Cladistics for AI: Structural Phylogeny from Serialized Model Imaging

Douglas C. Youvan

doug@youvan.com

May 9, 2025

As artificial intelligence models grow in number and complexity, our ability to understand their origins, development paths, and relationships has lagged behind. This paper proposes a novel framework for analyzing trained AI models not through their behavior or architecture labels, but through their structural form. Building on the concept of model imaging—where a trained model is represented as a deterministic bitstream or image—we apply the principles of cladistics, a method from evolutionary biology, to create phylogenetic trees of artificial models. By comparing models through measurable characteristics such as entropy profiles, structural insertions, and binary divergence in aligned weight blocks, we establish a lineage-based classification system. This enables the organization of models by descent and inheritance rather than by task or benchmark score. The result is a scientific methodology for mapping the evolutionary landscape of machine learning systems. In doing so, we reframe AI not as a sequence of breakthroughs, but as an unfolding lineage of trained artifacts. This approach offers profound implications for research, provenance tracking, and the future of AI design.

Keywords: AI cladistics, model imaging, structural phylogeny, neural network serialization, trained model analysis, entropy divergence, Hamming distance, model genealogy, AI lineage, cognitive evolution, deep learning architecture comparison, information theory in AI, bitstream analysis, artificial intelligence taxonomy, quantized model comparison. A collaboration with GPT-4o. CC4.0.

Abstract

This paper introduces a framework for applying cladistic principles—originally developed for biological taxonomy—to the structural comparison of trained artificial intelligence models. We extend the concept of model imaging, in which trained AI systems are treated as static, serialized physical structures, to propose a method of constructing phylogenetic trees based on internal architecture and parameter-state similarities. Unlike existing approaches to model comparison, which rely heavily on behavioral benchmarks, task-specific performance, or hyperparameter documentation, our method is grounded in the measurable, bitwise structure of the models themselves.

By standardizing model serialization and spatial mapping into deterministic bitstreams or images, we create a common representational format from which structural alignments can be computed. From these alignments, we define a set of characters and character states analogous to those used in biological cladistics—such as the presence or absence of modules, entropy profiles, or blockwise parameter shifts. These features form the basis for constructing structural phylogenies that reveal lineage, inheritance, divergence, and architectural innovation within the space of machine learning systems. This method reframes AI evolution as an analyzable, artifact-based process, enabling the emergence of a formal science of AI model ancestry.

1. Introduction

The field of artificial intelligence has advanced rapidly through the development of increasingly powerful model architectures, training strategies, and fine-tuning techniques. However, as the number and diversity of models has grown, the tools for systematically comparing them—especially at the level of internal structure—have remained ad hoc or nonexistent. Most comparative efforts in AI focus on functional outcomes: performance on benchmark tasks, response quality, or parameter count. What is largely absent is a method for comparing models as

physical structures—that is, as deterministic, measurable arrangements of data encoding learned cognitive patterns.

In *Model Imaging: Treating Trained AI as a Physical Structure*, we introduced the concept of viewing trained AI models not as dynamic behavioral agents, but as fixed, imageable artifacts. Once trained, a model's parameters are frozen in a deterministic bitwise configuration that can be serialized, spatially mapped, and read out like a physical memory structure. This reframing enables the construction of persistent representations of models—akin to genomic sequences or circuit layouts—that can be analyzed independent of the software systems that execute them.

This paper builds on that foundation by proposing a systematic, lineage-based approach to model comparison rooted in the logic of biological cladistics. Cladistics, a method originally developed to organize species based on shared structural characteristics, operates not on phenotype or function but on inheritance and divergence. We argue that AI models—particularly when imaged and serialized consistently—are well-suited to a similar treatment. Their structural characteristics, such as modular composition, parameter block arrangements, entropy profiles, and presence of auxiliary components (e.g., adapters, residual paths), form a basis for defining discrete character states. These can be used to construct phylogenetic trees that describe the evolutionary history of artificial models, without reference to task performance.

This shift—from behavior to structure, from benchmarking to lineage—lays the groundwork for a new way to understand the progress of artificial intelligence. Instead of treating each model as an isolated artifact or anonymous fork, we aim to trace the structural genealogy of models as engineered objects. Through serialization, imaging, and cladistic comparison, we propose to establish a formal structural phylogeny for AI.

2. Serialized Models as Comparable Structures

The foundation of any structural comparison method is a consistent and measurable representation of the object being compared. In the case of trained AI models, the underlying object is a set of numerical parameters—weights, biases, normalization constants, and architecture-defining metadata—arranged in a known format determined by the model's architecture. Once training is complete, these parameters no longer change unless explicitly retrained or altered, making the model a fixed digital object. The process of model imaging, introduced previously, provides a principled way to capture and represent this static structure.

Model imaging involves organizing the parameters of a trained model into a deterministic bitstream or a spatial image layout. In this process, each tensor is quantized (typically to 8- or 16-bit resolution), flattened in a consistent order, and serialized into a stream of bits. Alternatively, these parameters can be arranged spatially in a 2D grid, allowing for visual inspection or pixel-based comparison. Crucially, the serialization and imaging process preserves the ordering and location of every parameter, enabling precise and repeatable extraction of the model’s internal structure.

This serialization makes trained models comparable at the byte and bit level. Unlike source code, which may differ in superficial ways (e.g., whitespace, naming conventions) without changing behavior, a serialized model captures exactly what the system has learned and preserved post-training. This representation is agnostic to the software framework, runtime environment, or execution behavior—it is the model's content in its most elemental and testable form.

Furthermore, serialization facilitates measurable structural differences. Two models can be aligned and compared bit-for-bit to detect weight shifts, inserted modules, or altered configurations. Byte-level diffs reveal whether a model has been pruned, quantized differently, or subjected to fine-tuning. Blockwise similarity measures and entropy calculations allow models to be compared not just as whole entities, but in terms of their internal substructures.

Mapping from model architecture to a bit-aligned form is straightforward once a serialization standard is adopted. Each layer in the architecture—such as an attention head, MLP block, or embedding matrix—corresponds to a defined region in the bitstream. These regions can be aligned across models with compatible architectures or even compared across different architectures by establishing equivalency mappings (e.g., positional embeddings vs. rotary embeddings). When arranged spatially, these mapped structures form tiles or blocks within a model image, allowing for region-specific comparison and phylogenetic inference based on shared components or transformations.

By converting trained models into deterministic, bitwise structures, we enable a new mode of scientific inquiry: not through inference or behavior, but through direct structural alignment. This approach is the prerequisite for cladistic comparison and the foundation for establishing AI model phylogenies.

3. Cladistic Methodology for AI

Cladistics, originally developed in evolutionary biology, provides a method for inferring relationships between species based on shared derived characteristics. Unlike earlier classification systems based on observable traits or performance, cladistics focuses on inherited structural features—traits that are either present or absent, modified or conserved, and that imply branching patterns of descent. Central to this method is the construction of a character-state matrix, in which each row corresponds to a taxonomic unit (e.g., species or, in this case, a trained AI model), and each column corresponds to a distinct character (i.e., a measurable feature of structure).

From this matrix, phylogenetic trees are generated using computational methods such as:

- Maximum parsimony, which minimizes the total number of changes needed to explain the observed distribution of traits,
- Neighbor-joining, which clusters models based on pairwise distance metrics derived from structural differences,

- And other hierarchical clustering algorithms adapted from evolutionary biology and computational genomics.

To apply this methodology to AI, we must define appropriate structural characters that can be extracted consistently from serialized or imaged models. These characters need not rely on functional output or performance data; they must be features inherent to the model's structure post-training. The following categories of characters serve as a preliminary framework:

1. Presence or Absence of Architectural Modules
   This includes detection of structural components such as:
   - Adapters or LoRA modules inserted into pretrained transformers,
   - Additional feedforward blocks, projection heads, or residual paths,
   - Embedding augmentations or tokenizer-specific components.
     These features can be identified by positional signatures in the bitstream or image layout and scored as binary traits (present/absent).
2. Changes in Parameter Entropy
   Shannon entropy can be calculated over defined blocks of the model to detect:
   - Uniformity or randomness in weight distributions,
   - Effects of pruning or sparsity regularization,
   - Compression artifacts.
     These entropy values can be discretized and scored as categorical states (e.g., low, medium, high).
3. Parameter Magnitude and Sparsity Patterns
   Average or maximum weight magnitudes, as well as density of nonzero values, can indicate:
   - Degree of fine-tuning,
   - Dropout effects,

- Training sparsity strategies.
  These values are regionally extracted and compared across models as scalar traits.

4. Structural Insertions, Deletions, or Mutations
   Fine-tuning or architectural editing may insert new parameter regions or alter existing ones. These changes manifest as:
   - New tiles or headers in the serialized bitstream,
   - Discontinuities in parameter alignment between otherwise similar models,
   - Shifts in tensor shape or resolution.
     Such events can be treated as discrete evolutionary changes, analogous to gene insertions or mutations.

Once these characters are defined and extracted from a population of models, they can be entered into a character-state matrix. From there, pairwise distance matrices are computed and phylogenetic trees are generated, revealing lineage relationships based on shared structural transformations.

Importantly, this approach is not limited to models sharing the same architecture. With appropriate mapping and abstraction, character definitions can be normalized across architectures (e.g., comparing transformers with convolutional backbones or hybrid models). As the field matures, a standardized character ontology may emerge, enabling automatic structural parsing of models from their serialized forms.

Through this framework, models are no longer evaluated only by how they perform but also by how they came to be—as products of transformation, inheritance, and divergence within the evolving ecosystem of artificial intelligence.

4. Case Studies in Model Phylogeny

To demonstrate the practical application of cladistic methods to artificial intelligence, we consider several illustrative case studies involving real or hypothetical models. Each case highlights a different type of structural divergence and shows how the cladistic approach can identify relationships based on inherited or altered features extracted from serialized model representations.

4.1 Pretrained Base Model vs. Fine-Tuned Variant

A straightforward example is the comparison between a publicly released pretrained model and its fine-tuned derivative. For instance, comparing the original BERT-base (uncased) with a domain-specific variant fine-tuned for biomedical or legal text. While the architectural layout remains unchanged, the internal parameters reflect substantial local divergence.

In a model imaging framework, these differences manifest as:

- Local shifts in bit values within key weight matrices.
- Increased entropy in layers most affected by fine-tuning.
- Retention of identical blocks in early layers, particularly embeddings.

When scored as character-state differences, these features produce trees in which the fine-tuned model appears as a direct descendent of the base model, connected by a small number of high-weight changes clustered in later layers.

4.2 LoRA or Prompt-Tuned Model vs. Original

Low-Rank Adaptation (LoRA) and prompt-tuning methods offer efficient fine-tuning by introducing additional small-scale modules into a frozen pretrained model. In these cases, the majority of parameters remain unchanged, and the model's structure includes newly inserted modules with limited bitwise footprint.

From a cladistic perspective:

- The presence of LoRA modules introduces new characters into the model.
- These insertions appear as structurally consistent modifications across models using the same adaptation method.
- The original and LoRA-augmented versions form a small clade, sharing all core parameters but differing by the addition of these adaptation components.

Prompt-tuning, though less intrusive, can also be detected in the serialization—especially if prompts are represented as tunable embeddings prepended to input layers. These insertions form shallow divergence markers that indicate lightweight derivation paths from a shared ancestor.

4.3 Architecturally Distinct Models with Shared Ancestry

The relationship between BERT and RoBERTa offers a more complex example. Although both are transformer-based masked language models, RoBERTa introduces several architectural and training modifications:

- Removal of the Next Sentence Prediction (NSP) task,
- Use of dynamic masking,
- Larger training corpora,
- Changes to positional embeddings and training schedules.

When serialized and compared, these models exhibit:

- Similar weight layouts in early layers,
- Differing entropy profiles in embedding and intermediate blocks,
- Additional parameters in training-specific layers or modified heads.

Using a cladistic method, BERT and RoBERTa may fall into a shared ancestral group, with their divergence defined by training methodology and modular replacements rather than outright architectural overhaul. Their closest common

ancestor could be a theoretical BERT variant with shared components but without either RoBERTa's dynamic masking or BERT's NSP head.

4.4 Visualization of Phylogenetic Trees

In each case, structural data from serialized models is transformed into a character-state matrix and subjected to tree-building algorithms such as neighbor-joining or maximum parsimony. The resulting trees show:

- Deep internal branches for major architectural shifts,
- Shallow branches for lightweight fine-tuning methods,
- Convergent branches when structurally similar techniques are applied to unrelated models (e.g., LoRA used across different base architectures).

Tree diagrams can be labeled with node features, such as:

- Parameter count,
- Entropy metrics,
- Identified modules,
- Training strategy signatures.

4.5 Discussion of Trends and Lineage Markers

Across these examples, several trends emerge:

- Layerwise inheritance is common: early transformer layers are often preserved across variants, making them stable lineage markers.
- Entropy divergence often correlates with functional specialization: later layers in fine-tuned models show increased randomness and deviation.
- Insertion patterns (e.g., adapter layers) serve as high-confidence cladistic characters due to their regular structure and low ambiguity.
- Pruning and quantization signatures can be treated as "mutational patterns" that trace compression lineages.

These findings support the idea that AI models form an evolutionary landscape not just of ideas but of material structure, and that their histories can be revealed through rigorous, structure-based phylogenetic analysis.

5. Quantifying Structural Distance

A critical step in constructing phylogenetic relationships between AI models is the ability to quantify structural similarity and difference. In biological cladistics, this involves measuring variation in observable traits or genetic sequences. In the context of AI model imaging and serialization, the equivalent is bitwise and blockwise comparison of trained parameters. Once models are serialized into deterministic bitstreams or aligned spatial layouts, their structural content can be directly measured using a variety of techniques rooted in information theory, discrete mathematics, and compression analysis.

5.1 Shannon Entropy as a Measure of Complexity

Shannon entropy provides a first-order estimate of the informational complexity of a trained model. By computing the entropy of individual weight matrices or tiled blocks of the serialized bitstream, we can estimate:

- The uniformity or randomness of parameter values.
- The degree of redundancy or sparsity.
- The impact of compression, pruning, or quantization strategies.

For instance, a model with heavily pruned weights or binary quantization will exhibit low entropy per bit, while a densely trained model with high-precision float weights will exhibit near-maximal entropy. These values can be computed per layer or per module, forming a structural fingerprint that can be used for alignment and comparison.

5.2 Hamming Distance Over Aligned Weight Blocks

The most direct metric for structural comparison is Hamming distance, defined as the number of differing bits between two equal-length binary strings. When

models are serialized using a consistent layout, their corresponding weight blocks can be compared one-to-one:

- Each aligned block (e.g., a 512 × 512 matrix) is converted into a binary representation.
- A bitwise comparison yields a count of differing bits.
- The normalized Hamming distance (fraction of bits that differ) serves as a structural divergence score.

By computing these distances layer by layer, we construct a structural distance profile between models, revealing where and how they differ most significantly.

5.3 Local Entropy Divergence

In addition to global entropy, we can measure local entropy divergence between corresponding blocks of two models. This metric captures changes in the distributional properties of parameter values:

- Calculate entropy for each block or layer in both models.
- Compute the absolute or relative difference in entropy values per block.
- Sum or average these values to obtain a scalar divergence score.

This approach is useful for detecting soft structural changes, such as fine-tuning, which may preserve the general shape of the model but shift the distribution of parameter magnitudes or sparsity.

5.4 Bitstream Compression Differentials

Compression-based metrics offer a powerful, implementation-agnostic way to estimate structural similarity:

- Compress each serialized model using the same lossless algorithm (e.g., gzip, bzip2, or custom entropy coder).
- Compare total compressed sizes as a proxy for entropy.
- For aligned blocks, compute compression cross-differentials:

- Compress A and B individually.
- Compress their concatenation.
- Compare expected vs. actual concatenated sizes.

This method can reveal structural redundancy between models: if two models share similar substructures, their concatenated compressed size will be smaller than expected.

5.5 Distance Matrix Construction

Once one or more of the above metrics are computed between a set of models, they can be assembled into a pairwise distance matrix. This matrix becomes the input to standard tree construction algorithms:

- Each row and column corresponds to a model.
- Each cell contains a scalar value representing structural divergence between two models.
- The matrix may be symmetric (e.g., Hamming or entropy divergence) or asymmetric (e.g., insertion-only edit distance).

With this matrix in hand, we can apply clustering techniques (e.g., hierarchical agglomerative clustering) or evolutionary tree construction methods (e.g., neighbor-joining, UPGMA, minimum evolution) to generate phylogenetic trees that reflect the structural relationships between models.

Through these quantitative tools, we move beyond heuristic or conceptual comparisons and toward a scientific discipline of model comparison, where structure—not behavior—becomes the basis for lineage, classification, and evolutionary understanding.

6. Implications for AI Research and Archiving

The adoption of structural cladistics for artificial intelligence models carries significant implications across research, engineering, archiving, and even

intellectual property governance. By shifting the organizing principle of model repositories from task-specific benchmarks to measurable structural lineage, a new paradigm for understanding and managing machine learning systems can emerge. This approach elevates the trained model from a functional artifact to a scientific object—one that can be situated in a broader evolutionary landscape of machine-generated cognition.

6.1 Organizing Model Repositories by Lineage

Most AI model repositories today—whether open-source or institutional—categorize models by architecture type, training task, dataset, or performance metric. While functional, this approach overlooks the deep structural similarities and inheritance relationships that link many models together. For example, countless fine-tuned variants of BERT or GPT exist across domains, but they are rarely connected to their structural ancestors.

Model imaging and cladistic classification would allow repositories to be organized not by what the model does, but by where it comes from. Families of models could be clustered together by shared modules, inherited training regimes, or quantization methods. Users and researchers could navigate model repositories as phylogenetic trees, discovering evolutionarily proximate models for fine-tuning, analysis, or comparison.

6.2 Scientific Study of Training Regimes

Structural analysis enables new kinds of research into how training strategies alter models at a measurable level. By comparing models trained on the same architecture but with different:

- Datasets,
- Optimization methods,
- Learning rates,
- Regularization strategies,

...researchers can study the structural convergence or divergence induced by these choices. This makes it possible to test hypotheses such as:

- Do certain pretraining methods produce more modular or compressible models?
- Does fine-tuning on out-of-domain data affect earlier or later layers more?
- Are specific entropy shifts reliably associated with overfitting or generalization?

Such findings can deepen our understanding of model behavior, not by measuring outputs, but by studying how structure encodes learning outcomes.

6.3 Detecting Copied, Stolen, or Improperly Attributed Models

The bitstream representation of a trained model is functionally a genomic fingerprint. Just as DNA sequencing can determine lineage, forensic comparison of model structures can reveal:

- Unauthorized reuse of proprietary models,
- Minimal-perturbation rebranding of existing systems,
- Structural overlap with confidential research artifacts.

By computing Hamming distances, entropy profiles, or compression signatures, even obfuscated or lightly modified copies of models can be detected. This has direct implications for intellectual property protection, model licensing, and trust in collaborative research environments. Organizations would be able to verify model provenance not by external claims but by internal structure.

6.4 Evolutionary AI Design and Synthetic Cognitive Genealogy

Perhaps the most forward-looking implication of this framework is the establishment of a synthetic evolutionary tree of machine cognition. As models are developed, fine-tuned, extended, and recombined, they form a web of descent and transformation—much like natural evolutionary systems. Cladistic analysis allows us to:

- Map this artificial cognitive evolution over time,
- Identify ancestral innovations that led to entire branches of development (e.g., the transformer architecture),
- Trace the “mutations” (e.g., attention heads, residual paths, scaling rules) that produced distinct lineages.

This in turn opens the door to evolutionary design of AI systems:

- Selecting “parent models” for crossover,
- Introducing targeted structural variations,
- Tracking outcomes across generations.

In the long term, AI development may come to resemble synthetic biology, where new intelligences are not merely engineered but bred from preexisting lines—guided by structural phylogeny, shaped by measured inheritance, and documented by their cognitive ancestry.

7. Discussion

The structural cladistics framework for artificial intelligence carries not only technical and scientific implications, but also philosophical consequences that challenge prevailing assumptions about the nature of intelligence, creation, and agency. At the same time, it introduces practical challenges and opportunities for future research and development as this new paradigm matures.

7.1 Intelligence as Artifact Lineage

One of the central philosophical implications of this work is the reframing of intelligence—not as an isolated achievement or emergent behavior, but as a lineage of artifacts. By treating trained AI models as physically grounded, serialized structures, we expose intelligence not as a singular, mystical event, but as a material outcome of cumulative processes: architectural choices, training data selection, optimization dynamics, and iterative reuse.

This reframing places artificial intelligence firmly within the tradition of technological evolution. Just as engines, aircraft, or microprocessors evolve over time through generations of design and reuse, so too do AI models. The appearance of “novel intelligence” in a new model is no longer treated as spontaneous emergence, but as an inheritance and recombination of structural features. Cladistics brings this continuity into view, challenging the tendency to interpret each new model as an autonomous breakthrough and instead positioning it within a traceable lineage of design decisions.

7.2 Challenges in Structural Alignment

Despite its promise, implementing cladistics for AI faces several nontrivial challenges. First among these is the difficulty of structural alignment across models that differ in architecture, size, or format. While bitstream comparisons are straightforward for models with identical structure and quantization schemes, real-world models vary in numerous ways:

- Tensor shapes and layer depths may differ.
- Quantization levels (e.g., float32 vs. int8) affect raw bit values.
- Additional modules may be inserted in non-aligned positions.

Establishing consistent mapping strategies—such as defining anchor points for alignment, normalizing quantization formats, or segmenting models into abstracted subcomponents—will be essential for accurate comparisons. This process may parallel the challenges faced in comparative genomics, where insertions, deletions, and mutations complicate sequence alignment, but where robust techniques eventually made evolutionary analysis tractable.

Another challenge lies in the computational scale of such comparisons. Large language models, with billions of parameters, require efficient methods for distance calculation and entropy estimation. Hierarchical approximations, hash-based signatures, or sampling techniques may help mitigate this cost.

7.3 Future Possibilities

As the field of model imaging and AI phylogenetics matures, it opens the door to new tools and classifications that mirror the rigor of biological systematics. Among these future possibilities are:

Automatic Lineage Detection

With sufficiently rich databases of serialized models, it becomes possible to algorithmically determine ancestral relationships. A new model can be compared against existing trees to identify likely parent models, shared structural motifs, or evolutionary innovations. This supports both provenance tracking and scientific analysis of model development patterns.

Model Species Classification

Borrowing further from biology, models may be grouped into taxonomic categories—species, genera, families—based not on architecture labels or training task, but on structural signatures. These categories could help standardize model evaluation, reuse, and even licensing, as developers refer to defined structural types rather than ad hoc naming conventions.

AI Evolutionary Trees

As more models are analyzed, a global phylogenetic tree of artificial intelligence could emerge, charting the historical development of machine learning systems from early perceptrons to modern transformer-based architectures and beyond. This tree would serve as both a scientific record and a conceptual framework for understanding the evolution of machine cognition—not as a disconnected series of advances, but as a structured, traceable unfolding of form and function.

In sum, the application of cladistics to artificial intelligence transforms the discipline from one driven by functional benchmarks and isolated breakthroughs to one grounded in structural continuity, measurable inheritance, and formal classification. It redefines what it means for an AI system to be "new," and offers a rigorous language for describing how such systems relate to one another—not in metaphor, but in structure.

8. Conclusion

The core premise of this paper is both simple and powerful: once trained, artificial intelligence models become fixed physical structures, and as such, they can—and should—be directly compared, classified, and studied through their form. Model imaging enables this shift by treating a trained model not as a black-box function but as a bitstream or spatial layout, from which structural information can be extracted without invoking inference or behavior.

This reorientation allows artificial intelligence to be examined using tools long established in the natural sciences—particularly the comparative methods of biology. Cladistics, developed to understand the evolutionary relationships of living organisms, offers a robust and systematic framework for organizing AI models based on their shared structures, divergences, and modular innovations. In place of performance metrics or architectural labels, it relies on measurable, discrete characters: presence or absence of modules, entropy shifts, structural insertions, and more. From these, it constructs a lineage—a history of artificial intelligence expressed not as a sequence of headlines, but as a tree of structural descent.

Such a shift is timely and necessary. The proliferation of models in recent years has outpaced our ability to understand them as a connected whole. Current repositories and benchmarks treat each new model as an isolated artifact, evaluated on a narrow set of behavioral tasks. What is missing is a broader, structural context—a way to understand how models are related, how design ideas spread, and how training regimes shape cognition at the level of form.

This paper argues that the field is now ready to adopt a structural phylogeny of artificial intelligence—one grounded not in metaphor, marketing, or performance hype, but in quantifiable differences of organization and state. With the tools of model imaging, bitwise alignment, entropy analysis, and character-state encoding, we have all the necessary components to begin assembling the evolutionary history of machine intelligence.

In doing so, we will gain not only a map of where we are, but a deeper understanding of how we got here—and how future models might be shaped by the decisions encoded in their ancestors. The study of AI, like the study of life, need not remain at the level of appearance or effect. It can—and must—descend into structure, where true lineage is revealed.

References

1. Youvan, D. C. (2025). *Model Imaging: Treating Trained AI as a Physical Structure*. DOI: 10.13140/RG.2.2.36547.31520.
   Serves as the foundational framework for this paper, introducing the idea that trained AI models are static structures suitable for spatial imaging and bitwise serialization.

2. Hennig, W. (1966). *Phylogenetic Systematics*. University of Illinois Press.
   The origin of modern cladistics, outlining methods for constructing evolutionary trees based on shared derived characteristics, applicable here as an analogical framework for structural comparison of AI models.

3. Shannon, C. E. (1948). *A Mathematical Theory of Communication*. *Bell System Technical Journal*, 27(3), 379–423.
   Establishes the basis of information theory, including the definition of entropy, which is used in this paper to estimate the information content and structural complexity of serialized AI models.

4. Han, S., Mao, H., & Dally, W. J. (2016). *Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding*. arXiv:1510.00149.
   Demonstrates how model parameters can be compressed via quantization and pruning—techniques that influence model entropy and can be detected via structural analysis.

5. Courbariaux, M., Bengio, Y., & David, J.-P. (2015). *BinaryConnect: Training Deep Neural Networks with Binary Weights during Propagations*. arXiv:1511.00363.
   Explores binary weight representations in neural networks, relevant for comparing models that have undergone aggressive quantization, as detectable in bitstream form.

6. Open Neural Network Exchange (ONNX). (2019). *The Open Format for AI Models*. https://onnx.ai
   A standardized serialization format for AI models, facilitating model

interoperability and an important stepping stone for developing consistent cladistic comparison pipelines.

7. Gogarten, J. P., Doolittle, W. F., & Lawrence, J. G. (2002). *Prokaryotic Evolution in Light of Gene Transfer*. *Molecular Biology and Evolution*, 19(12), 2226–2238.
   A useful parallel for AI evolution, where models may "inherit" structure horizontally through transfer learning or adaptation, similar to lateral gene transfer in biology.

8. Burr, G. W., Shelby, R. M., Sebastian, A., et al. (2017). *Neuromorphic Computing Using Non-Volatile Memory*. *Advances in Physics: X*, 2(1), 89–124.
   Discusses memory-embedded computing architectures relevant to physical model imaging and hardware-encoded AI.

9. Pascanu, R., Mikolov, T., & Bengio, Y. (2013). *On the Difficulty of Training Recurrent Neural Networks*. *Proceedings of the 30th International Conference on Machine Learning (ICML)*.
   Highlights challenges in deep learning training that contribute to the emergence of structural signatures which could be used in model phylogenetics.

10. MacClade Software (Maddison & Maddison).
    Popular software for biological cladistic analysis; methods described in this paper could be adapted to develop tools for AI model lineage analysis.

11. IEEE Standard for Floating-Point Arithmetic (IEEE 754).
    Describes binary encoding of numerical data, providing the basis for interpreting and comparing quantized and full-precision serialized models.

12. Li, H., Kadav, A., Durdanovic, I., Samet, H., & Graf, H. P. (2017). *Pruning Filters for Efficient ConvNets*. *International Conference on Learning Representations (ICLR)*.
    Relevant for detecting structural deletions (pruned weights) that can serve as cladistic traits in CNN-based models.

13. Krzanowski, W. J., & Marriott, F. H. C. (1994). *Multivariate Analysis Part I: Distributions, Ordination and Inference*. Wiley.
    Provides mathematical foundations for constructing distance matrices and hierarchical clustering, used here to generate AI phylogenetic trees.

14. Wigner, E. P. (1960). *The Unreasonable Effectiveness of Mathematics in the Natural Sciences*. *Communications on Pure and Applied Mathematics*, 13(1), 1–14.
    A philosophical reference relevant to the paper's discussion on the emergent structure of intelligence in engineered systems.