Towards Intrinsic Interpretability of Large Language Models: A Survey of Design Principles and Architectures

Yutong Gao[1,4,*], **Qinglin Meng**[5,*], **Yuan Zhou**[5], **Liangming Pan**[1,2,3,†]

[1]MOE Key Laboratory of Computational Linguistics, Peking University
[2]School of Computer Science, Peking University
[3]Beijing Academy of Artificial Intelligence, Beijing, China
[4]Nanjing University of Science and Technology, [5]Purdue University
yutongkkk@njust.edu.cn, {meng160, zhou1475}@purdue.edu
liangmingpan@pku.edu.cn

Abstract

While Large Language Models (LLMs) have achieved strong performance across many NLP tasks, their opaque internal mechanisms hinder trustworthiness and safe deployment. Existing surveys in explainable AI largely focus on post-hoc explanation methods that interpret trained models through external approximations. In contrast, intrinsic interpretability, which builds transparency directly into model architectures and computations, has recently emerged as a promising alternative. This paper presents a systematic review of the recent advances in intrinsic interpretability for LLMs, categorizing existing approaches into five design paradigms: functional transparency, concept alignment, representational decomposability, explicit modularization, and latent sparsity induction. We further discuss open challenges and outline future research directions in this emerging field. The paper list is available at: Survey-Intrinsic-Interpretability-of-LLMs

1 Introduction

Large Language Models have achieved remarkable success across diverse tasks (Brown et al., 2020; Raffel et al., 2020; Chowdhery et al., 2022; Team et al., 2025). However, their complexity often makes them "black boxes" (Bommasani et al., 2022), hiding their internal decision-making. This lack of transparency creates trust and safety risks, especially in high-stakes fields like healthcare and law (Rudin, 2019; Pawar et al., 2020).

To address these concerns, interpretability research is often divided into two paradigms: post-hoc explanation and intrinsic design. Post-hoc methods analyze trained, fixed models using external tools such as LIME, SHAP, sparse autoencoders, or causal interventions (Ribeiro et al., 2016; Lundberg and Lee, 2017; Huben et al., 2024; Meng et al., 2022). Many rely on surrogate models or statistical attributions, resulting in a well-known fidelity gap between the explanation and the model's true computation (Jacovi and Goldberg, 2020). Causal based post hoc methods partially address this issue by intervening directly on internal components, yielding stronger local faithfulness (Meng et al., 2022; Wang et al., 2023). However, their explanations remain highly fine grained and are difficult to aggregate into coherent, high level accounts of overall model behavior.

In contrast, intrinsic interpretability builds transparency directly into the model architecture and training process (Fedus et al., 2022; Gao et al., 2025). By ensuring that the model's internal computation is itself interpretable, these approaches aim to achieve *structural fidelity*, namely a direct correspondence between model behavior and its explanation, without relying on external surrogates or post-hoc aggregation. Historically, however, intrinsic methods were constrained by a severe tradeoff: models that were transparent by construction typically lacked the expressive power required for complex language tasks (Linardatos et al., 2021).

Recent advances demonstrate that interpretability and performance need not be mutually exclusive, showing that large-scale models can be designed with interpretable internal structure while retaining competitive task performance (Rudin, 2019; Sharkey et al., 2025). By incorporating inductive biases such as modularity, sparsity, disentanglement, and structured representations directly into modern architectures and training objectives (Shazeer et al., 2017; Louizos et al., 2018; Fedus et al., 2022; Gao et al., 2025), these methods enable interpretability to emerge as a property of the model itself rather than as an after-the-fact analysis.

Despite this rapid progress, the literature on intrinsic interpretability remains fragmented, spanning disparate model classes, architectural choices,

* Equal contribution.
† Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 34741–34754

and training principles. Unlike post-hoc explanation methods whose taxonomy and limitations have been extensively surveyed (Molnar, 2025; Madsen et al., 2022; Zhao et al., 2024a; Palikhe et al., 2025), there remains a need for a unified framework that organizes intrinsic approaches around shared design principles or clarifies how different mechanisms contribute to transparency in LLMs. This survey aims to fill this gap by systematically reviewing intrinsic interpretability methods for LLMs, distilling common design principles, and highlighting open challenges and promising future directions.

Our contributions are threefold. First, we distinguish post-hoc explanation from intrinsic interpretability, clarifying their differences in faithfulness, scope, and design philosophy. Second, we introduce a structured taxonomy of intrinsic interpretability methods organized around five core design principles: *Functional Transparency*, *Concept Alignment*, *Representational Decomposability*, *Explicit Modularity*, and *Latent Sparsity Induction*. Finally, we synthesize existing work within this framework, analyze methodological strengths and limitations, and identify key open challenges and future research directions.

2 Two Paradigms of LLM Interpretability

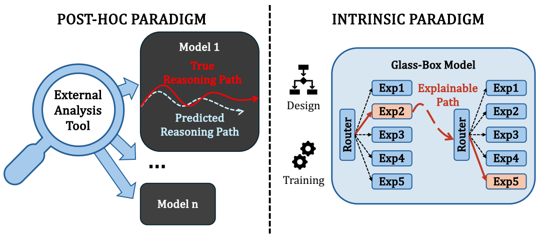

Figure 1: Comparison of the Post-hoc analysis versus Intrinsic design in LLM interpretability.

Research on interpretability for modern neural models has largely converged around two paradigms: (1) *post-hoc analysis*, which applies external tools to a trained, fixed model, and (2) *intrinsic interpretability*, which incorporates transparency directly into the model's architecture and training process. We distinguish these paradigms by causal necessity: an interpretability method is intrinsic if its interpretable components (e.g., sparse experts or concepts) lie on the critical computation path, such that modifying them directly alters the model's output. While recent hybrid approaches (Havasi et al., 2022; Tack et al., 2025) blur this line by allowing information to flow through residual side channels to preserve performance, we classify them as intrinsic designs that trade partial structural fidelity for enhanced capability. Figure 1 summarizes the key conceptual differences between these two approaches.

2.1 Post-hoc Interpretability

Post-hoc analysis has long dominated interpretability research as the default approach for explaining complex neural models. Existing surveys extensively cover these methods, ranging from early feature attribution techniques to modern mechanistic and causal analyses (Madsen et al., 2022; Zhao et al., 2024a; Ji et al., 2025; Palikhe et al., 2025).

Most post-hoc methods operate at one of two levels. At the *behavioral level*, feature attribution techniques such as LIME and SHAP (Ribeiro et al., 2016; Lundberg and Lee, 2017) estimate input importance by perturbing inputs and observing output changes, treating the model largely as a black box. At the *internal level*, inspection methods analyze intermediate representations. Probing classifiers train external predictors to detect concepts in hidden states (Raffel et al., 2020), while LogitLens projects hidden representations into the vocabulary space to expose transient computations (nostalgebraist, 2020). More recently, SAEs have emerged as a mechanistic tool for decomposing polysemantic activations into sparse, interpretable features (Huben et al., 2024).

Despite their flexibility, post-hoc methods share a fundamental limitation: they rely on auxiliary approximations rather than the model's native computation. (Jacovi and Goldberg, 2020) Attribution methods depend on local surrogate models, probing approaches identify correlations without establishing causal use (Ravichander et al., 2021), and mechanistic tools such as SAEs introduce reconstruction error by approximating, rather than exactly reproducing, forward-pass activations.

Causal-based post-hoc methods partially mitigate these issues by intervening on internal components and measuring their effects on model outputs (Meng et al., 2022; Wang et al., 2023). While such interventions provide stronger local faithfulness, their fine-grained nature makes it difficult to aggregate localized causal effects into coherent, high-level explanations of overall model behavior.

2.2 Intrinsic Interpretability

Intrinsic interpretability addresses the fidelity gap by designing models whose internal computation is transparent by construction. Rather than analyzing a trained black-box model, intrinsic approaches aim to build models in which the explanation is inseparable from the computation itself. As a result, interpretability is achieved without relying on post-hoc approximations.

Historically, intrinsic interpretability was largely confined to simple and low-dimensional models, such as linear regressors or generalized additive models, whose transparency comes at the cost of limited expressive power (Nelder and Wedderburn, 1972; Hastie and Tibshirani, 1986; Linardatos et al., 2021). While effective for certain tasks, these models were insufficient for complex NLP tasks. However, recent progress in sparse modeling, modular architectures, and structured representations suggests that transparency and scalability need not be mutually exclusive, enabling intrinsically interpretable designs that retain competitive performance at scale (Fedus et al., 2022; Gao et al., 2025; Tamkin et al., 2024). The following sections present the core design principles in Section 3 and representative methods in Section 4 underlying this line of work.

3 Design Principles of Intrinsic Interpretability

As illustrated in Figure 2, we categorize intrinsic interpretability into five design principles. These design philosophies dictate *how* transparency is constructed within a model. In this section, we analyze the rationale, formulation, and trade-offs of each principle, connecting them to the specific methodologies detailed in Section 4.

Functional Transparency. This principle advocates architectures whose computations are both structurally explicit and semantically meaningful. Rather than relying on opaque compositions of dense layers, such models are organized so that both the *where* (through structured or decomposed components) and the *what* (through operations with clear mathematical semantics) of computation are directly inspectable. As a result, these models behave less like black boxes and more like readable algorithms. Representative implementations are discussed in Section 4.1. Key trade-offs of this approach include reduced expressivity and training efficiency.

Concept Alignment. While functional transparency emphasizes mathematical structure, concept alignment targets semantic interpretability. This principle encourages latent variables to correspond directly to human-understandable concepts , thereby reducing *polysemanticity*, where individual units encode multiple unrelated features. By aligning representations with explicit concepts, models become easier to interpret and reason about. The primary trade-off is an *alignment tax*: constraining representations to be human-interpretable may limit expressive capacity or require additional supervision. Representative approaches following this principle are discussed in Section 4.2.

Representational Decomposability. Extending alignment, this principle focuses on the geometry of the latent space. It seeks to disentangle representations into independent subspaces so that distinct factors of variation can be manipulated separately without interference. This separation enables more precise and controllable generation. The central challenge is enforcing such decomposability, for example through orthogonality constraints, without relying on extensive supervision or sacrificing flexibility. Recent architectures that instantiate this principle are reviewed in Section 4.3.

Explicit Modularization. Whereas traditional models operate as a single monolithic block, this principle advocates decomposing computation into distinct, independently functioning modules. A routing mechanism explicitly selects which modules process a given input, yielding a clear and traceable computational pathway. A prominent instantiation of this principle is the Mixture-of-Experts (MoE) architecture (Section 4.4), which introduces transparency by structuring the model around specialized functional units. A key trade-off of this approach is the added complexity of routing and coordination, which can complicate optimization and limit global expressivity.

Latent Sparsity Induction. Rather than imposing a hand-crafted modular structure, this principle aims to induce modularity within otherwise standard neural architectures. The core insight is that the opacity of dense networks often stems from uniformly active and highly entangled pathways. Selective activation can be encouraged through sparsity-inducing training objectives, such as L_0 or structured regularization, or through competitive gating mechanisms such as Gated Linear Units (GLUs), which conditionally route information. These mechanisms encourage the model to sup-

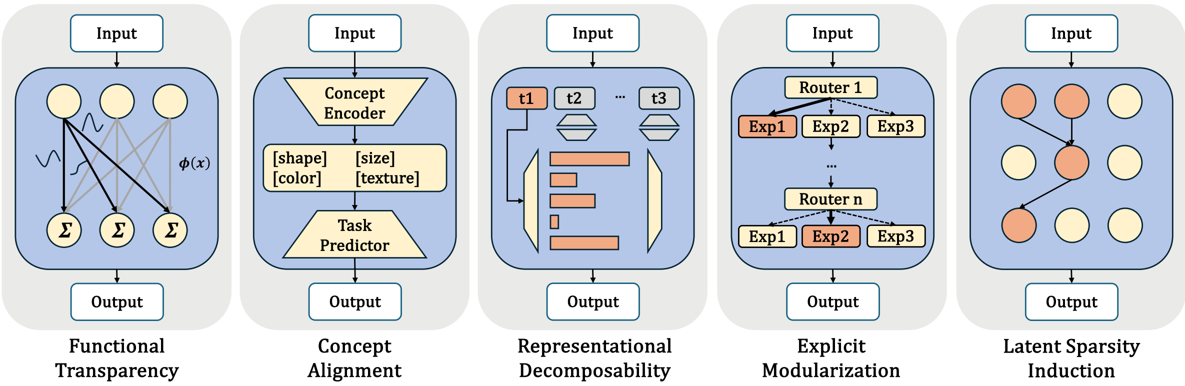

Figure 2: **A taxonomy of intrinsic architectural designs for interpretable LLMs.** We categorize existing approaches into five primary families based on their core mechanism for transparency.

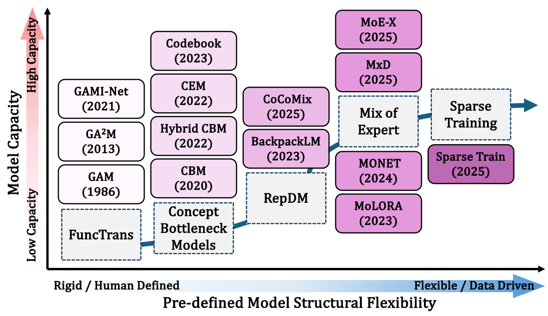

Figure 3: **Evolution of intrinsic interpretability.** The field has shifted from rigid, human-defined structures (e.g., GAMs) to scalable, data-driven sparse architectures (e.g., Specialized MoEs) that balance interpretability with performance.

press redundant channels and form task-specific subcircuits. Representative techniques following this principle are discussed in Section 4.5. A key trade-off is that strong sparsity or gating constraints can complicate optimization and may reduce expressivity or robustness if not carefully tuned.

We emphasize that these five paradigms serve as organizing design principles rather than strictly disjoint categories. Since some methods may instantiate multiple principles, we classify them according to the mechanism that most directly embeds interpretability into the model's architecture, training objective, or primary computation path, while noting cross-category connections where appropriate.

4 Intrinsic Interpretability Methods

In this section, we organize existing intrinsic interpretability methods according to the design principles introduced in Section 3. Figure 3 provides an overview of the methods discussed in this section, situating them along two dimensions: structural flexibility and model capacity.

4.1 Functional Transparency

In this subsection, we introduce three representative model families that realize functional transparency through architectural design, progressing from simple to more complex structures.

Generalized Additive Models. GAMs were originally proposed by Hastie and Tibshirani (1986) as an extension of GLMs (Nelder and Wedderburn, 1972). Instead of modeling the response as a linear combination of input features, GAMs replace the linear predictor with a sum of smooth univariate functions, yielding the formulation

$$F(\mathbf{x}) = f_0 + \sum_i f_i(x_i)$$

where each f_i is a learned smooth function of a single feature. These functions are typically estimated using iterative backfitting or local scoring procedures, which preserve interpretability by keeping each feature's contribution explicit and separable.

To capture limited feature interactions while retaining interpretability, Lou et al. (2013) introduced GA^2M, defined as

$$F(\mathbf{x}) = f_0 + \sum_i f_i(x_i) + \sum_{i,j} f_{ij}(x_i, x_j)$$

While effective, modeling pairwise interactions significantly increases computational and statistical complexity. This challenge was later addressed by EBMs (Nori et al., 2019), which use modern boosting techniques to efficiently learn additive and low-order interaction terms.

Neural Additive Models. More recently, researchers have leveraged neural networks to replace the smooth functions in GAMs, increasing expressivity while preserving the additive structure. Representative examples include GAMI-Net (Yang et al., 2021), as well as NODE-GAM and NODE-GA^{2}M models (Chang et al., 2022). In these approaches, each feature (or feature pair) is modeled by a small neural subnetwork, enabling nonlinear function approximation while maintaining per-feature transparency and interpretability.

Self-Explaining Neural Networks. Beyond additive models, Alvarez Melis and Jaakkola (2018) proposed Self-Explaining Neural Networks (SENN), which build predictions from an explicit combination of interpretable basis concepts and corresponding relevance scores. Concretely, SENN generalizes linear models by allowing both the concepts and their coefficients to be learned functions of the input, while imposing stability and interpretability constraints so that the contribution of each concept to the final prediction remains transparent.

B-cos Networks Another line of work modifies the predictive transformation itself to better align model parameters with task-relevant evidence. Böhle et al. (2022) proposed B-cos Networks, which replace standard linear transformations with B-cos transforms that encourage weight–input alignment during training. A B-cos transformer is defined as

$$\text{B-cos}(x; w) = \|\hat{w}\|\|x\||c(x, \hat{w})|^B \times \text{sgn}(c(x, \hat{w}))$$

where $c(x, \hat{w}) = \cos(\angle(x, w))$ As a result, the overall computation yields linear explanations that more faithfully reflect the evidence used for prediction. More recently, this idea has been extended to language models through B-cos LMs (Wang et al., 2025), which adapt pretrained language models with B-cos style transformations and fine-tuning to improve explanation faithfulness in NLP settings.

Kolmogorov–Arnold Networks. A more radical departure from the standard perceptron architecture is the Kolmogorov–Arnold Network (KAN) (Liu et al., 2025). While traditional multilayer perceptrons place learnable weights on edges and fixed activation functions on nodes, KANs invert this design by assigning learnable univariate functions, often parameterized as splines, to the edges. Based on the Kolmogorov–Arnold representation theorem, a KAN represents a multivariate function as

$$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$$

This formulation offers a high degree of functional transparency, as each $\phi_{q,p}$ can be directly visualized as a one-dimensional curve. As a result, KANs are relatively *symbolic-friendly*: in some cases, trained networks can be pruned and further simplified via symbolic regression into concise mathematical expressions. However, Hou et al. (2025) showed that KANs often suffer from significant computational overhead, optimization instability and inferior performance compared to standard MLPs when model size or input dimensionality grows.

4.2 Concept Alignment

Concept alignment is primarily realized through CBMs, which enforce interpretability by structurally constraining information flow within the network. Unlike post-hoc probes that analyze fixed representations, intrinsic CBMs explicitly design the architecture as a composition of a concept encoder $g : \mathcal{X} \to \mathcal{C}$ and a predictor $f : \mathcal{C} \to \mathcal{Y}$

Standard CBMs (Hard Bottlenecks). First formalized by Koh et al. (2020), standard CBMs impose a strict bottleneck where the final prediction relies exclusively on the predicted concepts $\hat{c} = g(x)$. This can be achieved via *independent training* (training g and f sequentially) or *joint training*. Vandenhirtz et al. (2024) proposed SCBMs to relax the assumption that concepts are conditionally independent by learning a joint distribution over concept rather than predicting each concept separately. While this architecture guarantees that the reasoning process is grounded in the defined concepts, it often suffers from an accuracy-interpretability trade-off, as the bottleneck may discard task-relevant information not captured by the predefined concept set.

Hybrid CBMs. To mitigate the performance degradation of hard bottlenecks, Mahinpei et al. (2021) and Havasi et al. (2022) proposed Hybrid CBMs. These models introduce a side channel, allowing the predictor to access both the explicit concepts c and uncontrolled latent embeddings z (i.e., $y = f(c, z)$). CB-LLM (Sun et al., 2025) extends this hybrid paradigm to LLMs, which introduces an unsupervised latent pathway alongside the concept bottleneck and employs adversarial training to remove concept-related information from the latent channel. Interpretability is maintained by applying regularization during training to maximize the model's reliance on concept while using z without encoding concept-related information.

Concept Embedding Models (CEMs). In NLP tasks, compressing a concept to a single scalar activation limits expressivity. To address this issue, CEMs and IntCEMs (Zarlenga et al., 2022) represent each concept as a high-dimensional vector in a learnable subspace rather than a scalar. This design allows the model to capture nuances (e.g., polysemy) while strictly restricting the downstream predictor to linear interactions between these concept embeddings, preserving the distinct attribution of the bottleneck design.

Unsupervised Discrete Bottlenecks. A limitation of the preceding approaches is their reliance on predefined concept annotations. To address this, Tamkin et al. (2024) proposed Codebook Features, which introduce an intrinsic bottleneck in a fully unsupervised manner. The method applies vector quantization (Gray, 1984; van den Oord et al., 2017) to approximate continuous hidden states using sparse combinations of vectors from a learned codebook, trained by jointly optimizing the language modeling objective and a reconstruction loss. By restricting representations to a discrete vocabulary, the approach promotes the emergence of distinct, often human-interpretable features without manual annotation. However, empirical results are reported on relatively small language models and a limited set of tasks, leaving its behavior at larger scales an open question.

4.3 Representational Decomposability Models

This class of methods operationalizes this design principle by explicitly structuring the model's latent space. Unlike standard Transformer architectures, where information is distributed across a single dense hidden representation, these approaches impose geometric constraints that separate distinct semantic factors into orthogonal subspaces or parallel processing streams.

Backpack Language Models. Standard Transformers entangle contextual information and lexical identity within a unified hidden state, making it difficult to isolate the contribution of individual word senses. To address this limitation, Hewitt et al. (2023) propose the Backpack Language Model (BLMs), which decomposes prediction into interpretable components. In this architecture, each vocabulary item is associated with a set of learnable, non-contextual *sense vectors*, capturing different meanings of the same surface form. The self-attention mechanism is constrained to produce non-negative weights, which combine these sense vectors additively:

$$y = \text{Unembed}\left(\sum_{i=1}^{n} \alpha_i(\mathbf{x}) \cdot v_{\text{sense}}^{(i)}\right)$$

By construction, the output representation is a weighted sum of independent sense vectors, enabling direct inspection and targeted intervention. Subsequent work extends this framework to non-alphabetic languages via Character-level Chinese BLMs (Sun and Hewitt, 2023), which learn interpretable sense decompositions at the character level, as well as to downstream control tasks, including model editing via canonical examples, where modifying or fine-tuning specific sense vectors enables localized behavioral changes without broadly perturbing the model (Hewitt et al., 2024). However, representing contextual meaning as additive combinations of fixed sense vectors may limit expressivity when sense interactions are non-linear.

Semantic Concept Integration. While Backpack Language Models decompose lexical inputs, Tack et al. (2025) introduce CoCoMix to enforce decomposability at the level of higher-level semantic concepts. Instead of operating solely over discrete token representations, CoCoMix integrates SAE into pretraining, training the model to predict continuous concept representations alongside next-token probabilities. These predicted concept vectors are interleaved with hidden states, encouraging explicit reasoning over disentangled semantic features during generation. By treating interpretable concepts as structured components of the forward pass, CoCoMix enables targeted control over generation while preserving output coherence, at the cost of introducing additional training structure and reliance on the quality of concept representations.

4.4 Explicit Modularization

In practice, explicit modularization is most often realized through MoE architectures. While standard MoE models are primarily designed for scalability, their expert representations and routing mech-

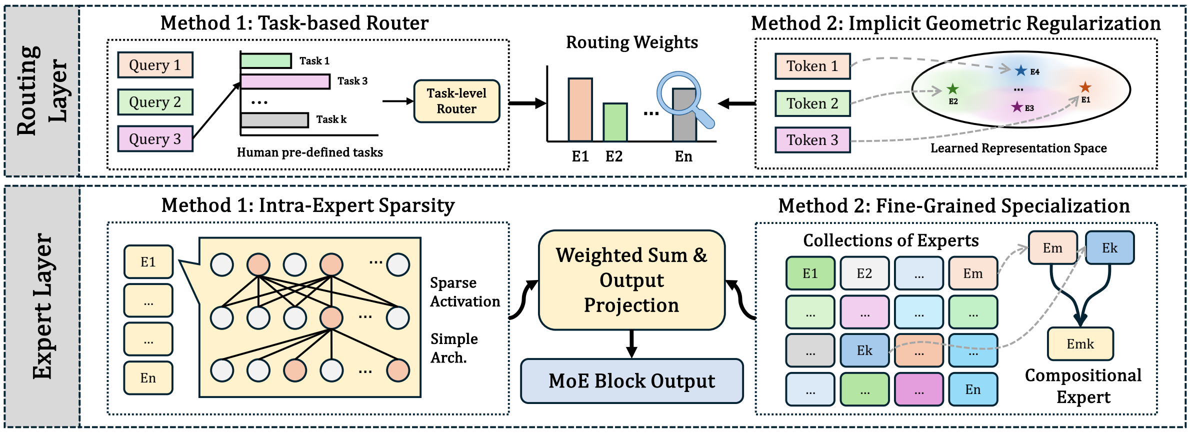

Figure 4: **Architectural strategies for intrinsically interpretable MoEs.** We distinguish between methods enforcing intra-expert sparsity, fine-grained decomposition, and semantically aligned routing.

anisms are typically optimized for load balancing rather than semantic transparency (Fedus et al., 2022). Recent work revisits MoE design with interpretability as a central goal. We organize these methods into three architectural strategies, illustrated in Figure 4, which we discuss in turn below.

Enforcing Intra-Expert Sparsity and Simplicity. A direct approach is to constrain the experts themselves. One strategy replaces smooth activations (e.g., GeLU) with hard thresholds like ReLU. For instance, MoE-X (Yang et al., 2025) uses this to enforce sparsity on hidden states, helping to disentangle features. A parallel strategy simplifies the expert architecture. Methods like MoV and MoLORA (Zadouri et al., 2024) replace full MLPs with lightweight vectors or low-rank adapters. While primarily efficient, these linear or low-rank experts are also much easier to analyze than deep, non-linear MLPs. However, despite simplifying expert internals, these models still rely on routing and sparse expert selection, and therefore remain sensitive to load imbalance during training.

Architecting for Fine-Grained Decomposition. Another strategy seeks monosemanticity by scaling the number of experts to match the number of features. Building on tensor decomposition methods like MPO-MoE (Gao et al., 2022), architectures such as MONET (Park et al., 2025) and MxD (Oldfield et al., 2025) use product key composition and flexible tensor factorization. These techniques construct hundreds of thousands of fine-grained sublayers from a compact parameter set, effectively treating the MoE layer as a sparse dictionary of specialized linear transformations. Despite their improved granularity, the expansion of the expert space amplifies routing sensitivity, making training vulnerable to routing imbalance and expert underutilization.

Designing Semantically Aligned Routing Policies. While early efforts simply mapped experts to languages (Zhao et al., 2024b), recent work distinguishes between explicit structural alignment and implicit geometric regularization. In the explicit paradigm, models like Task-Based MoE (Pham et al., 2023) and THOR-MoE (Liang et al., 2025) directly integrate task context into the router, whereas Apollo-MoE (Zheng et al., 2025) organizes experts by linguistic families. In the implicit paradigm, researchers enforce constraints on the routing space itself: RoMA (Li et al., 2025) aligns routing manifolds with task embeddings, and US-MoE (Do et al., 2025) reframes selection as a linear programming problem. Supporting these directions, recent analyses confirm that routing decisions follow distinct layer-wise patterns and can be predictably steered to alter model behavior (Bandarkar et al., 2025; Zheng et al., 2025).

4.5 Latent Sparsity Induction

Unlike explicit modular architectures such as Mixture-of-Experts, latent sparsity induction aims to encourage modular and interpretable structure to *emerge* within otherwise standard Transformer architectures. Rather than prescribing a fixed decomposition, these methods introduce inductive biases that promote selective activation and reduce superposition, allowing the model to organize its computation into sparse, task-specific subcircuits.

Enforcing Weight Sparsity. A primary approach to latent sparsity induction enforces sparsity directly at the level of model parameters. The underlying hypothesis is that polysemanticity arises from dense connectivity, where individual neurons participate in many unrelated computations. To address this, recent work trains Transformer models with strong sparsity constraints (Gao et al., 2025), imposing sparsity throughout optimization rather than post-hoc pruning, forcing the model to allocate its limited connections more selectively and reducing feature superposition (Elhage et al., 2022). A direct consequence of sparse training is the emergence of compact and interpretable computational circuits.

While weight-sparse models offer strong interpretability benefits, they are often inefficient on current hardware. Gao et al. (2025) train sparse and dense models jointly, coupled through linear mappings between their representations, making interpretable features discovered in the sparse model usable to explain or annotate the latent space of the dense model. However, Gao et al. (2025) note that enforcing weight sparsity introduces a capability-interpretability trade-off and remains difficult to scale.

Conditional Activation via Gated Architectures. Latent sparsity can also be induced at the level of activations rather than weights. Gated architectures, such as GLU and SwiGLU (Dauphin et al., 2017; Shazeer, 2020), introduce conditional computation within Transformer feed-forward layers. Unlike standard pointwise activations (e.g., ReLU (Glorot et al., 2011) or GeLU (Hendrycks and Gimpel, 2023)), GLUs compute an element-wise product between a value projection and a learned gate:

$$\mathrm{GLU}(x) = (xW) \odot \sigma(xV)$$

Here, the gating term selectively suppresses or amplifies features on a per-input basis, effectively routing information through different subspaces. While GLUs do not enforce strict sparsity, their conditional structure encourages selective pathway activation, reducing entanglement and promoting emergent modularity. This form of activation-level sparsity complements weight-sparse approaches by enabling input-dependent specialization without explicitly defined modules.

5 Open Challenges and Future Directions

Despite recent progress, intrinsic interpretability for LLMs remains an open and rapidly evolving research area. We highlight several key challenges and promising directions for future work.

Defining and Evaluating Intrinsic Interpretability. A central challenge is the lack of rigorous and widely accepted definitions and evaluation metrics for intrinsic interpretability (Doshi-Velez and Kim, 2017; Lipton, 2018). Although intrinsic approaches aim to align model structure with explanation, it remains unclear how to quantitatively assess the quality, completeness, or usability of such explanations. Existing evaluations rely primarily on proxy measures such as sparsity, modularity, or disentanglement, which do not reliably reflect human interpretability or task relevance. In particular, structural properties like sparsity do not guarantee semantic clarity, as features may remain polysemantic or lack stable human interpretable meaning (Elhage et al., 2022). Moreover, without principled verification, intrinsically generated explanations may appear plausible while failing to faithfully represent the model's true reasoning (Turpin et al., 2023; Singh et al., 2024). Developing evaluation frameworks that balance faithfulness, human comprehensibility, and downstream utility therefore remains an important open problem.

Balancing Interpretability and Expressivity. Although recent work suggests that interpretability and performance need not be mutually exclusive, intrinsic constraints may still limit model expressivity or generalization in practice (Gao et al., 2025; Sun et al., 2025). Understanding when and how architectural biases such as modularity, sparsity, or concept alignment improve or hinder learning remains an open question. Future research should aim to characterize these trade-offs more precisely, identifying regimes in which intrinsic interpretability enhances robustness and generalization rather than restricting model capacity.

Scalability to Large-Scale Language Models. Most intrinsically interpretable architectures have so far been evaluated only at small or moderate scales (Park et al., 2025; Tamkin et al., 2024; Tack et al., 2025). Extending these designs to large language models with billions of parameters introduces additional challenges, including increased routing complexity, memory overhead, and optimization instability. Demonstrating that intrinsic interpretability can be preserved at scale therefore remains a critical step toward practical deployment.

Training Efficiency and Optimization Stability. Intrinsic interpretability often introduces additional constraints or architectural components, such as sparse activations, modular routing, or complex functional parameterizations, which can complicate optimization and increase training cost (Gao et al., 2025; Jin et al., 2025; Liu et al., 2025). Improving the efficiency and stability of training intrinsically interpretable models is therefore an important direction for future work. This includes developing better optimization strategies, regularization schemes, and hardware-aware implementations that make intrinsic designs competitive with standard dense architectures.

Complementarity with Post-hoc Analysis. Although intrinsic and post hoc interpretability are often framed as distinct paradigms, they need not be mutually exclusive. Post-hoc tools can act as diagnostic instruments for validating and stress testing intrinsically interpretable models, while intrinsic structural constraints can in turn improve the faithfulness of post-hoc analyses. Moreover, insights derived from post-hoc methods can inform intrinsic design, for example by guiding concept discovery, feature selection, or module construction (Tack et al., 2025). Developing principled frameworks that integrate intrinsic architectures with post-hoc analysis therefore represents a promising direction for building more transparent and reliable LLMs.

6 Conclusion

In this paper, we present a comprehensive survey of intrinsic interpretability for large language models. We first clarify the key distinctions between intrinsic interpretability and post-hoc explanation methods, highlighting their conceptual differences and respective strengths. We then categorize and analyze existing approaches through the lens of five core design principles: functional transparency, concept alignment, explicit modularization, latent sparsity induction, and representational decomposability. In addition, we synthesize recent advances across model architectures and training strategies, and identify five key challenges and future directions for intrinsically interpretable model design. Our goal is to provide a structured and accessible resource for researchers interested in building transparent, interpretable-by-design LLMs. To aid in this effort, we create a continually updated paper list in a GitHub repository as follows: Survey-Intrinsic-Interpretability-of-LLMs.

Limitations

While this survey aims to provide a comprehensive overview of intrinsic interpretability for large language models, several limitations regarding its scope should be noted. The field is evolving rapidly, and despite efforts to incorporate work up to late 2025, new architectures and training techniques continue to emerge. Accordingly, this survey reflects a snapshot of the current literature and may not cover unpublished or concurrent preprints. To balance breadth and clarity, we emphasize unifying principles rather than exhaustive technical detail for individual methods.

Acknowledgement

This work was supported in part by the Beijing Major Science and Technology Project under Contract No. Z251100008125054. This work was supported by the Beijing Academy of Artificial Intelligence (BAAI).

References

David Alvarez Melis and Tommi Jaakkola. 2018. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Lucas Bandarkar, Chenyuan Yang, Mohsen Fayyaz, Junlin Hu, and Nanyun (Violet) Peng. 2025. Multilingual routing in mixture-of-experts. *CoRR*, abs/2510.04694.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, and 95 others. 2022. On the opportunities and risks of foundation models. *Preprint*, arXiv:2108.07258.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Moritz Böhle, Mario Fritz, and Bernt Schiele. 2022. B-cos networks: Alignment is all we need for interpretability. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10319–10328.

Chun-Hao Chang, Rich Caruana, and Anna Goldenberg. 2022. Node-gam: Neural generalized additive model for interpretable deep learning. *Preprint*, arXiv:2106.01613.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, and 48 others. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 933–941. PMLR.

Giang Do, Hung Le, and Truyen Tran. 2025. Unified sparse mixture of experts. *Preprint*, arXiv:2503.22996.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *Preprint*, arXiv:1702.08608.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Preprint*, arXiv:2209.10652.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39.

Leo Gao, Achyuta Rajaram, Jacob Coxon, Soham V. Govande, Bowen Baker, and Dan Mossing. 2025. Weight-sparse transformers have interpretable circuits. *Preprint*, arXiv:2511.13653.

Ze-Feng Gao, Peiyu Liu, Wayne Xin Zhao, Zhong-Yi Lu, and Ji-Rong Wen. 2022. Parameter-efficient mixture-of-experts architecture for pre-trained language models. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3263–3273, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pages 315–323. JMLR.org.

Robert M. Gray. 1984. Vector quantization. *IEEE ASSP Magazine*, 1:4–29.

Hongcan Guo, Haolang Lu, Guoshun Nan, Bolun Chu, Jialin Zhuang, Yuan Yang, Wenhao Che, Sicong Leng, Qimei Cui, and Xudong Jiang. 2025. Advancing expert specialization for better moe. *Preprint*, arXiv:2505.22323.

Trevor Hastie and Robert Tibshirani. 1986. Generalized Additive Models. *Statistical Science*, 1(3):297 – 310.

Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. 2022. Addressing leakage in concept bottleneck models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Dan Hendrycks and Kevin Gimpel. 2023. Gaussian error linear units (gelus). *Preprint*, arXiv:1606.08415.

John Hewitt, Sarah Chen, Lanruo Lora Xie, Edward Adams, Percy Liang, and Christopher D. Manning. 2024. Model editing with canonical examples. *Preprint*, arXiv:2402.06155.

John Hewitt, John Thickstun, Christopher D. Manning, and Percy Liang. 2023. Backpack language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9103–9125. Association for Computational Linguistics.

Yuntian Hou, Tianrui Ji, Di Zhang, and Angelos Stefanidis. 2025. Kolmogorov-arnold networks: A critical assessment of claims, performance, and practical viability. *Preprint*, arXiv:2407.11075.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4198–4205. Association for Computational Linguistics.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Lukas Vierling, Donghai Hong, Jiayi Zhou, Zhaowei Zhang, Fanzhi Zeng, Juntao Dai, Xuehai Pan, Kwan Yee Ng, Aidan O'Gara, Hua Xu, Brian Tse, and 7 others. 2025. Ai alignment: A comprehensive survey. *Preprint*, arXiv:2310.19852.

Chao Jin, Ziheng Jiang, Zhihao Bai, Zheng Zhong, Juncai Liu, Xiang Li, Ningxin Zheng, Xi Wang, Cong Xie, Qi Huang, Wen Heng, Yiyuan Ma, Wenlei

Bao, Size Zheng, Yanghua Peng, Haibin Lin, Xuanzhe Liu, Xin Jin, and Xin Liu. 2025. Megascale-moe: Large-scale communication-efficient training of mixture-of-experts models in production. *Preprint*, arXiv:2505.11432.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR.

Zhongyang Li, Ziyue Li, and Tianyi Zhou. 2025. Routing manifold alignment improves generalization of mixture-of-experts llms. *Preprint*, arXiv:2511.07419.

Yunlong Liang, Fandong Meng, and Jie Zhou. 2025. THOR-MoE: Hierarchical task-guided and context-responsive routing for neural machine translation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21433–21445, Vienna, Austria. Association for Computational Linguistics.

Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. 2021. Explainable AI: A review of machine learning interpretability methods. *Entropy*, 23(1):18.

Zachary C. Lipton. 2018. The mythos of model interpretability. *Commun. ACM*, 61(10):36–43.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y. Hou, and Max Tegmark. 2025. KAN: kolmogorov-arnold networks. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. 2013. Accurate intelligible models with pairwise interactions. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Christos Louizos, Max Welling, and Diederik P. Kingma. 2018. Learning sparse neural networks through l_0 regularization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. Post-hoc interpretability for neural nlp: A survey. *ACM Computing Surveys*, 55(8):1–42.

Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. 2021. Promises and pitfalls of black-box concept learning models. *Preprint*, arXiv:2106.13314.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Christoph Molnar. 2025. *Interpretable Machine Learning*, 3 edition.

John Ashworth Nelder and Robert WM Wedderburn. 1972. Generalized linear models. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 135(3):370–384.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. 2019. Interpretml: A unified framework for machine learning interpretability. *Preprint*, arXiv:1909.09223.

nostalgebraist. 2020. Interpreting gpt: The logit lens. LessWrong blog post.

Tuomas P. Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. 2023. Label-free concept bottleneck models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

James Oldfield, Shawn Im, Sharon Li, Mihalis A. Nicolaou, Ioannis Patras, and Grigorios G Chrysos. 2025. Towards interpretability without sacrifice: Faithful dense layer decomposition with mixture of decoders. *Preprint*, arXiv:2505.21364.

Avash Palikhe, Zhenyu Yu, Zichong Wang, and Wenbin Zhang. 2025. Towards transparent ai: A survey on explainable large language models. *Preprint*, arXiv:2506.21812.

Jungwoo Park, Ahn Young Jin, Kee-Eung Kim, and Jaewoo Kang. 2025. Monet: Mixture of monosemantic experts for transformers. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Urja Pawar, Donna O'Shea, Susan Rea, and Ruairi O'Reilly. 2020. Explainable AI in healthcare. In *2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment, CyberSA 2020, Dublin, Ireland, June 15-19, 2020*, pages 1–2. IEEE.

Michael T. Pearce, Thomas Dooms, Alice Rigg, José Oramas, and Lee Sharkey. 2025. Bilinear mlps enable weight-based mechanistic interpretability. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Hai Pham, Young Jin Kim, Subhabrata Mukherjee, David P. Woodruff, Barnabas Poczos, and Hany Hassan Awadalla. 2023. Task-based moe for multitask multilingual machine translation. *Preprint*, arXiv:2308.15772.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2021. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3363–3377, Online. Association for Computational Linguistics.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144. ACM.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.*, 1(5):206–215.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeffrey Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Isaac Bloom, Stella Biderman, Adrià Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, William Saunders, and 10 others. 2025. Open problems in mechanistic interpretability. *Trans. Mach. Learn. Res.*, 2025.

Noam Shazeer. 2020. Glu variants improve transformer. *Preprint*, arXiv:2002.05202.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. 2024. Rethinking interpretability in the era of large language models. *Preprint*, arXiv:2402.01761.

Chung-En Sun, Tuomas P. Oikarinen, Berk Ustun, and Tsui-Wei Weng. 2025. Concept bottleneck large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Hao Sun and John Hewitt. 2023. Character-level Chinese backpack language models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 106–119, Singapore. Association for Computational Linguistics.

Jihoon Tack, Jack Lanchantin, Jane Yu, Andrew Cohen, Ilia Kulikov, Janice Lan, Shibo Hao, Yuandong Tian, Jason Weston, and Xian Li. 2025. Llm pretraining with continuous concepts. *Preprint*, arXiv:2502.08524.

Alex Tamkin, Mohammad Taufeeque, and Noah D. Goodman. 2024. Codebook features: Sparse and discrete interpretability for neural networks. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1332 others. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6306–6315.

Moritz Vandenhirtz, Sonia Laguna, Ricards Marcinkevics, and Julia E. Vogt. 2024. Stochastic concept bottleneck models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yifan Wang, Sukrut Rao, Ji-Ung Lee, Mayank Jobanputra, and Vera Demberg. 2025. B-cos LM: Efficiently transforming pre-trained language models for improved explainability. *Transactions on Machine Learning Research*.

Xingyi Yang, Constantin Venhoff, Ashkan Khakzar, Christian Schröder de Witt, Puneet K. Dokania, Adel Bibi, and Philip Torr. 2025. Mixture of experts made intrinsically interpretable. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

Zebin Yang, Aijun Zhang, and Agus Sudjianto. 2021. Gami-net: An explainable neural network based on generalized additive models with structured interactions. *Pattern Recognit.*, 120:108192.

Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermis, Acyr Locatelli, and Sara Hooker. 2024. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frédéric Precioso, Stefano Melacci, Adrian Weller, Pietro Lió, and Mateja Jamnik. 2022. Concept embedding models: Beyond the accuracy-explainability trade-off. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024a. Explainability for large language models: A survey. *ACM Trans. Intell. Syst. Technol.*, 15(2):20:1–20:38.

Xinyu Zhao, Xuxi Chen, Yu Cheng, and Tianlong Chen. 2024b. Sparse moe with language guided routing for multilingual machine translation. In *The Twelfth International Conference on Learning Representations*.

Guorui Zheng, Xidong Wang, Juhao Liang, Nuo Chen, Yuping Zheng, and Benyou Wang. 2025. Efficiently democratizing medical llms for 50 languages via a mixture of language family experts. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

A Summary of Intrinsic Interpretability Methods

This appendix provides a consolidated summary of the intrinsic interpretability methods discussed throughout the survey. Table 1 organizes these approaches by design principle, highlighting their key mechanisms, sources of interpretability, and practical trade-offs in training cost, inference cost, and performance relative to black-box baselines. The table is intended as a reference for cross-method comparison rather than a comprehensive evaluation.

Method	Reference	Key Mechanism	Interp. Source	Train Cost	Infer. Cost	Perf.
Functional Transparency (Sec 4.1)						
GAMs	(Hastie and Tibshirani, 1986)	Additive smooth functions	Shape functions	Low	Low	Linear
GA^2M	(Lou et al., 2013)	Pairwise interaction terms	Interaction maps	Medium	Low	Moderate
EBMs	(Nori et al., 2019)	Boosting for additive terms	Shape/Interaction	Medium	Low	Moderate
GAMI-Net	(Yang et al., 2021)	Neural shape functions	Individual NNs	Medium	Medium	Moderate
NODE-GAM	(Chang et al., 2022)	Neural subnetwork per feature	Shape functions	Medium	Medium	Moderate
SENN	(Alvarez Melis and Jaakkola, 2018)	Basis concepts + Relevance scores	Concepts & Relevance	Medium	Medium	≈
B-cos Networks	(Böhle et al., 2022)	Weight–input alignment transform	Linear explanations	Medium	Low	≈
B-cos LMs	(Wang et al., 2025)	B-cos transforms + fine-tuning	Linear explanations	Medium	Low	≈
KANs	(Liu et al., 2025)	Learnable splines on edges	1D edge functions	High	High	↓
Bilinear MLPs	(Pearce et al., 2025)	Bilinear interactions	Weight tensors	High	High	≈
Concept Alignment (Sec 4.2)						
Standard CBMs	(Koh et al., 2020)	Hard concept bottleneck	Concept scores	Low	Low	↓
SCBMs	(Vandenhirtz et al., 2024)	Joint concept distribution	Concept dependencies	Medium	Low	↓
Hybrid CBMs	(Havasi et al., 2022; Mahinpei et al., 2021)	Residual side-channel	Concepts + Residual	Low	Low	≈
CB-LLM	(Sun et al., 2025)	Hybrid bottleneck + Adversarial	Concepts + Latent	High	Low	≈
Label-free CBM	(Oikarinen et al., 2023)	Auto-discovery via CLIP	Concept scores	Medium	Low	↓
CEMs / IntCEMs	(Zarlenga et al., 2022)	Concept embeddings	Concept Vectors	Medium	Low	≈
Codebook Features	(Tamkin et al., 2024)	Vector Quantization (VQ)	Discrete Codes	Medium	Low	↓
Representational Decomposability (Sec 4.3)						
Backpack	(Hewitt et al., 2023)	Sense vectors + Context weights	Sense vectors	Medium	High	↓
Char-BLM	(Sun and Hewitt, 2023)	Character-level sense vectors	Character senses	Medium	High	↓
CoCoMix	(Tack et al., 2025)	Concept prediction & mixing	Continuous concepts	Medium	High	≈
Explicit Modularization (MoEs) (Sec 4.4)						
— Intra-Expert Sparsity —						
MoE-X	(Yang et al., 2025)	Sparsity-aware routing + ReLU	Sparse Experts	Low	Low	≈
MoV	(Zadouri et al., 2024)	Mixture of Vectors (Linear)	Linear Units	Low	Low	≈
MoLORA	(Zadouri et al., 2024)	Mixture of LoRA adapters	Low-rank Adapters	Low	Low	≈
— Fine-Grained Decomposition —						
MONET	(Park et al., 2025)	Product Key Composition	Monosemantic Exp.	High	Medium	≈
MxD	(Oldfield et al., 2025)	Tensor factorization	Linear sublayers	High	Medium	≈
MPO-MoE	(Gao et al., 2022)	Matrix Product Operator	Shared tensors	Medium	Medium	≈
— Semantically Aligned Routing —						
Task-Based MoE	(Pham et al., 2023)	Task embeddings + Adapters	Task Adapters	Low	Low	↑
Lingual-SMoE	(Zhao et al., 2024b)	Language-guided routing	Language Experts	Low	Low	↑
THOR-MoE	(Liang et al., 2025)	Hierarchical Context-Routing	Domain Experts	Medium	Low	↑
Apollo-MoE	(Zheng et al., 2025)	Language Family Grouping	Family Experts	Medium	Low	↑
RoMA	(Li et al., 2025)	Manifold Alignment Reg.	Routing Geometry	Medium	None	↑
USMoE	(Do et al., 2025)	Linear Programming Routing	Unified Scores	Low	Low	↑
Orthogonality	(Guo et al., 2025)	Orthogonality & Variance loss	Exclusive Experts	Low	Low	≈
Latent Sparsity Induction (Sec 4.5)						
Weight-Sparse	(Gao et al., 2025)	L_0 Regularization	Sparse Circuits	Very High	Low	↓
GLUs / SwiGLU	(Dauphin et al., 2017; Shazeer, 2020)	Gated Linear Units	Activation Paths	Low	Low	↑

Table 1: A comprehensive summary of intrinsic interpretability architectures covering all methods discussed in this survey. **Perf.** denotes reported performance vs. black-box baselines: ↑ (Improved), ≈ (Similar), ↓ (Trade-off). Note: KANs and Bilinear MLPs are categorized under Functional Transparency following the text structure.