An Information-Theoretic Approach to Benign Leakage in Static Concept Embedding Models

Tianchao Li
UiT – The Arctic University of Norway
tianchao.li@uit.no

Shujian Yu
Vrije Universiteit Amsterdam, UiT – The Arctic University of Norway
s.yu3@vu.nl

Robert Jenssen
UiT – The Arctic University of Norway, University of Copenhagen, Norwegian Computing Center
robert.jenssen@uit.no

Abstract

Self-explanatory Concept Bottleneck Models (*CBMs*) integrate human-defined concepts into their internal representations to achieve interpretability, predictability, and intervenability. However, *CBM*s often suffer from concept leakage, where concept embeddings encode input information beyond the concept. Concept leakage undermines the core properties of *CBM*s as concept representations are dynamically extracted from inputs by the neural architecture, entangling with input-specific information, like end-to-end neural networks. To address this issue, we are the first to introduce vector quantization into *CBM*s for learning static concept embeddings under binary concept supervision, which we term Static Concept Embedding Models (*StaticCEM*). During the training forward process and test inference, static concept embeddings remain fixed, ensuring theoretical leakage resistance, a claim that is also empirically validated by our experiments. Moreover, since the input contains more substantial information than the human-defined concepts, predictions based solely on concepts may underperform compared to models that utilize the full input. To bridge the performance gap, we further inject controlled and limited input information into the leakage-resistant static embeddings via a dot-product projection, governed by the trade-off of a dual Information Bottleneck mechanism. We term this injection *Benign Leakage*, as it largely preserves *CBM*'s properties while boosting performance. Our experimental results demonstrate that this approach matches or surpasses state-of-the-art methods.

1 Introduction

Deep neural networks (*NN*s) have gained widespread adoption across various applications due to their state-of-the-art performance. However, in high-stakes domains such as medical diagnostics Møller et al. (2024) and anomaly detection Pei et al. (2021), interpretability and transparency are non-negotiable for AI-driven decision-making. Many self-explanatory models were proposed, such as the prototype-based Gautam et al. (2022); Shen (2025); Gautam et al. (2023), and the information bottleneck-based Choi et al. (2024); Zheng et al. (2024). Although both methods operate on latent representations, their final explanations are grounded in the input space. Explanations are barely understood by non-domain experts and offer no clear path for intervention.

Concept Bottleneck Models (*CBM*s) Koh et al. (2020) utilize high-level and human-understandable concepts as their internal representation, allowing for the interpretation of the prediction to be attributed to the contribution of each concept representation, and for the intervention of the concept correction. The preliminary and related work about *CBM*s are given in Section A.1 and Section A.2, respectively. Despite intuitive properties, concerns about *CBM*s have been raised regarding their

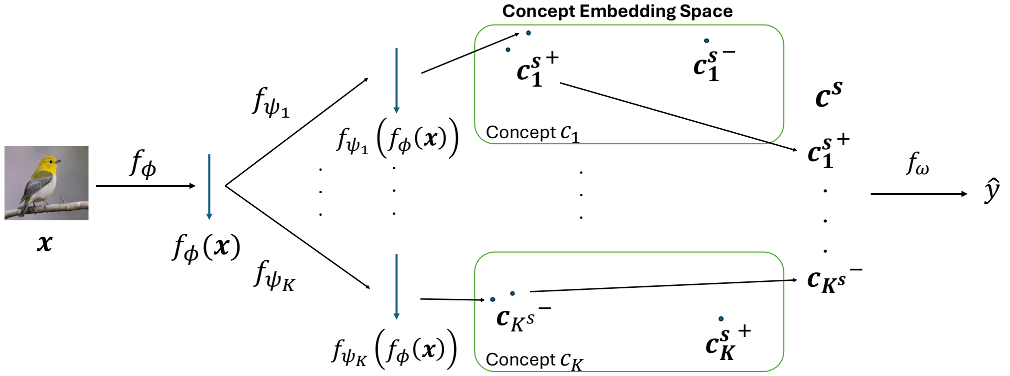

Figure 1: *StaticCEM*: we learn the static concept embeddings $\{\boldsymbol{c}_i^{s+}, \boldsymbol{c}_i^{s-}\}_{i=1}^K$ under the supervision of $\boldsymbol{c}$. $\boldsymbol{c}^s$ is the final concept embedding, concatenating all the selected static concept embeddings $\{\boldsymbol{c}_i^s\}_{i=1}^K$ ($\boldsymbol{c}_i^s$ is $\boldsymbol{c}_i^{s+}$ or $\boldsymbol{c}_i^{s-}$).

fidelity to the design properties Margeloiu et al. (2021). Particularly, concept leakage is weakening its faithfulness. Concept leakage refers to the unintended inclusion of input-specific information within the concept representation, so that concept information does not primarily determine the final prediction Mahinpei et al. (2021). Besides, another design limitation of *CBM*s lies in the mismatch between rich visual information and sparse linguistic semantics He et al. (2022). Specifically, it is difficult to capture the full content of an image using only a few language-based concepts, which typically results in *CBM*s lagging behind *NN*s. To address both issues, this paper makes two key conceptual contributions and presents their practical implementation. In summary, our main contributions are:

- **Static Concept Embedding:** We first introduce the vector quantization into *CBM*s. The learnable codebooks make the concept embeddings static, where the concept embeddings will not change with the input. In principle, this provides resistance to concept leakage, and design *Static Concept Embedding Models* (*StaticCEM*);
- **Benign Leakage:** We inject input-specific information into the leakage-resistant static concept embeddings by the dot product. We propose the dual Information Bottleneck (IB) Tishby et al. (2000) mechanism to make the injected information related to concepts. The intended leakage negligibly affects core properties of *CBM*s, so we refer to our approach as *Benign Leakage*;
- **Comprehensive Experiments:** We conduct experiments on diverse datasets and show that ours outperforms dataset-specific state-of-the-art *CBMs* Zarlenga et al. (2023); Kim et al. (2023); Laguna et al. (2024).

2 Methodology

We propose Static Concept Embedding Models (*StaticCEM*), incorporating the vector quantization technique to learn static concept embeddings supervised by concepts. These embeddings are termed static because they depend solely on each concept's activation state (0 or 1) and remain fixed without respect to the input during the forward process in training and at inference time. The static concept embedding results in leakage resistance. To achieve performance on par with or better than *NN*s, we further use the dot product to inject the information into the selected static concept embedding, and apply IB Tishby et al. (2000) during training, encouraging the leakage source to align closely with the concepts. After information injection, our approach still maintains the core properties of *CBM*s, which we refer to as *Benign Leakage*.

2.1 Static Concept Embedding Model

In the past works, high-dimensional concept embeddings are extracted from $\boldsymbol{x}$ via the neural architecture, so it dynamically depends on the sample $\{\boldsymbol{x}^{(j)}\}_{j=1}^N$ of $\boldsymbol{x}$. The information of $\boldsymbol{x}$ is leaked to the concept embedding Zarlenga et al. (2022). Although the intervention apparently exists by selecting the concept-corresponding neural projector, the model bypasses the concept bottleneck to make the final prediction Zarlenga et al. (2025). Therefore, the internal mechanism of *CBM*s becomes identical to that of *NN*s. The interpretability, predictability, and intervenability of the *CBM* exist in name only. To alleviate this issue, we assert that static concept embeddings, which depend solely on the activation state of each concept (inactive vs. active), are essential to preserve the core properties of the *CBM*. Precisely, the static term means that when making the prediction (the forward process of the training and the inference of the test), the learnable static concept embeddings are fixed, which will be updated in the backward process of the training. Based on this principle, we propose Static Concept Embedding Models (*StaticCEM*).

Each dynamic concept embedding $f_{\psi_i}(f_\phi(\boldsymbol{x}))$ should become a discrete representation (the active concept embedding $\boldsymbol{c}_i^{s+}$ or the inactive concept embedding $\boldsymbol{c}_i^{s-}$) between concept statuses in the vector quantization. For instance, if the concept is 1, the dynamic embedding should align tightly with $\boldsymbol{c}^{s+}$ while remaining distant from $\boldsymbol{c}^{s-}$. In our framework as shown in Figure 1, f_{ψ_i} continues to extract the i-th concept's dynamic embedding $f_{\psi_i}(f_\phi(\boldsymbol{x}^{(j)}))$ from the $\boldsymbol{x}^{(j)}$'s representation $f_\phi(\boldsymbol{x}^{(j)})$. The dimension of $f_{\psi_i}(f_\phi(\boldsymbol{x}^{(j)}))$ is the same as the static concept embeddings, so the Euclidean distances between the dynamic concept embedding and the two-state static concept embeddings can be computed by:

$$d_i^{(j)-} = \|f_{\psi_i}(f_\phi)(\boldsymbol{x}^{(j)}) - \boldsymbol{c}_i^{s-}\|_2^2, \tag{1}$$

$$d_i^{(j)+} = \|f_{\psi_i}(f_\phi)(\boldsymbol{x}^{(j)}) - \boldsymbol{c}_i^{s+}\|_2^2. \tag{2}$$

The i-th concept's prediction $\widehat{c_i^{(j)}}$ obtained by:

$$\widehat{c_i^{(j)}} = \begin{cases} 0 & \text{if } \min\{d_i^{(j)+}, d_i^{(j)-}\} = d_i^{(j)-} \\ 1 & \text{if } \min\{d_i^{(j)+}, d_i^{(j)-}\} = d_i^{(j)+} \end{cases} \tag{3}$$

The training objective supervised by the concept is:

$$\mathcal{L}_c = \sum_{j=1}^{N} \frac{1}{K} \sum_{i=1}^{K} \begin{cases} L_{\boldsymbol{c}}(c^{(j)}, \widehat{c_i^{(j)}}) + d_i^{(j)-} - d_i^{(j)+} & \text{if } \widehat{c_i^{(j)}} = 0 \\ L_{\boldsymbol{c}}(c^{(j)}, \widehat{c_i^{(j)}}) + d_i^{(j)+} - d_i^{(j)-} & \text{if } \widehat{c_i^{(j)}} = 1 \end{cases}, \tag{4}$$

In principle, minimizing $L_{\boldsymbol{c}}(c^{(j)}, \widehat{c_i^{(j)}})$ will correct the wrong prediction. However, the rest of the terms will enforce the dynamic concept embedding and the wrong-state static concept embedding closer, and the other farther.

Practically, Equation (4) cannot be optimized directly. We utilize the Gumbel-Softmax method in Baevski et al. (2020), which learns the codebook for vector quantization. The Gumbel-Softmax in Equation (5) can convert the distances $d_i^{(j)-}, d_i^{(j)+}$ to the probabilities $\widehat{c_i^{(j)}}$ of two states ($\widehat{c_i^{(j)}}[0]$ for 0 and $\widehat{c_i^{(j)}}[1]$ for 1) with the temperature τ, and the Gumbel-Softmax is differentiable.

$$\widehat{c_i^{(j)}} = \text{Gumbel-Softmax}([-d_{ij}^{-}, -d_{ij}^{+}], \tau) \tag{5}$$

The Gumbel-Softmax still maintains the relationship between the distance and the predicted concept, so it does not involve extra information or information leakage. Therefore, the training objective can be simplified as:

$$\mathcal{L}_c = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{K} \sum_{i=1}^{K} L_{\boldsymbol{c}}(c_i^{(j)}, \widehat{c_i^{(j)}}[1]), \tag{6}$$

where we choose binary cross entropy as $L_{\boldsymbol{c}}(\cdot,\cdot)$. Since Gumbel-Softmax is differentiable, the backward process will update the two distances to correct the wrong prediction. It will make the dynamic concept embedding closer to the right static concept embedding and farther from the wrong

Dataset	`XOR`				`Dot`				`Trigonometry`			
Metric	Acc(%)		*CLM*		Acc(%)		*CLM*		Acc(%)		*CLM*	
	Task	Concept	Predictor	Matrix	Task	Concept	Predictor	Matrix	Task	Concept	Predictpr	Matrix
Hard *CBM*	74.17	99.17	-	-	64.83	99.67	-	-	82.33	99.33	-	-
StaticCEM	73.17	99.58	**-0.0046**	**0.0183**	60.67	98.42	**-0.0011**	**-0.0583**	84.33	97.72	**0.0258**	**-0.6794**
CEM	99.33	99.67	0.0147	0.0205	97.33	99.00	0.5019	0.3468	98.00	99.39	0.3608	0.0994

Table 1: Evaluation of *StaticCEM* and *CEM*s across datasets among the task accuracy, mean concept accuracy, and *CLM*.

one, which is achieved by updating the learnable parameters in f_ϕ, f_ψ, and $\{\boldsymbol{c}_i^{s+}, \boldsymbol{c}_i^{s-}\}_{i=1}^K$. Equation (7) illustrates the selection between static concept embeddings.

$$\boldsymbol{c}_i^{s(j)} = \begin{cases} \boldsymbol{c}_i^{s+} & \text{if } \widehat{c_i^{(j)}}[1] \geq \widehat{c_i^{(j)}}[0] \\ \boldsymbol{c}_i^{s-} & \text{if } \widehat{c_i^{(j)}}[1] < \widehat{c_i^{(j)}}[0] \end{cases} \tag{7}$$

$\boldsymbol{c}^{s(j)} = \texttt{cat}(\{\boldsymbol{c}_i^{s(j)}\}_{i=1}^K)$ is the final concept embedding based on static concept embeddings, and the input of f_ω. In *StaticCEM*, f_ω can be a fully-connected layer Koh et al. (2020) or learnable class prototypes Kim et al. (2023) Gautam et al. (2022) by minimizing $\mathcal{L}_{\boldsymbol{y}}$.

The conventional vector quantization utilizing a sufficiently large codebook guarantees the lossless discrete representation Chen et al. (2024). By contrast, we pretend to have the information loss from our codebooks with size 2. Supervised by $\boldsymbol{c}^{(j)}$, $\{\boldsymbol{c}_i^{s+}, \boldsymbol{c}_i^{s-}\}^K$ preserve the concept-relevant information as much as possible, and ignore the concept-irrelevant information. In conclusion, *StaticCEM*s are leakage resistant as Theorem 1. The proof of Theorem 1 is given in Section A.3.

Theorem 1 *StaticCEMs are leakage-resistant.*

We use the Hard *CBM* as the anchor for the comparative leakage measure (*CLM*) based on Equation (14) :

$$L_{CLM} = H(y|\boldsymbol{c}, \texttt{round}(\hat{\boldsymbol{c}})) - H(y|\boldsymbol{c}, \vec{\boldsymbol{c}}), \tag{8}$$

where $\vec{\boldsymbol{c}}$ represents the concept embedding as the input of f_ω. $\vec{\boldsymbol{c}}$ is much more informative than $\texttt{round}(\hat{\boldsymbol{c}})$, so the smaller L_{CLM} is, the less concept leakage is. Here, we consider two methods to estimate both conditional entropy terms in Equation (8): the non-parametric kernel matrix-based estimator Yu et al. (2025), and the parametric predictor-based estimator Makonnen et al. (2025). We provide details of two estimators in the supplementary material. We evaluate *CLM* on synthetic datasets (`XOR`, `Dot`, and `Trigonometry`) introduced by Zarlenga et al. (2023) to validate *StaticCEM*'s strong resistance to concept leakage. Table 6 shows *StaticCEM* consistently yields much lower *CLM* values compared to the *CEM*. In some cases, *StaticCEM*'s *CLM* values are even negative. These negative values likely result from variance around 0 estimation and further support the interpretation that *StaticCEM* should not leak information for the task prediction. *StaticCEM*'s task accuracy lags significantly behind *CEM*, demonstrating the need for information injection to bridge the performance gap.

2.2 Benign Leakage

Besides the information of $\boldsymbol{c}$, $\boldsymbol{x}$ also contains other task-relevant information Schrodi et al. (2025). Taking the object detection task in Figure 2 as an example, the tree and the sun would remain undetected if relying solely on the human-language caption. So, $\boldsymbol{c}^s$ is not a sufficient embedding statistic of $\boldsymbol{x}$ and $f_\phi(\boldsymbol{x})$. To bridge the performance gap, we propose the concept of benign leakage, where we intentionally inject partial information from $f_{\psi_i}(f_\phi(\boldsymbol{x}))$ into $\boldsymbol{c}_i^s$.

Definition 1 (Benign Leakage (BL)) *If the injected information is constrained by the concept and the injected representation preserves the three CBM properties: interpretability, intervenability, and predictability, the injection operation is called benign leakage.*

In this work, we intentionally inject the information from $f_{\psi_i}(f_\phi(\boldsymbol{x}))$ into $\boldsymbol{c}_i^s$ by the dot product, and obtain a concept embedding $\hat{\boldsymbol{c}^s} \in \mathbb{R}^K$:

$$\hat{c_i^s}^{(j)} = \boldsymbol{c}_i^{sT} f_{\psi_i}(f_\phi(x^{(j)})) \in \mathbb{R}, \tag{9}$$

$$\hat{\boldsymbol{c}^s}^{(j)} = \texttt{cat}(\{\hat{c_i^s}^{(j)}\}_{i=1}^K). \tag{10}$$

The dot product represents the scalar projection of the predicted concept embedding $f_{\psi_i}(f_\theta(\boldsymbol{x}))$ onto the selected static concept embeddings $\boldsymbol{c}_i^s$. These static embeddings directly determine the axes along which the learned representation is projected, thereby playing a crucial role in shaping the resulting scalar value.

IB Tishby et al. (2000) is represented as the maximization of the Lagrangian optimization:

$$\mathcal{L}_{IB} = I(y; \boldsymbol{t}) - \beta I(\boldsymbol{t}; \boldsymbol{x}), \tag{11}$$

where $\boldsymbol{t}$ is the representation variable of the hidden layer and β controls the fundamental tradeoff between these two information terms. IB is interpreted as the trade-off between the complexity of the representation $\boldsymbol{t}$ and the amount of relevant information preserved by $\boldsymbol{t}$. We can detect that leakage sources are the neural embeddings $f_\phi(\boldsymbol{x})$ and $f_{\psi_i}(f_\phi(\boldsymbol{x})$. We implement a dual IB mechanism during the training. The first is Concept-based Information Bottleneck (*CIB*), as shown in Equation (12). The first term encourages $f_{\psi_i}(f_\phi(\boldsymbol{x}))$ to retain information relevant to $\boldsymbol{c}_i$, while the second term compresses the information in $f_{\psi_i}(f_\phi(\boldsymbol{x}))$ unrelated to c_i from $f_\phi(\boldsymbol{x})$. Due to the sufficient encoder assumption Tian et al. (2020), *CIB* compresses the redundant information about $\boldsymbol{x}$ into $f_{\psi_i}(f_\phi(\boldsymbol{x}))$. β_{CIB} controls the tradeoff of $f_{\psi_i}(f_\phi(\boldsymbol{x}))$ between the concept and input.

$$\mathcal{L}_{CIB} = \frac{1}{K}\sum_{i=1}^{K} I(c_i; f_{\psi_i}(f_\phi(\boldsymbol{x}))) - \frac{\beta_{CIB}}{K}\sum_{i=1}^{K} I(f_{\psi_i}(f_\phi(\boldsymbol{x})); f_\phi(\boldsymbol{x})) \tag{12}$$

The second is Label-based Information Bottleneck (*LIB*) as shown in Equation (13), making $\hat{\boldsymbol{c}^s}$ informative for y, while compressing the redundant information from $\boldsymbol{x}$ as well. β_{CIB} controls the tradeoff of $\hat{\boldsymbol{c}^s}$ between the target and input.

$$\mathcal{L}_{LIB} = I(y; \hat{\boldsymbol{c}^s}) - \beta_{LIB} I(\hat{\boldsymbol{c}^s}; f_\phi(\boldsymbol{x})), \tag{13}$$

Maximizing the first term in Equation (12) and Equation (13) can be approximated by minimizing $\mathcal{L}_{\boldsymbol{c}}$ and $\mathcal{L}_{\boldsymbol{y}}$ respectively Zheng et al. (2024). However, *LIB* in Equation (13) can only work on the joint training. If not, its first term will be trained in the concept part, and its second term will be trained in the task part. With the trade-off decoupled, minimizing $I(\hat{\boldsymbol{c}}^s; f_\theta(\boldsymbol{x}))$ will make $\hat{\boldsymbol{c}^s}$ as compact as possible and lose task-relevant information. After information injection, our approach is more suitable for joint training. There is no neural mapping from $f_\phi(\boldsymbol{x})$ to $\hat{\boldsymbol{c}^s}$, so the variational estimation does not work. Therefore, we implement the statistical estimation, Cauchy-Schwarz Quadratic Mutual Information (CS-QMI) Yu et al. (2024). Further implementation details are provided in the supplementary material.

As aforementioned, concept leakage is the preliminary of the comparative performance to *NN*s. However, the leakage should not affect the properties of *CBM*s. We define benign leakage by preserving the variant properties of *CBM*s:

- Interpretability: The contribution of each concept to the final prediction is calculable.
- Intervenability: The intervention really generates a positive effect for the final prediction.
- Predictability: Concepts dominate the final concept embedding, so the model is transferable and robust to domain shift because concepts are high-level.

Our approach satisfies the benign leakage's properties. Firstly, the final concept embedding $\hat{\boldsymbol{c}^s}$ has the same dimension as the concept $\boldsymbol{c}$. So, it is easy to obtain the contribution of every concept scalar representation $\hat{c_i^s}$. Secondly, minimizing $\mathcal{L}_c$ enforces $\boldsymbol{c}_i^{s+}$ and $\boldsymbol{c}_i^{s-}$ far from each other. For the latent prototypes, the intervened elements in $\hat{\boldsymbol{c}}$ will increase/decrease the distance contribution to probably trigger the correction. Lastly, the scalar projection significantly depends on the selected static concept embedding $\boldsymbol{c}_i^s$, so the static concept embedding still dominates the concept representation and the inference of the task.

3 Experiments

In this section, we first introduce the datasets for experiments. Next, the experimental results are illustrated compared to the state-of-the-art baselines on the corresponding dataset. Furthermore, we investigate the concept embedding $\hat{\boldsymbol{c}}$ to validate that it satisfies benign leakage. The details about the dataset and experimental setup are shown in Section A.8 and Section A.9

3.1 Experimental Results

Table 2 compares the task accuracy and concept accuracy between ours and the baselines across datasets. Our method outperforms baselines in both task accuracy and concept accuracy. The higher concept accuracy means our final prediction depends more on the concept prediction. During training, *CEM* achieves 91.32% task accuracy but only 68.03% concept accuracy. This discrepancy suggests that the final prediction in *CEM* does not fully rely on concept predictions. We argue that the concept embeddings extracted by the neural architecture may bypass the concept bottleneck.

Dataset	Model	Task Acc (%)	Concept Acc (%)
MNIST	*StaticCEM+BL*	**89.67**	93.42
	ProbCBM	86.67	**93.80**
CheXpert	*StaticCEM+BL*	**85.04**	79.55
	BeyondCBM	84.19	–
CUB	*StaticCEM+BL*	**53.57**	**87.55**
	CEM	43.51	71.15

Table 2: Performance comparison across datasets.

3.2 Predictability

The predictability can be reflected in the domain shift. When we design the new dataset derived from MNIST, the shifted domain (MNIST Shift) has been provided. CheXpert and MIMIC-CXR have identical concepts and the similar input. We regard MIMIC-CXR as the shifted domain of CheXpert. Table 3 compares our performances on the shifted domain to the baselines'. Our approach performs better than *ProbCBM* on MNIST Shift, and similarly to *BeyondCBM* (almost *NN*s) on MIMIC-CXR. Our approach remains effective under domain shift, indicating that the predictability of *CBM*s can be partially preserved.

Dataset	Model	Task Acc (%)	Concept Acc (%)
MNIST Shift	*StaticCEM+BL*	**52.19**	**61.49**
	ProbCBM	52.10	55.40
MIMIC-CXR	*StaticCEM+BL*	64.24	89.42
	BeyondCBM	**65.80**	–

Table 3: Model Generalization on Shifted Domains.

Dataset	Intervened Acc (%)	Acc Gain (%)
MNIST	94.42	4.75
CUB	61.79	8.22
CheXpert	85.04	0.00
MIMIC-CXR	65.80	1.56

Table 4: Accuracy gain of *ours* after concept intervention.

3.3 Intervenability

The intervenability is a vital property of *CBM*s. Ideally, the intervened concept embedding would correct the final prediction. Figure 3 visualizes the first sample of our model on MNIST Original. From the figure, the label is 0, so $\hat{c}^{(1)}$ should be close to the 1st prototype. If the concept prediction of digit 1 is 0, $\hat{c_2}^{(1)} = 2.2124$, otherwise $\hat{c_2}^{(1)} = 1.0443$. If the concept prediction is intervened from 0 to 1, the distance contribution of $\hat{c_2}^{(1)}$ to the 1st prototype clearly decreases, and the final prediction is more probably 0. Table 4 shows the accuracy gain after intervention. Our model triggers a significant accuracy improvement on MNIST and CUB, but the improvement on MIMIC-CXR and MIMIC-CXR is small or even none. We hypothesize that replacing the value -1, which denotes uncertainty, with 0 may lead to information loss. Since the annotations contain numerous missing or uncertain entries, such a replacement may distort the underlying relationship between concepts and the target labels. The concept quality negatively affects our interventability.

In summary, the concept bottleneck layer aims to improve interpretability rather than performance, and in fact, the concept bottleneck may sacrifice performance. Compared to the variants of *CBMs*, even the end-to-end, our approach maintains competitiveness in performance with benign leakage. We also run ablation studies in Section A.11 to validate the importance of β_{CIB} and β_{LIB}.

4 Conclusion

In this paper, we first propose the leakage-resistant model, Static Concept Embedding Models (*StaticCEM*), learning the static concept embeddings in contrast to the dynamic concept embedding extracted from the input by the neural architecture. Leakage resistance is empirically validated in our experiment. Due to the information imbalance between the visual input and the language concept, we inject the input information into the static concept embedding with the constraint of IB to bridge the performance gap. We term our approach as *Benign Leakage* because properties of *CBM*s are largely preserved. In the future, we will explore a more controllable injection method than the dot product, such that the predictability and intervenability can be greatly enhanced.

Acknowledgments

The authors would like to thank the anonymous reviewers for constructive comments. This work was supported by the Research Council of Norway, Integreat - Norwegian Centre for knowledge-driven machine learning, project number 332645.

References

Alexei Baevski, Steffen Schneider, and Michael Auli. vq-wav2vec: Self-supervised learning of discrete speech representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rylwJxrYDS.

Hang Chen, Sankepally Sainath Reddy, Ziwei Chen, and Dianbo Liu. Balance of number of embedding and their dimensions in vector quantization, 2024. URL https://arxiv.org/abs/2407.04939.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794. ACM, August 2016. doi: 10.1145/2939672.2939785. URL http://dx.doi.org/10.1145/2939672.2939785.

Changkyu Choi, Shujian Yu, Michael Kampffmeyer, Arnt-Børre Salberg, Nils Olav Handegard, and Robert Jenssen. Dib-x: Formulating explainability principles for a self-explainable model through information theoretic learning. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7170–7174, 2024. doi: 10.1109/ICASSP48485.2024.10447094.

Srishti Gautam, Ahcène Boubekki, Stine Hansen, Suaiba Salahuddin, Robert Jenssen, Marina Höhne, and Michael Kampffmeyer. Protovae: A trustworthy self-explainable prototypical variational model. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17940–17952. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/722f3f9298a961d2639eadd3f14a2816-Paper-Conference.pdf.

Srishti Gautam, Marina M.-C. Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. This looks more like that: Enhancing self-explaining models by prototypical relevance propagation. *Pattern Recognition*, 136:109172, 2023. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2022.109172. URL https://www.sciencedirect.com/science/article/pii/S0031320322006513.

Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. Addressing leakage in concept bottleneck models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 23386–23397. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/944ecf65a46feb578a43abfd5cddd960-Paper-Conference.pdf.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15979–15988, 2022. doi: 10.1109/CVPR52688.2022.01553.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. Chexpert: a large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.3301590. URL `https://doi.org/10.1609/aaai.v33i01.3301590`.

Alistair E. W. Johnson, Tom J. Pollard, Nathaniel R. Greenbaum, Matthew P. Lungren, Chih ying Deng, Yifan Peng, Zhiyong Lu, Roger G. Mark, Seth J. Berkowitz, and Steven Horng. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs, 2019. URL `https://arxiv.org/abs/1901.07042`.

Patrick Kidger and Terry Lyons. Universal Approximation with Deep Narrow Networks. In Jacob Abernethy and Shivani Agarwal (eds.), *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pp. 2306–2327. PMLR, 09–12 Jul 2020. URL `https://proceedings.mlr.press/v125/kidger20a.html`.

Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. Probabilistic concept bottleneck models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8245. doi: 10.1561/2200000056. URL `http://dx.doi.org/10.1561/2200000056`.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/koh20a.html`.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Sonia Laguna, Ričards Marcinkevičs, Moritz Vandenhirtz, and Julia E. Vogt. Beyond concept bottleneck models: How to make black boxes intervenable? In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 85006–85044. Curran Associates, Inc., 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/9a439efaa34fe37177eba00737624824-Paper-Conference.pdf`.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Tianchao Li and Yulong Pei. Mole: Modular learning framework via mutual information maximization, 2023. URL `https://arxiv.org/abs/2308.07772`.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/32cbf687880eb1674a07bf717761dd3a-Paper.pdf.

Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. Promises and pitfalls of black-box concept learning models, 2021. URL https://arxiv.org/abs/2106.13314.

Mikael Makonnen, Moritz Vandenhirtz, Sonia Laguna, and Julia E Vogt. MEASURING LEAKAGE IN CONCEPT-BASED METHODS: AN INFORMATION THEORETIC APPROACH. In *ICLR 2025 Workshop: XAI4Science: From Understanding Model Behavior to Discovering New Scientific Knowledge*, 2025. URL https://openreview.net/forum?id=u0oGjnshKt.

Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. Do concept bottleneck models learn as intended?, 2021. URL https://arxiv.org/abs/2105.04289.

Bjørn Leth Møller, Sepideh Amiri, Christian Igel, Kristoffer Knutsen Wickstrøm, Robert Jenssen, Matthias Keicher, Mohammad Farid Azampour, Nassir Navab, and Bulat Ibragimov. NEMt: Fast targeted explanations for medical image models via neural explanation masks. In *Northern Lights Deep Learning Conference 2025*, 2024. URL https://openreview.net/forum?id=PenPJYfmaA.

Sejun Park, Chulhee Yun, Jaeho Lee, and Jinwoo Shin. Minimum width for universal approximation. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=O-XJwyoIF-k.

Yulong Pei, Fang Lyu, Werner van Ipenburg, and Mykola Pechenizkiy. Subgraph anomaly detection in financial transaction networks. In *Proceedings of the First ACM International Conference on AI in Finance*, ICAIF '20, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450375849. doi: 10.1145/3383455.3422548. URL https://doi.org/10.1145/3383455.3422548.

Angelos Ragkousis and Sonali Parbhoo. Tree-based leakage inspection and control in concept bottleneck models, 2024. URL https://arxiv.org/abs/2410.06352.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 05 2019. doi: 10.1038/s42256-019-0048-x.

Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022. doi: 10.1109/ACCESS.2022.3167702.

Simon Schrodi, David T. Hoffmann, Max Argus, Volker Fischer, and Thomas Brox. Two effects, one trigger: On the modality gap, object bias, and information imbalance in contrastive vision-language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=uAFHCZRmXk.

Ke-Yuan Shen. Learn hybrid prototypes for multivariate time series anomaly detection. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8TBGdH3t6a.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6827–6839. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c2e5eaae9152079b9e95845750bb9ab-Paper.pdf.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, 2000. URL https://arxiv.org/abs/physics/0004057.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Xinyue Xu, Yi Qin, Lu Mi, Hao Wang, and Xiaomeng Li. Energy-based concept bottleneck models: Unifying prediction, concept intervention, and probabilistic interpretations. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=I1quoTXZzc.

Shujian Yu, Xi Yu, Sigurd Løkse, Robert Jenssen, and Jose C Principe. Cauchy-schwarz divergence information bottleneck for regression. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7wY67ZDQTE.

Shujian Yu, Hongming Li, Sigurd Løkse, Robert Jenssen, and José C. Príncipe. The conditional cauchy-schwarz divergence with applications to time-series data and sequential decision making. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5901–5917, 2025. doi: 10.1109/TPAMI.2025.3552434.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *ICLR 2022 Workshop on PAIR^2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2022. URL https://openreview.net/forum?id=HAMeOIRD_g9.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lio, and Mateja Jamnik. Concept embedding models: beyond the accuracy-explainability trade-off. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Mateo Espinosa Zarlenga, Katherine M. Collins, Krishnamurthy (Dj) Dvijotham, Adrian Weller, Zohreh Shams, and Mateja Jamnik. Learning to receive help: intervention-aware concept embedding models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Mateo Espinosa Zarlenga, Gabriele Dominici, Pietro Barbiero, Zohreh Shams, and Mateja Jamnik. Avoiding leakage poisoning: Concept interventions under distribution shifts. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=7mxDGiF01U.

Kaizhong Zheng, Shujian Yu, Baojuan Li, Robert Jenssen, and Badong Chen. Brainib: Interpretable brain network-based psychiatric diagnosis with graph information bottleneck. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2024. doi: 10.1109/TNNLS.2024.3449419.

A Appendix

A.1 Preliminary

*CBM*s Koh et al. (2020) have garnered significant attention due to their self-explanatory nature. Formally, let $\boldsymbol{x} \in \mathcal{X}$, $y \in \mathcal{Y}$, and $\boldsymbol{c} \in \mathcal{C}$ be the input, label, and the concept, and $\boldsymbol{c}_i$ denote the i-th concept. In this paper, we use boldface to represent the multi-dimensional vector. *CBM*s learn the mapping from $\mathcal{X}$ to $\mathcal{Y}$ via $\mathcal{C}$, namely $\mathcal{X} \to \mathcal{C} \to \mathcal{Y}$ instead of $\mathcal{X} \to \mathcal{Y}$ in *NN*s. Thus, *CBM*s can be interpreted by the contribution of each human-understandable concept to the final prediction. The wrongly predicted concept can be intervened before inputting the task predictor, which desirably and potentially corrects the prediction. Overall, *CBM*s have three properties:

- Interpretability: Understanding which concepts significantly influence the model's predictions;

- Predictability: Ensuring that the model can accurately predict the final output using only the inferred concepts;
- Intervenability: Allowing improvements in prediction accuracy by correcting the wrongly predicted concepts.

In this paper, each concept is represented as a binary indicator of its presence. We consider a dataset of N triplets denoted by $\mathcal{D} = \{(\boldsymbol{x}^{(j)}, \boldsymbol{c}^{(j)}, y^{(j)})\}_{j=1}^{N}$, where j indexes the sample. For each sample, $\boldsymbol{x}^{(j)}$ is the input, $\boldsymbol{c}^{(j)} \in \{0,1\}^K$ is the concept, and $y^{(j)} \in \{0,\ldots,C-1\}$ is the label. Here, K denotes the number of concepts, and C is the number of target classes. $c_i^{(j)} = 1$ indicates that the i-th concept is present in the j-th input, while $c_i^{(j)} = 0$ indicates its absence. A *CBM* f_θ, parameterized by θ, consists of three components: input encoder f_ϕ, concept projector f_ψ, and task predictor f_ω. f_ϕ extracts information from $\boldsymbol{x}$, and outputs the vectorized representation $f_\phi(\boldsymbol{x})$. Due to the universal approximation capability of deep neural networks Lu et al. (2017); Kidger & Lyons (2020); Park et al. (2021), the sufficient encoder assumption Tian et al. (2020) posits that the encoding process preserves all information contained in $f_\phi(\boldsymbol{x})$ Zheng et al. (2024). f_ψ extracts the concept-relevant information from $f_\phi(\boldsymbol{x})$ supervised by $\boldsymbol{c}$ with the output $\hat{\boldsymbol{c}}$, the probabilities of the concepts with 1 status. The pair (f_ϕ, f_ψ) approximates $\mathcal{X} \to \mathcal{C}$ by minimizing the concept loss $\mathcal{L}_{\boldsymbol{c}} = \sum_{j=1}^{N} L_{\boldsymbol{c}}(f_\psi(f_\phi(\boldsymbol{x}^{(j)}), \boldsymbol{c}^{(j)})))$, where $L_{\boldsymbol{c}} : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}^+$ is the loss function to measure the discrepancy between the predicted and true concept.

Vanilla *CBM*s Koh et al. (2020) proposed that *CBM*s feed $\hat{\boldsymbol{c}}$ (the soft concept representations) to f_ω as Soft *CBM*s, and *CBM*s feed $\texttt{round}(\hat{\boldsymbol{c}})$ (hard concept representations) as Hard *CBM*s, where $\texttt{round}(\cdot)$ converts the probability to the $0/1$ status. f_ω approximates $\mathcal{C} \to \mathcal{Y}$ by minimizing the task loss $\mathcal{L}_{\boldsymbol{y}} = \sum_{j=1}^{N} L_{\boldsymbol{y}}(\hat{y}^{(j)}, y^{(j)})$, where $\hat{y}^{(j)}$ denotes the prediction of f_ω, and $L_{\boldsymbol{y}} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ is the loss function to measure the discrepancy between the predicted and true label. Koh et al. (2020) proposed three training strategies:

- Independent Training: (f_ϕ, f_ψ) and f_ω are independently trained, and f_ω use the true concept as its input;
- Sequential Training: (f_ϕ, f_ψ) and f_ω are sequentially trained, and f_ω use the output of f_ψ as its input;
- Joint Training: (f_ϕ, f_ψ) and f_ω are jointly trained by minimizing the weighted sum of two losses, $\mathcal{L}_{\boldsymbol{y}} + \lambda\mathcal{L}_{\boldsymbol{c}}$.

The hyperparameter λ controls the trade-off between the concept loss and the task loss.

Usually, Soft *CBM*s perform better than Hard *CBM*s on task prediction. The soft concept representation encodes more unintended information than the hard concept representation. The soft probability inherently encodes both presence and absence information (e.g., 0.9 implies a 10% chance of absence), and f_ω may exploit this additional signal. This phenomenon is known as concept leakage, which refers to the information contained within the estimated concept representation $\hat{\boldsymbol{c}}$ that is informative about $\boldsymbol{x}$ but not about $\boldsymbol{c}$ Makonnen et al. (2025). Formally, concept leakage can be measured by the mutual information of targets $\boldsymbol{y}$ and the concept representation $\hat{\boldsymbol{c}}$ given the concept $\boldsymbol{c}$ Ragkousis & Parbhoo (2024):

$$L_{Leakage} = I(y; \hat{\boldsymbol{c}}|\boldsymbol{c}) = H(y|\boldsymbol{c}) - H(y|\boldsymbol{c}, \hat{\boldsymbol{c}}), \tag{14}$$

where $I(\cdot;\cdot|\cdot)$ denotes mutual information and $H(\cdot|\cdot)$ denotes the conditional entropy. Concept leakage compromises the reliability of interpretability and intervenability. f_ω is no longer required to predict the underlying concepts for accurate label predictions precisely. Instead, its primary role becomes encoding the input information as much as possible within the soft concept probability distributions Havasi et al. (2022).

On the other hand, vanilla *CBM*s lag behind *NN*s. *CBM* variants Zarlenga et al. (2023); Kim et al. (2023); Zarlenga et al. (2023; 2025) utilized the high-dimensional concept embedding as the input of f_ω. f_ψ first projects $f_\phi(\boldsymbol{x})$ to K high-dimensional embeddings $\{\boldsymbol{c}_i\}_{i=1}^{K}$ respectively corresponding to K concepts, and then f_ψ predicts $\boldsymbol{c}$ using $\{\boldsymbol{c}_i\}_{i=1}^{K}$. Finally, the concatenated $\{\boldsymbol{c}_i\}_{i=1}^{K}$, $\texttt{cat}(\{\boldsymbol{c}_i\}_{i=1}^{K})$, becomes the input of f_ω for task prediction. Since the higher-dimensional

$\texttt{cat}(\{\boldsymbol{c}_i\}_{i=1}^K)$ can encode enriched information about $\boldsymbol{x}$, f_ω can perform better. Under this condition, $\hat{\boldsymbol{c}}$ in Equation (14) should become $\texttt{cat}(\{\boldsymbol{c}_i\}_{i=1}^K)$. Consequently, the use of high-dimensional embeddings exacerbates concept leakage, as it allows more concept-irrelevant information to be encoded. We summarize the notation in Table 5.

Symbol	Description
$\boldsymbol{x}, \boldsymbol{c}, c_i, y$	Input variable, concept variables, the i-th concept variable and target variable
$\boldsymbol{x}^{(j)}, \boldsymbol{c}^{(j)}, c_i^{(j)}, y^{(j)}$	Input, concepts, the i-th concept and target of the j-th sample
$\boldsymbol{c}_i^{s+}, \boldsymbol{c}_i^{s-}$	Learnable 1-state (active) and 0-state (inactive) static concept embedding for i-th concept
$f_{\psi_i}(f_\phi(\boldsymbol{x}))$	Dynamic concept embedding variable for i-th concept from input $\boldsymbol{x}$
$f_{\psi_i}(f_\phi(\boldsymbol{x}^{(j)}))$	Dynamic concept embedding for i-th concept from sample $\boldsymbol{x}^{(j)}$
$d_i^{(j)+}, d_i^{(j)-}$	Distance to 1-state and 0-state static embedding for i-th concept from $\boldsymbol{x}^{(j)}$
$\widehat{c_i^{(j)}}$	Predicted value of concept i for $\boldsymbol{x}^{(j)}$
$\boldsymbol{c}_i^{s\,(j)}$	Selected static concept embedding (0- or 1-state) for the i-th concept and $\boldsymbol{x}^{(j)}$
$\boldsymbol{c}^{s\,(j)}$	Concatenation of all selected static embeddings for $\boldsymbol{x}^{(j)}$
$\hat{c_i^s}^{(j)}$	$\boldsymbol{x}^{(j)}$'s injected representation of the i-th concept: dot product between $\boldsymbol{c}_i^{s\,(j)}$ and $f_{\psi_i}(f_\phi(\boldsymbol{x}^{(j)}))$
$\hat{\boldsymbol{c}^s}$	Injected representation variable of all concepts
$\hat{\boldsymbol{c}^s}^{(j)}$	Concatenation of all injected concept representations for $\boldsymbol{x}^{(j)}$

Table 5: Summary of notation in *StaticCEM* and *Benign Leakage*

A.2 Related Works

In *CBM*s, the input typically comes from the visual domain (e.g., an image), while concepts represent interpretable, often language-based, attributes. As Figure 2 shows, the information is imbalanced between both domains Schrodi et al. (2025), so the human-specific natural language hardly describes all the information in the image. Intuitively, it is hard to achieve a comparable performance with *NN*s if only the limited information of the concept is utilized Li & Pei (2023). Among *CBM* variants with competitive performance, we categorize them into two approaches: high-dimensional concept embedding and information injection. While these methods improve predictive performance, they often do so at the cost of concept leakage.

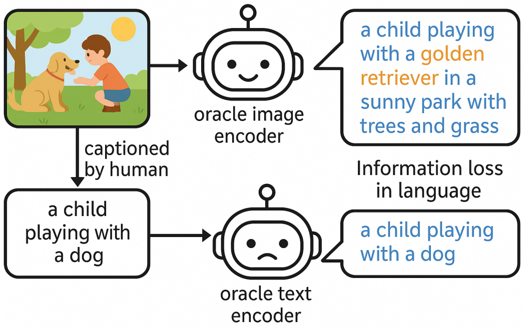
oracle image encoder
a child playing with a golden retriever in a sunny park with trees and grass
captioned by human
Information loss in language
a child playing with a dog
a child playing with a dog
oracle text encoder

Figure 2: Illustration of information imbalance between the image (top left) and natural language (bottom left). Inspired by Schrodi et al. (2025).

The high-dimensional concept embedding approach has gained popularity and is actively explored in recent works, including Zarlenga et al. (2022; 2023; 2025); Kim et al. (2023); Xu et al. (2024). The high-dimensional concept embedding approach explicitly enriches concept information by leveraging high-dimensional embeddings. The representative one is Concept Embedding Models (*CEM*) Zarlenga et al. (2022). Their f_ψ consists of the K unshared mapping pairs $\{f_{\psi_i}^+, f_{\psi_i}^-\}_{i=1}^K$, and a shared concept predictor $s(\cdot)$. $\{f_{\psi_i}^+, f_{\psi_i}^-\}_{i=1}^K$ extract K pairs of concept-state $(0/1)$ high-dimensional embeddings $\{\boldsymbol{c}_i^+, \boldsymbol{c}_i^-\}_{i=1}^K$ from $f_\theta(\boldsymbol{x})$, the i-th concept present probability p_i is predicted by $s(\texttt{cat}(\boldsymbol{c}_i^+, \boldsymbol{c}_i^-)) = p_i$, and the every concept embedding $\boldsymbol{c_i}$ is obtain by weighted mixup of two embeddings $(p_i\boldsymbol{c}_i^+ + (1-p_i)\boldsymbol{c}_i^-)$ for task prediction $f_\omega(\texttt{cat}(\{\boldsymbol{c}_i\}_{i=1}^K))$. Probabilistic Concept

Bottleneck Models (*ProbCBM*s) Kim et al. (2023) utilize the parameterized distributions Kingma & Welling (2019) to sample the concept embedding and estimate uncertainty of the final prediction. Beyond Concept Bottleneck Models (*BeyondCBM*s) Laguna et al. (2024) train the edited representation under the supervision of the concept and the label to implicitly inject the concept effect into $f_\phi(\boldsymbol{x})$. Overall, their high-dimensional concept embeddings are dynamic, as they are extracted from $\boldsymbol{x}$ using a neural architecture. These embeddings contain much information about $\boldsymbol{x}$, and pass this information to f_ω for the final prediction, which is much like *NN*s. Therefore, we conjecture that the static concept embeddings, which do not change with different $x^{(j)}$, should benefit the leakage resistance by not encoding input-specific details, unlike dynamic embeddings. Furthermore, high-dimensional concept embeddings complicate interpretability, as it becomes challenging to quantify the concept contribution to the final prediction.

Information injection approach preserves the concept representation $\hat{\boldsymbol{c}}$ of the vanilla *CBM*, so available concepts can partially interpret the final prediction. Sawada & Nakamura (2022) and Havasi et al. (2022) use f_ψ to extract a concept-based representation $\hat{\boldsymbol{c}} \in \mathbb{R}^{K+D}$ from $f_\phi(\boldsymbol{x})$, where D denotes the dimensionality of the unsupervised concepts. Yuksekgonul et al. (2022) apply a neural architecture $r(\cdot)$ using $f_\phi(\boldsymbol{x})$ to fit the residual of the concept-based prediction, i.e., $\boldsymbol{y} - f_\omega(\hat{\boldsymbol{c}})$. Although the leakage origin is known, the injected information is usually not limited or controlled. Under these circumstances, it is apparent that the injected information would likely dominate the final prediction, as a direct path exists from the input to the target through the state of the injected information within the neural architecture. Consequently, we inject the information into the concept representation to enhance the static concept embedding, rather than creating a separate path for the unknown concept and prediction residual. Moreover, since the known leakage origin is typically neural embeddings, we apply the information bottleneck Tishby et al. (2000) principle to regulate these sources, thereby limiting the encoding of excessive information beyond the concept.

A.3 Proof of Theorem 1

This section demonstrates the proof of Theorem 1 as the following shows:

Proof 1 *As Havasi et al. (2022) proved that Hard CBMs are leakage-resistant, we prove Theorem 1 by the equivalence between the static embedding and 0/1 embedding. For any sample $x^{\{j\}}$ of the i-th concept, the Gumbel-Softmax without learnable parameters constructs a bijection between the distance comparison and the concept state ($d_i^{(j)+} \geq d_i^{(j)-} \leftrightarrow 1$ and $d_i^{(j)+} < d_i^{(j)-} \leftrightarrow 0$). Since StaticCEM and Hard CBM will be trained until convergence, we can regard both learn the identical mapping $f_{\psi_i}(f_\phi(\boldsymbol{x})) \rightarrow \{0,1\}$. With the same dynamic concept embedding, the learned codebook decides the concept compared to Hard CBMs. Therefore, the 1 or 0 -state embedding in StaticCEMs deterministically corresponds to 1 or 0 in Hard CBMs.*

A.4 Non-parametric Kernel Matrix-based Estimator

Measuring the conditional mutual information is alternatively the independence between the label $\boldsymbol{y}$ and the concept embedding $\texttt{round}(\hat{\boldsymbol{c}})$ or $\vec{\boldsymbol{c}^s}$ conditioning on $\boldsymbol{c}$. If $\boldsymbol{x}$ is independent on $\boldsymbol{y}$ conditionally on $\boldsymbol{z}$, we have $p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{y}|\boldsymbol{z})$. From this equation, we have $p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})p(\boldsymbol{z}) = p(\boldsymbol{x},\boldsymbol{z})p(\boldsymbol{y},\boldsymbol{z})$. Hence, the measure of the conditional independence via CS divergence is by

$$
\begin{aligned}
&D_{\mathrm{CS}}(p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})p(\boldsymbol{z});\ p(\boldsymbol{x},\boldsymbol{z})p(\boldsymbol{y},\boldsymbol{z})) && (15)\\
&\quad = -2\log\left(\int p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})\,p(\boldsymbol{z})\,p(\boldsymbol{x},\boldsymbol{z})\,p(\boldsymbol{y},\boldsymbol{z})\,d\boldsymbol{x}\,d\boldsymbol{y}\,d\boldsymbol{z}\right)\\
&\qquad + \log\left(\int p^2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})\,p^2(\boldsymbol{z})\,d\boldsymbol{x}\,d\boldsymbol{y}\,d\boldsymbol{z}\right)\\
&\qquad + \log\left(\int p^2(\boldsymbol{x},\boldsymbol{z})\,p^2(\boldsymbol{y},\boldsymbol{z})\,d\boldsymbol{x}\,d\boldsymbol{y}\,d\boldsymbol{z}\right) && (16)
\end{aligned}
$$

Proposition 1 *Yu et al. (2025) Given N samples drawn from an unknown and fixed joint distribution $p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$, where $\boldsymbol{x}x \in \mathbb{R}^{d_{\boldsymbol{x}}}$, $\boldsymbol{y} \in \mathbb{R}^{d_{\boldsymbol{y}}}$, and $\boldsymbol{z} \in \mathbb{R}^{d_{\boldsymbol{z}}}$. Let $K \in \mathbb{R}^{N\times N}$ be the Gram (kernel) matrix* *for variable $\boldsymbol{x}$, with entries defined as*

$$K_{ji} = \exp\left(-\frac{\|x^{(j)} - x^{(i)}\|_2^2}{2\sigma_{\boldsymbol{x}}^2}\right),$$

where $\sigma_{\boldsymbol{x}}$ is the kernel width. Similarly, let $L \in \mathbb{R}^{N\times N}$ and $M \in \mathbb{R}^{N\times N}$ be the Gram matrices for variables $\boldsymbol{y}$ and $\boldsymbol{z}$, respectively. Then, the empirical estimator of $D_{CS}\left(p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})p(\boldsymbol{z})\,;\,p(\boldsymbol{x},\boldsymbol{z})p(\boldsymbol{y},\boldsymbol{z})\right)$ is given by:

$$\begin{aligned}
&\widehat{D}_{CS}((p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})p(\boldsymbol{z})\,;\,p(\boldsymbol{x},\boldsymbol{z})p(\boldsymbol{y},\boldsymbol{z}))) = \\
&-2\log\left(\sum_{j=1}^{N}\left(\left(\sum_{i=1}^{N}M_{ji}\right)\left(\sum_{i=1}^{N}K_{ji}M_{ji}\right)\left(\sum_{i=1}^{N}L_{ji}M_{ji}\right)\right)\right) \\
&+\log\left(\sum_{j=1}^{N}\left(\left(\sum_{i=1}^{N}K_{ji}L_{ji}M_{ji}\right)\left(\sum_{i=1}^{N}M_{ji}\right)^2\right)\right) \\
&+\log\left(\sum_{j=1}^{N}\left(\frac{(\sum_{i=1}^{N}K_{ji}M_{ji})^2(\sum_{i=1}^{N}L_{ji}M_{ji})^2}{\sum_{i=1}^{N}K_{ji}L_{ji}M_{ji}}\right)\right) \qquad (17)
\end{aligned}$$

A.5 Parametric Predictor-based Estimator

Makonnen et al. (2025) proposed estimating the conditional entropy terms $H(\boldsymbol{y}|\boldsymbol{c},\texttt{round}(\hat{\boldsymbol{c}}))$ and $H(\boldsymbol{y}|\boldsymbol{c},\vec{\boldsymbol{c}^s})$ is straightforwardlly by:

$$\begin{aligned}
H(\boldsymbol{y}|\boldsymbol{c},\texttt{round}(\hat{\boldsymbol{c}})) &= \mathbb{E}[-\log p(\boldsymbol{y}|\boldsymbol{c},\texttt{round}(\hat{\boldsymbol{c}}))] \\
&\approx -\frac{1}{N}\sum_{j=1}^{N}\log\, g_{\alpha_1}\left(\texttt{round}(\hat{c}^{(j)}),c^{(j)}\right)_{y^{(j)}},
\end{aligned} \qquad (18)$$

$$\begin{aligned}
H(\boldsymbol{y}|\boldsymbol{c},\vec{\boldsymbol{c}^s}) &= \mathbb{E}[-\log p(\boldsymbol{y}|\boldsymbol{c},\vec{\boldsymbol{c}^s})] \\
&\approx -\frac{1}{N}\sum_{i=1}^{N}\log\, g_{\alpha_2}\left((\vec{c^s}^{(j)},c_i\right)_{y_i},
\end{aligned} \qquad (19)$$

where g_{α_1} and g_{α_2} are classifiers trained to predict y from $(\boldsymbol{c},\texttt{round}(\hat{\boldsymbol{c}}))$ and $(\boldsymbol{c},\vec{\boldsymbol{c}^s})$, respectively. As Makonnen et al. (2025) experimentally concluded, *XGBoost* Chen & Guestrin (2016) is the most reliable classifier for the leakage measure, and remains stable, aligns well with expectations, and rarely produces negative leakage estimates. Therefore, in our experiment, we implement *XGBoost* as our classifier.

A.6 Experimental Result on Leakage-Resistance

This section uses Table 6 to validate that the *StaticCEM* is leakage-resistant compared to the Hard *CBM*.

Dataset	XOR				Dot				Trigonometry			
Metric	Acc(%)		*CLM*		Acc(%)		*CLM*		Acc(%)		*CLM*	
	Task	Concept	Predictor	Matrix	Task	Concept	Predictor	Matrix	Task	Concept	Predictpr	Matrix
Hard *CBM*	74.17	99.17	-	-	64.83	99.67	-	-	82.33	99.33	-	-
StaticCEM	73.17	99.58	**-0.0046**	**0.0183**	60.67	98.42	**-0.0011**	**-0.0583**	84.33	97.72	**0.0258**	**-0.6794**
CEM	99.33	99.67	0.0147	0.0205	97.33	99.00	0.5019	0.3468	98.00	99.39	0.3608	0.0994

Table 6: Evaluation of *StaticCEM* and *CEM*s across datasets among the task accuracy, mean concept accuracy, and *CLM*.

A.7 Cauchy-Schwarz Quadratic Mutual Information

Cauchy-Schwarz Quadratic Mutual Information Yu et al. (2025) is defined based on Cauchy-Schwarz divergence (CS divergence) Rudin (2019):

$$D_{CS}(p;q) = -\log\left(\frac{\left(\int p(\boldsymbol{x})q(\boldsymbol{x})\,d\boldsymbol{x}\right)^2}{\left(\int p(\boldsymbol{x})^2\,d\boldsymbol{x}\right)\left(\int q(\boldsymbol{x})^2\,d\boldsymbol{x}\right)}\right), \tag{20}$$

The CS divergence is symmetric for any two probability density functions (PDFs) p and q, such that $0 \leq D_{CS} \leq \infty$, where the minimum is obtained iff $p(\boldsymbol{x}) = q(\boldsymbol{x})$. The CS divergence is a measure of the “distance” between $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$. The independence between $\boldsymbol{x}$ and $\boldsymbol{y}$ can be measured by any (valid) distance or divergence measure over the joint distribution $p(\boldsymbol{x},\boldsymbol{y})$ with respect to the product of marginal distributions $p(\boldsymbol{x})p(\boldsymbol{y})$. We obtain the Cauchy-Schwarz Quadratic Mutual Information (CS-QMI) by substituting $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ in Equation (20) with $p(\boldsymbol{x},\boldsymbol{y})$ and $p(\boldsymbol{x})p(\boldsymbol{y})$:

$$\begin{aligned} I_{CS}(\boldsymbol{x},\boldsymbol{y}) &= D_{\mathrm{CS}}(p(\boldsymbol{x},\boldsymbol{y}) \,\|\, p(\boldsymbol{x})p(\boldsymbol{y})) \\ &= -\log\left(\frac{\left|\int p(\boldsymbol{x},\boldsymbol{y})\,p(\boldsymbol{x})\,p(\boldsymbol{y})\,d\boldsymbol{x}\,d\boldsymbol{y}\right|^2}{\int p^2(\boldsymbol{x},\boldsymbol{y})\,d\boldsymbol{x}\,d\boldsymbol{y}\int p^2(\boldsymbol{x})\,p^2(\boldsymbol{y})\,d\boldsymbol{x}\,d\boldsymbol{y}}\right). \end{aligned} \tag{21}$$

Distinct from KL divergence Kullback & Leibler (1951), which is notoriously hard to estimate. CS-QMI and the divergence can be elegantly estimated in a non-parametric way with closed-form expressions, enabling efficient implementation of deep IB without approximations.

Proposition 2 (Empirical Estimator of CS-QMI Yu et al. (2025)) *Given N pairs of samples $\{(x^{(i)}, y^{(i)}\}_{i=1}^{N}$, each sample contains two different types of measurements $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \mathcal{Y}$ obtained from the same sample. Let K and Q denote the Gram matrices for variable $\boldsymbol{x}$ and variable $\boldsymbol{y}$, respectively, which are symmetric. The empirical estimator of CS-QMI is given by:*

$$\begin{aligned} I_{\mathrm{CS}}(\boldsymbol{x};\boldsymbol{y}) = \log&\left(\frac{1}{N^2}\sum_{i,j=1}^{N} K_{ij}Q_{ij}\right) \\ &+ \log\left(\frac{1}{N^4}\sum_{i,j,q,r=1}^{N} K_{ij}Q_{qr}\right) - 2\log\left(\frac{1}{N^3}\sum_{i,j,q=1}^{N} K_{ij}Q_{iq}\right) \\ = \log&\left(\frac{1}{N^2}\mathrm{tr}(KQ)\right) + \log\left(\frac{1}{N^4}\mathbf{1}^\top K\mathbf{1}\cdot\mathbf{1}^\top Q\mathbf{1}\right) \\ &- 2\log\left(\frac{1}{N^3}\mathbf{1}^\top KQ\mathbf{1}\right), \end{aligned} \tag{22}$$

where $\mathbf{1}$ is is a $N \times 1$ vector of ones. The second line of Equation (22)reduces the complexity to $\mathcal{O}(N^2)$.

A.8 Dataset

MNIST We create more challenging *CBM* task using the `MNIST` dataset Lecun et al. (1998) than *ProbCBM* Kim et al. (2023). We also combine four digit images into a single image as the input, using digit labels as concepts. However, the combination order of four images is completely random. Table 7 shows all combinations for the original and shifted domains. We generate 2000 samples for every combination, so there are 12000 samples for the original and shifted domains. In our experiments, the model is trained on the original domain and tested on the shifted, allowing us to verify that it adheres to the predictability of benign leakage.

Chest X-ray We consider two datasets: `CheXpert`[1] Irvin et al. (2019), and `MIMIC-CXR`[2] Johnson et al. (2019). Both datasets have the same 14 labels. We select *No Finding* as the task label, and

[1] In our training, we only use the small dataset of `CheXpert` in https://www.kaggle.com/datasets/ashery/chexpert.

[2] In our test, we use the subset of `MIMIC-CXR`, the folder `p10`.

Original Domain	Even Count	Shifted Domain	Even Count
(0, 1, 2, 3)	2	(0, 2, 5, 9)	2
(4, 5, 6, 7)	2	(3, 6, 7, 8)	2
(8, 1, 3, 5)	1	(1, 2, 4, 7)	2
(0, 2, 4, 6)	4	(0, 3, 6, 8)	3
(1, 3, 5, 7)	0	(1, 4, 5, 8)	2
(2, 5, 6, 9)	2	(2, 4, 7, 9)	2

Table 7: Original and shifted digit combinations with their respective even digit counts.

the remaining 13 labels as concepts. Similar to MNIST, we train models on CheXpert and test them on MIMIC-CXR to validate their predictability. In the preprocessing, we replace -1 and the missing value with 0.

CUB Caltech-UCSD Birds-200-2011 (CUB) dataset Wah et al. (2011) is tailored for concept learning models. After preprocessing, CUB contains 11,788 images from 200 bird species classes with 112 concepts.

A.9 Experimental Setup

We select *ProbCBM* as the baseline on MNIST, *BeyondCBM* on CheXpert and MIMIC-CXR, and *CEM* on CUB. We select *CEM* for CUB due to its state-of-the-art performance, while *ProbCBM* and *BeyondCEM* are included as they would uniquely report results on their respective datasets. We split the train set into the train set and the validation set with the proportions 90% and 10%, and transfer *Resnet50* He et al. (2015) as f_ϕ for all models with ϕ frozen. We use the prototype on MNIST, and one fully-connected layer on the rest. We only fine-tune the parameters of ψ and ω. We set λ as 1.0 for MNIST and 5.0 for CUB and CheXpert, and both β_{CIB} and β_{LIB} as 0.25 for all datasets. We use AdamW Loshchilov & Hutter (2019) with a learning rate of $1e-3$ and weight decay $4e-5$.

In our experiments, we do not compare our approach to *NN*s, because the selected baselines have experimentally proved the matching performance compared to NNs. Furthermore, *BeyondCEM* is still in end-to-end mode, such that it can represent the end-to-end *NN*s. Furthermore, we will not investigate the interpretation, because after *Benign Leakage* (BL), the concept representation is scalar and pairwise to the concept. Under this condition, we can easily compute the contribution of each concept for the task, like vanilla *CBM*s.

A.10 An Intervention Example

Figure 3 is an example of MNIST. This example shows that although benign leakage happens, the intervention would work.

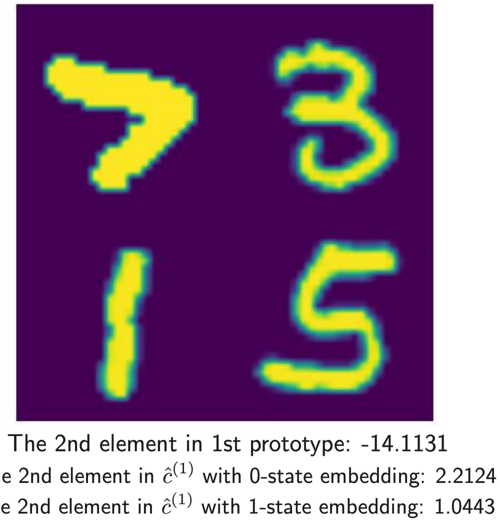
The 2nd element in 1st prototype: -14.1131
The 2nd element in $\hat{c}^{(1)}$ with 0-state embedding: 2.2124
The 2nd element in $\hat{c}^{(1)}$ with 1-state embedding: 1.0443

Figure 3: The first sample of MNIST Original dataset with the interventable effect on the 2nd concept.

A.11 Ablation Study

In Table 8, we conduct ablation studies to evaluate the effect of β_{CIB} and β_{LIB} on `MNIST`. When $\beta_{CIB} = 0$, more information unrelated to the concept will be injected into $\hat{\boldsymbol{c}}^s$. Therefore, the final prediction depends less on the concept, such that the concept accuracy becomes less priority. This reflects the low concept accuracy as shown in Table 8. When $\beta_{LIB} = 0$, the redundant information will be injected into $\hat{\boldsymbol{c}}^s$. Under these conditions, the task performance will decay to some extent. Overall, β_{CIB} and β_{LIB} play desirable roles, and both hyperparameters are necessary.

β_{CIB}	β_{LIB}	**Task Acc (%)**	**Concept Acc (%)**
×	×	87.92	93.06
×	✓	88.42	92.60
✓	×	76.67	92.88
✓	✓	89.67	93.42

Table 8: Ablation Study of β_{CIB} and β_{LIB} on `MNIST`.