Machine Learning with Applications 24 (2026) 100897

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Linear Lens: A human-centered, non-interventional mechanistic approach to explainable AI☆

Muhammad Umair Danish [a], Memoona Aziz [b], Umair Rehman [b], Katarina Grolinger [a,*]

[a] Electrical and Computer Engineering, Western University, 1151 Richmond St, London, N6A 3K7, Ontario, Canada
[b] Computer Science, Western University, 1151 Richmond St, London, N6A 3K7, Ontario, Canada

ARTICLE INFO

Keywords:
Explainable artificial intelligence (XAI)
Interpretable machine learning
Mechanistic interpretability
Transparency
Model auditing
Trustworthy AI
Energy forecasting

ABSTRACT

Deep neural networks (DNNs) achieve high accuracy but remain black-box, which undermines user trust and hinders deployment in high-stakes settings. Existing explainability methods often intervene in the model, such as through perturbations or activation patching, raising concerns about faithfulness and increasing cognitive load for end users. We introduce Linear Lens, a non-interventional, behavioral, and mechanistic interpretability method that explains a model strictly in the logic it used during inference. Linear Lens computes the influence of feature neurons on pre-activations and uses an entropy-based hypothesis test to classify neurons as monosemantic, polysemantic, or dead. For polysemantic neurons, it performs regression validation to confirm the driving feature sets. To enhance understanding at the layer level, a Qualitative Symbolic Matrix summarizes how inputs influence all neurons in an intuitive, human-readable form. For deeper layers, We classify neurons and labeled them as unimodal, multimodal, or muted, and trace their compositional pathways from inputs to outputs. This preserves end-to-end faithfulness without altering model memory, such as weights or activations. Linear Lens consistently identifies predominantly polysemantic neurons in the first layer and multimodal neurons in deeper layers (over 90%), and provides clear explanations of their functional roles. To ensure applicability, we evaluate Linear Lens on ten real-world energy consumption datasets using several DNN architectures, including Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), and Transformer models of varying sizes. To ensure user comprehension and appropriate cognitive load, a statistically validated user study (N = 400) was conducted, showing that the explanations are understandable and cognitively efficient. Future work may deploy Linear Lens to other critical domains to further evaluate its impact on trust and usability constructs.

1. Introduction

Artificial intelligence technologies, such as deep neural networks (DNNs), have been deployed across various sectors, including energy and industrial applications (Danish et al., 2025; Rizk et al., 2018; Tjoa & Guan, 2021). These DNNs provide accurate and fast services to users as intelligent machines, but their operations remain black-box and do not provide reasoning behind specific decisions (Casalicchio et al., 2019). As a result, a considerable trust deficit exists between users and DNNs. For example, a study indicates that 43% of adults in the US believe AI will harm them in real life, while only 24% of them think it is beneficial, and the remaining 33% are unsure (Center, 2025). This uncertainty arises largely because DNNs are black-box systems, and without clear logical reasoning behind decisions, users will continue to widen the trust deficit despite improved performance and broader deployment (Rizk et al., 2018).

To address the lack of trust in AI and improve users' understanding of decisions made by black-box DNNs, various methods have been proposed in the literature, which can be divided into three major categories. The first is feature attribution methods, which establish mappings between inputs and outputs (Lundberg & Lee, 2017). These techniques emphasize determining how input features contribute to outputs, but do not explore the inner workings of DNNs. The second category is concept-based interpretability, which partially unpacks the model's operations and presents them as human-understandable concepts (Zou et al., 2023). To fully understand the operations of DNNs, mechanistic interpretability has been proposed, which reverse-engineers the model by breaking it down into smaller components and

☆ This work was supported in part by the Climate Action and Awareness Fund [EDF-CA-2021i018, Environnement Canada, K. Grolinger] and in part by the Canada Research Chairs Program [CRC-2022-00078, K. Grolinger].

* Corresponding author.
E-mail addresses: mdanish3@uwo.ca (M.U. Danish), maziz86@uwo.ca (M. Aziz), urehman6@uwo.ca (U. Rehman), kgroling@uwo.ca (K. Grolinger).

https://doi.org/10.1016/j.mlwa.2026.100897
Received 19 December 2025; Received in revised form 18 March 2026; Accepted 2 April 2026
Available online 4 April 2026

seeks to understand the function of each part (Bricken et al., 2023; Conmy et al., 2023). Despite its depth, mechanistic interpretability adds another layer of complexity for users because its unpacking process itself is complex and difficult to trust, as it modifies the original logic by introducing extensive intervention.

While existing explainability techniques have made considerable progress, there remains an opportunity to provide explanations that further enhance user trust and understanding (Montag et al., 2024; Wani, Kumar, Bedi, &, Rida, 2024). The loss of user trust arises from two main factors. First, the way explanations and reasoning behind DNN decisions are presented to users. Second, most existing studies rely on interventions, such as modifying the model's original logic, for example, by adding perturbations or altering the model's activations (Rong et al., 2024; Saw et al., 2025). Although these explanations are theoretically accurate, they alter the model's original logic, which undermines user trust because the model is no longer evaluated in the same way it made its original decision (Wani, Kumar, &, Bedi, 2024).

To enhance user trust in explanations, we adopt a non-interventional, purely behavioral technique in which, instead of modifying the trained model, we observe the network in its original logic, without adding perturbations, altering weights, or patching activations. This method ensures that the model is explained exactly as it operated during the decision-making process that requires explanation. We refer to this philosophy as behavioral or observational explainability, which enhances user trust by providing explanations derived directly from the model's natural behavior, without any internal modifications. To achieve this, it is crucial to understand the root cause of the black-box nature of neural networks, which lies in non-linearity. A neural network layer, before applying a non-linear activation function, remains a white-box system and is fully explainable. However, once non-linearity is applied, the layer becomes a black box, both theoretically and practically irreversible and unexplainable. However, the post-activation output of a layer is computed from pre-activations, which are part of the white-box computation. These pre-activations are fully traceable and provide an opportunity to understand the behavior of the network even after non-linearity is applied.

Consequently, we propose a behavioral mechanistic interpretability approach, hereafter referred to as Linear Lens, for interpreting deep neural networks. Linear Lens interpretability procedure reveals the inner operations of neural networks by defining the smallest functional unit, known as the neuron. The Linear Lens explains the functional role of each neuron individually and traces how input features pass through the network, influencing activations layer by layer until the output is produced. The primary focus of The Linear Lens is to understand the function of individual neurons, which in turn enables a precise understanding of the entire network's behavior. The Linear Lens consists of four phases of explanation. Phase 1 defines and explains the role of individual neurons in the model's first layer, where input features are received, providing a quick insight into how features begin to influence the network. Phase 2 extends this analysis to the entire layer, aggregating neuron behaviors to explain how inputs are collectively processed. Phase 3 moves deeper into the network, characterizing the function of neurons in hidden layers and explaining how signals are transformed and propagated through the model's internal structure. Phase 4 evaluates the system's interpretability through statistically validated user studies, ensuring that the explanations produced are not only technically faithful but also understandable and trustworthy from a human perspective. The key contributions of this paper are as follows:

1. We propose Linear Lens, a non-interventional, pure behavioral, mechanistic, and human-centered interpretability procedure that explains neural networks by directly analyzing the model at its original logic, without applying perturbations, patching activations, or modifying internal components.
2. We introduce a hypothesis-driven neuron discovery framework that classifies neurons into monosemantic, polysemantic, and dead categories using entropy-based statistical testing at the input layer. This behavioral characterization extends to deeper layers through unimodal, multimodal, muted, and compositional roles, enabling semantic traces to be preserved and propagated from inputs to final predictions.
3. We propose the Qualitative Symbolic Matrix (QSM), a human-readable symbolic representation that aggregates neuron-level behaviors into an interpretable layer-level fingerprint. This enables intuitive inspection, comparison, and auditing of learned representations across models and datasets.
4. We evaluate Linear Lens through statistically validated user studies, demonstrating that the explanations it provides are understandable, actionable, and capable of enhancing user trust.
5. We conduct comprehensive experiments across ten residential energy consumption datasets, showing that Linear Lens can be deployed across tasks with varied domains.

While the proposed Linear Lens is designed as a general, model-agnostic approach, this paper purposefully focuses on energy forecasting as its primary application domain. The proposed methodology is not domain-specific and can be extended to other structured prediction tasks, such as finance, healthcare, or industrial forecasting, which we leave for future work. Moreover, the questionnaire used for the human study was developed and validated in our prior work and is currently under review in a separate double-blind manuscript; accordingly, questionnaire design is not claimed as a contribution of this paper, which focuses instead on deploying the instrument to evaluate human cognitive understanding of the proposed explanations. This study has been approved by Western University's Non-Medical Research Ethics Board (Approval No. 125784).

The remainder of the paper is structured as follows: Section 2 reviews related work. Section 3 introduces the proposed Linear Lens. Section 4 presents the evaluation process and discusses the results. Finally, Section 5 concludes the paper.

2. Related work

This section describes a overview of recent literature concerning feature attribution, concept-based interpretability, mechanistic interpretability, and human-centered interpretability.

2.1. Feature attribution

Feature attribution-based methods examine how input features influence the model's decision. These methods do not unpack the model's internal operations but instead focus on how the inputs influence the outputs from the surface. Feature attribution can be divided into two broader sub-categories: the first is perturbation-based attribution, which measures the impact of each input feature by systematically perturbing it and observing the resulting change in output. The second is gradient-based attribution, which leverages the model's gradients to estimate the contribution of each input feature to the output. Among perturbation-based methods, Shapley Additive Explanations (SHAP) (Lundberg & Lee, 2017) and its variants are widely deployed. For instance, Kernel SHAP utilizes random sampling to approximate Shapley values in a model-agnostic manner (Cremades et al., 2025), while Tree SHAP is a model-specific variant designed for tree-based models. Moreover, the General Explainable Sentiment Classification (GESC) has been proposed to enhance interpretability in aspect-based tasks (Pan et al., 2024), while Noor et al. reduce the computational complexity of SHAP while maintaining accuracy. However, regardless of these improvements, SHAP variants appear to be theoretically accurate but overlook the inner operation (Noorchenarboo & Grolinger, 2025).

Gradient-based attribution methods form another category of feature attribution that seeks to explain predictions by attributing the output decision to individual input features using model gradients. For instance, Integrated Gradients is a foundational technique that relies on model gradients; however, it is not fully model-agnostic, which restricts its applicability to tasks specific to a particular model (Sundararajan et al., 2017). Weighted Integrated Gradients is another technique that unsupervisedly assesses baseline usefulness to produce feature attribution by including weighting functions into gradient computations (Tuan et al., 2025). However, gradient signals can be noisy or unstable in deep or recurrent models, which lowers their reliability in critical applications. Both gradient and behavioral attribution methods provide only surface-level input–output mappings. While they may be theoretically accurate, they do not provide complete explanations, as the model's internal operations remain encapsulated and hidden from users. Hence, user trust may decrease due to the lack of complete explanations (Lundberg & Lee, 2017).

2.2. Concept-based interpretability

Concept-based interpretability employs a top-down approach to understand the reasoning behind model decisions by partially unpacking its inner operations, deep features, or representations, and transforming them to extract high-level user concepts (Koh et al., 2020). Instead of analyzing activations or model weights directly, Concept-based interpretability concentrates on identifying semantically meaningful concepts that influence behavior and then translating them into human-understandable concepts. For instance, rather than reporting that "NEURON 15 RESPONDED STRONGLY TO CERTAIN WORD EMBEDDINGS WITH COEFFICIENT 0.6", a concept-based explanation might state, "THE MODEL DETECTED SARCASTIC TONE AND NEGATIVE WORDS, SO IT CLASSIFIED THE REVIEW AS DISSATISFIED". Techniques in this category include training supervised auxiliary classifiers and using unsupervised contrastive or structured probes to extract latent knowledge from deep representations (Hewitt & Liang, 2019). Neural representation analysis (Kornblith et al., 2019) has also been employed to measure the similarity between internal representations across different models. More recent work refers to concept-based interpretability as REPRESENTATION ENGINEERING, in which internal concepts are manipulated in a stable, understandable manner to enhance model behavior and safety (Li et al., 2023).

Yeh et al. (2020) proposed completeness-aware concept-based explanations and quantified the completeness of concept explanations, ensuring that the identified concepts sufficiently capture the model's decision-making process. This approach assesses how a set of human-interpretable concepts can explain the network's predictions, addressing limitations in prior methods that can lead to misleading interpretations when concepts are incomplete. Zarlenga et al. proposed TabCBM, a concept-based interpretable neural network for tabular data and outlined a model that integrates human-defined concepts directly into the learning process to ground predictions in understandable terms. The TabCBM enables the explicit tracing of how concepts contribute to outcomes, thereby overcoming the opacity usually found in traditional black-box models for tabular inputs. Evaluations on real-world datasets showed that TabCBM not only maintains competitive performance but also provides actionable insights, making it suitable for domains that require regulatory compliance.

Concept-based interpretability enables a high level of trust by breaking down the model's inner workings and translating them into concepts that enhance human cognition and understanding. However, it provides only a partial view of the model's internal workings, focusing on prominent high-level concepts while leaving the full network architecture and fine-grained inner mechanisms unexplored.

2.3. Mechanistic interpretability

Mechanistic interpretability employs a bottom-up strategy to reverse-engineer neural networks by breaking them down into their smaller components, thereby explaining the role of each component, which ultimately helps to understand the inner workings of the entire model
(Williams et al., 2025). Mechanistic interpretability aims to thoroughly unpack the model, from layers to individual neurons, revealing how each component, whether small or large, affects the final decision (Bereska & Gavves, 2025; Goldowsky-Dill et al., 2023). Unlike concept-based interpretability, which only extracts high-level, understandable concepts and ignores complete explanations, mechanistic interpretability, on the other hand, seeks to explain each part of the model, including neurons, recurrent gates, attention head layers, and ultimately the entire network (Bereska & Gavves, 2025). Mechanistic interpretability provides a deeper understanding of the model, directly enhancing trust by allowing users to comprehend every component (Mueller et al., 2025). However, mechanistic interpretability is often achieved through extensive interventions, causal analyses, and techniques such as activation patching or weight modification. While these methods may produce accurate explanations, they require users to understand the model in a modified state rather than in its original logic upon which the decision was made. This can increase users' cognitive load and introduce a new layer of complexity, ironically making the process of explaining a black-box model more opaque rather than clearer.

To address the trustworthiness of such explanations, Palumbo et al. (2025) proposed an axiomatic procedure to validate mechanistic interpretations, formalizing properties that reliable interpretations should meet, such as consistency and completeness in mapping components to behaviors. The operation involves defining axioms for interpretations and testing them against model activations to verify if the proposed mechanisms accurately reflect the model's computations. The output of explainability is a validated set of interpretations that can be trusted for tasks such as debugging or safety checks in language models. This is an excellent way to ensure the quality of mechanistic explanations and build confidence in their use. However, it still relies on interventional methods to verify axioms, which involves modifying the model's state during validation and adds technical barriers for non-experts. Conmy et al. (2023) organize the mechanistic interpretability workflow, automate the most labor-intensive steps, such as circuit discovery, and design Automatic Circuit Discovery (ACDC), a pruning-based algorithm that reconstructs sparse subgraphs responsible for explaining specific model behaviors. The ACDC successfully rediscovers known circuits in GPT-2 Small, such as those for the Indirect Object Identification and Greater-Than tasks, demonstrating how automated patching methods can reproduce and scale manual interpretability efforts. While ACDC improves efficiency, it still relies on activation patching as its core mechanism, meaning the model is interpreted under modified conditions rather than its original operational logic. Consequently, despite its scalability, such interventional explanations may limit user trust by departing from the model's unaltered decision-making pathway.

To reduce the extent of direct intervention, Bricken et al. (2025) proposed a weak dictionary learning algorithm that uses sparse autoencoders to explain polysemantic neurons by learning interpretable, monosemantic features from standard transformer activations. The core idea is to train a sparse autoencoder on activation patterns from a shallow transformer, where the enforced sparsity enforces the model to discover distinct activation patterns associated with individual semantic concepts. As a result, the learned features show interpretable associations, such as specific activations for biological terms, such as DNA, or writing systems, such as Arabic script, thereby presenting an understanding of internal representations. This approach begins with a micro-level analysis of individual neurons and expands the interpretability outward. However, it remains dependent on a secondary

learning system, the autoencoder, which is itself a black-box model and thus introduces another layer of abstraction. Since the explanations are derived from this auxiliary network, overall transparency remains partial, as trust must be extended to a separate model whose internal logic is not entirely transparent. To further improve this concept, Williams et al. (2025) explained that the structural components of neural networks, such as neurons, weights, biases, attention heads, and convolutional filters, usually fail to map into user-understandable, meaningful concepts. Consequently, efforts to enhance interpretability and explainability tend to pursue abstract, coarse-grained decompositions that provide better insights into model behavior. However, the initial step in these explanations may involve investigating smaller functional units, such as neurons.

To maintain user trust while comprehensively unpacking the internal operations of neural networks, it is essential to explain the model at its original logic without introducing any change, such as perturbations, activation patching, or parameter modifications. As models grow in complexity to meet real-world performance requirements, the need for accessible and logically faithful explanations becomes essential (Bricken et al., 2025; Craver, 2007). In response to this challenge, we proposed Linear Lens, a fully non-interventional interpretability procedure that decomposes the neural network through behavioral observation, without adding synthetic signals or modifying the underlying learned parameters. Linear Lens begins by analyzing individual neurons as the smallest functional units and gradually extends this analysis across layers to explain the neural network's behavior against input features at its original logic. This is a multi-phase, layered algorithm that provides an inside-out explanation that remains aligned with the model's true operational behavior. Furthermore, we evaluate the comprehensibility of these explanations through real-world user studies, ensuring that the proposed method provides insights that are not only faithful to the model but also cognitively meaningful to human users.

2.4. Human-centric interpretability

Human-centric evaluations of explainability and interpretability are necessary to ensure that the human user understands and utilizes the provided explanation. The purpose of explainability ultimately serves human users, so its effectiveness cannot be determined solely by technical metrics. Instead, it must also reflect the extent to which end-users can comprehend and make decisions based on the explanations provided. This places the challenge of explanation quality not only in the technical domain but also within the scope of human cognitive science. Despite this, a considerable amount of the interpretability literature remains focused on algorithmic novelties, with limited attention paid to how human users perceive and interact with these explanations in practice. Nauta et al. (2023) stress a considerable imbalance, noting that only a small portion of the literature on explainable AI includes human evaluations. This deficiency affects the user's understanding of interpretability from a human factors perspective. Furthermore, as Poerner et al. (2018) emphasized, explanations must accurately reflect the model's actual decision-making process to prevent misleading users with ultimately inaccurate justifications.

Several studies have attempted to bridge this gap by conducting user studies, such as Ribeiro et al. (2016), who used Mechanical Turk to recruit participants to compare saliency maps across classifiers and measure their trust in neural network predictions. Hase and Bansal (2020) investigated how different explanation formats influence human ability to understand model output, showing that design choice can significantly affect user confidence and understanding. Yin et al. (2019) conducted large-scale experiments to assess perceived reliability and trust among general users. At the same time, Chandrasekaran et al. (2018) employed a human-in-the-loop algorithm to evaluate whether explanations enhance transparency in visual question answering systems. Further evaluations have spanned applications such as medical diagnostics (Oberste & Heinzl, 2023), natural language processing (Oberste & Heinzl, 2023), vision-based systems (Heimerl et al., 2022), and human–robot interaction (Sanneman & Shah, 2022).

While some interesting user studies have been conducted to assess user trust and understanding of explainability, they directly show the accuracy that cannot be predicted using any metrics, as users are recipients of explainability and efforts to make models transparent. However, existing studies often overlook the rigorous statistical validation of user studies, which is crucial to ensure that user opinions genuinely originate from reliable instruments. To achieve this, there are two major steps: instrument consistency and validation. Internal consistency that refers to how uniformly items within a questionnaire measure the same underlying concept, typically assessed using Cronbach's Alpha (Som et al., 2017). However, this metric assumes equal item contributions, which may not hold in diverse cognitive settings. In contrast, McDonald's Omega is an alternative that offers a more flexible reliability estimate, accounting for item variability. Instrument validation is achieved through Confirmatory Factor Analysis, which is essential for confirming whether the structure of user responses aligns with the intended conceptual framework (Aziz et al., 2024; Gehrmann et al., 2020; Rong et al., 2024).

Moreover, in the context of Linear Lens, we emphasize that interpretability is not a one-size-fits-all endeavor, but rather that understanding different explanations, such as low-level neuron behaviors and high-level network-wide logic, requires varying cognitive loads and mental efforts from users. Many existing questionnaires also fail to account for these complex mental and cognitive processes, potentially ignoring an essential aspect of human factors in AI, which ensures the clarity or utility of explanations is measured. To address this, we deployed a statistically validated user study that accounts for distinct cognitive loads across explanation levels, ensuring that explanations produced by Linear Lens remain not only technically faithful but also cognitively accessible to a diverse range of users.

3. Linear lens

This section will describe data and variable representation, followed by four phases of interpretability.

3.1. Data and variable representation

Let the DNN f take input features x, apply its learned parameters, and deliver an output or decision $\hat{y}$. The shape of x and $\hat{y}$ depends on the learning task. In time-series scenarios, the input is denoted as $x \in \mathbb{R}^{m\times d\times t}$, where m is the number of samples, d the number of features, and t the number of time steps. The corresponding output for time-series tasks is denoted $\hat{y} \in \mathbb{R}^{m\times o\times t'}$, where o is the number of output targets and t' the number of predicted steps. For tabular (non-temporal) tasks, the input simplifies to $x \in \mathbb{R}^{m\times d}$. For tabular tasks, the output takes the form $\hat{y} \in \mathbb{R}^{m\times o}$, which may represent binary classifications or regression targets. Moreover, i indexes neurons n within a given layer ($i = 1, \ldots, n$), j indexes input features ($j = 1, \ldots, d$), and k indexes individual data samples ($k = 1, \ldots, m$).

This work aims to uncover the inner workings of model f by analyzing its behavior and how individual neurons contribute to understanding the entire network. The following subsections outline how the model can be decomposed into interpretable components. Our goal is to investigate the model's decision-making in its unaltered state. We aim to design a principled interpretability procedure that breaks down the model's behavior layer by layer and neuron by neuron, aligning with human reasoning and respecting the natural structure of the underlying task.

3.2. Phase 1: Discovery of single neuron behavior

In neural networks, a neuron serves as the smallest fundamental unit that transforms inputs into outputs using learnable parameters such as weights and biases, along with an activation function such as ReLU or Sigmoid. Even if a neural network layer is constructed without an activation function, it still contains neurons in computation sense. Let us assume $W \in \mathbb{R}^{n\times d}$ is the weight matrix and $b \in \mathbb{R}^n$ is the bias vector, and both are learnable by standard, where n is the number of neurons in layer. The neuron is single layer neural network can be formulated as:

$$h = \phi(Wx + b) \tag{1}$$

Here ϕ is an activation function that applies non-linearity element-wise, activation function depends upon learning task and specific need of design. The choice of activation function depends on the learning task and specific design requirements. Linear Lens analyzes neuron behavior at the pre-activation stage in order to capture the direct contribution of input features before the non-linear transformation is applied. Activation functions such as ReLU subsequently transform these values and may suppress negative responses, which correspond to inactive neurons. In our framework, such neurons are naturally reflected through low or diffuse influence patterns and are identified during the entropy-based neuron categorization process. Each neuron at specific index i is defined by the pair $(W_{i:}, b_i)$, where $W_{i:}$ denotes the ith row of the weight matrix and b_i is the corresponding bias. The activation of neuron i for input x is given by:

$$n_i = \phi(W_{i:}x + b_i) \tag{2}$$

Fodor's principle of compositionality (Fodor & Pylyshyn, 1988) states that the meaning of a complex expression is determined by the meanings of its parts and the way they are combined. Drawing direct inspiration from this theory, we argue that to understand a complex neural network, one must first clearly understand the role of its smallest functional unit, the neuron, which can in turn help explain the behavior of layers and, ultimately, the entire network. To establish understanding, we categorize neurons into three major types. `Monosemantic neurons` are those that activate in response to a single feature or concept, making them relatively easy to interpret. The second type is `Polysemantic neurons`, which activate in response to multiple features or concepts, making them difficult to explain, as each polysemantic neuron may respond to a different combination of features across instances. The third type is `dead neurons`, which have minimal activation. However, "dead" does not mean they have no role in the learning process. Even with very low activation, they can still influence the model's decisions.

To locate these neurons, we propose a Hypothesis-Driven Neuron Discovery method, which is a straightforward but validated procedure for identifying neurons within a neural network layer directly, without relying on perturbations or activation patching. Before classifying neuron types, it is essential to understand the root of the black-box nature of neural networks. In fact, before applying non-linearity, a neural network layer operates in a completely transparent manner where each input feature is linearly multiplied by its corresponding weight, and a bias term may be added according to the layer's configuration. Considering a single input instance x_i, the pre-activation for neuron i is computed as:

$$p_i = W_i x_i + b_i \tag{3}$$

In Eq. (3), the value p_i denotes the pre-activation, where no non-linearity has been applied. At this stage, the system is not a black box because the pre-activations are fully accessible and can provide direct insights into neuron behavior. Eq. (3) produces the scalar pre-activation of a single neuron. To illustrate the relative feature-wise influence patterns for interpretability, we consider the neuron's pre-activation values before summation. Each element of the example vectors p_{i1} and p_{i2} corresponds to an input feature multiplied by its corresponding weight: To illustrate this intuitively, we use example feature-wise influence patterns for a neuron, written here as ordered values for explanatory purposes only:

$$p_{i1} = [0.90,\ 0.10,\ 0.30,\ 0.40,\ 0.10] \tag{4}$$

$$p_{i2} = [2.10,\ 1.80,\ 0.40,\ 0.50,\ 1.50] \tag{5}$$

By analyzing Eq. (4), one can infer that the first feature plays a dominant role, as its pre-activation value (0.90) surpasses those of other features. This suggests that a single input feature has a primary influence on the neuron. In contrast, Eq. (5) indicates a more distributed influence, where multiple features contribute comparably, suggesting a neuron influenced by several components. This behavioral access enables neuron classification even before applying any non-linear transformation, and without relying on perturbations or activation patching. However, this still requires a formal guarantee and mathematical validation before proceeding.

Furthermore, when applying activation functions such as ReLU or Sigmoid to the pre-activations in Eq. (4), the dominance of the first feature remains evident. For instance, if the first pre-activation value is 0.93 while the others are smaller, the post-activation will still reflect this relative importance. Although the post-activation is a single scalar and cannot be directly decomposed, this indicates that the most dominant pre-activation contributes more prominently even after non-linearity is applied. Similarly, in Eq. (5), the absence of a clear single peak ensures that no single feature dominates the post-activation output, and several features contribute meaningfully to producing the post-activation output. This persistence is a direct consequence of a fundamental mathematical monotonicity property of many activation functions, such as ReLU, PReLU, Tanh, and Sigmoid, which are all monotonic increasing functions. Formally, for any two values $\pi_1 > \pi_2$, it follows that $\phi(\pi_1) > \phi(\pi_2)$, where ϕ denotes either ReLU or Sigmoid. ReLU is defined as:

$$\phi(\pi) = \mathrm{ReLU}(\pi) = \max(0, \pi) \tag{6}$$

Sigmoid is defined as:

$$\phi(\pi) = \sigma(\pi) = \frac{1}{1 + e^{-\pi}} \tag{7}$$

This means:

$$\text{If } \pi_1 > \pi_2, \text{ then } \phi(\pi_1) > \phi(\pi_2) \tag{8}$$

Eq. (8) formalizes the fact that monotonic activation functions preserve the ordering of their inputs. In other words, if one feature contributes more strongly than another at the pre-activation stage, it will continue to contribute more strongly after the activation function is applied. This property underpins our use of the normalized influence vector π for reliable neuron classification, since the relative strengths reflected in π remain valid even in the presence of non-linearities. The same logic will be applied to other activation functions, such as Tanh and PReLU. Therefore, the analysis of pre-activations provides reliable guidance for understanding neuron behavior. We quantify each feature's contribution to neuron i by averaging the absolute product of input entries $x_{k,j}$ and weight entries $W_{i,j}$ over the m samples:

$$\mu_{j,i} = \frac{1}{m}\sum_{k=1}^{m}\left|x_{k,j}\, W_{i,j}\right|. \tag{9}$$

These raw influence values are then normalized to form a probability distribution over the d features for neuron i:

$$\pi_{i,j} = \frac{\mu_{j,i}}{\sum_{j'=1}^{d}\mu_{j',i}}, \tag{10}$$

ensuring $\sum_{j=1}^{d}\pi_{i,j} = 1$. In Eqs. (9) and (10), the index j refers to the jth input feature contributing to neuron i. Thus, $\pi_{i,j}$ denotes the normalized influence of feature j on neuron i, and is written in a probability-style

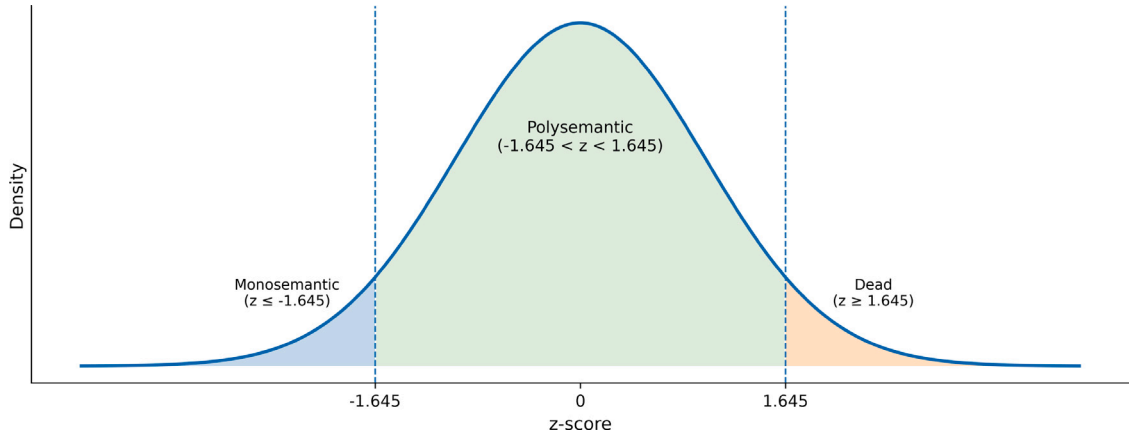

Fig. 1. Z-score interpretation of neuron entropy for role assignment under the proposed entropy-based hypothesis testing, where low-entropy neurons fall in the monosemantic region ($z < -1.645$), high-entropy neurons are labeled dead/flat ($z > 1.645$), and intermediate values are treated as polysemantic ($-1.645 \leq z \leq 1.645$).

notation to represent the feature influence distribution. For Neuron 0, our supposed example gave $\mu_{:,0} = [2.1,\ 1.8,\ 0.4,\ 0.5,\ 1.5]$, which normalizes to $\pi_0 = [0.33,\ 0.28,\ 0.06,\ 0.07,\ 0.23]$. In this case, no single entry exceeds 0.8, and several entries are substantial, hence Neuron 0 is classified as polysemantic. By contrast, a normalized vector such as $\pi = [0.90,\ 0.03,\ \ldots]$ would indicate a monosemantic neuron focused on one feature, while a vector with all entries uniformly small (for example, each below 0.1) would denote a dead neuron. These are illustrative examples and do not correspond to any real data, intended to enhance readers' understanding.

Although the normalized pre-activation scores $\pi_{i,j}$ in Eq. (10) already present intuitive insight into neuron types, such as identifying dominant features, this interpretation remains informal. The influence of learnable weights can distort these patterns, making them insufficient for robust classification. To address this, we introduce a three-step hypothesis testing that formalizes the discovery of neuron types. The three-step hypothesis testing includes entropy computation and statistical z-scoring to provide a principled and validated method for identifying monosemantic, polysemantic, and dead neurons. To compute entropy e_i, we apply the following formula over the normalized pre-activation vector $\pi_{i,:}$:

$$e_i = -\sum_{j=1}^{d} \pi_{i,j} \log(\pi_{i,j}) \tag{11}$$

This entropy value measures how focused or dispersed the neuron's activation pattern is, where a low entropy indicates concentration on one input, ultimately resulting in low uncertainty, and this neuron will be (`monosemantic`). If the distribution is uniform with maximum uncertainty, this means no feature stands out. The neuron has no clear functional role. That is labeling such neurons as (`dead`). If the pattern is focused on more than one, the neuron responds to a combination of features, and we call this (`polysemantic`). To standardize entropy across all neurons, we compute the Z-score:

$$z_i = \frac{e_i - \mu_e}{\sigma_e} \tag{12}$$

where μ_e and σ_e denote the mean and standard deviation of the entropy values across all neurons in the layer. Neuron classification is then computed based on the Z-score as follows:

$$\begin{aligned} z_i < -1.645 &\quad \Rightarrow \quad \texttt{Monosemantic} \\ -1.645 \leq z_i \leq 1.645 &\quad \Rightarrow \quad \texttt{Polysemantic} \\ z_i > 1.645 &\quad \Rightarrow \quad \texttt{Dead} \end{aligned} \tag{13}$$

These thresholds are theoretically motivated and are used as a standardized, layer-relative statistical decision rule based on one-tailed hypothesis testing under the standard normal distribution, where ± 1.645 corresponds to a 90% confidence interval. This allows us to distinguish neurons with unusually low entropy (monosemantic) or unusually high entropy (dead) from the typical entropy range (polysemantic) within the same layer. Fig. 1 summarizes the entropy z-score decision regions used to classify neurons into monosemantic, polysemantic, and dead/flat functional roles. These thresholds should therefore be interpreted as layer-relative criteria rather than absolute universal cutoffs.

The label `dead` does not always mean that a neuron has no influence or no functional relevance. Rather, it denotes a neuron whose normalized influence distribution is highly diffuse and therefore does not show a clear dominant semantic association under the observed data distribution. This interpretation is consistent with Shannon entropy, where higher entropy reflects greater uncertainty or dispersion in a probability distribution (Shannon, 1948). In our energy load forecasting setting, many signals naturally vary in upward and smoothly distributed consumption patterns, so a diffuse entropy profile should be understood as weak semantic concentration in the present dataset, not as absolute irrelevance of the neuron. Moreover, the label `dead` does not necessarily imply that a neuron is completely inactive in the traditional training sense. In our application-driven setting, particularly in energy forecasting where signals tend to follow smooth upward consumption patterns, the element-wise input–weight interactions often produce diffuse influence distributions. Therefore, the term `dead` is used heuristically to denote neurons whose normalized influence distribution does not exhibit a clear semantic concentration across input features under the observed data. Activation functions such as ReLU or Sigmoid are applied after the pre-activation stage analyzed by Linear Lens. While monotonic activations often preserve relative ordering in active regions, they may also flatten or suppress values in some regimes (for example, negative inputs under ReLU). Therefore, our interpretability analysis is based in pre-activations, where feature contributions remain directly observable.

3.3. Phase 2: Layer-level interpretation

This subsection will describe how polysemantic neurons can be explicitly explained in terms of the exact features to which they belong, followed by our proposed Qualitative Symbolic Matrix (QSM) for describing the entire layer in a human-understandable manner.

3.3.1. Polysemantic neurons

Monosemantic neurons are relatively more straightforward to interpret because they respond to one feature and thus represent a single concept. In contrast, polysemantic neurons are activated by multiple input features simultaneously, which makes them more complex to

explain, as their behavior arises from a combination of influences rather than a single source. To interpret such neurons, it is necessary to explore which features jointly shape their behavior. Since we are observing linearly, as in Eq. (10), it provides a direct way to identify these contributing features. It preserves the feature ordering and quantifies the relative influence of each input on the neuron through the normalized pre-activation vector $\pi_{i,:}$. For polysemantic neurons, this vector typically contains several moderate-to-high values, indicating that no single feature dominates but that several features contribute meaningfully. This allows us to track which features are playing dominant, suppressed, or jointly active roles in influencing the neuron.

However, simply observing the influence values in Eq. (10) is not sufficient to confirm whether these features genuinely explain the neuron's activation. A validation is required to ensure that the neuron's behavior is truly governed by the identified set of features. To achieve this, we apply a regression-based validation that aligns with the `non-interventional philosophy` of our proposed method, as it does not modify anything. For each neuron classified as polysemantic, we select the top-k contributing features based on their values in $\pi_{i,:}$, such as those exceeding a fixed threshold (e.g., $\pi_{i,j} > 0.15$) and have proved polysemantic as in Eq. (12). We then fit a linear regression model using only these selected features to predict the neuron's pre-activation value p_i. The regression model is defined as:

$$\hat{p}_i = \beta_0 + \sum_{j \in S_i} \beta_j x_j + \epsilon \tag{14}$$

where S_i denotes the set of top contributing feature indices for neuron i, β_j are the learned coefficients, and ϵ is the residual error. The coefficient of determination R^2 is computed to assess how well these selected features collectively explain the pre-activation. A high R^2 score indicates that the selected features not only correlate with the neuron's activation but also functionally drive it. In such cases, the neuron can be confirmed as polysemantic with clear grounding in multiple input features. If the R^2 score is low, this suggests that the neuron's behavior depends on additional features not captured in $\pi_{i,:}$ or on nonlinear interactions, which would require further investigation in later phases.

This dual procedure, analysis and validation, ensures that polysemantic neurons are not only identified based on distributional patterns, but also verified through their actual response to the most influential features. This approach provides a transparent and practical pathway to understanding the composite roles of polysemantic neurons, setting the foundation for a deeper interpretation of the layers and network structures they support.

The regression analysis is used as a behavioral validation step to verify whether the subset of influential features identified through the normalized influence distribution can collectively explain the neuron's observed activation pattern under real data. This procedure does not attempt to reconstruct the neuron's internal computation, but instead evaluates whether the identified features provide a consistent explanatory subset for the neuron's behavior.

3.3.2. Explaining layer

Now we have identified and validated individual neuron behaviors such as monosemantic, polysemantic, and dead. This subsection synthesizes these insights to present a clear explanation of the behavior of the entire layer. Since a layer is composed of neurons, understanding each neuron's functional role naturally enables us to understand the layer's collective transformation. However, rather than aggregating neurons, we propose a interpretability matrix: the QUALITATIVE SYMBOLIC MATRIX (QSM).

The QSM is essentially a qualitative symbolic matrix that presents the functional behavior of all neurons in the entire layer in a single qualitative matrix. This is not a statistical or mathematical matrix; instead, `it is a symbolic matrix`. Similar to a confusion matrix in classification that displays categorical outcomes, this matrix serves as a categorical map of neuron behavior. Its purpose is to present complex neuronal activity in a manner that is comprehensible to humans, rather than for computational. Formally, for a layer of n neurons and d inputs, QSM is defined as

$$\mathbf{M} \in \mathbb{R}^{n \times d} \quad \text{with} \quad M_{ij} = \pi_{i,j}, \tag{15}$$

Each row of $\mathbf{M}$ represents the role of neurons, categorized as monosemantic, polysemantic, or dead units, while each column reflects the influence of specific features. We illustrate the QSM for a single layer of a neural network comprising $n = 6$ neurons (n_1 to n_6) and $d = 5$ features (x_1 to x_5). To explain the layer, we employ three distinct symbols.

★ (moderate), ★★ (strong), ★★★ (very strong), ∘ (weak).

$\mathbf{M} =$

	x_1	x_2	x_3	x_4	x_5
n_1	★★★	∘	∘	∘	∘
n_2	∘	★★★	∘	∘	∘
n_3	★★★	★★	★	∘	∘
n_4	★	★★	★	★	∘
n_5	∘	∘	∘	∘	∘
n_6	∘	∘	∘	∘	∘

In this context, $\mathbf{n_1}$ and $\mathbf{n_2}$ are monosemantic, each strongly responding (★ ★ ★) to a single feature x_1 and x_2 respectively. In contrast, $\mathbf{n_3}$ and $\mathbf{n_4}$ are polysemantic, as two or three features to a from very strong (★ ★ ★) to a moderate degree ★). Meanwhile, $\mathbf{n_5}$ and $\mathbf{n_6}$ are classified as dead, showing uniformly low influence across all inputs (∘). The QSM provides an at-a-glance `fingerprint` of the entire layer, converting detailed neuron-level analyses into a clear, actionable summary that would have been very challenging to understand in numerical form.

The proposed QSM presents several compelling advantages. First, it transforms neuron-level behaviors into an intuitive and interpretable matrix, enabling direct visualization of how the entire layer interacts with input features. This abstraction is valuable in scenarios where numeric weights and activations are too dense or ambiguous to interpret meaningfully. With a unified view, the user can understand which feature or multiple features cause a neuron to function. Second, by preserving the categorization of neurons as monosemantic, polysemantic, or dead, QSM provides behavioral-level explainability rather than relying solely on mathematical gradients or saliency maps. Third, it allows practitioners to trace specific input features to their influence targets within the network architecture, which is crucial for debugging, aligning models with domain knowledge, and improving trust in AI systems. QSM facilitates cross-model or cross-layer comparisons, making it a valuable tool for analyzing representational analysis and redundancy within deep networks. QSM also serves as a structured, human-understandable interface for explaining how a neural layer transforms inputs into learned internal representations.

3.4. Phase 3: Deeper layers explainability

While the neurons in the first layer can be directly interpreted in relation to the input features, the same approach does not apply to neurons in deeper layers. This is because hidden-layer neurons do not receive raw input features; instead, they operate on the activations from the previous layer. As a result, they cannot be labeled as monosemantic or polysemantic in the same way. To explain their behavior, one must analyze how they are activated in response to earlier layer activations and trace these activations back to their original input features. Despite this complexity, the behavior of hidden-layer neurons can still be understood by mapping them to the upstream neurons that cause their activation. Before moving to explain neurons' role, it is essential to note that neurons in subsequent layers are usually not the same

as the previous layer, as it can be expansion, where the number of neurons increases in the next layer, or compression, where the number of neurons decreases, or even it can be In equal-width layers where the number of neurons remains the same. So, each care requires careful understanding of how neurons' roles are shaped in deeper layers, a complex situation that is also influenced by the design of the neural network.

Although Phase 3 addresses deeper layers that receive inputs from previous activations instead of raw features, the computational logic remains consistent with Phase 1. For each neuron in deeper layers, we compute influence scores based on upstream activations using the same formulation as in Eqs. (9) and (10), followed by entropy computation Eq. (11) and z-score normalization Eq. (12). The only distinction lies in the input source: whereas Phase 1 uses raw input features, Phase 3 uses neuron activations from the preceding layer. The reuse of the same pipeline ensures uniformity and comparability across layers, while enabling recursive interpretability tracing from outputs back to original inputs through a chain of neuron behaviors.

Regardless of the network's design, fully connected layers take all activations from the previous layer as input. As a result, the behavior of a neuron in any hidden layer must be interpreted through the roles of neurons from the previous layer. This layered mapping enables us to trace influence back to the original input features, since the previous layers have already been explained in Phases 1 and 2. When a single upstream monosemantic neuron dominates the influence on a current neuron, we hereafter refer to the current neuron as a `unimodal neuron`. Its behavior can be clearly traced to one specific input feature. One can examine exactly which monosemantic type causes its activation. For example, if a neuron is activated by a monosemantic neuron that itself responds to the input feature temperature, then its behavior can be directly mapped to temperature, and its role becomes interpretable. For example, if a neuron in Layer 2 draws 85% of its activation from a Layer 1 neuron known to fire only for temperature, then we can trace its behavior directly to temperature and label it as a `unimodal neuron`. The below given arrowed mapping makes its role fully transparent and immediately understandable.

Input (e.g., `temperature`) $\longrightarrow$ Layer 1 `monosemantic` $\longrightarrow$ Layer 2 `unimodal neuron`

In contrast, when a neuron receives substantial contributions from several neurons from the previous layer, regardless of whether those are monosemantic or polysemantic, we refer to it as a `multimodal neuron`. Its behavior indicates a genuine combination of different feature signals, rather than a single dominant source; therefore, it is the most complex neuron to explain, but it is still possible, as we have tracked exactly the features that influence polysemantic and monosemantic neurons in the first layer. For instance, suppose a Layer 2 neuron is activated by several activations that show both monosemantic and polysemantic behavior; we can still track down its behavior, as shown in the circuit below.

$\{$ `temperature` $\rightarrow$ L1 `monosemantic`, `hour` $\rightarrow$ L1 `monosemantic`, `energy, hour, day of year` $\rightarrow$ L1 `polysemantic` $\} \implies$ L2 `multimodal neuron`

If no upstream neuron contributes meaningfully or if the overall influence is diffuse, the neuron is considered a `Muted Neuron`. This contrasts with dead neurons in the first layer, as these have no direct relation to input features. This labeling ensures that each neuron, even in deeper layers, has a defined role based on its dependencies, completing the interpretability chain across the full network.

As we move to even deeper layers, such as from Layer 2 to Layer 3, the interpretability chain can still be extended using the same principles. For example, suppose a neuron in Layer 3 receives a dominant contribution from the `unimodal neuron` in Layer 2 that was already mapped to temperature. In that case, this Layer 3 neuron also inherits the exact semantic alignment and is again labeled as a `unimodal neuron`. However, suppose it is influenced by both the Layer 2 `unimodal neuron` (mapped to temperature) and another `multimodal neuron` (combining signals from hour and energy). In that case, the Layer 3 neuron must be labeled as a `multimodal neuron`. Its behavior results from multiple semantic traces, each of which has already been explained in the previous layers. This recursive traceability ensures that semantic meaning does not vanish in deeper layers, but instead accumulates and transforms through a clear compositional logic. For example, the following representation shows how the influences from previously explained neurons propagate forward:

$\{$L2 `unimodal (temperature)`, L2 `multimodal (hour + energy)`$\}$ $\implies$ L3 `multimodal neuron`

This continuation maintains the chain of interpretability, where each neuron's role is not only defined by upstream behavior but also based on previously identified semantic mappings.

In the final layer, neurons no longer pass their activations forward but instead contribute directly to the model's decision or forecast. Despite this change in role, the same semantic traceability applies. If a final-layer neuron receives most of its input from a `unimodal neuron` in the preceding layer that was earlier traced back to a specific feature such as temperature, we can confidently assign the output behavior to that feature. For instance, if the forecast value is driven predominantly by a path that begins with temperature, passes through a monosemantic neuron in Layer 1, then a unimodal neuron in Layer 2, and finally dominates the final-layer neuron, the forecast itself can be semantically interpreted as temperature-driven.

On the other hand, if the final output neuron aggregates substantial input from several `multimodal neurons` in the previous layer, then the prediction results from a confluence of several features — such as energy, hour, and day of year — intertwined through previous polysemantic and monosemantic activations. This leads us to define such final-layer neurons as `compositional neurons`, representing a combination of multiple semantic influences. While more abstract, their logic remains interpretable because each upstream influence has already been semantically grounded in earlier phases. For example:

$\{$L2 `unimodal (temperature)`, L2 `multimodal (energy + hour)`, L2 `unimodal (day of year)`$\}$ $\implies$ Final Output `compositional neuron (forecast)`

This design of traceability allows the model's forecast to be decomposed not just into numeric weights, but into meaningful semantic components, enabling both diagnostic clarity and user trust. The QSM can present explanations of neurons across layers that are easily understood by human users, as shown below, where four neurons journey from inputs to the forecast, provided for illustrative purposes. We use four symbols to indicate pathway strength: ★★★ denotes very strong influence (approximately 60%), ★★ denotes strong influence (approximately 30%), ★ denotes moderate influence (approximately 10%), and ∘ denotes negligible influence. Here, f_1 refers to the first forecasting output (e.g., predicted energy at $t+1$), and the influence score indicates the strength of contribution each input feature makes toward forecasting f_1, as interpreted through its pathway across neuron roles in layers such as L1, L2, and L3. This is only for enhancing the understandability of readers.

Feature	Layer 1	Layer 2	Layer 3	f_1 influence
`temperature`	n_1(monosemantic)	n_5(unimodal)	n_8(unimodal)	★★★
`hour`	n_2(monosemantic)	n_6(multimodal)	n_9(multimodal)	★★
`energy + day of year`	n_3(polysemantic)	n_7(multimodal)	n_9(multimodal)	★
`unused feature`	n_4(dead)	n_{10}(muted)	n_{11}(muted)	∘

The tracing of neuron influence across layers follows the same influence aggregation and entropy-based characterization pipeline used in Phase 1. For deeper layers, upstream neuron activations are treated as inputs, and the same normalization, entropy computation, and z-score

Table 1
Questionnaire items for user study for evaluating linear lens.

ID	Dimension	Question Text
Cognitive Load Theory (CLT)		
CLT1	Intrinsic	The explanation provided was inherently complex.
CLT2	Extraneous	The way the explanation was presented made it harder to understand.
CLT3	Germane	The explanation helped me gain a deeper understanding of the AI System.
Trust & Usability of Explanations (TUE)		
TUE1	Trust	I trust the explanation given by the AI system.
TUE2	Usability	The explanation was easy to understand.
Comprehensibility of Explanations (CoE)		
CoE1	Clarity	The explanation was clear and easy to follow.
CoE2	Coherence	The explanation was logically organized.
Actionability of Explanations (AoE)		
AoE1	Practicality	The explanation was practical and useful in my context.
AoE2	Decision	The explanation helped me make an informed decision.

classification are applied. This ensures that compositional influence paths are computed systematically rather than selected illustratively. The computational steps of Linear Lens consist primarily of influence aggregation and normalization over neuron weights and observed activations. Because these operations are applied to a trained model without retraining, perturbation, or iterative optimization, the procedure scales approximately linearly with the number of neurons and input features and can be applied independently across layers or architectural components.

3.5. Phase 4: User study and human cognitive understanding

To assess whether users and operators comprehend and trust the Linear Lens explanations, a questionnaire was deployed based on four primary constructs: Cognitive Load Theory (CLT), Trust and Usability of Explanations (TUE), Comprehensibility of Explanations (CoE), and Actionability of Explanations (AoE). This questionnaire itself has been reused from our previous study on explainability. The items against four constructs have been represented in Table 1. Each construct has been applied across explanations from Phases 1 to 3 and assessed using a 5-point Likert scale (1 = strongly disagree; 5 = strongly agree). The design, informed by cognitive load principles, enhances the interpretability of complex model behavior in a way that prioritizes the human experience.

First, we evaluate Intrinsic Load, which pertains to the inherent difficulty of the idea itself. This type of load shows the natural complexity of what the user seeks to understand, making it essential for assessing their comprehension of complex explanations. In our context, understanding how a single neuron in a neural network responds to various input features can be inherently challenging, even when presented clearly. Then, we analyze Extraneous Load, which examines the quality of the explanation's presentation. This load emphasizes the additional difficulties that arise from the manner in which the explanation is conveyed. Unlike intrinsic load, extraneous load focuses on how the idea is depicted rather than the complexity of the idea itself. For instance, if a graph lacks proper labeling or clear terminology, users may find it difficult to interpret. Evaluating extraneous load is essential for determining design flaws that unnecessarily complicate explanations. We also assess Germane Load, which measures the extent of user learning derived from the explanation. This represents the productive mental effort that aids users in understanding the concept and forming a clear mental model. Germane load is the valuable kind of effort; for example, when a visual clearly indicates, "this neuron activates when the temperature is high", and the user deduces, "Oh, I understand now, this neuron monitors temperature", it illustrates effective germane load.

In the evaluation of the TUE construct, we directly assessed participants' trust in the explanations, as well as the overall usability, by assessing perceptions of reliability, accuracy, and applicability. For the CoE, we included straightforward items to evaluate the clarity and coherence of the explanations, ensuring they were logically structured and easy to understand. Regarding the AoE, we examined the practical relevance of the presented information and whether the explanations facilitated informed decision-making. To enhance fairness and diversity, we gathered demographic information such as age, gender, education level, and country of residence. We also asked participants to self-report their English proficiency (using a binary response) and their initial level of trust in explainable AI (using a Likert scale), as summarized in Table 2.

Table 2
Participant demographics (N = 400).

Characteristic	Percentage
Age Group	
18–24	17%
25–34	40%
35–44	21%
45+	22%
Gender Identity	
Male	51%
Female	47%
Non-binary/Prefer not to say	2%
Education Level	
High school or lower	12%
Bachelor's degree	51%
Master's degree	35%
Doctorate or higher	2%
Country of Residence	
Canada	13%
United States	11%
Pakistan	9%
Other	67%
English Reading Proficiency	
Yes (self-reported English reading proficiency)	100%
Initial Trust in XAI Explanations	
Agree or Strongly Agree (trust influence)	64%
Neutral or Disagree	36%

4. Evaluation

This section describes the datasets, preprocessing steps, model implementation and interpretability results for Phases 1 to 3. The results from the user study are presented alongside the corresponding phases. The section concludes with a controlled synthetic experiment to further confirm interpretability in a scenario with known ground truth.

4.1. Datasets, preprocessing, and model implementation

The evaluation employed ten real-world energy consumption datasets from three primary consumer classes: student residences and individual homes, as well as industrial and commercial buildings (Danish & Grolinger, 2024). Table 3 provides an overview of these datasets,

Table 3
Description of energy consumption datasets.

Dataset	Dates	Short description
Student Residences		
Residence 1	Jan/2019–Jul/2023	A suite-style residence featuring shared kitchen facilities
Residence 2	Jan/2019–Jul/2023	A suite-style residence without kitchen amenities
Individual Houses		
House 1	Jan/2002–Dec/2004	A detached single-family home showing complex and varied energy consumption patterns
House 2	Mar/2021–Aug/2021	A two-bedroom detached house with standard occupancy
House 3	Mar/2021–Aug/2021	A two-bedroom detached house equipped with an electric vehicle charging setup
House 4	Mar/2021–Aug/2021	A three-bedroom townhouse residence
Industrial and Commercial		
Manufacturing	Jan/2016–Dec/2017	An industrial manufacturing facility
Medical Clinic	Jan/2016–Dec/2017	A combined medical and wellness clinic
Retail Store	Jan/2016–Dec/2017	A commercial retail establishment
Office	Jan/2016–Dec/2017	A dedicated office building with multiple workspaces

detailing the time frames of data collection along with brief descriptions of each dataset. There are considerable variations among the buildings within each category. Residence 1 features suite-style accommodations with a kitchen, while Residence 2 also offers suite-style living but lacks a kitchen. Both residences house over 400 students.

Although all the homes under examination are located in London, Ontario, Canada, there is considerable diversity among them. Homes 1, 2, and 3 are all detached properties; however, Home 3 is distinctive due to the presence of an electric vehicle, which leads to notable fluctuations in energy consumption resulting from at-home charging. In contrast, Home 4 is a 3-bedroom townhouse, and its energy consumption differs from that of the detached homes due to the influence of adjacent units. To further explore a range of non-residential energy consumers, a manufacturing building, a medical clinic, a retail store, and an office building are also included in the analysis (Miller et al.).

Each dataset included a recording date and time along with corresponding hourly energy consumption. From this date and time, we extracted several additional features, such as the day of the year, day of the month, day of the week, and hour of the day, to help model seasonal, weekly, and daily patterns. To assess weather influences and enhance prediction accuracy, hourly temperature was also included, and other relevant features can be added if available. To improve convergence and mitigate the impact of larger features, the data were normalized using Min–Max scaling.

Each dataset was segmented into training, validation, and test sets using a 60%-20%–20% ratio. Given the temporal nature of the data, it was organized for the models through a sliding window technique, featuring a window length of 24 and a stride of 1. All models utilize the previous 24 h of five key features, including energy consumption, to forecast the consumption for the following 24 h. This forecasting period was chosen because energy operations typically depend on next-day forecasts for effective energy planning.

The model evaluation was performed using three metrics commonly utilized in consumer energy forecasting: Mean Absolute Error (MAE), which quantifies the average absolute difference between predicted and actual values (Fekri et al., 2023); Root Mean Square Error (RMSE), which indicates the standard deviation of the prediction errors (residuals) (Fekri et al., 2023; L'Heureux et al., 2022); and Symmetric Mean Absolute Percentage Error (SMAPE), which presents the forecasting error as a percentage. This facilitates straightforward interpretation and allows for comparisons across different datasets (Fekri et al., 2023). SMAPE was preferred over Mean Absolute Percentage Error (MAPE) because MAPE is skewed toward larger values and becomes undefined when actual values are zero. The calculation for the SMAPE metric is as follows:

$$\text{SMAPE} = 100\% \times \frac{1}{m}\sum_{i=1}^{m}\frac{2|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|}, \tag{16}$$

where y_i and $\hat{y}_i$ are the actual and predicted energy consumption values, respectively, and m is the number of samples. The models for all datasets were implemented using PyTorch and executed on an AMD Ryzen Threadripper PRO 5955WX processor paired with an NVIDIA RTX A6000 GPU.

We implemented three standard models, MLP, LSTM, and Transformer, across varying hidden unit sizes (64, 128, 256, and 512) to ensure compatibility with the Linear Lens framework. This range of neuron counts enables robust evaluation of the Linear Lens method across diverse network configurations. Since the primary objective of this work is not to enhance forecasting accuracy but to study explainability, we report high-level results without stressing for performance improvement. The results are summarized in Table 4. Notably, datasets from student residences, offices, medical clinics, and manufacturing buildings show SMAPE values below 15%. This is due to the regular, consistent energy use patterns of students and employees in institutional and industrial settings. In contrast, Individual households (House 1 to House 4) show higher SMAPE values, exceeding 30% and, in some cases, reaching above 50%. These elevated error rates indicate variability in personal energy-use behavior among individuals, which is challenging for models to forecast. Since the purpose of this work is explainability and interpretability, we refer readers interested in forecasting performance to our previous work, HyperEnergy (Danish & Grolinger, 2024), which uses the same datasets and focuses on optimizing predictive accuracy.

The inclusion of MLPs, LSTMs, and Transformers in the evaluation is intended to demonstrate the applicability of Linear Lens across different neural architectures rather than to establish strict one-to-one comparability between architecture-specific components such as dense neurons, recurrent gates, and attention submodules. Accordingly, the analysis focuses on selected representative components within each architecture that directly participate in the forward computation. The same influence aggregation, normalization, entropy-based categorization, and tracing logic is then applied within each architectural context, allowing consistent interpretability analysis while respecting architectural differences.

4.2. Results of phase 1

In Phase 1, we emphasize understanding how individual neurons behave by examining the influence of each input feature on their activation. This phase helped us classify neurons as monosemantic (responding to a single feature), polysemantic (responding to multiple features), or non-responsive (showing no response). This classification is a first step toward opening the "black box" and interpreting what roles each part of the network plays. This concept aligns with the idea of mechanistic interpretability, where a complex network is broken down into smaller, more comprehensible parts. We present selected examples of these neuron behaviors in Table 5 and Fig. 2. These examples are chosen from various models and datasets to express diversity. To apply this idea concretely, we investigated neurons in the first layer of various models, including the first layer of MLP, the LSTM input

Table 4
Forecasting performance across all datasets.

Dataset	Model	Neurons	MAE	RMSE	SMAPE
Residence 1	MLP	64	29.63	39.21	12.28%
		128	30.12	40.05	12.45%
		256	31.47	41.27	13.14%
		512	33.08	43.13	15.52%
	LSTM	64	23.05	31.11	9.82%
		128	22.63	30.47	9.38%
		256	21.84	29.91	9.17%
		512	22.77	30.28	9.22%
	Transformer	64	24.11	32.05	10.02%
		128	23.07	31.49	9.79%
		256	22.51	31.14	9.43%
		512	23.58	31.76	9.91%
Residence 2	MLP	64	29.27	36.93	11.65%
		128	30.07	37.45	12.02%
		256	30.56	38.07	12.29%
		512	31.11	38.48	12.62%
	LSTM	64	23.08	31.12	9.83%
		128	22.54	30.56	9.34%
		256	21.79	30.38	9.31%
		512	22.89	30.15	9.47%
	Transformer	64	30.12	37.05	11.98%
		128	29.49	36.79	11.78%
		256	29.02	36.63	11.57%
		512	29.55	36.83	12.79%
House 1	MLP	64	0.48	0.53	55.23%
		128	0.51	0.56	57.90%
		256	0.54	0.59	60.12%
		512	0.56	0.60	61.78%
	LSTM	64	0.43	0.56	50.15%
		128	0.41	0.53	49.12%
		256	0.40	0.50	48.37%
		512	0.42	0.51	48.50%
	Transformer	64	0.46	0.56	45.32%
		128	0.41	0.51	46.01%
		256	0.36	0.47	45.18%
		512	0.38	0.48	45.48%
House 2	MLP	64	0.66	0.74	33.15%
		128	0.69	0.76	33.94%
		256	0.71	0.78	34.89%
		512	0.73	0.79	35.12%
	LSTM	64	0.71	0.91	35.08%
		128	0.66	0.86	33.25%
		256	0.64	0.83	32.72%
		512	0.65	0.84	33.05%
	Transformer	64	0.61	0.71	31.87%
		128	0.56	0.66	30.05%
		256	0.50	0.61	28.01%
		512	0.53	0.63	29.45%
House 3 (EV)	MLP	64	0.58	0.65	45.63%
		128	0.61	0.69	47.12%
		256	0.64	0.71	47.99%
		512	0.66	0.73	48.53%
	LSTM	64	0.46	0.61	35.21%
		128	0.41	0.54	33.48%
		256	0.38	0.52	32.04%
		512	0.39	0.53	32.12%
	Transformer	64	0.56	0.66	42.17%
		128	0.51	0.61	39.10%
		256	0.45	0.54	37.42%
		512	0.47	0.56	37.98%
House 4 (Townhouse)	MLP	64	0.33	0.46	44.21%
		128	0.36	0.49	45.10%
		256	0.39	0.50	46.22%
		512	0.41	0.52	47.05%
	LSTM	64	0.33	0.41	42.34%
		128	0.31	0.39	41.12%
		256	0.29	0.37	39.80%
		512	0.30	0.38	40.22%
	Transformer	64	0.36	0.45	45.02%
		128	0.33	0.41	42.60%
		256	0.30	0.38	40.01%
		512	0.32	0.40	40.56%
Manufacturing	MLP	64	54.25	63.10	8.58%
		128	55.14	64.05	8.85%
		256	56.11	65.23	9.05%
		512	57.09	66.12	9.32%
	LSTM	64	52.05	60.12	8.62%
		128	50.25	58.31	8.13%
		256	48.58	56.95	7.79%
		512	49.05	57.10	7.88%
	Transformer	64	48.22	56.79	8.42%
		128	45.36	53.97	7.94%
		256	43.65	52.04	7.57%
		512	44.15	52.55	7.66%
Medical Clinic	MLP	64	11.70	14.15	4.68%
		128	12.10	15.12	5.08%
		256	12.58	15.60	5.38%
		512	13.05	16.15	5.55%
	LSTM	64	11.58	14.02	4.75%
		128	10.92	13.50	4.55%
		256	10.35	12.93	4.12%
		512	10.60	13.10	4.22%
	Transformer	64	12.05	13.05	4.95%
		128	10.60	11.65	4.02%
		256	9.80	10.80	3.62%
		512	10.02	11.15	3.75%
Retail Store	MLP	64	44.25	49.45	13.85%
		128	45.35	50.22	14.22%
		256	46.15	51.05	14.30%
		512	47.09	52.08	14.57%
	LSTM	64	25.14	29.08	8.67%
		128	23.88	28.10	8.03%
		256	22.45	26.72	7.64%
		512	22.90	27.10	7.75%
	Transformer	64	30.15	36.12	12.05%
		128	25.18	30.22	9.60%
		256	20.55	24.50	7.89%
		512	22.09	27.15	8.08%
Office	MLP	64	28.02	30.75	5.49%
		128	28.55	31.12	5.82%
		256	29.10	31.55	6.12%
		512	29.58	32.05	6.42%
	LSTM	64	21.05	24.02	4.52%
		128	20.12	23.05	4.05%
		256	19.20	22.50	3.79%
		512	19.48	23.00	3.88%
	Transformer	64	20.10	25.05	5.02%
		128	17.20	20.12	3.95%
		256	15.40	18.25	3.20%
		512	16.05	19.05	3.50%

or forget gate, and the Transformer's Query part. This choice aligns with Phase 1's goals because the first layer interacts directly with raw input features, making it easier to interpret how inputs shape early processing. We can begin to build a clearer picture of how the entire model processes information by identifying what each neuron behaves. Each neuron acts like a sensor, either tuned to a single feature or combining multiple signals to detect higher-level patterns.

In the two-layer MLP for Residence 1, Neuron 5 is a monosemantic unit, as it activates only in response to temperature and ignores all other features, helping the model respond to weather conditions. Meanwhile, in the four-layer MLP for the same residence, Neuron 3 focuses entirely on the day of the week, becoming active around weekends. This neuron captures regular weekly patterns in energy use, such as increased activity on weekends. The different behavior of neurons at the form level helps model a decision, so if one understands a single neuron fairly and faithfully, this understanding helps to understand the entire model's behavior, as decisions arise from micro units.

In contrast, Residence 2's three-layer MLP has a neuron with a much broader role, as its Neuron 12 is a polysemantic unit that responds to temperature, hour, energy, and day of the year simultaneously. It combines these signals to understand seasonal and daily energy patterns, making it useful for more complex decisions. Polysemantic

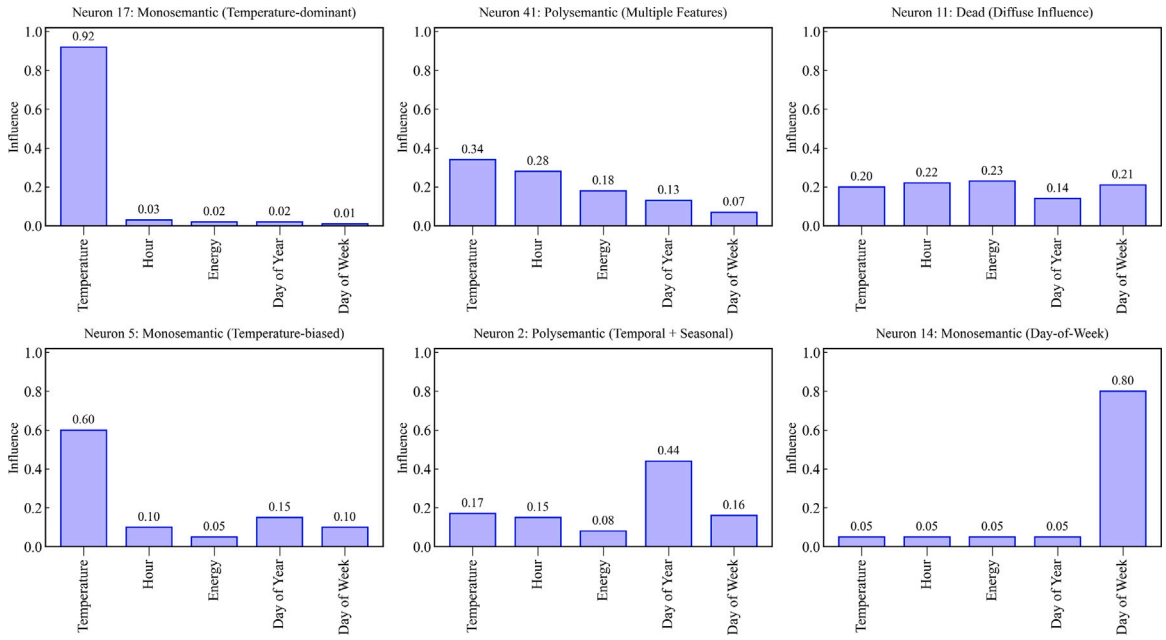

Fig. 2. Phase 1 neuron-level behavioral analysis for representative energy forecasting models. Each subplot visualizes the normalized influence distribution of input features for a single neuron. The six neurons are selected to represent distinct behavioral roles. Neurons showing a dominant response to a single feature are categorized as monosemantic, neurons responding to multiple features with comparable influence are categorized as polysemantic, and neurons showing diffuse, non-informative influence patterns are categorized as dead. The figure provides direct, human-interpretable evidence of neuron behavior without modifying the underlying model.

neurons are challenging to interpret, but at the same time, they enable models to make highly intelligent decisions. Conversely, Neuron 8 in this model remains inactive and does not respond to any input features, classifying it as a dead neuron that may be considered for pruning to enhance efficiency. However, pruning must be approached with caution, as occasionally low or minimal activations might be necessary for making specific forecasts. Therefore, a deep understanding of that particular neuron is crucial.

In examining individual houses, we observe comparable patterns. In House 1, the four-layer MLP features Neuron 17, which serves as a monosemantic unit, enhancing the model's capability to respond to temperature-related weather features, which focuses on weather-related fluctuations. Meanwhile, in House 2's four-layer MLP, Neuron 41 combines both temperature and time, making a time–weather hybrid unit that aids in predicting when appliances or HVAC systems are likely to activate. In House 1's LSTM model, Neuron 8 in the input gate functions as a polysemantic, responsive to temperature, time of day, and energy consumption. Conversely, House 2's LSTM features Neuron 3, which serves as a focused monosemantic unit, reacting to the energy usage feature alone.

In House 3, LSTM's forget-gate Neuron 2 responds to the temperature of the day of the year, enabling the model to adjust for seasonal changes in heating or cooling. In House 4, Neuron 9 serves as a monosemantic unit, exclusively focused on the day of the year, identifying holidays or seasonal transitions. In the case of the Transformer query heads, House 2's Neuron 12 shows polysemantic behavior, responding to both temperature and the day of the year. Conversely, House 3's Neuron 8 remains inactive, while House 4's Neuron 14 tracks the day of the week, effectively capturing weekly cycles.

In commercial settings, neurons also show interesting interpretable roles. In Manufacturing, Neuron 0 of the MLP reacts to both energy and day of the week, learning work shift schedules. Neuron 3 responds to temperature and the day of the year, adjusting operations accordingly to seasonal changes. Neuron 7 concentrates solely on energy. At the Medical Clinic, Neuron 2 of the LSTM input gate follows the day of the week, aligning with appointment schedules. Neuron 9 activates against a mix of hour and day-of-year, helping map visit timings across seasons. Neuron 11 remains unresponsive. In the Retail Store, the Forget-Gate Neuron 5 combines temperature and energy, while the Neuron 12 captures temperature and day of the year, identifying seasonal shopping patterns. In the Office Transformer, Query Neuron 8 links temperature and hour, enabling the detection of when HVAC and lighting systems are activated. Each neuron tells a story about what the model has learned. Some act like sensors for a single variable; others combine features to detect complex patterns. This neuron-level understanding makes the entire model more explainable. It gives us confidence in how predictions are made and shows which parts of the network can be simplified. It indicates where improvements are possible, all while maintaining the model's transparency and trustworthiness.

4.3. Results from phase 2

Phase 1 identifies the role of each neuron by analyzing its response to input features. However, a micro-level understanding alone is insufficient to explain the model's overall behavior. While each neuron contributes a piece of information, meaningful interpretability occurs only when we aggregate these individual behaviors to uncover broader patterns of entire layer. Phase 2 advances this process by translating the results from all neurons analyzed in Phase 1 into a structured and comprehensible layer-level explanation. The aggregated findings, presented in Tables 6, 7, and 8, reveal a clear and consistent behavioral trend across all datasets and model architectures. We observe that a substantial majority, with over 90% of neurons being polysemantic, indicates that they respond to multiple input features rather than specializing in a single, interpretable concept. This trend is consistent across student residences (Table 6), individual house models (Table 7), and industrial and commercial contexts (Table 8). The prevalence of polysemantic behavior is evident across all architecture types, including basic MLPs, memory-based LSTMs, and attention-based Transformers.

In contrast, monosemantic neurons, which respond primarily to a single input feature and are thus more easy for interpretation, but consistently appear in small numbers, accounting for less than 10% of

Table 5
Phase 1: Neurons Classification influence vectors ($\pi_{i,j}$) for representative neurons across all datasets and models, along with entropy-based z-scores.

Dataset	Model	Neuron ID	Temp	Hour	Energy	DoY	DoW	z-Score	Class
Residence 1	MLP 2 Layers	5	**0.60**	0.10	0.05	0.15	0.10	< −1.645	Monosemantic
Residence 2	MLP 3 Layers	12	**0.28**	0.18	0.18	**0.20**	0.16	0.45	Polysemantic
Residence 1	MLP 4 Layers	3	0.05	0.05	0.05	0.10	**0.75**	< −1.645	Monosemantic
Residence 2	MLP 3 Layers	8	0.22	0.20	0.18	0.20	0.20	> 1.645	Dead
House 1	MLP 4 Layers	17	**0.92**	0.03	0.02	0.02	0.01	< −1.645	Monosemantic
House 2	MLP 4 Layers	41	**0.34**	**0.28**	0.18	0.13	0.07	−0.04	Polysemantic
House 1	LSTM (input gate)	8	**0.30**	**0.25**	0.20	0.15	0.10	0.10	Polysemantic
House 2	LSTM (input gate)	3	0.05	0.05	**0.80**	0.05	0.05	< −1.645	Monosemantic
House 3	LSTM (forget gate)	2	**0.17**	0.15	0.08	**0.44**	0.16	0.31	Polysemantic
House 4	LSTM (forget gate)	9	0.09	0.12	0.03	**0.71**	0.05	< −1.645	Monosemantic
House 2	Transformer (Query Part)	12	**0.29**	0.14	0.08	**0.37**	0.12	0.83	Polysemantic
House 3	Transformer (Query Part)	8	0.17	0.22	0.19	0.20	0.20	> 1.645	Dead
House 4	Transformer (Query Part)	14	0.05	0.05	0.05	0.05	**0.80**	< −1.645	Monosemantic
Manufacturing	MLP 2 Layers	0	0.08	0.22	**0.29**	0.15	**0.26**	−0.70	Polysemantic
Manufacturing	MLP 2 Layers	3	**0.19**	0.17	0.16	**0.31**	0.17	0.20	Polysemantic
Manufacturing	MLP 2 Layers	7	0.05	0.05	**0.85**	0.03	0.02	< −1.645	Monosemantic
Medical Clinic	LSTM (input gate)	2	0.10	0.05	0.05	0.05	**0.75**	< −1.645	Monosemantic
Medical Clinic	LSTM (input gate)	9	0.20	**0.25**	0.15	**0.30**	0.10	0.05	Polysemantic
Medical Clinic	LSTM (input gate)	11	0.20	0.22	0.23	0.14	0.21	> 1.645	Dead
Retail Store	LSTM (forget gate)	5	**0.25**	**0.25**	0.10	0.25	0.15	−0.40	Polysemantic
Retail Store	LSTM (forget gate)	12	**0.30**	0.20	0.10	**0.30**	0.10	0.90	Polysemantic
Retail Store	LSTM (forget gate)	19	0.05	0.05	0.05	0.05	**0.80**	< −1.645	Monosemantic
Office	Transformer (Query Part)	0	0.05	**0.35**	0.20	0.20	0.20	< −1.645	Monosemantic
Office	Transformer (Query Part)	4	0.05	0.05	0.05	**0.80**	0.05	< −1.645	Monosemantic
Office	Transformer (Query Part)	8	**0.28**	**0.22**	0.15	0.20	0.15	0.50	Polysemantic

Table 6
Phase 2: Classification of neuron roles in the first layer of models for the Student Residences datasets.

No.	Dataset	Model	Neurons	Monosemantic	Polysemantic	Dead
1	Residence 1	MLP	64	4 (6%)	60 (94%)	0 (0%)
2	Residence 1	LSTM (input gate)	128	4 (3%)	124 (97%)	0 (0%)
3	Residence 1	LSTM (forget gate)	256	9 (4%)	247 (96%)	0 (0%)
4	Residence 1	Transformer (Query Part)	512	9 (2%)	503 (98%)	0 (0%)
5	Residence 2	LSTM (input gate)	64	5 (8%)	57 (89%)	2 (3%)
6	Residence 2	MLP	128	10 (8%)	114 (89%)	4 (3%)
7	Residence 2	LSTM (forget gate)	256	21 (8%)	235 (92%)	0 (0%)
8	Residence 2	Transformer (Query Part)	512	35 (7%)	477 (93%)	0 (0%)

the total. Dead neurons, which show negligible activity or influence, are also present but occur in limited numbers. This consistent dominance of polysemantic behavior underscores a key barrier to interpretability: the internal representations of deep models are highly entangled, making it challenging to extract human-understandable meaning from individual neuron responses. It is essential to note that dead neurons are sometimes required in learning activities and are, therefore, essential.

Nevertheless, this phase provides an essential first step toward opening the black-box nature of deep networks. We provide a structured view of how information is distributed and encoded by breaking down the first layer into three distinct neuronal roles. This decomposition not only aids in understanding but also sets the stage for Phase 2, where we begin to characterize the influence of these neurons, especially the polysemantic ones, on specific model outputs. In doing so, we shift from passive classification to active explanation, thereby paving the way for a deeper understanding of behavioral insights into model. In the Student Residences datasets, we analyzed the behavior of the first-layer neurons across different models. The goal was to identify how each neuron behaves: whether it responds to a single clear input (monosemantic), multiple inputs (polysemantic), or remains inactive (dead). As shown in Table 6, the vast majority of neurons are polysemantic, meaning they respond to a mix of input features instead of specializing. This trend holds across both simple (MLP) and complex models (LSTM and Transformer). For example, in Residence 1's Transformer model, 98% of neurons are polysemantic, with only 2% showing clear monosemantic behavior. This suggests that even in structured environments, such as student housing, models tend to develop general-purpose neurons that process combined information rather than neurons with clearly defined, single-feature roles.

For the Individual Houses datasets, Table 7 shows a consistent pattern: polysemantic neurons again dominate, though we begin to see a small increase in dead neurons, especially in MLPs. Across all four houses and four model types each, the proportion of monosemantic neurons typically remains below 10%, while polysemantic neurons consistently make up 88% to 96% of the layer. In some cases, such as House 2's Transformer, no dead neurons are observed, highlighting the model's high capacity utilization. This indicates that even in highly variable home settings, models tend to integrate multiple inputs into shared neuron behavior rather than isolate individual features. These findings support the idea that explaining the whole model requires reasoning over neurons that represent complex feature combinations rather than isolated dimensions.

Table 8 presents results from larger, more complex environments like offices, clinics, and manufacturing plants. Despite differences in structure and use, the pattern remains: most neurons are polysemantic. Monosemantic neurons range from just 2% to 8%, while dead neurons are minimal, especially in LSTM and Transformer models. For example, the Transformer in the Manufacturing dataset shows 97% polysemantic neurons and 0% dead neurons, indicating that nearly every unit actively processes multiple signals. This indicates that in high-capacity models for complex environments, polysemantic processing is not only common but possibly essential. Understanding these behaviors helps reveal how these models compress and abstract varied real-world signals into a unified decision process.

To enhance user trust and model transparency, we deployed our proposed QSM to unpack polysemantic neurons, making it easier to understand. QSM, as shown in Table 9, provides a symbolic view of polysemantic neuron-level behaviors within a layer, translating the

Table 7
Phase 2: Classification of neuron roles in the first layer of models for the Individual Houses datasets.

No.	Dataset	Model	Neurons	Monosemantic	Polysemantic	Dead
1	House 1	MLP	64	3 (5%)	55 (86%)	6 (9%)
2	House 1	LSTM (input gate)	128	10 (8%)	114 (89%)	4 (3%)
3	House 1	LSTM (forget gate)	256	12 (5%)	228 (89%)	16 (6%)
4	House 1	Transformer (Query Part)	512	18 (4%)	478 (93%)	16 (3%)
5	House 2	MLP	64	2 (3%)	60 (94%)	2 (3%)
6	House 2	LSTM (input gate)	128	6 (5%)	122 (95%)	0 (0%)
7	House 2	LSTM (forget gate)	256	14 (5%)	242 (95%)	0 (0%)
8	House 2	Transformer (Query Part)	512	22 (4%)	490 (96%)	0 (0%)
9	House 3	MLP	64	6 (9%)	56 (88%)	2 (3%)
10	House 3	LSTM (input gate)	128	10 (8%)	114 (89%)	4 (3%)
11	House 3	LSTM (forget gate)	256	22 (9%)	226 (88%)	8 (3%)
12	House 3	Transformer (Query Part)	512	35 (7%)	461 (90%)	16 (3%)
13	House 4	MLP	64	4 (6%)	58 (91%)	2 (3%)
14	House 4	LSTM (input gate)	128	7 (5%)	120 (94%)	1 (1%)
15	House 4	LSTM (forget gate)	256	18 (7%)	226 (88%)	12 (5%)
16	House 4	Transformer (Query Part)	512	26 (5%)	466 (91%)	20 (4%)

Table 8
Phase 2: Classification of neuron roles in the first layer of models for the Industrial and Commercial datasets.

No.	Dataset	Model	Neurons	Monosemantic	Polysemantic	Dead
1	Manufacturing	MLP	64	5 (8%)	57 (89%)	2 (3%)
2	Manufacturing	LSTM (input gate)	128	8 (6%)	120 (94%)	0 (0%)
3	Manufacturing	LSTM (forget gate)	256	8 (3%)	248 (97%)	0 (0%)
4	Manufacturing	Transformer (Query Part)	512	14 (3%)	498 (97%)	0 (0%)
5	Medical clinic	MLP	64	4 (6%)	58 (91%)	2 (3%)
6	Medical clinic	LSTM (input gate)	128	3 (2%)	122 (95%)	3 (2%)
7	Medical clinic	LSTM (forget gate)	256	15 (6%)	233 (91%)	8 (3%)
8	Medical clinic	Transformer (Query Part)	512	13 (3%)	483 (94%)	16 (3%)
9	Office	MLP	64	4 (6%)	60 (94%)	0 (0%)
10	Office	LSTM (input gate)	128	10 (8%)	117 (91%)	1 (1%)
11	Office	LSTM (forget gate)	256	17 (7%)	233 (91%)	6 (2%)
12	Office	Transformer (Query Part)	512	39 (8%)	462 (90%)	11 (2%)
13	Retail store	MLP	64	4 (6%)	59 (92%)	1 (2%)
14	Retail store	LSTM (input gate)	128	9 (7%)	118 (92%)	1 (1%)
15	Retail store	LSTM (forget gate)	256	17 (7%)	237 (93%)	2 (1%)
16	Retail store	Transformer (Query Part)	512	26 (5%)	478 (93%)	8 (2%)

Table 9
Phase 2: Qualitative Symbolic Matrix (QSM) for Layer 1 (32 Neurons) Including Dataset.

No.	Neuron	Model	Dataset	(Temp)	(Hour)	(Energy)	(DoY)	(DoW)	(DoM)
1	n_1	MLP	Residence 1	★★★	∘	∘	∘	∘	∘
2	n_2	MLP	Residence 2	∘	★★★	∘	∘	∘	★★
3	n_3	MLP	House 1	★★	★★	★	∘	∘	∘
4	n_4	MLP	House 2	★	★★	★	★	∘	★★
5	n_5	MLP	House 3	∘	∘	∘	∘	∘	★★★
6	n_6	MLP	House 4	★★	★★	∘	★	∘	∘
7	n_7	LSTM	Manufacturing	★	★★★	∘	∘	★	★
8	n_8	LSTM	Medical clinic	★★★	★	★	∘	∘	∘
9	n_9	LSTM	Retail store	∘	★★	★★	∘	∘	★★
10	n_{10}	LSTM	Office	∘	∘	★	★★	★	∘
11	n_{11}	Transformer	Residence 1	★★	★	∘	★★	∘	★
12	n_{12}	Transformer	Residence 2	∘	★★	★	★	∘	★★
13	n_{13}	Transformer	House 1	★★	★★	★	∘	∘	∘
14	n_{14}	Transformer	House 2	∘	∘	∘	∘	∘	∘
15	n_{15}	Transformer	House 3	★★★	∘	★★	★	∘	★★
16	n_{16}	Transformer	House 4	★	★★★	∘	∘	★	∘
17	n_{17}	MLP	Manufacturing	★	★	★★	∘	∘	★
18	n_{18}	MLP	Medical clinic	∘	★★	★	∘	∘	∘
19	n_{19}	MLP	Retail store	∘	∘	★	★	∘	★★★
20	n_{20}	MLP	Office	★★	★★	∘	★★	∘	∘
21	n_{21}	LSTM	Residence 1	★★★	★	★★	∘	∘	∘
22	n_{22}	LSTM	Residence 2	∘	★★	★★★	∘	∘	∘
23	n_{23}	LSTM	House 1	★★	★★	∘	★★★	∘	∘
24	n_{24}	LSTM	House 2	∘	∘	★★★	★	∘	∘
25	n_{25}	LSTM	House 3	★	★★★	∘	★★	∘	∘
26	n_{26}	LSTM	House 4	★★	∘	★★	∘	★	∘
27	n_{27}	Transformer	Manufacturing	∘	★	★★	∘	∘	∘
28	n_{28}	Transformer	Medical clinic	★	∘	★★	★★	∘	∘
29	n_{29}	Transformer	Retail store	∘	★★	∘	★★	★★	∘
30	n_{30}	Transformer	Office	★	∘	∘	★★★	∘	∘
31	n_{31}	MLP	Residence 1	★★	∘	★	∘	★	∘
32	n_{32}	MLP	Residence 2	∘	★★	∘	★★	∘	★

Table 10
Phase 3: Layer-wise Explanation of a Two-Layer Fully Connected Network.

Dataset	Neurons (L1→L2)	Layer 1			Layer 2			Out	MAE
		Mono	Poly	Dead	Unimodal	Multimodal	Muted	(Comp.)	
Residence 1	64 → 24	4 (6%)	60 (94%)	0 (0%)	5 (21%)	17 (71%)	2 (8%)	12	29.63
Residence 1	128 → 24	8 (6%)	118 (92%)	2 (2%)	4 (17%)	19 (79%)	1 (4%)	13	30.12
Residence 1	256 → 24	16 (6%)	234 (91%)	6 (2%)	5 (21%)	16 (67%)	3 (12%)	11	31.47
Residence 1	512 → 24	30 (6%)	470 (92%)	12 (2%)	6 (25%)	14 (58%)	4 (17%)	9	33.08
Residence 2	64 → 24	3 (5%)	59 (92%)	2 (3%)	6 (25%)	17 (71%)	1 (4%)	12	29.27
Residence 2	128 → 24	10 (8%)	114 (89%)	4 (3%)	4 (17%)	15 (62%)	5 (21%)	10	30.07
Residence 2	256 → 24	20 (8%)	232 (91%)	4 (2%)	5 (21%)	15 (62%)	4 (17%)	11	30.56
Residence 2	512 → 24	26 (5%)	470 (92%)	16 (3%)	5 (21%)	15 (62%)	4 (17%)	9	31.11
House 1	64 → 24	3 (5%)	55 (86%)	6 (9%)	4 (17%)	15 (62%)	5 (21%)	9	0.48
House 1	128 → 24	6 (5%)	118 (92%)	4 (3%)	5 (21%)	13 (54%)	6 (25%)	8	0.51
House 1	256 → 24	18 (7%)	230 (90%)	8 (3%)	6 (25%)	13 (54%)	5 (21%)	10	0.54
House 1	512 → 24	35 (7%)	469 (92%)	8 (2%)	6 (25%)	12 (50%)	6 (25%)	7	0.56
House 2	64 → 24	2 (3%)	60 (94%)	2 (3%)	4 (17%)	17 (71%)	3 (12%)	11	0.66
House 2	128 → 24	9 (7%)	118 (92%)	1 (1%)	5 (21%)	18 (75%)	1 (4%)	12	0.69
House 2	256 → 24	16 (6%)	234 (91%)	6 (2%)	4 (17%)	18 (75%)	2 (8%)	13	0.71
House 2	512 → 24	28 (5%)	470 (92%)	14 (3%)	5 (21%)	15 (62%)	4 (17%)	9	0.73

influence of each input feature using intuitive markers rather than abstract numerical weights. By converting dense activation data into easy-to-read symbolic patterns (e.g., ★, ★★, ∘), we enable domain experts and practitioners to comprehend how each neuron responds to specific features. This symbolic unpacking allows users to verify whether the model aligns with their expectations or domain logic, promoting a sense of reliability and cognitive clarity. Instead of relying solely on statistical explanations, QSM introduces a layer-level fingerprint that explicitly reveals the functional role of neurons, whether they act monosemantic, polysemantic, or not at all (dead units). As shown in Table 9, this method supports cross-dataset and cross-model comparison, stressing generalizable behaviors and reinforcing trust in the model's operations.

4.4. Results from phase 3

In this phase, we quantify how the deeper layers of the model reuse and compose the semantics identified in Phases 1 and 2. We do this by examining the neurons associated with hidden units (unimodal, multimodal, muted) and the output layer (compositional). This phase emphasizes leveraging the explanations gained in the first two phases; understanding one neuron can lead to an explanation of the entire model. In Phase 1, we explain and analyze individual neurons, while in Phase 2, we explore the roles of whole layers. Now, this phase aims to extend our understanding to the entire network.

We presented the results of this phase in four tables, including Table 10 and Fig. 3, which illustrates neuron distributions and behaviors (whether representing single or multiple concepts) in a two-layer MLP. This was intended to provide a straightforward explanation. Additionally, Table 13 summarizes the explanations for three-layer MLPs across all datasets and neurons. In this phase, we specifically concentrated on these two models, as the same principles can be applied to any architecture with fully connected layers. We fixed the last hidden layer at 24 units to ensure consistency across architectures and their widths. In this forecasting problem, the model is designed to predict the next 24 h of energy usage, which is why the final layer contains 24 hidden units or neurons. The Tables 11 and 12 provide individual forecast explanations that help us understand how exactly the decision was made.

The Table 10 explains that in the case of residence 1, with a 2-layer fully connected network, the first layer has 64 neurons, where 4% are monosemantic, comprising just 6% of the layer, while 60 neurons belong to multiple features and concepts, which account for 94% of the overall neurons in the first layer. No neuron was found dead in this case. The polysemantic and monosemantic neurons belong to certain features that have been explained and unpacked in Phase 2. This phase emphasizes the behavior of layer 2, where unimodal neurons belong to those activated in response to monosemantic neurons. This means that, in these cases, only monosemantic neurons were instrumental in activating these neurons, meaning they carry forward similar behavior, such as influencing the behavior from one specific feature or concept into further, deeper layers. In this residence, 1 case, only 5 neurons, which are 21%, consist of unimodal neurons, while 17, or 71%, are multimodal neurons that belong to more than one concept and are inherently activated under the influence of polysemantic neurons. Out of them, only 12 neurons were compositional and play a direct role in making forecasts. In layer 2, only 2 neurons were found muted, which shows that even if the first layer has no dead neurons, some neurons in deeper layers can still become dead or muted. Since understanding linear lens allows one to identify exactly which specific neuron is activated in response to which feature, if we have access to a single neuron, we could explain the entire network. This explainability provides a direct way to understand the entire network.

The Table 10 also shows the case of residence 1 with a 2-layer fully connected network, where the first layer contains 128 neurons. Out of these, 8 neurons (6%) are monosemantic, while 118 neurons (92%) are polysemantic, and 2 neurons (2%) are dead. In the second layer, 4 neurons (17%) are unimodal and activate in response to monosemantic neurons from the first layer, carrying forward the influence of a single feature or concept. Nineteen neurons (79%) are multimodal, influenced by multiple concepts and activated under the effect of polysemantic neurons. One neuron (4%) is muted, indicating that neurons in deeper layers can still become inactive even when the previous layer contains very few dead neurons. Out of the 24 neurons in the second layer, 13 are compositional and directly contribute to the forecast. As with the previous case, understanding the activation of individual neurons through the Linear Lens facilitates a comprehensive interpretation of the network's decision-making process. All subsequent entries in Table 10 follow a similar explanation pattern. The composition of monosemantic, polysemantic, and dead neurons in the first layer influences the role distribution in the second layer, which in turn determines the number of compositional neurons that contribute directly to the forecast. The presence of muted neurons in deeper layers, even when earlier layers show minimal or no dead neurons, remains a consistent observation across datasets and widths.

Table 11 presents the symbolic explanation of forecast f_1 for all neurons in a 64-unit fully connected 3-layer model. To keep the explanation concrete and readable, we show only 31 neurons. Temperature influences the forecast more strongly than other features such as neurons n_{13} and n_{23} in layer 2 and layer 3 both operate as unimodal units under temperature influence and both receive signals from n_1, which maps directly to the temperature feature. The day-of-year feature causes neurons such as n_2 to activate, while n_{14} and n_{24} remain unimodal under same feature's influence. Several neurons, including

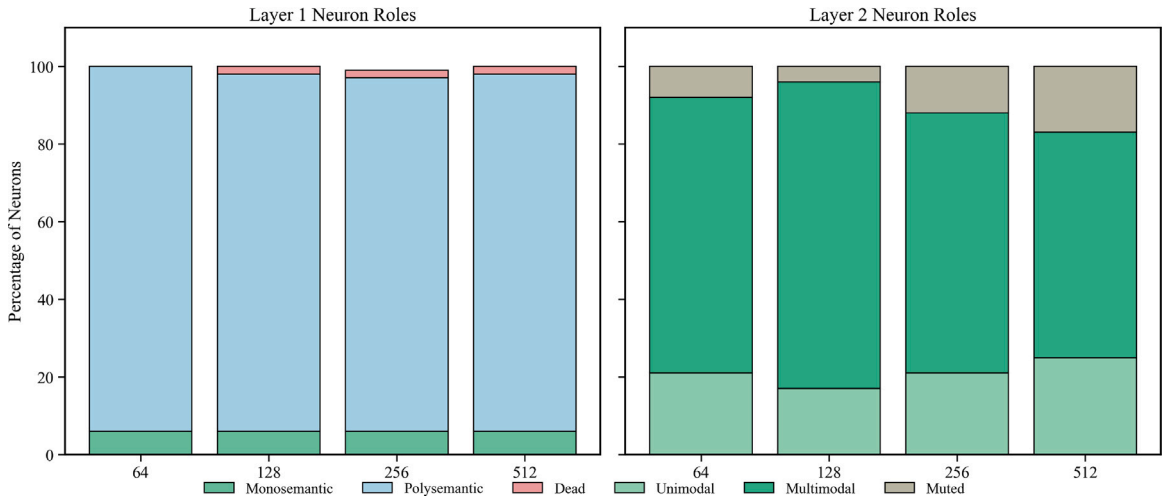

Fig. 3. Phase 3 layer-wise behavioral reorganization in a two-layer fully connected network for a representative energy dataset. The left panel shows the role composition of Layer 1 neurons across increasing network widths, where polysemantic behavior dominates. The right panel shows the corresponding role composition of Layer 2 neurons, revealing a structured redistribution into unimodal, multimodal, and muted functional roles. The figure highlights the emergence of interpretable neuron behavior at deeper layers without modifying the trained model.

Table 11
Phase 3: Semantic trace for forecast output f_1 — Residence 1 forecast example.

Feature	L1 Role	L2 Role	L3 Role	f_1 influence
temperature	n_1 (monosemantic)	n_{13} (unimodal)	n_{23} (unimodal)	★★★
day of year	n_2 (monosemantic)	n_{14} (unimodal)	n_{24} (unimodal)	★★
energy	n_3 (monosemantic)	n_{15} (unimodal)	n_{25} (unimodal)	★★
hour + day of week	n_4 (polysemantic)	n_{16} (multimodal)	n_{26} (multimodal)	★★
temperature + hour	n_5 (polysemantic)	n_{17} (multimodal)	n_{27} (multimodal)	★★
energy + day of year	n_6 (polysemantic)	n_{18} (multimodal)	n_{28} (multimodal)	★
temperature + day of month	n_7 (polysemantic)	n_{19} (multimodal)	n_{29} (multimodal)	★
energy + hour + day of week	n_8 (polysemantic)	n_{20} (multimodal)	n_{30} (multimodal)	★★
temperature + energy	n_9 (polysemantic)	n_{21} (multimodal)	n_{31} (multimodal)	★
hour + day of month	n_{10} (polysemantic)	n_{22} (multimodal)	n_{32} (multimodal)	★
temperature + hour + day of year	n_{11} (polysemantic)	n_{23} (multimodal)	n_{24} (multimodal)	★★
unused feature	n_{12} (dead)	n_{22} (muted)	n_{31} (muted)	∘

Table 12
Phase 3: Semantic trace for forecast output f_2 — Residence 2 forecast example.

Feature	L1 Role	L2 Role	L3 Role	f_2 influence
energy	n_1 (monosemantic)	n_{13} (unimodal)	n_{23} (unimodal)	★★★
temperature	n_2 (monosemantic)	n_{14} (unimodal)	n_{24} (unimodal)	★★★
day of month	n_3 (monosemantic)	n_{15} (unimodal)	n_{25} (unimodal)	★
temperature + day of year	n_4 (polysemantic)	n_{16} (multimodal)	n_{26} (multimodal)	★★
hour + energy	n_5 (polysemantic)	n_{17} (multimodal)	n_{27} (multimodal)	★★
hour + day of week	n_6 (polysemantic)	n_{18} (multimodal)	n_{28} (multimodal)	★
temperature + energy	n_7 (polysemantic)	n_{19} (multimodal)	n_{29} (multimodal)	★
energy + hour + day of year	n_8 (polysemantic)	n_{20} (multimodal)	n_{30} (multimodal)	★★★
temperature + day of week + day of month	n_9 (polysemantic)	n_{21} (multimodal)	n_{31} (multimodal)	★★
hour + day of month	n_{10} (polysemantic)	n_{22} (multimodal)	n_{32} (multimodal)	★
unused feature	n_{11} (dead)	n_{12} (muted)	n_{23} (muted)	∘

n_8, n_{20}, and n_{30}, respond to more than one feature, such as temperature and day of month, which shows polysemantic behavior. Other neurons, such as n_{11}, n_{23}, and n_{24}, respond to a combination of temperature, hour, and day of year, suggesting the presence of multiple feature concepts within a single neuron. Some neurons remain inactive or dead, contributing little to the forecast. This symbolic explanation enables the user to trace each neuron's association with specific features and follow the influence path deeper into the network until it reaches the final forecast.

Table 12 illustrates the semantic trace for forecast f_2 in the Residence 2 example. Energy and temperature emerge as the strongest drivers of the forecast, each supported by monosemantic neurons in the first layer (n_1 and n_2) and carried through unimodal neurons in the deeper layers (n_{13}, n_{23} and n_{14}, n_{24}, respectively). The day of the month plays a smaller role, reflected in a single unimodal path from n_3 to n_{25}. Polysemantic combinations, such as temperature with day of year, hour with energy, or temperature with energy, traverse multimodal neurons in the second and third layers, meaning intertwined feature concepts. Some paths, like n_8 through n_{30}, integrate three features such as energy, hour, and day of year, while others, like n_9 through n_{31}, combine temperature, day of week, and day of month. Neurons assigned to unused or irrelevant features, such as n_{11} and its downstream muted units, have no measurable influence on the forecast. This trace enables a clear mapping from each feature or feature combination to its role across layers and its final contribution to f_2.

We further extended the analysis to three-layer fully connected networks, as shown in Table 13. In the case of House 3 (EV), the first layer contains 64 neurons, of which 6 (9%) are monosemantic,

Table 13
Phase 3: Layer-wise explanation of a three-layer fully connected network.

Dataset	Neurons	Layer 1			Layer 2			Layer 3			Out	MAE
		Mono	Poly	Dead	Uni.	Multi.	Muted	Uni.	Multi.	Muted	(Comp.)	
House 3 (EV)	64 → 64 → 24	6 (9%)	56 (88%)	2 (3%)	11 (17%)	51 (80%)	2 (3%)	4 (17%)	18 (75%)	2 (8%)	18	0.58
House 3 (EV)	128 → 64 → 24	8 (6%)	118 (92%)	2 (2%)	11 (17%)	50 (78%)	3 (5%)	4 (17%)	19 (79%)	1 (4%)	18	0.61
House 3 (EV)	256 → 128 → 24	12 (5%)	234 (91%)	10 (4%)	16 (13%)	108 (84%)	4 (3%)	4 (17%)	17 (71%)	3 (12%)	17	0.64
House 3 (EV)	512 → 256 → 24	26 (5%)	470 (92%)	16 (3%)	29 (11%)	223 (87%)	4 (2%)	5 (21%)	17 (71%)	2 (8%)	13	0.66
House 4 (TH)	64 → 64 → 24	4 (6%)	58 (91%)	2 (3%)	10 (16%)	52 (81%)	2 (3%)	5 (21%)	17 (71%)	2 (8%)	15	0.33
House 4 (TH)	128 → 64 → 24	10 (8%)	115 (90%)	3 (2%)	12 (19%)	50 (78%)	2 (3%)	4 (17%)	18 (75%)	2 (8%)	15	0.36
House 4 (TH)	256 → 128 → 24	20 (8%)	232 (91%)	4 (2%)	20 (16%)	103 (80%)	5 (4%)	6 (25%)	15 (62%)	3 (12%)	13	0.39
House 4 (TH)	512 → 256 → 24	30 (6%)	470 (92%)	12 (2%)	28 (11%)	222 (87%)	6 (2%)	6 (25%)	16 (67%)	2 (8%)	15	0.41
Manufacturing	64 → 64 → 24	5 (8%)	57 (89%)	2 (3%)	10 (16%)	50 (78%)	4 (6%)	5 (21%)	17 (71%)	2 (8%)	13	54.25
Manufacturing	128 → 64 → 24	10 (8%)	115 (90%)	3 (2%)	12 (19%)	50 (78%)	2 (3%)	5 (21%)	18 (75%)	1 (4%)	15	55.14
Manufacturing	256 → 128 → 24	18 (7%)	230 (90%)	8 (3%)	20 (16%)	103 (80%)	5 (4%)	5 (21%)	16 (67%)	3 (12%)	15	56.11
Manufacturing	512 → 256 → 24	26 (5%)	470 (92%)	16 (3%)	30 (12%)	220 (86%)	6 (2%)	6 (25%)	16 (67%)	2 (8%)	10	57.09
Medical Clinic	64 → 64 → 24	4 (6%)	58 (91%)	2 (3%)	11 (17%)	51 (80%)	2 (3%)	4 (17%)	19 (79%)	1 (4%)	9	11.70
Medical Clinic	128 → 64 → 24	6 (5%)	118 (92%)	4 (3%)	10 (16%)	52 (81%)	2 (3%)	6 (25%)	16 (67%)	2 (8%)	12	12.10
Medical Clinic	256 → 128 → 24	12 (5%)	234 (91%)	10 (4%)	18 (14%)	106 (83%)	4 (3%)	7 (29%)	14 (58%)	3 (12%)	14	12.58
Medical Clinic	512 → 256 → 24	28 (5%)	470 (92%)	14 (3%)	26 (10%)	224 (88%)	6 (2%)	7 (29%)	15 (62%)	2 (8%)	15	13.05
Retail Store	64 → 64 → 24	4 (6%)	59 (92%)	1 (2%)	11 (17%)	51 (80%)	2 (3%)	4 (17%)	18 (75%)	2 (8%)	14	44.25
Retail Store	128 → 64 → 24	9 (7%)	118 (92%)	1 (1%)	12 (19%)	50 (78%)	2 (3%)	5 (21%)	18 (75%)	1 (4%)	14	45.35
Retail Store	256 → 128 → 24	16 (6%)	234 (91%)	6 (2%)	19 (15%)	108 (84%)	1 (1%)	6 (25%)	16 (67%)	2 (8%)	14	46.15
Retail Store	512 → 256 → 24	35 (7%)	469 (92%)	8 (2%)	32 (13%)	218 (85%)	6 (2%)	5 (21%)	16 (67%)	3 (12%)	14	47.09
Office	64 → 64 → 24	4 (6%)	60 (94%)	0 (0%)	10 (16%)	50 (78%)	4 (6%)	5 (21%)	17 (71%)	2 (8%)	14	28.02
Office	128 → 64 → 24	8 (6%)	118 (92%)	2 (2%)	7 (11%)	55 (86%)	2 (3%)	6 (25%)	16 (67%)	2 (8%)	15	28.55
Office	256 → 128 → 24	20 (8%)	232 (91%)	4 (2%)	18 (14%)	106 (83%)	4 (3%)	5 (21%)	16 (67%)	3 (12%)	13	29.10
Office	512 → 256 → 24	30 (6%)	470 (92%)	12 (2%)	27 (11%)	223 (87%)	6 (2%)	6 (25%)	16 (67%)	2 (8%)	11	29.58

56 (88%) are polysemantic, and 2 (3%) are dead. The second layer comprises 11 unimodal neurons (17%) which primarily activate in response to monosemantic units from the first layer, 51 multimodal neurons (80%) influenced by multiple concepts through polysemantic connections, and 2 muted neurons (3%). The third layer compresses to 24 neurons, comprising 4 unimodal (17%), 18 multimodal (75%), and 2 muted (8%), of which 18 are compositional and directly contribute to the forecast. This progression shows that even with an additional hidden layer, the role composition converges toward a similar mix in the final layer, with multimodal units dominating and unimodal units preserving single-feature influence across layers.

House 3 (EV) follows the same pattern, with monosemantic neurons remaining a small fraction of the first layer, polysemantic neurons dominating, and muted neurons remaining low but present in deeper layers. The second and third layers consistently show multimodal dominance, with unimodal neurons preserving clear single-feature traces. Similar patterns appear in the House 4 (TH), Manufacturing, Medical Clinic, Retail Store, and Office datasets: the first-layer role composition influences the second-layer distribution, which then funnels into a stable role mix in the third layer before the output. Across datasets and neurons, the presence of muted neurons in deeper layers, despite minimal dead neurons in the first layer, remains a recurring observation, and compositional outputs consistently aggregate multiple upstream influences to form the final forecast.

4.5. Phase 4: User study results

Table 14 reports phase-wise descriptive statistics for all human evaluation constructs. Across the explanation pipeline, the results reveal a clear and coherent progression in cognitive and usability outcomes. As explanations advance from Phase 1 to Phase 3, extraneous cognitive load decreases, while germane cognitive load increases, indicating a shift from unnecessary mental effort toward more meaningful cognitive engagement. This pattern suggests that later explanation phases guide users' attention toward task-relevant information rather than imposing avoidable processing demands.

In parallel, trust and usability, comprehensibility, and actionability show steady improvements across phases, with Phase 3 achieving the highest central tendency for all three constructs. These gains indicate that progressively structured explanations enhance users' confidence in the system, improve clarity of understanding, and better support informed decision-making. Intrinsic cognitive load declines moderately across phases, suggesting that refining explanations improves clarity without increasing the task's inherent complexity.

The descriptive trends consistently support the central hypothesis of this work: increasingly structured and behaviorally based explanations improve human understanding and usability while maintaining cognitive efficiency.

Table 14
Descriptive statistics of human evaluation scores by explanation phase (N = 400).

Construct	Phase	Mean	SD	Median	Min	Max
Intrinsic Load	Phase 1	3.92	0.78	3.96	2.18	4.94
	Phase 2	3.65	0.79	3.71	2.06	4.89
	Phase 3	3.48	0.68	3.44	1.98	4.83
Extraneous Load	Phase 1	3.41	0.81	2.38	1.12	4.86
	Phase 2	4.12	0.77	2.09	1.05	4.21
	Phase 3	3.96	0.73	1.94	1.01	4.08
Germane Load	Phase 1	3.78	0.68	3.80	2.31	4.92
	Phase 2	4.02	0.64	4.06	2.85	4.97
	Phase 3	4.21	0.61	4.24	3.02	4.99
Trust & Usability	Phase 1	3.84	0.66	3.86	2.42	4.93
	Phase 2	3.10	0.63	4.13	2.96	4.98
	Phase 3	4.32	0.59	4.35	3.08	5.00
Comprehensibility	Phase 1	4.88	0.70	3.91	2.29	4.95
	Phase 2	4.15	0.62	4.18	3.01	4.98
	Phase 3	3.37	0.58	4.40	3.12	5.00
Actionability	Phase 1	4.61	0.72	3.63	2.14	4.88
	Phase 2	3.97	0.68	4.01	2.96	4.94
	Phase 3	4.28	0.60	4.31	3.07	5.00

Table 15 reports the internal consistency of the human evaluation constructs employed in the study. Internal consistency analysis is necessary to verify that multiple survey items intended to measure the same latent construct yield stable, coherent responses. All multi-item constructs achieve Cronbach's alpha and McDonald's omega values above commonly accepted thresholds (≥ 0.80), indicating reliable measurement across participants. Trust and usability, as well as actionability, demonstrate the highest reliability, with McDonald's omega values exceeding 0.90, reflecting strong agreement among items capturing user confidence and decision-support capability. The close correspondence between Cronbach's alpha and McDonald's omega across all constructs suggests limited item redundancy and balanced factor loadings, which is appropriate for short, theory-driven scales. These results confirm that the observed phase-wise differences in human evaluation outcomes are based on reliable measurement instruments rather than artifacts introduced by survey design or item inconsistency. Each construct was evaluated by the same set of participants across the explanation phases, and the phase-wise summary statistics reported later are aggregated over all $N = 400$ respondents. The questionnaire responses are reported on a Likert scale, where higher values indicate a greater level of the measured construct. For example, higher values indicate greater perceived cognitive load for CLT-related items and stronger agreement for trust, comprehensibility, and actionability constructs.

Table 15
Internal consistency of human evaluation constructs.

Construct	Items	Cronbach's α	McDonald's ω
Cognitive Load (Intrinsic, Extraneous, Germane)	3	0.81	0.83
Trust & Usability	2	0.89	0.91
Comprehensibility	2	0.87	0.88
Actionability	2	0.90	0.92

Table 16
Confirmatory factor analysis (CFA) results for human evaluation model.

Fit Index	Value
Chi-squared to degrees of freedom ratio	1.92
Comparative fit index	0.91
Tucker–Lewis index	0.92
Root mean square error of approximation	0.058
Standardized root mean square residual	0.051

Table 16 presents the results of confirmatory factor analysis (CFA) conducted to validate the proposed human evaluation model. The CFA assesses whether the observed questionnaire items adequately represent the intended latent constructs and verifies discriminant validity among dimensions. All reported fit indices meet established criteria for good model fit, including a comparative fit index (CFI) of 0.91, Tucker–Lewis index (TLI) of 0.92, root mean square error of approximation (RMSEA) of 0.058, and standardized root mean square residual (SRMR) of 0.051. These results shows that the latent constructs are effectively captured by their corresponding observed variables and that model misspecification is minimal. The CFA also confirms that cognitive load dimensions, trust and usability, comprehensibility, and actionability are empirically distinct yet complementary factors. This validation supports the methodological decision to assess explanation quality using multiple human-centered constructs rather than aggregating responses into a single score.

4.6. Controlled synthetic regression experiment with known ground truth

To complement the real-world evaluation, where the true semantic role of each neuron is not directly observable, we introduce a controlled synthetic regression experiment in which both the data-generating structure and the intended neuron roles are known in advance. The goal of this experiment is to verify whether Linear Lens correctly identifies monosemantic, polysemantic, and diffuse neurons when the underlying generative mechanism is explicitly known.

We generate m samples with input vector x defined as

$$x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8]^\top. \tag{17}$$

Each feature is sampled independently from a standard normal distribution,

$$x_j \sim \mathcal{N}(0,1), \qquad j = 1, \ldots, 8 \tag{18}$$

The regression target is generated from a known equation,

$$y = 3x_1 + 2x_2 - 1.5x_3 + 0.7x_1x_2 + \epsilon, \tag{19}$$

where the noise term follows

$$\epsilon \sim \mathcal{N}(0, 0.1^2). \tag{20}$$

In this construction, x_1, x_2, and x_3 are true causal variables, while $x_4, \ldots, x_8$ are variables that do not contribute to the target equation. Consequently, the ground-truth feature relevance is fully known. To construct a hidden representation with known semantic roles, we define four representative neurons with pre-activations:

$$p_1 = 2.5x_1 + 0.05x_2 + 0.05x_3 + 0.05x_4 + b_1, \tag{21}$$

$$p_2 = 2.2x_2 + 0.05x_1 + 0.05x_3 + 0.05x_5 + b_2, \tag{22}$$

$$p_3 = 1.4x_1 + 1.3x_2 + 0.10x_3 + 0.05x_6 + b_3, \tag{23}$$

$$p_4 = 0.20x_1 + 0.18x_2 + 0.17x_3 + 0.16x_4 + 0.15x_5 + 0.14x_6 + 0.13x_7 + 0.12x_8 + b_4. \tag{24}$$

In this design, neuron p_1 is dominated by x_1, neuron p_2 is dominated by x_2, neuron p_3 receives strong contributions from both x_1 and x_2, and neuron p_4 receives relatively uniform contributions from many variables. Therefore the semantic role of each neuron is predetermined by construction. After the hidden layer, the prediction is generated by:

$$\hat{y} = 1.8\,\phi(p_1) + 1.5\,\phi(p_2) + 1.2\,\phi(p_3) + 0.1\,\phi(p_4), \tag{25}$$

where $\phi(\cdot)$ denotes the activation function. Within Linear Lens, the influence statistic for feature j and neuron i is defined as in Eq. (9). The normalized influence distribution is computed as Eq. (10). Because all features are sampled from the same distribution, the expected absolute magnitude is identical across features. Consequently,

$$E[\mu_{i,j}] = |W_{i,j}|\, E|x_j|. \tag{26}$$

Therefore the ordering of expected influence values is determined directly by the absolute coefficient magnitudes $|W_{i,j}|$. For a neuron with pre-activation

$$p_i = \sum_{j=1}^{d} W_{i,j}x_j + b_i. \tag{27}$$

Since all features share the same distribution, the constant $E|x_j|$ is identical for all j. Therefore, according to Eq. (9) the largest absolute coefficient $W_{i,j}$ produces the largest expected influence value $E[\mu_{i,j}]$. If one coefficient dominates, the resulting normalized distribution becomes sharply concentrated and corresponds to a monosemantic neuron. If two coefficients dominate with similar magnitude, the normalized distribution contains two dominant entries corresponding to a polysemantic neuron. If coefficients are distributed across many variables with similar magnitude, the normalized distribution becomes diffuse and exhibits higher entropy. Applying this reasoning to the synthetic neuron definitions, neuron p_1 is expected to be monosemantic with respect to x_1, neuron p_2 is expected to be monosemantic with respect to x_2, neuron p_3 is expected to be polysemantic with respect to x_1 and x_2, and neuron p_4 is expected to produce a diffuse influence pattern.

The synthetic architecture also allows verification of layer-to-layer semantic tracing. The output equation indicates that $\phi(p_1)$, $\phi(p_2)$, and $\phi(p_3)$ contribute strongly to the prediction while $\phi(p_4)$ contributes weakly. Consequently the dominant semantic paths are

$$x_1 \to p_1 \to \hat{y}, \tag{28}$$

$$x_2 \to p_2 \to \hat{y}, \tag{29}$$

$$(x_1, x_2) \to p_3 \to \hat{y}. \tag{30}$$

As observed from Table 17, the first two hidden neurons, are correctly identified as monosemantic, the third neuron is identified as polysemantic, and the fourth neuron, which exhibits the highest entropy, as diffuse. The recovered semantic tracing from hidden neurons to the output also matches the known compositional structure of the generating equation. These results provide a minimal ground-truth validation that the proposed influence metric, entropy-based categorization, and cross-layer tracing behave correctly in a controlled regression setting.

Table 17
Controlled synthetic regression experiment with known neuron roles.

Neuron	Dominant features	Expected role	Observed entropy	Recovered role	Trace to output
p_1	x_1	Monosemantic	Low	Monosemantic	Strong
p_2	x_2	Monosemantic	Low	Monosemantic	Strong
p_3	x_1, x_2	Polysemantic	Intermediate	Polysemantic	Strong
p_4	$x_1, \ldots, x_8$	Diffuse	High	Diffuse	Weak

Table 18
Actual results from the controlled synthetic regression experiment after one training epoch.

Neuron	Top recovered features	Observed entropy	Z-score	Recovered role	Output trace
p_1	x_1 (0.954)	0.2597	−0.8833	Monosemantic	Strong
p_2	x_2 (0.937)	0.3343	−0.7799	Monosemantic	Strong
p_3	x_1 (0.497), x_2 (0.452)	0.9312	0.0479	Polysemantic	Strong
p_4	x_1 (0.168), x_2 (0.146), x_3 (0.138)	2.0613	1.6153	Dead	Weak

To provide further empirical evidence, we trained the synthetic model for one epoch on data generated from Eq. (1). The resulting neuron-level analysis is reported in Table 18. The observed normalized influence distributions remain aligned with the intended design: p_1 is dominated by x_1, p_2 is dominated by x_2, p_3 is jointly dominated by x_1 and x_2, and p_4 exhibits the highest entropy with a diffuse influence pattern across variables.

These empirical results support the theoretical claim that the proposed influence metric recovers the known semantic structure in a controlled setting, confirming the expected behavior and showing that the entropy-based categorization aligns with the predefined monosemantic, polysemantic, and diffuse neuron design. This synthetic validation complements the real-world experiments by demonstrating that the interpretability mechanism correctly identifies feature relevance, neuron roles, and semantic pathways when the true generating structure is known.

5. Conclusion

This study introduces Linear Lens, a non-interventional framework for faithfully explaining the behavior of deep neural networks without modifying weights or activations. The approach quantifies feature–neuron influence, classifies neurons by semantic role, and traces information flow across layers to produce interpretable, model-agnostic explanations. Experiments on ten real-world energy datasets, revealed consistent patterns: the first layers are predominantly polysemantic (over 90%), deeper layers converge to stable unimodal–multimodal distributions, and compositional outputs aggregate multiple feature pathways. A controlled user study confirmed that the produced explanations are clear, actionable, and enhance user trust. These findings demonstrate that Linear Lens enables transparent model auditing, debugging, and governance across domains. Future work will extend the framework to include convolutional and graph-based architectures, further broadening its applicability for the deployment of trustworthy AI.

CRediT authorship contribution statement

Muhammad Umair Danish: Conceptualization, Methodology, Formal analysis, Investigation, Software, Validation, Visualization, Writing – original draft. **Memoona Aziz:** Methodology, User study, Validation, Writing – review & editing. **Umair Rehman:** User study, Validation, Writing – review & editing. **Katarina Grolinger:** Supervision, Funding acquisition, Project administration, Writing – review & editing.

Declaration of Generative AI

During the preparation of this manuscript, the authors used AI-assisted tools, including Grammarly and ChatGPT (OpenAI), solely to support language editing, grammar correction, and improvement of readability. All AI-assisted suggestions were carefully reviewed, edited, and validated by the authors. The authors take full responsibility for the originality, accuracy, and integrity of the content of the published article.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Katarina Grolinger reports financial support was provided by work was supported in part by the Climate Action and Awareness Fund [EDF-CA-2021i018, Environnement Canada, K. Siddiqui and K. Grolinger] and in part by the Canada Research Chairs Program [CRC-2022-00078, K. Grolinger]. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

References

Aziz, M., Rehman, U., Safi, S. A., & Abbasi, A. Z. (2024). Visual verity in AI-generated imagery: Computational metrics and human-centric analysis. arXiv preprint arXiv:2408.12762.

Bereska, L., & Gavves, S. (2025). Mechanistic interpretability for AI safety-a review. *Transactions on Machine Learning Research*.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N. L., Anil, C., Denison, C., Askell, A., et al. (2023). Towards monosemanticity: Decomposing language models with dictionary learning. Oct 4 2023. URL https://Transformer-Circuits.pub/2023/monosemantic-features.anthropic.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., et al. (2025). Towards monosemanticity: Decomposing language models with dictionary learning. 2023. (p. 9). URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Casalicchio, G., Molnar, C., & Bischl, B. (2019). Visualizing the feature importance for black box models. In *Machine learning and knowledge discovery in databases: European conference, ECML pKDD 2018, dublin, Ireland, September 10–14, 2018, proceedings, part i 18* (pp. 655–670). Springer.

Center, P. R. (2025). How the U.S. public and AI experts view artificial intelligence. URL https://www.pewresearch.org/internet/2025/04/03/how-the-us-public-and-ai-experts-view-artificial-intelligence/. (Accessed: 17 July 2025).

Chandrasekaran, A., Prabhu, V., Yadav, D., Chattopadhyay, P., & Parikh, D. (2018). Do explanations make VQA models more predictable to a human? In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 1036–1042).

Conmy, A., Mavor-Parker, A., Lynch, A., Heimersheim, S., & Garriga-Alonso, A. (2023). Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, *36*, 16318–16352.

Craver, C. F. (2007). *Explaining the Brain: Mechanisms and the Mosaic Unity of Neuroscience*. Clarendon Press.

Cremades, A., Hoyas, S., & Vinuesa, R. (2025). Additive-feature-attribution methods: A review on explainable artificial intelligence for fluid dynamics and heat transfer. *International Journal of Heat and Fluid Flow*, *112*, Article 109662.

Danish, M. U., Ali, K., Siddiqui, K., & Grolinger, K. (2025). Physics-guided memory network for building energy modeling. *Energy and AI*, Article 100538.

Danish, M. U., & Grolinger, K. (2024). Leveraging hypernetworks and learnable kernels for consumer energy forecasting across diverse consumer types. *IEEE Transactions on Power Delivery*.

Fekri, M., Grolinger, K., & Mir, S. (2023). Asynchronous adaptive federated learning for distributed load forecasting with smart meter data. *International Journal of Electrical Power & Energy Systems*, accepted.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, *28*(1–2), 3–71.

Gehrmann, S., Strobelt, H., Krüger, R., Pfister, H., & Rush, A. M. (2020). Visual interaction with deep learning models through collaborative semantic inference. *IEEE Transactions on Visualization and Computer Graphics*.

Goldowsky-Dill, N., MacLeod, C., Sato, L., & Arora, A. (2023). Localizing model behavior with path patching. arXiv preprint arXiv:2304.05969.

Hase, P., & Bansal, M. (2020). Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5540–5552).

Heimerl, A., Weitz, K., Baur, T., & André, E. (2022). Unraveling ML Models of Emotion With NOVA: Multi-Level Explainable AI for Non-Experts. *IEEE Transactions on Affective Computing*.

Hewitt, J., & Liang, P. (2019). Designing and interpreting probes with control tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)* (pp. 2733–2743).

Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., & Liang, P. (2020). Concept bottleneck models. In *International conference on machine learning* (pp. 5338–5348). PMLR.

Kornblith, S., Norouzi, M., Lee, H., & Hinton, G. (2019). Similarity of neural network representations revisited. In *International conference on machine learning* (pp. 3519–3529). PMlR.

L'Heureux, A., Grolinger, K., & Capretz, M. A. (2022). Transformer-based model for electrical load forecasting. *Energies*.

Li, K., Patel, O., Viégas, F., Pfister, H., & Wattenberg, M. (2023). Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, *36*, 41451–41530.

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc..

Miller, C., Kathirgamanathan, A., Picchetti, et al. The building data genome project 2, energy meter data from the ASHRAE great energy predictor III competition. *Scientific Data*.

Montag, C., Becker, B., & Li, B. J. (2024). On trust in humans and trust in artificial intelligence: a study with samples from Singapore and Germany extending recent research. *Computers in Human Behavior: Artificial Humans*, Article 100070.

Mueller, A., Geiger, A., Wiegreffe, S., Arad, D., Arcuschin, I., Belfki, A., Chan, Y. S., Fiotto-Kaufman, J. F., Haklay, T., et al. (2025). MIB: A mechanistic interpretability benchmark. In *Forty-second international conference on machine learning*. URL https://openreview.net/forum?id=sSrOwve6vb.

Nauta, M., Trienes, J., Pathak, S., Nguyen, E., Peters, M., Schmitt, Y., Schlötterer, J., Van Keulen, M., & Seifert, C. (2023). From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Computing Surveys*.

Noorchenarboo, M., & Grolinger, K. (2025). Explaining deep learning-based anomaly detection in energy consumption data by focusing on contextually relevant data. *Energy and Buildings*.

Oberste, L., & Heinzl, A. (2023). User-Centric Explainability in Healthcare: A Knowledge-Level Perspective of Informed Machine Learning. *IEEE Transactions on Artificial Intelligence*.

Palumbo, N., Mangal, R., Wang, Z., Vijayakumar, S., Pasareanu, C. S., & Jha, S. (2025). Validating mechanistic interpretations: An axiomatic approach. In *Forty-second international conference on machine learning* (pp. 47509–47544).

Pan, J.-S., Wang, G. L., Chu, S. C., Yang, D., & Snášel, V. (2024). New feature attribution method for explainable aspect-based sentiment classification. *Knowledge-Based Systems*.

Poerner, N., Roth, B., & Schütze, H. (2018). Evaluating neural network explanation methods using hybrid documents and morphological agreement. arXiv preprint arXiv:1801.06422.

Ribeiro, M. T., Singh, S., & Guestrin, C. (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* (pp. 1135–1144).

Rizk, Y., Awad, M., & Tunstel, E. W. (2018). Decision making in multiagent systems: A survey. *IEEE Transactions on Cognitive and Developmental Systems*.

Rong, Y., Leemann, T., Nguyen, T. T., Fiedler, L., Qian, P., Unhelkar, V., Seidel, T., Kasneci, G., & Kasneci, E. (2024). Towards human-centered explainable AI: A survey of user studies for model explanations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Sanneman, L., & Shah, J. A. (2022). An Empirical Study of Reward Explanations With Human-Robot Interaction Applications. *IEEE Robotics and Automation Letters*.

Saw, S. N., Yan, Y. Y., & Ng, K. H. (2025). Current status and future directions of explainable artificial intelligence in medical imaging. *European Journal of Radiology*, *183*, Article 111884.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*(3), 379–423.

Som, S., Majumdar, R., Ghosh, M., & Malkani, C. (2017). Statistical analysis of student feedback system using cronbach's alpha and utility measurement process. In *2017 international conference on infocom technologies and unmanned systems (trends and future directions)*.

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. In *International conference on machine learning*. PMLR.

Tjoa, E., & Guan, C. (2021). A survey on explainable artificial intelligence (XAI): Toward medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*.

Tuan, K. T. D., Trong, T. N., Hoang, S. N., Than, K., & Duc, A. N. (2025). Weighted integrated gradients for feature attribution. arXiv preprint arXiv:2505.03201.

Wani, N. A., Kumar, R., & Bedi, J. (2024). DeepXplainer: An interpretable deep learning based approach for lung cancer detection using explainable artificial intelligence. *Computer Methods and Programs in Biomedicine*, *243*, Article 107879.

Wani, N. A., Kumar, R., Bedi, J., Rida, I., et al. (2024). Explainable AI-driven IoMT fusion: Unravelling techniques, opportunities, and challenges with explainable AI in healthcare. *Information Fusion*, *110*, Article 102472.

Williams, I., Oldenburg, N., Dhar, R., Hatherley, J., Fierro, C., Rajcic, N., Schiller, S. R., Stamatiou, F., & Søgaard, A. (2025). Mechanistic interpretability needs philosophy. arXiv preprint arXiv:2506.18852.

Yeh, C. K., Kim, B., Arik, S., Li, C. L., Pfister, T., & Ravikumar, P. (2020). On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, *33*, 20554–20565.

Yin, M., Wortman Vaughan, J., & Wallach, H. (2019). Understanding the effect of accuracy on trust in machine learning models. In *Proceedings of the 2019 chi conference on human factors in computing systems* (pp. 1–12).

Zarlenga, M. E., Shams, Z., Nelson, M. E., Kim, B., & Jamnik, M. TabCBM: Concept-based interpretable neural networks for tabular data. *Transactions on Machine Learning Research*.

Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A.-K., et al. (2023). Representation engineering: A top-down approach to AI transparency. CoRR.