ADAPTIVE CONCEPT BOTTLENECK FOR FOUNDATION MODELS UNDER DISTRIBUTION SHIFTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Advancements in foundation models (FMs) have led to a paradigm shift in machine learning. The rich, expressive feature representations from these pre-trained, large-scale FMs are leveraged for multiple downstream tasks, usually via lightweight fine-tuning of a shallow fully-connected network following the representation. However, the non-interpretable, black-box nature of this prediction pipeline can be a challenge, especially in critical domains, such as healthcare, finance, and security. In this paper, we explore the potential of Concept Bottleneck Models (CBMs) for transforming complex, non-interpretable foundation models into interpretable decision-making pipelines using high-level concept vectors. Specifically, we focus on the test-time deployment of such an interpretable CBM pipeline "in the wild", where the distribution of inputs often shifts from the original training distribution. We first identify the potential failure modes of such pipelines under different types of distribution shifts. Then we propose an *adaptive concept bottleneck* framework to address these failure modes, that dynamically adapts the concept-vector bank and the prediction layer based solely on unlabeled data from the target domain, without access to the source dataset. Empirical evaluations with various real-world distribution shifts show our framework produces concept-based interpretations better aligned with the test data and boosts post-deployment accuracy by up to 28%, aligning CBM performance with that of non-interpretable classification.

1 INTRODUCTION

Foundation Models (FMs), trained on vast data, are powerful feature extractors applicable across diverse distributions and downstream tasks (Bommasani et al., 2021; Rombach et al., 2022). They can be applied to classification tasks off-the-shelf via zero-shot prediction, or via linear probing using task-specific fine-tuning data (Kumar et al., 2022; Radford et al., 2021). Despite these strong advantages, foundation model-based systems often operate as inscrutable black-boxes, presenting a barrier to user trust and wider deployment in safety-critical settings. Another challenge faced in the standard deployment of FM-based deep classifiers is their vulnerability to distribution shifts at test time caused *e.g.*, due to environmental changes, which can cause a drop in performance (Bommasani et al., 2021). This is particularly challenging in high-stakes domains such as healthcare (AlBadawy et al., 2018; Eslami et al., 2023), autonomous driving (Yu et al., 2020), and finance (Wu et al., 2023a).

In this work, we address these challenges by developing an *interpretable classification* framework that enjoys the rich, expressive feature representations of FMs, while also having enhanced robustness towards *distribution shifts at test time*. To tackle interpretability, we utilize *Concept Bottleneck Models (CBMs)* (Koh et al., 2020), transforming FM-based classifiers into interpretable, concept-based prediction pipelines. With the rapid advancements in FMs, there is strong opportunity to utilize them as powerful backbones, providing robust feature representations from which high-quality concepts can be extracted. Unlike early CBM approaches that required expensive concept annotations, recent advances show potential for constructing concept bottlenecks without any annotations by leveraging vision-language models (Oikarinen et al., 2023; Wu et al., 2023b), and achieving performance on par with non-interpretable models. Concept-based predictions provide not only interpretability, but are also beneficial for *robustness*; a central premise of CBMs is that as complex feature embeddings go through the concept bottleneck, the resulting predictions should, in theory, become more invariant to inconsequential input changes (Kim et al., 2018; Adebayo et al., 2020).

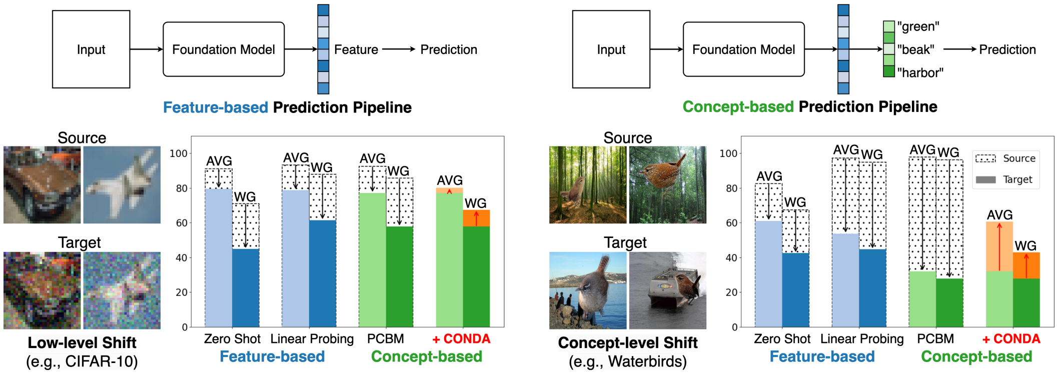


Figure 1: **Concept-based predictions are not inherently more robust to distribution shifts than feature-based predictions, necessitating dynamic adaptation after deployment.** We observe significant drops in the averaged group accuracy (AVG) and worst-group accuracy (WG) from the source to the target (test) domain under two types of distribution shifts: (1) *low-level shift* (left), where inputs are perturbed without modifying class-level semantics (*e.g.*, Gaussian noise); and (2) *concept-level shift* (right), where some high-level semantics change. On the left, predictions made through high-level concepts (*e.g.*, by PCBM (Yuksekgonul et al., 2023) here) are not necessarily more robust to low-level input perturbations. On the right, the performance of concept-based predictions suffers an even more drastic drop, failing to leverage the expressiveness of the foundation model's high-level features, and falling behind direct feature-based predictions (here zero-shot and linear-probing). However, with CONDA (ours), we can boost the performance of the deployed concept-based predictor to be on par with, or even better than, its non-interpretable counterparts.

However, we observe that CBMs directly deployed under distribution shifts often do not produce more robust predictions compared to FM-based classifiers (either in zero-shot or fine-tuned configurations). For instance, as illustrated in Figure 1, even when a concept-based prediction pipeline matches or outperforms a feature-based prediction pipeline in the training (source) domain, its test-time (deployment) performance can drop as severely, or even more, under distribution shifts. This highlights that a naive adoption of CBMs is insufficient for fully leveraging the robustness and expressiveness of FM features under test-time shifts, necessitating a dynamic approach for adapting concept-based predictions in real-world deployments.

The problem of test-time (or source-free domain) adaptation (TTA) has recently been explored extensively (Wang et al., 2021; Jung et al., 2023; Liang et al., 2023). The goal is to adapt a deep classifier, trained on a source domain, to a test-time deployment setting where there could be distribution shifts (*e.g.*, corruptions, environment changes), and given access to *only* unlabeled test data and the source domain classifier. While the main focus of TTA methods has been on non-interpretable, deep classifier networks, to our knowledge we present the first approach for *TTA of concept bottlenecks* with a foundation model backbone. Our contributions are as follows: given unlabeled test data, a frozen FM, and a pre-constructed concept bottleneck, we

1. formally categorize the types of distribution shifts expected post-deployment, identifying possible failure modes of the concept bottleneck pipeline under these shifts (Section 2);
2. propose a novel framework, CONDA (CONcept-based Dynamic Adaptation), where each component of the framework is adapted based on the identified failure modes, without requiring access to the source dataset or labels for the test dataset (Section 3);
3. empirically demonstrate the robustness and interpretability of CONDA across various FM backbones (*e.g.*, CLIP:ViT-L/14) and concept bottleneck construction methods (*e.g.*, post-hoc CBM), showing that CONDA improves the test-time accuracy by up to 28%, and provides concept-based interpretations better tailored towards test inputs (Section 4).

Related Work. Distribution shifts occur when the data distribution during deployment differs from that during training, leading to degraded model performance (Quiñonero-Candela et al., 2022). To address this issue, TTA methods adapt model parameters using unlabeled test data to enhance the robustness under such shifts. Representative methods include entropy minimization (Wang et al., 2021; Zhang et al., 2022), self-supervised learning at test time (Sun et al., 2020), class-aware feature alignment (Jung et al., 2023), and updating batch normalization statistics based on test data (Nado et al., 2020). These methods enable models to adapt on-the-fly without requiring access to the training data. In the era of foundation models, recent efforts have been made to enhance their zero-shot inference robustness under distribution shifts without modifying their internal parameters (Chuang et al., 2023; Adila et al., 2023). However, improving the robustness of the foundation model itself is not the focus of our work. Instead, given *any* foundation model, regardless of its inherent robustness, we aim to construct an interpretable framework without sacrificing the utility, striving for performance that matches or exceeds that of the foundation model's feature-based predictions.

2 Concept Bottleck Model under Distribution Shifts

2.1 Background: Foundation Models with a Concept Bottleneck

Consider a foundation model $\boldsymbol{\phi} : \mathcal{X} \mapsto \mathbb{R}^d$, which is any pre-trained backbone model or feature extractor (Eslami et al., 2023; Jia et al., 2021; Girdhar et al., 2023) that maps the input $\mathbf{x}$ to an intermediate feature embedding $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^d$. $\boldsymbol{\phi}(\mathbf{x})$ is pre-trained on a large-scale, broad mixture of data for general purposes, *i.e.*, not restricted to a specific domain. For a specific downstream classification task, the general practice is to either apply zero-shot prediction on $\boldsymbol{\phi}(\mathbf{x})$, or to train a shallow label predictor $\mathbf{g}_s : \mathbb{R}^d \mapsto \mathbb{R}^L$, that maps $\boldsymbol{\phi}(\mathbf{x})$ to the un-normalized class predictions $\mathbf{g}_s(\boldsymbol{\phi}(\mathbf{x}))$, using a supervised loss (*e.g.*, cross-entropy).

A CBM (Koh et al., 2020) first projects the high-dimensional feature embedding to a lower m-dimensional ($m \ll d$) *concept-score space* (acting like a bottleneck), and follows it with a *label predictor*, which is a simple affine or fully-connected layer that maps the concept scores into class predictions. The concept bottleneck is represented by a matrix of m unit-norm *concept vectors* $\mathbf{C}_s = [\mathbf{c}_{s1} \,/\, \|\mathbf{c}_{s1}\|_2 \,\cdots\, \mathbf{c}_{sm} \,/\, \|\mathbf{c}_{sm}\|_2]^\top \in \mathbb{R}^{m \times d}$, where each $\mathbf{c}_{si} \in \mathbb{R}^d$ represents a high-level concept (*e.g.*, "stripes", "fin", "dots"). The m concept scores are obtained via a linear projection $\mathbf{v}_{\mathbf{C}_s}(\mathbf{x}) = \mathbf{C}_s\, \boldsymbol{\phi}(\mathbf{x})$, which is followed by a fully-connected layer to obtain the CBM model as

$$\mathbf{f}_s^{(\text{cbm})}(\mathbf{x}) \,:=\, \mathbf{W}_s\, \mathbf{v}_{\mathbf{C}_s}(\mathbf{x}) \,+\, \mathbf{b}_s \,=\, \mathbf{W}_s \mathbf{C}_s\, \boldsymbol{\phi}(\mathbf{x}) + \mathbf{b}_s \,=\, \mathbf{g}_s(\boldsymbol{\phi}(\mathbf{x})) \qquad (1)$$

The label predictor $\mathbf{g}_s(\mathbf{z})$ is defined by the parameters $\mathbf{W}_s \in \mathbb{R}^{L \times m},\ \mathbf{b}_s \in \mathbb{R}^L$, and $\mathbf{C}_s$. A key advantage of the CBM is that its predictions are an affine combination of the high-level concept scores, which allows for better interpretability of the model. Since the label predictor of a CBM is chosen to be simple, its performance is strongly dependent on the construction of the concept bank.

2.2 Distribution Shifts in the Wild

Let $\mathcal{T} = \{\mathbf{t}_0, \mathbf{t}_1, \ldots, \mathbf{t}_k\}$ be a finite set of measurable input transformations, where each $\mathbf{t}_i : \mathcal{X} \to \mathcal{X}$ is a measurable function. We also define a transformed input space encompassing all possible transformed inputs: $\mathcal{X}_\mathcal{T} = \bigcup_{i=0}^{k} \{\mathbf{t}_i(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\}$. Without loss of generality, we set $\mathbf{t}_0$ to be the identity function $\mathbf{t}_0(\mathbf{x}) = \mathbf{x},\ \forall \mathbf{x} \in \mathcal{X}$. Let μ_s and μ_t be probability measures on $\mathcal{T}$ representing the distributions over input transformations in the source and target domains, respectively. We define the *source domain* D_s, equipped with μ_s such that $\mu_s(\{\mathbf{t}_0\}) = 1,\ \ \mu_s(\{\mathbf{t}_i\}) = 0\ \ \forall i \neq 0$. Its joint distribution is denoted by $\mathbb{P}_s$ over $\mathcal{X}_\mathcal{T} \times \mathcal{Y}$ such that $\mathbb{P}_s(\mathbf{x}, y) = \mathbb{P}(\mathbf{x}, y)\ \ \forall \mathbf{x}, y$, where $\mathbb{P}$ is the underlying distribution over inputs and labels. Similarly, we define the *target domain* D_t with a probability measure μ_t such that $\mu_t(\{\mathbf{t}_i\}) > 0$ for some $i \in [k]$. Its joint distribution is denoted by $\mathbb{P}_t$ over $\mathcal{X}_\mathcal{T} \times \mathcal{Y}$ such that $\mathbb{P}_t(\mathbf{x}, y) = \sum_{i=0}^{k} \mu_t(\{\mathbf{t}_i\})\, \mathbb{P}(\mathbf{t}_i^{-1}(\mathbf{x}), y)$, assuming that the $\mathbf{t}_i$ are invertible or appropriately measurable for their pre-images.

Let $\mathcal{H}$ be a *concept hypothesis* class, defined as the space of measurable concept mappings $\mathbf{h} : \mathbb{R}^d \to \mathbb{R}^m$ from the feature representation $\boldsymbol{\phi}(\mathbf{x})$ to concept scores. We also define the *concept set* $\mathcal{C} := \{c_1, c_2, \cdots, c_m\}$, where each $c_i : \mathbb{R}^d \mapsto \mathbb{R}$ represents a high-level concept mapping (*e.g.*, stripe pattern, grass, beach, *etc.*). For a domain $D_j,\ j \in \{s, t\}$, we define the concept score distribution as $\mathbb{P}_{\text{con}}(D_j, \boldsymbol{\phi}, \mathbf{h}) = (\mathbf{h} \circ \boldsymbol{\phi})_* \mathbb{P}_j$, where $(\mathbf{h} \circ \boldsymbol{\phi})_* \mathbb{P}_j$ is the push-forward measure of $\mathbb{P}_j$ under $\mathbf{h} \circ \boldsymbol{\phi}$. Note that $\mathbf{h}$ is determined by $\mathcal{C}$ such that $\mathbf{h}(\boldsymbol{\phi}(\mathbf{x})) = [c_1(\boldsymbol{\phi}(\mathbf{x})), \cdots, c_m(\boldsymbol{\phi}(\mathbf{x}))]^T$ [1].

[1] A common approach is to define $c_i(\boldsymbol{\phi}(\mathbf{x}))$ as the inner product of a (unit-normalized) concept vector with the feature representation $\boldsymbol{\phi}(\mathbf{x})$, which results in a score for concept i.

Let $\mathcal{G}$ be a *classification hypothesis* class, defined as a set of measurable classifiers $\mathbf{g} : \mathbb{R}^m \to \mathbb{R}^L$ mapping the concept scores to prediction logits. Finally, we define the distribution of predictions as the push-forward measure of $\mathbb{P}_{\text{con}}(D_j, \boldsymbol{\phi}, \mathbf{h})$ under $\mathbf{g}$: $\mathbb{P}_{\text{pred}}(D_j, \boldsymbol{\phi}, \mathbf{h}, \mathbf{g}) \;=\; \mathbf{g}_* \mathbb{P}_{\text{con}}(D_j, \boldsymbol{\phi}, \mathbf{h})$.

Given $\mathbf{h} \in \mathcal{H}$ and $\mathbf{g} \in \mathcal{G}$, we categorize the distribution shifts in the target domain, $\{\mu_t(\mathbf{t}_i) > 0 \mid \mathbf{t}_i \in \mathcal{T}\}$, into one of the following broad categories:

1. **Low-level shift:** This type of transformation does not change the concept score distribution across the domains. Examples include additive Gaussian noise, blurring, and pixelization, which employ low-level changes to the input (*e.g.*, CIFAR10-C (Hendrycks & Dietterich, 2019)):
$$\mathbb{P}_{\text{con}}(D_t, \boldsymbol{\phi}, \mathbf{h}) \;=\; \mathbb{P}_{\text{con}}(D_s, \boldsymbol{\phi}, \mathbf{h}) \tag{2}$$
Naturally, the resulting distribution of predictions based on the concept scores also remains the same across the domains, *i.e.*, $\mathbb{P}_{\text{pred}}(D_s, \boldsymbol{\phi}, \mathbf{h}, \mathbf{g}) \;=\; \mathbb{P}_{\text{pred}}(D_t, \boldsymbol{\phi}, \mathbf{h}, \mathbf{g})$.
2. **Concept-level shift:** This type of transformation alters the concept score distribution, but not the prediction distribution across the domains. Examples include replacing water background with a land background in images (*e.g.*, Waterbirds, Metashift (Sagawa et al., 2019; Liang & Zou, 2021)):
$$\begin{aligned} \mathbb{P}_{\text{con}}(D_t, \boldsymbol{\phi}, \mathbf{h}) \;&\neq\; \mathbb{P}_{\text{con}}(D_s, \boldsymbol{\phi}, \mathbf{h}) \\ \mathbb{P}_{\text{pred}}(D_t, \boldsymbol{\phi}, \mathbf{h}, \mathbf{g}) \;&=\; \mathbb{P}_{\text{pred}}(D_s, \boldsymbol{\phi}, \mathbf{h}, \mathbf{g}) \end{aligned} \tag{3}$$

Definition 1 *The concept set $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ is **complete** if there exists a classifier $\mathbf{g} \in \mathcal{G}$ such that, for both low-level and concept-level shifts, the prediction distributions conditioned on the concepts are identical:*
$$\mathbb{P}_{\textit{pred}}(D_s, \boldsymbol{\phi}, \mathbf{h}, \mathbf{g}) \;=\; \mathbb{P}_{\textit{pred}}(D_t, \boldsymbol{\phi}, \mathbf{h}, \mathbf{g}). \tag{4}$$
This implies that there exists a mapping from concept scores to labels encompassing both the source and target domains.

2.3 Failure Modes of Concept Bottleneck for Foundation Models

Based on the definitions above, we categorize the possible failure modes of the decision-making pipeline of a foundation model equipped with a CBM, defined by a given $D_s, D_t, \boldsymbol{\phi}$, $\mathbf{h} \circ \boldsymbol{\phi} = [c_1 \circ \boldsymbol{\phi}, \cdots, c_m \circ \boldsymbol{\phi}]$, and $\mathbf{g}$ as follows.

1. **Non-robust concept bottleneck under low-level shift:** the concept mapping $\mathbf{h}$ is *not* robust to low-level shifts, causing discrepancies in the concept-level predictions:
$$\mathbb{P}_{\text{con}}(D_t, \boldsymbol{\phi}, \mathbf{h}) \;\neq\; \mathbb{P}_{\text{con}}(D_s, \boldsymbol{\phi}, \mathbf{h}),$$
violating the requirement for a low-level shift in Eqn. 2. Such discrepancies in the concept predictions can lead to degraded performance in D_t, resulting from mismatched prediction distributions, *i.e.*, $\mathbb{P}_{\text{pred}}(D_t, \boldsymbol{\phi}, \mathbf{h}, \mathbf{g}) \;\neq\; \mathbb{P}_{\text{pred}}(D_s, \boldsymbol{\phi}, \mathbf{h}, \mathbf{g})$.

2. **Non-robust classifier under concept-level shift:** Given that the concept score distributions differ due to a concept-level shift as in Eqn. 3, the given classifier $\mathbf{g}$ fails to produce consistent prediction distributions across the domains, violating Eqn 3:
$$\mathbb{P}_{\text{pred}}(D_t, \boldsymbol{\phi}, \mathbf{h}, \mathbf{g}) \neq \mathbb{P}_{\text{pred}}(D_s, \boldsymbol{\phi}, \mathbf{h}, \mathbf{g})$$

3. **Incomplete concept set:** The concept set $\{c_1, c_2, \ldots, c_m\}$ is *not* complete, and there does not exist *any* $\mathbf{g} \in \mathcal{G}$ such that $\mathbb{P}_{\text{pred}}(D_s, \boldsymbol{\phi}, \mathbf{h}, \mathbf{g}) = \mathbb{P}_{\text{pred}}(D_t, \boldsymbol{\phi}, \mathbf{h}, \mathbf{g})$. Intuitively, it fails to capture all the necessary information for consistent predictions across domains, and Definition 1 is not achievable in the first place.

3 CONDA: CONcept-based Dynamic Adaptation

In this section, we propose a dynamic approach for adaptation of a CBM-based *only* on unlabeled test data. We follow the setting of test-time adaptation, where the foundation model $\boldsymbol{\phi}(\mathbf{x})$ and CBM, consisting of the concept bank $\mathbf{C}_s$ and label predictor $(\mathbf{W}_s, \mathbf{b}_s)$, trained on the source domain are given (see Eqn 1), but the source (training) dataset is not available. Let $\mathcal{D}_t = \{\mathbf{x}_{tn}\}_{n=1}^{N_t}$ be the unlabeled test set from the target distribution. To address the potential failure modes in a CBM pipeline identified in Section 2.3, we propose the following three-step adaptation:

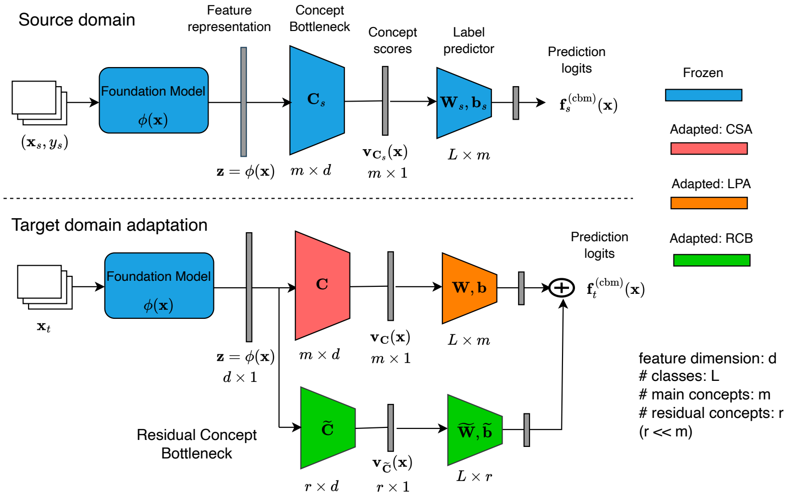


Figure 2: **Overview of CONDA, our proposed adaptation framework.** The foundation model and CBM pipeline trained on the source domain is shown at the top, while the adapted CBM, consisting of a main branch and residual branch, is shown at the bottom. The components of CBM that are adapted during each stage of the proposed method (*i.e.*, CSA, LPA, and RCB) are shown in different colors.

1. **Concept-Score Alignment (CSA):** The goal of this step is to perform a feature alignment of the concept scores of test inputs $\mathbf{v}_{\mathbf{C}}(\mathbf{x}_t) \in \mathbb{R}^m$ such that their class-conditional distributions are close to that of the concept scores in the source dataset [2]. By adapting the concept vectors $\mathbf{C}$, this will ensure that the label predictor continues to "see" very similar class-conditional input distributions at test time, thereby maintaining accurate predictions.

2. **Linear Probing Adaptation (LPA):** To further address any discrepancy or mismatch in the feature alignment CSA step (*e.g.*, due to distribution assumptions), here we adapt the label predictor $(\mathbf{W}, \mathbf{b})$ of the CBM, with the concept vectors fixed at their updated values from the CSA step.

3. **Residual Concept Bottleneck (RCB):** As discussed in Section 2.3, the concept bank from the source domain could be incomplete and new concepts may be required to bridge the distribution gap between the domains. In this step, we introduce a residual CBM with additional concept vectors and a linear predictor, which are jointly optimized (with the parameters of the main CBM fixed) to improve the test accuracy.

Target Domain CBM. Figure 2 shows the overall architecture of CONDA. The residual concept bottleneck is shown as a separate branch, where we introduce r additional concept vectors $\widetilde{\mathbf{C}} = [\,\widetilde{\mathbf{c}}_1 \,/\, \|\widetilde{\mathbf{c}}_1\|_2 \;\cdots\; \widetilde{\mathbf{c}}_r \,/\, \|\widetilde{\mathbf{c}}_r\|_2\,]^\top \in \mathbb{R}^{r \times d}$. The concept scores are obtained by projecting the feature representation $\phi(\mathbf{x})$ on these residual concept vectors, and the scores are passed to another linear predictor $(\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}})$ to obtain the un-normalized class predictions (logits) of the residual CBM: $\widetilde{\mathbf{W}}\widetilde{\mathbf{C}}\,\phi(\mathbf{x}) \,+\, \widetilde{\mathbf{b}}$. The un-normalized predictions of the target domain CBM are obtained by adding that of the main and the residual branch CBMs, giving

$$\begin{aligned}\mathbf{f}_t^{(\text{cbm})}(\mathbf{x}) \;&=\; \mathbf{W}\mathbf{C}\,\phi(\mathbf{x}) \,+\, \mathbf{b} \,+\, \widetilde{\mathbf{W}}\widetilde{\mathbf{C}}\,\phi(\mathbf{x}) \,+\, \widetilde{\mathbf{b}} \\ &=\; (\mathbf{W}\mathbf{C} + \widetilde{\mathbf{W}}\widetilde{\mathbf{C}})\,\phi(\mathbf{x}) \,+\, \mathbf{b} + \widetilde{\mathbf{b}} \;=\; \mathbf{W}_{\text{con}}\mathbf{C}_{\text{con}}\,\phi(\mathbf{x}) \,+\, \mathbf{b}_{\text{con}},\end{aligned} \tag{5}$$

where $\widetilde{\mathbf{W}} \in \mathbb{R}^{L \times r}$ and $\widetilde{\mathbf{b}} \in \mathbb{R}^L$. For comparison with the source domain CBM (Eqn. 1), we have defined the combined parameters from the main and residual branch CBMs as $\mathbf{W}_{\text{con}} = [\mathbf{W}\;\widetilde{\mathbf{W}}] \in \mathbb{R}^{L \times (m+r)}$, $\mathbf{C}_{\text{con}} = [\mathbf{C}\,;\widetilde{\mathbf{C}}] \in \mathbb{R}^{(m+r) \times d}$, and $\mathbf{b}_{\text{con}} = \mathbf{b} + \widetilde{\mathbf{b}} \in \mathbb{R}^L$. That is, adding the residual CBM is equivalent to introducing r additional rows (columns) in the concept (weight) matrix. For adaptation, the parameters of the main CBM $\{\mathbf{C}, \mathbf{W}, \mathbf{b}\}$ are initialized to their corresponding values from the source domain, while the parameters of the residual CBM $\{\widetilde{\mathbf{C}}, \widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}\}$ are initialized randomly.

Pseudo-labeling. Since the test samples are unlabeled, it becomes challenging to design adaptation objectives that can minimize a smooth proxy of the classification error rate on the target distribution.

[2] We drop the subscript 's' to denote that they are adaptation parameters, not specific to the source domain.

We utilize the idea of pseudo-labeling to address this, as commonly done in the TTA and semi-supervised learning literature (Chen et al., 2022; Lee et al., 2013; Sohn et al., 2020). A simple approach for pseudo-labeling the test set is to use the class predictions of the (un-adapted) source-domain CBM, referred to as "self-labeling". However, since this CBM is often not robust to distribution shifts in the first place, this can produce poor-quality pseudo-labels for adaptation. We leverage the fact that the feature extraction backbone $\boldsymbol{\phi}(\mathbf{x})$ is a foundation model that is pre-trained on diverse data distributions, and as a result is likely to be relatively robust to distribution shifts. We take an ensemble of the commonly used *zero-shot* predictor (as done *e.g.*, in Radford et al. (2021)) and a *linear probing* predictor (trained on the source dataset on top of the foundation model) to get the pseudo-labels for test samples. We combine the two by taking the class predicted with higher confidence across both predictors. We note that more sophisticated pseudo-labeling methods *e.g.*, involving weak- and strong-augmentations, and soft nearest-neighbor voting (Chen et al., 2022) can be used to potentially improve our method.

Following the convention in the TTA literature (Wang et al., 2021; Chen et al., 2022), we randomly split the test data into fixed-size batches $\mathcal{D}_t = \bigcup_{b=1}^{B} \mathcal{D}_t^b$, and perform adaptation sequentially on each batch b, obtaining the adapted model's predictions on the same batch, before moving to the next one. Also, the parameters of the CBM (main and residual) are adapted in an online fashion (not episodically) (Wang et al., 2021), *i.e.*, the adapted parameters learned from a batch are used to initialize the next batch and so on [3]. For convenience, we define the test dataset with paired pseudo-labels as $\widehat{\mathcal{D}}_t = \{(\mathbf{x}_{tn}, \widehat{y}_{tn})\}_{n=1}^{N_t}$, and a corresponding pseudo-labeled test batch as $\widehat{\mathcal{D}}_t^b,\ b \in [B]$. We next expand on each stage of the CBM adaptation outlined earlier, and provide a complete algorithm for the same in Algorithm 1 in the Appendix.

3.1 Concept Score Alignment

From Figure 2 (top half) and Eqn. 1, the concept scores $\mathbf{v}_{\mathbf{C}_s}(\mathbf{x}) \in \mathbb{R}^m$ are input to the linear label predictor $\mathbf{W}_s\mathbf{v} + \mathbf{b}_s$. Let $\{\mathbb{P}(\mathbf{v}_{\mathbf{C}_s}(\mathbf{x}_s) \,|\, y_s = y),\ \ y \in \mathcal{Y}\}$ be the class-conditional distributions of these concept scores on the source domain. At test time, if the distribution of the input changes such that $\mathbf{x}_t \sim p_{\text{t}}(\mathbf{x})$, then there is a corresponding change in the class-conditional distributions of concept scores $\{\mathbb{P}(\mathbf{v}_{\mathbf{C}_s}(\mathbf{x}_t) \,|\, y_t = y) = \mathbb{P}(\mathbf{C}_s\boldsymbol{\phi}(\mathbf{x}_t) \,|\, y_t = y),\ \ y \in \mathcal{Y}\}$. The goal of concept-score alignment (CSA) is to adapt the source domain concept bank $\mathbf{C}_s$ to a target domain-specific one $\mathbf{C}_t$ such that the class-conditional distributions after adaptation are close to that of the source domain under some distributional distance (*e.g.*, Kullback-Leibler or Total-variation). Informally, we wish to find an adapted concept bank $\mathbf{C}_t$, starting from $\mathbf{C}_s$, such that

$$\mathbb{P}(\mathbf{C}_t\boldsymbol{\phi}(\mathbf{x}_t) \,|\, y_t = y) \ \approx\ \mathbb{P}(\mathbf{C}_s\boldsymbol{\phi}(\mathbf{x}_s) \,|\, y_s = y),\ \ \forall y \in \mathcal{Y}.$$

If the class priors $\{\mathbb{P}(y_t = y),\ \forall y\}$ do not change significantly, this can ensure that the label predictor of the main CBM continues to receive concept scores from a similar distribution as the source domain.

We model the class-conditional distributions of the concept scores in the source domain as multivariate Gaussians: $\mathbb{P}(\mathbf{v}_{\mathbf{C}_s}(\mathbf{x}_s) \,|\, y_s = y) = \mathcal{N}(\mathbf{v}_{\mathbf{C}_s}(\mathbf{x}_s)\,; \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y),\ \forall y \in \mathcal{Y}$. Given a labeled source-domain dataset, it is straight-forward to estimate $\boldsymbol{\mu}_y$ and $\boldsymbol{\Sigma}_y$ using the sample mean and sample covariance of $\mathbf{v}_{\mathbf{C}_s}(\mathbf{x}_s)$ on the data subset from class y (max-likelihood estimate). Although we cannot access the source domain dataset during adaptation, we assume to have access to these distribution statistics $\{(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)\}_{y \in \mathcal{Y}}$. At test time, changes to the distribution of the concept scores can be captured by a concept matrix $\mathbf{C}$ (to be adapted). For a test input $\mathbf{x}_t$, the distance of its concept scores $\mathbf{v}_{\mathbf{C}}(\mathbf{x}_t)$ from the Gaussian distribution of class y is given by the *Mahalanobis metric* $D_{\text{mah}}(\mathbf{x}_t\,; \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y) = (\mathbf{v}_{\mathbf{C}}(\mathbf{x}_t) - \boldsymbol{\mu}_y)^\top \boldsymbol{\Sigma}_y^{-1}(\mathbf{v}_{\mathbf{C}}(\mathbf{x}_t)) - \boldsymbol{\mu}_y)$.

Intra-class and Inter-class Distances. Taking the pseudo-label $\widehat{y}_t$ as a proxy for the true label of $\mathbf{x}_t$, the *intra-class (or within-class)* distance measures the closeness of $\mathbf{x}_t$ to samples from its own class, while the *inter-class (or between-class)* distance measures the separation of $\mathbf{x}_t$ to samples from the other classes. They are defined as follows:

$$D_{\text{intra}}(\mathbf{x}_t, \widehat{y}_t) \ =\ D_{\text{mah}}(\mathbf{x}_t\,; \boldsymbol{\mu}_{\widehat{y}_t}, \boldsymbol{\Sigma}_{\hat{y}_t}) \quad \text{and} \tag{6}$$

$$D_{\text{inter}}(\mathbf{x}_t, \widehat{y}_t) \ =\ \frac{1}{L-1} \sum_{\ell=1:\ell \neq \widehat{y}_t}^{L} D_{\text{mah}}(\mathbf{x}_t\,; \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell). \tag{7}$$

[3] In the episodic approach, parameters would be reset to their source domain values to initialize each batch.

Motivated by class-aware feature alignment CAFA (Jung et al., 2023), we explore an adaptation loss ℓ_{ada} that is specifically designed to achieve concept-score alignment on a per-class level. This loss is based on the idea that for discriminative feature alignment, the intra-class distances should be small and the inter-class distances should be large on the test samples (Ye et al., 2021; Ming et al., 2023).

$$\ell_{ada}(\mathbf{v}_{\mathbf{C}}(\mathbf{x}_t), \widehat{y}_t) \;=\; \log \frac{D_{\text{intra}}(\mathbf{x}_t, \widehat{y}_t)}{D_{\text{inter}}(\mathbf{x}_t, \widehat{y}_t)}. \tag{8}$$

With this setup, we propose the adaptation objective for CSA to minimize on a test batch:

$$L_{\text{CSA}}(\mathbf{C}) \;=\; \frac{1}{|\widehat{\mathcal{D}}_t^b|} \sum_{(\mathbf{x}_t, \widehat{y}_t) \in \widehat{\mathcal{D}}_t^b} \ell_{ada}(\mathbf{v}_{\mathbf{C}}(\mathbf{x}_t), \widehat{y}_t) \;+\; \lambda_{\text{frob}} \, \|\mathbf{C} - \mathbf{C}_s\|_F^2. \tag{9}$$

The second term is a regularization on how much the concept vectors can deviate from their source domain values in terms of the Frobenius norm.

3.2 Linear Probing Adaptation

In this step, we focus on improving the test accuracy of the label predictor of the main CBM branch $(\mathbf{W}, \mathbf{b})$, with the concept vectors $\mathbf{C}$ fixed at their updated values from the CSA step (the residual CBM parameters are also frozen). For this, we use the cross-entropy loss between the predictions of the target domain CBM (Eqn. 5) and the pseudo-labels of a test batch $\widehat{\mathcal{D}}_t^b$. In order to enhance the interpretability of the label predictor, we impose sparsity and grouping effect in its weights via an Elastic-net penalty term (Zou & Hastie, 2005; Yuksekgonul et al., 2023) given by

$$L_{\text{sparse}}(\mathbf{W}) \;=\; \frac{1}{m\,L} \sum_{\ell=1}^{L} \left(\alpha \, \|\mathbf{w}_\ell\|_1 \;+\; (1-\alpha) \, \|\mathbf{w}_\ell\|_2^2\right), \tag{10}$$

where $\mathbf{w}_\ell \in \mathbb{R}^m$ is the ℓ-th row of $\mathbf{W}$, and $\alpha = 0.99$. The adaptation objective for LPA is given by

$$L_{\text{LPA}}(\mathbf{W}, \mathbf{b}) \;=\; -\frac{1}{|\widehat{\mathcal{D}}_t^b|} \sum_{(\mathbf{x}_t, \widehat{y}_t) \in \widehat{\mathcal{D}}_t^b} \log \boldsymbol{\sigma}_{\widehat{y}_t}(\mathbf{f}_t^{(\text{cbm})}(\mathbf{x}_t)) \;+\; \lambda_{\text{sparse}} \, L_{\text{sparse}}(\mathbf{W}), \tag{11}$$

where $\boldsymbol{\sigma}_k(\mathbf{r})$ is the Softmax probability for class k given the logits $\mathbf{r}$, and $\lambda_{\text{sparse}} \geq 0$ is a sparsity regularization hyper-parameter. Using this objective, the label predictor is adapted such that the CBM's predictions on a test batch are consistent with their pseudo-labels.

3.3 Residual Concept Bottleneck

We next discuss adaptation of the residual branch of the CBM whose parameters are $\{\widetilde{\mathbf{C}}, \widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}\}$. The r additional concept vectors in $\widetilde{\mathbf{C}}$ are expected to capture new concepts in the target data and compensate for the potentially incomplete coverage of the main CBM (see Section 2.3). By increasing the expressiveness of the concept subspace, we expect to improve the accuracy on the target dataset beyond the CSA and LPA steps. Therefore, we first have a cross-entropy loss term in this adaptation objective (as in Eqn. 11). We also introduce a *cosine similarity* based regularization in the objective to ensure that the new concept vectors in $\widetilde{\mathbf{C}}$ are minimally redundant with each other, while also having minimal overlap with the existing concept vectors $\mathbf{C}$ (obtained from the CSA step).

$$L_{\text{sim}}(\widetilde{\mathbf{C}}) \;=\; \frac{1}{m\,r} \sum_{i \in [m]} \sum_{j \in [r]} \cos(\mathbf{c}_i, \widetilde{\mathbf{c}}_j) \;+\; \frac{2}{r\,(r-1)} \sum_{\substack{(i,j) \in [r]^2: \\ j > i}} \cos(\widetilde{\mathbf{c}}_i, \widetilde{\mathbf{c}}_j). \tag{12}$$

Finally, we include a *coherency regularization* term in the objective (modified from Yeh et al. (2020)) to improve the interpretability of the learned residual concepts, given by

$$L_{\text{coh}}(\widetilde{\mathbf{C}}) \;=\; \frac{1}{r\,k} \sum_{i \in [r]} \sum_{\mathbf{x}_t \in T_{\widetilde{\mathbf{c}}_i}} \frac{\langle \widetilde{\mathbf{c}}_i, \boldsymbol{\phi}(\mathbf{x}_t) \rangle}{\|\widetilde{\mathbf{c}}_i\|_2}, \tag{13}$$

where $T_{\widetilde{\mathbf{c}}_i}$ is the subset of the current target batch $\mathcal{D}_t^b$ that has the k-largest concept scores for residual concept vector $\widetilde{\mathbf{c}}_i$ (*i.e.*, the top-k nearest neighbors of $\widetilde{\mathbf{c}}_i$ among the feature representations from $\mathcal{D}_t^b$).

The objective to be minimized for adapting the residual concept bottleneck (with the parameters of the main CBM branch frozen) is given by:

$$L_{\text{RCB}}(\widetilde{\mathbf{C}}, \widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}) \;=\; -\frac{1}{|\widehat{\mathcal{D}}_t^b|} \sum_{(\mathbf{x}_t, \widehat{y}_t) \in \widehat{\mathcal{D}}_t^b} \log \boldsymbol{\sigma}_{\widehat{y}_t}(\mathbf{f}_t^{(\text{cbm})}(\mathbf{x}_t)) \;+\; \lambda_{\text{sim}}\, L_{\text{sim}}(\widetilde{\mathbf{C}}) \;-\; \lambda_{\text{coh}}\, L_{\text{coh}}(\widetilde{\mathbf{C}}). \quad (14)$$

The constants $\lambda_{\text{sim}} \geq 0$ and $\lambda_{\text{coh}} \geq 0$ are hyper-parameters that control the strength of the regularization terms. Note that for the residual CBM, we *jointly* adapt $\widetilde{\mathbf{C}}$ and $\widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}$, because we have a common objective of increasing the test accuracy, whereas for the main CBM, the adaptation is done in two stages (CSA and LPA), with CSA focusing on distribution alignment of the concept scores based on the intra-class and inter-class distances.

4 Experiments

In this section, we conduct experiments to answer the three following research questions:

RQ1: How effective is CONDA in improving the test-time performance of deployed classification pipelines that use foundation models with concept bottlenecks?

RQ2: How does each component of CONDA specifically address and remedy the failures caused by different types of distribution shifts?

RQ3: How do the concept-based explanations change before and after test-time adaptation?

4.1 Setup

A detailed description of the experimental setup is available in Appendix B.1. Anonymized repository for our implementation is at `https://anonymous.4open.science/r/CONDA-D7AE/`.

Datasets. We evaluate the performance of concept bottlenecks for FMs and the proposed adaptation on five real-world datasets with distribution shifts, following the setup in Lee et al. (2023): (1) CIFAR10 to CIFAR10-C and CIFAR100 to CIFAR100-C for low-level shift, (2) Waterbirds and Metashift for concept-level shift, and (3) Camelyon17 for natural shift.

Models. For CIFAR datasets, we use the CLIP:ViT-L/14 (FARE^2) (Schlarmann et al., 2024) as a backbone, which is adversarially fine-tuned to be more robust to (adversarial) low-level perturbations than standard CLIP variants. We employ CLIP:ViT-L/14 (Radford et al., 2021) for Waterbirds and Metashift. For Camelyon17, we utilize MedCLIP (Wang et al., 2022), which is trained to understand medical images and text jointly, making it suitable for zero-shot tasks in the medical domain.

Preparing the Concept Bottleneck. We evaluate CONDA using three popular approaches for constructing the concept bottleneck: (1) using a general-purpose concept bank where natural language concept descriptions and modern vision-language models (*e.g.*, Stable Diffusion (Rombach et al., 2022)) are being leveraged to automatically generate concept examples for finding concept vectors (Yuksekgonul et al., 2023; Wu et al., 2023b); (2) unsupervised learned concepts where concept vectors are learned via optimization to maximize the concept-based prediction accuracy (Yeh et al., 2020); and (3) employing GPT-3 with appropriate filtering to discover a tailored set of concepts for the bottleneck (Oikarinen et al., 2023).

Metrics. We report the performance in terms of two metrics: averaged group accuracy (AVG) and worst-group accuracy (WG). AVG is the average (per-class) accuracy across the classes, and WG is the minimum (per-class) accuracy across the classes.

4.2 RQ1: Effectiveness of CONDA under real-world distribution shifts

Table 1 presents our main results evaluating the effectiveness of CONDA on different real-world distribution shifts, when combined with different CBM baselines. First of all, we observe that leveraging the expressive power of the FM feature representations can enhance the performance of CBMs. For example, using the method from Oikarinen et al. (2023), their reported accuracies on CIFAR10 and CIFAR100 are 86.40% and 65.13% respectively when using the CLIP-RN50 backbone.

Dataset			ZS	LP	Yuksekgonul et al. (2023)		Yeh et al. (2020)		Oikarinen et al. (2023)	
					Unadapted	**w/ CONDA**	Unadapted	**w/ CONDA**	Unadapted	**w/ CONDA**
CIFAR10	Source	AVG	91.18	93.31	92.55 ± 0.05	-	96.26 ± 0.11	-	95.24 ± 0.08	-
		WG	71.1	88.0	85.64 ± 0.55	-	90.89 ± 0.97	-	90.11 ± 0.76	-
	Target	AVG	66.68 ± 15.88	84.11 ± 1.54	82.61 ± 1.65	84.38 ± 1.52	89.76 ± 1.10	85.14 ± 1.29	81.22 ± 2.77	84.56 ± 3.11
		WG	55.04 ± 2.05	71.37 ± 3.33	68.62 ± 2.93	72.69 ± 2.49	78.28 ± 2.43	76.09 ± 1.66	69.03 ± 2.47	72.88 ± 2.01
CIFAR100	Source	AVG	62.73	66.67	65.98 ± 0.10	-	83.87 ± 0.04	-	68.36 ± 0.09	-
		WG	5.12	4.28	9.5 ± 1.14	-	51.0 ± 1.40	-	12.09 ± 1.23	-
	Target	AVG	51.90 ± 1.76	55.30 ± 1.63	51.53 ± 0.13	53.88 ± 0.23	72.33 ± 0.15	70.82 ± 0.20	52.16 ± 0.14	54.79 ± 1.17
		WG	1.73 ± 0.4	2.47 ± 0.49	2.80 ± 0.71	2.56 ± 0.27	30.60 ± 1.42	28.44 ± 0.95	6.32 ± 0.38	6.01 ± 0.22
Waterbirds	Source	AVG	82.61	97.28	97.78 ± 0.16	-	98.80 ± 0.04	-	98.80 ± 0.17	-
		WG	67.45	94.86	96.31 ± 0.38	-	98.21 ± 0.08	-	97.03 ± 0.26	-
	Target	AVG	61.06	53.79	32.03 ± 0.58	60.69 ± 0.23	45.03 ± 0.34	61.11 ± 0.09	46.18 ± 0.42	62.71 ± 0.33
		WG	42.52	44.70	27.80 ± 1.24	43.01 ± 0.46	38.74 ± 0.68	41.86 ± 0.25	35.29 ± 1.52	44.01 ± 0.60
Metashift	Source	AVG	95.72	97.18	97.94 ± 0.10	-	97.18 ± 0.01	-	98.02 ± 0.10	-
		WG	93.44	96.0	96.94 ± 0.30	-	96.0 ± 0.01	-	97.25 ± 0.10	-
	Target	AVG	94.65	81.03	84.45 ± 1.39	93.69 ± 0.20	90.53 ± 0.09	93.81± 0.13	83.72 ± 2.21	93.90 ± 0.13
		WG	92.81	65.03	73.89 ± 3.21	92.02 ± 0.12	84.84 ± 0.20	91.41 ± 0.26	75.41 ± 1.68	91.77 ± 0.12
Camelyon17	Source	AVG	53.09	79.89 ± 0.05	76.92 ± 0.06	-	94.58 ± 0.10	-	79.15 ± 0.08	-
		WG	11.75	79.28 ± 0.01	76.21 ±0.16	-	92.20 ± 0.44	-	78.01 ± 0.15	-
	Target	AVG	48.87	68.37 ± 0.07	67.35 ± 0.12	67.56 ± 0.11	88.72 ± 0.28	86.04 ± 0.19	66.29 ± 0.18	67.05 ± 0.08
		WG	14.66	68.32 ± 0.05	62.15 ± 0.19	65.36 ± 0.14	81.42 ± 1.15	81.01 ± 1.65	59.35 ± 0.21	65.17 ± 0.14

Table 1: Performance of CONDA on different distribution shifts when combined with different CBM baselines. Zero-shot (ZS) and Linear probing (LP) are the non-interpretable FM baselines. Low-level shifts are covered by the CIFAR datasets, concept-level shifts by Waterbirds and Metashift, and natural shifts by the Camelyon17 benchmark. CONDA significantly improves the AVG and WG accuracy on the target domain in many scenarios.

In our experiments, by employing the adversarially fine-tuned CLIP-ViT-L/14, we achieve higher accuracies of 95.24% and 68.36% respectively (source domain). This demonstrates the potential for improved utility in concept-based interpretable pipelines as foundation models continue to get better.

However, this improved performance in the source domain often does not translate to robustness post-deployment. Under low-level shifts, the performance of CBMs may be comparable to that of non-interpretable counterparts (ZS and LP), but is not inherently more robust to low-level shifts. The performance drop is particularly severe under concept-level shifts when the CBM is not adapted. But with adaptation using CONDA, the test-time accuracy under different distribution shifts increases significantly in most cases. The performance is on par with or even surpasses that of the non-interpretable methods, particularly in terms of the WG accuracy.

4.3 RQ2: Effectiveness of Individual Components of CONDA

We now analyze the individual contributions of the components in CONDA: CSA, LPA, and RCB. Figure 3 illustrates the relative AVG and WG (%) when adapting the CBM of Yeh et al. (2020). Under low-level shifts, CSA plays a crucial role in performance improvement by encouraging the high-level concept scores to remain similar. Interestingly, using CSA alone even surpasses the performance achieved when all components are combined. This trend is also observed with the Camelyon17 dataset, which resembles a low-level shift due to lighting differences across hospitals. On the other hand, under concept-level shifts, LPA and RCB become the key contributors. These components allow the model to adjust concept reliance to the target domain and address the incompleteness of the deployed concept set, tailoring it to the target data. In this context, CSA has minimal impact, while using only LPA leads to performance gains comparable to, or even exceeding those achieved when all components are included.

This phenomenon aligns with the findings of Lee et al. (2023) that fine-tuning only a subset of layers can be more effective than fine-tuning all layers, depending on the type of distribution shift. In our case, the concept-based prediction pipeline can be considered a special instance of their framework with a two-layer classifier. The concept bottleneck layer corresponds to the first layer, which is particularly effective in addressing input-level shifts (following their definition), while the linear probing layer corresponds to the second layer, which is more effective in handling output-level shifts (see Section 3 of their paper). These empirical observations confirm our design motivation for CONDA: different components play specific roles in adapting to different types of distribution shifts.

4.4 RQ3: Interpretability of CONDA

We investigate how the concept-based explanations change through adaptation by CONDA. In Figure 4a, we present the top five most prominent concepts contributing to the predictions for each class.

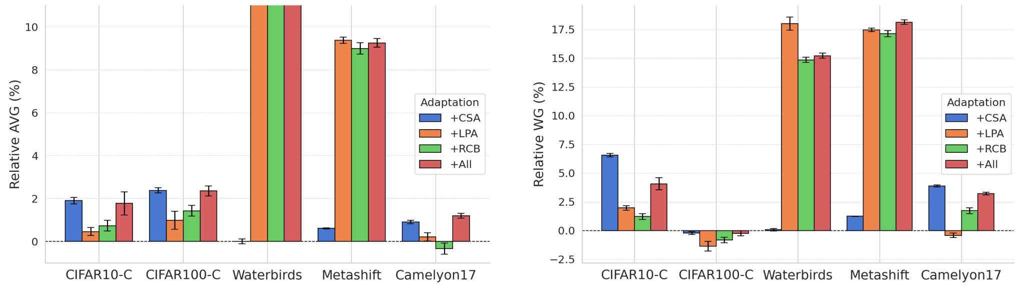
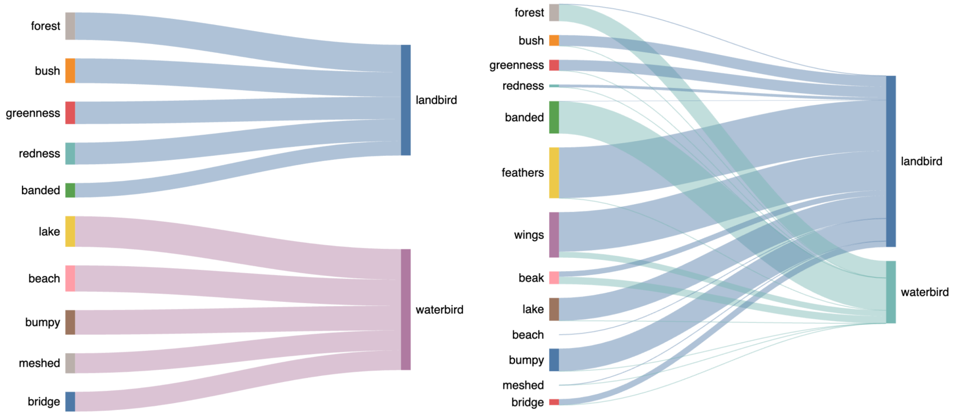


Figure 3: Effectiveness of individual components of CONDA for the CBM method of Yuksekgonul et al. (2023). We report the relative AVG and WG, which is the (accuracy after adaptation) − (accuracy before adaptation).



(a) Before adaptation (source)

(b) After adaptation (target)

Figure 4: **CONDA adapts the concept weights to be tailored to the target data.** We visualize the linear probing layer weights (width of each mapping) before vs after applying CONDA to Yuksekgonul et al. (2023) with Watershift data. We only show the mappings with positive weights.

As expected, in the source domain, land-related concepts are most important for predicting "landbird", and do not positively contribute to "waterbird", and vice versa for water-related concepts. After adapting to the target domain, we observe adjustments in the concept-to-class mappings. Notably, land-related concepts begin to positively contribute to the prediction of "waterbird". This shift indicates that CONDA successfully adapts the concept-based explanations to reflect the new correlations in the target domain. Moreover, in the original concept bottleneck constructed following Wu et al. (2023b), there were no bird-related concepts that could help make robust predictions independent of spurious background correlations. By employing RCB with five residual concepts, we identified that three of them correspond to bird-related concepts: feathers, wings, and beak [4]. This demonstrates that CONDA adapts in a manner aligned with human intuition, just like a human intervening in CBMs to correct predictions. More importantly, RCB captures concepts that may have been missed during the initial construction of the concept bottleneck, enhancing both interpretability and robustness.

5 Conclusions and Future Work

This work made the first attempt to study the post-deployment performance of concept bottlenecks for foundation models. We formalized potential failure modes under low-level and concept-level distribution shifts and proposed a novel test-time adaptation framework. Each component of our framework is designed to address specific failure modes, effectively improving the test-time performance of a deployed CBMs. Limitations and Future work are discussed in Appendix A.3.

[4] To interpret the residual concepts, we use automated concept annotations; see details in Appendix A.2

Reproducibility Statement

We fully comply with the reproducibility policy. Relevant implementation details, hyperparameters, and experimental setups are further clarified in Appendix B.1. Additionally, we provide a source code in an anonymized repository at `https://anonymous.4open.science/r/CONDA-D7AE/`, which contains the code necessary to reproduce the key results presented in the paper.

References

Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 700–712, 2020.

Dyah Adila, Changho Shin, Linrong Cai, and Frederic Sala. Zero-shot robustification of zero-shot models with foundation models. *arXiv preprint arXiv:2309.04344*, 2023.

Ehab A AlBadawy, Ashirbani Saha, and Maciej A Mazurowski. Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. *Medical physics*, 45(3):1150–1158, 2018.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021. URL `https://arxiv.org/abs/2108.07258`.

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 295–305. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00039. URL `https://doi.org/10.1109/CVPR52688.2022.00039`.

Jihye Choi, Jayaram Raghuram, Ryan Feng, Jiefeng Chen, Somesh Jha, and Atul Prakash. Concept-based explanations for out-of-distribution detectors. In *International Conference on Machine Learning*, pp. 5817–5837. PMLR, 2023.

Ching-Yao Chuang, Varun Jampani, Yuanzhen Li, Antonio Torralba, and Stefanie Jegelka. Debiasing vision-language models via biased prompts. *arXiv preprint arXiv:2302.00070*, 2023.

Sedigheh Eslami, Christoph Meinel, and Gerard De Melo. PubMedCLIP: How much does CLIP benefit visual question answering in the medical domain? In *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 1151–1163, 2023.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *CVPR*, 2023.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

Sanghun Jung, Jungsoo Lee, Nanhee Kim, Amirreza Shaban, Byron Boots, and Jaegul Choo. CAFA: Class-aware feature alignment for test-time adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19060–19071, 2023.

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

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`.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896. Atlanta, 2013.

Yoonho Lee, Annie S. Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. In *The Eleventh International Conference on Learning Representations, ICLR*. OpenReview.net, 2023. URL `https://openreview.net/pdf?id=APuPRxjHvZ`.

Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *CoRR*, abs/2303.15361, 2023. doi: 10.48550/ARXIV.2303.15361. URL `https://doi.org/10.48550/arXiv.2303.15361`.

Weixin Liang and James Zou. Metashift: A dataset of datasets for evaluating contextual distribution shifts and training conflicts. In *International Conference on Learning Representations*, 2021.

Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. How to exploit hyperspherical embeddings for out-of-distribution detection? In *The Eleventh International Conference on Learning Representations (ICLR)*. OpenReview.net, 2023. URL `https://openreview.net/pdf?id=aEFaE0W5pAd`.

Zachary Nado, Shreyas Padhy, D Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020.

Tuomas Oikarinen and Tsui-Wei Weng. CLIP-Dissect: Automatic description of neuron representations in deep vision networks. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=iPWiwWHc1V`.

Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=FlCg47MNvBA`.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

Christian Schlarmann, Naman Deep Singh, Francesco Croce, and Matthias Hein. Robust CLIP: Unsupervised adversarial fine-tuning of vision embeddings for robust large vision-language models. In *Forty-first International Conference on Machine Learning (ICML)*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=WLPhywf1si`.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/06964dce9addb1c5cb5d6e3d9838f733-Abstract.html`.

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=uXl3bZLkr3c`.

Zifeng Wang, Zhenbang Wu, Dinesh Agarwal, and Jimeng Sun. Medclip: Contrastive learning from unpaired medical images and text. *arXiv preprint arXiv:2210.10163*, 2022.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023a.

Shirley Wu, Mert Yuksekgonul, Linjun Zhang, and James Zou. Discover and cure: Concept-aware mitigation of spurious correlation. *arXiv preprint arXiv:2305.00650*, 2023b.

Haotian Ye, Chuanlong Xie, Tianle Cai, Ruichen Li, Zhenguo Li, and Liwei Wang. Towards a theoretical framework of out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34: 23519–23531, 2021.

Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in neural information processing systems*, 33:20554–20565, 2020.

Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2636–2645, 2020.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=nA5AZ8CEyow`.

Marvin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642, 2022.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320, 2005.

APPENDICES

A ALGORITHMIC DETAILS

A.1 TEST-TIME ADAPTATION IN CONDA

Here we describe the comprehensive algorithm of CONDA.

Algorithm 1 CONDA: CONCEPT-BASED DYNAMIC ADAPTATION

```
Inputs: Foundation model $\phi(\mathbf{x})$. Source domain CBM: $\mathbf{C}_s, \mathbf{W}_s, \mathbf{b}_s$. Concept scores distribution
    statistics: $\{(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)\}_{y \in \mathcal{Y}}$. Unlabeled test dataset $\mathcal{D}_t$.
Set constants and hyper-parameters:
    # batches $B$, # gradient steps $n_{\text{grad}}$, # residual concepts $r$
    Regularization constants: $\lambda_{\text{frob}}, \lambda_{\text{sparse}}, \lambda_{\text{sim}}, \lambda_{\text{coh}}$
Initialize the main CBM branch using source domain parameters: $\mathbf{C} = \mathbf{C}_s, \mathbf{W} = \mathbf{W}_s, \mathbf{b} = \mathbf{b}_s$.
Initialize the residual CBM branch parameters $\widetilde{\mathbf{C}}, \widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}$ randomly.
Split the test dataset randomly into $B$ fixed-size batches $\{\mathcal{D}_t^b\}_{b=1}^B$.
for batch $b = 1, 2, \cdots, B$ do
    Pseudo-labeling: Using the foundation model, take an ensemble of the zero-shot predictor
        and the linear-probing predictor to obtain pseudo-labels for the test batch.
    CSA Step: Adapt $\mathbf{C}$ with the remaining parameters fixed at their current values.
    for step $i = 1, 2, \cdots, n_{\text{grad}}$ do
        Compute the intra-class and inter-class Mahalanobis distances for the pseudo-labeled test
            batch $\widehat{\mathcal{D}}_t^b$ (Eqns. 6 and 7).
        Compute the CSA adaptation objective $L_{\text{CSA}}(\mathbf{C})$ (Eqns. 8 and 9).
        Perform a gradient descent step to update $\mathbf{C}$.
    end for
    LPA Step: Adapt $(\mathbf{W}, \mathbf{b})$ with the remaining parameters fixed at their current values.
    for step $i = 1, 2, \cdots, n_{\text{grad}}$ do
        Compute the Elastic-net regularization term $L_{\text{sparse}}(\mathbf{W})$ (Eqn. 10).
        Compute the LPA adaptation objective $L_{\text{LPA}}(\mathbf{W}, \mathbf{b})$ (Eqn. 11).
        Perform a gradient descent step to update $\mathbf{W}, \mathbf{b}$.
    end for
    RCB Step: Adapt $(\widetilde{\mathbf{C}}, \widetilde{\mathbf{W}}, \widetilde{\mathbf{b}})$ with the remaining parameters fixed at their current values.
    for step $i = 1, 2, \cdots, n_{\text{grad}}$ do
        Compute the cosine similarity regularization term $L_{\text{sim}}(\widetilde{\mathbf{C}})$ (Eqn. 12).
        Compute the coherency regularization term $L_{\text{coh}}(\widetilde{\mathbf{C}})$ (Eqn. 13).
        Compute the RCB adaptation objective $L_{\text{RCB}}(\widetilde{\mathbf{C}}, \widetilde{\mathbf{W}}, \widetilde{\mathbf{b}})$ (Eqn. 14).
        Perform a gradient descent step to update $\widetilde{\mathbf{C}}, \widetilde{\mathbf{W}}, \widetilde{\mathbf{b}}$.
    end for
    Using the adapted parameters, obtain the target domain CBM predictions $\mathbf{f}_t^{(\text{cbm})}(\mathbf{x})$ for the
        current batch (Eqn. 5).
    Initialize parameters for the next batch using the adapted parameters from the current batch.
end for
Outputs: Predictions of the target domain CBM on the test dataset. Final adapted parameters of the
    target domain CBM: $\mathbf{C}_t, \mathbf{W}_t, \mathbf{b}_t, \widetilde{\mathbf{C}}_t, \widetilde{\mathbf{W}}_t, \widetilde{\mathbf{b}}_t$.
```

A.2 Automatically annotating concepts

Adopt and modify CLIP-DISSECT (Oikarinen & Weng, 2023) as follows.

Say $\mathcal{S}$ is the set of potential annotations. We use ConceptNet Speer et al. (2017) to obtain texts that are relevant to the classes. ConceptNet is an open knowledge graph, where we can find concepts that have particular relations to a query text. For instance, for a class "cat", one can find relations of the form "A Cat has {whiskers, four legs, sharp claws, ..}". Similarly, we can find "parts" of a given class (*e.g.*, "bumper", "roof" for "truck"), or the superclass of a given class (*e.g.*, "animal", "canine" for "dog"). Following the setup in Yuksekgonul et al. (2023), we restrict ourselves to five sets of relations for each class: the hasA, isA, partOf, HasProperty, MadeOf relations in ConceptNet. We collect all concepts that have these relations with classes in each classification task to build the concept subspace. But for Waterbirds dataset, since the classes of {"waterbird", "landbird"} is too specific terminologies, and we cannot find relevant nodes in ConceptNet, we instead use {"bird", "water", "land"} as the query. When we have the concept annotations for the main concept bottleneck from before-deployment (*e.g.*, Yuksekgonul et al. (2023); Oikarinen et al. (2023); Wu et al. (2023b), we set $\mathcal{S}$ as the union set of those pre-defined concepts and those identified by ConceptNet.

Let $\mathcal{D}_t$ be the entire target domain test set. Let ϕ^I_{CLIP} and ϕ^T_{CLIP} be the image encoder and text encoder of CLIP:ViT-B/16. Recall that $\boldsymbol{\phi}$ is the backbone foundation model used in our framework.

To determine the annotation for i-th concept $\mathbf{c}_a \in \mathbf{C}_t$: our goal is to assign an appropriate $t_b \in \mathcal{S}$ as follows,

1. Compute the normalized text embedding of concepts in $\mathcal{S}$ using ϕ^T_{CLIP}; let T_j be the normalized text embedding of the j-th concept in $\mathcal{S}$. Also, compute the image embedding of all images in $\mathcal{D}_t$ using ϕ^I_{CLIP}; let I_i be the image embedding of the i-th data in $\mathcal{D}_t$. Then we take the inner product of the two; the image-text matrix $P = I \times T' \in \mathbb{R}^{|\mathcal{D}_t|\times|\mathcal{S}|}$ where $I \in \mathcal{R}^{|\mathcal{D}_t|\times d}$ and $T \in \mathcal{R}^{|\mathcal{S}|\times d}$ and d is the dimension of the CLIP embeddings. That is, $P_{i,j}$ is the inner product of the normalized embeddings of i-th target image and j-th candidate annotation.

2. For all images in the target dataset, compute and collect their concept scores, $\mathbf{v}_{\mathbf{c}_a} = \{\langle \boldsymbol{\phi}(\mathbf{x}_i), \mathbf{c}_a \rangle\}_{\mathbf{x}_i \in \mathcal{D}_t} \in \mathbb{R}^{|\mathcal{D}_t|}$.

3. The annotation for $\mathbf{c}_a$ is determined by calculating the most *similar* concept in $\mathcal{S}$ with respect to the its concept scores $\mathbf{V}_{\mathbf{c}_a}$. The similarity is defined as,

$$\texttt{sim}(t_l, \mathbf{v}_{\mathbf{c}_a}; P) = \frac{\langle \mathbf{v}_{\mathbf{c}_a}, P \rangle}{\|\mathbf{v}_{\mathbf{c}_a}\| \cdot \|P\|} \tag{15}$$

   which is the cosine similarity between the corresponding concept scores and the corresponding column of image-text matrix, $P_{:,a}$, and $\texttt{sim}(\mathbf{v}_{\mathbf{c}_a}, P)\mathbb{R}^{|\mathcal{S}|}$. Then, the annotation for $\mathbf{c}_a$ becomes the concept in $\mathcal{S}$ with the maximum similarity; $b = \arg\max_l \texttt{sim}(t_l, \mathbf{v}_{\mathbf{c}_a}; P)$. Note that we only accept t_b as the annotation of $\mathbf{c}_a$ only when $\texttt{sim}(t_l, \mathbf{v}_{\mathbf{c}_a}; P) > 0.8$.

To annotate the concepts in our residual concept bottleneck $\widetilde{\mathbf{C}}$, we repeat the same process.

A.3 Limitations and Future Work.

As noted in the results of Section 4.2, we acknowledge that the effectiveness of our framework is limited by the inherent robustness of the backbone foundation model, especially due to its reliance on pseudo labels.

We note that there are instances where our adaptation does not yield improvements with the CBM method of Yeh et al. (2020). In cases such as the CIFAR datasets and Camelyon17, the unadapted CBM already outperforms ZS or LP in the target domain, and adaptation using pseudo-labels based on these methods can negatively impact its performance. This is likely because the concept learning algorithm in Yeh et al. (2020) is designed to optimize accuracy, with the concept bottleneck layer serving as an additional layer that can be optimized alongside the subsequent LP layer. However, a caveat of this approach is that the interpretability of the concept bottleneck is not guaranteed, whereas methods like Yuksekgonul et al. (2023) and Oikarinen et al. (2023) provide clear textual annotations for concepts, enhancing interpretability.

Future work could involve employing more sophisticated pseudo-labeling techniques or robustifying the foundation model itself. Despite these limitations, we believe our work is an important first step toward leveraging off-the-shelf foundation models in interpretable decision-making processes while preserving post-deployment utility. Our framework stands to benefit further from the rapid evolution of foundation models.

When the backbone foundation model remains robust (*e.g.*, against low-level shifts with less severity or concept-level shifts), concept-based predictions get more brittle than feature-based predictions. When the backbone foundation model is not robust (*e.g.*, against low-level shifts with severity level 5), concept-based predictions indeed remain more robust than feature-level predictions; see Table 2.

Dataset			ZS	LP	Yuksekgonul et al. (2023)				Yeh et al. (2020)			
					w/o adaptation	+ CSA	+ LPA	+ CSA + LPA	w/o adaptation	+ CSA	+ LPA	+ CSA + LPA
Metashift	Source	AVG	0.957	0.972	0.979 ± 0.001	-	-	-	0.972 ± 0.001	-	-	-
		WG	0.934	0.960	0.969 ± 0.003	-	-	-	0.960 ± 0.001	-	-	-
	Target	AVG	0.705	0.835	0.890 ± 0.006	0.620 ± 0.049	0.713 ± 0.005	0.676 ± 0.009	0.840 ± 0.009	0.834 ± 0.009	0.749 ± 0.008	0.690 ± 0.005
		WG	0.460	0.720	0.850 ± 0.013	0.279 ± 0.110	0.476 ± 0.017	0.398 ± 0.018	0.712 ± 0.018	0.700 ± 0.020	0.512 ± 0.016	0.400 ± 0.010

Table 2: **Negative results of our test-time adaptation.** In the target domain, the model faces Metashift images with random Gaussian noise Hendrycks & Dietterich (2019). When the performance of zero-shot inference is poor in the target domain, the pseudo-label cannot serve as a reliable reference for the test-time adaptation.

B Experiments

B.1 Experimental Details

All the experiments are run on a server with thirty-two AMD EPYC 7313P 883 16-core processors, 528 GB of memory, and four 884 Nvidia A100 GPUs. Each GPU has 80 GB of 885 memory. For each setup, we repeated each experiment for 10 trials (using seed 40-49) and report the mean and standard error.

2.1 Datasets

CIFAR10. It consists of 60k RGB images of size 32x32 (50k images for the train set, and 10k images for the test set), equally balanced over 10 different classes (*e.g.*, airplane, car, dog, cat, etc.). We follow the given train/test split to report the performance in the source domain.

CIFAR100. It is similar to CIFAR10, but in a larger-scale; there are 100 classes, and each class has 500 32x32 RGB training images and 100 test images, making the classification more challenging.

CIFAR10-C and CIFAR100-C. To report the accuracies, we take the average over 15 different types of corruptions with the severity level of two (out of the scale from one to five); Gaussian Noise, Shot Noise, Impulse Noise, Defocus Blur, Frosted Glass Blur, Motion Blur, Zoom Blur, Snow, Frost, Fog, Brightness, Contrast, Elastic, Pixelate, JPEG Compression. Conventionally, studies in out-of-distribution generalization literature, severity level five is used, but we observe that it severely hurts the performance of the foundation model, making it impossible to be used as a decent oracle for the pseudo labeling. Hence, we chose the severity level that is still causing the performance drop due to the distribution shift, but against which, the backbone model still presents decent performance compared to the CBMs.

Waterbirds. Waterbirds dataset is for a two-class classification task (“landbird” vs. “waterbird”). In the source domain, landbird (waterbird) images are always associated with the land (water) background, while in the target domain, the correlation with the background is flipped, *i.e.*, landbird (waterbird) images are always on the water (land) background.

Metashift. Metashift has two classes of “cat” and “dog”, and it simulates the disparate correlation to the backgrounds in a similar way. Source cat images are always correlated with a sofa or bed in the background, while dog images are always correlated with a bench or bike in the background. For evaluation, we randomly split 90:10 equally across the correlation types, *i.e.*, 10% of dog images with sofa, 10% of dog images with bed, 10% of cat images with bench, and 10% of cat images with bike.

Camelyon17. This dataset is a collection of histopathology whole-slide images used for the detection of metastases in lymph nodes; classifying the given slide into benign tissue vs cancerous tissue. It includes images from five medical centers, each with different staining protocols, equipment, and imaging settings. These differences simulate natural real-world distribution shifts. We use the train set (hospital 1-3) for source, and the test set (hospital 5) for target.

2.2 Preparing Concept Bottleneck

Preparing the concept bottleneck. There are various ways of defining the concept vectors $\{\mathbf{c}_{si}\}_{i=1}^{m}$ in the concept prediction layer $\mathbf{v}_{\mathbf{C}_s}(\mathbf{x})$ (see Appendix **??** for detailed discussion). Early works on CBM required the training dataset to have concept annotations from domain experts in addition to the class labels for training the concept predictor (Koh et al., 2020). Subsequent works have also explored learning the concept vectors in an unsupervised manner (without any concept annotations) (Yeh et al., 2020; Choi et al., 2023). More recently, natural language concept descriptions and modern vision-language models (*e.g.*, Stable Diffusion (Rombach et al., 2022)) are being leveraged to automatically generate concept examples (Yuksekgonul et al., 2023; Wu et al., 2023b) for finding the Concept Activation Vectors (CAVs) (Kim et al., 2018) (each CAV corresponds to a $\mathbf{c}_{si}$), or to directly guide the construction of concept bank $\mathbf{C}_s$ (Oikarinen et al., 2023). We highlight that in all prior works (to our knowledge) the *concept bank remains static*, *i.e.*, once the set of concept vectors is defined and the CBM is deployed, its predictions are made based on these predefined concepts, regardless of any distribution shift at test time.

Yuksekgonul et al. (2023). For CIFAR10 and CIFAR100, we use the BRODEN visual concepts datasets Bau et al. (2017) to learn concept activation vectors, which are used to initialize the weights and bias parameters of the concept bottleneck layer, as described in Yuksekgonul et al. (2023). For Waterbirds and Metashift, we use the images belonging to the concept categories as follows; nature, color, and textures for Waterbirds, and nature, color, texture, city, household, and others for Metashift. For Camelyon17, we use color and textures categories, following the setting in Wu et al. (2023b).

Yeh et al. (2020). For fair comparison, we set the number of the concepts to be the same as the size of concept bottleneck by Yuksekgonul et al. (2023) except with Metashift where we use 100 concepts instead, since with over 100 concepts, we found there are much unnecessary redundancy between them.

Oikarinen et al. (2023). Following their instructions, we create the initial concept set using GPT-3, followed by concept filtering. For the sparsity of the linear probing layer, we set $\lambda = 0.001$ and $\alpha = 0.5$.

Table 3 shows the summary of the major hyperparameters used in the experiments.

Dataset	Backbone	Batch Size	# Epochs	lr (CSA, LPA, RCB)	Adaptation steps	$\{\lambda_{\text{frob}}, \lambda_{\text{sparse}}, \lambda_{\text{sim}}, \lambda_{\text{coh}}\}$
CIFAR10	CLIP:ViT-L-14 (FARE^2)	128	50	Adam, 0.01	50	$\{0.1, 1.0, 0.1, 2.0\}$
CIFAR100	CLIP:ViT-L-14 (FARE^2)	512	50	Adam, 0.01	50	$\{0.1, 1.0, 0.1, 2.0\}$
Waterbirds	CLIP:ViT-L-14	32	20	SGD, 0.1	20	$\{2.5, 1.0, 0.1, 0.1\}$
Metashift	CLIP:ViT-L-14	32	20	SGD, 0.1	50	$\{5.0, 2.0, 1.0, 0.1\}$
Camelyon17	MedCLIP	64	30	SGD, 0.01	20	$\{0.5, 1.0, 0.5, 1.0\}$

Table 3: **Overview of parameters used in the experiments.**