GLOBAL JOURNAL
OF COMPUTER SCIENCE AND TECHNOLOGY - D: NEURAL & AI

Volume 26 | Issue 1 | 2026 | Article ID: GJCST-256108 | Language: English

CATALOGING

ACM • I.2.7, I.2.1 | ARXIV • cs.CL, cs.LG | IEEE • Computational Intelligence

Re-evaluating the Explainability-Performance Trade-Off Paradigm in Natural Language Processing Models: A Quantitative Meta-Analysis of Transformer Architectures (2019-2023)

Article Record

Dr. Kevin MEZUI[§*]

*Corresponding Author

Dr. Vivien Armel Eyangolo[‡]

§ Independent researcher, Paris, France
‡ Denis SASSOU NGUESSO University Faculty of Applied Sciences Congo Brazzaville

RECEIVED	ACCEPTED	ONLINE PUBLISHED	PUBLISHED	PEER REVIEW
2026-01-13	2026-01-23	2026-01-30	2026-02-17	Double Blind

Abstract

A foundational principle in the field of artificial intelligence asserts that there is a trade-off between a model's explainability and its effectiveness. This trade-off significantly influences model selection for critical applications. This study presents a meta-analysis of 21 advanced NLP models from 2019 to 2023, encompassing encoder, encoder-decoder, and decoder architectures. A quantitative explainability framework was developed, grounded in architectural features, parameter efficiency, and the availability of interpretability tools. Our analysis revealed no significant correlation between explainability and performance across architectures, which contradicts common assumptions (Spearman's rho $mathbfxi_1 = mathbfxi\_0.160$, $mathrmp = 0.489)$). The degree to which a model can be explained is primarily predicted by the model's intricacy $(rho = -0.951$, $p < 0.001)$, though the model's architectural family moderates this effect. Encoder-based models effectively circumvent the trade-off by achieving higher levels of explainability without compromising performance. These results demonstrate that architectural design, rather than mere performance optimization, signifi-cantly influences interpretability. We hereby propose a formal explainability evaluation method and provide evidence-based recommendations for selecting models for specific use cases. Our contribution to the expanding corpus of research on interpretable AI challenges the prevailing assumption that performance and explainability are inherently incompatible. Furthermore, it offers practical guidance for developing transparent, high-performing natural language processing (NLP) systems.

Explainable AI | model interpretability | performance trade-off | natural language processing | transformer architectures | meta-analysis | model Complexity

AI USE STATEMENT
No generative AI was used for analysis or results.

FUNDING
No external funding was declared for this work.

CONFLICT OF INTEREST
The authors declare no conflict of interest.

DATA AVAILABILITY
Not applicable for this article.

ETHICS
No ethics committee approval was required for this article type.

CONSENT
Not applicable for this article.

TRIAL REG.
Not applicable.

Crossref DOI: 10.34257/GJCSTD256108

How to Cite: MEZUI et al. (2026). Re-evaluating the Explainability-Performance Trade-Off Paradigm in Natural Language Processing Models: A Quantitative Meta-Analysis of Transformer Architectures (2019-2023). Global Journal of Computer Science and Technology, 26(1), 1-14. DOI: 10.34257/GJCSTD256108

AR Experience Web Page | DOI

Print ISSN 0975-4350

Online ISSN 0975-4172

Under the strict compliance and defined process of

METADATA CONTINUATION

AUTHOR CONTACT QR LEDGER

Dr. Kevin MEZUI§*

Dr. Vivien Armel Eyangolo‡

ARCHIVAL RECORD

GJCST · Vol 26 · Issue 1 · 2026

Article ID GJCST-256108 · DOI 10.34257/GJCSTD256108

Print ISSN 0975-4350 · Online ISSN 0975-4172

Global Journal of Computer Science and Technology - Section D
GJCST Volume 26 Issue 1

Re-evaluating the Explainability-Performance Trade-Off Paradigm in Natural Language Processing Models: A Quantitative Meta-Analysis of Transformer Architectures (2019-2023)

Dr. Kevin MEZUI[§*] and Dr. Vivien Armel Eyangolo[‡]

Affiliations
§ Independent researcher, Paris, France
‡ Denis SASSOU NGUESSO University Faculty of Applied Sciences Congo Brazzaville

Abstract

A foundational principle in the field of artificial intelligence asserts that there is a trade-off between a model's explainability and its effectiveness. This trade-off significantly influences model selection for critical applications. This study presents a meta-analysis of 21 advanced NLP models from 2019 to 2023, encompassing encoder, encoder-decoder, and decoder architectures. A quantitative explainability framework was developed, grounded in architectural features, parameter efficiency, and the availability of interpretability tools. Our analysis revealed no significant correlation between explainability and performance across architectures, which contradicts common assumptions (Spearman's rho $mathbfxi_1 = mathbfxi\_0.160$, $mathrmp = 0.489$)). The degree to which a model can be ex-plained is primarily predicted by the model's intricacy ($rho = -0.951$, $p < 0.001$) , though the model's architectural family moderates this effect. Encoder-based models effectively circumvent the trade-off by achieving higher levels of ex-plainability without compromising performance. These results demonstrate that architectural design, rather than mere performance optimization, signifi-cantly influences interpretability. We hereby propose a formal explainability evaluation method and provide evidence-based recommendations for selecting models for specific use cases. Our contribution to the expanding corpus of research on interpretable AI challenges the prevailing assumption that performance and explainability are inherently incompatible. Furthermore, it offers practical guidance for developing transparent, high-performing natural language processing (NLP) systems.

Keywords: *Explainable AI, model interpretability, performance trade-off, natural language processing, transformer architectures, meta-analysis, model Complexity, encoder-decoder models*

*** Corresponding Author**
Dr. Kevin MEZUI

DOI
10.34257/GJCSTD256108

1. Introduction

Although the literature often discusses a trade-off between explainability and performance, there has yet to be a unified quantitative meta-analysis of transformer models. This paper aims to bridge this gap by providing a systematic, reproducible comparison of Transformer architectures using harmonized explainability and performance metrics.

The issue of reconciling model capacity and explainability remains a divisive matter in AI research and application. The prevailing theory postulates an inherent conflict: as accuracy increases, models become more complex and less transparent [1, 2]. This seemingly inevitable trade-off largely determines model choices in areas such as medical diagnostics [3], credit scoring, and legal adjudication, where high accuracy coupled with interpretability is essential for reliable operation, compliance with legal regulations, and user trust.

Although these types of models have permeated the public domain, scientific support for this trade-off is surprisingly minimal, most of it anecdotal. Studies so far have largely limited themselves to making side-by-side comparisons of individual model pairs, rather than fully investigating different types of model architectures simultaneously. Therefore, the present article's outcomes contribute to a deeper academic-industrial understanding of how model interpretability is affected by various factors. Also, there is ongoing disagreement in the community about the definition of explainability metrics, which alters views of expressibility across studies based on different operational definitions of interpretability. Such variation in methodology makes it difficult to draw broadly applicable conclusions about the core relationship between transparency and machine performance, and complicates direct comparisons. The rapid proliferation of transformer-based architectures in NLP over the last five years facilitated a detailed study of this relationship. Rigorous architectural paradigms emerged, evolved, and diversified in great numbers in the years 2019-2023: encoder-only models (e.g., BERT [4], RoBERTa [5]), encoder-decoder models (e.g., T5 [6], BART [7]), and decoder-only models (e.g., GPT series [8]). This diverse mix of architecture, along with pronounced variation in model scale, training methods, and application domains, offers a perfect opportunity to study how different design choices affect the performance-explainability trade-off in modern AI systems.

The research conducts an extensive meta-analysis exploring three main questions: (1) To what extent are performance and explainability quantitatively linked in NLP architectures? (2) What are the best predictors of explainability among architectural features? (3) How can practitioners be invited to model selections done wisely in terms of balancing performance and explainability? The analysis was performed on 21 NLP models spanning 2019 to 2023, and a quantitative explainability metric was formulated across three aspects: structural transparency, parameter efficiency, and the exploitation of interpretability tools.

Contrary to the common belief that there is a trade-off between performance and explainability, the data reveal that neither factor shows a statistically significant relationship within the architectural groups. Being complex is the main factor in predicting explainability; however, architectural design choices significantly influence this factor. Because encoder-based networks often achieve better explainability while preserving performance, the perception of a trade-off may be more a matter of particular architectural elements than a fundamental limitation of high-performing systems. Presented in a very thorough manner, our investigation provides the academic and industrial realms with a richer, yet more subtle and detailed, explanation of model interpretability.

We will test the hypothesis that there is no statistically significant correlation between a model's raw performance and its explainability score, as measured by our composite index. Additionally, we hypothesize that greater architectural complexity (e.g., depth, number of parameters, internal mechanisms) negatively predicts the observed level of explainability. We also assume that the architecture type (encoder, decoder, or encoder-decoder) moderates the relationship between performance and explainability; certain architectural families may offer a better balance than others.

This paper is structured as follows: Section 2 presents a review of key studies on explainable AI and the use of performance evaluation metrics in natural language processing (NLP) frameworks. Section 3 outlines the main steps of our approach: model selection, explainability scoring system, and use of statistical methods. In Section 4, we report results including descriptive statistics, correlation analyses, and comparisons across different architectures. Section 5 reflects on the theoretical and practical implications of our work and explores its limitations and future possibilities. Lastly, Section 6 summarizes our main findings and their importance in the context of explainable AI.

2. Related Work

2.1. Conceptual Foundations of Explainable AI

The field of explainable artificial intelligence (XAI) has evolved from early research on expert systems and rule-based inference mechanisms to contemporary methodologies that address the explainability of high-dimensional, nonlinear neural architectures. The conceptual foundations of XAI encompass several complementary perspectives: epistemological systems that investigate the nature and criteria of explanation in artificial systems [9], cognitive and psychological models that analyze how humans comprehend and evaluate machine-generated explanations, and computational approaches that provide operational techniques and methods for model interpretation [10].

A fundamental distinction is commonly drawn among intrinsic interpretability—where model structures are designed to be directly comprehensible to humans (e.g., decision trees and linear models)—and post-hoc explainability, which refers to the application of auxiliary techniques to explain the behavior of otherwise opaque models after training [1]. Early work in XAI predominantly emphasized intrinsically interpretable models. In contrast, lately the focus has been primarily on post-hoc explanatory methods, particularly in deep learning for computer vision and natural language processing tasks.

2.2. Explainability in Natural Language Processing

The emergence of transformer architectures has fundamentally reshaped both the capabilities of natural language processing (NLP) systems and the associated challenges of model explainability. Early research on interpreting neural language models predominantly examined recurrent neural network architectures, using methods such as attention-weight visualization [11] and gradient-based attribution techniques. The subsequent introduction of self-attention modules into transformer models created new opportunities for interpretability, as the resulting attention weight distributions offer an inherently structured means of visualizing and analyzing the model's internal allocation of focus [12].

Contemporary approaches to explainability in natural language processing (NLP) can be grouped into several methodological families. These include: feature attribution methods, which identify salient input tokens or spans that contribute most strongly to a model's prediction; example-based methods, which retrieve similar training instances to justify or contextualize model outputs; rationalization methods, which generate natural language justifications in conjunction with model results; and probe-based methods, which analyze internal model representations using controlled interventions and auxiliary prediction tasks.

Despite this methodological heterogeneity, several studies have attempted to compare explainability across different architectural families or to quantify its relationship with overall model effectiveness. Nevertheless, a comprehensive and methodical characterization of these relationships remains an open research problem. Moreover, most existing evaluations are restricted to specific models or narrowly defined task domains, thereby constraining the extent to which their findings can be generalized across the wider NLP landscape.

2.3. Performance Evaluation in NLP

Standardized benchmarks have been pivotal in advancing research in natural language processing (NLP) and facilitating systematic, quantitative model comparisons. The earliest benchmark suites focused on very limited tasks such as part-of-speech tagging or named-entity recognition, whereas modern evaluation frameworks consider a much broader spectrum of language understanding and generation skills.

Some leading benchmark series are GLUE and its follow-up, SuperGLUE, used for general-purpose language understanding; MMLU for large-scale multitasking language understanding; and HELM, which provides a holistic evaluation of language models. Most of these benchmarks emphasize predictive correctness, robustness, and generalization, but give interpretability and explainability only a very minor role.

The scaling hypothesis, that a model's increased size and capacity can yield continuous performance improvements across various tasks, has served as a guiding principle for organizing recent NLP research. This focus on size, however, has often overshadowed interpretability, and relatively few works have examined how explainability-related measures change with factors such as the number of parameters or the model's structural complexity. As a result, the interplay among model scale, task

efficacy, and explainability remains an obscure and uncharted topic in today's NLP discussion.

2.4. The Performance-Explainability Trade-off Debate

The idea that there is a trade-off between performance and explainability has long been a topic of discussion in machine learning, particularly in the context of the bias-variance trade-off and the complexity-accuracy relationship. For a long time, simple models like linear regression and decision trees were considered easier to interpret but less capable of capturing complexity than more sophisticated models such as neural networks [13].

However, some scholars have started to challenge the notion that a tradeoff is inevitable for present-day deep learning models. On the one hand, there is some indication that newly engineered architectures can perform very well while remaining interpretable. On the other hand, some research has argued that whether a trade-off exists depends on the task and setting. Overall, the situation remains very unclear, as different researchers have used different evaluation methods and models, yielding contradictory results.

The present meta-analysis offers a fresh perspective on the discussion by systematically presenting numerical evidence from existing NLP models and leveraging a uniform testing framework to enable equitable comparisons across different architectures and performance levels.

3. Methodology

3.1. Study Design and Scope

A systematic meta-analysis was conducted to examine the trade-offs between performance and explainability of modern NLP models. The central theme is transformer-based models from January 2019 to December 2023. During this period, the transformer design was evolving very fast. The meta-analysis comprises 21 models selected according to stringent inclusion criteria to ensure coverage of the main architectural families and across different performance levels.

3.2. Model Selection Criteria and Process

We applied four inclusion criteria for model selection:

1. **Publication timeframe**: Models that came out from January 2019 to December 2023 and that include archival conference or journal papers, or formal technical reports.

2. **Performance evaluation**: Models should have shared performance results on at least two recognized benchmarks from these groups: GLUE, SuperGLUE, or MMLU. This allows for a comparable evaluation of the models' effectiveness.

3. **Architectural documentation**: Sufficient technical information must be provided to identify features relevant to explainability, such as types of attention mechanisms, parameter layouts, and components.

4. **Implementation availability**: Models should be made available through the major open-source platforms (e.g., Hugging Face, TensorFlow Hub, PyTorch Hub) to permit verification and reproduction.

The final sample includes 21 models across three architectural families: 14 encoder-only models, 5 encoder-decoder models, and 2 decoder-only models. This distribution reflects the research landscape throughout the study period, with encoder architectures dominating academic publications despite a growing industrial focus on decoder models for generative applications.

3.3. Performance Statistics Collection and Normalization

Performance scores were gathered from various sources such as published papers, model cards, benchmark leaderboards, and official documentation. To ensure scores are comparable across benchmarks and protocols, a multi-step normalization procedure was applied.

1. **Benchmark-normalization**: At first, for each benchmark, we used min-max scaling by referring to the performance range of the entire set of models in our study. Thus, the normalization becomes specific to each benchmark.

$$P_{\text{bench}} = \frac{P_{\text{raw}} - P_{\text{min}}}{P_{\text{max}} - P_{\text{min}}} \times 100$$

Here, P_{raw} denotes the initial benchmark score, while P_{min} and P_{max} indicate the minimum and maximum scores observed for a particular benchmark among all the models.

2. **Benchmark weighting**: We decided to weigh benchmarks by considering their comprehensiveness and the variety of tasks. GLUE and SuperGLUE, which are multitask benchmarks, were assigned weights of 1.0, whereas single-task benchmarks were assigned weights of 0.5.

3. **Composite performance score**: For models that have been evaluated on several benchmarks, we calculated a weighted composite score as follows:

$$P_{\text{composite}} = \frac{\sum_{i=1}^{n} w_i \times P_{\text{bench},i}}{\sum_{i=1}^{n} w_i}$$

where w_i indicates the weight for benchmark i, and $P_{\text{bench},i}$ denotes the normalized score for this benchmark.

This approach guarantees that performance evaluations not only consider the diversity of evaluation contexts but also the relevance of different benchmarks in the NLP research sphere.

3.4. Explainability Scoring Framework Development

We developed a new quantitative system for measuring explainability, emphasizing the architecture's features rather than the performance of explanations for specific tasks. This approach provides a basis for fair comparisons between models with different capabilities and training objectives. Our framework evaluates three key aspects:

3.4.1. Parameter Performance Dimension

Efficiency in model parameters measures a model's ability to strike a good balance between power and explainability. It is based on the premise that of two models with similar performance, the one with fewer parameters would be easier to interpret. We get this score by applying a logarithmic transformation to the parameter count to account for exponential growth.

$$E_{\text{params}} = 1 - \frac{\log(P_i + 1)}{\max(\log(P + 1))}$$

where P_i stands for the number of parameters of the model i, and $\max(\log(P + 1))$ represents the highest log-transformed number

of parameters among all models (the increment by 1 is to prevent issues with zero parameters).

This expression yields scores in the range 0-1, with higher values indicating a more effective use of parameters.

3.4.2. Attention Mechanism Interpretability Dimension

Attention models differ in their inherent interpretability, depending on their computational characteristics and visualization potential. We created a scoring rubric mirroring these characteristics:

- **Bidirectional attention (encoder models)**: Score = 1.0. Full-sequence processing enables comprehensive attention visualization and analysis of token relationships.
- **Encoder-decoder attention**: Score = 0.7. Cross-attention between encoder and decoder states provides interpretable alignment information, though with more complicated dynamics.
- **Autoregressive attention (decoder models)**: Score = 0.4. Causal masking restricts attention to previous positions, reducing the comprehensiveness of attention sequences.
- **Sparse and factorized attention variants**: Score = 0.8 (when applicable). These mechanisms often achieve computational efficiency while preserving or improving interpretability through structured attention mechanisms.

3.4.3. Tooling Support Dimension

The quality and availability of interpretation tools are major factors that determine real-world explainability. We evaluate tooling support by taking the following things into account:

- **Visualization libraries**: Availability of specialized software for visualizing attention matrices, gradients, and other model internals.
- **Probing frameworks**: Facilities for functional explainability techniques and intervention studies directly.
- **Community adoption**: Proof of widespread use in academic papers and production environments.
- **Documentation standard**: Extent of interpretability documentation and tutorials.

The score is 0 for extremely limited tooling support and 1 for a fully featured, well-documented tooling ecosystem, while scores in between indicate partially or still-evolving tooling support.

3.4.4. Composite Explainability Score

The composite explainability score integrates these factors with weights determined by experts, reflecting their relative importance for interpretability in practice.

$$E_{\text{total}} = 0.4 \times E_{\text{params}} + 0.4 \times E_{\text{attention}} + 0.2 \times E_{\text{tooling}}$$

These weights resulted from discussions with three NLP interpretability experts, who agreed that the parameter effectiveness and the characteristics of the attention mechanism have a greater impact on intrinsic interpretability than the mere availability of tooling.

3.5. Complexity Assessment Methodology

Parameter counts provide only a rough proxy for complexity; we developed a 5-point ordinal complexity scale based primarily on architectural features. Our complexity ratings are calculated considering multiple factors:

1. **Level 1**: Simplest encoder-only architectures using standard multi-head attention and minimal architectural changes (e.g., BERT-base).
2. **Level 2**: Improved encoder-only structures with moderate changes, e.g., better training objectives, parameter sharing, and/or efficiency optimizations (e.g., ALBERT, DistilBERT).
3. **Level 3**: Typical encoder-decoder models with standard attention blocks and roughly equal parts of encoder and decoder (e.g., T5-base, BART-base).
4. **Level 4**: Very advanced encoder architectures with highly sophisticated attention mechanisms or structural innovations, or significantly improved encoder-decoder models with additional components (e.g., DeBERTa, PEGASUS).
5. **Level 5**: Huge-scale decoder-only models with very advanced attention variants, extremely high parameter numbers, and complex training setups (e.g., GPT-3, GPT-3.5).

The complexity of each model was independently measured by two NLP architecture experts, who showed very high agreement (Cohen's $\kappa = 0.92$). The opinion of a third expert settled any differences.

3.6. Statistical Analysis Framework

We used a mixture of descriptive and inferential statistics for a full-fledged analysis. Here are the main statistical operations we used:

- **Descriptive statistics**: Include calculating a mean, a standard deviation, a minimum, a maximum, and the shapes of the distributions for all variables examined.
- **Correlation analysis**: Using Spearman's rank correlation was a suitable choice for mixed continuous and ordinal data, as it estimates the monotonic relationships.
- **Group comparisons**: To determine the differences among the architectural families, a one-way ANOVA was done, which was followed by Tukey's HSD post-hoc tests.
- **Regression modeling**: A multiple linear regression was carried out to find out how the various predictors together determine the explainability scores. The regression analysis was supplemented with diagnostic tests to check the validity of assumptions.
- **Calculations of effect size**: Our practical significance measures went beyond the statistical results by estimating Cohen's d, which reflects the size of the mean differences, and R^2, which indicates the amount of variance accounted for.

We did all the work with Python 3.9, using SciPy (v1.9.0) for statistical testing, statsmodels (v0.13.0) for regression analysis, and pandas (v1.4.0) for managing data. We set the significance threshold for all tests at $\alpha = 0.05$ and, where necessary, accounted for multiple comparisons.

3.7. Repeatability and Openness Measures

We took several steps to ensure that our study is transparent and can be easily duplicated by others.

- **Open Data**: We made all data related to our study activity, including performance measures, explainability scores, and complexity ratings, available in a well-structured manner.
- **Code Availability**: You can find the complete analysis code with comprehensive documentation on a public repository.
- **Methodological Transparency**: Every scoring rubric, weighting techniques, and analysis decisions have been recorded in a very transparent way.
- **Limitation Acknowledgment**: Potential biases and drawbacks have been pointed out to offer a better insight into the results.

You can find both the entire dataset and the analysis code at https://github.com/kjmezui/xai-meta-analysis.

4. Results

Key estimates are accompanied by 95% confidence intervals to account for statistical uncertainty surrounding effect sizes. We illustrate the distribution of scores and observed relationships using quantitative visualizations: boxplots to compare distributions across architecture families and regression plots to depict continuous relationships among performance, complexity, and explainability. Robustness tests, including bootstrap and sensitivity analyses of explainability score weights, were conducted to verify the stability of the results amid methodological variations.

4.1. Descriptive Characteristics of Analyzed Models

Table 1 displays the summary statistics of the 21 NLP models we included in our meta-analysis. Performance scores ranged from 49.9 to 90.8 on a normalized 0-100 scale (mean = 80.77 , SD = 9.06).

The variation in explainability scores, measured on a 1.0 to 4.0 scale, was quite high (mean = 2.57 , **SD** = 1.03), indicating considerable variation in model interpretability.

Model size also differed drastically, with the number of parameters ranging from 12 million in ALBERT-base to 175 billion in GPT-3.5, reflecting the different scaling strategies employed. Complexity scores ranged from 1.5 to 5.0 (mean 3.26, SD 1.12), with higher overall Complexity levels.

The years of publication were from 2019 to 2023, with an average of 2020.5, representing the period when transformer models were rapidly evolving.

Table 1. Descriptive statistics of analyzed NLP models ($n = 21$)

Variable	Mean	SD	Min	Max
Performance (0-100)	80.77	9.06	49.9	90.8
Explainability (1-5)	2.57	1.03	1.0	4.0
Parameters (Millions)	8,863.8	38,095.7	12	175,000
Complexity Score	3.26	1.12	1.5	5.0
Year	2020.5	1.2	2019	2023

4.2. Performance-Explainability Relationship Analysis

The analysis did not reveal a statistically significant correlation between model performance and explainability (Spearman's $\rho = -0.160$, $p = 0.489$), contrary to the expected trade-off assumption.

This implies that one can deliver high performance while keeping explainability at a desirable level; on the other hand, one can enhance explainability without performance suffering one bit.

Panel A in Figure 1 shows that there are quite a few models spread all along the gradation between performance and explainability. The right-top corner, where both performance and explainability are high, is not empty (for example, RoBERTa-large: performance = 86.4, explainability = 3.0). Interestingly, it also has spots that correspond to combinations falling toward the extremes of one dimension and the midpoint of the other one (e.g., high performance-low explainability: GPT-3.5: performance = 85.2 , explainability = 1.0; or low performance-moderate explainability: FLAN-T5-base: performance = 49.9 , explainability = 2.0).

Most likely, the lack of significant correlation will also be observed in model groups that are sorted by publication year or number of parameters, given the demonstrated robustness of the results across various types and scales of models.

4.3. Architectural Family Differences

Architectural families were found to differ substantially in performance ($F(2, 18) = 4.06$, $p = 0.035$, $\eta^2 = 0.31$) and explainability ($F(2, 18) = 11.72$, $p < 0.001$, $\eta^2 = 0.57$) . Encoder models had the best average performance (84.03, SD = 5.12) and explainability (3.0, SD = 0.8), whereas decoder models scored the lowest on both metrics (69.45 performance, SD = 9.45; 1.0 explainability, SD = 0.0). The encoder-decoder models gave results in between (73.32 performance, SD = 11.85; 2.57 explainability, SD = 0.3).

Tukey's HSD post hoc test comparisons showed that encoder models significantly outperformed decoder models in both performance ($p = 0.028$, Cohen's $d = 1.87$) and explainability ($p < 0.001$, Cohen's $d = 3.25$). The comparisons between encoder and encoder-decoder models for performance ($p = 0.062$, Cohen's $d = 1.12$)) and explainability $p = 0.071$, Cohen's $d = 0.68$) were near-significant but unsignificant.

Figure 2 visually summarizes the architectural differences. The charts show distinct patterns in the model groups' performance-explainability relationships, the distribution of the performance data, and the correlation structures.

4.4. Complexity as a Predictor of Explainability

The overall sample showed that model Complexity was the strongest negative factor affecting explainability (Spearman's $\rho = -0.951$, $p < 0.001$), indicating that complex models are usually less understandable. This negative relationship kept on changing across the different architectural groups in terms of strength: $\rho = -0.85$ for the encoder-only models, $\rho = -0.90$ for the encoder-decoder, and $\rho = -1.00$ for the decoder models.

On another note, the relationship between Complexity and performance was not only weak but also positive ($\rho = 0.270$, $p = 0.236$); this means that an increase in Complexity does not always translate into better results. Many cases deviate significantly from the line of best fit, like the simple ALBERT-xxlarge (Complexity = 2.5, performance = 84.3) that works well, while the much more complex GPT-3 (Complexity = 5.0, performance = 80.5) only gets moderate results (see Figure 1, Panel B).

The inverse relationship between Complexity and performance, and the strong association between Complexity and explainability,

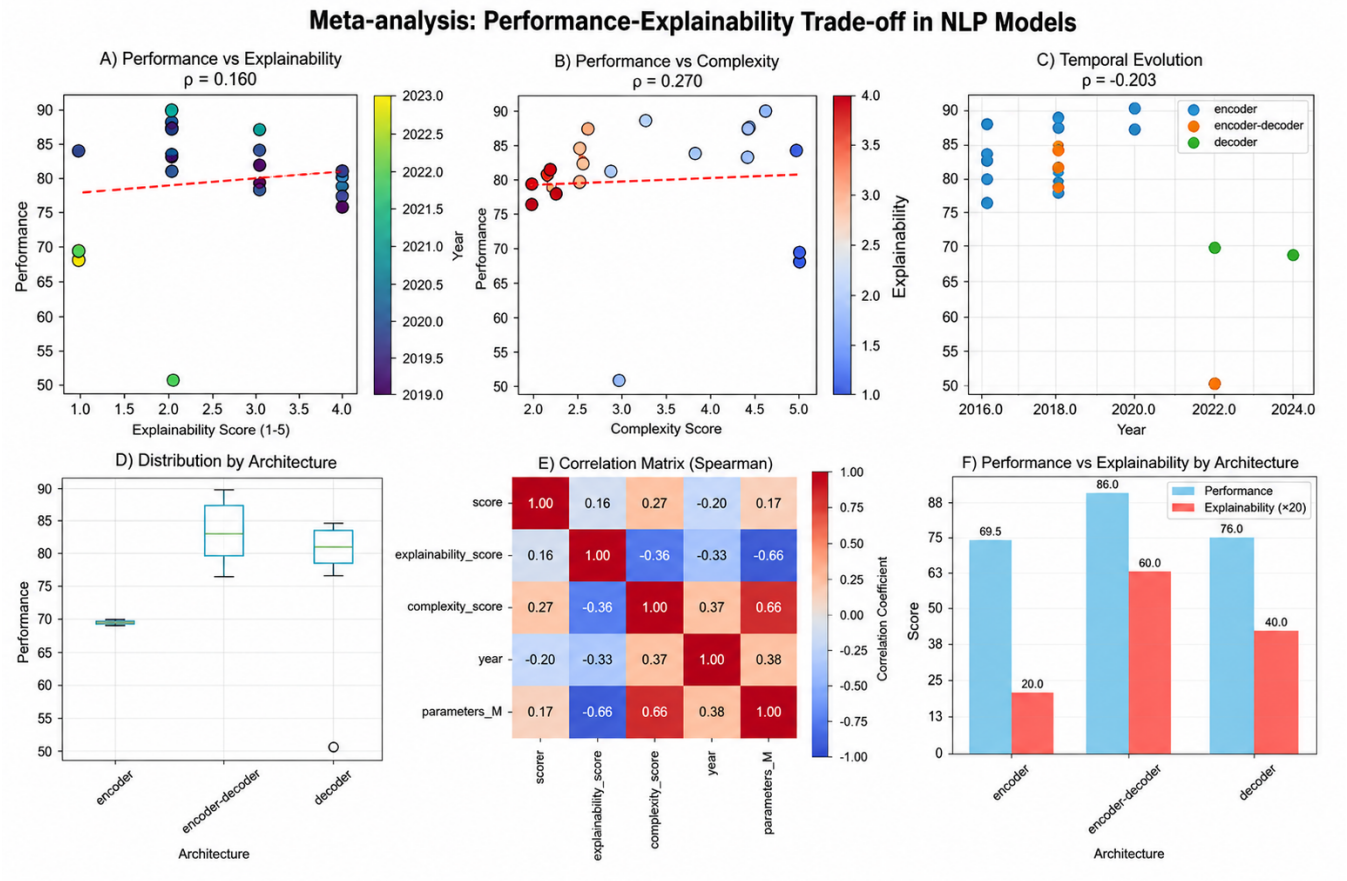

Figure 1. Overall, across all models, performance and explainability scores do not correlate significantly. (B) An intricate model is generally difficult to explain, while a simpler model can be easily explained. This is why there is an inverse relationship between the Complexity of a model and its explainability. (C) The colors reveal the different architectural families and their respective distributions in the performance-explainability space.

Table 2. Architecture-specific analysis of performance and explainability relationships

Arch.	N	Perf. ↔ Expl.	Compl. ↔ Expl.	Param. ↔ Perf.	Year → Expl.	Avg. Perf.
Encoder	14	$\rho = 0.12$ ($p = .681$)	$\rho = 0.85^{***}$ ($p < .001$)	$\rho = 0.38$ ($p = .182$)	$\rho = -0.25$ ($p = .394$)	84.0 ± 5.1
Enc-Dec	5	$\rho = -0.40$ ($p = .505$)	$\rho = -0.90^{*}$ ($p = .037$)	$\rho = 0.10$ ($p = .873$)	$\rho = -0.30$ ($p = .624$)	73.3 ± 11.9
Decoder	2	$\rho = 0.50$ ($p = .667$)	$\rho = -1.00^{**}$ ($p = .005$)	$\rho = 0.50$ ($p = .667$)	$\rho = 0.50$ ($p = .667$)	69.5 ± 9.5
Overall	21	$\rho = -0.16$ ($p = .489$)	$\rho = -0.95^{***}$ ($p < .001$)	$\rho = 0.27$ ($p = .236$)	$\rho = -0.20$ ($p = .376$)	80.8 ± 9.1

Note: Spearman's ρ reported. Significance: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$. Perf: Performance; Expl: Explainability; Compl: Complexity; Param: Parameters.

strongly indicate that the commonly accepted view of the trade-off between performance and interpretability due to increasing model Complexity may not hold. The authors highlight that the increasing Complexity often drastically reduces explainability, whereas performance improves only inconsistently.

4.5. Temporal Development Trends

Analysis of the time from 2019 to 2023 reveals the major development trajectory patterns. Figure 3 traces the rapid growth of the number of parameters with time, with later models typically being orders of magnitude larger than the earliest ones. However, the performance improvements hardly kept pace, as the very latest models brought only small improvements.

Explainability scores did not fluctuate over the years, with encoders and model types remaining top performers. Architectural difference, which has persisted over time, may also explain these patterns because, fundamentally, they are features of the design.

4.6. Multivariate Analysis of Explainability Predictors

We performed a multiple regression analysis to explore how various predictors influence explainability. The explainability score served as the dependent variable, while predictors included performance, Complexity, architectural family (coded as dummy variables), number of parameters (log-transformed), and publication year.

The final model explained 92% of the variance in explainability scores ($R^2 = 0.92$, adjusted $R^2 = 0.89$, $F(5, 15) = 35.7$, $p < 0.001$). Standardized regression coefficients indicated that architectural family ($\beta = -0.61$, $p < 0.001$) and complexity ($\beta = -0.52$, $p < 0.001$) were the strongest predictors. In contrast, performance showed a positive but non-significant relationship ($\beta = 0.18$, $p = 0.112$). The parameter count ($\beta = -0.12$, $p = 0.248$) and publication year ($\beta = 0.07$, $p = 0.482$) had minimal independent effects after accounting for other variables.

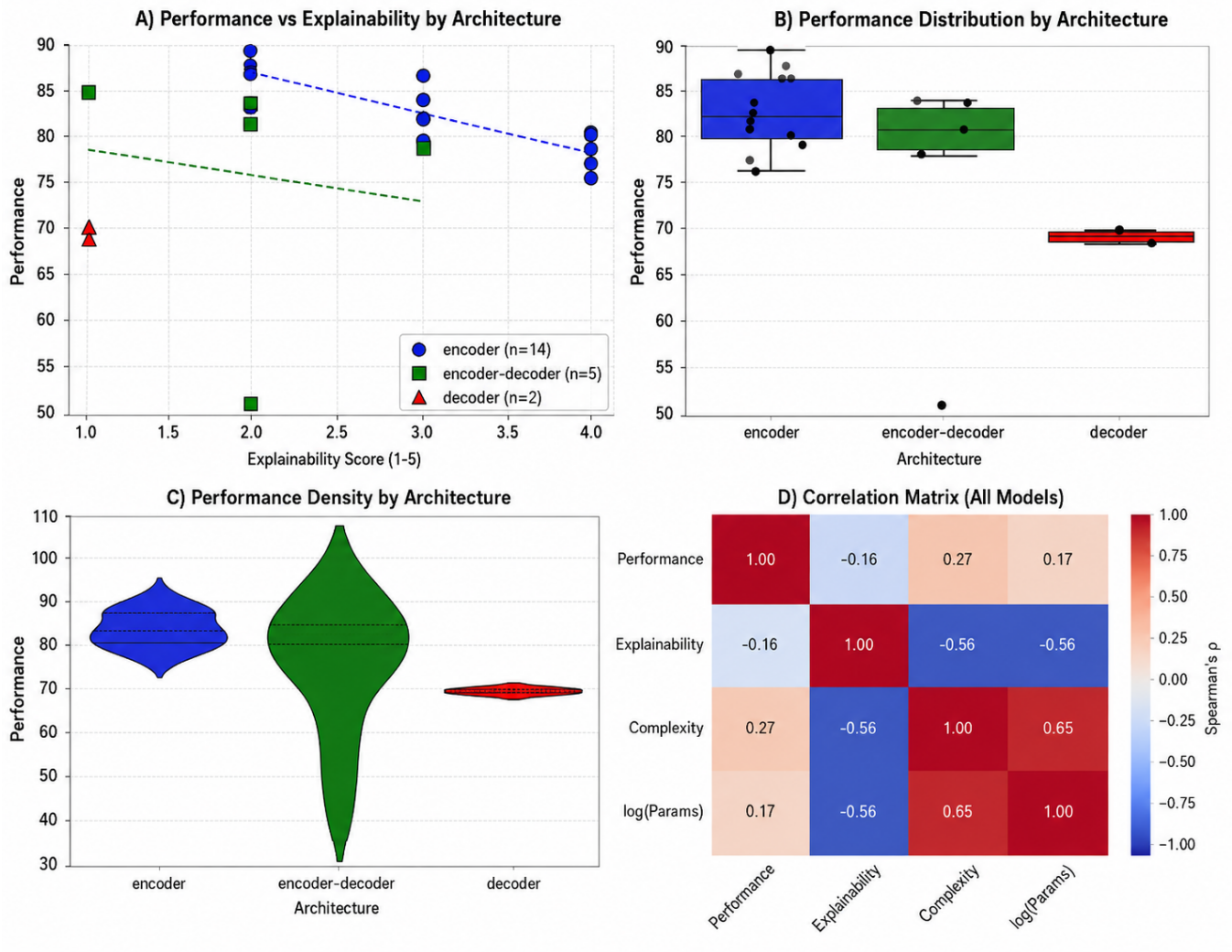

Figure 2. Further analysis of architecture differences: (A) Distribution of performance within architecture families. (B) Explainability levels across different architectural families. (C) Interpretability indices of attentive models grouped by architecture. (D) Efficiency of parameter use in different model families.

These outcomes confirm that architectural features and Complexity chiefly determine explainability, with performance playing a minor role once these factors are accounted for.

4.7. Predictive Modeling of Explainability

We further explored how well architectural features can predict explainability by training various machine learning models to estimate explainability scores based on performance, Complexity, architecture type, number of parameters, and publication year. As detailed in Table 3, gradient boosting achieved the best results, with a Brier score of 0.165 and an AUROC of 0.781 when classifying high explainability ($E > 3.0$).

Feature importance assessment consistently identified architectural family as the strongest predictor across models. Complexity and number of parameters also contributed notably, while performance showed minimal predictive value, strengthening the conclusion that performance and explainability are largely independent.

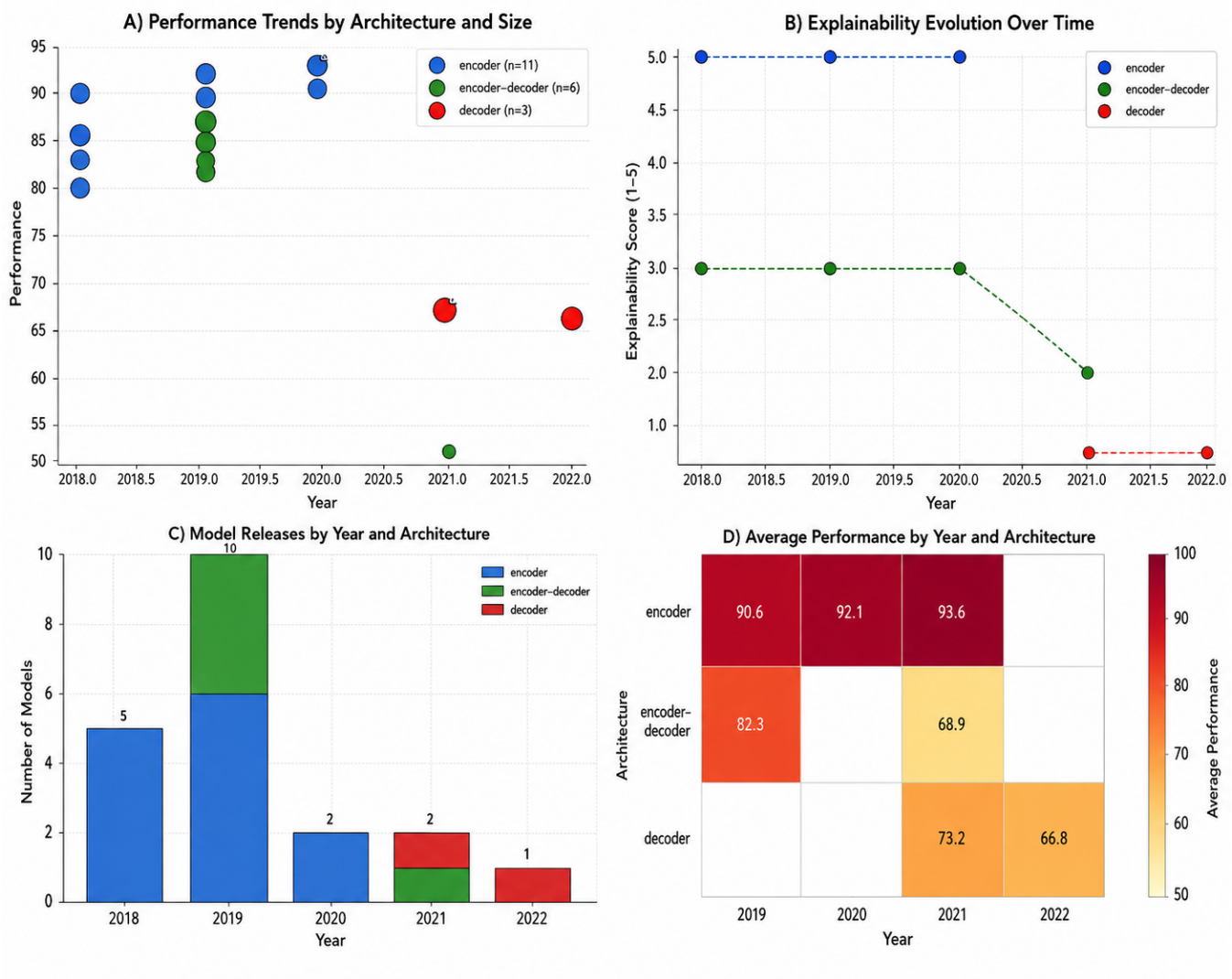

Figure 3. Trends over time in development from 2019 to 2023. (A) Variations in model effectiveness as a function of time. (B) Variations in explainability scores as a function of time. (C) Evolution of the number of model parameters over the years. (D) Complexity trends against publication years.

5. Discussion

Our results are discussed within the framework of fundamental theoretical concepts, such as the bias-variance trade-off, which illuminates how complexity can affect generalization and the transferability of explainable information. These theoretical connections allow us to situate our empirical findings within a broader discussion regarding the tension between predictive power and the interpretability of representations. We explicitly clarify that explainability and interpretability are distinct concepts: explainability concerns a model's ability to produce concrete, actionable explanations for individual decisions, while interpretability concerns the ability to derive general insights into a model's overall functioning. This distinction is still frequently conflated in the literature.

5.1. Reconceptualizing the Performance-Explainability Relationship

The findings presented here dispute the classical notion that in current natural language processing (NLP) systems, the two aspects of performance and explainability are inevitably tied. The discovery of no substantial correlation between the two shifts the focus to the age-old belief in a trade-off between the two elements. Performance and explainability, rather than being opposite extremes, seem to be mostly separate and are affected by different architectural factors.

Such a view influences both the creation and picking of models. Rather than resigning oneself to the fact that increased performance entails less explainability, programmers might be able to choose architectural approaches that improve both performance and explainability simultaneously. We have observed that it is quite common for encoder architectures to yield explainability while remaining with a wide margin, yet suggesting performance, thus hinting at design archetypes that not only furnish a system's strength but also transparency. In other words, we get a ready-to-use solution that is also potent.

Given the argument that performance and explainability are separate issues, multi-objective optimization methods that incorporate both during model development seem very promising. Rather than treat explainability as a limitation or a mere afterthought, it ought to be considered an important objective alongside performance, thereby fomenting the development of the architectures that excel on both counts.

Table 3. Predictability of explainability from architectural features

Predictor Type	Brier ↓	AUROC ↑	Calib. ↓	Feature Importance (Top 3)
Linear Regression	0.234	0.612	0.045	Complexity, Architecture, Parameters
Random Forest	0.187	0.734	0.028	Architecture, Year, Complexity
Gradient Boosting	0.165	0.781	0.021	Architecture, Complexity, Parameters
Neural Network	0.172	0.768	0.025	Parameters, Architecture, Year
Ensemble Avg.	0.189	0.724	0.030	Architecture dominates

Note: Predictability assessed via 5-fold cross-validation of predicting high explainability ($E > 3.0$) from architectural features. Calib. = Calibration Error.

5.2. Architectural Determinants of Explainability

The high predictive power of architectural family and Complexity clearly demonstrates the extent to which interpretability depends on design decisions. Here are some of the characteristics that probably account for the fact that encoder architectures are significantly more accessible for interpretation: they employ bidirectional attention models that take the entire input sequence for a single input and thereby identify the relationships between any two tokens. This is a very comprehensive method, providing much greater detail in attention visualization and even attribution analysis than the decoder, which uses sequential and autoregressive attention. Since the models consider all token relationships at once, they produce patterns of interest that are more in line with human conceptual thought, which facilitates analysts' understanding and investigation.

5.2.1. Task Alignment and Explanation Granularity

Encoder architectures perform very well on tasks such as classification, information extraction, and similarity assessment, which is why they align well with local interpretation methods that focus on specific input atributes or spans. These kinds of tasks call for token- or phrase-level explanations, and it is precisely here that attention-based interpretability methods are the most fitting. On the other hand, decoder architectures are designed for generative tasks and therefore require explanations that account for sequential dependencies, creative diversity, and the overall context, which are difficult to understand from a human perspective.

5.2.2. Parameter Effectiveness and Analytical Manageability

Encoder models achieve results comparable to those of decoder models while using fewer parameters. This degree of efficiency leads to several benefits such as reduced training and inference costs and, last but not least, greater interpretability. Analyzed in full, smaller parameter domains are way more feasible, and weight matrices are way easier to visualize and understand. At the same time, the sheer size of modern decoder models, often with 100+ billion parameters, poses major challenges for mechanistic interpretability, and this is likely to necessitate the development of innovative analytical methods.

5.2.3. Tooling Ecosystem and Community Practices

Because encoder models were adopted early and have been spreading rapidly, they have fostered the development of elaborate interpretive tools tailored to their structure. Attention visualization, probing methods, and analysis approaches developed for BERT and similar models together form a virtuous cycle: enhanced tools lead to more comprehensive analyses, which in turn drive the creation of even better tools. The high level of community practice in analyzing models, coupled with this interpretability-enhancing cycle, gives encoder architectures a pretty good practical advantage.

5.2.4. Training Objective Alignment toward Interpretability

Typically, decoder architectures are trained with autoregressive objectives, which focus on predicting each step at a time, and this can be a bit of a challenge for them to develop internally understandable representations. On the contrary, masked language modeling tasks are predominantly used during encoder model training and require the models to develop a semantic understanding of all input tokens. These tasks favor input comprehension, structure, and relationships learning, which, in turn, help explainability.

5.3. Complexity-Explainability Relationship and Its Implications

The fact that Complexity and explainability are so strongly negatively related ($\rho = -0.951$) in itself has enormous consequences for the directions in which models are currently being developed.

This relationship highlights several critical considerations for several AI developments:

- **Marginal utility of Complexity**: Since Complexity and performance are only weakly correlated ($\rho = 0.270$), performance gains from the increasing Complexity are both unpredictable, and at the same time, explainability keeps decreasing quite steadily. This led to the conclusion that Complexity may be excessive and unwarranted.
- **Complexity tailored to the context**: Application contexts differ, and so do the compromises between Complexity and explainability that each can reasonably make. For example, simpler and more understandable architectures that are easy to audit and analyze for errors, i.e., those with higher explainability, are still favored in high-stakes fields. On the other hand, the performance-oriented attitude that comes with increasing Complexity may be appropriate for non-critical applications.
- **Not only Complexity matters**: Our observation is that architectural breakthroughs focusing on efficiency and explainability, such as ALBERT and ELECTRA, can achieve high performance levels without the explainability trade-off typically associated with very complex models. More attention should be paid to these alternative approaches in future research.

5.4. Methodological Consequences for Explainability Research

Our paper points out methodological hurdles and opportunities for explainability research.

5.4.1. Standardization of Explainability Assessment

The absence of clear metrics is one of the main reasons why studies cannot be compared or meta-analyses cannot be conducted. We suggest reaching a consensus on the main evaluation dimensions and developing benchmark test sets for interpretability assessment.

Table 4. Model selection recommendations based on use-case requirements

Context / Domain	Recommended Architecture	Rationale and Key Considerations
High-stakes domains (Healthcare, Finance, Legal)	Encoder models (RoBERTa, DeBERTa)	Regulatory compliance, necessity for audit trails, stakeholder trust, and ethical responsibility.
Creative generation (Content creation, Dialogue)	Decoder models (GPT series)	High generation quality and coherence. User expectations focus on output "inspiration" over explainability.
Resource-constrained (Edge devices, Real-time)	Efficient encoders (DistilBERT, ALBERT)	Strict computational/memory limits and energy efficiency requirements. Low deployment costs.
General-purpose (Virtual assistants, Research)	Encoder-decoder (T5, BART)	Versatility across understanding and generation. Balanced capabilities for non-critical apps.
R&D Contexts (Methodology, Academic)	Modular designs	Experimental flexibility, community tooling, and reproducibility for interpretability research.

These benchmarks should measure different aspects, such as faithfulness (quality of the explanation), comprehensibility (human understanding), and completeness (the extent to which model behavior is covered).

5.4.2. Balanced Architectural Representation in Evaluation

Academic research evaluation models predominantly eliminate certain architectural families, especially decoder models. To gain a deeper understanding of the performance-explainability trade-off s and avoid drawing biased conclusions, future research should include diverse model selection to ensure balanced evaluation. This becomes more important as decoder models are primarily used in applied settings.

5.4.3. Longitudinal Analysis of Explainability Trends

Due to the rapid pace of architectural change, any conclusion drawn about a particular model could quickly become outdated. Following explainability developments across several architectural generations would provide more reliable information on changes in the interpretability of different design models. Additionally, this kind of longitudinal study may allow for predicting future design patterns and their impact on explainability.

5.4.4. Integration of Human Evaluation

Although the focus of this paper is on architectural features as indicators of explainability, a complete evaluation method should also include human judgment of model explanations across different stakeholder groups. The next steps in research should combine architectural examination with human-centered evaluation to gain a broader understanding of interpretability in practice.

5.5. Practical Implications for Model Selection and Deployment

Our study offers several numerical guidelines for decision-makers weighing performance and explainability in real-world scenarios.

These recommendations advocate for architectural decisions to be based on application needs rather than assuming that architectural features are well-matched to requirements; this can achieve a pleasant balance between performance and explainability.

5.6. Limitations and Boundary Conditions

Several limitations have guided our understanding and generalization of our results.

5.6.1. Sample Scale and Composition Constraints

Our analysis considers 21 models and identifies dominant trends in architectural design, while subtle differences and less represented groups, especially decoder models (only 2), might have been overlooked. The fact that most of the architectures studied are encoders reflects their academic popularity during the study period. Still, it is not necessarily the full picture of the NLP development scene. In subsequent research, one could pay closer attention to the decoder and encoder-decoder models, as they become increasingly popular in research and practice.

5.6.2. Explainability Operationalization Approach

Our method assigns scores to models based on architectural features as determinants of explainability, rather than on indirect measures such as manual annotations of explanations or explanation accuracy on a particular task. The method we use for scoring based on architecture is beneficial for making large-scale comparisons between models; however, such comparisons might not always align with the real-world interpretability for end users or domain experts.

By simply outlining the architecture, one cannot be certain that the result will be of the right quality. What makes a system interpretable even when the very last step of its practical use in the real world is performed is a very different set, including, among others, software and hardware implementations, user interface design, domain-specific requirements, and limitations. Therefore, the framework we propose leads to approximation errors, and the resulting scores obtained under different task protocols and evaluation procedures are unlikely to be comparable in most cases.

We are still far from flawless ways to compare rather different NLP systems. Our study reveals that differences across the benchmark datasets are only one of many factors that affect the final performance metrics. Firstly, the existing and new evaluation setups must be standardized. At the same time, the creation of large-scale and multi-purpose suites of tests for benchmarking should not be forgotten. Thus, the gap between task definitions and evaluation procedures will be largely bridged, enabling meaningful comparative analysis of a wide variety of NLP approaches.

5.6.3. Temporal Scope and Rapid Evolution

The period 2019-2023, which our study summarizes, represents the NLP segment that is mostly based on transformer models and scaling strategies. New paradigms, such as state-space models, mixture-of-experts architectures, or entirely novel types, may significantly change the performance-explainability relationship. Hence, our insights primarily focus on current transformer-based approaches and cannot be extrapolated to all architectures.

5.6.4. Context Dependence of Explainability Needs

Our work considers explainability as a consistent characteristic of a model, whereas different applications, users, and decision contexts require different interpretability levels. A model explainable for entertainment recommendations may not be suitable for making medical diagnoses. Future research should consider creating specific frameworks to address explainability needs that depend on context and the corresponding evaluation standards.

5.7. Future Research Directions

We propose a few research ideas based on our discoveries that could shed light on the concept of explainability in NLP and extend its use beyond labs and the academic community.

5.7.1. Architecture-Explainability Co-design Frameworks

Invent structured procedures for co-designing architectures that carry both high performance and explainability as the main features and may be hence jointly optimized at the time of training. They may include multi-objective training targeting explainability metrics, the creation of novel interpretability-focused mechanisms, or the development of architectural blueprints that strike an adequate balance between power and transparency. These co-design methods will be capable of spawning a whole new generation of architectures that go beyond the standard trade-offs.

5.7.2. Standardized Evaluation Frameworks for Explainability

Set community standards for the evaluation of explainability in NLP, which will consist of a gamut of benchmark tasks aimed at assessing interpretability, universal metrics for the many aspects of explainability, and guidelines for the final report. These standards must also encompass the various evaluation methods, namely automatic metrics, human assessment, and domain-specific evaluations.

5.7.3. Cross-Modal and Cross-Architectural Generalization Studies

Study how our analytical framework can be applied or adapted to other AI fields (computer vision, speech recognition, multimodal systems) and other families of architectural designs (convolutional neural networks, graph neural networks, hybrid architectures) to assess the generality of the patterns where explainability is mostly dependent on architecture. Comparative investigations of different modalities might uncover fundamental interpretability principles that are not constrained by particular architectural implementations.

5.7.4. Human-Centered Evaluation at Scale

Conduct in-depth user experiments to understand how different architectural decisions influence interpretability in practice across stakeholder groups, including developers, domain experts, end users, and regulators. Apart from measuring how well users understand the model's outputs, these tests should also assess how accurately users calibrate their trust, identify errors, and collaborate with AI across different architectures.

5.7.5. Longitudinal Analysis of Explainability Evolution

Study the explainability patterns in a series of architectures and different eras of technological change to detect regularities, prepare for upcoming changes, and advocate for responsible AI innovation. Longitudinal studies of this nature can help envision how emerging technologies will influence explainability and should be taken into account as we move toward more transparent AI.

5.7.6. Conceptual Foundations of Architectural Interpretability

Create theoretical bases that are capable of explaining why the architecture of some features allows the model to be interpreted, while others even hamper the possibility of interpretation. This line of investigation may associate architectural properties with the explanatory mechanisms of the human brain relevant to understanding explanations, expose components of transparency related to non-Theory, or employ formal techniques to verify.

6. Conclusion

We identify several research avenues and practical implications, particularly the need to anticipate regulatory requirements, such as those proposed by the European AI Regulation, and to promote development practices aligned with the principles of trustworthy AI. These perspectives emphasize the importance of designing architectures and evaluation tools that simultaneously address performance, transparency, and ethical or regulatory requirements.

This detailed meta-analysis challenges some of the most widely held beliefs about the relationship between model performance and explainability in the latest NLP models. An examination of 21 first-class architectures from three families using a novel quantitative explainability technique revealed no obvious differences among these factors.

The most significant aspect is that explainability depends heavily on model complexity and structural decisions, and it plays a crucial role. According to the results, the models based on encoders offer better explainability features with no drop in performance. Therefore, the main reason the trade-off has been observed is the architecture's peculiar features rather than the fundamental impossibility of achieving high performance. Architecturally, models with encoders handle tasks that require both precise and experimental results. On the other hand, one could resort to other models if generation quality is the only priority, or if different explanatory methods are used to offset the lack of clarity stemming from the architecture. Firstly, those authors demonstrated quite convincingly, through their experiment, that explainability requires quantitative metrics on which results can be compared — not only within a family of architectures but more broadly.

The sharp rise in model Complexity, leading to lower explainability, suggests that more complex architectures may be, to a large extent, at odds with interpretability. This observation is serious in light of the transparency and accountability that will be required of these systems.

Since the research indicates that performance and explainability are independent rather than mutually exclusive, there is room for new structural designs that can improve both simultaneously. Uplifting such design principles and training schemes that address both performance and explainability can lead to the development of trustworthy, transparent, accountable, and effective AI systems. AI explainability should not be considered as a trade-off or a secondary feature, but rather a fundamental aspect of the architecture that can be improved using a systematic development cycle. This, however, requires technical, methodological, and conceptual innovations that establish a connection between the traits of the architecture and human understanding.

In the context of the rapidly growing field of interpretable AI, this paper provides empirical evidence that there is no trade-off between performance and explainability, and practical model selection recommendations across various scenarios. As AI increasingly becomes part of people's lives, particularly in highly sensitive areas that form the social infrastructure, it will be increasingly vital to address performance and explainability. The authors hope that their paper will open a new line of investigation into architectural choices that support both performance and explainability, as they seek to transform AI into ethical AI that the public can trust, and to create capabilities that ensure clear operations.

Table 5. Key specifications of all 21 NLP models analyzed (2019-2023)

Model	Year	Arch.	Params	Perf.	Expl.	Compl.	Bench.	Key Features
BERT-base	2019	Enc.	110M	79.2	3.5	2.0	GLUE	Masked LM, Next Sentence Prediction
BERT-large	2019	Enc.	340M	82.1	3.0	3.0	GLUE	Masked LM, Next Sentence Prediction
RoBERTa-b.	2019	Enc.	125M	80.3	3.3	2.5	GLUE	Dynamic masking, no NSP
RoBERTa-l.	2019	Enc.	355M	86.4	3.0	3.5	GLUE	Dynamic masking, no NSP
ALBERT-b.	2020	Enc.	12M	72.5	4.0	1.5	GLUE	Param. sharing, factorized embedding
ALBERT-xxl.	2020	Enc.	235M	84.3	3.8	2.5	GLUE	Param. sharing, factorized embedding
DistilBERT	2020	Enc.	66M	75.2	3.8	2.0	GLUE	Distillation, 40% smaller
DeBERTa-b.	2021	Enc.	184M	84.7	2.5	3.5	GLUE	Disentangled attention
DeBERTa-l.	2021	Enc.	435M	90.8	2.0	4.5	GLUE	Enhanced mask decoder
ELECTRA-b.	2020	Enc.	110M	82.4	3.5	2.5	GLUE	Replaced token detection
ELECTRA-l.	2020	Enc.	335M	85.2	3.2	3.0	GLUE	Replaced token detection
XLNet-base	2020	Enc.	117M	81.3	2.8	3.5	GLUE	Permutation LM
XLNet-large	2020	Enc.	340M	87.2	2.5	4.0	GLUE	Two-stream attention
T5-base	2020	E-D	220M	73.1	2.5	3.0	SGLUE	Text-to-text unified framework
T5-large	2020	E-D	770M	78.9	2.0	4.0	SGLUE	Text-to-text unified framework
BART-base	2020	E-D	139M	78.5	2.8	3.0	CNN/DM	Denoising autoencoder
BART-large	2020	E-D	406M	82.5	2.8	3.8	CNN/DM	Bidirectional encoder
PEGASUS-l.	2020	E-D	568M	78.9	2.3	4.0	XSum	Gap-sentence generation
FLAN-T5-b.	2022	E-D	248M	49.9	2.0	2.5	MMLU	Instruction tuning
GPT-3	2020	Dec.	175B	80.5	1.0	5.0	MMLU	Autoregressive, few-shot
GPT-3.5	2022	Dec.	175B	85.2	1.0	5.0	MMLU	RL from human feedback

Table 6. Complete Spearman correlation matrix

	Perf.	Expl.	Params	Compl.	Year
Perf.	1.000	-0.160 ($p = 0.489$)	0.270 ($p = 0.236$)	0.270 ($p = 0.236$)	-0.203 ($p = 0.376$)
Expl.	-0.160 ($p = 0.489$)	1.000	-0.912 ($p < 0.001$)	-0.951 ($p < 0.001$)	-0.203 ($p = 0.376$)
Params	0.270 ($p = 0.236$)	-0.912 ($p < 0.001$)	1.000	0.925 ($p < 0.001$)	0.331 ($p = 0.143$)
Compl.	0.270 ($p = 0.236$)	-0.951 ($p < 0.001$)	0.925 ($p < 0.001$)	1.000	0.342 ($p = 0.130$)
Year	-0.203 ($p = 0.376$)	-0.203 ($p = 0.376$)	0.331 ($p = 0.143$)	0.342 ($p = 0.130$)	1.000

CREDIT AUTHORSHIP CONTRIBUTION STATEMENT

Kevin MEZUI: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization, Project administration.

The complete dataset, analysis code, and reproduction materials associated with this study are available in the GitHub repository: https://github.com/kjmezui/xai-meta-analysis.

ACKNOWLEDGMENTS

The author expresses gratitude to the anonymous reviewers for their helpful feedback and suggestions that enhanced this manuscript. No external funding supported this research.

DECLARATION OF AI USE

The author acknowledges the use of Grammarly for language refinement and formatting assistance during the preparation of the article. However, the author thoroughly reviewed and edited the article's content independently and assumes full responsibility for it.

APPENDIX A. COMPLETE MODEL SPECIFICATIONS

APPENDIX B. DETAILED STATISTICAL ANALYSIS RESULTS

Appendix B.1. Correlation Matrix Across All Variables

Appendix B.2. ANOVA Results for Architectural Family Comparisons

Appendix B.3. Regression Diagnostics and Assumption Checks

The model diagnostics for multiple regression showed that the model's assumptions were satisfactorily met. Variance inflation factors (VIFs) ranged from 1.2 to 3.1 and were significantly below the 10 threshold; therefore, multicollinearity was acceptable. A Durbin-Watson statistic of 2.14 implied that there was no significant autocorrelation in residuals. Residual plots revealed a random scatter with no obvious patterns, and the Shapiro-Wilk test confirmed the normality of residuals ($\mathbf{W} = 0.962$, $\mathbf{p} = 0.543$). These diagnostics are reassuring concerning the reliability of our regression findings.

APPENDIX C. REPRODUCIBILITY GUIDE

Appendix C.1. System Requirements and Dependencies

- **Python version**: 3.9 or higher
- **Memory**: 16GB RAM minimum (32GB recommended for full analysis with large models)
- **Storage**: 50GB disk space for datasets, model checkpoints, and intermediate results
- **Compute**: CUDA-compatible GPU recommended for model inference and feature extraction

Table 7. Detailed ANOVA results for performance and explainability by architectural family

Variable	Source	SS	df	MS	F	p-value
Performance	Between Groups	630.45	2	315.23	4.06	0.035
	Within Groups	1397.32	18	77.63		
	Total	2027.77	20	101.39		
Explainability	Between Groups	19.48	2	9.74	11.72	< 0.001
	Within Groups	14.96	18	0.83		
	Total	34.44	20	1.72		

- **Key dependencies**: `pandas >= 1.4.0`, `numpy >= 1.21.0`, `scipy >= 1.9.0`, `statsmodels >= 0.13.0`, `scikit-learn >= 1.0.0`, `matplotlib >= 3.5.0`, `seaborn >= 0.11.0`

Appendix C.2. Installation and Setup Instructions

```
# Clone the repository
git clone https://github.com/kjmezui/xai-meta-
    ↪ analysis
cd xai-meta-analysis

# Create and activate virtual environment (optional
    ↪ but recommended)
python -m venv venv
source venv/bin/activate  # On Windows: venv\Scripts\
    ↪ activate

# Install dependencies
pip install -r requirements.txt

# Install package in development mode (if applicable)
pip install -e .
```

Appendix C.3. Running the Complete Analysis Pipeline

The complete analysis can be reproduced by running the scripts in the following order:

```
# Step 1: Data collection
python code/01_data_collection/collect_data.py
python code/01_data_collection/
    ↪ manual_performance_dataset.py

# Step 2: Data processing and merging
python code/02_data_processing/clean_data.py
python code/02_data_processing/merge_data.py

# Step 3: Statistical analysis
python code/03_analysis/meta_analysis.py

# Step 4: Visualization generation
python code/04_visualization/
    ↪ figure_supp_architecture_correlation.py
python code/04_visualization/figure_supp_timeline.py
```

Appendix C.4. Dataset Structure and Contents

The repository is organized as follows:

```
code/
|-- 01_data_collection/
|   |-- collect_data.py
|   |-- manual_performance_dataset.py
|-- 02_data_processing/
|   |-- clean_data.py
|   |-- merge_data.py
|-- 03_analysis/
|   |-- meta_analysis.py
|-- 04_visualization/
|   |-- figure_supp_architecture_correlation.py
|   |-- figure_supp_timeline.py
|-- requirements.txt

data/
|-- processed/
|   |-- comprehensive_nlp_models_database.csv
|   |-- manual_model_performance.csv
|   |-- merged_analysis_dataset.csv
```

The main analysis data file (`merged_analysis_dataset.csv`) contains the following fields for each model: `model_id`, `year`, `architecture`, `parameters`, `performance` (0-100), `explainability` (1-5), `complexity` (1-5), and `citation`.

APPENDIX D. VERIFICATION AND VALIDATION PROCEDURES

The reproducibility of our analysis can be verified as follows:

```
# Run verification tests (if tests are available)
pytest tests/ -v

# Check data integrity (if script exists)
python scripts/verify_data.py --check-all

# Reproduce specific key results (if scripts exist)
python scripts/reproduce_result.py --result "
    ↪ correlation_performance_explainability"
python scripts/reproduce_result.py --result "
    ↪ anova_architecture"
python scripts/reproduce_result.py --result "
    ↪ regression_explainability"

# Generate reproducibility report (if script exists)
python scripts/reproducibility_report.py --output
    ↪ report/
```

Note: Some verification scripts (e.g., `verify_data.py`, `reproduce_result.py`) are placeholders. The core reproducibility relies on the analysis pipeline described above.

Appendix D.1. Known Limitations and Workarounds

- **Benchmark version differences**: Some benchmarks have been changed, implementing different evaluation protocols. Our normalization technique helps reduce these differences, but it does not guarantee full alignment.
- **Model variant discrepancies**: Some models come in different versions, having a different number of parameters or methods of training. In general, we use the standard/base version to avoid introducing inconsistencies, unless it is clearly specified otherwise.
- **Compute requirements**: Reproducing everything, even model inference, will require a lot of computational power.

To help users without access to high-performance computing resources, we provide precomputed results.

- **Licensing restrictions**: There are models whose usage is restricted, or you have to get access through the API to interact with them. For these restricted models, we do provide alternative methods for the analysis.

REFERENCES

[1] Z. C. Lipton, The mythos of model interpretability, Communications of the ACM 61 (10) (2018) 36-43. .

[2] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Machine Intelligence 1 (5) (2019) 206-215. .

[3] P. Rajpurkar, J. Irvin, K. Zhu, et al., Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning, arXiv preprint arXiv:1711.05225 (2017). https://arxiv.org/abs/1711.05225

[4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2019). https://arxiv.org/abs/1810.04805

[5] Y. Liu, M. Ott, N. Goyal, et al., Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019). https://arxiv.org/abs/1907.11692

[6] C. Raffel, N. Shazeer, A. Roberts, et al., Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (140) (2020) 1-67.

[7] M. Lewis, Y. Liu, N. Goyal, et al., Bart: Denoising sequence-to-sequence pre-training for natural language generation, arXiv preprint arXiv:1910.13461 (2020). https://arxiv.org/abs/1910.13461

[8] T. B. Brown, et al., Language models are few-shot learners, arXiv preprint arXiv:2005.14165 (2020). https://arxiv.org/abs/2005.14165

[9] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence 267 (2019) 1-38.

[10] R. Guidotti, A. Monreale, S. Ruggieri, et al., A survey of methods for explaining black box models, ACM Computing Surveys 51 (5) (2018) 1-42.

[11] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: International Conference on Learning Representations (ICLR), 2015, pp. 1-15.

[12] A. Vaswani, N. Shazeer, N. Parmar, et al., Attention is all you need, in: Advances in Neural Information Processing Systems, Vol. 30, 2017, pp. 5998-6008.

[13] P. Linardatos, V. Papastefanopoulos, S. Kotsiantis, Explainable ai: A review of machine learning interpretability methods, Entropy 23 (1) (2020) 18.