ConSim: Measuring Concept-Based Explanations' Effectiveness with Automated Simulatability

Anonymous ACL submission

Abstract

Concept-based explanations work by mapping complex model computations to human-understandable concepts. Evaluating such explanations is very difficult, as it includes not only the quality of the induced *space of possible concepts* but also how effectively the chosen concepts are *communicated* to users. Existing evaluation metrics often focus solely on the former, neglecting the latter.

We introduce an evaluation framework for measuring concept explanations via *automated simulatability*: a simulator's ability to predict the explained model's outputs based on the provided explanations. This approach accounts for both the concept space and its interpretation in an end-to-end evaluation. Human studies for simulatability are notoriously difficult to enact, particularly at the scale of a wide, comprehensive empirical evaluation (which is the subject of this work). We propose using large language models (LLMs) as simulators to approximate the evaluation and report various analyses to make such approximations reliable. Our method allows for scalable and consistent evaluation across various models and datasets. We report a comprehensive empirical evaluation using this framework and show that LLMs provide consistent rankings of explanation methods. Code available at GitHub.

1 Introduction

The need for transparent and interpretable models has remained a principal need in NLP, leading to the emergence of Explainable AI (XAI) as a means of fostering trust and understanding in these systems. Among the various XAI approaches, concept-based explanations stand out for their ability to bridge the gap between complex model computations and human-understandable concepts. Unlike feature attribution methods that focus on individual input features, concept-based explanations group features into higher-level abstractions or "concepts" more aligned with human cognition (Deveaud et al., 2014; Kim et al., 2018; Ghorbani et al., 2019; Fel et al., 2023b), facilitating better interpretation of the model's internal reasoning.

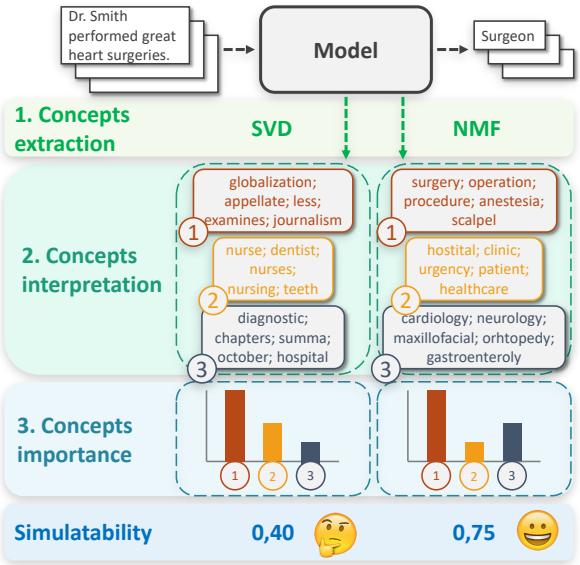


Figure 1: *How can we choose concept extraction (1) and interpretation methods (2) to make them more useful to humans?* Concept-based XAI relies on identifying relevant, interpretable concepts in the model's latent space. Different techniques yield varying concepts and importance scores (3). The simulatability score (bottom) evaluates how effectively these explanations help users understand model predictions.

However, evaluating such methods remains a challenge. Evaluation metrics often lack grounding in human interpretation (e.g., see Fig. 1–while SVD has a much higher score with many metrics, it leads to concepts that are much less useful for understanding the model's predictions). Current metrics are proxies for either faithfulness or plausibility (Jacovi and Goldberg, 2020), and the trade-off between the two is rarely explored in this setting. Furthermore, existing metrics focus on concept-space evaluation and overlook the interpretation of concept dimensions. Following previous work (Fel

et al., 2023a), we argue that concept-based explanation frameworks have three main components: constructing the concept space, evaluating concept importance, and interpreting concepts.

We propose using simulatability (Hase and Bansal, 2020; Colin et al., 2022) as a reliable method of enacting a comprehensive evaluation. Simulatability assesses the ability of a meta-predictor Ψ (simulator) to understand predictions of a model f by measuring the capacity of Ψ to simulate the predictions of f empirically. This approach evaluates both faithfulness and plausibility.

A simulatability experiment consists of three phases: i) Ψ is introduced to the task during the Initial Phase (IP); ii) learns the model's behavior in the Learning Phase (LP); and iii) attempts to simulate f's predictions during the Evaluation Phase (EP). We adapted simulatability to concept-based explanations, optionally introducing model-wise explanations at IP and sample-wise explanations at LP. However, explanations should never be provided at EP so that the labels are not leaked.

Simulatability is often evaluated through user studies. However, the number of participants necessary for an extensive method benchmark makes such studies prohibitively costly (Poursabzi-Sangdeh et al., 2021; De Bona et al., 2024) and notoriously sensitive to superficial confounders (Schuff et al., 2022). In this paper, we explore the use of large language models (LLMs) as meta-predictors, referred to as user-LLMs. Previous work (De Bona et al., 2024) has shown the potential of such meta-predictors, with results exhibiting high correlations with human performance.

We experiment with a wide variety of datasets, models, user-LLMs, and methods. As simulatability scores are only comparable for equivalent settings, we aggregated these scores using Copeland's ranked-choice voting method (Copeland, 1951; Szpiro, 2010). This gave us comparable method rankings regardless of the experimental setup. Furthermore, most of the differences between the pairwise methods were statistically significant. We tested five different methods across various datasets and meta-predictors. Non-negative Matrix Factorization (NMF; Lee and Seung, 1999) was overall the best-performing method.

Contributions:

A generalizing formalization of concept-based explanations components: (1) concept space, (2) concept importance, and (3) concept interpretation.

An evaluation framework using simulatability to assess the interpretability of concept-based explanation methods.

User-LLMs for simulatability: A demonstration of user-LLMs as effective meta-predictors in a simulatability framework.

A comprehensive empirical analysis across multiple use-cases, with statistical significance.

2 Concept Explanations: Background

The field of explainable artificial intelligence (XAI) for classification tasks has witnessed significant growth, driven by the widespread adoption of deep learning techniques. Among the various approaches, attribution methods (Zeiler and Fergus, 2014; Ribeiro et al., 2016; Shrikumar et al., 2017; Lundberg, 2017) have traditionally dominated the literature, offering insights by highlighting the contributions of input features to model predictions. However, concept-based methods (Kim et al., 2018; Ghorbani et al., 2019; Koh et al., 2020; Yeh et al., 2020; Zarlenga et al., 2022; Jourdan et al., 2023b) have recently gained increasing attention, providing a complementary perspective by focusing on high-level, human-interpretable concepts to explain model behavior.

Supervised vs. Unsupervised. Within concept-based explainability methods, two main categories can be identified. The first relies on supervised concepts constructed using labeled concept datasets. This category includes methods such as CAV (Concept Activation Vector) (Kim et al., 2018) for post-hoc approaches and CBM (Concept Bottleneck Model) (Koh et al., 2020) for by-design frameworks. However, finding labeled concepts is inherently difficult, and creating datasets to represent concepts often introduces substantial human bias (Ramaswamy et al., 2023).

By contrast, unsupervised concept-based methods do not rely on labeled concepts and instead extract them directly from the model's latent space. Neurons are not interpretable in themselves (Elhage et al., 2022; Colin et al., 2024; Dreyer et al., 2024). Hence, the most widely used approach treats concepts as a linear combination of neurons – directions in the latent space (Kim et al., 2018; Yeh et al., 2020; Zhang et al., 2021; Cunningham et al., 2023; Fel et al., 2023b; Jourdan et al., 2023b; Zhao et al., 2024). Recent advances in mechanistic interpretability have focused on Sparse Auto-Encoders

(SAEs; Ng et al., 2011; Makhzani and Frey, 2013; Domingos, 2015) to find these concepts (Bricken et al., 2023; Rajamanoharan et al., 2024a,b; Templeton et al., 2024; Gao et al., 2024; Lieberum et al., 2024; Fel et al., 2024).

Evaluations. Post-hoc, unsupervised concept-based explanations evaluation typically focuses on two main properties: faithfulness and plausibility (Jacovi and Goldberg, 2020). Faithfulness-oriented metrics – such as completeness (Yeh et al., 2020), fidelity (Zhang et al., 2021), relative ℓ_2 (Fel et al., 2023a), FID and OOD (Fel et al., 2023a), and MAE (Bricken et al., 2023) – measure how well the identified concepts preserve the information from the model's original embeddings. In addition, plausibility is often inferred from simplicity-based proxies such as sparsity (Fel et al., 2023a; Bricken et al., 2023) and conciseness (Vielhaben et al., 2023).

Many evaluation frameworks rely on labeled concepts (e.g., CEBaB (Abraham et al., 2022)), which are often challenging to define, validate, and align with a model's internal representations. Although some studies have performed human evaluations (Zhang et al., 2021; Barua et al., 2024), to the best of our knowledge, no previous work has applied simulatability to concept-based explanations.

Simulatability. Simulatability can be defined as the degree to which "a user can correctly and efficiently predict the method's results" (Kim et al., 2016; Hase and Bansal, 2020; Colin et al., 2022). It evaluates how useful and understandable an explanation is to a user. Recent findings indicate that large language models (LLMs) can approximate human judgments at scale (De Bona et al., 2024). Hence, we propose using LLMs as meta-predictors to evaluate simulatability and explainability without relying on predefined labeled concepts.

3 A Theoretical Framework for Post-hoc Unsupervised Concept-XAI

Consider classification models $f : \mathcal{X} \longrightarrow \mathcal{Y}$ with input space $\mathcal{X}$ and output space $\mathcal{Y}$. The model is decomposed into: $f = g \circ h : \mathcal{X} \xrightarrow{h} \mathcal{H} \xrightarrow{g} \mathcal{Y}$ with $\mathcal{H} \subseteq \mathbb{R}^p$ the embedding space. In our experiments, we divide the model at the penultimate layer, in DistilBERT (Sanh et al., 2019a) h outputs would be the token [CLS]. Concept-based explanations have three main components described in the three following subsections and illustrated in Fig. 2.

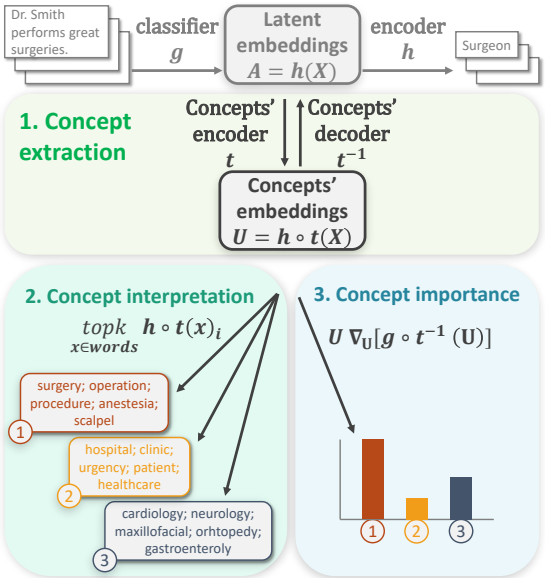


Figure 2: *A generalizing formalization of Concept-based explanations.* For a model $f = g \circ h$, concepts can be extracted from its activations $A = h(X)$ using the *concept encoder* t, and can be decoded using the *concept decoder* t^{-1}. The explanation can interpreted by keeping the most relevant words for each concept. Finally, an importance score can be attributed to each concept to understand their role in the model's rationale.

3.1 Concepts Space

The first step of post-hoc unsupervised concept-based explainability is to define the concept space $\mathcal{C} \subseteq \mathbb{R}^k$ through concept extraction methods. *Concept extraction methods* allow the construction of a projection $t : \mathcal{H} \longrightarrow \mathcal{C}$ and its bijection (or approximation) $t^{-1} : \mathcal{C} \longrightarrow \mathcal{H}$ (appendix C.3 defines how are obtained such concept projection). Note the input-to-concept part $f_{ic} : \mathcal{X} \xrightarrow{h} \mathcal{H} \xrightarrow{t} \mathcal{C}$ and the concept-to-output part $f_{co} : \mathcal{C} \xrightarrow{t^{-1}} \mathcal{H} \xrightarrow{g} \mathcal{Y}$. Finally, we can construct $f_c = f_{co} \circ f_{ic} : \mathcal{X} \xrightarrow{f_{ic}} \mathcal{C} \xrightarrow{f_{co}} \mathcal{Y}$, an unsupervised CBM (Concept-Bottleneck Model) (Koh et al., 2020). Note that in many concept-based explainability methods for classification, concept extractions are done class-by-class. In our case, we treat all classes at the same time to obtain a common concept space, as in Jourdan et al. (2023a).

3.2 Concepts Interpretability

The second part of post-hoc unsupervised concept-based explainability is to interpret concepts. Concepts are directions in the latent space and are not interpretable as is. How to represent a concept is still an open question. It is possible to represent concepts as word clouds (Dalvi et al., 2022), give

examples that activate the concepts and highlight important words (Jourdan et al., 2023b), or label the given concepts by either: asking human annotators (Dalvi et al., 2022); finding the most aligned label in a concept bank (Sajjad et al., 2022); or asking an LLM to label the concept based on maximally activating examples (Bricken et al., 2023; Templeton et al., 2024). The last solution has been the most popular in the mechanistic interpretability literature. However, its computational cost is high for interpreting a single concept. In this paper, we explored two different interpretability methods:

Concept Maximally Activating Words (CMAW) selects the five words that most strongly activate a concept and, if negative activations exist, also the five least activating words. These words are selected from words frequent enough in the dataset. With regards to concept dimension i, CMAW can be computed as follows:

$$CMAW(cpt_i) = \underset{x \in words}{\text{topk}} \; f_{ic}(x)_i \tag{1}$$

o1 Concept Alignment (o1CA) . For o1CA, we prompt GPT o1 (OpenAI, 2024a) for potential concept labels and corresponding representative sentences, then align discovered concepts to these labels by choosing the label with the highest mean activation on the corresponding sentence. Thus, for our concept dimension i, with X_j the sentences corresponding to o1 concept j, we have:

$$o1CA(cpt_i) = \max_{j \in o1_cpt} \; \underset{x \in X_j}{\text{mean}} \, f_{ic}(x)_i \tag{2}$$

3.3 Concepts Importance

Concept attribution methods $\varphi : \mathcal{C} \longrightarrow \mathbb{R}^k$ provide the importance of each concept for a given prediction based on the concepts. Fel et al. (2023a) show (theorem 3.2) that when the model is divided at the penultimate layer, certain attribution methods (e.g., Gradient Input (Shrikumar et al., 2017)) are optimal. We, therefore, choose Gradient Input for its simplicity and efficiency. *Local concepts' importance* φ can be defined for a given sample $x \in \mathcal{X}$, with concepts representation $u = f_{ic}(x) \in \mathcal{C}$, by Eq. 3. Through this, with $X \in \mathcal{X}^n$ the train set samples and $U = f_{ic}(x) \in \mathcal{C}^n$ their concepts representations, we can define *global concepts importance* Φ with regard to class c through Eq. 4.

$$\varphi_{f_{co}}(u) = u \nabla_u f_{co}(u) \tag{3}$$

$$\Phi_{f_{co},c} = \underset{u \in U | f_{co}(u) = c}{\text{mean}} \; \varphi_{f_{co}}(u) \tag{4}$$

4 Our Evaluation Framework

4.1 Simulatability

Simulatability aims to quantify how well a meta-predictor Ψ (also called simulator) can replicate the predictions of an AI model f (Kim et al., 2016; Hase and Bansal, 2020; Colin et al., 2022). The meta-predictor is usually a human, but in our experiments, we use an LLM as a meta-predictor. The meta-predictor is given samples and tasked to predict what would have predicted the AI model.

A simulatability experiment consists of three phases. These parts are illustrated in Fig. 3 through reduced examples of prompt parts:

Initial Phase (IP): The meta-predictor receives a description of the task and possibly some global explanations of the model. Global explanations consist of global concepts' importance Φ as defined by Eq. 4 and the important concepts' interpretation.

Learning Phase (LP): The meta-predictor is shown examples with the model's predictions and, optionally, local explanations. The explanations are concepts' importance φ as defined by Eq. 3.

Evaluation Phase (EP): The meta-predictor must predict the model's outputs on new samples without access to these predictions. No explanation is given at the phase as it would leak the label.

In summary, Ψ is introduced to the task during IP, learns the model's behavior in LP, and attempts to simulate f's predictions in EP. Since Ψ's performance may depend on the experimental settings s and the chosen concept extraction method m, we denote it as $\Psi_{s,m}$. By assessing how accurately Ψ replicates f's outputs, this approach mitigates issues like confirmation bias and prediction leakage (Colin et al., 2022). We measure simulatability as the accuracy of the meta-predictor's guesses on EP samples X_{EP}:

$$acc_{\Psi,s,m} = \underset{x \in X_{EP}}{\mathbb{E}} \mathbb{1}\left\{\Psi_{s,m}(x_i) = f(x_i)\right\} \tag{5}$$

In each setting, samples for LP and EP were selected to represent the dataset and better differentiate methods explaining performance. Each setting had different seeds for more statistically significant results. Details are described in appendix B.1.

4.2 User-LLM and Prompting

We refer to LLMs replacing users in user studies as user-LLMs (De Bona et al., 2024). User-LLMs

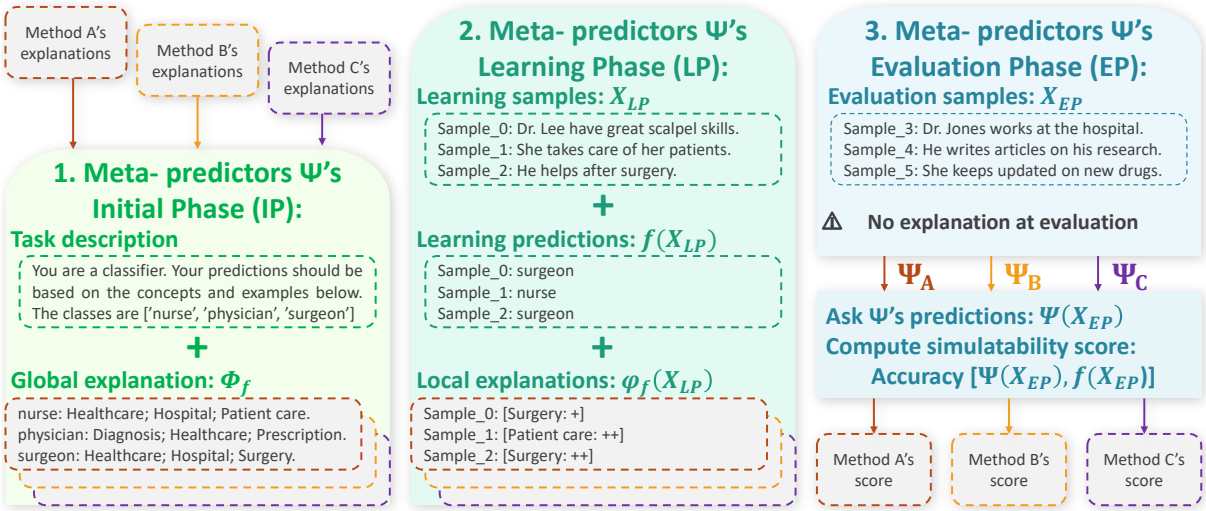


Figure 3: **Overview of our simulatability framework.** For a given *meta-predictor* Ψ (User-LLM or human), our simulatability framework is composed of three distinct stages: *(i)* an **initial phase** (**IP**) where the task is carefully described to Ψ and the global explanation is shown to it; *(ii)* a **learning phase** (**LP**) where some samples are shown to Ψ, along with the model f predictions; *(iii)* a final **evaluation phase** (**EP**) where a different set of samples is input to Ψ without the corresponding predictions, and it is asked to predict what the model f would have predicted. With this information, the simulatability score can be computed as the accuracy in guessing the model's outputs.

do not replace studies with real humans, but they allow experiments at a much larger scale to provide an approximation and motivation for future investment in human user studies. Furthermore, it was shown the conclusions of studies through the lens of user-LLMs tend to correlate with human studies (De Bona et al., 2024).

In our case, we leverage GPT-4o-mini (OpenAI, 2024b, §5.1) and Gemini-1.5 Flash and Pro (Team et al., 2024, §6.1). The Gemini experiments cover a representative subset of the full experiment scope.

Selecting the concepts. Some of the induced concept spaces had 500 concepts. Showing them all would complicate the prompt unnecessarily. Therefore, for global and local explanations, we only show concepts with normalized global importance $\hat{\Phi}_{c,cpt}$ in absolute value above a threshold of 0.05 for at least one class. For a given class c and concept cpt, its normalized global importance is defined in appendix B.2, Eq. 9. Similarly, for the remaining concepts, local explanations only include concepts with importance values above 0.05 for the given sample. Normalized local importance is defined in appendix B.2, Eq.10. In prompts, concepts' importance is encoded into four buckets for simplicity; details in appendix B.3.

4.3 Ranking

Different settings are constructed by fixing the dataset, model, seed, concepts' extraction method, concepts' interpretation method, prompt type, and user-LLM. However, the simulatability scores $acc_{\Psi,s,m}$ (Eq. 5) between the two methods can only be compared when the setting s is the same. Therefore, to rank methods, we make the parallel with ranked-choice voting systems. We consider the simulatability score from a setting as a vote with order between methods and aggregate these votes using Copeland's method (Copeland, 1951; Szpiro, 2010) with the "0/1/2" rule, a kind of Condorcet method (Pomerol and Barba-Romero, 2012). Afterward, with S the settings, i, and j methods, we construct the pairwise comparison matrix P through Eq. 6. Note that here, "methods" are either concept extraction methods, concept interpretation methods, or concept importance methods.

$$P_{i,j} = \sum_{s \in S} \begin{cases} 0 & \text{if } acc_{\Psi,s,i} < acc_{\Psi,s,j} \\ 1 & \text{if } acc_{\Psi,s,i} = acc_{\Psi,s,j} \\ 2 & \text{if } acc_{\Psi,s,i} > acc_{\Psi,s,j} \end{cases} \quad (6)$$

Each value is then normalized to obtain a value between 0 and 100 comparable to a percentage of wins. Finally, the ranking of a method i is constructed from the number of times method i is preferred over method j, with M the list of concepts explanation methods:

$$rank_P(i) = |M| + 1 - \sum_{j \in M} \mathbb{1}\{P_{i,j} \geq 50\} \quad (7)$$

Furthermore, another pairwise comparison matrix was computed to determine if the pairwise

differences were statistically different 0. We used a student's test (Student, 1908) with a p-value threshold of 0.05. To do so, the mean differences between accuracies were computed with:

$$Diff_{i,j} = \underset{s \in S}{\text{mean}}[acc_{\Psi,s,i} - acc_{\Psi,s,j}] \quad (8)$$

5 Ranking Methods with GPT-4o-mini

The first experiment was conducted with GPT-4o-mini (OpenAI, 2024b) as meta-predictor Φ. Comparison of the ranking between several user-LLMs are described in Sec. 6.

5.1 GPT-4o-mini Experiments Description

Experiments with GPT-4o-mini were conducted with an extended set of settings compared to the latter comparison. We use 4 datasets, 5 models, 7 seeds, 5 concept extraction methods, 2 concept interpretation methods, 6 prompt types for explanations, 4 other prompt types for baselines, and, for some settings, 7 different numbers of concepts. There are also several baseline prompts. Resulting in $23,360$ different experiment settings reported and used for GPT-4o-mini. Prompt mean size was about $2,000$ tokens; hence, through the OpenAI API, this cost around 7$. The different settings variables are listed below:

Datasets. We consider four classification datasets: (i) A reduced version of BIOS (De-Arteaga et al., 2019), limited to the 10 most frequent classes; (ii) IMDB (Maas et al., 2011); (iii) Rotten Tomatoes (Pang and Lee, 2005); (iv) The "emotion" subset of the Tweet Eval dataset (Barbieri et al., 2020). For concept extraction, we often augment the original datasets by including split samples (partial sentences) derived from the initial samples. See extended details in Appendix C.1.

Models. We evaluate three model architectures: an encoder model DistilBERT (Sanh et al., 2019a), an encoder-decoder model T5 (Raffel et al., 2020), and a decoder model Llama3-8B (Dubey et al., 2024). DistilBERT and T5 were fine-tuned for the classification tasks, while Llama3-8B used prompting. Details of model fine-tuning and adaptation are in appendix C.2. DistilBERT and T5 were fine-tuned with positive embeddings to enable NMF-based concept extraction. These modified models are denoted with $+$ in Tab. 2 and Tab. 3, resulting in five distinct models in total.

Concept extraction methods. We employed five concept extraction methods, each representing a form of dictionary learning as generalized in (Fel et al., 2023a): (i) Independent Component Analysis (ICA) (Ans et al., 1985; Hyvärinen and Oja, 2000), (ii) Non-negative Matrix Factorization (NMF) (Lee and Seung, 1999; Sra and Dhillon, 2005), (iii) Principal Component Analysis (PCA) (Pearson, 1901; Hotelling, 1992), (iv) Sparse Auto-Encoder (SAE) (Ng et al., 2011; Makhzani and Frey, 2013; Domingos, 2015), and (v) Singular Value Decomposition (SVD) (Eckart and Young, 1936). See appendix C.3 for details on their implementation and the corresponding notation.

Simulatability Phase		L1	E1	L2	E2	E3
IP	Task desc.	✓	✓	✓	✓	✓
	Global expl.		✓		✓	✓
LP	X_{LP}, $f(X_{LP})$			✓	✓	✓
	Local expl.					✓

Table 1: Elements present in the simulatability prompt depending on the experiment (E1, E2, or E3) or the baseline (L1 or L2). Details in Sec. 4.1 and Fig. 3.

Concept interpretation methods. Experiments use the two concept interpretability methods introduced in Section 3.2, namely CSAW and o1CA.

Prompt types. We explored several prompt configurations to answer questions, such as whether a learning phase (LP) improves user-LLM performance and whether local explanations are beneficial. Tab. 1 details these prompt settings. The simplest baselines, L1 and L2, include no explanations. E1 is compared to L1, and any setting that includes an LP is compared to L2. All instructions are provided in the system prompt, except those for the evaluation phase (EP), which are given in the user prompt.

Anonymous prompt types. While the user-LLMs (Ψ) can achieve performance levels close to those of the fine-tuned models on the initial task, our objective is for them to predict exactly what model f would predict. To increase complexity, we introduced experiments where class labels are anonymized (denoted by "-a" in Lx-a or Ex-a variants), ensuring that Ψ must rely more on the provided concepts and explanations rather than directly recognizing class names.

Number of concepts. Finally, each concept extraction method includes a hyperparameter specifying the number of concepts. We tested $k \in \{3, 5, 10, 20, 50, 150, 500\}$ concepts. Some config-

Experiment setting subset		Concept extraction						Concept interpretation		
		NMF	SAE	ICA	PCA	SVD	Baseline	CMAW	o1CA	Baseline
Datasets	BIOS10	**1**	2	3	4	5	6	**1**	2	3
	IMDB	**1**	4	3	6	5	2	**1**	3	2
	rotten tomatoes	**1**	2	4	6	5	3	**1**	2	3
	tweet eval	**1**	3	2	5	6	4	**1**	2	3
Models	DistilBERT	N/A	**1**	**1**	4	5	3	**1**	2	3
	DistilBERT+	**1**	2	3	4	6	5	**1**	2	3
	Llama-3-8B	N/A	2	**1**	5	4	3	**1**	2	3
	T5	N/A	**1**	2	4	5	3	2	**1**	3
	T5+	**1**	2	3	5	6	4	**1**	2	3
All settings		**1**	2	2	5	6	4	**1**	2	3

Table 2: **Methods ranking with GPT-4o-mini.** Comparison of concept extraction methods and concept interpretation methods rankings across different sets of settings. In a setting (a line), we fix either one of the datasets or models. The last line shows the ranking for all settings of the extended GPT-4o-mini experiments.

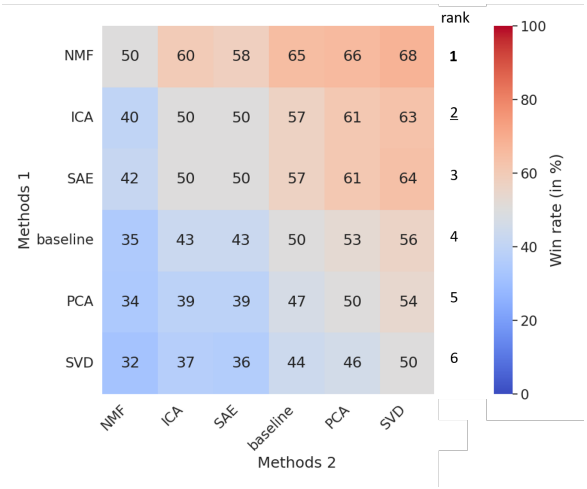


Figure 4: Pairwise comparison matrices on GPT-4o-mini experiments described in Sec. 5.1. Percentage of simulatability experiments where method 1 is over method 2. Ranking by number of pairwise victories.

urations timed out with a large k (ICA and NMF). Instead of reporting results for every setting or always selecting the best outcome, we followed the validation procedure described at the end of Sec. 4.1, using two 40-sample sets to determine the optimal number of concepts for each dataset-method pair. The best number of concepts was often very high, which can be explained by the fact that we only showed the most important concepts.

In summary, a large variety of settings were explored to obtain statistically robust and generalizable results. This experiment has shown that NMF, SAE, and ICA are the most promising concept extraction methods. Furthermore, the concept interpretation method CMAW – the simplest of the two – is above o1CA in most cases.

5.2 GPT-4o-mini Results

GPT-4o-mini experiments can be analyzed from different angles: first, comparing concept extraction methods; second, comparing concept interpretation methods; and third, comparing the prompt types. In any case, results are primarily aggregated in pairwise comparison matrices Eq. 6 and Eq. 8, then the ranking is constructed following Eq. 7.

Concept extraction methods. Examples of pairwise comparison matrices defined in Eq. 6 and Eq. 8 are respectively shown in Fig. 4 and Fig. 5. Fig. 4 shows the percentage of wins between two methods and the final ranking of methods, putting the NMF above the others. Fig. 4 shows that most differences are statistically significant with respect to a student's test (Student, 1908) with a p-value threshold of 0.05.

Tab. 2 summarizes the ranking across settings with GPT-4o-mini as the user-LLM. The first line shows that overall, NMF ranks higher than SAE and ICA, which also rank higher than the baseline (no explanation). Finally, the PCA and SVD rank, overall, below the baseline. Tab. 2 also shows that across subsets of settings with either one of the dataset, model, or concept interpretation methods fixed, the ranking is similar. Indeed, NMF, when applicable, is always ranked first, SAE and ICA occupy the top 3 apart from one time, and finally, PCA and SVD stay in the bottom three in any case. However, the baseline rank, thus the performance of methods overall, varies a lot with the dataset.

Concept interpretation method. With regards to the comparison of the concepts' interpretability methods, Tab. 2 shows that CMAW is better than o1CA. Both methods also appear above the baseline (prompts, no explanation). This can be

User-LLMs	Concept extraction						Concept interpretation		
	NMF	SAE	ICA	PCA	SVD	Baseline	CMAW	o1CA	Baseline
GPT-4o-mini	**1**	2	3	4	6	5	**1**	2	3
Gemini-1.5-flash	**1**	2	3	4	5	6	**1**	2	3
Gemini-1.5-pro	**1**	2	3	4	6	5	**1**	2	3

Table 3: **User-LLMs ranking comparison.** Comparison of concept extraction methods and concept interpretation methods rankings across different user-LLMs on the representative subset of experiments described in Sec. 6.1.

explained by the fact that PCA and SVD were removed from these experiments, knowing that their concepts were not interpretable; comparing interpretability methods with them did not make sense. Finally, Tab. 2 shows consistent results across settings, with about half of them being statistically significant.

Prompt types. The statistical differences between prompt types are illustrated in Fig. 6, where significant differences are in bold. It shows that no difference can be made between settings with real class names, suggesting that GPT-4o-mini shortcuts the task and ignores concepts. However, for setting anonymous classes representing more complex tasks, the GPT-4o-mini simulatability score obtains a clear gain with explanations. Finally, it seems that local explanations do not help if global explanations are given.

6 User-LLMs Comparison Experiments

6.1 Comparison Experiments Description

For the second set of experiments, we compared the previous GPT-4o-mini results with more advanced user-LLMs, such as Gemini-1.5 (flash and Pro) (Team et al., 2024). This comparison was done on a subset of the previously defined settings. We restricted this comparison to the two non-binary classification datasets (BIOS10 and Tweet Eval Emotion). Additionally, we only considered the positively fine-tuned versions of DistilBERT and T5, ensuring that all concept extraction methods were compatible. Finally, we used prompt types E1 and E2 and their anonymized variants, along with the corresponding baselines, to enable a consistent and fair comparison across different user-LLMs.

6.2 Comparison Experiments Results

Concept extraction methods. Through the comparison of the 3 user-LLMs in Tab. 3, the ranking is maintained apart between SVD and the baseline; details in Fig. 7a and Fig. 8a. Not all pairwise differences are significant: Fig. 8b and Fig. 8b show that the difference between SAE and ICA is not statistically significant for all user-LLMs. Similarly, the order of the three last methods, namely, PCA, SVD, and the baseline, is not statistically significant. However, NMF ranks first in all settings Similarly, SAE and ICA remain in the top 3.

Concept interpretation methods. Tab. 3 shows that the ranking is conserved across the different user-LLMs, placing CMAW on top. Differences are statistically significant.

7 Conclusion

We present a simulatability experiment for post-hoc unsupervised concept-based explanations with user-LLMs. The results show that concept-based explanations can help user-LLMs predict what would have predicted a classification model, and that user-LLM accuracy can be used to rank methods with statistical significance, across multiple user-LLMs.

Recommendations. Our evaluation framework and empirical report gives concrete recommendations with regard to the different parts of concept-based explanations: the NMF method appears to be the most interpretable, however, it requires positive embeddings. Hence, without positive embeddings we recommend the use of SAEs. These methods are popular in recent literature as they can create over-complete concept banks which are necessary for generative tasks. However, these models are fragile and difficult to implement. Thus, the ICA would be the most simple to apply as it does not have such constraints.

With regards to the concept interpretability methods, using the CMAW only requires the model and have a constant cost, regardless of the number of methods or concepts. This makes it suitable as a baseline. It obtained better results than the second method. The method used by (Bricken et al., 2023; Templeton et al., 2024) seems to be more interpretable but requires much more computing, a more complex pipeline, and the use of an LLM.

Limitations

Despite our efforts to design thorough and comprehensive experiments, we acknowledge that certain blind spots and limitations may still remain, reflecting the inherent challenges in achieving complete coverage in such analyses. Namely, three major points could be raised:

- We followed the suggestion in (Fel et al., 2023a) of computing the concepts in the penultimate layer of the model. We assume Computer Vision models behave similarly to NLP models in this regard, but this might not be the case. However, our framework can also be applied elsewhere in the residual stream or MLP layers of a transformer model.
- Due to the sudden popularity and speed at which the state-of-the-art of SAEs changes at the time of writing, the SAE studied in this work – described in appendix C.3 – did not include the latest improvements (Rajamanoharan et al., 2024a,b; Gao et al., 2024; Leask et al., 2024; Bussmann et al., 2024). Therefore, SAEs results are probably underestimated.
- Although previous work seems to provide evidence towards LLM's being a useful proxy for human behavior (De Bona et al., 2024), there is no actual proof that the ranking would be similar to one calculated using humans as meta-predictor Ψ.

References

Eldar D Abraham, Karel D'Oosterlinck, Amir Feder, Yair Gat, Atticus Geiger, Christopher Potts, Roi Reichart, and Zhengxuan Wu. 2022. Cebab: Estimating the causal effects of real-world concepts on nlp model behavior. *Advances in Neural Information Processing Systems (NeurIPS)*.

AI@Meta. 2024. Llama 3 model card. https://huggingface.co/meta-llama/Meta-Llama-3-8B.

B Ans, J Hérault, and C Jutten. 1985. Architectures neuromimétiques adaptatives: Détection de primitives. *Proceedings of Cognitiva*.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa-Anke, and Leonardo Neves. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Adrita Barua, Cara Widmer, and Pascal Hitzler. 2024. Concept induction using llms: a user experiment for assessment. *ArXiv e-print*.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bart Bussmann, Patrick Leask, and Neel Nanda. 2024. Batchtopk sparse autoencoders. In *Workshop in Advances in Neural Information Processing Systems (NeurIPS)*.

Julien Colin, Thomas Fel, Rémi Cadène, and Thomas Serre. 2022. What i cannot predict, i do not understand: A human-centered evaluation framework for explainability methods. *Advances in Neural Information Processing Systems (NeurIPS)*.

Julien Colin, Lore Goetschalckx, Thomas Fel, Victor Boutin, Jay Gopal, Thomas Serre, and Nuria Oliver. 2024. Local vs distributed representations: What is the right basis for interpretability? *ArXiv e-print*.

Arthur H Copeland. 1951. A reasonable social welfare function. Technical report, mimeo, 1951. University of Michigan.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Fahim Dalvi, Abdul Rafae Khan, Firoj Alam, Nadir Durrani, Jia Xu, and Hassan Sajjad. 2022. Discovering latent concepts learned in bert. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*.

Francesco Bombassei De Bona, Gabriele Dominici, Tim Miller, Marc Langheinrich, and Martin Gjoreski. 2024. Evaluating explanations through llms: Beyond traditional user studies. *Workshop in Advances in Neural Information Processing Systems (NeurIPS)*.

Romain Deveaud, Eric SanJuan, and Patrice Bellot. 2014. Accurate and effective latent concept modeling for ad hoc information retrieval. *Document numérique*.

Pedro Domingos. 2015. *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Basic Books.

Maximilian Dreyer, Erblina Purelku, Johanna Vielhaben, Wojciech Samek, and Sebastian Lapuschkin. 2024. Pure: Turning polysemantic neurons into pure features by identifying relevant circuits. *ArXiv e-print*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *ArXiv e-print*.

Carl Eckart and Gale Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. 2022. Toymodelsof superposition. *Transformer Circuits Thread*.

Thomas Fel, Louis Bethune, Andrew Kyle Lampinen, Thomas Serre, and Katherine Hermann. 2024. Understanding visual feature reliance through the lens of complexity. *ArXiv e-print*.

Thomas Fel, Victor Boutin, Mazda Moayeri, Rémi Cadène, Louis Bethune, Mathieu Chalvidal, Thomas Serre, et al. 2023a. A holistic approach to unifying automatic concept extraction and concept importance estimation. *Advances in Neural Information Processing Systems (NeurIPS)*.

Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. 2023b. Craft: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *ArXiv e-print*.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. 2019. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems (NeurIPS)*.

Peter Hase and Mohit Bansal. 2020. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Harold Hotelling. 1992. Relations between two sets of variates. In *Breakthroughs in statistics: methodology and distribution*. Springer.

Aapo Hyvärinen and Erkki Oja. 2000. Independent component analysis: algorithms and applications. *Neural networks*.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Fanny Jourdan, Louis Béthune, Agustin Picard, Laurent Risser, and Nicholas Asher. 2023a. Taco: Targeted concept erasure prevents non-linear classifiers from detecting protected attributes. *ArXiv e-print*.

Fanny Jourdan, Agustin Picard, Thomas Fel, Laurent Risser, Jean Michel Loubes, and Nicholas Asher. 2023b. Cockatiel: Continuous concept ranked attribution with interpretable elements for explaining neural net classifiers on nlp tasks. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. 2016. Examples are not enough, learn to criticize! Criticism for Interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *Proceedings of the International Conference on Machine Learning (ICML)*.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Patrick Leask, Bart Bussmann, Joseph Isaac Bloom, Curt Tigges, Noura Al Moubayed, and Neel Nanda. 2024. Stitching sparse autoencoders of different sizes. In *Workshop in Advances in Neural Information Processing Systems (NeurIPS)*.

Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *nature*.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *ArXiv e-print*.

Scott Lundberg. 2017. A unified approach to interpreting model predictions. In *31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Alireza Makhzani and Brendan Frey. 2013. K-sparse autoencoders. *ArXiv e-print*.

Andrew Ng et al. 2011. Sparse autoencoder. *CS294A Lecture notes*.

OpenAI. 2024a. Introducing openai o1-preview: A new series of reasoning models for solving hard problems. https://openai.com/index/introducing-openai-o1-preview/.

OpenAI. 2024b. Openai o1-mini: Advancing cost-efficient reasoning. https://openai.com/index/openai-o1-mini-advancing-cost-efficient-reasoning/.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Karl Pearson. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research (JMLR)*.

Jean-Charles Pomerol and Sergio Barba-Romero. 2012. *Multicriterion decision in management: principles and practice*. Springer Science & Business Media.

Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Wortman Vaughan, and Hanna Wallach. 2021. Manipulating and measuring model interpretability. In *Proceedings of the ACM Conference on Human Factors in Computing systems (CHI)*.

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

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. 2024a. Improving dictionary learning with gated sparse autoencoders. *ArXiv e-print*.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. 2024b. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *ArXiv e-print*.

Vikram V Ramaswamy, Sunnie SY Kim, Ruth Fong, and Olga Russakovsky. 2023. Overlooked factors in concept-based explanations: Dataset choice, concept learnability, and human capability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Firoj Alam, Abdul Khan, and Jia Xu. 2022. Analyzing encoded concepts in transformer language models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019a. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv e-print*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019b. Distilbert model card. https://huggingface.co/distilbert/distilbert-base-uncased.

Hendrik Schuff, Alon Jacovi, Heike Adel, Yoav Goldberg, and Ngoc Thang Vu. 2022. Human interpretation of saliency-based explanation over text. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 611–636, New York, NY, USA. Association for Computing Machinery.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Suvrit Sra and Inderjit Dhillon. 2005. Generalized nonnegative matrix approximations with bregman divergences. *Advances in Neural Information Processing Systems (NIPS)*.

Student. 1908. The probable error of a mean. *Biometrika*.

George Szpiro. 2010. *Numbers rule: the vexing mathematics of democracy, from Plato to the present*. Princeton University Press.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al.

2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *ArXiv e-print*.

HuggingFace team. 2020. T5 model card. https://huggingface.co/google-t5/t5-base.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

Johanna Vielhaben, Stefan Bluecher, and Nils Strodthoff. 2023. Multi-dimensional concept discovery (mcd): A unifying framework with completeness guarantees. *ArXiv e-print*.

Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. 2020. On completeness-aware concept-based explanations in deep neural networks. *Advances in neural information processing systems*.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, et al. 2022. Concept embedding models: beyond the accuracy-explainability trade-off. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*.

Ruihan Zhang, Prashan Madumal, Tim Miller, Krista A Ehinger, and Benjamin IP Rubinstein. 2021. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Ruochen Zhao, Tan Wang, Yongjie Wang, and Shafiq Joty. 2024. Explaining language model predictions with high-impact concepts. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

A Experiments Supplementary Visualizations

A.1 Ranking Methods with GPT-4o-mini

Fig. 4 and Fig. 5 illustrate concept extraction methods pairwise comparison matrices when all GPT-4o-mini settings are taken into account.

Fig. 6 shows the pairwise comparison matrix with the percentages of wins between prompt types.

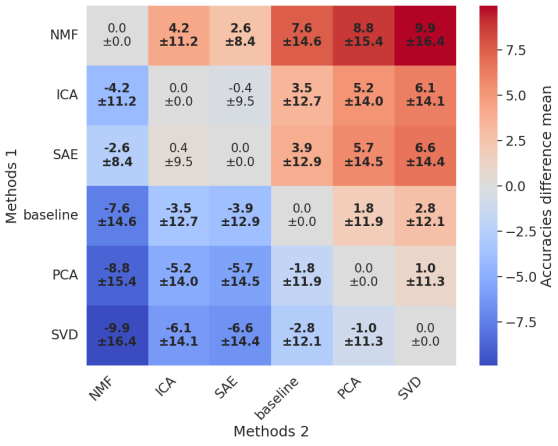


Figure 5: Pairwise comparison matrices on GPT-4o-mini experiments described in Sec. 5.1. Difference means and standard deviations between method 1 and method 2 simulatability scores across experiments. Bold differences are statistically significant.

Interpretation	NMF	SAE	ICA	PCA	SVD	Baseline
CMAW	**1**	3	2	5	6	4
o1CA	**1**	2	3	5	6	4

Table 4: **Experiments with GPT-4o-mini as a user-LLM.** Concepts extraction methods ranking for the two concept interpretation methods.

It takes into the top3 concept extraction methods (NMF, SAE, and ICA) and all of the GPT-4o-mini settings on other variables.

Tab. 4 and Tab. 4 shows the rankings of concept extraction methods and concept interpretation methods when the other is fixed. In both cases, the order is conserved, and the NMF-CMAW pair emerged on the first rank.

A.2 User-LLMs Comparison

Fig. 7 illustrates concept extraction methods pairwise comparison matrices for Gemini-1.5-flash as the meta-predictor.

Fig. 8 illustrates concept extraction methods pairwise comparison matrices for Gemini-1.5-pro as the meta-predictor.

B Simulatability Prompting

B.1 Prompt Samples

In our simulatability experiments, we select 40 samples for each dataset-model pair: 20 for the Learning Phase (LP) and 20 for the Evaluation Phase (EP). These samples are chosen to cover each class uniformly. Among them, 20 are correctly classified

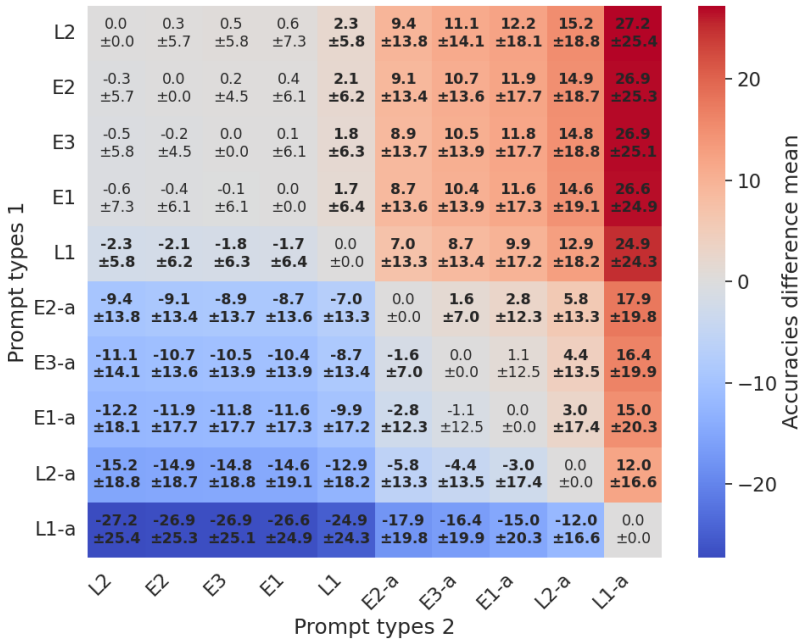


Figure 6: Percentage of simulatability experiments where method 1 is over method 2. Ranking by number of pairwise victories. GPT-4o-mini experiments described in Sec. 5.1 subset to the top 3 methods NMF, SAE, and ICA. Experiments (E1, E2, and E3) and baselines (L1 and L2) are described in Sec. 5.1 and Tab. 1. They differ in the simulatability elements present in the prompt. Experiments and baselines with "-a" are done with anonymous classes.

Extraction	**CMAW**	**o1CA**	**Baseline**
ICA	**1**	2	3
NMF	**1**	2	3
SAE	**1**	2	3

Table 5: **Experiments with GPT-4o-mini as a user-LLM.** Concepts interpretation methods ranking for the top 3 concept extraction methods.

by f, and 20 are misclassified, ensuring a balanced challenge for the meta-predictor. We then randomly distribute these samples between the LP and EP.

To increase statistical robustness, we repeat this selection with 5 different random seeds, resulting in 5 distinct sets of 40 samples. Additionally, we use 2 more sets of 40 samples to determine the optimal number of concepts for each dataset-method pair.

Finally, this paper reports $23,360$ for GPT-4o-mini and 960 for both Gemini-1.5 Flash and Pro. These prompts had a mean number of tokens around $2,000$, mostly represented by the samples.

B.2 Normalizing Concept Importance

For a given class c and concept cpt, its normalized global importance is defined by:

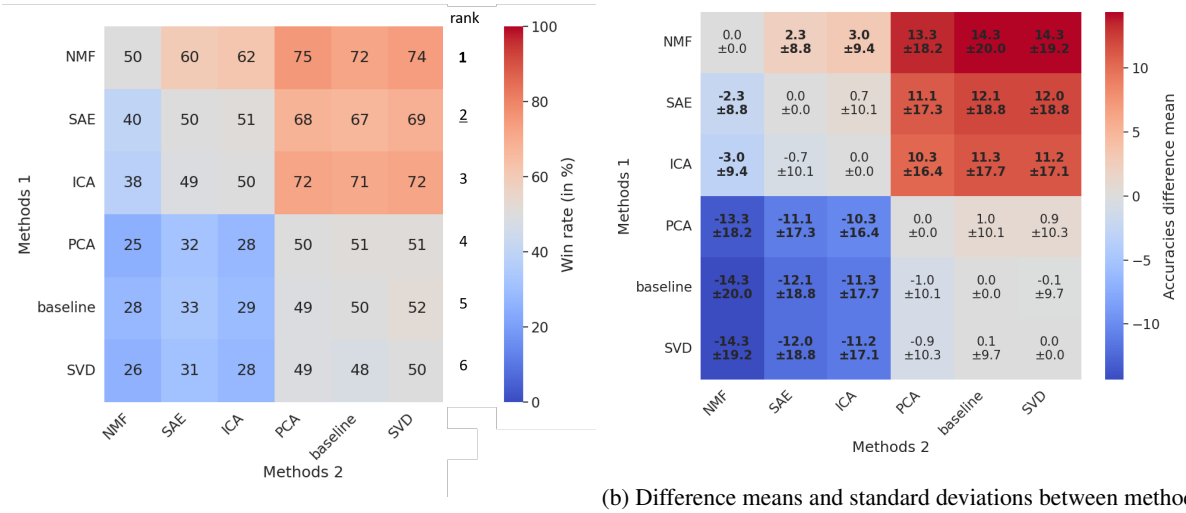
$$\hat{\Phi}_{f_{co},c,cpt} = \frac{\Phi_{f_{co},c,cpt}}{\sum_{i=1}^{k} |\Phi_{f_{co},c,i}|} \quad (9)$$

For a local explanation of sample x with concepts projection $u = f_{ic}(x)$, the normalized local importance for a given concept cpt is given by:

$$\hat{\varphi}_{f_{co}(u)_{cpt}} = \frac{\varphi_{f_{co}(u)_{cpt}}}{\sum_{i=1}^{k} |\varphi_{f_{co}(u)_i}|} \quad (10)$$

B.3 Communicate Importance in Prompts

In several cases, communicating how important some concepts are is necessary. However, LLMs have been proven to be unable to compare numerical values. Furthermore, encoding importance value via a single token would make the comparison far easier. Finding a one-token word to encode these values was not trivial. Hence we opted for the signs "- -", "-", "+", and "+ +". Concepts with low local importance were not shown for local explanations either. To decide what sign to show, we used arbitrary thresholds. The correspondences can be found in Tab. 6.



(a) Percentage of simulatability experiments where method 1 is over method 2. Ranking by number of pairwise victories.

(b) Difference means and standard deviations between method 1 and method 2 simulatability scores across experiments. Bold differences are statistically significant.

Figure 7: Pairwise comparison matrices on Gemini-1.5-Flash experiments described in Sec. 6.1. NMF comes first, and with SAE and ICA, these methods are significantly improving over the baseline (*i.e.* without explanations).

$\hat{\varphi}$ intervals	[-1, -0.3]	]-0.3, -0.05]	[0.05, 0.3[	[0.3, 1]
Encoding	"- -"	"-"	"+"	"+ +"

Table 6: Table of buckets for concept importance encoding in prompts.

C Experiment Settings

C.1 Datasets

Some of the datasets were too small for concept extraction methods to converge to satisfying results. Therefore we artificially increased the datasets by adding modified versions of the samples. The new samples were obtained by splitting the initial ones by punctuation marks. Hence `["This is a first example, made up for understanding."]` becomes `["This is a first example, made up for understanding.", "This is a first example", "made up for understanding"]`.

C.2 Models

DistilBERT and T5. The models used were extracted from HuggingFace. The model cards are: DistilBERT (Sanh et al., 2019b), T5 (team, 2020), Llama-3-8B (AI@Meta, 2024). For DistilBERT and T5, we used the `ModelForSequenceClassification` fine-tuned for each dataset.

DistilBERT+ and T5+. To build the positive versions, we added a ReLU function in the forward pass before the latent space we wanted to study. Then, these models were fine-tuned for the task.

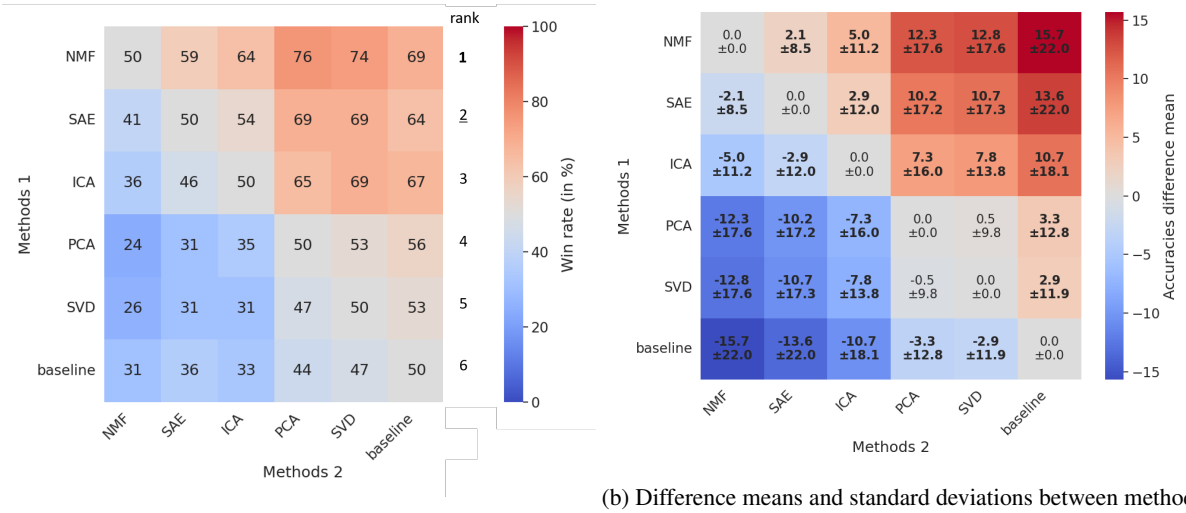
Llama. We adapted `LlamaForCausalLM` to our task through prompting and only considered the next predicted token. The unembedding operation was used as our g part of the model, and we limited it to the classes present in the dataset. The h part was all the rest of the model.

C.3 Concepts Extraction Methods

The goal is to define a concept space $\mathcal{C} \subseteq \mathbb{R}^k$ through the concept transformation $t : \mathcal{H} \longrightarrow \mathcal{C}$ and its bijection (or approximation) $t^{-1} : \mathcal{C} \longrightarrow \mathcal{H}$. We note $X \in \mathcal{X}^n$ a set of samples, $A \in \mathcal{H}^n$ their latent embeddings, and $U \in \mathcal{C}^n$ their projection in the concept space. Similarly, respectively, we note x, a, and u as elements of these sets. Unlike many unsupervised concept-based explicability methods, here we make **a single projection** for the task and not a projection for each predicted class (similar to (Jourdan et al., 2023a)).

Apart from SAE, all implementations are from scikit-learn (Pedregosa et al., 2011) with default parameters apart from the 'n_components' that we vary with the number of required concepts. The used classes are `FastICA`, `NMF`, `PCA`, and `TruncatedSVD`.

Independent Component Analysis (ICA) (Ans et al., 1985; Hyvärinen and Oja, 2000) extracts independent components or sources S such that $S = W \cdot whiten(A)$. The whitening function centers the data on 0. We could write $whiten(a) =$



(a) Percentage of simulatability experiments where method 1 is over method 2. Ranking by number of pairwise victories.

(b) Difference means and standard deviations between method 1 and method 2 simulatability scores across experiments. Bold differences are statistically significant.

Figure 8: Pairwise comparison matrices on Gemini-1.5-Pro experiments described in Sec. 6.1. NMF comes first, and with SAE and ICA, these methods are significantly improving over the baseline (*i.e.* without explanations).

$a - \mu$. Therefore, in our case we could define $t_{ICA}(a) := W \cdot (a - \mu)$. Then we can compute the Moore-Penrose pseudo-inverse W^+, hence we can define $t_{ICA}^{-1}(u) := W^+ u + \mu$.

We use the `FastICA` implementation from scikit-learn (Pedregosa et al., 2011) with default parameters apart from the 'n_components' that we vary with the number of required concepts.

Non-negative Matrix Factorization (NMF) (Lee and Seung, 1999; Sra and Dhillon, 2005) factorizes the matrix A into two matrices U and W such that $A = UW$. The particularity of the NMF is that all three matrices have non-negative weights.

It is easy to construct t^{-1} as U is the concepts activations, thus $t^{-1}(u) = uW$. But this factorization is nonlinear, and we cannot inverse W. Therefore, to obtain U_2 corresponding to other latent embeddings A_2, an U_2 is optimized to fit $A_2 = U_2 W$ with W fixed. Hence, t cannot be defined by matrix multiplications and can only be done by solving an equation.

We use the `NMF` implementation from scikit-learn (Pedregosa et al., 2011) with default parameters apart from the 'n_components' that we vary with the number of required concepts.

Principal Component Analysis (PCA) (Pearson, 1901; Hotelling, 1992) transforms a zero-centered matrix $A - \mu$ into another matrix U through linear combinations W such that $U = (A - \mu)W$. Hence we can define t by $t(a) = (a - \mu)W$ once W is computed, then by investing W we define $t^{-1}(u) = uW^{-1} + \mu$.

We use the `PCA` implementation from scikit-learn (Pedregosa et al., 2011) with default parameters apart from the 'n_components' that we vary with the number of required concepts.

Sparse Auto-Encoder (SAE) (Ng et al., 2011; Makhzani and Frey, 2013; Domingos, 2015) are neural networks whose outputs should be the same as the inputs, the particularity is that some constraints are applied in the middle during their training. Hence t is the encoder and t^{-1} the decoder.

In our case, we follow most of the recommendations from Bricken et al. (2023). In some, we use a $\ell 1$ component with a $1e-3$ coefficient in the loss to push toward sparsity. We apply dead neuron resampling; this part is very sensitive to modifications in the hyperparameters. Finally, we do $100,000$ steps with a learning rate starting at $1e-3$. Note that in some cases, early stopping fires.

Singular Value Decomposition (SVD) (Eckart and Young, 1936) factorizes the matrix A into three components, such that $A = U\Sigma V^T$. In our case, we use $U\Sigma$ as concept activations that we usually denote U. Hence with our notations, we have $A = UV^T$. Since with the SVD, $V^T V = I$, then we can define the projections by $t(a) = aV$ and $t^{-1}(u) = uV^T$.