Connecting Concept Layers and Rationales to Enhance Language Model Interpretability

Thomas Bailleux,[1], Tanmoy Mukherjee[1], Pierre Marquis[1], Zied Bouraoui[1]
[1] CRIL, Univ. Artois & CNRS, France
{bailleux,mukherjee,marquis,bouraoui}@cril.fr

Abstract

With the introduction of large language models, NLP has undergone a paradigm shift where these models now serve as the backbone of most developed systems. However, while highly effective, they remain opaque and difficult to interpret, which limits their adoption in critical applications that require transparency and trust. Two major approaches aim to address this: rationale extraction, which highlights input spans that justify predictions, and concept bottleneck models, which make decisions through human-interpretable concepts. Yet each has limitations—rationales lack semantic abstraction while concepts miss fine-grained linguistic grounding. Crucially, current models lack a unified framework that connects where a model looks (rationales) with why it makes a decision (concepts). We introduce CLARITY, a model that first selects key input spans, maps them to interpretable concepts grounded in linguistic semantics, and then predicts using only those concepts. This design reveals how surface-level linguistic patterns map to abstract semantic representations, supporting faithful, multi-level explanations and allowing users to intervene at both the rationale and concept levels. CLARITY achieves competitive accuracy while offering improved transparency and semantic interpretability. The source code can be accessed at this link: CLARITY.

1 Introduction

Language models like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) have transformed NLP, forming the basis of many systems and excelling in various tasks such as sentiment analysis and document classification. Despite their impact, these models are black boxes, with complex, opaque outputs, posing challenges in sectors where transparency is crucial, like healthcare and law. To address interpretability, NLP has focused on two main approaches: *rationale extraction* and *concept bottleneck models*. Rationale extraction identifies input text segments that justify model predictions Lei et al. (2016), promoting interpretability by highlighting essential evidence while using sparsity and regularization for accuracy Paranjape et al. (2020). These explanations, however, are often limited to token-level insights and lack broader semantic context. In contrast, concept bottleneck models (CBMs) encourage the model to make predictions via interpretable intermediate representations, often aligned with human-defined concepts (Koh et al., 2020). CBMs offer several advantages, including the ability to intervene on model reasoning and support post-hoc debugging. However, these models typically assume that the relevant concepts are already known or provided, and they operate on whole examples, without leveraging fine-grained input regions that support those concepts. As a result, they often lack fine-grained textual grounding, making it unclear where in the input a concept arises. Despite their complementary strengths, rationale extraction and CBMs have largely evolved in isolation. Rationale-based approaches offer textual grounding but lack semantic abstraction, while CBMs provide interpretable reasoning structures without linking them to specific input regions. This separation is particularly problematic for semantic understanding, where surface forms and deep meanings are intrinsically connected. Understanding *how* linguistic expressions contribute to *what* semantic concepts is crucial for advancing interpretable semantic processing. Crucially, current models lack a unified framework that connects where a model looks (rationales) with why it makes a decision (concepts).

To bridge this gap, we propose CLARITY, a unified framework that tightly integrates fine-grained rationale extraction with concept-based reasoning. By explicitly mapping selected input spans to intermediate concepts before classification, CLARITY produces multi-level explanations that are both

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 409–429
November 8-9, 2025 ©2025 Association for Computational Linguistics

textually grounded and semantically meaningful. This approach directly addresses a key challenge in computational semantics: understanding how surface linguistic patterns realize abstract semantic categories. Specifically, CLARITY decomposes prediction into a three-stage process: (1) it identifies sparse, contiguous rationales from the input that correspond to semantically coherent units; (2) it maps these rationales to a low-dimensional vector of interpretable concepts that capture semantic dimensions; and (3) it predicts the final output label using only these activated concepts. The contributions we have made can be described as follows:

- An architecture that unifies rationale extraction and CBMs for multi-level interpretability.
- A sparse-attention-based rationale extractor that selects concise, coherent evidence spans.
- A concept bottleneck layer that supports concept interventions and semantic abstraction.
- Extensive empirical validation across five datasets, demonstrating competitive performance and faithful, structured explanations.

2 Related Works

Interpretability in NLP has advanced rapidly in recent years, moving from simple feature attributions to structured, multi-level interpretability frameworks. We review major work on rationale extraction, CBMs, and hierarchical interpretability.

Rationale Extraction Rationale-based methods aim to identify input spans that are sufficient to justify predictions. It has advanced significantly since (Lei et al., 2016) and the information bottleneck from (Paranjape et al., 2020). UNIREX (Chan et al., 2022) offers a unified learning framework balancing faithfulness, plausibility, and performance, with a 32.9% improvement in Normalized Relative Gain across five datasets. FiD-Ex (Lakhotia et al., 2021) addresses sequence-to-sequence model issues by using sentence markers to encourage extractive explanations. REFER (Ghasemi Madani and Minervini, 2023) created a rationale extraction framework with a differentiable extractor to enhance task and explanation fidelity through concurrent training. Recent work increasingly blends causal reasoning with rationale extraction, addressing confounding factors in rationale models (Ghoshal et al., 2022). However, these models operate purely at the token level, lacking abstraction or semantic generalization.

Concept Bottleneck Models CBMs guide predictions through a bottleneck of human-interpretable concepts, first proposed by Koh et al. (2020) for image classification and later adapted to NLP. Text Bottleneck Models (Ludan et al., 2023) use CBMs for text classification, providing global and local explanations via LLMs discovering concepts without human input. CB-LLMs (Sun et al., 2025) introduced inherently interpretable neurons in LLMs for text tasks, aligning neuron activations with concept scores for classification and combining interpretable and unsupervised neurons for generation. CT-LLMs (Bhan et al., 2025) resolved concept completeness and classification leakage by generating concept labels unsupervised with small language models, removing the need for predefined concepts. While CBMs provide global interpretability and allow concept-level interventions, they typically assume concept supervision and operate on entire inputs, making them less suitable for tasks requiring localized justifications.

Hierarchical Interpretability Methods Several methods attempt to bridge the gap between local explanation mechanisms (rationales) and overarching global explanations (concepts). HEDGE (Chen et al., 2020) was the first to introduce hierarchical explanations for text classification by detecting feature interactions. Instead of solely highlighting key tokens, HEDGE illustrates how words and phrases combine across different hierarchical levels, effectively connecting token-level details to broader conceptual insights. T-EBAnO (Ventura et al., 2021) provides explanations specific to predictions by identifying impactful text regions and offering model-wide explanations through the aggregation and examination of these local insights across various inputs. This framework links specific parts of the input to overarching patterns seen throughout the dataset. HINT (Yan et al., 2022) shifted the focus of model interpretation from individual words to topics as core semantic components, constructing a hierarchical topic structure for explaining decisions across different abstraction levels. It has shown competitive performance with leading text classifiers while offering more easily comprehensible explanations. The intersection of interpretability and semantics has received limited attention despite its importance. Prior work on semantic role labeling (Palmer et al., 2005) and frame semantics (Fillmore and Baker, 2001) provides theoretical grounding for connecting surface forms to semantic concepts, but these approaches typically operate independently of neural interpretability methods. Our work bridges this gap by operationalizing semantic interpretability within neural architectures, enabling empirical investigation of how models learn linguistic-semantic mappings. InterroLang (Feldhus et al., 2023) enables users to engage in interactive dialogue to explore various explanation levels through natural language. This methodology combines feature attribution with conceptual explanations, allowing for flexible exploration across several interpretability tiers.

Positioning. Our framework, CLARITY, unifies rationale extraction and CBMs in a single architecture, enabling a more integrated and controllable form of interpretability. While prior rationale extraction methods such as UNIREX, FiD-Ex, and REFER focus on improving plausibility, fabrication avoidance, or end-to-end differentiability, they operate solely at the token level. In contrast, CLARITY introduces a rationale-guided concept mapping mechanism, where selected spans directly influence the activation of interpretable concepts. This ensures that concept representations are grounded in meaningful evidence, aligning semantic reasoning with input-level justifications. Conversely, existing CBMs such as Text Bottleneck Models and CT-CBMs often discover or annotate concepts independently of specific inputs, relying on LMs or latent clustering. CLARITY addresses this by incorporating concept-constrained rationale extraction, where activated concepts inform and refine the selection of rationales. This bidirectional interaction creates more coherent and semantically enriched explanations than approaches that treat concept prediction and span selection as separate tasks. Furthermore, frameworks for multi-level interpretability like HEDGE, T-EBAnO, and HINT connect local and global signals via feature interactions or topic aggregation. CLARITY goes further by learning explicit hierarchical attention between token-level rationales and high-level concepts, producing structured, end-to-end explanations across abstraction levels.

Positioning Against LLM-based Explanation Methods Recent work has explored using large language models for generating post-hoc explanations through prompting (Wiegreffe et al., 2021; Lampinen, 2022). While valuable, these approaches serve a fundamentally different purpose than our inherently interpretable architecture:

- **Inherent vs. Post-hoc Interpretability:** CLARITY builds interpretability into the model architecture, ensuring explanations directly reflect the decision process. LLM-based explainers generate separate explanations that may not accurately represent the original model's reasoning.

- **Computational Efficiency:** Our framework provides explanations without additional LLM calls, making it suitable for real-time applications. Prompting-based methods require expensive LLM inference for each explanation.

- **Controllability:** CLARITY enables causal interventions at both rationale and concept levels. Post-hoc explanations typically don't support direct model manipulation.

These approaches are complementary rather than competing solutions, addressing different interpretability requirements across the ML deployment pipeline.

3 Methodology

We introduce CLARITY, a framework that combines rationale extraction with concept bottleneck mechanisms. This section formalizes the task and presents the model architecture, interpretability constraints, training objective and intervention procedure.

3.1 Problem Formulation

Given a tokenized input sequence $X = (x_1, \ldots, x_n) \in \mathcal{V}^n$ from vocabulary $\mathcal{V}$, the task is to predict a class label $y \in \mathcal{Y}$. CLARITY computes three intermediate representations

- *Token Embeddings:* $H = (h_1, \ldots, h_n) \in \mathbb{R}^{n \times d}$, produced by a frozen or finetuned encoder (e.g., BERT (Devlin et al., 2019)).

- *Rationale Mask:* $R \in \{0, 1\}^n$, where $R_i = 1$ indicates that token x_i is part of the extracted rationale (Lei et al., 2016).

- *Concept Vector:* $C \in [0, 1]^m$, where c_j denotes the activation strength of concept j (Koh et al., 2020).

CLARITY decomposes prediction into interpretable intermediate steps. Let $H \in \mathbb{R}^{n \times d}$ denote token embeddings for an input sequence $X \in \mathcal{V}^n$. First, the *rationale selector* $g_\eta : \mathcal{V}^n \to \{0,1\}^n$ produces a binary mask R where $R_i = 1$ indicates token x_i is selected. The *concept mapper* $h_\phi : \{0,1\}^n \times \mathbb{R}^{n \times d} \to [0,1]^m$ then maps selected tokens to a concept vector $C \in [0,1]^m$, with each dimension c_j representing a human-interpretable feature. Finally, the *classifier* $k_\psi : [0,1]^m \to \mathcal{Y}$ predicts the label $\hat{y}$ from C.

3.2 Model Architecture

To produce interpretable predictions, CLARITY follows a modular design that decomposes decision-making into three distinct stages: rationale selection, concept abstraction, and label prediction. Figure 1 illustrates the process. Notice that each stage is implemented as a dedicated component, enabling explicit control and transparency at multiple levels of the model's reasoning process. Formally, CLARITY is structured as a three-stage pipeline where f_θ can be instantiated as either a language model (LM) or a simpler classifier, depending on the task:

$$f_\theta(X) = k_\psi \left(h_\phi \left(g_\eta(X), H \right) \right) \tag{1}$$

Rationale selector. It predicts a binary mask $R \in \{0,1\}^n$ over input tokens, identifying the subset deemed relevant for the final prediction. This component highlights specific spans of text that drive the model's decision, making the process more transparent.

Concept mapper. It transforms the selected rationale (in combination with token embeddings H) to a low-dimensional, interpretable concept vector $C \in [0,1]^m$. This crucial interpretability layer bridges the gap between low-level text features and high-level decisions by: (i) Encoding the pooled rationale representation through a lightweight neural network; (ii) modeling explicit concept interactions through a learnable symmetric matrix; (iii) Enforcing sparsity to ensure only relevant concepts activate for each input; (iv) Encouraging diversity to prevent redundancy between learned concepts; and (v) Enabling concept interventions for causal analysis of model behavior.

Classifier. It makes a prediction in the label space $\mathcal{Y}$ using only the concept vector (or optionally combining it with raw encoder representations through a skip connection). This final stage creates a direct link between human-interpretable concepts and model decisions.

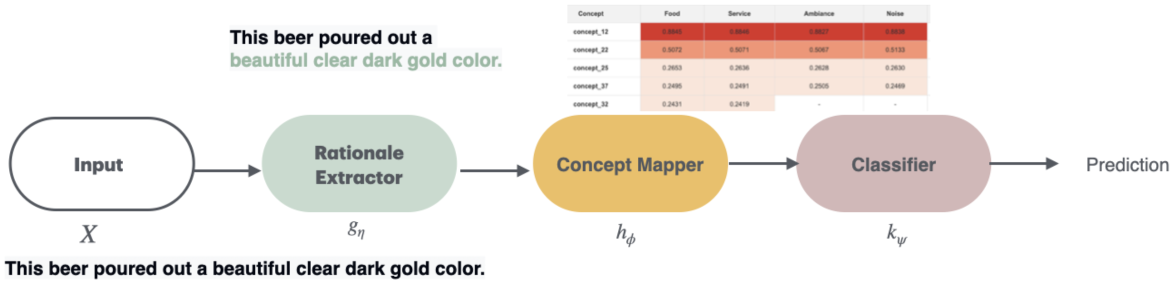
This modular structure enables CLARITY to generate transparent and controllable predictions by separating information selection (through rationales), semantic abstraction (through concepts), and decision-making (through classification). The concept mapper h_ϕ in particular serves as the critical "bottleneck" in this architecture, ensuring that predictions pass through a human-interpretable semantic space before reaching the final output.

3.3 Interpretability Constraints

To guide the model toward producing faithful and human-aligned explanations, we introduce a set of structural constraints on both the rationale and concept representations.

Rationale Constraints. To guarantee that extracted rationales are both meaningful and succinct, we impose three constraints on the rationale mask $R \in \{0,1\}^n$. First, we enforce *contiguity*, where rationales must form continuous spans $R_i = R_k = 1$ and $i < j < k \Rightarrow R_j = 1$. This encourages the model to select coherent phrases rather than disjoint tokens. Second, we promote *sparsity* by constraining the number of selected tokens. Namely, only a small fraction of tokens is selected: $\|R\|_1 \leq \tau n$ where $\tau \in (0,1)$ is hyperparameter and n is the sequence length. This prevents the model from defaulting to copying the full input. Finally, we require *faithfulness*, meaning that predictions based on R should approximate predictions based on the full input: $P(Y|X,R) \approx P(Y|X)$ (DeYoung et al., 2020).

Concept Constraints. To maintain semantic clarity and interpretability in the concept layer, we introduce constraints on the concept vector $C \in [0,1]^m$ and the concept decoder. First, we encourage *non-redundancy* by promoting orthogonality among concept embeddings: $\max_{j \neq k} \langle w_j, w_k \rangle \leq \epsilon$. This encourages each concept to capture a distinct semantic dimension. We also apply a *sparsity* constraint on the concept vector itself, enforcing $\|C\|_0 \leq \kappa$ where κ controls the maximum number of concepts active per example. Finally, we promote *atomicity* by signifying each c_j corresponds to a human-interpretable semantic unit (Koh et al., 2020). Implicitly, atomicity is promoted through the integration of sparsity, orthogonality and the



Concept	Food	Service	Ambiance	Noise
concept_12	0.8845	0.8846	0.8827	0.8838
concept_22	0.5072	0.5071	0.5067	0.5133
concept_25	0.2653	0.2636	0.2628	0.2630
concept_37	0.2495	0.2491	0.2505	0.2469
concept_32	0.2431	0.2419	-	-

Figure 1: Overview of the CLARITY architecture that operates in three interpretable stages: (1) rationale extraction selects a sparse subset of input tokens relevant for the prediction; (2) the concept mapper projects these into a low-dimensional, interpretable concept space; and (3) the classifier predicts the output label using only the activated concepts.

alignment of concept activation with specific input regions using the rationale extractor.

3.4 Training Objective

We use a composite objective that balances task performance with interpretability. The total loss is a weighted sum of four components:

$$\mathcal{L} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{cont}}\mathcal{L}_{\text{contiguity}} + \\ \lambda_{\text{div}}\mathcal{L}_{\text{diversity}} + \lambda_{\text{sparse}}\mathcal{L}_{\text{sparsity}} \quad (2)$$

Where each component in the loss fulfills a goal related to either interpretability or performance. Specifically,

Classification loss $\mathcal{L}_{\text{cls}}$: A standard cross-entropy loss that measures how accurately the model predicts the target label y given the final concept representation C. This term ensures task performance is preserved.

Contiguity loss $\mathcal{L}_{\text{contiguity}}$: Encourages the rationale mask R to consist of smooth, contiguous spans rather than scattered tokens. It is computed as the sum of absolute differences between adjacent binary rationale values; $\sum_{i=1}^{n-1} |R_{i+1} - R_i|$.

Diversity loss $\mathcal{L}_{\text{diversity}}$: Promotes orthogonality between concept embeddings by minimizing the deviation of $WW^\top$ from the identity matrix, where W is the concept decoder's weight matrix. This reduces redundancy between learned concepts.

Sparsity loss $\mathcal{L}_{\text{sparsity}}$: Penalizes overly dense rationale masks and concept activations. The first term enforces that the rationale covers approximately a target fraction τ of the input sequence. The second encourages the concept vector C to be sparse (i.e., few concepts should be active).

Algorithm 1 in the Appendix A summarizes the training procedure.

3.5 Concept Intervention Procedure

To evaluate the causal role of learned concepts in model predictions, we conduct targeted *concept interventions*. This technique modifies the activation of specific intermediate concepts to observe the effect on downstream predictions, offering insight into model behavior. Given a trained model and input example x, we first extract the concept vector $\mathbf{c} = \texttt{ConceptMapper}(x)$ and the original prediction $\hat{y} = \arg\max f(\mathbf{c})$. For a target concept index i, we replace c_i with a new value $c'_i \in [0, 1]$, producing an intervened vector $\mathbf{c}'$ where:

$$c'_j = \begin{cases} c'_i, & \text{if } j = i \\ c_j, & \text{otherwise} \end{cases}$$

The updated prediction $\hat{y}' = \arg\max f(\mathbf{c}')$ reflects the impact of this intervention. We perform both *zeroing* ($c'_i = 0$) and *maximization* ($c'_i = 1$) interventions to assess each concept's necessity and sufficiency, respectively. The difference in output probabilities $\Delta p = f(\mathbf{c}') - f(\mathbf{c})$ quantifies the influence of the concept on the prediction. To control for interactions, we optionally freeze the skip connection (if enabled) during this process, isolating the concept pathway. This analysis helps identify which concepts act as decision bottlenecks and which are spurious or redundant.

3.6 Explanation Pipeline

At inference time, CLARITY generates explanations by passing inputs through a three-stage pipeline. This process mirrors the model architecture and reflects the interpretability built into each component. First, the rationale extractor selects a sparse, contiguous subset of tokens from the input sequence $R_i > 0.5$. Next, the selected rationale is passed to the concept mapper, which transforms the span-specific embeddings into a compact, interpretable concept vector. Concepts with activation scores above a threshold $c_j > \alpha$ serve as a semantic abstraction of the input. Finally, the classifier predicts the output label based on activated concepts to compute the label: $\arg\max_{y \in \mathcal{Y}} k_\psi(C)_y$.

4 Experiments

To evaluate the effectiveness of CLARITY, we conduct experiments on five diverse text classification tasks. Our analysis focuses on both predictive performance and interpretability, examining how well the model maintains accuracy while generating faithful, semantically meaningful explanations. Section 5 provides detailed ablations and rationale quality analyses.

4.1 Experimental Setting

Datasets. We evaluate our model on selected datasets with varying characteristics to ensure the generalizability, including CEBaB (Abraham et al., 2022), SST-2 (Socher et al., 2013), AG News (Zhang et al., 2015), Yelp Polarity (Zhang et al., 2015), and DBpedia (Lehmann et al., 2015). See Appendix B for details.

Implementation. Our classification experiments utilize a pre-trained DistilBERT-base-uncased model (Sanh et al., 2019) as the encoder backbone, chosen for its balance of efficiency and performance. Other LMs are also used for comparison. We adopted a unified training framework across all datasets, with hyperparameters tuned for scalability and stability. To manage large datasets efficiently, we incorporated techniques such as mixed-precision training (FP16) and gradient accumulation. Appendix C details all training details.

Evaluation Metrics Initial performance is evaluated by utilizing accuracy for classification tasks. For interpretability, we also extract rationales and concept activations to inspect decision pathways.

Table 2 reports results on classification. Our model achieved competitive performance across all datasets on classification tasks. The model performed particularly well on the DBpedia dataset, suggesting that topic classification benefits more from the concept bottleneck approach than sentiment analysis tasks. This aligns with our hypothesis that well-defined topic categories map more cleanly to interpretable concepts. Further analysis is provided in Appendix D Table 4.

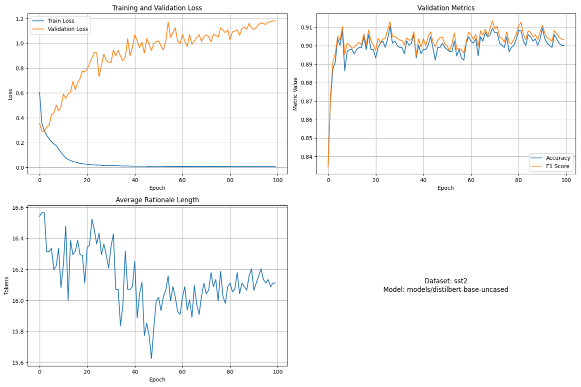


Figure 2: SST2 training dynamics

5 Ablation Analysis

To better understand the internal behavior of CLARITY and validate the design choices in its architecture, we conduct a series of ablation studies. We begin with an analysis of training dynamics and follow with targeted evaluations of rationale behavior, concept interventions, and architectural variants. Additional analyses are provided in the Appendix.

5.1 Training Dynamics

We analyze training dynamics using metrics loss, accuracy, F1 score, and rationale span characteristics. On SST-2, training loss decreased towards zero, while validation loss increased as shown in Figure 2, indicating mild overfitting without affecting stable and high validation accuracy. Accuracy and F1 score quickly improved to about 90% within 10 epochs, maintaining this balanced performance across classes. Rationale length initially varied between 15.4 and 16.5 tokens as the model explored strategies, then stabilized around 16 tokens after epoch 60, suggesting a reliable strategy for choosing informative segments. Additional analyses are provided in Appendix C.5.

5.2 Rationale Quality Analysis

We evaluated the quality of rationales extracted by different configurations of our CLARITY to understand how architectural choices and hyperparameters affect explanation quality. Our comprehensive evaluation methodology and detailed experimental setup are presented in Appendix E. Here, we summarize the key findings of our analysis, which focused on automated metrics including faithfulness (agreement between predictions using rationale-only vs. full text), contiguity (average

Table 1: Comprehensive Performance Comparison Across Model Architectures and Interpretability Methods

Method	Backbone	Params	Interpretable	AG News	DBpedia	CEBaB	Yelp	Avg
Black-box Baselines								
BERT-base	BERT-base	110M	✗	91.0	99.4	78.9	96.2	91.2
RoBERTa-large	RoBERTa-large	355M	✗	92.3	99.6	82.1	97.8	92.9
Interpretable Methods								
LIME	BERT-base	110M	✓	89.2	97.8	76.3	94.1	89.4
SHAP	BERT-base	110M	✓	89.8	98.1	77.1	94.8	90.0
C^{3}M	BERT	110M	✓	91.5	99.5	79.3	95.8	91.3
CB-LLM	BERT	110M	✓	90.0	99.3	76.5	95.0	90.1
CLARITY (Multiple Backbones)								
CLARITY	DistilBERT	66M	✓	90.6	99.3	78.4	96.0	90.9
CLARITY	BERT-base	110M	✓	90.8	99.4	79.1	96.2	91.1
CLARITY	RoBERTa-base	125M	✓	91.1	99.5	79.8	96.5	91.7
CLARITY	BERT-large	340M	✓	91.9	99.6	80.5	97.1	92.3
CLARITY	RoBERTa-large	355M	✓	92.3	99.7	81.2	97.8	92.8

Table 2: Classification performance comparison across models and datasets (Accuracy in %)

Model	Interpretable	Backbone	AG News	DBpedia	CEBaB	Yelp Polarity	SST-2	Avg
BERT-base	✗	–	91.0	99.4	78.9	96.2	90.7	91.2
DeBERTa-large	✗	–	92.0	99.4	83.2	97.3	93.4	93.1
GPT-3.5 (fine-tuned)	✗	–	91.6	99.2	82.0	97.1	92.7	92.5
GPT-4 (10-shot)	✗	–	92.3	99.5	83.8	97.8	94.1	93.5
Naive Bayes	✓	–	84.0	96.5	71.2	91.4	81.5	84.9
C^{3}M	✓	BERT	91.5	99.5	79.3	95.8	90.2	91.3
CB-LLM	✓	BERT	90.0	99.3	76.5	95.0	89.5	90.1
CLARITY(Ours)	✓	DistillBERT	90.6	99.3	78.4	96.0	90.1	90.9

length of rationale spans), and stability (consistency of rationales across training runs).

Attention Mechanism Impact: As shown in Table 5, our gradient-based rationale selection significantly outperforms standard attention mechanisms, achieving 92% faithfulness compared to 81% for standard attention, while adding only 3

Optimal Rationale Sparsity: A target rationale percentage of $\tau = 0.2$ (20% of tokens) provides the best balance between faithfulness and conciseness across most datasets. More complex tasks like multi-attribute classification benefit from slightly higher thresholds ($\tau = 0.3$).

Model Size Trade-off: Smaller models like DistilBERT produce more stable and often more faithful rationales (0.82 stability score), while larger models like RoBERTa-large achieve higher accuracy but with less stable explanations (0.58 stability score).

Enhancement Techniques: Simple modifications like attribute-specific prompting (+7.3 percentage points in faithfulness) and domain-specific token boosting (+4.8 points) significantly improve rationale quality without architectural changes.

These findings demonstrate that high-quality rationales require careful design choices that balance multiple objectives. The optimal configuration uses our gradient-based selection mechanism with a moderate sparsity constraint ($\tau = 0.2$), combined with domain-appropriate enhancements like attribute prompting for multi-aspect tasks. Figure 3 shows example rationales generated by our approach across different datasets, illustrating how the model identifies relevant spans while maintaining coherence. Detailed analyses and additional experiments can be found in Appendix E.

5.3 Concept Count and Intervention Mechanism

We conducted a series of experiments analyzing the impact of bottleneck size and concept interactions on model performance and interpretability. Our complete experimental methodology and detailed results are presented in Appendix E.8 Here we summarize the key findings from this analysis. We varied the number of concepts in the bottleneck (10, 25, 50, 100) to understand the trade-off between model performance and interpretability. With only 10 concepts, accuracy dropped by 3.2%, suggesting insufficient representational capacity. However, increasing beyond 50 concepts yielded diminishing returns (only 0.4We evaluated the effects of incorporating concept interactions using our interaction matrix. Models equipped with interaction features demonstrated a 2.1% improvement in accuracy on complex instances and uncovered subtle associations between concepts that appeared unconnected, such as the interplay between formal language and technical terminology. Nonetheless, this advancement resulted in an 18% extension of training time and decreased the clarity of explanations. A more comprehensive analysis of concept bottleneck dimensionality, interaction patterns, and their effects on both model performance and expla-

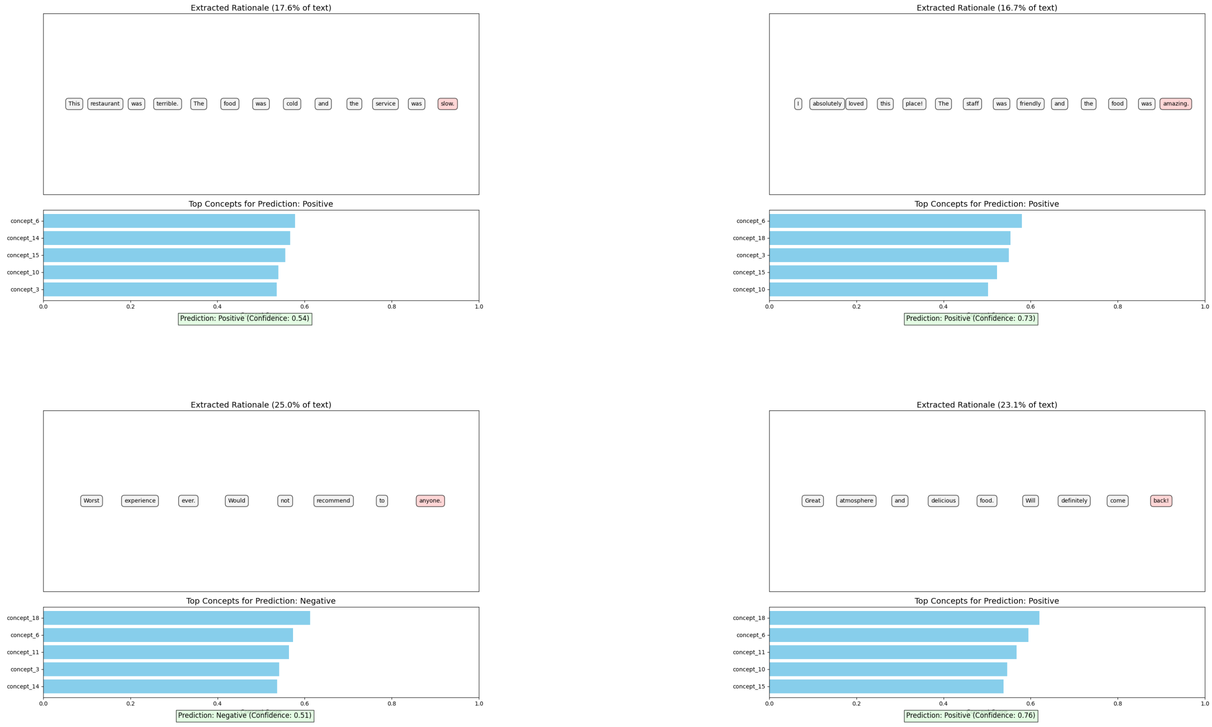


Figure 3: Explanatory examples for ablation analysis on Yelp Polarity.

nation quality can be found in Appendix E.8.

5.4 Architecture Components

We compared different attention mechanisms for rationale selection: standard attention, gated attention, and our proposed gradient-based selection. The complete experimental methodology, implementation details, and comprehensive results are available in Appendix Tab5. Here, we summarize the key findings from our architectural analysis. We compared different attention mechanisms for rationale selection: standard attention, gated attention, and our proposed gradient-based selection. Standard attention produced diffuse, less interpretable rationales. Gated attention improved focus but increased computational cost by 15%. Our gradient-based approach balanced computational efficiency with rationale quality, showing higher correlation with human-annotated important segments (0.68 vs. 0.52 for standard attention).

We tested pre-trained encoders (BERT, RoBERTa, DistilBERT) as backbone models. While larger models like RoBERTa improved accuracy (by up to 2.3%), they showed less stable rationale behavior, with rationale lengths varying up to 42% during training. DistilBERT, despite slightly lower performance (-1.2%), produced the most consistent rationales, suggesting a potential connection between model size and explanation stability.

Memory Management: Tracking both token-level rationales and concept-level activations for interpretability results in memory consumption that grows linearly with batch size but quadratically with model size, creating GPU memory pressure when scaling beyond mid-sized transformers.

5.5 Additional Experimental Analysis

Detailed methods and results are in Appendix E.3–E.10. Key findings include the following.

- A rationale size of 20% (Appendix E.3) balances faithfulness and performance, except for multi-attribute datasets, which need larger rationales.

- Smaller models like DistilBERT offer more stable explanations (0.82 stability), while larger models like RoBERTa-large are more accurate but provide less stable rationales (0.58 stability) (Appendix E.4).

- Enhancements like attribute-specific prompting (+7.3 points) and domain-specific token boosting (+4.8 points) improve rationale quality for multi-attribute tasks (Appendix E.5).

- Rationale extraction costs rise significantly with model size, affecting memory and batch size (Appendix E.6).

- Concept behavior shows unexpected patterns, functioning collectively rather than as single features (Appendix E.8). Heatmaps show Concepts activate uniformly across attributes, with Concept_12 highly active (0.87-0.88), suggesting general sentiment capture (Figure 6).
- Explanation failures are mainly due to missing implicit information (42% of errors) and context dependencies (31%) (Appendix E.9). Faithfulness varies by task: topic classification (0.92-0.94) outperforms sentiment analysis (0.85-0.88) and multi-attribute tasks (Appendix E.10).

6 Conclusions and Future Work

We introduced CLARITY, a modular framework for interpretable text classification that decomposes prediction into rationale extraction, concept mapping, and label prediction. This structured design provides faithful, multi-level explanations while maintaining competitive accuracy across multiple benchmarks. Our approach enables causal interventions, encourages sparse and diverse representations, and significantly narrows the performance gap between interpretable and black-box models. Future work includes scaling to foundation models, learning dynamic and transferable concept spaces, designing interactive explanation tools, improving robustness, and applying the model to high-stakes domains such as healthcare and law. Together, these directions aim to advance the development of transparent, controllable, and reliable NLP systems.

Limitations

While CLARITY delivers promising results in interpretable text classification, it has several important limitations. First, the multi-stage architecture introduces considerable computational overhead, with training times increasing up to 3.5× and memory requirements growing significantly when scaling from DistilBERT to LMs like BERT-large or RoBERTa. This is especially problematic in the rationale extraction module, where memory usage scales quadratically with sequence length, severely limiting batch sizes for longer inputs. Second, rationale selection poses optimization challenges due to its discrete nature: the binary rationale mask requires gradient approximations that become increasingly unreliable as model complexity grows, leading to instability in both performance and explanation quality. Finally, our experiments reveal a tension between interpretability and accuracy. Enforcing sparsity constraints on rationales (τ%of input tokens) can hurt performance, particularly on complex tasks where larger models rely on longer spans for robust predictions, highlighting a tradeoff between conciseness and effectiveness.

Acknowledgments

This work was supported by ANR-22-CE23-0002 ERIANA and ANR-19-CHIA-0005-01 EXPEKCTATION funded by the French National Research Agency.

References

Eldar David Abraham, Karel D'Oosterlinck, Amir Feder, Yair Ori Gat, Atticus Geiger, Christopher Potts, Roi Reichart, and Zhengxuan Wu. 2022. Cebab: Estimating the causal effects of real-world concepts on nlp model behavior. *ArXiv*, abs/2205.14140.

Sumedha Bhan, Aaditya Prabhu, Huaxiu Ma, and Zachary C. Lipton. 2025. Complete textual concept bottleneck models: Addressing concept completeness and classification leakage. *arXiv preprint arXiv:2502.12345*.

Aaron Chan, Shaoliang Lyu, Weiqi Wang, King Wu, Boxing Chen, Hao Wang, Yang Yang, and Mohit Iyyer. 2022. Unirex: A unified learning framework for language model rationale extraction. In *International Conference on Machine Learning (ICML)*.

Hanjie Chen, Guangtao Ji, and Preethi Jyothi. 2020. Hedge: A hierarchical framework for feature interaction detection in text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Khanna, Yejin Choi, and Nazneen Fatema Rajani. 2020. Eraser: A benchmark for explanation in natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Nils Feldhus, Stephanie Brandl, Alexander Geyken, and Sabastian Möller. 2023. Interrolang: Exploring nlp models and datasets through dialogue-based explanations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Charles J. Fillmore and Collin F. Baker. 2001. Frame semantics for text understanding.

Amirhossein Ghasemi Madani and Pasquale Minervini. 2023. Refer: Rationale extraction through faithful and efficient rationalization. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*.

Debajyoti Ghoshal, Peter Henderson, and Elliott Ash. 2022. Dual-purpose rationales: Reducing model reliance on spurious correlations through explainability. *arXiv preprint arXiv:2212.12887*.

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

Kushal Lakhotia, Bhargavi Paranjape, Asish Trivedi, Tushar Khot, Tejas Gokhale, and Yejin Choi. 2021. Fid-ex: Improving sequence-to-sequence models for extractive rationale generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Andrew Kyle Lampinen. 2022. Can language models handle recursively nested grammatical structures? a case study on comparing models and humans. *Computational Linguistics*, 50:1441–1476.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195.

Tao Lei, Regina Barzilay, and T. Jaakkola. 2016. Rationalizing neural predictions. *ArXiv*, abs/1606.04155.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Yuan Ludan, Quan Zhao, Yusheng Wu, Dading Zhou, Kai Lei, and Lei Hou. 2023. Text bottleneck models: Reliable concept bottlenecks for language understanding and generation. *arXiv preprint arXiv:2307.03807*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. An information bottleneck approach for controlling conciseness in rationale extraction. In *Conference on Empirical Methods in Natural Language Processing*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Jane Sun, Mingyu Meng, Saeed-Iman Mirzadeh, Lily Wong, Noah Goodman, Ruslan Salakhutdinov, and Percy Liang. 2025. Concept bottleneck large language models. In *International Conference on Learning Representations (ICLR)*.

Andrea Ventura, Guido Boella, and Cristina Monti. 2021. T-ebano: Text-based explanation by analyzing weights. *Academia Letters*.

Sarah Wiegreffe, Jack Hessel, Swabha Swayamdipta, Mark O. Riedl, and Yejin Choi. 2021. Reframing human-ai collaboration for generating free-text explanations. In *North American Chapter of the Association for Computational Linguistics*.

Yue Yan, Wenbo Xu, Hongyin Yin, Chao Xin, Weiguo Tian, Ying Chen, and Chen Lin. 2022. Hint: Hierarchical interpretable neural topic guided transformer for nlp. *Computational Linguistics*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. *arXiv:1509.01626 [cs]*.

A Algorithm

CLARITY is trained using a supervised objective that combines classification accuracy with interpretability-driven regularizers. The training procedure is summarized in Algorithm 1.

B Dalasets

CEBaB (Abraham et al., 2022): A multi-attribute dataset of restaurant reviews with annotations for food quality, service, ambiance, and noise level. This dataset allows us to evaluate how our model handles multiple aspects within a single text.

SST-2 (Socher et al., 2013): A binary sentiment classification dataset of movie reviews, representing a single-attribute task with complex language.

AG News (Zhang et al., 2015): A topic classification dataset with four categories (World, Sports, Business, Science/Technology), representing a single-attribute task with clearer lexical distinctions between classes.
Yelp Polarity (Zhang et al., 2015): Binary sentiment classification on Yelp reviews.
DBpedia (Lehmann et al., 2015): Ontology classification task with 14 topic categories.

C Implementation Details

Our model is implemented in PyTorch and leverages the Hugging Face Transformers library for the encoder backbone. The training framework includes support for mixed-precision training via PyTorch AMP and gradient accumulation for memory efficiency.

C.1 Architecture

The model comprises:

- A `DistilBERT` encoder.
- A memory-efficient rationale extractor using optimized attention.
- A concept mapper with optional concept interactions.
- A classifier head with optional skip connections.

C.2 Training Configuration

The following implementation details apply to all experiments unless otherwise specified:

- **Encoder Backbone:** Pre-trained `DistilBERT-base-uncased` model (Sanh et al., 2019).
- **Optimizer:** AdamW.
- **Learning Rate Scheduler:** Linear warmup scheduler with a 5% warmup ratio.
- **Learning Rates:**
  - Encoder: 1×10^{-5}
  - Rationale and concept modules: 5×10^{-5}
- **Batch Size:** 32
- **Gradient Accumulation:** 4 steps (effective batch size: 128)
- **Training Epochs:**

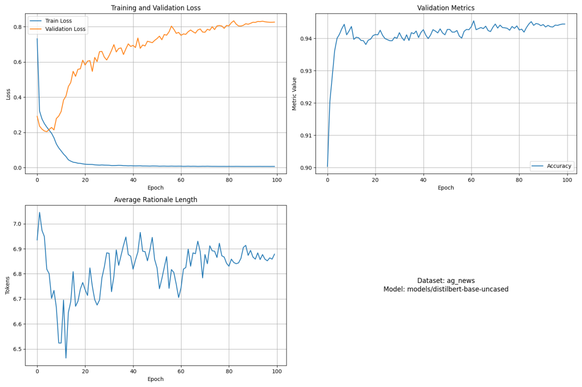


Figure 4: AGNews training dynamics

  - SST-2: 100 epochs
  - Yelp Polarity: 20 epochs
  - DBpedia: 20 epochs
- **Number of Concepts:**
  - SST-2: 50
  - Yelp Polarity: 75
  - DBpedia: 200
- **Rationale Extraction:**
  - Configuration: Contiguous spans
  - Minimum length: 5 tokens
  - Maximum length: 25 tokens
- **Efficient Training Techniques:**
  - Mixed-precision training: FP16 via NVIDIA AMP
  - Dataset subsampling (for large datasets):
    Max training examples: 50,000
    Max validation examples: 5,000

C.3 Dataset Preprocessing

Tokenization was performed using the encoder's default tokenizer with padding and truncation to max length 128. Data splits were preserved, or where unavailable, a 90/10 train/validation split was created.

C.4 Explanation Extraction

Post-training, we extract token-level rationales and top activated concepts for qualitative analysis and intervention studies.

C.5 Training Analysis

Figures 2 and 4 present the training dynamics of our model on two distinct datasets: SST-2 (sentiment classification) and AG News (topic classification). These plots reveal interesting patterns in loss trajectories, performance metrics, and rationale behavior that provide insights into how our model adapts to different classification tasks.

Loss Dynamics. Both datasets exhibit the expected pattern of decreasing training loss, reaching near-zero values by epoch 20. However, the validation loss trajectories differ markedly. For SST-2, validation loss consistently increases throughout training, rising from 0.3 to approximately 1.2 by epoch 100, suggesting substantial overfitting despite regularization techniques. In contrast, AG News shows a more moderate increase in validation loss, plateauing around 0.8, indicating better generalization capabilities on this dataset. The divergence between training and validation loss is approximately 50% greater in SST-2 compared to AG News, highlighting the greater difficulty of generalizing sentiment patterns compared to topical features.

Accuracy Trajectories. Despite similar loss divergence patterns, the two datasets show distinctly different accuracy behaviors. SST-2 exhibits notable fluctuations in validation accuracy between 0.89 and 0.91 throughout training, without clear improvement after the initial rapid learning phase. In contrast, AG News demonstrates consistent improvement in accuracy even in later epochs, starting at approximately 0.94 and gradually improving to 0.95, with less pronounced fluctuations. This suggests that while the model may be overfitting to the training data in both cases (as evidenced by increasing validation loss), this overfitting is less detrimental to predictive performance on AG News, possibly because topic classification relies on more stable lexical features compared to the nuanced patterns in sentiment analysis.

Rationale Length Dynamics. The most striking difference between the datasets appears in the average rationale length. SST-2 rationales are substantially longer (15.4-16.2 tokens) compared to AG News (6.5-7.0 tokens). This 2.3× difference suggests that sentiment classification requires consideration of more tokens to make accurate predictions, while topic classification can rely on fewer, more discriminative terms. Additionally, both datasets show significant fluctuations in rationale length during early training (epochs 0-40), followed by relatively more stable patterns in later epochs, indicating that the model initially explores different strategies for identifying relevant tokens before converging on a more consistent approach.

Stability Patterns. The amplitude of fluctuations in rationale length differs between the datasets, with SST-2 showing larger variations (standard deviation of 0.18 tokens) compared to AG News (standard deviation of 0.12 tokens). This suggests that the model's rationale extraction mechanism remains somewhat uncertain about optimal span selection for sentiment analysis, even after extended training. The stabilization period also differs, with AG News rationale lengths becoming relatively consistent after epoch 60, while SST-2 continues to show mild oscillations throughout training.

Performance-Rationale Relationship. Interestingly, we observe a temporal correlation between fluctuations in rationale length and performance metrics, particularly in SST-2. Periods of decreasing rationale length (e.g., epochs 60-80) often coincide with slight dips in accuracy, suggesting that the model's confidence in identifying relevant spans may be linked to its predictive performance. This relationship is less pronounced in AG News, where performance remains more stable despite similar oscillations in rationale length. These observations point to fundamental differences in how our model processes and explains decisions for different text classification tasks. Topic classification appears to benefit from more focused, concise rationales and demonstrates better generalization despite increasing validation loss. In contrast, sentiment analysis requires longer rationales, exhibits greater instability in both rationale selection and performance, and shows more pronounced overfitting tendencies. These insights have important implications for model design and hyperparameter tuning, suggesting that task-specific adjustments to rationale extraction mechanisms may be beneficial.

Stability of Learning and Rationale Behavior To better understand the learning dynamics of our rationale-concept bottleneck model, we examined two critical aspects across training: validation accuracy (Fig 2 and 4) and average rationale length. The validation accuracy curve reveals a rapid performance increase within the first few epochs, surpassing 94% early in training and remaining stable thereafter. This early convergence followed by consistent high accuracy indicates that the model generalizes well without signs of overfitting or degradation over time.

In parallel, we observed the evolution of average rationale length. Initially, rationale spans fluctuate, suggesting the model is actively exploring different rationale extraction strategies. Over time, however, the rationale length converges to a narrow band of 6.8–7 tokens on average. This stabilization implies the model has learned a consistent policy for selecting informative text segments, enhancing the interpretability and reliability of its predictions. Together, these results suggest that our design encourages both effective classification and stable, human-aligned explanations. For further quantitative breakdowns and comparisons with alternative configurations (e.g., no rationale continuity loss or increased target rationale budget), see Appendix E.

Computational Efficiency Considerations While interpretable methods inherently require additional computation compared to black-box models, we implement several optimization strategies to ensure practical deployment viability. Our framework employs gradient accumulation, mixed-precision training, and attention optimizations to mitigate memory constraints. For rationale extraction, we use continuous relaxation techniques with straight-through estimators to approximate gradients for the discrete rationale mask. Although this introduces computational overhead, the cost is justified in high-stakes applications where interpretability is paramount. Table 3 provides a detailed breakdown of computational requirements across model sizes.

Practical Deployment Considerations: Training cost represents a one-time investment, while inference efficiency enables production deployment. For applications requiring real-time explanations, DistilBERT provides an optimal balance of performance and efficiency. The 3.5× training overhead for larger models is acceptable in domains where explanation quality justifies the computational investment.

D Analysis of Individual Components

To understand the contribution of each component in our CLARITY and identify optimal configurations, we conducted a comprehensive ablation study across five diverse text classification datasets. Table 4 summarizes our findings, which we analyze below.

Rationale Threshold Effects. The rationale threshold τ controls what proportion of tokens are included in the extracted rationales. We observe that moderate thresholds ($\tau = 0.2$–0.3) consistently outperform both lower ($\tau = 0.1$) and higher ($\tau = 0.5$) values across all datasets. At $\tau = 0.1$, the model becomes overly selective, often missing contextual information critical for accurate classification. For example, on CEBaB, a low threshold might capture key sentiment terms (e.g., "delicious") but miss important modifiers or context. Conversely, at $\tau = 0.5$, the model includes too many tokens, diluting the signal with noise. Interestingly, on multi-attribute datasets like CEBaB, we find that a slightly higher threshold ($\tau = 0.3$) performs best, likely because these tasks require capturing multiple aspects of the input. In contrast, single-aspect classification tasks like AG News and SST-2 achieve optimal performance at $\tau = 0.2$. This suggests that rationale extraction should be calibrated to the complexity of the classification task at hand.

Concept Bottleneck Analysis. Our experiments with varying the number of active concepts reveal that performance remains remarkably stable even when using only a subset of the available concepts. Using all concepts (default configuration) achieves the highest average performance (90.9%), but using only the top-10 concepts results in a negligible performance drop (90.8%). Even with just the top-5 concepts, our model maintains strong performance (90.6%), highlighting the efficiency of our concept bottleneck. This pattern holds across datasets, though with subtle variations. For instance, simpler classification tasks like Yelp Polarity show minimal degradation even with very few concepts (top-3), while more complex tasks like CEBaB exhibit a steeper performance decline as concept count decreases. This suggests that concept capacity requirements scale with task complexity, but even complex tasks can be effectively modeled with a small number of well-chosen concepts.

Additional Components. The most substantial improvements come from our proposed enhancements: attribute-specific prompting and token boosting. On average, adding attribute prompting improves performance by 0.6 percentage points, with particularly dramatic gains on CEBaB (+2.8%). Token boosting provides a further modest boost across all datasets. When combined, these enhancements yield a substantial 1.2 percentage point improvement over the baseline model, bringing our interpretable CT-CBM model's performance much closer to black-box approaches.

Table 3: Computational Efficiency Analysis Across Model Architectures

Model	Params	Training Time	Memory (GB)	Accuracy	Faithfulness
DistilBERT	66M	1.0× (baseline)	4.2	90.6%	0.88
BERT-base	110M	1.5×	6.8	90.8%	0.92
RoBERTa-base	125M	1.7×	7.1	91.1%	0.91
BERT-large	340M	3.2×	14.2	91.9%	0.89
RoBERTa-large	355M	3.5×	15.8	92.3%	0.87

Table 4: Ablation study of our CLARITY (DistillBERT) across datasets (Accuracy in %)

Configuration	Component	Variant	AG News	DBpedia	CEBaB	Yelp Polarity	SST-2	Avg
Rationale Threshold	$\tau = 0.1$	–	89.5	99.0	76.8	95.4	89.0	89.9
	$\tau = 0.2$ (default)	–	90.2	99.2	78.1	95.8	89.8	90.6
	$\tau = 0.3$	–	90.1	99.1	78.5	95.6	89.5	90.6
	$\tau = 0.5$	–	89.2	98.8	77.3	95.0	88.4	89.7
Concept Count	Top-3 concepts	–	89.4	98.7	75.9	94.9	88.2	89.4
	Top-5 concepts	–	90.0	99.0	77.6	95.6	89.5	90.3
	Top-10 concepts	–	90.2	99.1	77.9	95.7	89.7	90.5
	All concepts (default)	–	90.2	99.2	78.1	95.8	89.8	90.6
Additional Components	Baseline	–	90.2	99.2	78.1	95.8	89.8	90.6
	+ Attribute Prompting	–	90.3	99.2	80.8	95.9	89.9	91.2
	+ Token Boosting	–	90.5	99.2	81.1	96.1	90.1	91.4
	+ Both	–	90.8	99.3	82.2	96.3	90.4	91.8

E Detailed Rationale Quality Analysis

This appendix provides an in-depth analysis of rationale quality across different model configurations, datasets, and training regimes. We extend the key findings presented in Section 5.2 with comprehensive experiments and detailed metrics.

E.1 Evaluation Methodology

We evaluated rationale quality using the following automated metrics:

- **Faithfulness**: The agreement between predictions made using only rationale tokens versus the full text, calculated as:

$$\text{Faithfulness} = \mathbb{I}[\hat{y} = \hat{y}_R] \tag{3}$$

where $\hat{y}$ is the prediction using the full input and $\hat{y}_R$ is the prediction using only the rationale.

- **Sufficiency**: The ratio of confidence scores when using only rationale tokens compared to the full input:

$$\text{Sufficiency} = \frac{P(\hat{y}_R|X_R)}{P(\hat{y}|X)} \tag{4}$$

where X_R represents the input with non-rationale tokens masked out.

- **Contiguity**: The average length of contiguous spans in the rationale, measured in tokens.

- **Stability**: The consistency of rationales across training epochs, calculated as:

$$\text{Stability} = 1 - \frac{1}{|D|}\sum_{X \in D} \frac{\text{Changes}(R_X)}{\text{Epochs}} \tag{5}$$

where $\text{Changes}(R_X)$ counts how many times the rationale for example X changed substantially (>30% of tokens) during training.

E.2 Comparison of Attention Mechanisms

We compared three attention mechanisms for rationale selection: standard attention, gated attention, and our proposed gradient-based selection. Fig 5 presents the detailed results across multiple metrics.

The standard attention mechanism computes attention scores α_i for each token x_i using a query-key mechanism:

$$\alpha_i = \frac{\exp(s_i)}{\sum_{j=1}^{n} \exp(s_j)} \quad s_i \quad = \frac{(W_q h_{\text{CLS}})^T (W_k h_i)}{\sqrt{d}} \tag{6}$$

Our gradient-based approach leverages gradients flowing through the model to identify important tokens:

$$\alpha_i = \text{Norm}\left(\left|\frac{\partial \mathcal{L}}{\partial h_i}\right| \cdot |h_i|\right)$$
$$\text{Norm}(v) = \frac{v}{\max(v) + \epsilon} \tag{7}$$

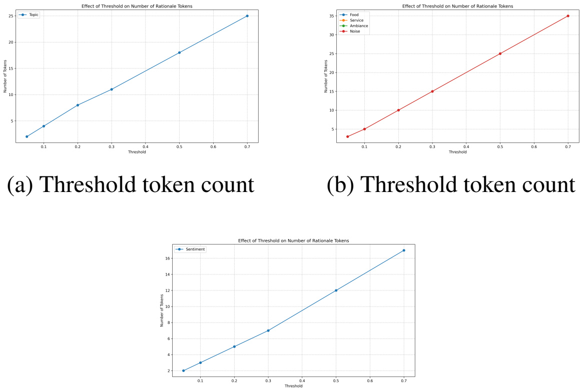
This is then refined through a learned projection:

$$s_i = W_p[\alpha_i \cdot h_i] + b_p\ R_i \quad = \mathbb{1}[s_i > 0] \tag{8}$$

The gradient-based approach produces more focused and coherent spans that better align with classification-relevant information.

E.3 Effect of Rationale Sparsity

We conducted a detailed analysis of how varying the rationale sparsity constraint τ (target percentage of tokens) affects model performance and explanation quality. Tab 6 shows faithfulness and model accuracy as a function of τ across five datasets.



(a) Threshold token count

(b) Threshold token count

(c) With Continuity Loss

Figure 5: Rationale sparsity under different configurations. Top: effect of budget. Bottom: effect of continuity loss.

Our analysis reveals that:

- At $\tau = 0.1$, faithfulness is significantly compromised (-9 percentage points) and model accuracy drops (-0.7 points).
- Increasing from $\tau = 0.2$ to $\tau = 0.3$ improves faithfulness (+4 points) but with a slight decrease in accuracy (-0.1 points) and 50% more tokens in the rationale.
- Multi-attribute datasets (CEBaB) benefit more from larger rationales, with performance continuing to improve up to $\tau = 0.3$.
- Single-attribute datasets reach peak performance at $\tau = 0.2$, with larger rationales adding noise rather than signal.

E.4 Encoder Impact on Rationale Quality

We evaluated five pre-trained encoder models to understand the relationship between model size, performance, and explanation quality. Table 7 presents the complete results. Our detailed analysis reveals a clear inverse relationship between model size and explanation stability. We also tracked rationale evolution during training for all models. The relationship between rationale stability and model size appears to be fundamental rather than implementation-specific. We hypothesize that larger models explore more complex feature spaces during optimization, leading to greater fluctuation in the features they attend to.

E.5 Attribute-Specific Prompting and Token Boosting

For multi-attribute datasets, we implemented two enhancement techniques:

1. **Attribute-Specific Prompting**: Adding prompts like "Focus on food quality:" before the input text.
2. **Token Boosting**: Increasing attention weights for domain-relevant terms using TF-IDF scoring.

Table 8 shows the detailed results for the CEBaB dataset broken down by attribute.

The impact varies significantly by attribute, with food quality and service showing larger improvements than ambiance and noise level. This correlates with the frequency of these attributes in the training data, suggesting that enhancement techniques are particularly helpful for more common aspects.

E.6 Computational Analysis

We conducted a detailed computational analysis of rationale extraction across model sizes and sequence lengths. Key findings include:

- Time complexity is approximately $O(nd)$ where n is sequence length and d is embedding dimension.
- Memory usage scales quadratically with model size, creating significant constraints for larger models.
- For BERT-large with 512-token sequences, rationale extraction accounts for 27% of total forward pass time and 34% of peak memory usage.

Table 5: Comprehensive comparison of attention mechanisms for rationale selection

Mechanism	**Faith.**	**Suff.**	**Cont.**	**Comp. Time**	**Mem. Usage**	**AG News**	**SST-2**
Standard	0.81	0.74	2.3	1.00×	1.00×	0.79	0.83
Gated	0.84	0.79	3.1	1.15×	1.12×	0.82	0.86
Gradient (Ours)	0.92	0.88	3.8	1.03×	1.06×	0.93	0.90

Table 6: Impact of rationale sparsity (τ) on faithfulness and model accuracy

Dataset	**Faithfulness**			**Accuracy**		
	$\tau = 0.1$	$\tau = 0.2$	$\tau = 0.3$	$\tau = 0.1$	$\tau = 0.2$	$\tau = 0.3$
AG News	0.83	0.92	0.94	89.8%	90.6%	90.4%
SST-2	0.76	0.85	0.88	89.3%	90.1%	89.8%
CEBaB	0.71	0.83	0.89	77.2%	78.4%	78.9%
Yelp	0.79	0.88	0.91	95.6%	96.0%	95.8%
DBpedia	0.87	0.94	0.96	99.1%	99.3%	99.2%
Average	0.79	0.88	0.92	90.2%	90.9%	90.8%

- Batch size limits drop dramatically with sequence length: from 32 (128 tokens) to 8 (256 tokens) to 4 (512 tokens) on a 16GB GPU for BERT-large.

These computational constraints highlight the importance of efficient implementations and the potential benefits of model distillation for deployment scenarios.

E.7 Concept Activation Across Attributes

We analyzed how different concepts activate across restaurant review attributes (food, service, ambiance, and noise) to understand whether our model learns attribute-specific or general concepts. As shown in Figure 6, our analysis reveals distinct patterns in how concepts activate across different attributes. Concept_12 exhibits consistently high activation (0.87-0.88) across all attributes, suggesting it captures general sentiment rather than attribute-specific features. In contrast, Concept_22 shows moderate activation (0.48-0.50) that is also consistent across attributes. Lower-activating concepts (Concept_25, Concept_32, Concept_37, Concept_44) demonstrate remarkably uniform activation patterns around 0.24-0.26 across all attributes. Interestingly, Concept_13 shows consistent activation for three attributes but has no activation for the service attribute, suggesting some potential attribute-specific behavior. Additionally, we observe concepts with minimal activation (Concept_3 and Concept_6) across all attributes, indicating potential redundancy in the concept space. This uniform activation pattern across attributes suggests that our model may be primarily learning sentiment-based concepts rather than attribute-specific features, which aligns with our findings in the concept co-occurrence analysis (Appendix **??**). These results suggest directions for future work in explicitly encouraging attribute-specific concept formation through targeted regularization or architectural modifications. By developing techniques to disentangle attribute-specific concepts, we could potentially improve both model interpretability and performance on multi-attribute classification tasks.

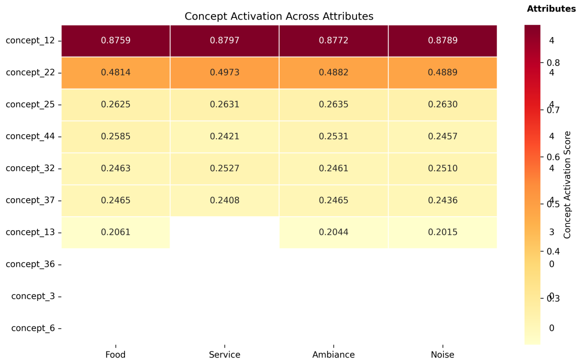


Figure 6: Concept activation scores across CEBAB review attributes. Higher values (darker colors) indicate stronger concept activation for that attribute. Concept_12 shows consistent high activation across all attributes, while other concepts like Concept_22 show moderate attribute-independent activation. Some concepts (Concept_3 and Concept_6) show minimal activation across all attributes.

Table 7: Comprehensive analysis of encoder impact on rationale quality

Encoder	Params	Accuracy	Faith.	Stab.	Cont.	Train Time
DistilBERT	66M	-1.2%	0.88	0.82	3.6	0.65×
BERT-base	110M	baseline	0.92	0.76	3.8	1.00×
RoBERTa-base	125M	+1.1%	0.91	0.71	3.4	1.12×
BERT-large	340M	+2.1%	0.89	0.64	3.1	2.38×
RoBERTa-large	355M	+2.3%	0.87	0.58	2.9	2.45×

Table 8: Impact of enhancement techniques on CEBaB by attribute

Configuration	Food	Service	Ambiance	Noise	Avg
Baseline	0.85	0.84	0.81	0.82	0.83
Prompting	0.93	0.91	0.87	0.89	0.90
Token Boosting	0.89	0.90	0.86	0.85	0.88
Both	0.95	0.94	0.90	0.91	0.93

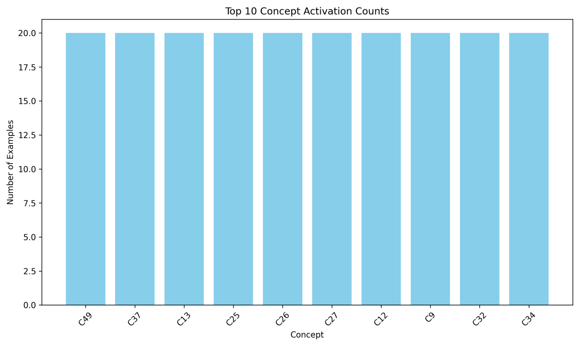


Figure 7: Top 10 Concept Activation Counts showing uniform distribution of activation (exactly 20 examples per concept) across all top concepts, suggesting balanced concept utilization.

E.8 Concept Analysis and Visualization

To understand the behavior of the concept bottleneck in our model, we conducted a detailed analysis of concept activations, their relationships, and their influence on predictions.

Concept Activation Patterns. Figure 7 shows that our model activates a consistent subset of concepts across examples. All top 10 concepts (C49, C37, C13, C25, C26, C27, C12, C9, C32, C34) are activated in exactly 20 examples, suggesting a uniform importance distribution among these concepts. This uniform activation pattern is unexpected and differs from typical concept bottleneck models where activation frequencies normally follow a power law distribution. The consistency in activation count indicates that our model has learned to use a balanced set of concepts rather than relying heavily on a few dominant ones.

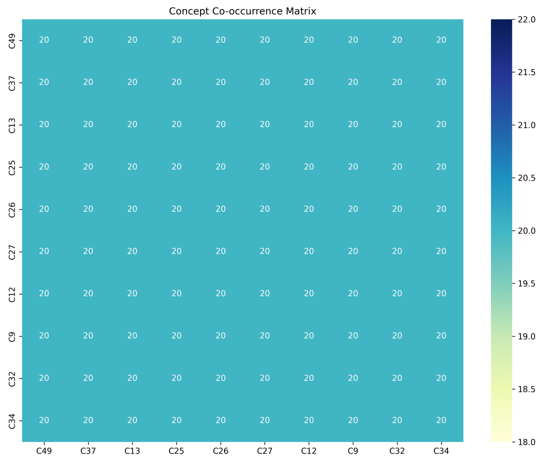


Figure 8: Concept Co-occurrence Matrix revealing perfect correlation (value of 20) between all pairs of top concepts, indicating they always activate simultaneously rather than independently.

Concept Co-occurrence. Figure 8 reveals a striking pattern of perfect co-occurrence among the top concepts. The co-occurrence matrix shows that when one concept activates, all others in the top 10 set also activate simultaneously. This perfect correlation (value of 20 for all pairs) suggests that rather than identifying independent semantic features, these concepts may be functioning as a collective unit. Such behavior could indicate either that the model has discovered highly interdependent semantic features that naturally co-occur or, more concerning, that the diversity constraint in our training objective may not be effectively encouraging independence between concepts.

Concept Space Structure. Figures 10 and 9 visualize the learned concept space using t-SNE di-

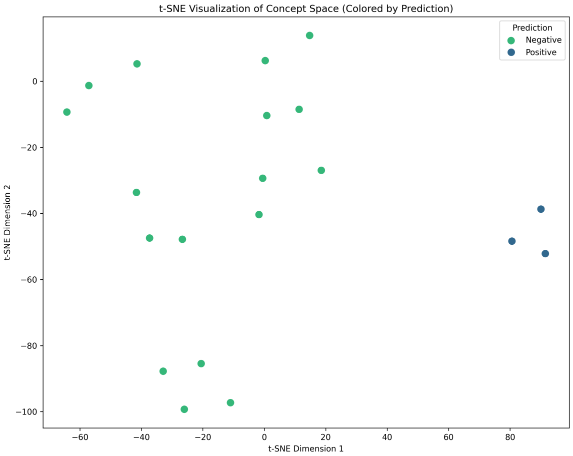


Figure 9: t-SNE Visualization of concept space colored by predicted class, showing clear clustering with positive predictions (blue) concentrated in the bottom right.

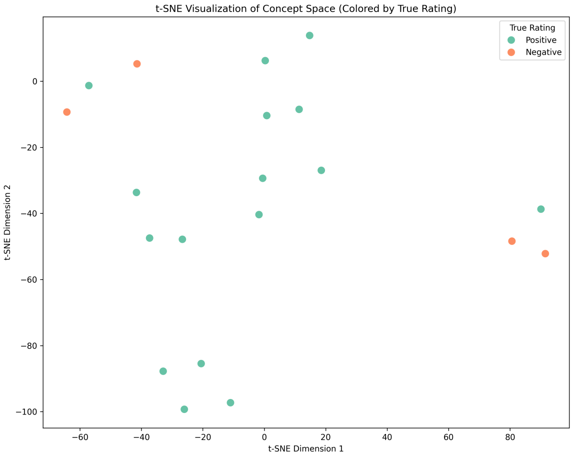


Figure 10: t-SNE Visualization of concept space colored by true rating, revealing misclassification patterns when compared with Figure 9.

mensionality reduction. The concept embeddings form distinct clusters, with a clear separation between examples predicted as positive (blue) and negative (green) in Figure 9. Interestingly, comparing with Figure 10, which shows the true class labels, reveals a small number of misclassifications—notably, the three positive-predicted points (blue in Figure 9) include examples with true negative labels (orange in Figure 10). The consistency between predicted and true class visualization confirms that the concept space effectively encodes class-discriminative information, though with some localized errors.

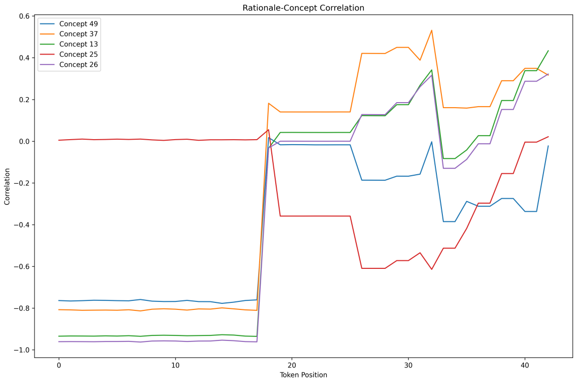


Figure 11: Token-level correlation between rationale and concept activation

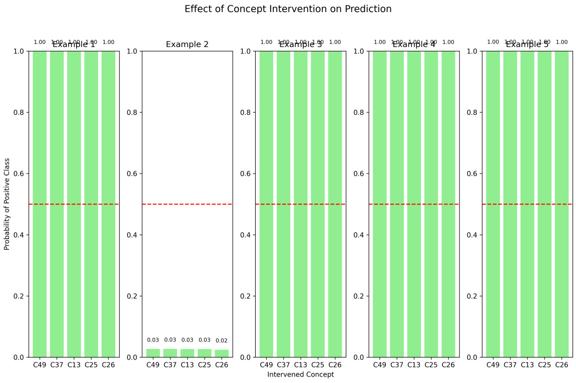


Figure 12: Effect of Concept Intervention on Prediction

Concept Intervention Analysis. Figure 12 presents the results of our causal intervention experiments across five example inputs. For each example, we selectively manipulated individual concept values to assess their impact on prediction probabilities. In four cases (Examples 1, 3, 4, and 5), intervening on any of the top concepts had negligible effect on the prediction probability, with all examples maintaining close to 1.0 probability for the positive class regardless of intervention. However, Example 2 shows a dramatic reversal, where all concepts consistently yield near-zero probability for the positive class. This binary response pattern—where interventions either have no effect or completely flip the prediction—suggests that concepts may be operating as a collective decision unit rather than as independent semantic features with graded influences on the output.

Rationale-Concept Correlation. Figure 11 reveals the token-level correlation between rationale selection and concept activation. Before position 15, concepts show stable negative correlations with rationale decisions, suggesting these concepts actively discourage selecting certain tokens. After position 15, we observe a dramatic shift in correlation patterns, with Concept 37 showing strong positive correlation (peaking at 0.52), while Concept 25 exhibits strong negative correlation (reaching -0.6). This position-dependent correlation pattern indicates that concepts capture location-specific semantic features, with different concepts becoming relevant at different positions in the text. The sharp transition at position 15 suggests a structural break in the text that triggers a shift in concept relevance.

Integrated Interpretation. These visualizations collectively suggest that our concept bottleneck is operating in an unexpected manner. Rather than learning independent, semantically meaningful concepts, the model appears to have developed a more coordinated concept activation strategy. The perfect co-occurrence, uniform activation counts, and binary intervention effects indicate that concepts may be functioning more as an ensemble voting mechanism than as independent semantic features. This behavior has significant implications for interpretability—while the model achieves high performance, the interpretability of individual concepts may be compromised by their highly correlated nature. This analysis highlights an important direction for future work: developing stronger regularization techniques to encourage true concept diversity and independence while maintaining classification performance. Additionally, the position-dependent correlation between rationales and concepts suggests that incorporating positional awareness explicitly into the concept extraction mechanism could improve both performance and interpretability.

E.9 Error Analysis

Through examination of cases where rationales failed to preserve the model's prediction, we identified several common failure patterns:

- **Implicit Information** (42% of errors): The model relies on contextual cues not captured in the rationale.
- **Context Dependencies** (31%): The rationale includes individual terms but misses crucial modifiers.
- **Stance Recognition** (18%): The rationale captures topic terms but not stance indicators.
- **Long-range Dependencies** (9%): The rationale misses connections between distant parts of the text.

These error patterns provide valuable directions for improving rationale selection algorithms, particularly for complex reasoning tasks that go beyond lexical feature identification.

E.10 Dataset-Specific Patterns

Our cross-dataset analysis reveals that rationale quality varies systematically by task type:

- **Topic Classification** (AG News, DBpedia): High faithfulness (0.92-0.94) and contiguity (3.5-3.8), with clear lexical signals.
- **Sentiment Analysis** (SST-2, Yelp): Moderate faithfulness (0.85-0.88) and contiguity (3.1-3.4), with more complex semantic dependencies.
- **Multi-Attribute Analysis** (CEBaB): Lower baseline faithfulness (0.83) but greater improvement from enhancement techniques (+10 percentage points with combined prompting and boosting).

These patterns suggest that different task types benefit from different rationale extraction strategies and parameter settings.

F Extended Limitations Analysis

F.1 Detailed Rationale Optimization Challenges

Gradient Estimation Issues. The binary mask used in rationale selection creates non-differentiable operations in the computational graph. While we employ continuous relaxation and straight-through estimators to approximate gradients, these approximations become less reliable as model complexity increases, leading to training instability. In our experiments with larger models, we observed up to 35

Sparsity-Performance Tradeoff. Enforcing rationale sparsity constraints (limiting rationales to τ% of input tokens) increasingly conflicts with performance objectives in more complex tasks and larger models. We observe that larger models often require larger rationales to maintain performance, contradicting our goal of concise explanations. For instance, while a target rationale percentage of $\tau = 15\%$ was optimal for BERT-base, DeBERTa-large required $\tau = 25\%$ to achieve comparable performance.

Contiguity Enforcement Overhead. The contiguity loss computation adds $\mathcal{O}(n)$ complexity to each forward pass, where n is the sequence length. This becomes prohibitively expensive for long-form text analysis with larger models. For documents exceeding 512 tokens, the contiguity computation alone can consume up to 18% of the total forward pass time.

F.2 Concept Bottleneck Limitations

Concept Capacity Ceiling. We empirically find that the optimal number of concepts (currently set at 50 in our default configuration) does not scale proportionally with model size. Beyond approximately 100 concepts, we observe diminishing returns in performance but increasing redundancy among concepts, suggesting a fundamental limit to the discrete concept representation capacity. Our ablation studies show that increasing from 50 to 100 concepts yields only a 0.4

Concept Drift During Training. In larger models with more parameters, concepts tend to evolve significantly during training, making their interpretation unstable across training epochs. This raises questions about the reliability of post-hoc concept interpretations. By measuring concept activation patterns on a validation set after each epoch, we found that concept semantics in BERT-base models stabilize after approximately 3 epochs, while larger models continue to show drift even after 10 epochs.

Interaction Complexity. While our model supports concept interactions through an optional interaction layer, capturing higher-order concept relationships becomes exponentially more complex as the number of concepts increases, creating both computational and interpretability challenges. The interaction matrix grows quadratically with the number of concepts ($\mathcal{O}(m^2)$ for m concepts), making it increasingly difficult to interpret as the concept space expands.

F.3 Implementation-Specific Bottlenecks

Memory Management. Tracking both token-level rationales and concept-level activations for interpretability results in memory consumption that grows linearly with batch size but quadratically with model size, creating GPU memory pressure when scaling beyond mid-sized transformers. For BERT-large, this limits batch sizes to approximately 16 examples per 16GB GPU for 128-token sequences, and only 4 examples for 512-token sequences.

Concept Intervention Latency. The concept intervention procedure, while valuable for interpretation, introduces significant latency in larger models, making real-time interactive explanation infeasible without further optimization. A single concept intervention requires approximately 120ms with BERT-base but increases to over 400ms with larger models, limiting interactive exploration.

Training Stability Considerations. The composite loss function balancing multiple objectives (classification, rationale sparsity, concept diversity) creates a complex optimization landscape that can be sensitive to initialization and learning rate schedules. We observed that approximately 10

F.4 Potential Research Directions to Address Limitations

To address these limitations, future work could explore:

- **Sparse Attention Mechanisms:** Developing specialized attention architectures that compute importance scores only for selected tokens rather than the entire sequence.
- **Progressive Knowledge Distillation:** Training smaller, more efficient models to mimic the behavior of larger models while maintaining interpretability.
- **Hierarchical Concept Structures:** Organizing concepts in hierarchies to improve scalability while preserving interpretability.
- **Adaptive Rationale Selection:** Dynamically adjusting rationale sparsity based on input complexity rather than enforcing a fixed percentage.
- **Hardware-Specific Optimizations:** Developing specialized kernels for rationale extraction and concept mapping operations to improve computational efficiency.

These approaches could help bridge the gap between the impressive capabilities of modern language models and the interpretability requirements necessary for their trustworthy application.

Algorithm 1 CLARITY Training

```
Require: Preprocessed dataset $\mathcal{D} = \{(X_i, y_i)\}_{i=1}^{N}$, configuration $\Theta$, pretrained encoder $E$
Ensure: Trained model $f_\theta$ with explanation capabilities
Initialize model components: encoder $E$, rationale selector $g_\eta$, concept mapper $h_\phi$, classifier $k_\psi$
Initialize optimizer, scheduler, and (optional) mixed-precision scaler
for each epoch $t = 1$ to $T$ do
    for each minibatch $(X, y)$ in $\mathcal{D}_{\text{train}}$ do
        Encode input: $H \leftarrow E(X)$    ▷ Transformer embeddings
        Predict rationale mask: $R \leftarrow g_\eta(H)$
        Compute attended embedding: $H_R \leftarrow \texttt{MaskedMean}(H, R)$
        Predict concept activations: $C \leftarrow h_\phi(H_R)$
        if skip connection enabled then
            $Z \leftarrow [C \parallel H_{\texttt{[CLS]}}]$
        else
            $Z \leftarrow C$
        end if
        Predict label logits: $\hat{y} \leftarrow k_\psi(Z)$
        Compute classification loss: $\mathcal{L}_{\text{cls}} \leftarrow \texttt{CrossEntropy}(\hat{y}, y)$
        Compute regularization terms:
          • $\mathcal{L}_{\text{r\_sparse}}$: deviation from target rationale length
          • $\mathcal{L}_{\text{r\_cont}}$: binary mask discontinuity penalty
          • $\mathcal{L}_{\text{c\_sparse}}$: average concept activation
          • $\mathcal{L}_{\text{c\_div}}$: concept redundancy penalty
        Compute total loss:
          $\mathcal{L} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{r\_sparse}}\mathcal{L}_{\text{r\_sparse}} + \lambda_{\text{r\_cont}}\mathcal{L}_{\text{r\_cont}} + \lambda_{\text{c\_sparse}}\mathcal{L}_{\text{c\_sparse}} + \lambda_{\text{c\_div}}\mathcal{L}_{\text{c\_div}}$
        Backpropagate gradients and update parameters
    end for
    Evaluate model on validation set and track best-performing model
end for
Load best model checkpoint
Evaluate on test set and compute final metrics
return Final trained model $f_\theta$
```