Measuring Concept Completeness for Explainability in AI-Driven Network Functions

Alexander Niedermayer, David Monschein, Oliver P. Waldhorst
Institute of Data-Centric Software Systems, Karlsruhe University of Applied Sciences, Karlsruhe, Germany
{alexander.niedermayer,david.monschein,oliver.waldhorst}@h-ka.de

***Abstract*—Network functions increasingly rely on machine learning models. Recent work–such as the Agua framework presented at SIGCOMM 2025–has shown that explanations grounded in networking concepts can improve the interpretability of these models. However, this requires a predefined set of concepts that is assumed to completely capture the model's internal reasoning, without offering any way to verify this assumption. As a result, explanations may appear plausible while omitting substantial parts of the learned representation. In this paper, we introduce a representation-level framework for quantifying concept completeness. Instead of evaluating completeness solely via prediction reconstruction, we measure how much variance in a model's internal representations is captured by the subspace spanned by predefined networking concepts. This yields a global, post-hoc, model-agnostic completeness metric. To address the high dimensionality and noise typical of large language model-based explanations, we further propose a significant-component completeness measure. Additionally, we derive an importance metric to fairly attribute each concept's contribution to overall completeness. We apply our approach to assess the completeness of the concept sets used to explain three AI-driven networking scenarios in the Agua paper. We find that the concept set is only moderately complete for two and substantially complete for one of these scenarios.**

***Index Terms*—AI-driven Networking, Explainable AI, Concept Explanations**

I. Introduction

Artificial Intelligence (AI)-driven network functions are a new and rapidly growing area in networking research and practice. By leveraging machine learning models, these functions can adapt to changing network conditions, optimize performance, enhance user experience and provide security-centric services. However, understanding how these models make decisions remains a challenge, particularly in terms of interpretability and explainability. The latter is crucial for building trust in AI-driven systems and improving their reliability.

In other domains, such as computer vision or natural language processing, interpretability methods have been developed to explain model decisions based on human-understandable concepts. Common techniques include feature importance methods and saliency maps. However, while being useful especially for image data, these methods do not directly translate to networking applications, where the data and features are often more abstract and less intuitive due to their numerical forms [1]. A promising approach to address this challenge is the use of concept-based interpretability methods [2], [3], [4], which aim to explain model decisions based on high-level concepts that are meaningful to humans. As an example in a networking context, one could take multiple traffic data series that exhibit frequent packet loss and use them as inputs to a model. Together, these inputs represent the higher-level concept "frequent packet loss". By observing the model's output for these inputs, it becomes possible to infer how the model internally responds when this condition is present, and which behaviors are influenced by sustained packet loss.

While the use of concept-based explanations in frameworks such as Agua [5] have shown promise in improving interpretability for AI-driven network functions, the authors note a key limitation: it remains unclear how comprehensively these concepts capture the model's behavior. Therefore the possibility remains that important aspects of the model's decision-making process are not captured by the predefined concepts, leading to incomplete or misleading explanations. Yeh et al. [6] introduced the notion of *concept completeness*, which assesses how well a set of predefined concepts can reconstruct the predictions of a model. However, this approach focuses solely on prediction-level reconstruction accuracy and does not quantify how much of the internal representation space is covered by the concepts. As a result, a concept set may be deemed complete even if it explains only a small or highly specific subspace of the learned features, while large portions of the latent representation remain unaccounted for. Furthermore, with the rise of Large Language Models (LLMs) and their application in networking tasks, there is a need to extend concept completeness methods to handle large and complex feature spaces in LLM-generated embeddings.

We address these gaps by proposing a novel method to quantify the concept completeness of AI-driven network functions at the representation level. Our approach measures how well a set of predefined concepts captures the variance in the model's internal representations, providing a more comprehensive assessment of concept completeness beyond prediction accuracy. Additionally, we extend our method to efficiently handle high-dimensional feature spaces common in LLM-based networking applications. Our contributions are summarized as follows:

This work was supported by the bwNET2.0 project, funded by the Ministry of Science, Research and the Arts Baden-Wurttemberg (MWK). The authors alone are responsible for the content of this paper.

ISBN 978-3-903176-82-9

- We propose an efficient method to quantify the concept completeness of AI-driven network functions, enabling a better understanding of how well predefined concepts are represented in the model's internal features.
- We extend the concept completeness method to handle large feature spaces typical in AI-driven networking applications and LLM-generated embeddings.
- We derive an importance measure for individual concepts based on their contribution to the global concept completeness.
- We evaluate our methods on three different AI-driven network scenarios, demonstrating their effectiveness in quantifying concept completeness and importance.

In our experiments we first demonstrate the validity of our method in a synthetic setup with known ground truth concept completeness. We then apply our method to three real-world AI-driven networking scenarios [5]: adaptive bitrate streaming, congestion control, and Distributed Denial-of-Service (DDoS) detection. We address the open question regarding the quality of base concepts raised in [5]. Substantial completeness is achieved for the congestion control concept set, whereas the concept sets for the remaining scenarios attain only moderate completeness.

The paper is structured as follows: In Section II, we review related work on concept-based explainability and concept completeness, as well as highlighting the gap in existing methods. In Section III, we present our proposed method for quantifying concept completeness and importance. In Section IV, we evaluate our method in a synthetic setup with known ground truth concept completeness. In Section V, we apply our method to three real-world AI-driven networking scenarios. Finally, in Section VI, we conclude the paper and discuss future research directions.

II. Related Work

A. Concept-Based Explainability

Concept-based explainability aims to bridge the gap between low-level model representations and human-understandable abstractions. The idea evolves around using multiple data series sharing a similar concept (e.g. average high packet rate or frequent packet loss), projecting these into the latent space of the model and afterwards using the similarity to these vectors as metric how much the given concept impacted the model decision. An example for early work using this idea is Testing with Concept Activation Vectors (TCAV) [2]. Subsequent approaches extended this idea by learning linear probes or interpretable classifiers on top of internal representations to associate neurons or layers with predefined concepts [7], [8]. Complementary to post-hoc methods, concept bottleneck models explicitly incorporate concepts as intermediate variables during training, thereby enforcing interpretability by design [3].

Recently in the context of AI-driven networking, concept-based explanations have been proposed to improve trust and debuggability of learning-enabled controllers. Prior work introduces domain-specific concepts for tasks such as adaptive bitrate streaming, congestion control, and DDoS detection, and demonstrates how these concepts can be used to generate human-interpretable explanations of model behavior [5].

While providing valuable insights into model decisions, all mentioned approaches implicitly assume that the predefined concepts are representative of the model's internal reasoning processes. This assumption is also acknowledged as a limitation in prior work on AI-driven networking, where suboptimal base concepts are reported to degrade explanation fidelity [5]. They largely focus on local or prediction-level explanations, providing limited insight into how comprehensively a set of predefined concepts captures the overall structure of a model's learned representations.

B. Concept Completeness

Yeh et al. [6] introduced the notion of *concept completeness*, providing an important theoretical step toward assessing the sufficiency of concept sets. In their framework, a set of concepts is considered complete if a classifier operating on those concepts can reconstruct the predictions of the original model with comparable performance. This formulation offers a principled way to compare different concept sets and to reason about whether a given explanation vocabulary is adequate for a specific task.

Despite its significance, concept completeness as defined by Yeh et al. is inherently prediction-centric. Completeness is evaluated solely based on the ability to reproduce model outputs, without explicitly considering how concepts align with the internal representation space of the model. As a consequence, a concept set may be deemed complete even if it explains only a small or highly specific subspace of the learned features, while large portions of the latent representation remain unaccounted for. Moreover, the framework does not quantify how much variance or structure in the internal representations is captured by concepts, nor does it provide a global scalar measure of representation-level alignment.

C. Gap Addressed by This Work

This work addresses the previously mentioned limitation by introducing a representation-level notion of concept completeness. Instead of scoring concepts by output reconstruction accuracy, our concept completeness measures how much variance in the model's internal representations is captured by the subspace spanned by concept activation vectors. This yields a single global scalar that describes latent-space alignment and can be reported under a standardized procedure across different models or settings without requiring prediction reconstruction.

In addition, we extend concept completeness analysis to high-dimensional embeddings by focusing on significant components of the representation (via Singular Value Decomposition (SVD)/Principal Component Analysis (PCA)-style variance thresholds), enabling more interpretable and robust reporting when representations are large, sparse, or noisy—common for LLMs. The latter has a direct impact on the usage for AI-driven networking scenarios, since recent research has

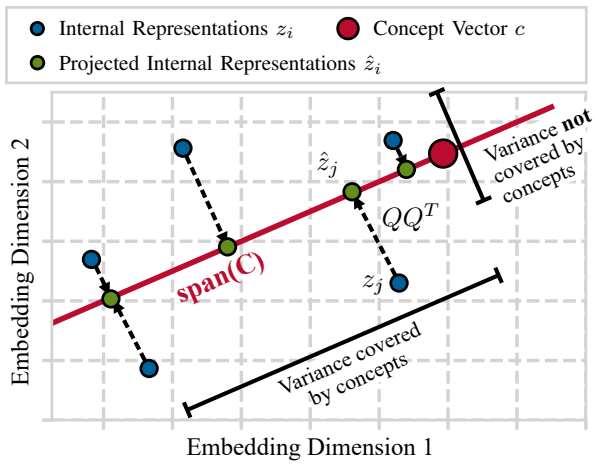

Fig. 1. Visualization of the main steps for our completeness method.

shown that LLMs deliver promising results for concept-based explanations in this domain [5].

Finally, this work introduces a method for attributing concept importance using Shapley values computed over the proposed concept completeness metrics. This allows for fair ranking and prioritization of individual concepts based on their marginal contributions to overall representation alignment, enabling more informed concept selection and pruning strategies.

III. On the Calculation of Concept Completeness

In this section, we present our framework designed to evaluate the completeness of concepts within neural network models applied to networking tasks. For our framework, we define the following four requirements for computing the global concept completeness and the significant component completeness:

1) A trained neural network model for a specific networking task.
2) A dataset representative of the task domain, used to probe the model's internal representations. Since the internal representations must match what the model sees in operation, using the test dataset is recommended.
3) A method for extracting internal representations from the neural network model.
4) A set of predefined concepts relevant to the networking task, along with corresponding concept activation vectors.

All algorithms are implemented as post-hoc analysis, allowing them to be applied to any trained model without requiring modifications to the training process.

A. General Concept Completeness

To assess the general concept completeness of a neural network model, we propose a metric that quantifies the extent to which the model's internal representations align with a predefined set of concepts relevant to networking tasks. The main idea behind our method is illustrated in Figure 1.

We define the dataset representative of the task domain as $X = \{x_1, x_2, \ldots, x_N\} \subseteq \mathcal{X}^{d_x}$, where $\mathcal{X}$ is the data type (e.g., $\mathbb{R}$ for real numbers) and d_x is the dimensionality of the input data. The trained model consists of a function $\Phi : \mathcal{X}^{d_x} \to \mathcal{X}^{d_z}$ mapping input data into an intermediate layer and a function $h : \mathcal{X}^{d_z} \to \mathcal{X}^{d_y}$ mapping the intermediate representation to a final output (e.g. classification, regression). The full model is then given by $h \circ \Phi$. Finally, the set of predefined concept activation vectors is denoted as $C = \{c_1, c_2, \ldots, c_K\} \subseteq \mathcal{X}^{d_z}$.

For every input sample $x_i \in X$ $i \in \{1, 2, \ldots, N\}$, we compute the internal representation $z_i = \Phi(x_i)$, giving us a set of internal representations $Z = \{z_1, z_2, \ldots, z_N\}$ for the entire dataset X.

On this set of representations, we calculate the variance for each dimension across all samples. The total variance is then given by the sum of the variances across all dimensions:

$$\text{Var}_{\text{global}} = \frac{1}{N} \sum_{i=1}^{N} \|z_i - \mu_z\|_2^2. \tag{1}$$

Here, $\mu_z = \frac{1}{N} \sum_{i=1}^{N} z_i$ denotes the mean of the internal representations. To evaluate the completeness of all predefined concepts C within the model's internal representations, we project each representation z_i onto the subspace spanned $span(C) \subseteq \mathcal{X}^{d_z}$ by the concept activation vectors $\{c_1, c_2, \ldots, c_K\}$. The projection is computed using QR decomposition to obtain an orthonormal basis $Q \in \mathcal{X}^{d_z \times K}$ for $span(C)$ with $C = QR$ [9]. The projected representation $\hat{z}_i$ is then given by $\hat{z}_i = z_i Q Q^T$. Afterwards, we calculate the variance explained by the concepts as the variance of the representations projected onto the concept subspace $span(C)$:

$$\text{Var}_{\text{covered},C} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{z}_i - \mu_{\hat{z}}\|_2^2. \tag{2}$$

Finally the global concept completeness κ_C for a set of concepts C is defined as the ratio of the variance explained by the concepts to the total variance in the internal representations. When using all concepts, we denote the global concept completeness as κ^*.

$$\kappa^* = \frac{\text{Var}_{\text{covered},C}}{\text{Var}_{\text{global}}} \in [0, 1]. \tag{3}$$

A higher ratio indicates that a larger portion of the model's internal representations can be explained by the predefined concepts, suggesting better concept completeness. A value close to 1 implies that the concepts effectively capture the essential features learned by the model for the networking task.

The computational complexity of the algorithm can be attributed to three main parts: (1) the computation of the variance of the internal representations given by $O(Nd_z)$, (2) the QR decomposition of the concept matrix C with complexity $O(d_z K^2)$, and (3) the projection of the internal representations onto the concept subspace with complexity $O(Nd_z K)$. Thus, the overall computational complexity of the

algorithm is given as $O(Nd_z + d_zK^2 + Nd_zK)$ or simplified as $O(Nd_zK + d_zK^2)$.

While the algorithm is sufficient for evaluating small dimensional representation spaces ($d_z \leq 256$), higher-dimensional spaces tend to be sparse with individual dimensions encoding increasingly fine-grained features, making it difficult to achieve high concept completeness. To address this issue, we propose an extension to the algorithm for these scenarios in Section III-B.

B. Concept Explanation of Significant Components

To enhance the interpretability of neural network models in networking tasks, we extend our algorithm to provide concept explanations for significant components within the model's internal representations. This extension focuses on identifying and explaining the most influential dimensions that contribute to the model's decision-making process. We define significant components as the top k dimensions of the internal representation space that exhibit the highest variance across the dataset.

We select the top k dimensions based on the singular values of the internal representations Z. This enables us to steadily decrease the number of dimensions, giving us a trade-off between variance per dimension and the number of dimensions to explain. Formally, we perform SVD on the representations:

$$Z = U\Sigma V^T. \tag{4}$$

Here, $U \in K^{N \times d_z}$ contains the left singular vectors, $\Sigma \in K^{d_z \times d_z}$ is a diagonal matrix with singular values, and $V \in K^{d_z \times d_z}$ contains the right singular vectors. The singular values in Σ indicate the variance captured by each corresponding dimension in the internal representation space. For the top k dimensions we choose the first k columns of V, until a predefined variance threshold τ is met, such that the cumulative variance explained by these dimensions exceeds τ percent of the total variance.

Afterwards, we project the internal representations z_i onto the subspace spanned by the top k singular vectors, resulting in the projected representations $z_{i,k}$. Furthermore, we also project the concept activation vectors c_i onto the same subspace, yielding $c_{i,k}$. This allows us to analyze how well the concepts cover the significant components of the internal representations.

We define the concept completeness for this reduced set of significant components in an analogous manner to the global concept completeness presented in the previous section. We again compute the total variance for the reduced internal representations and the variance when projecting the reduced representations onto the subspace spanned by the reduced concept activation vectors. Since the intensity of our dimensionality reduction process is now controlled by the variance threshold τ, we denote the concept completeness for a given variance threshold as $\kappa_C(\tau)$, with the corresponding variances defined as $\text{Var}_{global}(\tau)$ and $\text{Var}_{covered,C}(\tau)$. This allows us to express the concept completeness for a significant component threshold τ as:

$$\kappa^*(\tau) = \kappa_C(\tau) = \frac{\text{Var}_{covered,C}(\tau)}{\text{Var}_{global}(\tau)}. \tag{5}$$

For $\tau = 100\ \%$, this metric converges to the global concept completeness defined in the previous section. By analyzing the concept completeness across different variance thresholds, we can gain insights into how well the set of predefined concepts explains the most significant features learned by the model for the networking task. The value of significant component completeness τ^* can then be defined as the highest variance threshold where a desired concept completeness $1 - \varepsilon$ is achieved. ε ($\varepsilon > 0$ and $\varepsilon \in \mathbb{R}$) is used as small tolerance value to account for numerical inaccuracies. Formally, we define τ^* for a concept set C as:

$$\tau^*_C = \arg\max_{\tau} \{\tau \mid \kappa_C(\tau) \geq 1 - \varepsilon\}. \tag{6}$$

If no concept space C is given, we denote $\tau^* = \tau^*_C$ when using all concepts C. A higher value of τ^*_C indicates that the predefined concepts effectively capture the essential features learned by the model. This extension of our algorithm provides a more nuanced understanding of concept completeness, particularly in scenarios where the internal representation space is high-dimensional and noisy.

Based on [10], we define the following taxonomy to classify the concept completeness of significant components:

- Comprehensive Completeness: $\tau^* \geq 90\ \%$
- Substantial Completeness: $70\ \% \leq \tau^* < 90\ \%$
- Moderate Completeness: $50\ \% \leq \tau^* < 70\ \%$ -
- Limited Completeness: $\tau^* < 50\ \%$
- No Completeness: τ^* does not exist - The concepts do not explain any significant components, suggesting a complete misalignment between the concepts and the model's learned features, with the concept space being effectively orthogonal to the internal representations.

The extended algorithm requires computing a SVD with complexity $O(Nd_z^2)$, in addition to the complexities of variance computation and projection from the previous section. Thus, the overall computational complexity of the extended algorithm is given as $O(Nd_z^2 + Nd_zK + d_zK^2)$. Since $\kappa_C(\tau)$ is monotonically decreasing (cf. Section VI), we use binary search for calculating τ^*_C.

C. Estimating Concept Importance

The importance of individual concepts can be estimated using their contribution to the explained variance in the internal representations. For this we define κ_S as the concept completeness when using a subset of concepts $S \subseteq C$. To estimate the importance of each concept $c_i \in C$, we evaluate the marginal contribution of c_i to the concept completeness when added to different subsets of concepts.

We use the Shapley value from cooperative game theory [11] to fairly attribute the contribution of each concept to the overall concept completeness. As value function, we use the concept completeness κ defined in the previous sections

for a given set of concepts $S \subseteq C$. The Shapley value ϕ_i for a concept $c_i \in C$ is computed as:

$$\phi_i = \frac{1}{K} \sum_{S \subseteq C \setminus \{c_i\}} \binom{K-1}{|S|}^{-1} \left[\kappa_{S \cup \{c_i\}} - \kappa_S\right]. \quad (7)$$

where $K = |C|$ is the total number of concepts. We therefore obtain ϕ_i as the average marginal contribution of concept c_i across all possible subsets of concepts. A higher Shapley value indicates that the concept is more important in explaining the model's internal representations. Additionally, the Shapley values for the significant component completeness τ^* can be computed in an analogous manner by reducing the dimensionality until τ^* is met and afterwards performing the importance estimation in a similar manner.

The computational complexity of calculating the Shapley values grows exponentially is given as $O(2^K)$, due to the need to evaluate all possible subsets of concepts. Usually, concepts are kept small ($K < 15$) to allow for a feasible computation. For larger sets of concepts, approximation methods such as Monte Carlo sampling [12] can be employed to estimate the Shapley values with reduced computational overhead.

IV. Method Evaluation

A. Validation of Concept Completeness Method

To validate the effectiveness of our proposed concept completeness method, we conduct a series of experiments in a controlled synthetic environment. In this setup, we generate latent feature representations with known concept structures, allowing us to compare the calculated concept completeness scores against ground truth values.

To simulate the latent space produced by a model, we first define a set of known C_K and unknown C_U orthogonal concept vectors. The n-th concept vector is generated as $c_n = (0, ..., 0, 1, 0, ..., 0)^T$, where the 1 is located at the n-th position. This ensures orthogonality between the concept vectors. Since models often do not fully utilize all dimensions we set the latent space dimensionality d_z to be larger than the number of concepts, i.e., $d_z > |C_K| + |C_U|$, so that the remaining dimensions act as unused or noise-like components.

An embedding vector z is then constructed to mimic a model's representation by taking a linear combination of all concept vectors with random weights w_n for each concept, plus Gaussian noise $\mathcal{N}(0, \sigma^2 I_{d_z})$:

$$z = \sum_{c_n \in C_K \cup C_U} w_n \cdot c_n + \mathcal{N}(0, \sigma^2 I_{d_z}). \quad (8)$$

The concept completeness is then calculated using our proposed method with the known concepts C_K on a set of generated embedding vectors. We vary two main parameters in our experiments: the noise level σ and the amount of known concepts $|C_K|$ relative unkown concepts $|C_U|$:

- 0 %: All concepts unknown $C_K = \{\ \}$
- 50 % Equal known and unknown concepts $|C_K| = |C_U|$
- 100 %: All concepts known $C_U = \{\ \}$

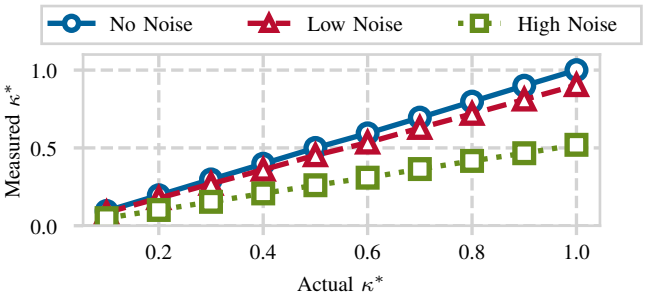

Fig. 2. Comparison between the calculated concept completeness scores and the ground truth concept completeness in a synthetic evaluation setup with varying noise levels and constant latent space dimension.

This allows us to observe how the concept completeness scores change under different conditions.

Figure 2 shows the results of our experiments when varying the noise level σ and the known concepts ratio while keeping the representation space dimensionality constant at $d_z = 256$. As expected, without any noise ($\sigma = 0$), the concept completeness scores match the ground truth concept completeness across all known concepts ratios. As the noise level increases ($\sigma = 0.1$ and $\sigma = 1$), the concept completeness scores decrease smoothly, indicating that our method effectively captures the degradation in concept representation due to noise.

Futhermore, one can observe that the concept completeness scores strictly monotonically increase with the known concepts ratio for all noise levels. This behavior aligns with our expectations, as a higher proportion of known concepts should lead to better completeness in the latent space.

B. Addressing Latent Space Dimensionality

Since most LLMs utilize very high-dimensional latent feature spaces, we further evaluate the performance of our concept completeness method when varying the latent space dimensionality d_z. In this experiment, we keep the noise level constant at $\sigma = 0.01$ and vary the latent space dimensionality from $d_z = 16$ to $d_z = 4096$, while reusing the latent space creation from Section IV-A. The known concepts ratio is also varied as in the previous experiment for three different levels (33 %, 66 %, 100 %). The results are presented in Figure 3.

As shown in Figure 3, the global concept completeness scores decrease when increasing the latent space dimensionality for all known concepts ratios. While still being usable for moderate latent space dimensions ($d \leq 512$), the concept completeness scores drop significantly for very high-dimensional latent spaces ($d \geq 1024$). This behavior can be attributed to the increased difficulty of accurately capturing concept representations in higher-dimensional spaces, where the signal-to-noise ratio gets lower.

As last experiment in this validation section, we compare the deviation between the measured completeness score and the ground truth completeness score for both the global concept completeness and the significant concept completeness

introduced in Section III-B. We vary the noise level σ from 0 to 1 in a high-dimensional latent space with $d_z = 2048$ and a known concepts ratio of 50 %. The results are presented in Figure 4.

The global concept completeness score shows no deviation when there is no noise present ($\sigma = 0$). However, as the noise level increases, the deviation between the measured completeness score and the ground truth completeness score also increases significantly. This indicates that the global concept completeness method struggles to accurately capture concept representations in high-dimensional latent spaces with substantial noise. In contrast, the significant concept completeness method exhibits a small deviation with little noise levels present, but lower deviations even for higher noise levels. This demonstrates the effectiveness of the significant concept completeness method in handling high-dimensional latent spaces with noise, providing a more reliable measure of concept completeness under these challenging conditions.

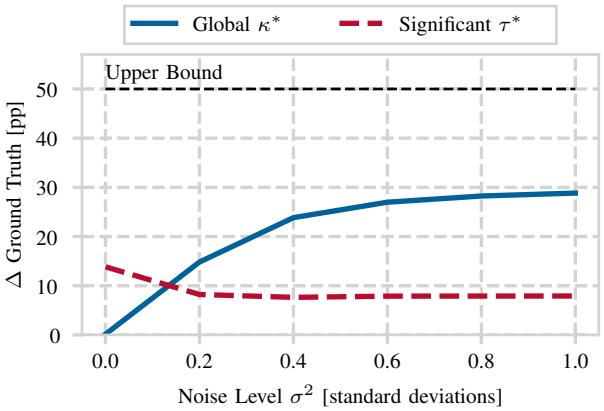

Fig. 4. Comparison between the mean deviation of global concept completeness scores κ^* and the significant concept coverage scores τ^* to the ground truth completeness for varying noise levels in an synthetic evaluation setup with high-dimensional latent spaces. The deviation is given in percentage points (pp).

V. AI-Driven Network Function Evaluation

In the following section we apply our method to estimate the concept completeness of three different AI-driven network functions [5].

A. AI-Driven Network Evaluation Scenarios

The three AI-driven concept-aware network scenarios are taken from Patel et al. [5]. Patel et al. present these use cases to demonstrate the applicability of concept-based explanations in networking contexts. The three scenarios are briefly described below.

1) Scenario: Adaptive Bitrate Streaming: The first scenario involves an Adaptive Bitrate Streaming (ABR) system [13] that dynamically adjusts video quality based on network conditions. The system utilizes a deep reinforcement learning model as controller to optimize user experience by selecting appropriate video bitrates. There are 16 predefined concepts related to network conditions (volatile network throughput, recent network improvement, ...), content characteristics (low content complexity, content requiring high quality, ...) and client information (startup of video, stable buffer, ...).

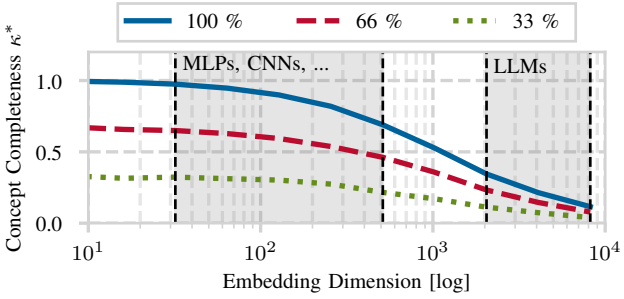

Fig. 3. Comparison between the calculated concept completeness scores and the ground truth concept completeness in a synthetic evaluation setup with varying latent space dimensions and constant noise level.

2) Scenario: Congestion Control: The second scenario focuses on AI-driven Congestion Control (CC) as presented by Jay et al. [14]. The AI-driven congestion control model aims to optimize data transmission rates in response to varying network conditions. The model is trained using reinforcement learning techniques to adaptively manage congestion and improve overall network performance. There are 8 predefined concepts related to network conditions (increasing latency, increasing packet loss) and traffic characteristics (high network utilization, ...).

3) Scenario: Distributed Denial-of-Service Detection: The last scenario considers an AI-driven DDoS detection system [15]. The system employs a neural network-based model to identify and mitigate DDoS attacks in real-time. The model analyzes network traffic patterns and distinguishes between legitimate and malicious activities. There are 10 predefined concepts related to request characteristics (protocol anomalies, high request rate, ...) and client behavior (geographical distribution, typical application behavior, ...).

B. Concept Completeness Results

For every scenario, we include 5000 input samples. The experiments were performed on an AMD EPYC 7742 CPU (64 Cores@2.8 GHz) and one NVIDIA A100 80 GB Graphics Processing Unit (GPU). The algorithms were implemented using Python 3.12 and CuPy 13.6.0 providing GPU-accelerated computing.

Calculating the global concept completeness scores for the three AI-driven network scenarios, as given in Section III-A, results in the values presented in Table I. We calculate the concept completeness scores across five different folds of input data using the same concepts as defined by Patel et al. [5] for each scenario. The scores are given as mean values with standard deviation.

It is clearly visible that the concept completeness scores are relatively low across all three scenarios, with none of

TABLE I
GLOBAL CONCEPT COMPLETENESS SCORES κ^* FOR AI-DRIVEN NETWORK SCENARIOS.

Scenario	# Concepts	Completeness Score
Adaptive Bitrate Streaming	16	$13.33\ \% \pm 0.40\ \%$
Congestion Control	8	$8.65\ \% \pm 0.37\ \%$
DDoS Detection	10	$12.67\ \% \pm 1.03\ \%$

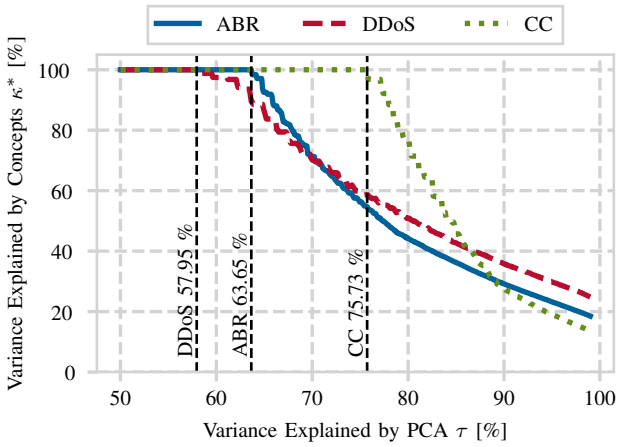

Fig. 5. Concept completeness scores for different variance thresholds in the three AI-driven network scenarios (Adaptive Bitrate Streaming (ABR), Congestion Control (CC), Distributed Denial-of-Service (DDoS)). Metrics are given as average across five folds.

them exceeding a completeness score of 15 %. The latter can be attributed to the high-dimensional latent spaces used by the models in these scenarios (3096 dimensions for all three scenarios). As shown in the validation experiments, high-dimensional latent spaces make it difficult to achieve high concept completeness scores due to the sparsity of the representation space. Since our algorithm's complexity scales quadratically with the number of concepts K, the runtimes remain low in practice because K is generally small (< 32). Moreover, we measure 127.42 ms $\pm$ 8.99 ms for ABR, 87.94 ms $\pm$ 10.58 ms for CC, and 50.35 ms $\pm$ 17.89 ms for DDoS detection.

To further analyze the concept completeness in these high-dimensional latent spaces, we apply the significant concept completeness method introduced in Section III-B. We calculate the concept completeness scores for different variance thresholds τ ranging from 50 % to 100 % in steps of 0.5 %. The results are presented in Figure 5.

As shown in Figure 5, the concept completeness scores increase significantly when lowering the variance threshold τ. For the CC scenario, a significant concept completeness score of 75.79 % is achieved, demonstrating *substantial completeness* according to our taxonomy. The ABR and DDoS scenarios achieve concept completeness scores of around 57.74 % and 62.89 %, respectively, indicating *moderate completeness*.

When comparing the number of concepts to the concept completeness scores as shown in Figure 6, it becomes evident that a small number of concepts can already achieve substantial completeness in high-dimensional latent spaces. For example, in the CC scenario, only 8 concepts are required to reach a significant concept completeness score of over 75 %. This highlights the effectiveness of our significant concept completeness method in identifying the most relevant concepts that contribute to the model's internal representations. Furthermore, it suggests that sometimes a small set of well-defined concepts is more valuable than a large set of less relevant ones, as seen in the ABR scenario.

The runtime is now mainly dominated by the SVD calculation, being quadratic in regards to the latent space dimensionality d_z. Latent space dimensionality for these scenarios is set at $d_z = 3096$. We are locating the best threshold τ^* using binary search with 10 iterations (cf. Theorem 1), further improving computation time, the measured times are 48.06 s $\pm$ 0.11 s for ABR, 1.75 s $\pm$ 0.28 s for CC and 1.81 s $\pm$ 0.01 s for DDoS detection.

C. Concept Importance Results

Using the concept importance method introduced in Section III-C, we calculate the importance scores for each predefined concept in the three AI-driven network scenarios. The importance can be calculated for two variance thresholds, $\tau = 100\ \%$ as global importance measure and τ^* for the significant importance measure. When comparing the concept importance scores for both thresholds, we observe a different distribution of importance across the concepts. The results for the significant concept completeness using the CC and DDoS scenario are presented in Figure 7.

In Figure 7, the concepts "increasing packet loss" and "rapidly increasing latency" show a substantial increase in importance when using the significant threshold τ^*. This aligns when considering that packet loss is the most critical factor in rule-based congestion control algorithms such as Transmission Control Protocol (TCP) Tahoe, TCP Reno and TCP Cubic [16], [17], [18]. Furthermore, the concept of

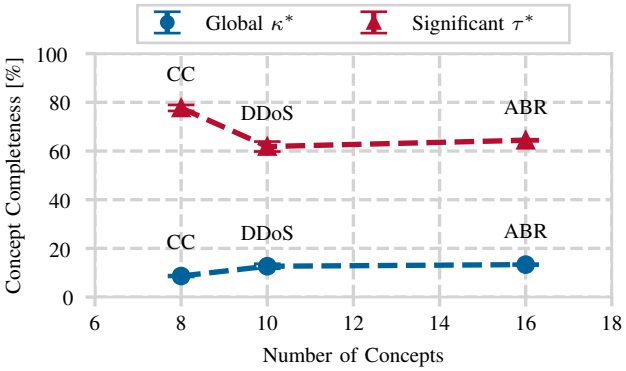

Fig. 6. Comparison of the number of concepts to the global and significant concept completeness scores for different variance thresholds in the three AI-driven network scenarios (Adaptive Bitrate Streaming (ABR), Congestion Control (CC), Distributed Denial-of-Service (DDoS)).

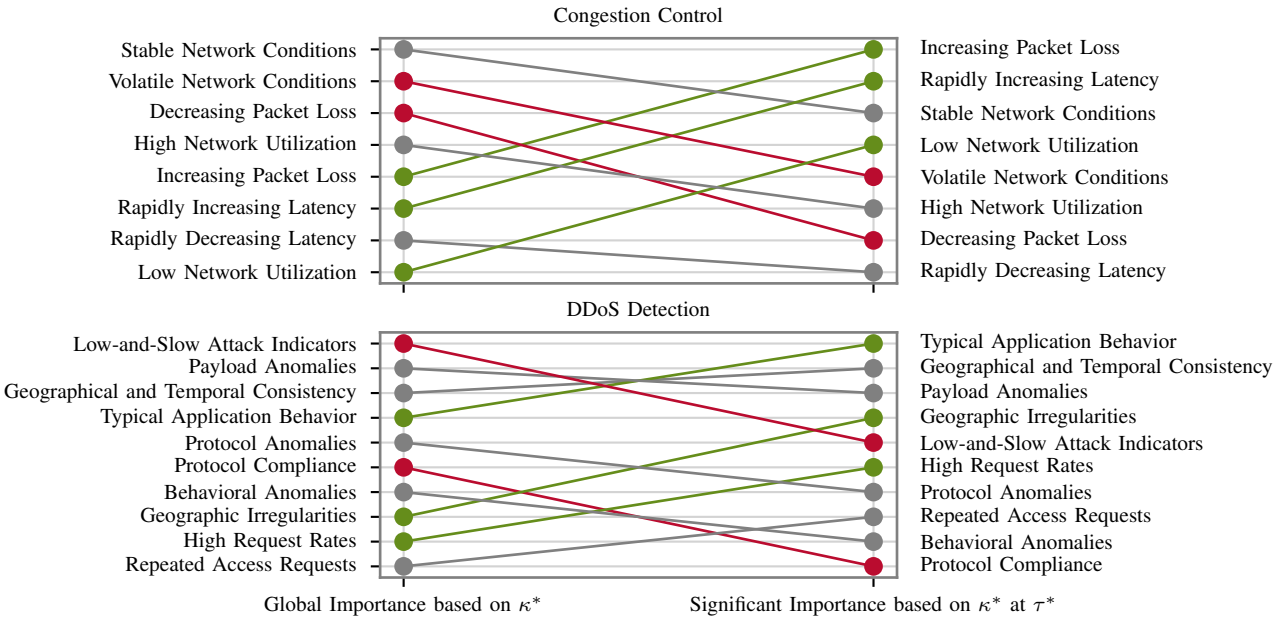

Fig. 7. Concept importance scores for the significant concept completeness in the AI-driven congestion control and Distributed Denial-of-Service (DDoS) detection scenario. Concepts that deviate by less than 2 ranks between the global and significant concept completeness are not highlighted.

"rapidly increasing latency" is also crucial for congestion control, as it changes the bandwidth delay product.

For the DDoS detection scenario, the concept importance scores are also presented in Figure 7. Similar to the congestion control scenario, certain concepts show a change in their importance ranking when comparing the global and significant thresholds. For example, the concepts "typical application behavior", "geographic irregularities", and "high request rates" show a substantial increase in importance when using the significant threshold. This aligns with the understanding that these concepts are critical indicators of DDoS attacks, as they reflect abnormal patterns in network traffic that are often associated with such attacks. Furthermore, for the global concept completeness-based importance scores "low-and-slow attack indicators" is ranked highest, which is less intuitive since low-and-slow attacks are generally harder to detect and also less common than volumetric attacks [19], [20].

Although directly validating the concept importance scores is challenging due to the lack of a ground truth, the comparison between global and significant measurements reveals notable shifts in concept rankings. Importantly, the concepts that gain prominence at τ^* correspond closely to features known to be effective in rule-based systems. This alignment provides supporting evidence that the τ^*-based concept importance captures meaningful concepts that a well-optimized model is likely to leverage for strong performance, suggesting that it offers a more informative assessment of concept importance than the global measure.

As seen in Section III-C the computation time for the Shapley values is dominated exponentially by the number of concepts K. To handle the complexity, we used a Monte-Carlo-based approximation with 1024 iterations. The computing times were measured at 48.06 s $\pm$ 0.11 s for ABR, 1.75 s $\pm$ 0.28 s for CC and 1.81 s $\pm$ 0.01 s for DDoS detection.

VI. Conclusion

In this paper, we introduced a representation-level notion of concept completeness for AI-driven network functions, addressing a fundamental limitation of existing concept-based explainability approaches that primarily focus on prediction reconstruction. By measuring how much variance in a model's internal representations is captured by a predefined set of concepts, our method provides a principled and model-agnostic way to assess whether a set of concepts truly aligns with the learned feature space.

We proposed a global concept completeness metric based on variance preservation under projection onto concept subspaces and extended it to handle high-dimensional and noisy latent spaces through a significant-component analysis using SVD. This extension enables meaningful completeness assessment even for large representation spaces typical of modern deep models and LLM-based systems. In addition, we introduced a Shapley value-based importance measure that fairly attributes each concept's contribution to overall completeness, supporting systematic concept ranking.

Our experiments on synthetic data validated the correctness of the proposed metrics under controlled noise and dimensionality conditions. Application on three real-world AI-driven networking scenarios—adaptive bitrate streaming, congestion control, and DDoS detection—demonstrated that while global completeness scores can be low in high-dimensional settings, the proposed significant concept completeness reveals sub-

stantial and interpretable alignment between concepts and the most influential latent components. The resulting concept importance rankings further align well with established domain knowledge in networking.

Future research directions can involve applying the proposed methods to a broader range of AI-driven networking tasks and models. Furthermore, investigating the interplay between concept completeness and model generalization or robustness could yield valuable insights into the design of more interpretable and reliable AI systems for networking applications. Moreover, our variance-based approach may omit low-variance but decision-critical features. Extending our method with additional importance measures such as gradient-based attributions can yield more robust results. Additionally, the Shapley value calculation is currently only feasible for moderately sized concept-sets.

REFERENCES

[1] O. Ayoub, C. Natalino, S. Troia, C. Rottondi, D. Andreoletti, F. Lelli, S. Giordano, and P. Monti, "Natural language interpretability for ML-based QoT estimation via large language models," in *2025 25th Anniversary International Conference on Transparent Optical Networks (ICTON)*. IEEE, Jul. 2025, pp. 1–4.

[2] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas *et al.*, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV)," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2668–2677.

[3] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang, "Concept bottleneck models," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5338–5348.

[4] E. Poeta, G. Ciravegna, E. Pastor, T. Cerquitelli, and E. Baralis, "Concept-based Explainable Artificial Intelligence: A Survey," *ACM Computing Surveys*, Nov. 2025.

[5] S. Patel, D. Han, N. Narodytska, and S. A. Jyothi, "Agua: A concept-based explainer for learning-enabled systems," in *Proceedings of the ACM SIGCOMM 2025 Conference*, ser. SIGCOMM '25. ACM, Aug. 2025, pp. 329–346.

[6] C.-K. Yeh, B. Kim, S. Arik, C.-L. Li, T. Pfister, and P. Ravikumar, "On completeness-aware concept-based explanations in deep neural networks," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 20 554–20 565. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/ecb287ff763c169694f682af52c1f309-Paper.pdf

[7] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jul. 2017, pp. 3319–3327.

[8] B. Zhou, D. Bau, A. Oliva, and A. Torralba, "Interpreting deep visual representations via network dissection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2131–2145, Sep. 2019.

[9] G. H. Golub, *Matrix computations*, fourth edition ed., ser. Johns Hopkins studies in the mathematical sciences, C. F. V. Loan, Ed. Baltimore: The Johns Hopkins University Press, 2013.

[10] I. T. Jolliffe, *Principal component analysis*, 2nd ed., ser. Springer series in statistics. New York: Springer, 2004.

[11] L. S. Shapley, *A value for n-person games*. Cambridge University Press, Oct. 1958, pp. 31–40.

[12] R. Mitchell, J. Cooper, E. Frank, and G. Holmes, "Sampling permutations for shapley value estimation," *Journal of Machine Learning Research*, vol. 23, no. 43, pp. 1–46, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-0439.html

[13] S. Patel, J. Zhang, N. Narodystka, and S. A. Jyothi, "Practically high performant neural adaptive video streaming," *Proceedings of the ACM on Networking*, vol. 2, no. CoNEXT4, pp. 1–23, Nov. 2024.

[14] N. Jay, N. Rotman, B. Godfrey, M. Schapira, and A. Tamar, "A deep reinforcement learning perspective on internet congestion control," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 3050–3059. [Online]. Available: https://proceedings.mlr.press/v97/jay19a.html

[15] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martinez-del Rincon, and D. Siracusa, "Lucid: A practical, lightweight deep learning solution for ddos attack detection," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 876–889, Jun. 2020.

[16] W. R. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," RFC 2001, Jan. 1997. [Online]. Available: https://www.rfc-editor.org/info/rfc2001

[17] E. Blanton, D. V. Paxson, and M. Allman, "TCP Congestion Control," RFC 5681, Sep. 2009. [Online]. Available: https://www.rfc-editor.org/info/rfc5681

[18] I. Rhee, L. Xu, S. Ha, A. Zimmermann, L. Eggert, and R. Scheffenegger, "CUBIC for Fast Long-Distance Networks," RFC 8312, Feb. 2018. [Online]. Available: https://www.rfc-editor.org/info/rfc8312

[19] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.

[20] N. Tripathi and N. Hubballi, "Application layer denial-of-service attacks and defense mechanisms: A survey," *ACM Computing Surveys*, vol. 54, no. 4, pp. 1–33, May 2021.

APPENDIX

Theorem 1. *For a given set of concepts C $\kappa_C(\tau)$ is monotonically decreasing.*

Proof. Assume there is $\epsilon > 0$ and τ such that:

$$\kappa_C(\tau + \epsilon) > \kappa_C(\tau). \tag{9}$$

Then with $A = \text{Var}_{\text{covered},C}$ and $B = \text{Var}_{\text{global}}$:

$$\kappa_C(\tau) = \frac{A(\tau)}{B(\tau)}, \quad \Delta A = A(\tau + \epsilon) - A \tag{10}$$

By assumption:

$$\kappa_C(\tau + \epsilon) > \kappa_C(\tau) \Leftrightarrow \frac{A + \Delta A}{B + \Delta B} > \frac{A}{B} \Leftrightarrow \frac{\Delta A}{\Delta B} > \frac{A}{B} \tag{11}$$

This implies that the newly added singular directions (between τ and $\tau+\epsilon$) have a higher fraction of concept-aligned variance than the original dominant subspace.

However, these directions correspond to lower singular values with lower-variance components. Under the assumption that concept vectors are not preferentially aligned with these lower-variance directions, such an increase in relative alignment is not possible. This contradicts the assumption. Hence:

$$\kappa_C(\tau + \epsilon) \leq \kappa_C(\tau) \tag{12}$$

□