shayne Longpre, Robert Mahari, Anthony Chen, Naana Obeng-Marnu, Damien Sileo, William Brannon, Niklas Muennighoff, Nathan Khazam, Jad Kabbara, Kartik Perisetla, Xinyi Wu, Enrico Shippole, Kurt Bollacker, Tongshuang Wu, Luis Villa, Sandy Pentland, and Sara Hooker. A large-scale audit of dataset licensing and attribution in AI. *Nat. Mach. Intell.*, 6(8):975–987, August 2024.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229, 2019.

Gabriel Nicholas and Aliya Bhatia. Lost in translation: Large language models in non-english content analysis, 2023. URL `https://arxiv.org/abs/2306.07377`.

Open Source Initiative. Mit license, 1988. URL `https://opensource.org/licenses/MIT`.

Tori Qiu, Benjamin Laufer, Jon Kleinberg, and Hoda Heidari. A formal model of the economic impacts of ai openness regulation. *arXiv preprint arXiv:2507.14193*, 2025.

Mohammad Shahedur Rahman, Peng Gao, and Yuede Ji. Hugginggraph: Understanding the supply chain of llm ecosystem. *arXiv preprint arXiv:2507.14240*, 2025.

Inioluwa Deborah Raji, Emily M Bender, Amandalynne Paullada, Emily Denton, and Alex Hanna. Ai and the everything in the whole wide world benchmark. *arXiv preprint arXiv:2111.15366*, 2021.

Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data, 2024. URL `https://arxiv.org/abs/2402.16991`.

Grant Shanklin, Emmie Hine, Claudio Novelli, Tyler Schroder, and Luciano Floridi. The case for contextual copyleft: Licensing open source training data and generative ai, 2025. URL `https://arxiv.org/abs/2507.12713`.

Mrinank Sharma, Meg Tong, Jesse Mu, Jerry Wei, Jorrit Kruthoff, Scott Goodfriend, Euan Ong, Alwin Peng, Raj Agarwal, Cem Anil, Amanda Askell, Nathan Bailey, Joe Benton, Emma Bluemke, Samuel R. Bowman, Eric Christiansen, Hoagy Cunningham, Andy Dau, Anjali Gopal, Rob Gilson, Logan Graham, Logan Howard, Nimit Kalra, Taesung Lee, Kevin Lin, Peter Lofgren, Francesco Mosconi, Clare O'Hara, Catherine Olsson, Linda Petrini, Samir Rajani, Nikhil Saxena, Alex Silverstein, Tanya Singh, Theodore Sumers, Leonard Tang, Kevin K. Troy, Constantin Weisser, Ruiqi Zhong, Giulio Zhou, Jan Leike, Jared Kaplan, and Ethan Perez. Constitutional classifiers: Defending against universal jailbreaks across thousands of hours of red teaming, 2025. URL `https://arxiv.org/abs/2501.18837`.

Aivin V. Solatorio, Gabriel Stefanini Vicente, Holly Krambeck, and Olivier Dupriez. Double jeopardy and climate impact in the use of large language models: Socio-economic disparities and reduced utility for non-english speakers, 2024. URL `https://arxiv.org/abs/2410.10665`.

Boaz Taitler and Omer Ben-Porat. Selective response strategies for genai. *arXiv preprint arXiv:2502.00729*, 2025.

Johan Ugander, Lars Backstrom, Cameron Marlow, and Jon Kleinberg. Structural diversity in social contagion. *Proceedings of the national academy of sciences*, 109(16):5962–5966, 2012.

David Gray Widder and Dawn Nafus. Dislocated accountabilities in the "ai supply chain": Modularity and developers' notions of responsibility. *Big Data & Society*, 10(1):20539517231177620, 2023.

Yanxuan Wu, Haihan Duan, Xitong Li, and Xiping Hu. Navigating the deployment dilemma and innovation paradox: Open-source versus closed-source models. In *Proceedings of the ACM on Web Conference 2025*, pages 1488–1501, 2025.

Jiang Yang and Scott Counts. Predicting the speed, scale, and range of information diffusion in twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 4, pages 355–358, 2010.

Takeshi Yoshimura, Tatsuhiro Chiba, Manish Sethi, Daniel Waddington, and Swaminathan Sundararaman. Speeding up model loading with fastsafetensors, 2025. URL `https://arxiv.org/abs/2505.23072`.

Li Yujian and Liu Bo. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095, 2007.

A Technical Appendices and Supplementary Material

B Further related work

This paper aims to measure and analyze the structure of AI *fine-tuning* and related adaptation and transfer learning procedures. These relationships connect finetuned and remixed AI models to their 'parent' model(s) whose weights, structures, and other elements might influence the child's development. The sources of inspiration for this work come from scholarship on **social networks and the web**, **multi-agent interactions and modeling AI development**, and finally, **approaches from theoretical ecology and genetics**. We cover relevant work from each of these categories in turn.

Social Networks on the Web. This work takes a quantitative approach to networks of viral propagation over the web, evocative of literature on virality and social media Kossinets and Watts [2006], Goel et al. [2012], Yang and Counts [2010] Goel et al. [2016] differentiate broadcast diffusion trees from viral trees using a metric they term structural virality – this concept helped inspire our work because we were surprised that fine-tuning trees are not exclusively broadcast graphs.Many have considered the dependence of graph features on local network topology, including in the context of attachment Ugander et al. [2012], link prediction Liben-Nowell and Kleinberg [2003], Leskovec et al. [2010], feature prediction Grover and Leskovec [2016], Hamilton et al. [2017] and community inference Gibson et al. [1998]. In contrast, our approach attempts to predict trait similarity and trait transitions over a tree network. Though empirical work on Hugging Face is limited, some strides have been made. Horwitz et al. [2025] calls for work mapping an 'atlas' of models on Hugging Face, demonstrating that directed acyclic graphs representing model relationships can be drawn for certain families and providing a dataset with 1.1 million models. Our work answers this call and offers an expanded dataset. Castaño et al. [2024] analyze the growth over time and commit patterns using the Hugging Face model hub, gathering a dataset of 380,000 models. Choksi et al. [2025] explore chats and conversations among community members and contributors, evidence of vibrancy and richness among contributing developers. Bommasani et al. [2023] coin *ecosystem graphs* as an abstraction for understanding AI development, and analyze a preliminary set of 128 models that they use to demonstrate the usefulness of ecosystems thinking for reasoning about social implications and regulation of AI. Duan et al. tracks the frequency of copyleft license violations across model derivatives using a dataset of around 15,000 models on Hugging Face. Rahman et al. [2025] use the Hugging Face API to create a graph of information about models totaling 402,654 nodes.

Multi-agent interactions and modeling. Scholars have developed theoretical models and theories of the multi-actor system surrounding the development of AI technologies. Laufer et al. [2024] create a game-theoretic model to understand how 'domain specialists' and 'generalists' interact to produce the technology. Others have developed depth-one tree structures as a model for understanding AI diffusion Jagadeesan et al. [2024], Qiu et al. [2025], Dean et al. [2024], Laufer et al. [2025]. Hopkins et al. [2025] use directed acyclic graphs (DAGs) of arbitrary depth to allow supply chains of interacting actors to understand the dynamics of AI supply chains. There is budding work on decision-making along these networks Widder and Nafus [2023], Taitler and Ben-Porat [2025], though much of it is theoretical. Further, we claim that perspectives on incentives, competition, cooperation have tended to be organized by economic—rather than ecological—metaphors. Here, we wish to go deeper with the ecological phenomenology of AI development and diffusion.

Theoretical Ecology and Genetics. This paper is inspired by perspectives of systems as *complex adaptive systems*, characterized by emergent properties that arise from small-scale interactions between components Levin [1998]. Sclocchi et al. [2024], taking a machine learning perspective, understand model 'phylogeny' as a prediction problem, and show that models with larger normed parameter vectors—weights and biases of greater magnitude—tend to be higher up in the family tree.

In a different genealogical approach to machine learning, Kalluri et al. [2025] draw links between ML papers and downstream produce developments, focusing on surveillance applications.

C Defining measures of genetic similarity

Here we provide additional details on how we measure genetic similarity between models, and we report results across the range of measures we define.

Our Figure 2 shows one of six ways we measure genetic similarity between models. These six methods align in the general trends and interpretations reported in the paper. Here we provide details on all six.

The measures can be divided by two *targets* of similarity analysis—the metadata and model cards. On each of these pieces of text, we implement three distinct measures. One measure—Levenshtein distance—computes the total character-by-character difference. The other two—Bag-of-words (BOW) and Term Frequency-Inverse Document Frequency (TF-IDF)—measure differences using the set of n-grams in the text.

C.1 Formal definitions

Below we state the formal definitions of our various measures of genetic similarity. All take as input a pair of strings s_1, s_2 and output a measure, between 0 and 1, of similarity between them.

Definition C.1 (Cosine similarity in term frequency)**.** Given two strings (s_1, s_2) in a set of strings S, we compute the **cosine similarity in term frequency** as follows. Over all strings in S, produce an ordered list of the n most frequently appearing terms (unigrams or bigrams). Then, for any string $s_i \in S$, define the vector $v_i \in \mathbb{R}^n$ such that every value $v_i[k]$ is the number of times the k^{th} term in the list appears in s_i. The similarity is $\frac{v_i v_j}{||v_i||||v_j||}$.

Definition C.2 (Cosine similarity in term frequency-inverse document frequency)**.** Given two strings (s_1, s_2) in a set of strings S, we compute the **cosine similarity in TF-IDF** as follows. Over all strings in S, produce an ordered list of the n most frequently appearing terms (unigrams or bigrams). Then, for any string $s_i \in S$, define the vector $v_i \in \mathbb{R}^n$ such that every value $v_i[k]$ is the product of the number of times the k^{th} term appears in s_i (its term frequency) and the inverse of the fraction of documents $s \in S$ which contain the term (its inverse document frequency). The similarity is $\frac{v_i v_j}{||v_i||||v_j||}$.

Definition C.3 (Normalized Levenshtein Similarity)**.** Given two strings (s_1, s_2), we define the normalized Levenshtein distance (NLD) as the minimum number of character-wise insertions, deletions, or substitutions to transform s_1 into s_2, divided by $\max(\text{length}(s_1), \text{length}(s_2))$. The **normalized Levenshtein distance** is $1 - \text{NLD}$.

The above definitions can be computed for a general set of strings, and we report results comparing two sets of strings specifically: The metadata, which is highly structured and recorded for every model on Hugging Face, and the model cards, which is unstructured, much more variable in length, and missing for roughly a third of all models. In the body of the text, we report results on the metadata.

C.2 Why we prefer term frequency based similarity metrics to edit distances

We report the TF-IDF similarities in the body of the paper, and the other similarity metrics (which match in qualitative conclusions) in the appendix. We do this for two reasons. First, we believe mutations over the metadata are more a function of differences in term-based tokens rather than character-based tokens. The difference between the snippets '`license: mit`' and '`license: gemma`' should not depend on how many letters 'mit' and 'gemma' share. Further, the use of traits that happen to have long names does not correspond to a further genetic distance in a meaningful. For instance, the tasks '`reinforcement-learning`' and '`fill-mask`' are not different because of the number of character deletions they require; rather they are different because they are different terms. Second, Levenshtein distance is significantly affected by the ordering of terms, such that the existance of a long tag somewhere in the middle of the string could skew the distance measure. We believe these attributes are much more a function of whether their semantic markers *appear* in the metadata, and less a function of their *ordering* in the metadata. This is why we prefer term frequency based measures. Finally, we choose to report the measures normalized by inverse-document frequency because it is a norm in the field, but generally we note that our qualitative insights and interpretations are consistent across the proposed measures.

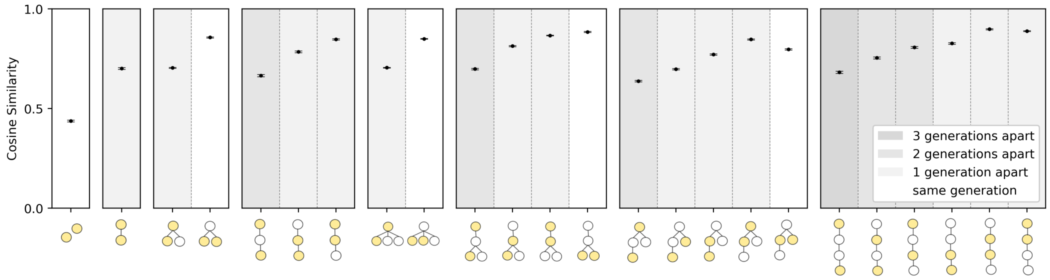


Figure 6: Bag of Words Cosine Similarity, Metadata.

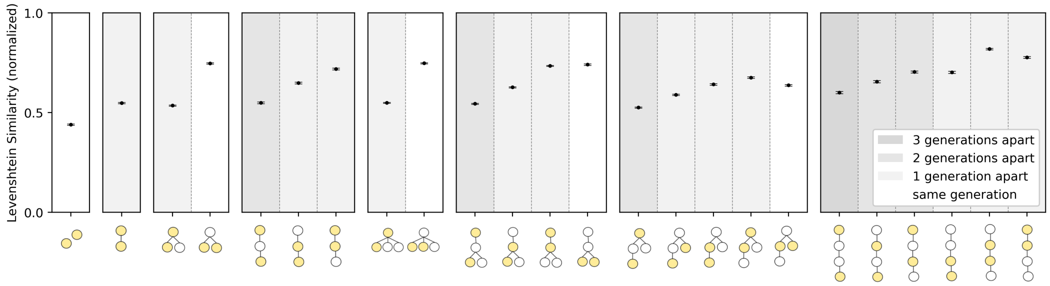


Figure 7: Levenshtein distance based similarity measure on the model metadata.

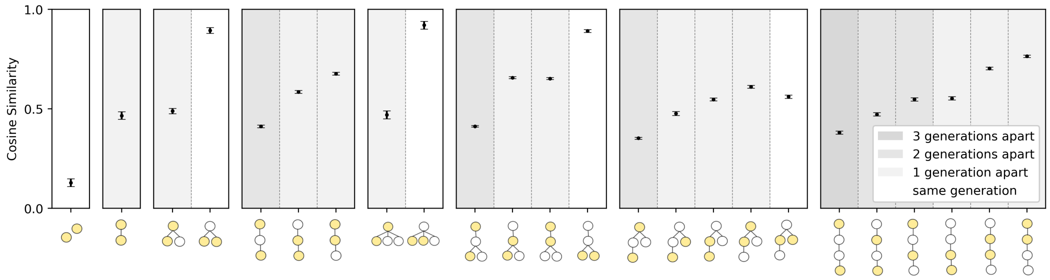


Figure 8: TF-IDF Cosine Similarity, Model Cards.

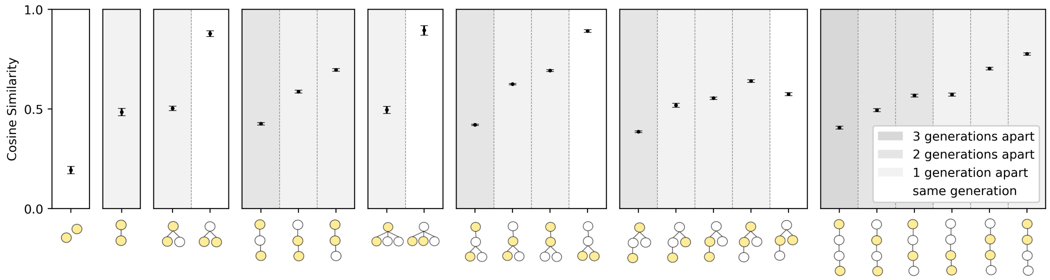


Figure 9: Bag of Words Cosine Similarity, Model Cards.

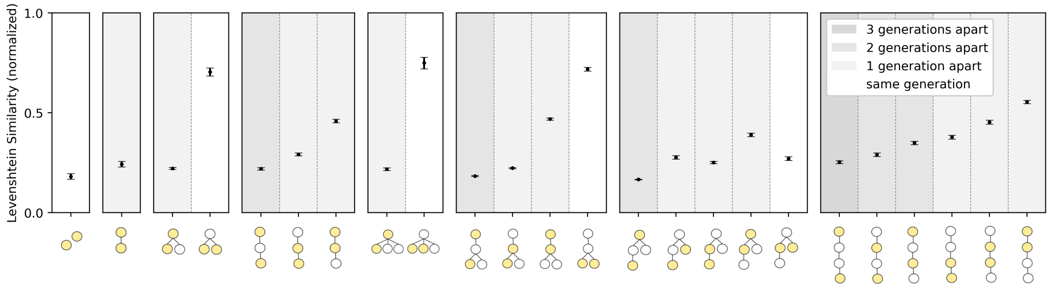


Figure 10: Levenshtein distance based similarity measure using the model cards. We have reason to believe this is the least reliable measure, as model cards are free text and Levenshtein distance relies heavily on text ordering, making it more suitable for structured strings. Directional patterns nonetheless resemble the findings using other metrics.

D Defining the mutation rate

In the paper, we attempt to measure the mutation rate over model *traits*. Depending on how various traits are logged in the metadata JSON, Hugging Face sometimes allows one model to list multiple traits in the same category. For other traits, however, a model can only have one categorical value. For example, models can be compatible with multiple languages, because languages are logged in the metadata as tags. Models can only have one task (or '`pipeline_tag`'), however. Here we provide a definition for the mutation rate over a category of traits. This is the definition used in all cases where mutation rate is reported in the paper. It is compatible with both types of traits listed below (those for which models can have multiple values, and those for which models can have only one value).

Definition D.1 (Mutation rate over traits T)**.** Given a set of categorical traits T. Every model i in our graph has a group of individual elements denoted $t_i = \{a, b, ...\} \in T$. Then the mutation rate over any directed edge (i, j) is given by $m(i, j) = 1 - \frac{t_i \cap t_j}{t_i \cup t_j}$. The mutation rate over the set T is equal to $\frac{\sum_{\text{edges } (i,j)} m(i,j)}{N_{\text{edges}}}$.

Notice that, in cases where every model must have a single categorical value in the set of traits (equivalently, t_i has cardinality one $\forall i$), the mutation rate on any edge is 0 if the parent and child have the same trait, and 1 if the parent and child have different traits.

E Data collection and summary statistics

Here we provide some additional information on the dataset and general exploratory data analysis conducted.

E.1 How we collected the data

We collected the data for our dataset in two stages. In the first stage, we used the Hugging Face 'model' API to collect the model features and relationships—that is, all pieces of information in our dataset aside from the model cards. Hugging Face provides API access to individual *lists* of models, but these lists are capped to only list 1000 models. Using pagination, we were able to iterate over all such lists of models to collect the information in our dataset in JSON format. In the second stage, we collected the full text of every model's model card through individual, per-model API calls to the model cards API. These cards were significantly more data-intensive—since model cards can be quite large and many more API calls were required to find all 1.86 million models in the dataset. In total, our full dataset uses memory on the order of 10GB (depending on the file format used), and the dataset without model cards uses significantly lower memory, at around 500MB. All calls to the API were conducted through the authors' registered accounts on Hugging Face, and in consultation with employees at Hugging Face, including Hugging Face's in-house librarian.

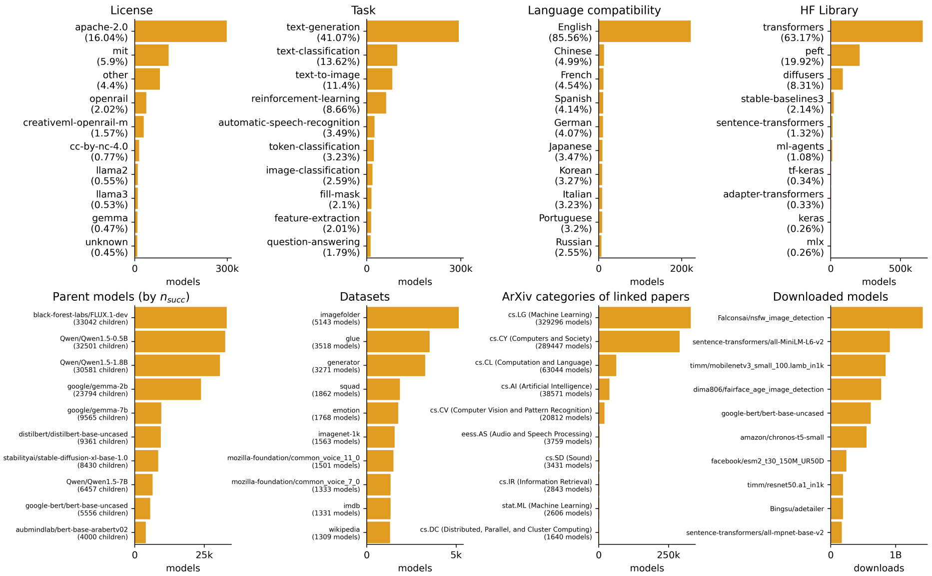


Figure 11: Top ten most frequent licenses, tasks, languages, and libraries (top row). Top ten models ranked by number of children, datasets, arXiv categories of linked papers, and downloaded models (bottom row).

E.2 Properties and summary statistics

Our dataset centers around snippets of text for every model known as the model's metadata. Model metadata comes in JSON format, and this JSON is made readily available for any model through Hugging Face's API. These JSONs include the `model_id` (a unique identifier for each model containing its author and name), `likes`, `trendingScore` (a trait defined by Hugging Face for ranking models on their website), `downloads`, `pipeline_tag` (also known as `task`—a categorization of models into e.g., `feature-extraction`, `text-generation`, `image-classification`, and other modalities), `library_name` (the Hugging Face library used to support development), `createdAt` (the date and time that the model was created[3]), and `tags`. Tags contain a structured list of strings, some with organized prefixes. For example, tags beginning with `base_model:finetune:` link a finetuned model to its parent's model id, tags beginning with `license:` contain the model's license, and those beginning with `arxiv:` contain links to the arXiv identifiers of accompanying papers. Other tags do not have these prefixes, but their meaning can still be inferred. For example, languages are listed using two- or three-letter ISO-639 codes.

A summary of the distributions of the various metadata traits is provided in Figure 11. These distributions convey the relative frequencies of different traits, as well as the absolute number of papers with these documented traits. Here, we provide some findings these figures convey about the state of the open source ecosystem on Hugging Face, reading from left to right and top to bottom through the figure. A few trends emerge from these summary statistics and rates. First, permissive licenses—especially `apache-2.0` and `mit` are dominant, constituting over 60% of all reported licenses. Text-based tasks—and especially `text-generation`—are most common. English is by far the dominant language compatibility on Hugging Face, with over 75% of models that document any language compatibility marking english as a supported language. Chinese is the second most-common at 4.4%. `transformers` is the most common Hugging Face library. `black-forest-labs/FLUX.1-dev` is the model that has the most children. `imagefolder` is the most commonly recorded dataset in metadata. `Machine Learning` and `Computers and Society` codes are the most common among linked arXiv papers. Finally, in the lower right figure, we show the most downloaded models, finding that the model `Falconsai/nsfw_image_detection` is the

---

[3] Tracking of the `createdAt` date and time began March 2, 2022. According to the Hugging Face documentation, and corroborated by our findings, all models created before that date are back-filled with that date; the date is accurate for all models uploaded thereafter.

most downloaded. This model's purpose is to detect and identify explicit imagery and is perhaps used for content moderation and compliance.

A remarkable amount of information is conveyed in text snippets that Hugging Face stores for every model. Throughout the paper, we treat the snippets of text provided by the metadata JSON as the models DNA, as it contains rich information about traits and allows us to track changes and differences over generations (illustrated in Figure 1c). Before embarking on this genetic analysis, we discuss one additional source of genetic information: the model cards.

Model cards are documents that carry information about the use, performance, compatibilities, risks, impacts, and many other pieces of information about models Mitchell et al. [2019]. Model cards are the main form of documentation for models on the hub, and they constitute much of the information that populates on any given model's associated webpage. Model cards can be considerably longer than metadata, and much less structured. They can therefore contain more information, however, not all models have corresponding model cards, and they are considerably less standardized and organized. According to our data, 67.04% of models currently have an associated model card. An analysis of the 1,247,149 cards available reveals an average model card length of 3575.60 characters ($\approx$ 436.06 words), with a median of 2073.0 characters ($\approx$ 238.0 words). This wide range, from a minimum of 11 characters to a maximum of 18,289,454 characters ($\approx$ 2,813,762 words), indicates that a small number of extremely verbose cards significantly influence the average.

E.3 Linking papers from arXiv

To investigate the research inspiring models on the Hub, we extracted all linked papers from model metadata. For available arXiv IDs, we queried the `arXiv` API to retrieve the corresponding titles, abstracts, and subject classifications, allowing us to systematically categorize the papers by domain.

arXiv subject classification IDs (like cs.AI, cs.CL) are extracted from the categories column in the full JSON dataset, maps them to readable subject names using a predefined dictionary, and counts the frequency of each subject across all models. The process handles both single categories and lists of categories per model, flattening all categories into a single list before counting occurrences, where models with multiple arXiv categories contribute to the count of each individual category (e.g., a model with ['cs.AI,' 'cs.CL'] adds +1 to both "Computer Science, Artificial Intelligence" and "Computer Science, Computation and Language"). The top 20 most frequent research domains are then visualized in Figure 11.

E.4 Documentation availability

We analyze model availability and observe low adoption of Hugging Face complimentary tools Goldin and Katz [1996]. Only 5.96% of the models are endpoint-compatible or accessible via the Hugging Face API without local hosting. Furthermore, 6.6% of the models released with weights use the safetensor file format—the default model weight format developed by Hugging Face in 2022 Face [2022].[4] Additionally, 23.69% of the models use automated training via Hugging Face Spaces—containerized web deployment environments. Although only a small subset of Hugging Face models have self-assigned DOIs, they are downloaded 29× more than those without. Possible explanations include DOIs make models more visible and trustworthy, and people tend to choose models that are already popular and well-documented.

F Further information on sampling subtree topologies

Here we provide a more complete table as an addendum to Table 4. For each shape of subgraph, we implemented a specific sampling method to get a representative sample of models. The sampling method is summarized in Table 1.

---

[4] Although the format was developed in 2022, it became the default (as a zero-copy alternative to `pickle`) in 2023 Yoshimura et al. [2025].

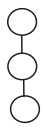
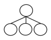
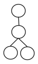
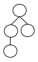
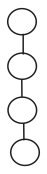
Subgraph	Occurrences	Sampling condition	Multiplicity|condition
	3,470,193,356,870	Two arbitrary nodes.	1
	191,072	Single edge (u, v).	1
	119,795,843	Node u with more than one successor.	$\binom{n_{\text{succ}}(u)}{2}$
	40,922	Edge (u, v) where v has successors.	$n_{\text{succ}}(v)$
	193,010,561,824	Node u with more than two successors.	$\binom{n_{\text{succ}}(u)}{3}$
	11,847,103	Edge (u, v) where v has more than one successor.	$\binom{n_{\text{succ}}(v)}{2}$
	19,932,645	Edge (u, v) where u has multiple successors and v has successors.	$n_{\text{succ}}(v)(n_{\text{succ}}(u) - 1)$
	10,965	Edge (u, v) where u has a predecessor and v has successors.	$n_{\text{succ}}(v)$

Table 1: Subgraph patterns, their total occurrences, sampling conditions, and associated multiplicities conditioned on each pattern. $n_{\text{succ}}(u)$ refers to the number of successors (or, equivalently, the out-degree) of node u.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main claims are summarized in the abstract and introductory paragraphs.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We detail our methodology in the paper and explicitly discuss the limitations of our approach in the discussion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We describe our sampling strategies and define different notions of similarity in the body of our work 1c in our appendix 1.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes] .

   Justification: We make both the dataset and the accompanying code publicly available, and we provide detailed documentation of our collection and analysis procedures within the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may

be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes] .

Justification: We provide links to our code on GitHub and to the datasets on Hugging Face, released under a CC-BY-4.0 license, with detailed documentation of our procedures in the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA] .

   Justification: [NA] .

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes] .

Justification: Upon review of the NeurIPS Code of Ethics, we affirm that our research adheres to its guidelines to the best of our knowledge and ability.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Our analysis of 1.86M Hugging Face models provides transparency into open-source AI development and governance patterns. Our findings reveal concerning trends: license drift, declining multilingual support, privacy vulnerabilities, and risks of misuse for surveillance or disinformation. By releasing this dataset to the research community, we aim to support empirical ML research while highlighting these challenges in large-scale open-source AI ecosystems.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: We release only metadata and aggregate properties of models (e.g., licensing information, documentation statistics, language support), not the actual model weights or training data. Our dataset consists entirely of summary statistics and derived features extracted from information already publicly available on Hugging Face, a vetted platform with established safety protocols. We provide comprehensive documentation and this paper to help researchers navigate the dataset responsibly. This approach enables transparency research while avoiding distribution of potentially harmful content.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We list all external assets used and respect their licenses and terms of use.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We provide links to the corresponding data and code repositories alongside this paper.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA] .

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLM do not impact the core methodology in this work and were used for for writing, editing, or formatting purposes.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.