Cross-Layer Discrete Concept Discovery for Interpreting Language Models

Ankur Garg, Xuemin Yu, Hassan Sajjad & Samira Ebrahimi Kahou

Abstract

Uncovering emergent concepts across transformer layers remains a significant challenge because the residual stream linearly mixes and duplicates information, obscuring how features evolve within large language models. Current research efforts primarily inspect neural representations at single layers, thereby overlooking this cross-layer superposition and the redundancy it introduces. These representations are typically either analyzed directly for activation patterns or passed to probing classifiers that map them to a limited set of predefined concepts. To address these limitations, we propose cross-layer VQ-VAE (CLVQ-VAE), a framework that uses vector quantization to map representations across layers and in the process collapse duplicated residual-stream features into compact, interpretable concept vectors. Our approach uniquely combines top-k temperature-based sampling during quantization with EMA codebook updates, providing controlled exploration of the discrete latent space while maintaining codebook diversity. We further enhance the framework with scaled-spherical k-means++ for codebook initialization, which clusters by directional similarity rather than magnitude, better aligning with semantic structure in word embedding space.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language processing tasks, yet their internal mechanisms remain largely opaque. This opacity poses major challenges for interpretability, limiting our scientific understanding and raising concerns around trust, accountability, and responsible use (Dodge et al., 2021; Sheng et al., 2021).

Current interpretability work typically analyzes representations at individual layers, either through direct examination of activation patterns (Zhang et al., 2021) or by employing probing classifiers that map hidden states to predefined concepts (Belinkov et al., 2017; Arps et al., 2022; Kumar et al., 2023). Such single-layer views ignore how transformer residual streams duplicate and mix information across layers, masking the true computational structure that emerges only when these layers are considered jointly (Team, 2024).

Recent advances in sparse autoencoder (SAE) methods (Härle et al., 2024; Lan et al., 2025) and transcoder architectures (Marks et al., 2024; Dunefsky et al., 2024a) highlight the value of analyzing layer pairs. Empirical studies show that cross-layer analysis often yields more interpretable features than per-layer approaches (Shi et al., 2025; Balagansky et al., 2025; Laptev et al., 2025). This is largely due to the additive residual stream in transformers: each layer contributes to the running representation, causing features to persist and appear duplicated when layers are viewed in isolation (Lindsey et al., 2025). However, SAE-based methods operate in continuous spaces, requiring arbitrary thresholds to interpret sparse activations. This continuity limits interpretability, as the resulting neurons lack discrete conceptual boundaries aligned with human categorical reasoning (Wu et al., 2024).

Vector quantized-variational autoencoder (VQ-VAE) have been extensively explored in computer vision to discretize continuous representations into codebook vectors (van den Oord et al., 2018; Takida et al., 2022; Razavi et al., 2019). Although VQ-VAEs share the reconstruction objective of SAEs, they critically differ by mapping inputs to discrete codebook

vectors rather than continuous activations. When applied to NLP, these codebook vectors can be interpreted as linguistic concepts that are essential for reconstruction.

Building on these insights, we propose CLVQ-VAE, a framework that discovers concepts between transformer layers. Unlike standard VQ-VAEs which reconstruct the same layer, our model acts as a transcoder, mapping activations from a lower layer l to a higher layer h through a discrete bottleneck, hence collapsing redundant residual-stream features into interpretable codebook vectors. We enhance this architecture with two key technical contributions: (1) a stochastic sampling mechanism that selects from top-k nearest codebook vectors using temperature-controlled probability distributions, significantly improving codebook utilization and concept diversity compared to deterministic approaches; and (2) directional similarity-based codebook initialization through scaled-spherical k-means++, which better captures the semantic structure of language embeddings where angular relationships often carry more meaning than Euclidean distances.

We evaluate CLVQ-VAE on ERASER (Pang & Lee, 2004), Jigsaw Toxicity (cjadams et al., 2017), and AGNEWS (Gulli, 2005) using fine-tuned RoBERTa (Liu et al., 2019b), BERT (Devlin et al., 2019), and LLaMA-2-7b models. Perturbation-based experiments show that our approach identifies salient concepts that strongly influence predictions, outperforming clustering, single-layer VQVAE, and SAE baselines. Human evaluation corroborates these findings, with CLVQ-VAE visualisations achieving markedly higher model-alignment and inter-annotator agreement scores than clustering approach, indicating that our discrete concepts faithfully interprets the reason of the prediction.

2 Methodology

We propose CLVQ-VAE, a modular framework for cross-layer concept discovery in transformers that aligns lower-layer quantized representations with higher-layer activations. As shown in Figure 1, the framework comprises three main components: **(i)** an **adaptive residual encoder** that applies minimal, learnable transformations to lower layer embeddings via a controllable interpolation mechanism; **(ii)** a **vector quantizer** that discretizes encoder outputs into concept vectors; and **(iii)** a **transformer decoder** that reconstructs higher-layer representations from the quantized concepts.

The model takes hidden states from a lower transformer layer (e.g., Layer 8), processes them through the encoder to preserve semantic content while enabling controlled transformation, and then maps the result to a discrete latent space via the quantizer. These concept vectors are decoded to approximate a higher-layer activation (e.g., Layer 12). This training is guided by a reconstruction loss that encourages the model to capture persistent, distributed features across layers.

2.1 Adaptive Residual Encoder

Transformer-based language models produce rich, contextualized embeddings at each layer that have been shown to encode diverse linguistic information (Liu et al., 2019a; Sajjad et al., 2022b). A major challenge in designing the encoder is to ensure the preservation of semantic knowledge present in the input embeddings while allowing for subtle transformations that facilitate effective cross-layer concept mapping.

We propose an adaptive residual encoder that implements a controlled manipulation of input embedding. Rather than completely transforming the input, which destroy valuable linguistic features due to random initialization of encoder parameters, or leaving it unchanged, which would limit concept discovery, we introduce a learnable interpolation mechanism. This design respects the information-rich nature of pre-trained language model embeddings while enabling targeted refinements for cross-layer concept identification.

Given an input embedding $\mathbf{x} \in \mathbb{R}^d$ from layer l, the encoder produces an output $\mathbf{z}_e$ through a weighted mixture:

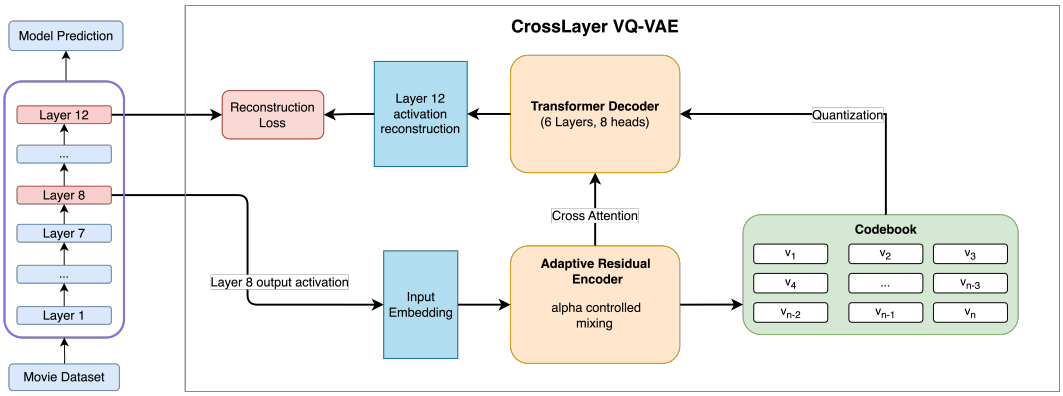

Figure 1: Overview of the CLVQ-VAE framework for cross-layer concept discovery. Lower-layer activations are passed through an adaptive residual encoder, discretized via vector quantization into concept vectors, and decoded to reconstruct higher-layer representations.

$$\mathbf{z}_e = (1 - \alpha) \cdot \mathbf{x} + \alpha \cdot \mathrm{LN}(W\mathbf{x} + \mathbf{b}) \tag{1}$$

where $\alpha = \sigma(a) * 0.5$ is a learnable mixing coefficient constrained to $[0, 0.5]$ and LN is the layer normalization applied to the linearly transformed x. We limit α to a maximum of 0.5 to prevent excessive modification of the original embedding, as we empirically found in Table 8 that allowing complete transformation ($\alpha \in [0,1]$) resulted in reduced codebook utilization. This architecture ensures that the original input can be adaptively modified, preserving the foundation of the representation while allowing the model to learn which aspects to adjust.

In Table 8, we observe that for an adaptive alpha, the model initially prefers a small value of α, which constrains the modification of the original embedding. As training progresses and encoder parameters get trained, the gradients gradually increase α, allowing the model to make progressively greater modifications.

2.2 Vector Quantizer

The vector quantizer maps encoder outputs to discrete concept representations. To promote stable training and effective codebook utilization, we introduce three key mechanisms: spherical k-means++ initialization, temperature-controlled top-k sampling and EMA-based codebook updates.

2.2.1 Codebook Initialization

We initialize our codebook using scaled-spherical k-means++, which clusters input embeddings based on directional similarity rather than Euclidean distance. For each input embedding $\mathbf{x}_i$ from layer l in the training dataset, we compute its unit-normalized form as $\hat{\mathbf{x}}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|_2}$.

After applying k-means++ to these normalized vectors $\hat{x}_i$ to identify cluster centroids c_j, we scale each centroid by the average magnitude of vectors in its cluster:

$$\mathbf{e}_j = \mathbf{c}_j \cdot \frac{1}{|C_j|} \sum_{i \in C_j} \|\mathbf{x}_i\|_2 \tag{2}$$

where e_j is the scaled centroid for cluster j and C_j is the set of indices for vectors assigned to cluster j. This initialization strategy better aligns with semantic relationships in embedding

spaces, where direction often captures meaning more effectively than absolute position, while still preserving magnitude information.

2.2.2 *Top-k Temperature-Based Codebook Sampling*

Our vector quantization mechanism employs temperature-based sampling (Takida et al., 2022) from the top-k nearest codebook vectors. For each encoder output $\mathbf{z}_e$, we compute distances to all codebook vectors $\{\mathbf{e}_j\}_{j=1}^{K}$ as:

$$d(\mathbf{z}_e, \mathbf{e}_j) = \|\mathbf{z}_e - \mathbf{e}_j\|_2^2 \tag{3}$$

Rather than deterministically selecting the closest vector, we identify the top-k nearest codebook vectors and sample from them using a temperature-controlled distribution:

$$p(j|\mathbf{z}_e) = \frac{\exp(-d(\mathbf{z}_e, \mathbf{e}_j)/\tau)}{\sum_{j' \in \text{top-}k} \exp(-d(\mathbf{z}_e, \mathbf{e}_{j'})/\tau)} \tag{4}$$

where τ is the temperature parameter. In our optimal configuration, we set $k = 5$ and $\tau = 1.0$, balancing exploration with exploitation. This controlled stochasticity during training encourages more uniform codebook utilization, reduces codebook collapse, and improves concept capture.

2.2.3 *EMA-Based Codebook Updates*

The codebook is updated using exponential moving average (EMA) [1] (Łukasz Kaiser et al., 2018) rather than backpropagation, providing more stable training dynamics essential for discrete representation learning.

Instead of gradient-based updates that can oscillate or become unstable with discrete assignments (van den Oord et al., 2018), we maintain running averages of both the assignment counts and accumulated vectors for each codebook entry, and then update each codebook vector to the EMA-weighted average of its assigned embeddings.

This update strategy, combined with our temperature-based sampling approach, ensures the codebook remains diverse and active throughout training while converging toward stable representations of cross-layer transformations.

2.3 Transformer Decoder

The final component of our CLVQ-VAE architecture is a transformer-based decoder that maps quantized representations to higher-layer activations. This decoder reconstructs the transformations that occur between layers l and h in the original language model.

Our decoder follows the standard transformer decoder architecture with self-attention and cross-attention mechanisms. It consists of 6 layers with 8 attention heads each. The self-attention employs causal masking to preserve autoregressive properties, while the cross-attention allows the decoder to leverage information from the unquantized encoder outputs, providing a residual connection around the quantization bottleneck.

2.4 Training Objectives

Our training incorporates two weighted objectives combined into a single loss function. The primary objective is the reconstruction loss, which minimizes the mean squared error between the decoder output $\hat{\mathbf{y}}$ and the target higher-layer representation $\mathbf{y}$, defined as $\mathcal{L}_{\text{rec}} = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$ (Dunefsky et al., 2024a). This ensures that the model captures the transformations occurring between neural network layers. Additionally, we employ a commitment loss

[1] More detail on implementation of EMA can be found in A.1

that encourages encoder outputs to commit to codebook vectors, calculated as $\mathcal{L}_{\text{commit}} = \|\mathbf{z}_e - \text{sg}(\mathbf{z}_q)\|_2^2$, where sg denotes the stop-gradient operator. The total loss is given by $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{commit}}$, where β controls the relative importance of the commitment term. We set $\beta = 0.1$ (van den Oord et al., 2018) to avoid constraining the encoder output too strictly (Wu & Flierl, 2019).

3 Experimental Setup

Data We conducted experiments for CLVQ-VAE on three datasets across different task types: ERASER movie review dataset (Pang & Lee, 2004) for sentiment classification, Jigsaw Toxicity dataset (cjadams et al., 2017) for toxicity classification, and AGNEWS dataset (Gulli, 2005) for multi-class news categorization.

Model We use RoBERTa base (Liu et al., 2019b) and BERT base (Devlin et al., 2019) after fine-tuning on the respective datasets, and LLaMA-2-7b (Touvron et al., 2023) with a task specific prompt and without finetuning. From these models, we extract paired activations: representations from a lower layer l serve as input to CLVQ-VAE, while representations from a higher layer h serve as reconstruction targets.

Activation Extraction We utilized the NeuroX toolkit (Dalvi et al., 2023) to extract paired activations from RoBERTa and BERT models. A key advantage of NeuroX is its ability to aggregate sub-word token representations of RoBERTa tokenizer into word-level activations. This aggregation enables the mapping of discovered concepts to complete words rather than fragmented tokens, allowing us to employ visualization techniques such as word clouds to represent and analyze the discrete concepts identified by CLVQ-VAE.

Evaluation For our primary evaluation across models and datasets, we apply CLVQ-VAE between transformer layers 8 and 12. This specific layer pair is chosen based on empirical analysis, which we detail in Appendix A.4. For comparison, we include three baselines: the clustering-based method from Yu et al. (2024), a single-layer VQ-VAE variant (referred to as "Single-Layer VQ-VAE"), and a cross-layer sparse autoencoder (SAE).

4 Evaluation

4.1 Faithfulness Evaluation

To quantitatively assess the ability of CLVQ-VAE to identify salient concepts between transformer layers, we adopt a perturbation-based evaluation protocol. This approach measures how effectively each method can identify concept representations that influence the model's predictions.

4.1.1 Perturbation Methodology

We evaluate the efficacy of CLVQ-VAE in identifying salient concepts through concept ablation experiments. Specifically, we remove the codebook vector corresponding to the salient token, from the CLS representation in test sentences and measure the resulting drop in model performance.

To perform this analysis, we first extract sentence-level embeddings from layer l (the input to CLVQ-VAE) for each sentence. For encoder-based models like BERT and RoBERTa, we use the CLS token embedding as sentence embedding. For decoder-only models like LLaMA, which lack a CLS token, we instead use the final token embedding from the generated sequence. This choice is motivated by the autoregressive nature of decoder-only models, where the last token captures contextual information from all preceding tokens, making it a functional equivalent to CLS. We apply the same saliency attribution and ablation methods to these embeddings across all models.

We then construct three variants of these embeddings:

Method	ERASER-Movie		Jigsaw	
	RoBERTa	BERT	RoBERTa	BERT
Clustering	0.6271 ± 0.1010	0.7757 ± 0.0723	**0.5628 ± 0.0715**	0.7577 ± 0.0895
Single-Layer (Layer 8)	0.0945 ± 0.0456	0.7430 ± 0.0655	0.8892 ± 0.0490	0.8177 ± 0.0624
Single-Layer (Layer 12)	0.6999 ± 0.0747	0.9328 ± 0.0352	0.6606 ± 0.1030	0.9135 ± 0.0528
SAE (2048 neurons)	0.2268 ± 0.0549	0.8645 ± 0.0526	0.9134 ± 0.0450	0.8944 ± 0.0618
SAE (4096 neurons)	0.8705 ± 0.0478	0.8713 ± 0.0524	0.9134 ± 0.0485	0.8931 ± 0.0568
CLVQ-VAE	**0.0782 ± 0.0454**	**0.6694 ± 0.0509**	0.6456 ± 0.1334	**0.7360 ± 0.0823**

Table 1: Comparison of faithfulness evaluation across RoBERTa and BERT on ERASER-Movie and Jigsaw. Lower values indicate better concept identification performance.

- **Original CLS**: Unmodified CLS token embeddings.
- **Perturbed CLS**: CLS embeddings with their most salient concept removed. For each method (Clustering, Single-Layer VQ-VAE, SAE, and CLVQ-VAE), we identify the most important concept vector associated with the most salient token. In VQ-VAE-based approaches, this corresponds to the nearest codebook vector. In clustering, it is the closest cluster centroid. For SAE, it is the decoder vector associated with the most highly activated neuron. The identified concept vector is then removed from the CLS embedding via orthogonal projection.
- **Random Perturbed CLS**: CLS embeddings with a random direction removed using orthogonal projection. This serves as a control condition to verify that performance drops are due to removing meaningful concepts rather than arbitrary perturbations.

To evaluate the effects of concept ablation experiments on CLS token, we employ a probe-based assessment approach. First, we train a simple probe (a 2-layer neural network with dropout) using the original unmodified CLS embeddings as input and task labels as targets. We then evaluate this trained classifier on the three variants of the embeddings: the original CLS, the perturbed CLS, and the random-perturbed CLS.

The key insight behind this approach is that removing truly important concepts from the embedding space should cause a significant drop in classification performance. In contrast, removing random vectors is expected to have minimal impact. Consequently, methods that yield larger drops in accuracy under embedding perturbation, relative to the original embeddings, demonstrate stronger concept identification capabilities.

4.1.2 Baseline Evaluation

As shown in Table 1, CLVQ-VAE achieves the lowest scores on ERASER-Movie across both RoBERTa and BERT, indicating the most faithful identification of influential concepts in that dataset. On Jigsaw, clustering outperforms CLVQ-VAE under RoBERTa, but CLVQ-VAE remains the best-performing method across all other model–dataset combinations.

These results highlight the robustness of CLVQ-VAE in consistently identifying predictive concept directions that impair model performance when removed. In contrast, random perturbations have minimal impact, confirming that CLVQ-VAE is not merely reacting to noise but targeting meaningful latent directions. [2]

4.2 Architectural Analysis

We further investigate the impact of different architectural choices on concept identification performance using the same ablation-based faithfulness metric.

Temperature and Top-k Sampling Analysis: Our stochastic sampling mechanism uses temperature-controlled probability distributions to select from top-k nearest codebook

[2]Reference baseline values for faithfulness evaluation (Original CLS and Random Perturbation accuracies across all model-dataset combinations and baselines) are provided in A.2.

Temperature	Top-k	Validation Perplexity
0.5	5	207.14
1.0	5	210.63
2.0	5	217.14
3.0	5	220.09
1.0	1	207.07
1.0	10	210.37
1.0	50	210.17
1.0	100	210.51

Table 2: Impact of temperature and top-k parameters on codebook utilization (validation perplexity) for Eraser Movie on RoBERTa model.

Model	Dataset	Spherical	K-means++	Random
BERT	Jigsaw	**0.7360 ± 0.0823**	0.7629 ± 0.0750	0.8216 ± 0.0553
BERT	ERASER-Movie	0.6694 ± 0.0509	0.7617 ± 0.0688	**0.6136 ± 0.0293**
BERT	AGNEWS	0.7208 ± 0.0513	0.7350 ± 0.0497	0.7142 ± 0.0214
RoBERTa	Jigsaw	**0.6456 ± 0.1334**	0.6596 ± 0.1282	0.6706 ± 0.0499
RoBERTa	ERASER-Movie	**0.0782 ± 0.0454**	0.1402 ± 0.0499	0.0924 ± 0.0404
RoBERTa	AGNEWS	0.1058 ± 0.0243	0.1067 ± 0.0305	0.1017 ± 0.0210
LLaMA-2-7b	ERASER-Movie	0.8095 ± 0.0596	**0.7779 ± 0.0691**	0.8075 ± 0.0303
LLaMA-2-7b	Jigsaw	**0.7524 ± 0.0836**	0.7736 ± 0.0864	0.7775 ± 0.0398

Table 3: CLVQ-VAE performance across multiple models and datasets with different initialization methods. Lower values indicate better concept identification performance.

vectors. Table 2 shows the impact of different temperature and top-k values on codebook utilization measured by validation perplexity.

Increasing temperature from 0.5 to 3.0 increases validation perplexity from 207.14 to 220.09, reflecting greater exploration in codebook selection. For top-k values with τ=1.0, we observe slower increases in perplexity, indicating that the exploration-exploitation balance shifts more gradually compared to temperature adjustments. We chose conservative values (τ=1.0, k=5) to provide stable training dynamics while maintaining reasonable codebook diversity.

Codebook Initialization Comparison: Table 3 shows results across BERT, RoBERTa, and LLaMA models on ERASER, Jigsaw, and AGNEWS[3] using three initialization strategies: spherical k-means++, standard k-means++, and random token selection. Spherical k-means++ consistently outperforms standard k-means++ in most configurations (e.g., RoBERTa-Jigsaw: 0.6456 vs. 0.6596; LLaMA-Jigsaw: 0.7524 vs. 0.7736), confirming that angular similarity provides a better inductive bias for capturing semantic structure in embedding space. Although spherical k-means++ and random initialization show comparable average performance, random initialization exhibits high variance across different seeds, leading to inconsistent outcomes. In contrast, spherical k-means++ provides more deterministic and reproducible initialization through robust cluster formation, as demonstrated by previous literature on k-means++ (Celebi et al., 2013).

Codebook Size: Table 7 shows the impact of varying codebook size K on performance and utilization. Although faithfulness scores vary only slightly across configurations, perplexity reveals meaningful trends in codebook usage. At $K = 400$, CLVQ-VAE achieves a perplexity of 198.776—corresponding to approximately 50% utilization—striking a good balance between capacity and efficiency. Smaller codebooks (e.g., $K = 50$) lead to high utilization but often force multiple concepts to share the same codebook vector, reducing

[3]AGNEWS results are not available for LLaMA.

Method	Fleiss' Kappa (κ)	Avg. Confidence	Model Alignment Rate
Clustering	0.59	5.981	54.14%
CLVQ-VAE	**0.864**	**8.44**	**78.20%**

Table 4: Human evaluation results comparing CLVQ-VAE with baseline clustering approach for ERASER Movie dataset. Higher values indicate better performance across all metrics

interpretability. In contrast, very large codebooks (e.g., $K = 800$ or $K = 1200$) underutilize available entries and may fragment the representation space.

4.3 Human Evaluation

To complement our quantitative experiments, we conducted a human evaluation study comparing CLVQ-VAE approach with clustering baseline. This evaluation aims to assess each method's effectiveness in visually communicating the model's reasoning process to human observers.

4.3.1 Methodology

We randomly selected 19 representative sentences from the ERASER movie review dataset, ensuring an equal proportion of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) with respect to the model's predictions. This balanced selection allows us to evaluate concepts underlying both correct and incorrect model predictions. For each sentence, we generate word clouds using the tokens that were mapped—during training—to the codebook vector associated with the sentence's most salient token, for both the CLVQ-VAE and clustering approaches.

14 annotators participated in the study, collectively reviewing a total of 266 samples. Each annotator was presented with the generated word cloud of each sentence without being shown the original sentence, model prediction, or ground truth label. This design ensures that annotators relied solely on the visualizations to infer model behavior. For each visualization, annotators were asked to:

1. Predict the model's sentiment label using only the information in the word cloud.
2. Rate their confidence in this prediction on a scale of 1-10 (where 10 is highest)

4.3.2 Evaluation Metrics

We assessed the quality of visualizations using three complementary metrics:

- **Fleiss' Kappa (κ)**: Measures inter-annotator agreement beyond chance. This metric indicates whether different annotators derive consistent interpretations from the same visualization, with values ranging from -1 (worse than chance) to 1 (perfect agreement).
- **Average Confidence**: The mean self-reported confidence (1-10) across all annotators and examples, indicating how certain users felt about their interpretations.
- **Model Alignment Rate**: The percentage of annotator predictions that matched the model's actual prediction, regardless of the model's correctness. This metric measures how effectively each visualization communicates the model's reasoning.

4.3.3 Results

Table 4 presents the results of the human evaluation comparing the clustering approach (Yu et al., 2024) with our CLVQ-VAE framework. CLVQ-VAE achieved substantially higher inter-annotator agreement ($\kappa = 0.864$, "almost perfect agreement") compared to clustering ($\kappa = 0.59$, "moderate agreement"), suggesting more consistent interpretations across users. Annotators also reported higher confidence (8.44 vs. 5.981) when interpreting CLVQ-VAE visualizations, indicating greater conceptual clarity. Most importantly, model alignment

was over 24 percentage points higher (78.20% vs. 54.14%), showing that CLVQ-VAE more faithfully communicates the model's reasoning.

5 Related Work

Traditional interpretability methods for NLP models, such as gradient-based and perturbation-based techniques (Sundararajan et al., 2017; Kapishnikov et al., 2021; Rajagopal et al., 2021; Zhao & Aletras, 2023), assess input feature contributions to predictions but often fail to reveal internal decision-making processes. Similarly, the representation analysis literature provides insights into whether a predefined concept is learned in the representation and how such knowledge is structured in neurons of the model (Dalvi et al., 2019; Sajjad et al., 2022a; Gurnee et al., 2023). However, it is limited by the need for predefined concepts and annotated data for probing (Antverg & Belinkov, 2022), while neuron-level interpretation is further complicated by the polysemantic and superpositional nature of neurons (Haider et al., 2025; Elhage et al., 2022; Fan et al., 2023).

Concept-based approaches aim to address some of these limitations by interpreting model behavior through high-level, human-understandable concepts. Techniques like TCAV measure model sensitivity to predefined concepts via directional derivatives in activation space (Kim et al., 2018), though they still rely on manually specified concepts. Recent methods move toward discovering latent concepts directly from internal representations, enabling a deeper and more flexible understanding of model functionality (Ghorbani et al., 2019; Dalvi et al., 2022; Jourdan et al., 2023; Yu et al., 2024).

Sparse autoencoders (SAEs) (Härle et al., 2024) have been utilized to extract interpretable features from large language models (LLMs). However, studies have highlighted challenges in their stability and utility. For instance, SAEs trained with different random seeds on the same data can learn divergent feature sets, indicating sensitivity to initialization (Paulo & Belrose, 2025). Furthermore, their performance on downstream tasks does not consistently surpass baseline methods, questioning their practical benefits (Kantamneni et al., 2025). To mitigate these issues, ensemble methods combining multiple SAEs have been proposed, resulting in improved reconstruction, feature diversity, and stability (Gadgil et al., 2025).

Cross-layer interpretability has garnered attention, with sparse crosscoders introduced to capture and understand features across different model layers (Lindsey et al., 2024). These methods facilitate tasks like model diffing and circuit analysis, enabling the tracking of shared and unique features across layers and providing deeper insights into model behavior (Minder et al., 2025; Dunefsky et al., 2024b).

While VQ-VAE have shown success in domains like image and speech processing (van den Oord et al., 2018; Łukasz Kaiser et al., 2018; Huang et al., 2023; Guo et al., 2020; Huang & Ji, 2020; Bhardwaj et al., 2022), their application in NLP interpretability remains underexplored. CLVQ-VAE framework addresses this gap by integrating transcoder-inspired objectives to map lower-layer representations to higher-layer ones.

6 Conclusion

We presented CLVQ-VAE, a framework for discovering discrete concepts that are superposed across transformers layers. CLVQ-VAE consistently outperformed baseline methods in identifying salient and meaningful concepts. Human evaluations further support its effectiveness against clustering approach. The effectiveness of our approach is supported by three design choices: an adaptive residual encoder that updates the codebook with minimal input distortion, temperature-based top-k sampling that preserves diversity and stability of codebook, and scaled-spherical k-means++ initialization that captures directional similarity aligned with semantic structure in language models.

CLVQ-VAE represents a significant step in the interpretability of LLMs, offering insights into the concepts captured and transformed across layers, and contributing to the broader goal of making deep learning models more explainable.

7 Limitations

Despite strong performance over baselines, several limitations affect how improvements in CLVQ-VAE can be measured and interpreted:

Faithfulness Metric Sensitivity: Our perturbation-based evaluation, though useful for comparing broad architectural choices, lacks the granularity to capture smaller changes—such as variations in temperature, top-k, or codebook size K (see Appendix 10). This highlights the need for more sensitive metrics to evaluate subtle shifts in concept quality.

Limited Human Evaluation: While our human study shows promising results, it was conducted on a small set of 19 examples. A larger-scale evaluation would offer more robust validation of our method's interpretability benefits.

Ethics Statement

CLVQ-VAE advances language model interpretability through cross-layer discrete concept discovery. While common concerns in explainability research still exist (including concept completeness and evaluation subjectivity), this work is primarily theoretical. Our experiments use standard benchmarks and focus on methodological innovations rather than applications posing ethical risks. These insights contribute to transparent AI systems and may support future work on bias detection without introducing novel ethical challenges beyond those inherent to interpretability research.

Acknowledgement

We acknowledge the support and funding of CIFAR, Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada Foundation for Innovation (CFI), and Research Nova Scotia. This research was enabled through computational resources provided by the Research Computing Services group at the University of Calgary, the Denvr Dataworks platform, ACENET—the regional partner in Atlantic Canada—and the Digital Research Alliance of Canada.

References

Omer Antverg and Yonatan Belinkov. On the pitfalls of analyzing individual neurons in language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=8uz0EWPQIMu.

David Arps, Younes Samih, Laura Kallmeyer, and Hassan Sajjad. Probing for constituency structure in neural language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 6738–6757, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.502. URL https://aclanthology.org/2022.findings-emnlp.502/.

Nikita Balagansky, Ian Maksimov, and Daniil Gavrilov. Mechanistic permutability: Match features across layers, 2025. URL https://arxiv.org/abs/2410.07656.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. What do neural machine translation models learn about morphology? In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 861–872, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1080. URL https://aclanthology.org/P17-1080/.

Rishabh Bhardwaj, Amrita Saha, Steven C.H. Hoi, and Soujanya Poria. Vector-quantized input-contextualized soft prompts for natural language understanding. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical*

Methods in Natural Language Processing, pp. 6776–6791, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.455. URL https://aclanthology.org/2022.emnlp-main.455/.

M. Celebi, H. Kingravi, and P. Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems With Applications*, 40:200–210, 2013. doi: 10.1016/j.eswa.2012.07.021.

cjadams, Jeffrey Sorensen, Julia Elliott, Lucas Dixon, Mark McDonald, nithum, and Will Cukierski. Toxic comment classification challenge, 2017. URL https://kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, D. Anthony Bau, and James Glass. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, March 2019.

Fahim Dalvi, Abdul Rafae Khan, Firoj Alam, Nadir Durrani, Jia Xu, and Hassan Sajjad. Discovering latent concepts learned in BERT. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=POTMtpYI1xH.

Fahim Dalvi, Nadir Durrani, and Hassan Sajjad. Neurox library for neuron analysis of deep nlp models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 75–83, Toronto, Canada, July 2023. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1286–1305, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.98. URL https://aclanthology.org/2021.emnlp-main.98/.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable llm feature circuits, 2024a. URL https://arxiv.org/abs/2406.11944.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable llm feature circuits. *arXiv preprint arXiv:2406.11944*, 2024b. URL https://arxiv.org/abs/2406.11944.

Nelson Elhage et al. Superposition, memorization, and double descent. *Transformer Circuits*, 2022.

Yimin Fan, Fahim Dalvi, Nadir Durrani, and Hassan Sajjad. Evaluating neuron interpretation methods of nlp models. In *Thirty-seventh Conference on Neural Information Processing Systems*, Dec 2023. URL https://openreview.net/forum?id=YiwMpyMdPX.

Soham Gadgil, Chris Lin, and Su-In Lee. Ensembling sparse autoencoders. *arXiv preprint arXiv:2505.16077*, 2025. URL https://arxiv.org/abs/2505.16077.

Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations, 2019. URL https://arxiv.org/abs/1902.03129.

Antonio Gulli. Ag's corpus of news articles, 2005. URL http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html.

Daya Guo, Duyu Tang, Nan Duan, Jian Yin, Daxin Jiang, and Ming Zhou. Evidence-aware inferential text generation with vector quantised variational autoencoder, 2020. URL https://arxiv.org/abs/2006.08101.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing, 2023. URL https://arxiv.org/abs/2305.01610.

Muhammad Umair Haider, Hammad Rizwan, Hassan Sajjad, Peizhong Ju, and A. B. Siddique. Neurons speak in ranges: Breaking free from discrete neuronal attribution, 2025. URL https://arxiv.org/abs/2502.06809.

Lifu Huang and Heng Ji. Semi-supervised new event type induction and event detection. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 718–724, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.53. URL https://aclanthology.org/2020.emnlp-main.53/.

Mengqi Huang, Zhendong Mao, Zhuowei Chen, and Yongdong Zhang. Towards accurate image coding: Improved autoregressive image generation with dynamic vector quantization, 2023. URL https://arxiv.org/abs/2305.11718.

Ruben Härle, Felix Friedrich, Manuel Brack, Björn Deiseroth, Patrick Schramowski, and Kristian Kersting. Scar: Sparse conditioned autoencoders for concept detection and steering in llms, 2024. URL https://arxiv.org/abs/2411.07122.

Fanny Jourdan, Agustin Picard, Thomas Fel, Laurent Risser, Jean Michel Loubes, and Nicholas Asher. Cockatiel: Continuous concept ranked attribution with interpretable elements for explaining neural net classifiers on nlp tasks, 2023. URL https://arxiv.org/abs/2305.06754.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are sparse autoencoders useful? a case study in sparse probing. *arXiv preprint arXiv:2502.16681*, 2025. URL https://arxiv.org/abs/2502.16681.

Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. Guided integrated gradients: An adaptive path method for removing noise. *CoRR*, abs/2106.09788, 2021. URL https://arxiv.org/abs/2106.09788.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Abhinav Kumar, Balaji Srinivasan, Avi Feng, Divyat Mahajan, et al. Probing classifiers are unreliable for concept removal and detection. In *International Conference on Machine Learning*, pp. –, 2023. URL https://arxiv.org/abs/2207.04153.

Michael Lan, Philip Torr, Austin Meek, Ashkan Khakzar, David Krueger, and Fazl Barez. Sparse autoencoders reveal universal feature spaces across large language models, 2025. URL https://arxiv.org/abs/2410.06981.

Daniil Laptev, Nikita Balagansky, Yaroslav Aksenov, and Daniil Gavrilov. Analyze feature flow to enhance interpretation and steering in language models, 2025. URL https://arxiv.org/abs/2502.03032.

Jack Lindsey, Adly Templeton, Jonathan Marcus, Tom Conerly, Joshua Batson, and Chris Olah. Sparse crosscoders for cross-layer features and model diffing. https://transformer-circuits.pub/2024/crosscoders/index.html, 2024.

Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. Sparse crosscoders for cross-layer features and model diffing, 2025. URL https://transformer-circuits.pub/2024/crosscoders/index.html.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1073–1094, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1112. URL https://aclanthology.org/N19-1112/.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv:1907.11692*, 2019b. URL https://arxiv.org/abs/1907.11692.

Samuel Marks, Adam Karvonen, and Aaron Mueller. Dictionary learning. https://github.com/saprmarks/dictionary_learning, 2024.

Julian Minder, Clément Dumas, Caden Juang, Bilal Chugtai, and Neel Nanda. Robustly identifying concepts introduced during chat fine-tuning using crosscoders. *arXiv preprint arXiv:2504.02922*, 2025. URL https://arxiv.org/abs/2504.02922.

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, pp. 271–es, USA, 2004. Association for Computational Linguistics. doi: 10.3115/1218955.1218990. URL https://doi.org/10.3115/1218955.1218990.

Gonçalo Paulo and Nora Belrose. Sparse autoencoders trained on the same data learn different features. *arXiv preprint arXiv:2501.16615*, 2025. URL https://arxiv.org/abs/2501.16615.

Dheeraj Rajagopal, Vidhisha Balachandran, Eduard H Hovy, and Yulia Tsvetkov. SELFEXPLAIN: A self-explaining architecture for neural text classifiers. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 836–850, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.64. URL https://aclanthology.org/2021.emnlp-main.64/.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2, 2019. URL https://arxiv.org/abs/1906.00446.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. Neuron-level interpretation of deep nlp models: A survey. *Transactions of the Association for Computational Linguistics*, 2022a.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Firoj Alam, Abdul Khan, and Jia Xu. Analyzing encoded concepts in transformer language models. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3082–3101, Seattle, United States, July 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.225. URL https://aclanthology.org/2022.naacl-main.225/.

Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. Societal biases in language generation: Progress and challenges. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4275–4293, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.330. URL https://aclanthology.org/2021.acl-long.330/.

Wei Shi, Sihang Li, Tao Liang, Mingyang Wan, Gojun Ma, Xiang Wang, and Xiangnan He. Route sparse autoencoder to interpret large language models, 2025. URL https://arxiv.org/abs/2503.08200.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *CoRR*, abs/1703.01365, 2017. URL http://arxiv.org/abs/1703.01365.

Yuhta Takida, Takashi Shibuya, WeiHsiang Liao, Chieh-Hsin Lai, Junki Ohmura, Toshimitsu Uesaka, Naoki Murata, Shusuke Takahashi, Toshiyuki Kumakura, and Yuki Mitsufuji. Sq-vae: Variational bayes on discrete representation with self-annealed stochastic quantization, 2022. URL https://arxiv.org/abs/2205.07547.

Transformer Circuits Team. Sparse crosscoders for cross-layer features and model diffing. *Distill*, 2024. URL https://transformer-circuits.pub/2024/crosscoders/index.html.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. URL https://arxiv.org/abs/2307.09288.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. URL https://arxiv.org/abs/1711.00937.

Hanwei Wu and Markus Flierl. Learning product codebooks using vector quantized autoencoders for image retrieval, 2019. URL https://arxiv.org/abs/1807.04629.

Yi-Fu Wu, Minseung Lee, and Sungjin Ahn. Neural language of thought models, 2024. URL https://arxiv.org/abs/2402.01203.

Xuemin Yu, Fahim Dalvi, Nadir Durrani, Marzia Nouri, and Hassan Sajjad. Latent concept-based explanation of nlp models, 2024. URL https://arxiv.org/abs/2404.12545.

Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *arXiv preprint arXiv:–*, 2021. URL https://petertino.github.io/web/PAPERS/Yu_Survey_Interpret_DNN.pdf.

Zhixue Zhao and Nikolaos Aletras. Incorporating attribution importance for improving faithfulness metrics. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4732–4745, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.261. URL https://aclanthology.org/2023.acl-long.261/.

Łukasz Kaiser, Aurko Roy, Ashish Vaswani, Niki Parmar, Samy Bengio, Jakob Uszkoreit, and Noam Shazeer. Fast decoding in sequence models using discrete latent variables, 2018. URL https://arxiv.org/abs/1803.03382.

A Appendix

A.1 EMA update details

During training, we perform temperature-based top-k sampling to select codebook vectors, then apply EMA updates using hard assignments:

For each codebook vector $\mathbf{e}_j$:

$$N_j^{(t)} = \gamma N_j^{(t-1)} + (1-\gamma) \sum_i \mathbb{I}[j \text{ sampled for } \mathbf{z}_e^{(i)}]$$

$$\mathbf{m}_j^{(t)} = \gamma \mathbf{m}_j^{(t-1)} + (1-\gamma) \sum_i \mathbf{z}_e^{(i)} \mathbb{I}[j \text{ sampled for } \mathbf{z}_e^{(i)}]$$

$$\mathbf{e}_j^{(t)} = \frac{\mathbf{m}_j^{(t)}}{N_j^{(t)}}$$

where $\gamma = 0.99$, $N_j^{(t)}$ tracks assignment counts, $\mathbf{m}_j^{(t)}$ accumulates vectors, and $\mathbb{I}[\cdot]$ indicates whether vector j was selected via stochastic top-k sampling. This provides stable codebook updates with improved utilization.

A.2 Reference baseline values

Table 5 provides the original CLS and random perturbed CLS accuracy values used as baselines across all faithfulness evaluation experiments. These values serve as reference points for calculating performance drops when salient concepts are removed from model representations.

Model	Dataset	Original CLS	Random Perturbed CLS
BERT	Jigsaw	0.8983 ± 0.0568	0.8995 ± 0.0549
BERT	ERASER-Movie	0.8248 ± 0.0477	0.8247 ± 0.0454
BERT	AGNEWS	0.7492 ± 0.0581	0.7492 ± 0.0620
RoBERTa	Jigsaw	0.9134 ± 0.0464	0.9134 ± 0.0464
RoBERTa	ERASER-Movie	0.7604 ± 0.1317	0.7264 ± 0.1152
RoBERTa	AGNEWS	0.7275 ± 0.0528	0.6850 ± 0.0591
LLaMA-2-7b	ERASER-Movie	0.9078 ± 0.0491	0.9054 ± 0.0528
LLaMA-2-7b	Jigsaw	0.8217 ± 0.0802	0.8217 ± 0.0802

Table 5: Reference baseline values for faithfulness evaluation across different model-dataset combinations.

A.3 Hyperparameter for reproducibility

We list the key hyperparameters used across model architecture, training, and quantization in Table 6. Unless otherwise noted, all experiments use this configuration. Adaptive α is capped at 0.5, and all weights are initialized using standard PyTorch defaults, with the encoder residual path initialized as an identity transformation.

Component	Hyperparameter Value
Model Architecture	
Embedding dim (d)	768
Codebook size (K)	400
Commitment cost (β)	0.1
Decoder layers	6
Decoder heads	8
Feedforward dim	2048
Dropout	0.1
Quantization and Sampling	
Sampling method	Top-k sampling
Top-k	5
Sampling temperature (τ)	1.0
Encoder Interpolation	
α schedule	Adaptive (learned)
Max α scaling factor	0.5
Optimization and Training	
Optimizer	Adam
Learning rate	5e−3
Weight decay	1e−4
Scheduler	ReduceLROnPlateau
Batch size	128
Training epochs	50 (with early stopping)
Seed	42
Device	GPU (if available)

Table 6: Hyperparameter configuration used in all primary experiments.

A.3.1 Codebook Size Analysis

Codebook Size (K)	Perturbed CLS	Perplexity (Utilization %)
50	0.0769 ± 0.0498	39.813 (79.6%)
100	0.0665 ± 0.0436	58.145 (58.1%)
400	0.0782 ± 0.0454	198.776 (49.7%)
800	0.1214 ± 0.0502	257.776 (32.2%)
1200	0.1004 ± 0.0460	316.051 (26.3%)
Reference values:		
Original CLS: 0.7604 ± 0.1317	Random Perturbed CLS: 0.7264 ± 0.1152	

Table 7: Impact of codebook size (K) on concept identification performance and codebook utilization.

A.3.2 Adaptive Alpha Parameter Study

Table 8 shows the impact of different α strategies on training dynamics and codebook utilization across training epochs on Eraser-movie dataset and RoBERTa model.

These results reveal that training of adaptive α behaves like a curriculum mechanism: it begins with low values that preserve the original input embeddings and gradually increases to allow more expressive transformations. For instance, the limited adaptive setting starts around 0.28 and converges to 0.45, achieving high perplexity from early epochs and maintaining it consistently. This facilitates effective concept discovery while optimizing for low validation loss. In contrast, fixed low α values such as 0.1 retain high perplexity but restrict the model's ability to adapt representations, resulting in higher loss. On the other hand, fixed high α values (e.g., 0.75 or 1.0) take significantly longer to

Alpha Strategy	Initial	Epoch 10	Epoch 30	Final	Best Val Loss
Adaptive (Limited)	1.97	198.5	216	210.6	0.033
Adaptive (Complete)	1.86	25.54	50.70	63.21	0.033
Fixed α=0.0	238.1	237.3	239.8	237.2	0.045
Fixed α=0.1	180.9	232.3	238.1	230.8	0.040
Fixed α=0.4	1.95	126.9	160.2	157.2	0.036
Fixed α=0.75	1.077	1.883	27.517	40.106	0.033
Fixed α=1.0	1.155	2.397	15.219	147.470	0.032

Table 8: Alpha parameter analysis showing perplexity evolution across training epochs and final validation loss.

reach useful perplexity levels, delaying convergence. Notably, when $\alpha = 0$, the encoder remains an identity function throughout training and becomes decoupled from decoder and quantization gradients, leading to stagnation.

We limit the adaptive α to do a maximum of 0.5 change. We noticed that allowing complete change of input embedding using adaptive α resulted in a high α which reduced final perplexity.

A.3.3 Commitment Weight Analysis

Table 9 shows the impact of commitment cost β on codebook utilization across ERASER and Jigsaw datasets for RoBERTa model.

Commitment Cost (β)	ERASER Perplexity	Jigsaw Perplexity
0.0	213.45	163.45
0.1	210.26	164.07
0.3	189.74	145.65
0.6	170.76	81.38
1.0	23.94	30.71

Table 9: Impact of commitment cost β on validation perplexity across datasets.

Higher β values force stronger commitment to assigned codebook vectors, reducing perplexity but limiting concept diversity. Lower β values allow more flexible assignments, promoting diverse concept identification. β=0.1 achieves optimal balance between concept diversity and training stability.

A.3.4 Temperature and Top-k

Top-k	Temperature	Perturbed CLS
1	1.0	0.0911 ± 0.0416
10	1.0	0.0864 ± 0.0488
100	1.0	0.0817 ± 0.0462
400	1.0	0.0877 ± 0.0355
400	0.1	0.0806 ± 0.0407
400	1.0	0.0877 ± 0.0355
400	2.0	0.0783 ± 0.0378
400	4.0	0.0911 ± 0.0360
Baseline values:		
Original CLS: 0.7604 ± 0.1317		Random Perturbed CLS: 0.7264 ± 0.1152

Table 10: Impact of temperature and top-k values on concept identification performance

Table 10 demonstrates that our perturbation-based faithfulness metric has limited sensitivity to temperature and top-k hyperparameter. Despite significant differences in temperature and top-k values, the perturbed CLS accuracies remain within a narrow range (0.0783-0.0911), suggesting that this evaluation approach, while effective for comparing architectural differences, cannot reliably discriminate between subtle hyperparameter configurations.

A.4 Layer Pair Analysis

We analyze the impact of layer pair selection on concept discovery by evaluating several combinations across RoBERTa models fine-tuned on ERASER-Movie and Jigsaw. Our final choice of layers 8–12 is motivated both by theoretical understanding and empirical evidence.

Theoretical Motivation. Transformer-based architectures such as RoBERTa are known to exhibit hierarchical processing, where lower layers capture surface-level linguistic patterns and intermediate layers encode rich semantic information (Yu et al., 2024). We hypothesize that the transformation between Layers 8 and 12 best captures the transition from semantically meaningful representations to task-specific decision-making features. While our method is layer-agnostic in design, selecting this range enables optimal interpretability.

Empirical Validation. We conducted systematic experiments across multiple layer pairs, comparing the impact of concept removal using our perturbation-based faithfulness metric. Table 11 presents accuracy after perturbing the [CLS] token using discovered concepts, alongside baselines using original and randomly perturbed inputs.

Model–Dataset	Layer Pair	Perturbed CLS	Original CLS	Random Perturbed
RoBERTa–ERASER	0–4	0.5023 ± 0.010	0.5023 ± 0.010	0.5047 ± 0.016
RoBERTa–ERASER	4–8	0.7357 ± 0.073	0.5327 ± 0.063	0.5420 ± 0.065
RoBERTa–ERASER	8–12	0.0782 ± 0.045	0.7604 ± 0.132	0.7264 ± 0.115
RoBERTa–Jigsaw	0–4	0.5013 ± 0.0179	0.5026 ± 0.011	0.5026 ± 0.014
RoBERTa–Jigsaw	4–8	0.1491 ± 0.0539	0.7781 ± 0.076	0.7576 ± 0.072
RoBERTa–Jigsaw	8–12	0.6456 ± 0.1334	0.9134 ± 0.046	0.9134 ± 0.046

Table 11: Layer pair analysis showing that layers 8–12 capture the most meaningful transformations for both datasets.

These results support three key observations. First, early layers (0–4) encode minimal task-relevant concepts, as perturbing them leads to no meaningful change in prediction accuracy. Second, the 4–8 layer range shows inconsistent behavior across datasets—on ERASER, we observe an unexpected accuracy increase following concept perturbation, possibly due to the removal of noisy or redundant features, whereas on Jigsaw, the expected performance drop suggests some level of concept relevance. Finally, the 8–12 configuration consistently reveals meaningful concepts across both datasets: perturbing this range significantly degrades model performance, indicating that it captures the most faithful and impactful transformations from semantic features to final task-specific representations.

A.5 Qualitative Evaluation

To show how CLVQ-VAE represents concepts related to sentiment analysis, we examine examples from the ERASER movie review dataset. We present word clouds generated by our method for different prediction outcomes.

A.5.1 False Negative Example

In Figure 2, we show a false negative example where the model incorrectly predicts negative sentiment for a positive review. The review describes actors performing imitations, with Lloyd Bridges doing a "decent imitation of Brando's godfather" and Pamela Gidley performing a "dead-on mockery of Sharon Stone".

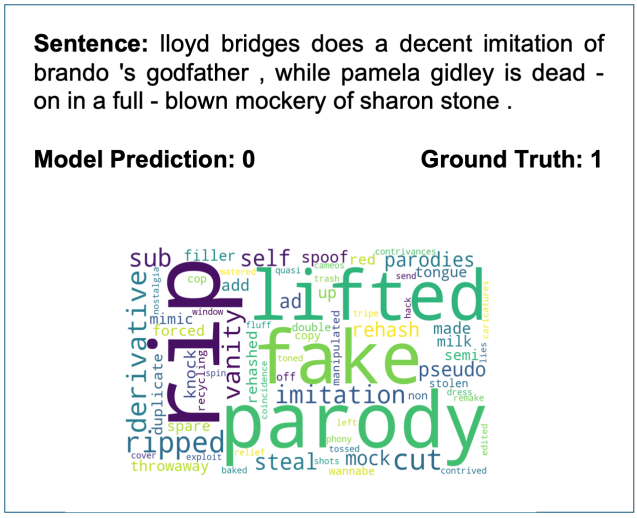

Figure 2: False negative example (Model: 0, Ground Truth: 1) showing concept clusters related to imitation and parody.

The word cloud in Figure 2 contains many terms related to imitation ("lifted", "fake", "parody", "imitation", "ripped"). Despite the review framing these imitations positively as "decent" and "dead-on", the model associates these imitation concepts with negative sentiment. This shows a limitation in distinguishing between criticism of unoriginality and praise for good impersonations.

A.5.2 False Positive Example

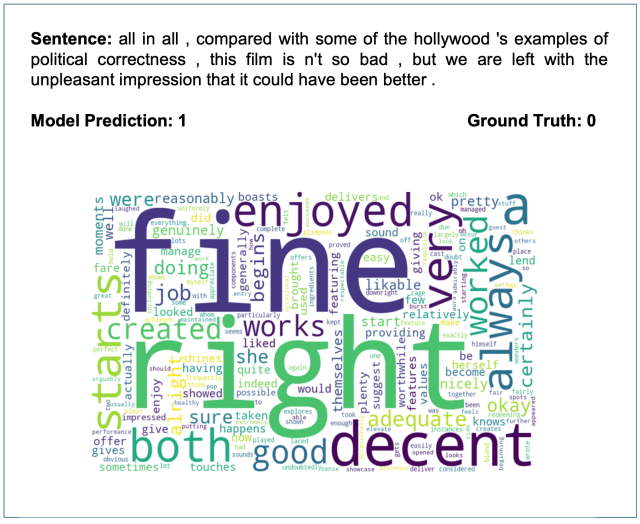

Figure 3: False positive example (Model: 1, Ground Truth: 0) showing terms of moderate approval despite the overall negative sentiment.

Figure 3 shows a false positive case where the model incorrectly predicts positive sentiment for a negative review. The review describes a film that "isn't so bad" but leaves "the unpleasant impression that it could have been better".

The word cloud in Figure 3 shows terms of moderate approval (“fine”, “enjoyed”, “decent”, “right”, “good”). The model focused on the mild praise while missing the more subtle negative sentiment. This shows a limitation in distinguishing between faint praise and genuine positive sentiment in reviews with mixed language.

A.5.3 True Negative Example

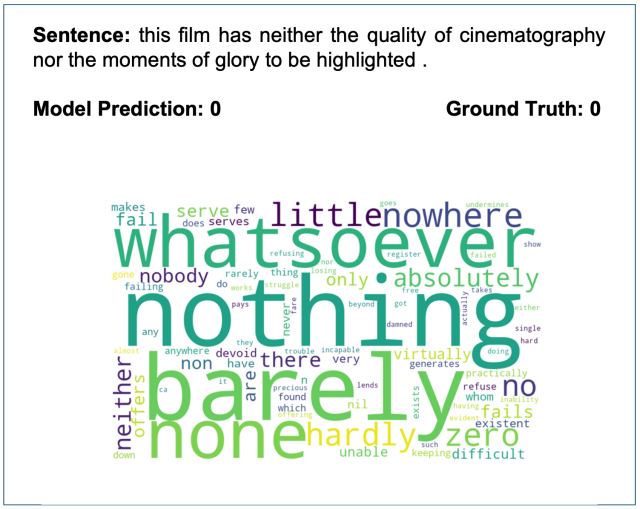

Figure 4: True negative example (Model: 0, Ground Truth: 0) showing terms expressing absence or deficiency.

Figure 4 shows a true negative example where the model correctly predicts negative sentiment. The review states “this film has neither the quality of cinematography nor the moments of glory to be highlighted”.

The word cloud in Figure 4 contains terms expressing absence or lack (“nothing”, “barely”, “whatsoever”, “little”, “nowhere”, “zero”). This shows how CLVQ-VAE effectively captures concepts related to deficiency, correctly identifying negative sentiment in the review.

A.5.4 True Positive Example

In Figure 5, we show a true positive example where the model correctly predicts positive sentiment for “the acting is superb from everyone involved”. This direct praise is an ideal case for sentiment analysis.

The word cloud in Figure 5 shows many positive descriptors (“outstanding”, “fantastic”, “awesome”, “magnificent”). This demonstrates how our model captures related positive terms, particularly for straightforward expressions of praise.

These examples show how CLVQ-VAE captures discrete concepts for sentiment classification, which often includes sentiment-laden terms and their semantic relationships. The false prediction cases (Figures 2 and 3) highlight limitations identified by CLVQ-VAE in RoBERTa model.

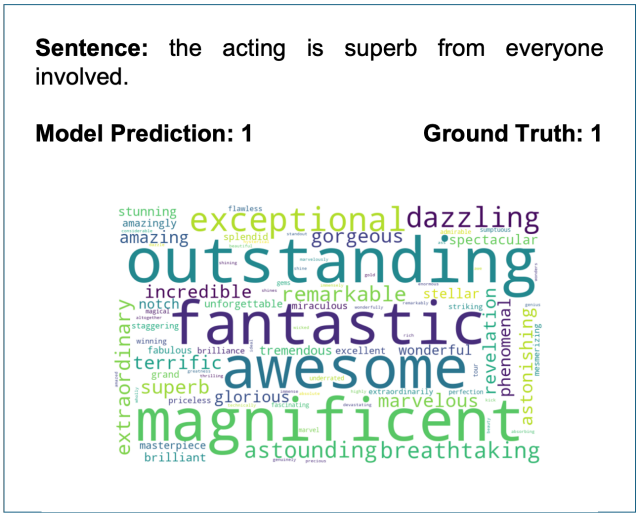

Figure 5: True positive example (Model: 1, Ground Truth: 1) showing positive descriptors for "the acting is superb from everyone involved".