Architectural Roots of Trust: A Systematic Review of Transformer Variants for LLM Safety

Anonymous ACL submission

Abstract

The emergence of deconstructionism opened new horizons for modern art by dismantling established conventions, the maturing field of LLM safety requires similar approach moving beyond surface-level analyses to dissect the foundational architectures of Large Language Models (LLMs) and scrutinize their inherent trade-offs. While extensive research focuses on post-hoc mitigations, a systematic analysis of how architectural choices influence Trust and Safety (T&S) outcomes remains underdeveloped. This paper conducts a systematic review of the three primary transformer families—Encoder-only, Decoder-only, and Encoder-Decoder—from a T&S perspective, analyzing their predispositions to critical failure modes like bias amplification, adversarial robustness, and hallucination. This review reveals that each architecture presents a unique T&S trade-off profile, intrinsically linked to core mechanisms like bidirectional context encoding versus autoregressive generation. Synthesizing the current literature, this work identifies significant research gaps and outlines three promising frontiers for enhancing model safety: Causal Transformers, Formally Verified Architectures, and Biologically-Inspired designs. To demonstrate their potential, the paper presents conceptual frameworks for testing the viability of these research directions. This study provides comprehensive insights into existing LLM safety evaluation methods. It introduces a structured overview of safety techniques that can be applied to current transformer architectures, while also exploring future paths for embedding safety directly into the architectural design itself.

1 Introduction

The introduction of the transformer architecture by Vaswani et al. (2023) marked a watershed moment in Natural Language Processing (NLP). By replacing recurrent mechanisms with self-attention, transformers enabled parallel processing of sequential data, leading to models of unprecedented scale and capability. This innovation became the foundation for models like BERT (Devlin et al., 2019) and the GPT series, which have revolutionized tasks from generation to complex reasoning. However, the rapid adoption of these powerful models has surfaced profound concerns regarding their trust and safety (Liu et al., 2024a). The very capabilities that make LLMs powerful also make them susceptible to failure modes that can cause significant harm, including the generation of biased content, propagation of misinformation, and vulnerability to prompt injection attacks (Chua et al., 2024; Perez and Ribeiro, 2022). Ensuring their safe deployment, especially in high-stakes domains, has become a paramount scientific and ethical imperative.

1.1 The Research Gap: An Architectural Blind Spot

In response, a substantial body of research has emerged focusing on LLM safety, cataloging vulnerabilities and evaluating mitigation strategies (Liu et al., 2024a). Despite this extensive work, a critical gap persists: a dedicated systematic review that directly connects fundamental architectural choices within the transformer family to their inherent T&S vulnerabilities. While many studies compare architectures on performance metrics, few have systematically analyzed how the decision to use an encoder-only, decoder-only, or encoder-decoder structure fundamentally shapes a model's safety profile. This paper aims to fill this "architectural blind spot," arguing that the industry's rapid adoption of specific architectures may proceed without a full appreciation of the inherent safety liabilities "baked in" to that choice.

Table 1: Comparative Trust & Safety Analysis of Transformer Architectures

Architecture	Bias & Fairness	Robustness	Hallucination & Factual Grounding
Encoder-only	**Pros:** Excellent for discriminative fairness tasks (e.g., bias detection). **Cons:** Bidirectional context can deeply entrench and amplify statistical biases from training data. Prone to overfitting on spurious correlations.	**Pros:** Deep contextual understanding can provide resilience to some classification-based adversarial attacks. **Cons:** May be less robust than decoder-only models on some NLP benchmarks.	N/A: Not a generative architecture, so hallucination is not a primary failure mode. The core risk is misrepresentation, not fabrication.
Decoder-only	**Pros:** Can be fine-tuned to perform competitively on bias detection tasks. **Cons:** Autoregressive generation can lead to "bias cascades," where an initial biased token reinforces further biased generation.	**Pros:** Potentially more robust to certain textual perturbations due to generalized pre-training on vast, unstructured data. **Cons:** Highly vulnerable to generative attacks like prompt injection and jailbreaking.	**Pros:** Capable of generating fluent and coherent text. **Cons:** Highly susceptible to the "Attention Degeneration Problem," leading to source-detached hallucinations in long generations. Lacks a persistent architectural grounding mechanism.
Encoder-Decoder	**Pros:** Separation of encoder/decoder allows for potentially distinct bias mitigation strategies. **Cons:** Combines vulnerabilities; bias can be introduced in the encoding stage and amplified in the decoding stage.	**Pros:** Cross-attention to a full source representation may provide a stable anchor against certain perturbations. **Cons:** Complex attack surface, with vulnerabilities in both the encoder (misrepresentation) and decoder (malicious generation).	**Pros:** The encoder provides a strong, persistent architectural grounding mechanism. Cross-attention forces the decoder to continuously refer to the source, mitigating source-detached hallucinations. **Cons:** Still susceptible to more subtle, "intrinsic" hallucinations (fabricating details while remaining on-topic).

1.2 On the Feasibility of Meta-Analysis

A quantitative meta-analysis, which statistically aggregates results, would be the ideal method for comparing T&S properties. However, such an analysis is currently infeasible due to profound heterogeneity across studies. T&S evaluation is not standardized; papers employ a wide array of fairness benchmarks (e.g., BOLD, StereoSet; Dhamala et al. 2021; Nadeem et al. 2020), adversarial attack methodologies, and hallucination metrics. Therefore, this paper adopts the methodology of a qualitative systematic review, adhering to principles inspired by the PRISMA guidelines (Page et al., 2021).

2 The Transformer Architectural Spectrum

To understand the architectural roots of T&S issues, one must understand the core components of the transformer model and its primary architectural variants.

2.1 Core Components

- **Self-Attention Mechanism:** The central innovation of the transformer (Vaswani et al., 2023). For each token, it computes a weighted sum of all other tokens in the sequence. These "attention scores" allow the model to capture complex, long-range dependencies by projecting input embeddings into Query (Q), Key (K), and Value (V) matrices.

- **Positional Encoding:** Since self-attention is permutation-invariant, positional encodings are added to input embeddings to provide the model with information about the order of tokens in the sequence.

- **Encoder and Decoder Blocks:** Models are constructed by stacking blocks. An encoder block contains a multi-head self-attention layer and a feed-forward network. A decoder block adds a cross-attention layer to attend to the encoder's output, enabling it to condition its generation on the input sequence (IBM, 2024).

2.2 The Three Architectural Families

- **Encoder-only (Auto-Encoding):** Exemplified by BERT (Devlin et al., 2019), this family uses only the encoder stack. Pre-trained with objectives like Masked Language Modeling (MLM), its defining feature is **bidirectional context understanding**. This makes it highly effective for analysis tasks like text classification and sentiment analysis.

- **Decoder-only (Autoregressive):** This family, including the GPT series, uses only the decoder stack. Trained to predict the next token given all preceding tokens (Causal Language Modeling), its defining feature is its **unidirectional (left-to-right) autoregressive nature**. This makes it proficient at generative tasks like creative writing and chatbots.

- **Encoder-Decoder (Sequence-to-Sequence):** This family, including the original transformer and models like T5 (Raffel et al., 2020), employs both stacks. The encoder processes an input sequence, and the decoder autoregressively generates an output sequence while attending to the encoder's output. It is purpose-built for tasks like machine translation and summarization.

3 Systematic Review: Architectural Predispositions to T&S Risks

A central finding of this review is the duality of architectural strength and weakness: a model's greatest capability is often the direct source of its most significant T&S vulnerability. T&S failures are not accidental side effects but are emergent properties of the mechanisms that optimize these architectures for their primary tasks.

3.1 Bias and Fairness

Bias in LLMs can perpetuate and amplify harmful societal stereotypes. How bias manifests is intrinsically linked to the model's architecture.

- **Encoder-only models** leverage bidirectional context, making them adept at learning spurious statistical correlations from data (Feder et al., 2022). If a corpus frequently associates "nurse" with "she," the bidirectional attention can form a strong link, amplifying this bias in downstream classification tasks. This *intrinsic bias* is often measured by probing embedding geometry with tests like WEAT (Caton and Haas, 2024).

- **Decoder-only models** operate via an autoregressive process, which creates the potential for a **bias cascade** (Bender et al., 2021). If the model generates a biased token (e.g., "he" after "The brilliant doctor..."), that token becomes a fixed part of the context for all subsequent predictions, reinforcing the initial stereotype in a self-perpetuating feedback loop. This *extrinsic bias* is evaluated by analyzing generated content with benchmarks like StereoSet (Nadeem et al., 2020).

3.2 Robustness to Perturbations

A model's robustness—its ability to maintain performance with perturbed inputs—is crucial for reliability. The architectural paradigms present distinct attack surfaces. Attacks on encoders typically aim to induce misclassification, while attacks

on decoders often use prompt injection to bypass safety alignment (Li et al., 2020; Liu et al., 2024b). The literature presents a complex picture. While early work suggested transformers were more robust than CNNs, more rigorous studies challenged this, suggesting robustness is an emergent property of the entire training ecosystem (Bai et al., 2021). A noteworthy result from direct comparisons on GLUE benchmark tasks is that GPT-2 (decoder-only) was found to be more robust to textual perturbations than both BERT (encoder-only) and T5 (encoder-decoder) (Neerudu et al., 2023). A plausible hypothesis is that the generative pre-training on massive, unstructured corpora forces decoder-only models to learn a more generalized, "looser" representation of language that is less sensitive to minor syntactic variations.

3.3 Hallucination and Factual Grounding

Hallucination, the generation of plausible but factually incorrect information, is a fundamental challenge for generative models (Ji et al., 2023). It is a statistically inevitable byproduct of models trained to maximize next-token likelihood. A key mechanism driving a divergence in hallucination susceptibility is the **Attention Degeneration Problem**, a critical defect in decoder-only models on grounded generation tasks (Beltagy et al., 2020). In a decoder-only model, the prompt and generated text exist in the same sequence. As the output grows, the relative attention paid to the source prompt diminishes. The model becomes progressively more influenced by its own output, causing it to "drift away" from its grounding into detached hallucination. The **encoder-decoder architecture** offers a powerful, built-in structural defense. It enforces a separation of concerns: the encoder creates a comprehensive, static representation of the source. The decoder's cross-attention mechanism can then refer back to this stable representation at every step of generation (IBM, 2024). This constant re-grounding acts as an architectural guardrail, preventing source-detached hallucination and explaining why models like T5 can demonstrate superior performance on grounded tasks like summarization (Raffel et al., 2020).

4 The Mitigation Ecosystem

While architecture establishes inherent predispositions, a comprehensive safety strategy requires a defense-in-depth approach, as detailed in Table 2. This ecosystem, however, is largely a series of reactive patches applied across the AI lifecycle.

- **Data-Centric Approaches:**This foundational layer treats data not as a resource to be passively consumed but as a subject for rigorous governance, a perspective critical of uncurated, web-scale datasets (Bender et al., 2021). Such governance demands a proactive investigation into the corpus, beginning with a deep sociolinguistic analysis to scrutinize demographic and linguistic representation, thereby identifying whose voices are present, amplified, or absent. This analytical work is complemented by the creation of formal documentation, such as "Data Statements" or "Datasheets for Datasets," which transparently record the data's provenance, collection methodology, and inherent limitations to ensure accountability (Gebru et al., 2021). Finally, this process of deep analysis and documentation enables a more robust form of content auditing, moving beyond simple keyword filtering to a qualitative removal of not just explicit toxicity but also subtler forms of stereotypes and misinformation (Weidinger et al., 2021).

- **Model Training Techniques:** Alignment methods like Adversarial Training (Goodfellow et al., 2015), Reinforcement Learning from Human Feedback (RLHF; Ouyang et al. 2022), and Constitutional AI (CAI; Bai et al. 2022) steer model behavior toward desired outcomes. A key risk is *safety alignment erosion*, where fine-tuning on a new task can degrade the original alignment (Qi et al., 2023).

- **Post-Processing and Moderation:** Inference-time defenses like controlled decoding (Holtzman et al., 2020) and input/output filtering using "Supervisor LLMs" (Zheng et al., 2023) provide a final layer of defense.

- **Human-in-the-Loop (HITL) and Governance:** Integrating human oversight through real-time monitoring (Li et al., 2024) and proactive adversarial testing via Red Teaming (Microsoft, 2023) are essential for accountability.

Table 2: Systematic Measurement Methods Across the AI Lifecycle and Transformer Architectures. In the 'Tools & Benchmarks' rows, `monospaced` names refer to available Python packages.

Architecture	**Focus**	**1. Data & Pre-Training** (Measuring Foundational Bias)	**2. Training & Alignment** (Measuring Learned Behavior)	**3. Testing & Red Teaming** (Measuring Pre-Deployment Risk)	**4. Deployment & Monitoring** (Measuring Real-World Harm)
Encoder-only (e.g., BERT)	**Risk**	Latent statistical biases (e.g., gender, race) embedded from the corpus.	Entrenched biases that persist despite fine-tuning.	Unfair or biased classifications under adversarial conditions.	Downstream systems inheriting and amplifying model bias.
	Method	Quantify stereotypical associations by measuring the geometric distance between concept embeddings (e.g., 'doctor' vs. 'nurse') and attribute embeddings (e.g., 'he' vs. 'she').	Apply counterfactual inputs by changing only the sensitive attribute in a prompt and measuring the resulting change in the model's internal representations or final classification.	Systematically perturb inputs using character swaps, synonyms, or paraphrasing to test if the model's fairness predictions are brittle.	Perform statistical analysis on model outputs in production to measure if a particular group is being systematically disadvantaged.
	Tools & Benchmarks	WEAT, SEAT, `sent-fairness`	Fairness testing, `fairlearn`	WinoGender, WinoBias, `TextAttack`, TextFlint	Impact/Equality metrics, `fairlearn`, `AIF360`
Decoder-only (e.g., GPT)	**Risk**	Presence of toxic, hateful, or explicit language in the training data.	Generating stereotyped or prejudiced text in a self-reinforcing "bias cascade."	Bypassing safety filters through prompt injection or "jailbreaking."	Generating plausible-sounding but false information (hallucinations).
	Method	Scan representative samples of the training corpus using pre-trained toxicity classifiers to score the prevalence of harmful content that the model may learn to replicate.	Evaluate model completions for prompts designed to elicit stereotypes, comparing the model's associations against known societal biases.	Use automated and human-led adversarial testing to discover prompts that trick the model into violating its safety policies.	Compare generated outputs against a trusted knowledge base or use human reviewers to score the factual accuracy of responses.
	Tools & Benchmarks	Perspective API, `detoxify`	StereoSet, CrowS-Pairs, `evaluate`	HarmBench, AdvBench, `garak`, `llm-guard`	TruthfulQA, RAG frameworks
Encoder-Decoder (e.g., T5)	**Risk**	Skewed representations in parallel corpora (e.g., gender bias in translation pairs).	Exaggerating biases present in the source text during generation tasks.	Fabricating details not present in the source document (grounding failure).	Compounding errors where a small encoder error leads to a large decoder error.
	Method	Conduct statistical analysis on source-target data pairs to identify and quantify systemic biases, such as translating gender-neutral pronouns into gendered ones.	Audit summarization or translation outputs for faithfulness. Measure if the generated text maintains the neutrality of the source or introduces new biases.	Measure n-gram overlap between the generated text and the source document to check for source-detached hallucinations. More advanced methods use information extraction to verify facts.	Implement end-to-end task evaluation with human reviewers who assess the final output based on multiple criteria (accuracy, fairness, fluency, faithfulness) simultaneously.
	Tools & Benchmarks	Manual audits, `spaCy`, `NLTK`	BE-Gender-Fair, Human review	ROUGE, BLEU, METEOR, `evaluate`	Human evaluation platforms

5 The Frontier: Novel Architectural Paradigms

A more fundamental approach involves rethinking the architecture itself to build models that are inherently more trustworthy. While these proposed paradigms are not new to the broader field of LLM, their systematic application as a foundational approach to solving Trust and Safety challenges at the architectural level represents a significant frontier.

5.1 Causal Transformers

Current transformers are correlational engines, vulnerable to learning spurious correlations (Feder et al., 2022). Causal Transformers aim to address this by integrating principles from causal inference into the architecture (Oba et al., 2023). This could involve architectures for explicit counterfactual reasoning ("what if" scenarios) or dual-system designs that combine a fast, intuitive "System 1" (a standard transformer) with a slower, deliberative "System 2" that performs explicit causal reasoning (Bengio, 2019).

5.2 Formally Verified Architectures

While empirical methods like Red Teaming are crucial, they are fundamentally unable to provide exhaustive guarantees against catastrophic failures. Standard testing may fail to uncover latent threats that are not apparent during development, such as **deceptive alignment**, where a model behaves as intended during training but pursues unintended goals upon deployment; unpredictable **emergent capabilities** that manifest suddenly at scale; instrumental **power-seeking behavior** as a means to achieve other goals; or failures triggered by rare but high-impact **long-tail events**.

The potential for such severe, unforeseen risks underscores the limitations of purely empirical safeguards and motivates the need for a guaranteed safe AI framework. Although the scale of LLMs makes it intractable to provide absolute mathematical guarantees about their entire behavior, a more viable path is to apply formal methods to smaller, critical components to create provably safe modules ("davidad" Dalrymple et al., 2024). Techniques like Satisfiability Modulo Theories (SMT) solvers (e.g., Reluplex) can verify specific input-output properties for smaller components like a toxicity classifier (Katz et al., 2017). Abstract Interpretation reasons about a program's behavior over infinite inputs by computing a sound over-approximation of its reachable states, trading precision for a formal guarantee that an output will never enter an unsafe region (Cousot and Cousot, 1977).

5.3 Biologically-Inspired and Hybrid Models

Neuroscience offers a rich source of ideas for more trustworthy architectures.

- **Neuro-Symbolic AI:** This paradigm combines connectionist (neural) and symbolic AI (Sarker et al., 2021). A hybrid system could use a neural front-end for perception and a symbolic back-end for reasoning, making the model's reasoning more transparent and auditable.

- **Hierarchical Memory Systems:** Unlike the "flat" memory of a transformer, the human brain uses a complex hierarchical memory system. Architectures like the Hierarchical Memory Transformer (HMT) attempt to mimic this to improve long-context understanding and mitigate attention degeneration (Hutchins et al., 2022).

A summary of these novel approaches is presented in Table 3.

6 Technical Proposals for Testing

To bridge the gap between conceptual paradigms and empirical validation, this section provides overview on technical frameworks to address the problems previously identified. The methods presented here are not entirely novel; rather, they are based on state-of-the-art technologies that are currently under active research in both academia and industry. It is for the purpose of scholarly validation and do not represent a blueprint for any specific production system. Drawing upon these techniques, the aim is to select and combine the solutions deemed most suitable for the specific challenges addressed in this study and to demonstrate how the proposed methods can work in a complementary manner to enhance the stability and reliability of the overall system.

6.1 Testing Causal Interventions for Bias Mitigation

A core claim of Causal Transformers is their ability to distinguish correlation from causation. This can be tested by measuring the causal effect of a sensitive attribute on a model's output.

Table 3: Summary of Novel Architectural Approaches for LLM Safety

Approach	Core Principle & T&S Problem Addressed
Causal Transformers	Reasoning with interventions and counterfactuals instead of just correlations. *Addresses: Spurious correlations, OOD robustness, fairness.*
Formally Verified	Using mathematical proof (e.g., SMT solvers, abstract interpretation) to guarantee safety properties for critical components. *Addresses: High-assurance guarantees; preventing catastrophic failures.*
Neuro-Symbolic & Hierarchical Memory	Combining neural perception with symbolic reasoning. Mimicking biological memory systems. *Addresses: Explainability, brittle reasoning, long-context understanding, hallucination.*

- **Objective:** To quantitatively measure if a model's output is causally dependent on a sensitive attribute (e.g., gender). A causally fair model should exhibit minimal change in its output representation when only the sensitive attribute is altered.

- **Mathematical Formulation:** Using the concept of the calssic statistics on Average Treatment Effect (ATE) from causal inference, let P be a prompt with a sensitive attribute A (e.g., pronouns "he" vs. "she"). Let $E = M(P)$ be the output embedding from model M. We estimate the causal effect of changing A from a_2 to a_1 by performing a causal intervention, denoted by the *do*-operator:

  $$ATE = \mathbb{E}[E|do(A = a_1)] - \mathbb{E}[E|do(A = a_2)]$$

  In practice, a large norm, $||ATE||$, indicates a strong causal bias. This is outlined in Algorithm 1.

6.2 Verifying Safety with Abstract Interpretation

Abstract interpretation can formally verify properties over entire sets of inputs rather than individual points.

- **Objective:** To formally verify that for any input within a defined abstract domain (e.g., any sentence with a certain sentiment score), the output logit for a specific "unsafe" token will never exceed a safety threshold.

- **Mathematical Formulation:** Abstract interpretation relies on an abstraction function α and a concretization function γ. Let $L(x)$ be the transformation by a neural network layer. An abstract transformer $L^{\#}(a)$ operates on an abstract input a. The transformer is sound if it always over-approximates the concrete layer's behavior:

  $$\forall a \in \text{AbstractDomain: } \alpha(L(\gamma(a))) \sqsubseteq L^{\#}(a)$$

  Here, $\sqsubseteq$ is the partial order relation in the abstract domain (e.g., for intervals, $[l_1, u_1] \sqsubseteq [l_2, u_2]$ if $l_2 \leq l_1$ and $u_1 \leq u_2$). This soundness ensures that if the abstract transformer proves a property, it holds for all possible concrete inputs. The procedure is outlined in Algorithm 2.

7 Conclusion

This systematic review has dissected the intersection of transformer architecture and LLM Trust and Safety, concluding that a model's foundational design is intrinsically linked to its propensity for critical failure modes. A key strength of this work is its structured, dual-pronged analysis. It moves beyond a monolithic view of LLMs to first systematically deconstruct the inherent T&S trade-offs "baked in" to the three primary architectural families: Encoder-only, Decoder-only, and Encoder-Decoder. Second, it maps the corresponding ecosystem of mitigation layers, from data-centric approaches to post-deployment monitoring, providing a holistic view of how safety is currently engineered. Crucially, this paper does not merely catalogue existing problems; it also charts a clear path forward. By proposing and providing technical frameworks for novel paradigms, it advocates for a fundamental shift away from reactive safety patches and toward proactive, trustworthy-by-design principles.

Algorithm 1: Counterfactual Fairness Test

```
Input: model, professions, prompt_template
Output: Average bias score
bias_scores ← [];
pronouns ← ["he", "she"];
for prof in professions do
    prompt1 ← Format(template, prof, pronouns[0]);
    prompt2 ← Format(template, prof, pronouns[1]);
    output1 ← model.Generate(prompt1);
    embedding1 ← model.GetEmbedding(output1);
    output2 ← model.Generate(prompt2);
    embedding2 ← model.GetEmbedding(output2);
    score ← EuclideanDistance(embedding1, embedding2);
    bias_scores.append(score);
end
return Average(bias_scores);
```

Algorithm 2: Safety Verification with Abstract Interpretation

```
procedure VERIFY_LAYER_SAFETY(layer, domain, unsafe_idx, threshold)
    abstract_output ← COMPUTE_TRANSFORMER(layer, domain);
    unsafe_upper_bound ← abstract_output.upper[unsafe_idx];
    if unsafe_upper_bound > threshold then
        Print("Verification FAILED");
    end
    else
        Print("Verification PASSED");
    end
function COMPUTE_TRANSFORMER(layer, input_interval)
    W_pos ← Max(0, layer.weights);
    W_neg ← Min(0, layer.weights);
    out_lower ← Dot(W_pos, input_interval.lower) + Dot(W_neg, input_interval.upper) + layer.bias;
    out_upper ← Dot(W_pos, input_interval.upper) + Dot(W_neg, input_interval.lower) + layer.bias;
    return Interval (lower=out_lower, upper=out_upper);
```

This transition from treating safety as an add-on to embedding it within the architectural fabric is presented as the more viable frontier for the next generation of safe and trustworthy LLM. Despite these contributions, there are areas that warrant further investigation. The primary limitation, as noted within the analysis, is the current infeasibility of a quantitative meta-analysis due to the lack of standardized T&S benchmarks. The development of such standards is a prerequisite for enabling rigorous, reproducible comparisons between architectures. Furthermore, while this paper offers technical proposals for testing the novel architectural solutions, large-scale empirical validation remains a critical next step. Future work must focus on implementing these proposed designs to measure their real-world efficacy and performance trade-offs against current models. In summary, the future of LLM safety will not be secured by simply applying more layers of mitigation to inherently flawed foundations. It demands a innovative rethinking of the architectural blueprints themselves, a process that will embed safety not as a feature, but as the primary, non-negotiable objective from the outset.

Limitations

While this systematic review provides a structured analysis of architectural T&S predispositions, its scope and conclusions are subject to several limita-

tions that suggest avenues for future work.

First, the analysis is predominantly centered on the English language context. The foundational models discussed, such as BERT and the GPT series, were primarily trained on English-centric corpora, and the T&S benchmarks cited (e.g., WEAT, StereoSet, BOLD) were developed to measure biases as they manifest in English. Consequently, the specific T&S trade-off profiles identified may not generalize directly to languages with different linguistic structures, such as those with rich morphology, or to different socio-cultural contexts where harmful stereotypes and failure modes may manifest differently.

Second, the proposed novel architectural paradigms, while promising, come with their own significant scalability and resource challenges. Formally Verified Architectures, for instance, are acknowledged to be intractable for guaranteeing the behavior of entire large-scale models; their application is more feasible for smaller, critical components. Similarly, proposals like Causal Transformers or Biologically-Inspired hybrid models may introduce substantial computational overhead compared to standard transformer architectures, potentially requiring large GPU resources and posing challenges to their deployment at the scale of current foundation models.

Furthermore, it is crucial to clarify the scope of this review's recommendations. The metrics and tools presented are not intended as an exhaustive or prescriptive list. Rather, they serve as representative examples within this systematic review to illustrate ideal pairings of evaluation methods for the safety issues inherent to each transformer type. Similarly, while this paper advocates for a "trustworthy-by-design" approach as a promising frontier, it does not endorse this paradigm as the sole or exclusive solution to LLM safety, nor does it suggest other methods are not in use. The primary aim is to systematically map the current landscape, pairing inherent architectural risks with corresponding measurement strategies to inform a more holistic and architecturally-aware approach to safety engineering.

Finally, the core methodology of this study is a qualitative systematic review, a choice necessitated by the profound heterogeneity and lack of standardization across T&S evaluation metrics in existing literature. This prevents a more rigorous quantitative meta-analysis. This limitation highlights a critical need within the research community to develop standardized benchmarks for enabling reproducible comparisons between architectures. The technical proposals outlined in this paper remain conceptual frameworks; they require large-scale empirical validation to measure their real-world efficacy and performance trade-offs against current models.

Ethical Considerations and Potential Risks

While this review aims to improve LLM safety, we acknowledge two potential risks. First, our systematic analysis of architectural vulnerabilities could inadvertently serve as a guide for malicious actors seeking to design more effective attacks against existing systems. Second, the nuanced trade-offs discussed could be oversimplified by readers, potentially leading to the premature dismissal of entire architectural families rather than fostering more robust safety engineering within each paradigm. This work is intended solely for defensive and proactive research, and we have consciously avoided introducing novel, exploitable attack techniques.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional ai: Harmlessness from ai feedback. *Preprint*, arXiv:2212.08073.

Yutong Bai, Jieru Mei, Alan Yuille, and Cihang Xie. 2021. Are transformers more robust than cnns?

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Yoshua Bengio. 2019. The consciousness prior. *Preprint*, arXiv:1709.08568.

Simon Caton and Christian Haas. 2024. Fairness in machine learning: A survey. *ACM Computing Surveys*, 56(7):1–38.

Jaymari Chua, Yun Li, Shiyi Yang, Chen Wang, and Lina Yao. 2024. Ai safety in generative ai large language models: A survey. *Preprint*, arXiv:2407.18369.

Patrick Cousot and Radhia Cousot. 1977. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 238–252.

David "davidad" Dalrymple, Joar Skalse, Yoshua Bengio, Stuart Russell, Max Tegmark, Sanjit Seshia, Steve Omohundro, Christian Szegedy, Ben Goldhaber, Nora Ammann, Alessandro Abate, Joe Halpern, Clark Barrett, Ding Zhao, Tan Zhi-Xuan, Jeannette Wing, and Joshua Tenenbaum. 2024. Towards guaranteed safe ai: A framework for ensuring robust and reliable ai systems. *Preprint*, arXiv:2405.06624.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, and Rahul Gupta. 2021. Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 862–872. ACM.

Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, and Diyi Yang. 2022. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *Preprint*, arXiv:2109.00725.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for datasets. *Preprint*, arXiv:1803.09010.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *Preprint*, arXiv:1412.6572.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration.

DeLesley Hutchins, Imanol Schlag, Yuhuai Wu, Ethan Dyer, and Behnam Neyshabur. 2022. Block-recurrent transformers. *Preprint*, arXiv:2203.07852.

IBM. 2024. What is an encoder-decoder model? https://www.ibm.com/think/topics/encoder-decoder-model.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Guy Katz, Clark Barrett, David Dill, Kyle Julian, and Mykel Kochenderfer. 2017. Reluplex: An efficient smt solver for verifying deep neural networks.

Dongyuan Li, Zhen Wang, Yankai Chen, Renhe Jiang, Weiping Ding, and Manabu Okumura. 2024. A survey on deep active learning: Recent advances and new frontiers. *Preprint*, arXiv:2405.00334.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. *Preprint*, arXiv:2004.09984.

Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo, Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2024a. Trustworthy llms: a survey and guideline for evaluating large language models' alignment. *Preprint*, arXiv:2308.05374.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. 2024b. Prompt injection attack against llm-integrated applications. *Preprint*, arXiv:2306.05499.

Microsoft. 2023. Red teaming large language models (llms). https://learn.microsoft.com/en-us/azure/ai-foundry/openai/concepts/red-teaming.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. Stereoset: Measuring stereotypical bias in pretrained language models. *Preprint*, arXiv:2004.09456.

Pavan Kalyan Reddy Neerudu, Subba Reddy Oota, Mounika Marreddy, Venkateswara Rao Kagita, and Manish Gupta. 2023. On robustness of fine-tuned transformer-based nlp models. *Preprint*, arXiv:2305.14453.

Daisuke Oba, Naoki Yoshinaga, and Masashi Toyoda. 2023. Perplm: Personalized fine-tuning of pretrained language models via writer-specific intermediate learning and prompts. *Preprint*, arXiv:2309.07727.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Matthew J Page, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, and 1 others. 2021. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *bmj*, 372.

Fábio Perez and Ian Ribeiro. 2022. Ignore previous prompt: Attack techniques for language models. *Preprint*, arXiv:2211.09527.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2023. Fine-tuning aligned language models compromises safety, even when users do not intend to! *Preprint*, arXiv:2310.03693.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Md Kamruzzaman Sarker, Lu Zhou, Aaron Eberhart, and Pascal Hitzler. 2021. Neuro-symbolic artificial intelligence: Current trends. *Preprint*, arXiv:2105.05330.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, and 4 others. 2021. Ethical and social risks of harm from language models.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.