Objective:

Construct a synthetic dataset with artificially introduced leakage. This dataset enables testing the proposed leakage estimation measure, as the leakage amount in the data is precisely controlled.

Leakage:

Briefly recall that leakage occurs when additional information from the features—not encoded in the ground truth concepts but useful for predicting the target—is contained in the estimated concepts. Note that from here on, any mention of "information" refers to information useful for predicting the target.

High level overview synthetic data generation:

The problem setting is multiclass classification within the framework of concept bottleneck models, meaning that the synthetic dataset consists of features, ground truth concepts, estimated concepts, and targets.

The ground truth concepts are generated as a function of the features, establishing a realistic link between features and concepts, as in real-life settings where features naturally inform concept values. However, only a subset of feature components is used in constructing these ground truth concepts. To construct the estimated concepts, this subset is combined with additional, unused feature information. This ensures that the estimated concepts contain both the information in the ground truth concepts and extra predictive information. In this way, leakage is introduced, as the estimated concepts now include target-relevant information not found in the ground truth concepts.

Lastly, the targets are generated as a function of both the ground truth concepts and the additional information (leakage) in the estimated concepts. This setup ensures that the model leverages both the intended concept-based pathway and the leakage in the estimated concepts for predicting the target. In this way, the target still implicitly depends on the features, but this approach allows for greater control over the flow of information.

Mathematical overview synthetic data generation:

$i = 1, ..., n$ observations
$j = 1, ..., k$ concepts

First, draw the **features** $\boldsymbol{x}_i \overset{iid}{\sim} \boldsymbol{X} \in \mathbb{R}^d$ (note that $\mathbb{R}^d$ here refers to $\mathbb{R}^{d\times 1}$, i.e. default are column vectors) where

$$\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x), \quad \boldsymbol{\mu}_x \in \mathbb{R}^d \text{ and } \boldsymbol{\Sigma}_x \in \mathbb{R}^{d\times d}. \tag{1}$$

Next, the **binary ground truth concept vector** $\boldsymbol{c}_i \in \{0, 1\}^k$ is constructed by sampling each concept from a Bernoulli distribution. Sampling from a Bernoulli distribution, rather than e.g. directly applying a threshold, captures the inherent uncertainty and noise in the relationship between features and concepts. For each observation, a vector of success probabilities $\boldsymbol{\pi}_i \in \mathbb{R}^k$, one for each concept, is computed using a function of a subset of the feature information. This approach ensures that the features inform the ground truth concepts while not utilizing all their information, allowing the remaining information to be used later for modeling leakage. Specifically

$$c_{ij} \sim \text{Bernoulli}(\pi_{ij}), \tag{2}$$

$$\boldsymbol{\pi}_i = \sigma(\boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{\epsilon}_c), \tag{3}$$

$$\boldsymbol{\epsilon}_c \sim \mathcal{N}(0, \boldsymbol{\Sigma}_c), \quad \text{where } \boldsymbol{\epsilon}_c \in \mathbb{R}^k \text{ and } \boldsymbol{\Sigma}_c \in \mathbb{R}^{k\times k}. \tag{4}$$

Here, σ denotes the sigmoid activation function, applied element-wise to map the logits to the $[0, 1]$ range. Next, it is important to explain how only a subset of the feature information is used in constructing the success probabilities for the ground truth concepts. This is achieved through the matrix $\mathbf{A} \in \mathbb{R}^{k\times d}$.

The matrix $\mathbf{A}$ is designed to perform a random projection of the first b elements of the feature vector $\boldsymbol{x}_i$ into the k-dimensional concept space. By doing so, the information flow from the features to the concepts is intentionally restricted to only the first b features.

For this specifically, a random projection—as opposed to another type of projection—is employed. This is done to emulate the potentially black-box nature in which concept embeddings are generated by recombining input features in Concept Bottleneck Models. Importantly, the use of a random projection here preserves the relative geometry between observations with high probability, as described by the Johnson-Lindenstrauss lemma. A key point is that this approach requires $k < b$, meaning the number of features being projected (b) must exceed the dimensionality of the concept embedding (k).

In detail, the matrix $\mathbf{A}$ is constructed as:

$$\mathbf{A} = \left[\mathbf{R}_A \;\middle|\; \mathbf{0}_{k\times(d-b)} \right]_{k\times d} \tag{5}$$

where $\mathbf{R}_A \in \mathbb{R}^{k\times b}$ is a random projection matrix, and $\mathbf{0}_{k\times(d-b)}$ is a zero matrix ensuring that the remaining $d-b$ elements of the feature vector (the elements beyond the first b being projected) do not contribute to the concept generation. Here, the entries of $\mathbf{R}_A$ are sampled independently from a standard normal distribution, as is common with random projections, i.e.,

$$(\mathbf{R}_A)_{jp} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \quad \text{for } p = 1,\ldots,b. \tag{6}$$

Alternatively, $\mathbf{A}$ can be visualized as

$$\mathbf{A} = \begin{bmatrix} (R_A)_{11} & (R_A)_{12} & \ldots & (R_A)_{1b} & 0 & \ldots & 0 \\ (R_A)_{21} & (R_A)_{22} & \ldots & (R_A)_{2b} & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ (R_A)_{k1} & (R_A)_{k2} & \ldots & (R_A)_{kb} & 0 & \ldots & 0 \end{bmatrix}_{k\times d} . \tag{7}$$

This construction ensures that only the first b features of $\boldsymbol{x}_i$ are used in computing the success probabilities $\boldsymbol{\pi}_i$, while the remaining features are excluded. By restricting the feature set in this manner, the model allows the remaining information in $\boldsymbol{x}_i$ to be used later for modeling leakage.

Proceeding the **estimated concept vector** $\hat{\boldsymbol{c}}_i \in [0,1]^k$ is constructed, with values constrained to the $[0,1]$ range, as is standard in the soft concept bottleneck model setting. While termed "estimated concepts," they are not actually estimated but instead constructed in this synthetic data setting to retain control over the degree of leakage.

The estimated concept vector $\hat{\boldsymbol{c}}_i$ is computed as

$$\hat{\boldsymbol{c}}_i = \sigma\left(\mathbf{A}\boldsymbol{x}_i + \boldsymbol{l}_i + \boldsymbol{\epsilon}_{\hat{c}}\right), \tag{8}$$

where

- σ is the sigmoid activation function, applied element-wise to map logits to the $[0,1]$ range
- $\boldsymbol{\epsilon}_{\hat{c}} \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}_{\hat{c}}\right)$ introduces noise to model uncertainty, with $\boldsymbol{\epsilon}_{\hat{c}} \in \mathbb{R}^k$ and $\boldsymbol{\Sigma}_{\hat{c}} \in \mathbb{R}^{k\times k}$
- $\boldsymbol{l}_i \in \mathbb{R}^k$ represents the leakage term

The leakage term $\boldsymbol{l}_i$ is defined as:

$$\boldsymbol{l}_i = \mathbf{B}\boldsymbol{x}_i, \tag{9}$$

Here, $\mathbf{B} \in \mathbb{R}^{k\times d}$ is constructed to project specific elements of the feature vector $\boldsymbol{x}_i$ into the concept space, introducing additional information not present in the ground truth concepts. Like the generation of ground truth concepts, $\mathbf{B}$ uses a random projection to map features into the concept space. However, $\mathbf{B}$ specifically projects the elements of $\boldsymbol{x}_i$ from positions $b+1$ to $d-l$, effectively using the remaining $d-b$ features not used in $\mathbf{A}$ while excluding the last l features. Excluding the last l features provides fine-grained control over how much remaining feature information contributes to leakage, avoiding a direct complement relationship where less information in the ground truth concepts implies more in the leakage. This allows for settings with little information in both ground truth and estimated concepts, supporting a more robust assessment of the leakage measure.

Precisely, $\mathbf{B}$ is constructed as

$$\mathbf{B} = \left[\mathbf{0}_{k\times b} \;\middle|\; \mathbf{R}_B \;\middle|\; \mathbf{0}_{k\times l} \right]_{k\times d}, \tag{10}$$

where $\mathbf{0}_{k\times b}$ and $\mathbf{0}_{k\times l}$ are zero matrices, ensuring that the first b and last l elements of the feature vector are excluded. $\mathbf{R}_B \in \mathbb{R}^{k\times(d-b-l)}$ is a random projection matrix, with entries sampled according to

$$(\mathbf{R}_B)_{jq} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \quad \text{for } q = 1,\ldots,d-b-l. \tag{11}$$

To preserve the relative geometry between observations with high probability, as per the Johnson-Lindenstrauss lemma, it is necessary that $k < d-b-l$.

Alternatively, $\mathbf{B}$ can be visualized as

$$\mathbf{B} = \begin{bmatrix} 0 & \ldots & 0 & (R_B)_{11} & \ldots & (R_B)_{1,\,d-b-l} & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & (R_B)_{k1} & \ldots & (R_B)_{k,\,d-b-l} & 0 & \ldots & 0 \end{bmatrix}_{k\times d}. \tag{12}$$

Lastly, the **target variable** $\boldsymbol{y}_i \in \{1,\ldots,J\}$ is constructed. Operating in a multiclass setting and to introduce randomness (based on the prior argument for sampling from a Bernoulli rather than thresholding), the target y_i is sampled from a categorical distribution defined by the probability vector $\boldsymbol{p}_i$

$$y_i \sim \text{Categorical}\left(\boldsymbol{p}_i\right), \tag{13}$$

where $\boldsymbol{p}_i \in \mathbb{R}^J$ is computed using a nonlinear function $f : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}^J$ that combines the ground truth concepts $\boldsymbol{c}_i$ and the leakage term $\boldsymbol{l}_i$

$$\boldsymbol{p}_i = \text{softmax}\left(f\left(\boldsymbol{c}_i, \boldsymbol{l}_i\right) + \boldsymbol{\epsilon}_y\right). \tag{14}$$

Here, $\boldsymbol{\epsilon}_y \sim \mathcal{N}(0, \boldsymbol{\Sigma}_y)$ is a noise vector in $\mathbb{R}^J$ that introduces randomness into the target probabilities, with $\boldsymbol{\Sigma}_y \in \mathbb{R}^{J\times J}$.

By constructing the target labels y_i with this nonlinear function that integrates both ground truth concepts and leakage information, we ensure that the ground truth concepts are informative for predicting the target, while the leakage provides additional information to improve prediction accuracy. Note that this setup ensures the target implicitly depends on the original features through both the ground truth concepts and the leakage term.

To conclude, there are two avenues to control leakage:

- Via b: By choosing the number of elements from the feature vector that enter as information into the ground truth concepts, and therefore do not contribute to the leakage term
- Via l: This parameter provides a finer control over how much of the remaining information from the feature vector contributes to the leakage.

Given the constraints imposed by the random projections, namely $k < b$ and $k < d-b-l$, and combining these with $b < d$ (since b must be a subset of d), we have:

$$k < b < d - k - l \tag{15}$$

where $b, d, k, l \in \mathbb{N}$. This summarizes the necessary constraints that need to be accounted for.