PREFERENCE OPTIMIZATION FOR CONCEPT BOTTLENECK MODELS

Emiliano Penaloza[a,b], **Tianyue H. Zhang**[a,b], **Laurent Charlin***[b,c] **& Mateo Espinosa Zarlenga***[d]
[a]Université de Montréal, [b]Mila - Quebec AI Institute, [c]HEC Montréal, [d]University of Cambridge

ABSTRACT

Concept Bottleneck Models (CBMs) propose to enhance the trustworthiness of AI systems by constraining their decisions on a set of human-understandable concepts. However, CBMs typically assume that datasets contain accurate concept labels—an assumption often violated in practice, which we show can significantly degrade performance (by 25% in some cases). To address this, we introduce the *Concept Preference Optimization* (CPO) objective, a new loss function based on Direct Preference Optimization, which effectively mitigates the negative impact of concept mislabeling on CBM performance. We provide an analysis of some key properties of the CPO objective showing it directly optimizes for the concept's posterior distribution, and contrast it against Binary Cross Entropy (BCE) where we show CPO is inherently less sensitive to concept noise. We empirically confirm our analysis, finding that CPO consistently outperforms BCE in three real-world datasets with and without added label noise.

1 INTRODUCTION

It is a well-known adage that "garbage in" leads to "garbage out." Yet, when designing machine learning (ML) methods that rely on high-quality labelled data, this concern is often overlooked. In this work we show that Concept Bottleneck Models (CBMs) (Koh et al., 2020), a popular but label-hungry family of interpretable neural architectures, when trained with mislabelled concepts are specifically affected by this oversight. CBMs offer a promising solution to the opacity of Deep Neural Networks (DNNs) by using human-understandable concepts (e.g., "*has tail*", "*has whiskers*") as intermediate representations. This structure allows experts to *intervene* at test time by correcting mispredicted concepts, updating the model's final prediction (Shin et al., 2023). With their interpretability and intervenability (Marcinkevičs et al., 2024), CBMs are well suited for high-stakes tasks where verifiability is paramount. However, their reliance on labelled concepts makes them vulnerable to noise. To address this, we propose a learning objective that improves CBM robustness under mislabelled data. Although CBMs are promising, they assume the concept annotations are *correct* for all samples—an unrealistic assumption when labeling - potentially - hundreds of concepts per datum. A study found that 12% of ImageNet-1K animal validation images are mislabelled, with some classes exceeding 90% (Luccioni & Rolnick, 2023). Concept labels, being more granular, likely have even higher error rates, underscoring the need for robustness to label noise. CBMs are often deployed in noisy real-world domains like healthcare (Sylolypavan et al., 2023), where labels can be subjective (Wei et al., 2024). Even with correct labels, CBMs rely on data augmentations (e.g., random crops or flips; see Figure 1) that can distort concepts, making some mislabeling inevitable. Thus, improving CBMs' robustness to concept-label noise is crucial for real-world usability.

Inspired by Preference Optimization (PO), which relaxes the assumption that training data is optimally labelled, we propose *Concept Preference Optimization (CPO)*—a PO-based loss for CBMs. Unlike likelihood-based learning, CPO assumes only *preference* over labels, making it well-suited for noisy settings (Kaufmann et al., 2023; Bengs et al., 2021). Our analysis and empirical results demonstrate that CPO improves both in noiseless settings and mitigates the impact of mislabelled concepts.

*Equal Supervision

2 Related Work

Concept Learning (CL) Concept Learning is a subfield of eXplainable AI (XAI) where models are designed to explain their prediction using human-understandable units of information, or *concepts* (Bau et al., 2017; Kim et al., 2018), that are relevant for a task of interest (Poeta et al., 2023). While CL methods use diverse approaches to produce concept-based explanations, most can be framed within the context of a Concept Bottleneck Model (CBM) (Koh et al., 2020). A CBM is a neural architecture composed of (1) a *concept predictor* $\pi_\theta(c \mid x)$, which maps input features x to a predicted distribution c over a set of pre-defined concepts, and (2) a *label predictor* $f_\phi(c)$, which maps the set of predicted concepts c to a downstream label y. By conditioning their task predictions on a set of concepts, CBMs can explain their prediction through their predicted concepts. They also allow for *concept interventions*, where, at test time, an expert interacting with the CBM can correct a handful of its mispredicted concepts, leading to significant improvements in task accuracy (Shin et al., 2023).

Recent approaches have expanded the reach of CBMs across varying real-world setups. Concept Embedding Models (CEMs) (Espinosa Zarlenga et al., 2022) enhance the expressivity of concept representations to enable CBMs to be competitive in datasets with *incomplete* (Yeh et al., 2020) concept annotations. Post-hoc CBMs (Yuksekgonul et al., 2023), LaBOs (Yang et al., 2023), and Label-free CBMs (Oikarinen et al., 2023) instead address the difficulty of sourcing concept labels and retraining models by exploiting foundation and pretrained models. Further works improve the effectiveness of concept interventions by introducing new training losses (Espinosa Zarlenga et al., 2023), intervention policies (Chauhan et al., 2022), or considering inter-concept relationships (Havasi et al., 2022; Steinmann et al., 2023; Vandenhirtz et al., 2024; Raman et al., 2024).

Among these, the closest to our work are Probabilistic CBMs (ProbCBM) (Kim et al., 2023) and Stochastic CBMs (SCBMs) (Vandenhirtz et al., 2024). Both approaches frame CBMs probabilistically and learn to amortize the posterior distribution of an auxiliary latent variable between the concepts and the input data. ProbCBMs amortize the latent's posterior using a diagonal covariance matrix to estimate concept uncertainty. In contrast, SCBMs amortize the full covariance matrix and use it to estimate joint concept distributions for more efficient interventions. Both approaches have their benefits, but they both approximate the posterior of a latent variable and *not* the concept distributions. Conversely, we show how the CPO objective is equivalent to learning the posterior distribution of the concepts.

Preference Optimization (PO) Previous works show PO is a powerful learning framework for settings where the training labels are rarely optimal, such as in recommender and information retrieval systems (Yue & Joachims, 2009; Shivaswamy & Joachims, 2012; Radlinski et al., 2008; Dudík et al., 2015). At its core, PO algorithms focus on learning a policy in setups where we lack an explicit reward signal but instead have access to relative preferences between pairs of labels (a weaker constraint). PO has become particularly important in the training of large language models in the form of Reinforcement Learning from Human Feedback (RLHF) which is used to guide policy optimization based on qualitative feedback (Ouyang et al., 2022b; team, 2024). While powerful, this approach is computationally expensive as the reward function and policy are trained separately. To alleviate this, Rafailov et al. (2023) introduces the *Direct Preference Optimization (DPO)* objective, which streamlines the process by jointly training both the reward function and policy.

Akin to traditional likelihood-based optimization approaches, DPO has the added benefit of being end-to-end differentiable. In contrast to likelihood-based learning, however, DPO does not require a set of training labels sampled from the optimal data distribution, instead only assuming that a preference exists between any pair of labels (Bengs et al., 2021). The assumption of optimal labels has been shown to make likelihood-based prone to overfitting to simple patterns (Arpit et al., 2017) making them largely less robust to label noise (Goodfellow et al., 2016). Such a limitation is particularly relevant for traditional CBM training pipelines, which often include data augmentations and sample mislabels that can lead concept labels to differ from the optimal ones. To alleviate this, we extend the DPO objective to CBMs in this work. We find that, as in language and retrieval tasks, training CBMs with our objective alleviates the effect of label noise.

3 Background

In this work, we approach learning CBMs through the use of PO and thus, adapt our notation accordingly.

Concept Bottleneck Models Concept Bottleneck Models (CBMs) (Koh et al., 2020) assume their training sets $(\mathbf{X}, \mathbf{C}, \mathbf{Y})$ are sampled i.i.d from an empirical distribution $\mu(c, x, y)$, where $x \in \mathbf{X}$ are the input features, $c \in \mathbf{C}$ are the binary concepts sets composed of $c = \{c_1, ..., c_k\}$ and $y \in \mathbf{Y}$ are the task labels. Here, we assume that $\mu(c, x, y)$ may not necessarily be the same as $d^*(c, x, y)$, the *optimal data distribution* sampled from the optimal policy π^*. Specifically, we assume that they only differ at the concept level, meaning the empirical distribution's concept labels may be noisy while the task labels are always correct. We argue, however, that this difference is likely in practice as concept-specific noise may be accidentally added during training because of common data augmentations that may occlude concepts (e.g., random crops or shifts; see Figure 1). Moreover, concept-specific noise may further arise naturally due to subjective or fatigued labeling (Sylolypavan et al., 2023; Wei et al., 2024).

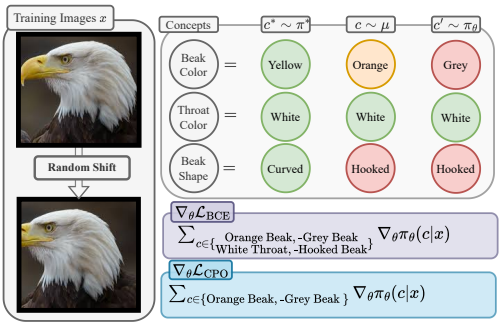


Figure 1: Concept scenarios for CBM: Beak color (orange preferred over red), throat color (green optimal), and beak shape (both incorrect).

CBMs consist of two sub-models. First, a concept predictor, $\pi_\theta : \mathbf{X} \to \mathbf{C}^k$, maps the input x onto an interpretable layer composed of predicted concepts, $\hat{c}$. The concept predictor is usually initialized using a pretrained image encoder k_θ. Then, a task predictor, $f_\phi : \mathbf{C}^k \to \mathbf{Y}^m$, maps these predicted concepts to the task labels $\hat{y}$. In this work, we focus on *jointly trained CBMs*, which are trained end-to-end by minimizing the following objective weighted by a hyperparameter $\lambda \in \mathbb{R}$:

$$\mathcal{L}_{\text{CBM}} = \mathcal{L}_{\text{CE}}(y, f_\phi(c)) + \lambda \mathcal{L}_{\text{BCE}}(c, \pi_\theta(c|x)).$$

The concept objective above optimizes the binary cross entropy (BCE) between the policy's predictions and the empirical data, which is known to be suboptimal under noisy settings (Goodfellow et al., 2016). Due to this sensitivity, we take inspiration from modern PO algorithms, deriving a simple and computationally efficient objective that is equivalent to approximating the concept's posterior distribution and is more robust to noise compared to BCE.

Direct Preference Optimization (DPO) Traditionally, preference optimization using RLHF algorithms (Kaufmann et al., 2023) relies on learning a reward function through the Bradley-Terry preference model (Ouyang et al., 2022a; Kaufmann et al., 2023). Given a preference dataset $(c^w, c^l, x, y) \sim \mu^p$, one can learn a reward function capable of distinguishing preferred concepts c^w from dispreferred ones c^l by optimizing

$$\max_{r_\psi} \mathbb{E}_{(c^w, c^l, x) \sim \mu^p} [\log \sigma(r_\psi(c^w, x) - r_\psi(c^l, x))]$$

Where r_ψ is a parameterized reward function learnt through the optimization process and σ is the sigmoid function. Using this learned reward function, a policy can be trained with any RL algorithm. Most commonly employed is the proximal policy optimization (Schulman et al., 2017) algorithm, which imposes a KL constraint with a prior $\pi_0(c|x)$ on the standard reward maximization objective,

$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mu, c \sim \pi_\theta} [r_\psi(x, c)] - \beta \mathbb{D}_{\text{KL}} \left(\pi_\theta(c|x) \parallel \pi_0(c|x)\right) \tag{1}$$

where β is a hyperparameter controlling the prior's strength. However, this two-step procedure is computationally expensive and unstable. To address this, Rafailov et al. (2023) proposed the Direct Preference Optimization (DPO) algorithm. This shows that the optimal policy for this optimization problem can be expressed as

$$\pi^*(c|x) = \frac{1}{Z(x)} \pi_0(c|x) \exp\left(\frac{1}{\beta} r^*(x, c)\right) \tag{2}$$

where $Z(x) = \sum_c \pi_0(c|x) \exp\left(\frac{1}{\beta} r^*(x, c)\right)$ is the partition function, and $r^*(x, c)$ represents the optimal reward. Consequently, the optimal reward function can be expressed in terms of the optimal

policy:

$$r^*(x, c) = \beta \log \frac{\pi^*(c|x)}{\pi_0(c|x)} + \beta \log Z(x) \tag{3}$$

Using this formulation, Equation 1 simplifies to

$$\max_{\pi_\theta} \mathbb{E}_{(c^w, c^l, x)\sim\mu} \left[\log \sigma \left(\log \frac{\pi_\theta(c^w|x)}{\pi_0(c^w|x)} - \log \frac{\pi_\theta(c^l|x)}{\pi_0(c^l|x)} \right) \right] \tag{4}$$

which is an offline objective that jointly trains the policy and reward functions.

4 Preference Optimization for CBMs

Although one could directly optimize the objective in Equation 4, doing so for CBMs would require a labelled dataset where preferences in concepts are explicitly specified. To circumvent this issue, we can leverage the empirical dataset and state its preference over a concept set sampled from π_θ. The preference over a pair of concepts should hold specifically early on in training where the policy is suboptimal compared to the empirical data. Throughout the rest of this section, we formally describe this algorithm, showing some key similarities and differences between it and $\mathcal{L}_{\text{BCE}}$.

4.1 Concept Bottleneck Preference Optimization

To leverage the DPO objective to learn π_θ, we re-formalize it as an online learning algorithm. We collect negative concept sets by sampling from the policy conditioned on the input image $c' \sim \pi_\theta(c|x)$.[1] We can then compare these negatively sampled concept sets with those sampled from the empirical data $c \sim \mu$ where we assume that the empirical set is *preferred* to the sampled set, that is $c \succ c'$. Note that this is a weaker assumption than that of traditional CBMs, as we only assume a preference over c rather than its correctness. Using this, we can formally introduce the *Concept Preference Optimization (CPO)* objective, an online formulation of Equation 4:

$$\mathcal{L}_{\text{CPO}} = -\mathbb{E}_{\substack{(x,c)\sim\mu \\ c'\sim\pi_\theta}} \left[\log \sigma \left(\log \frac{\pi_\theta(c|x)}{\pi_0(c|x)} - \log \frac{\pi_\theta(c'|x)}{\pi_0(c'|x)} \right) \right]. \tag{5}$$

When used in language modeling, π_0 is defined as the model after a supervised fine-tuning procedure. Here, we train the model from scratch. In practice, we could impose a prior on the concept labels which relate to either the input or the task label e.g., $\pi_0(c|x, y)$. We briefly explore such applications in § 6, but otherwise assume a uniform prior unless otherwise stated, leaving further explorations as future work. These assumptions simplify the CPO algorithm as follows:

Proposition 4.1. *Assuming that $\pi_0(c|x)$ follows a uniform distribution over binary concepts and that concept labels are conditionally independent given the inputs ($c_i \perp c_j \mid x$ for all $i \neq j$), we have:*

$$\mathcal{L}_{CPO} \propto -\mathbb{E}_{c,x\sim D, c'\neq c\sim\pi_\theta} \big[\log(\pi_\theta(c|x)) \big]. \tag{6}$$

A proof for this proposition is given in App. C.1. Simply put, the above states that $\mathcal{L}_{\text{CPO}}$ is proportional to optimizing the binary cross-entropy when π_θ samples concepts that differ from those in the empirical distribution. $\mathcal{L}_{\text{CPO}}$ is proportional to the objective in Equation 6, and not equal, because when the sampled concepts are equal to the empirical ones, the objective is always constant, i.e., $\log \frac{\pi_\theta(c|x)}{\pi_0(c|x)} - \log \frac{\pi_\theta(c'|x)}{\pi_0(c'|x)} = 0$ in Equation 5. Where the equivalence to the log-likelihood when the sampled concepts are not equal to the empirical ones relies on the stated assumptions.

Gradient Analysis Proposition 4.1 highlights a similarity between $\mathcal{L}_{\text{CPO}}$ and $\mathcal{L}_{\text{BCE}}$. Therefore, we can study their gradients to understand their key differences better. Under our previous assumptions,

[1] In practice, we use hard Gumbel-Softmax sampling (Jang et al., 2017) to ensure end-to-end differentiability. In this work, we sample a *single* concept for each image in each iteration, but one could potentially sample multiple per image to increase performance.

we can express the expected gradient of $\mathcal{L}_{\text{CPO}}$ as

$$\begin{aligned}\mathbb{E}[\nabla_\theta \mathcal{L}_{\text{CPO}}] &= \frac{1}{N} \sum_{\substack{(c,x)\sim\mu \\ c'\sim\pi_\theta}} (\pi_\theta(c|x) - 1)\pi_\theta(c'|x)\nabla_\theta k_\theta \\ &= \frac{1}{N} \sum_{(c,x)\sim\mu} (\pi_\theta(c|x) - 1)\left(1 - \pi_\theta(c|x)\right)\nabla_\theta k_\theta\end{aligned}$$

where k_θ refers to the pre-trained image encoder traditionally used to generate concept representations. A full derivation of this equality, which exploits the fact that we only have a nonzero gradient when $c' \neq c$ and thus $\pi(c'|x) = 1 - \pi(c|x)$, is given in App. C.1.

This result shows that $\mathcal{L}_{\text{CPO}}$'s gradient is $\mathcal{L}_{\text{BCE}}$'s gradient weighted by how confident the policy is in the sampled concept. This yields the following bound on the CPO gradient:

Proposition 4.2. *Under the same assumption as Proposition 4.1, the expected gradient norm of $\mathcal{L}_{CPO}$ is a lower bound of $\mathcal{L}_{BCE}$'s expected gradient. That is:*

$$\left\|\mathbb{E}[\nabla_\theta \mathcal{L}_{CPO}]\right\|_2 \leq \left\|\mathbb{E}[\nabla_\theta \mathcal{L}_{BCE}]\right\|_2$$

Proof. As $0 \leq \pi(c|x) \leq 1$, we have that

$$\left\| \frac{\nabla_\theta k_\theta}{N} \sum_{\substack{(c,x)\sim\mu \\ c'\sim\pi(c|x)}} (\pi(c|x) - 1)(1 - \pi(c|x)) \right\|_2 \leq \left\| \frac{\nabla_\theta k_\theta}{N} \sum_{(c,x)\sim\mu} (\pi(c|x) - 1) \right\|_2$$

Notice how the right-hand side is equivalent to $\nabla_\theta \mathcal{L}_{\text{BCE}}$ with equality only holding when $c_i \neq c'_i$ for all i. Thus, we can see that an implication of not assuming the correctness of the concepts is that $\mathcal{L}_{\text{CPO}}$ is more conservative in its gradient updates than $\mathcal{L}_{\text{BCE}}$. This means that $\mathcal{L}_{\text{CPO}}$ has a larger gradient when π_θ is confident in the sampled concepts and more conservative when it is uncertain. A visualization of the differences in the gradients is given in App. D. Next, we discuss the direct implications of these results and the relationship to the improved label noise robustness.

4.2 Noisy Concept Labels

To improve performance, CBMs are traditionally trained by randomly augmenting input images with transformations, such as cropping or blurring, which may obscure the represented concept. As a result, CBMs are often trained with some level of concept noise, regardless of the reliability of the empirical data. Moreover, commonly used benchmark datasets for CBMs, such as CUB (Wah et al., 2011) and AwA2 (Xian et al., 2019), are designed so that their images may not accurately reflect their concept labels. Intuitively, $\mathcal{L}_{\text{CPO}}$ dropping the assumption of correctness towards preferences should help under both noisy and optimal conditions. To illustrate why this is the case, we analyze the gradients of $\mathcal{L}_{\text{CPO}}$ and $\mathcal{L}_{\text{BCE}}$ in the presence of noise.

Empirical Best Gradient To show why $\mathcal{L}_{\text{CPO}}$ is more resilient to noise, we examine both losses' gradients under noisy conditions and compare them to their optimal counterparts. To do so, we first make the assumption that $d^*(c, x)$ is one if x and c are the ground truth concepts in the image and zero otherwise. In this case, given the empirical and optimal distributions μ and d^*, respectively, we can derive the expected gradient that approximates the ground truth as:

$$\begin{aligned}\mathbb{E}_{(c^*,x)\sim d^*}[\nabla_\theta \mathcal{L}] &= \mathbb{E}_{(c,x)\sim\mu}\Big[\frac{d^*(c,x)}{\mu(c,x)} \nabla_\theta \mathcal{L}(c, \pi_\theta(c|x))\Big] \\ &= \mathbb{E}_{(c^*,x)\sim\mu^+}\Big[\nabla_\theta \mathcal{L}(c^*, \pi_\theta(c|x))\Big].\end{aligned}$$

Here, $\frac{d^*(c,x)}{\mu(c,x)}$ is an importance sampling coefficient that equals 1 when c exists in both d^* and μ, and 0 otherwise, as we assume both μ and d^* are deterministic, and $\mu^+ \in \mu$ is the subset containing only optimal concepts. Conversely, $\mu^- \in \mu$ is the subset containing only suboptimal concepts c^-. The resulting gradient on the empirical data is

$$\begin{aligned}\mathbb{E}_{(c,x)\sim\mu}\big[\nabla_\theta \mathcal{L}\big] =& \mathbb{E}_{(c^*,x)\sim\mu^+}\big[\nabla_\theta \mathcal{L}(c^*, \pi_\theta(c^*|x)\big] \\ &+ \mathbb{E}_{(c^-,x)\sim\mu^-}\big[\nabla_\theta \mathcal{L}(c^-, \pi_\theta(c^-|x)\big].\end{aligned}$$

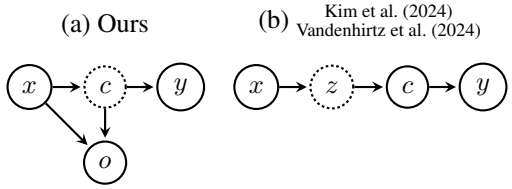


Figure 2: Comparison of graphical models for Bayesian CBMs. Dashed outlines indicate the variable over which an amortized posterior is taken. On the left (a) it can be seen that optimizing a CBM using $\mathcal{L}_{\text{CPO}}$ directly approximates the posterior distribution of the concepts. On the right (b) it can be seen how other methods instead obtain the posterior over a latent variable z.

It is a linear combination between the gradients on the noisy concepts c^- and the optimal ones c^*. This formulation allows us to analyze the difference between the optimal gradient and the gradient produced on a noisy distribution for both $\mathcal{L}_{\text{CPO}}$ and $\mathcal{L}_{\text{BCE}}$:

Theorem 4.3. *The gradient of $\mathcal{L}_{CPO}$ under a constant level of noise is closer in distance to its noise-free counterpart than the gradient of $\mathcal{L}_{BCE}$ under the same noise is to its respective noise-free counterpart. In other words:*

$$\left\|\mathbb{E}_{(c^*,x)\sim d^*}[\nabla_\theta \mathcal{L}_{CPO}] - \mathbb{E}_{(c,x)\sim\mu}[\nabla_\theta \mathcal{L}_{CPO}]\right\|_2 \leq \left\|\mathbb{E}_{(c^*,x)\sim d^*}[\nabla_\theta \mathcal{L}_{BCE}] - \mathbb{E}_{(c,x)\sim\mu}[\nabla_\theta \mathcal{L}_{BCE}]\right\|_2$$

We prove this theorem in App. C.3. Intuitively, Theorem 4.3 says that when examining the difference between optimal and noisy gradients, only terms from noisy observations remain. Thus, according to Proposition C.1, $\mathbb{E}_{c^-\sim\mu^-}[\nabla_\theta \mathcal{L}_{\text{DPO}}] \leq \mathbb{E}_{c^-\sim\mu^-}[\nabla_\theta \mathcal{L}_{\text{BCE}}]$. This implies that the gradient updates of $\mathcal{L}_{\text{CPO}}$ more closely approximate its optimal gradient, resulting in greater robustness to noise.

A simpler explanation lies in the update mechanisms, where CPO only modifies the policy when concepts are incorrectly sampled, creating situations where sampled concepts c' align with c^- and, thus, minimizing noise impact. In contrast, BCE updates continuously unless $\pi_\theta(c|x)$ exactly equals 1 or 0, making it inherently more susceptible to noise. Figure 1 illustrates these results.

4.3 Relationship to Amortized Posterior Approximation

Given their relationship, we seek to understand the fundamental difference between optimizing $\mathcal{L}_{\text{CPO}}$ and $\mathcal{L}_{\text{BCE}}$.

Control as Inference The bottleneck nature of CBMs is similar to that of a Variational Auto-Encoder (VAE) (Kingma & Welling, 2022). Traditionally, such Bayesian methods introduce a "bottleneck" in their inference that is formed by latent variables that are learned in an unsupervised fashion (Doersch, 2016). The graphical model representing a CBM often resembles this relationship, with the key difference that CBMs directly specify the factors within the bottleneck in the form of known concepts. One important outcome of this difference is that CBMs are traditionally trained to optimize the likelihood of the empirical concepts, which is *fundamentally different* from approximating the concept's posterior distribution (Koller & Friedman, 2009). On the other hand, Haarnoja et al. (2017) show that Equation 1 — and Equation 5 by extension — is equivalent to training an amortized posterior approximation of the actions (concepts in our contexts). This derivation relies on introducing an optimality latent variable o whose relationship to x and c is visualized in Figure 2 (Eysenbach & Levine, 2022; Levine, 2018). This optimality variable denotes whether or not the given state-action pair sampled from π is optimal $o = 1$ (c is the best visually represented concept in x) or not $o = 0$. The distribution over this variable is then given as:

$$p(o = 1|x, c) = \exp(r^*(c, x)) \tag{7}$$

where $r^*(c, x) \in (-\infty, 0]$ in our case is an unknown reward function indicating how well a given concept set is represented in an image. Here, one can interpret $p(o = 1|x, c)$ as the probability that the given concept set c is correct, or *optimal*, for input x, and $p(o = 1|x)$ as how optimal, on average, the concept sets sampled from π are for a given x. Given this, a posterior over the concepts is:

$$\pi(c|o = 1, x) = \frac{p(o = 1|c, x)\pi_0(c|x)}{p(o = 1|x)} \tag{8}$$

$$= \frac{1}{Z(o)}\pi_0(c|x)\exp\left(\frac{1}{\beta}r^*(x, c)\right) \tag{9}$$

where $Z(o) = p(o = 1|x)$. This objective is equivalent to that in Equation 2 as $Z(o)$ must be equivalent to $Z(x)$ for $\pi(c|o = 1, x)$ to be a valid probability distribution. Hence, optimizing π_θ using the objective in Equation 1 - and Equation 5 by extension - *directly approximates the optimal concept posterior distribution* where $\pi^*(c|x) = \pi(c|o = 1, x)$.

Table 1: Task and concept performances. The highest and second-highest values in each column are bolded and underlined, respectively.

	CUB		AwA2		CelebA	
	Task Accuracy	**Concept AUC**	**Task Accuracy**	**Concept AUC**	**Task Accuracy**	**Concept AUC**
ProbCBM Sequential	0.742 ± 0.004	0.900± 0.007	0.891 ± 0.003	0.960 ± 0.003	0.302 ± 0.008	**0.878 ± 0.006**
ProbCBM Joint	0.766 ± 0.012	0.943 ± 0.006	0.860 ± 0.017	0.945 ± 0.007	0.288 ± 0.023	0.863 ± 0.005
CoopCBM	0.760 ± 0.004	0.936 ± 0.001	0.888 ± 0.006	0.950 ± 0.003	0.288 ± 0.011	0.878 ± 0.002
CBM BCE	0.753 ± 0.009	0.937 ± 0.001	0.900 ± 0.008	0.959 ± 0.003	0.283 ± 0.007	0.873 ± 0.002
CBM CPO (Ours)	0.800 ± 0.003	**0.952 ± 0.001**	0.915 ± 0.004	**0.971 ± 0.001**	0.310 ± 0.009	0.857 ± 0.003
CEM BCE	0.800 ± 0.003	0.946 ± 0.001	0.889 ± 0.001	0.953 ± 0.000	0.351 ± 0.006	0.875 ± 0.004
CEM CPO (Ours)	**0.807 ± 0.004**	0.931 ± 0.003	**0.917 ± 0.003**	0.965 ± 0.001	**0.352 ± 0.004**	0.853 ± 0.003

5 EXPERIMENTS

We evaluate the task accuracy and mean concept AUC-ROC of models trained with $\mathcal{L}_{\text{CPO}}$ under un-noised and noisy settings. We in addition analyze the intervention performance across baselines.

5.1 BASELINES

We evaluate $\mathcal{L}_{\text{CPO}}$ against $\mathcal{L}_{\text{BCE}}$ on the following CBM-based architectures: (1) standard joint CBMs with sigmoidal concept representations (CBM), Concept Embedding Models (CEMs) (Espinosa Zarlenga et al., 2022), which employ more expressive concept representations to increase model capacity, (3) Coop-CBMs (Sheth & Ebrahimi Kahou, 2024), which use an auxiliary head to improve a CBM's information bottleneck, and (4) ProbCBMs (Kim et al., 2023), which introduce a latent parameter between inputs and concepts (Figure 2). ProbCBMs, in particular, allow us to compare amortizing the concepts' posterior distribution instead of a latent variable's. However, while ProbCBMs traditionally use sequential training, *we jointly train them* to ensure a fair comparison across baselines. We do not evaluate training traditional ProbCBMs with $\mathcal{L}_{\text{CPO}}$ as their loss function optimizes an evidence-lower-bound on the latent variables' posterior, which explicitly requires maximizing concept likelihood. We comment that while $\mathcal{L}_{\text{CPO}}$ introduces a new parameter β we choose not to tune it and set $\beta = 1$ for all experiments. We discuss other hyperparameters and implementation details in App. 5.1.

5.2 NO INTERVENTIONS

Base Performance Table 1 summarizes performance metrics for all datasets and baselines. Our results suggest that training with the $\mathcal{L}_{\text{CPO}}$ objective enhances the base task accuracy of both standard CBMs and CEMs with minimal-to-no-loss in mean concept AUC. Most notably, we observe that, in CUB and AwA2, CPO-trained CBMs match/outperform CEMs trained with BCE, a significant result since these benefits from CPO come *without any* additional parameters or significant computational cost.

Noisy Setting To empirically study $\mathcal{L}_{\text{CPO}}$ under various amounts of noise we randomly flip each training concept label with probability p and report the performance. Figure 3 shows our baselines' task accuracies and concept AUCs as we ablate the label noise probability p across $\{0.1, 0.2, 0.3, 0.4\}$. We observe that, under noisy label conditions, training CBMs and ProbCBMs with $\mathcal{L}_{\text{BCE}}$ leads to a significant drop in both task accuracy and concept AUC (except for AwA2). Interestingly, we see that CEMs trained with $\mathcal{L}_{\text{BCE}}$ are much more resilient to noise in comparison to CBMs. However, we still observe significant drops in concept AUCs in CEMs trained with $\mathcal{L}_{\text{BCE}}$ in all tasks but CelebA. We believe CEMs' more

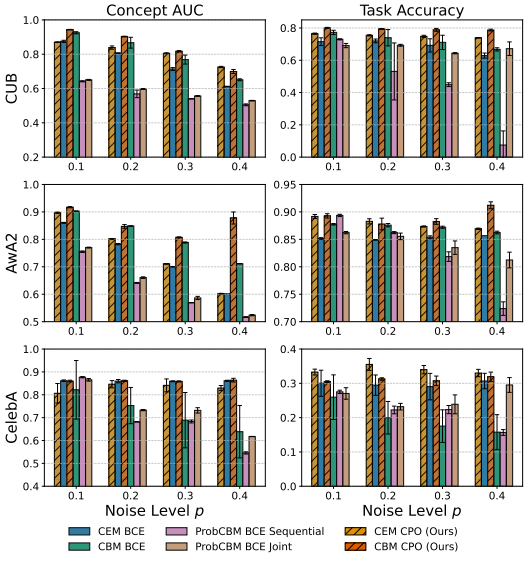


Figure 3: Performance under label noise.

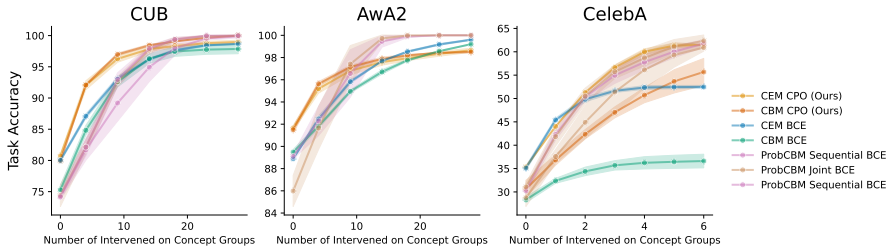


Figure 4: CUB Interventions without added label noise. In CUB and CelebA, $\mathcal{L}_{\text{CPO}}$ models lead to the best intervention performance. While in AwA2, a substantial ($\sim$8-15) number of interventions must be performed for $\mathcal{L}_{\text{BCE}}$-based models to outperform $\mathcal{L}_{\text{DPO}}$ ones.

robust performance in CelebA is due to the bottleneck imposed by this dataset being small (only 6 concepts), making any additional capacity extremely helpful. In contrast, we see that models trained with $\mathcal{L}_{\text{DPO}}$ are very resilient to noise. Specifically, we find that in terms of task accuracy, CBMs trained with $\mathcal{L}_{\text{DPO}}$ are the least affected by noise and largely surpass the performance of CEMs. Moreover, we find that $\mathcal{L}_{\text{DPO}}$-trained models have their concept AUCs better preserved, consistently holding the best or second-best ranks in concept AUC, often attaining significantly better concept AUCs than $\mathcal{L}_{\text{BCE}}$-based models. Most interestingly, we find that even at rather noisy levels ($p = 0.4$), CBMs trained with $\mathcal{L}_{\text{DPO}}$ can outperform more complex models trained with $\mathcal{L}_{\text{BCE}}$ and, in some cases, are largely unaffected by the noise. Overall, we find that using $\mathcal{L}_{\text{DPO}}$ is an effective way to counteract concept label noise.

6 Intervention Performence

Base Interventions A key advantage of CBMs is their ability to improve their task performance through test-time *concept interventions*. An advantage of $\mathcal{L}_{\text{CPO}}$ directly optimizing for the concepts' posterior distribution is that we can obtain accurate uncertainty estimations from the predicted concept values. To test the effectiveness of this uncertainty estimate, we study the effect of interventions when we choose the order in which concepts are intervened on based on their uncertainties, i.e., more uncertain concepts are intervened on first. We do this for similar approaches by using the concept prediction as an uncertainty estimate for CBMs and the determinant of the covariance matrix for ProbCBMs (as done by Kim et al. (2023)).

Figure 4 illustrates the responsiveness of models to interventions. Here, we see that, across all datasets, CEMs and standard CBMs trained with $\mathcal{L}_{\text{CPO}}$ exhibit better accuracies as they are intervened on than their $\mathcal{L}_{\text{BCE}}$ counterparts. This suggests that directly modelling the concept posterior distribution provides better uncertainty estimates, leading to more effective interventions. Additionally, CBMs and CEMs trained with $\mathcal{L}_{\text{CPO}}$ achieve stronger intervention performance than ProbCBMs on CUB, while CEMs using $\mathcal{L}_{\text{CPO}}$ outperform ProbCBMs on CelebA. The only exception is AwA2, where ProbCBMs, on average, still require $\sim$eight interventions before surpassing $\mathcal{L}_{\text{CPO}}$ models.

Noised Interventions In Figure 5 we provide intervention performance for all values of $p \in \{0.1, 0.2, 0.3, 0.3\}$. We find that again even at low noise levels $\mathcal{L}_{\text{CPO}}$ models consistently outperform their $\mathcal{L}_{\text{BCE}}$ counterparts. The one model not holding to that is ProbCBMs which outperform $\mathcal{L}_{\text{DPO}}$ consistently on AwA2. It illustrates the responsiveness of models to interventions. We find that overall, across all datasets, CEMs and standard CBMs trained with $\mathcal{L}_{\text{CPO}}$ exhibit better accuracies as they are intervened on than their $\mathcal{L}_{\text{BCE}}$ counterparts. This suggests that directly modelling the concept posterior distribution provides better uncertainty estimates, leading to more effective interventions. Additionally, CBMs and CEMs trained with $\mathcal{L}_{\text{CPO}}$ achieve stronger intervention performance than ProbCBMs on CUB, while CEMs using $\mathcal{L}_{\text{CPO}}$ outperform ProbCBMs on CelebA. The only exception is AwA2, where ProbCBMs, on average, still require approximately eight interventions before surpassing $\mathcal{L}_{\text{CPO}}$ models.

Learning on Streaming Data A byproduct of $\mathcal{L}_{\text{CPO}}$ optimizing for an approximate posterior is its ability to leverage a prior. So far, we have assumed a uniform prior over concepts, but here we explore adjusting it. One key benefit of CBMs is that practitioners can scrutinize concept representations at test time, enhancing trust, and accuracy and enable the ability to collect new training data through interventions. That is when a CBM is intervened on it obtains a new concept label that can be used to improve the system further (an idea that has been explored in other fields (Stephan et al., 2024;

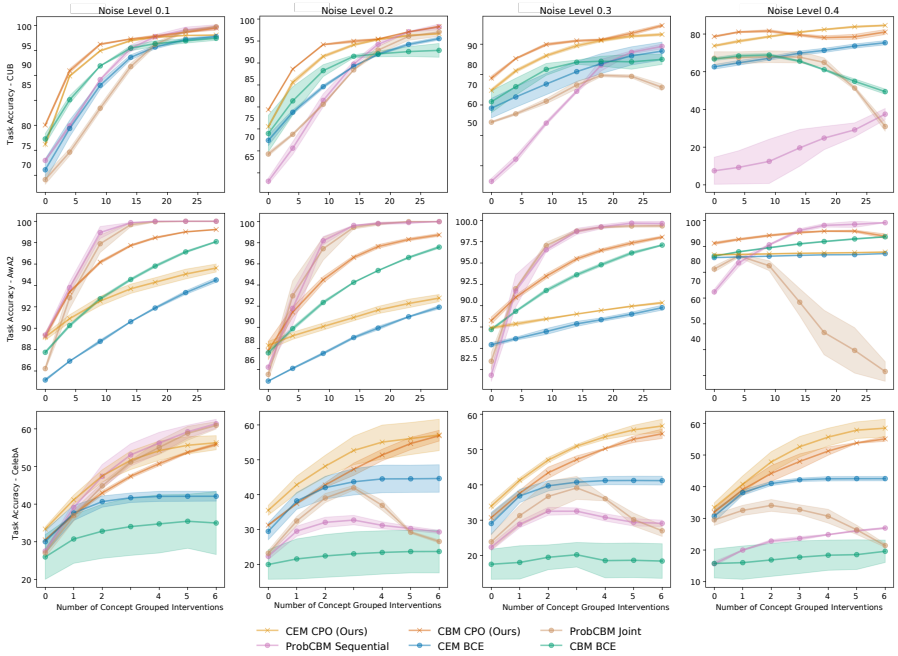


Figure 5: Intervention performance for all noise levels. We find that overall methods trained with $\mathcal{L}_{\text{DPO}}$ yield better intervention performance under noise. These findings are specifically relevant to CUB and Celeb where we see all other methods are harshly impacted.

Shi et al., 2024)). For this, we first partition the CUB training data into four equal-sized blocks, of which we use the first block ($= 25\%$ of the total data) to train an initial checkpoint using a CBM jointly trained using $\mathcal{L}_{\text{DPO}}$ on the task labels y and concepts c. Thereafter, we analyze training on the remaining data blocks *only* using concept labels in three different ways using $\mathcal{L}_{\text{BCE}}$, $\mathcal{L}_{\text{CPO}}$ with a uniform prior and $\mathcal{L}_{\text{CPO}}$ with the previous checkpoint as the prior. The main idea is that a priori can help prevent the model from drifting too far from the policy trained jointly with the task predictor f_ϕ. Figure 6 evaluates models using $k\% \in \{50\%, 100\%\}$ of total concepts. Curiously, we find that $\mathcal{L}_{\text{CPO}}$ using a uniform prior performs worse when using more concept labels. We believe this is due to the policy drifting further from that of the initial checkpoint. We find that indeed using a prior one can alleviate this drift enabling $\mathcal{L}_{\text{CPO}}$ to use all new concept labels. While here we find $\mathcal{L}_{\text{BCE}}$ underperforms $\mathcal{L}_{\text{DPO}}$, in App. 6 we show this gap narrows—though not fully closes—when the initial policy is trained with $\mathcal{L}_{\text{BCE}}$.

7 Conclusion

We present a DPO-inspired training objective for CBMs called $\mathcal{L}_{\text{CPO}}$. Our loss directly optimizes for the concept's posterior distribution, with concept representations that explicitly encode uncertainty, leading to improved intervention performance. We provide analysis demonstrating that $\mathcal{L}_{\text{CPO}}$ exhibits greater robustness to noise compared to $\mathcal{L}_{\text{BCE}}$ and empirically show that a simple CBM trained with the $\mathcal{L}_{\text{CPO}}$ objective can consistently outperform competing methods without any additional parameters. Moreover, our experiments complement our analysis on $\mathcal{L}_{\text{CPO}}$'s behaviour under noise, showing that $\mathcal{L}_{\text{CPO}}$ yields better concept AUC and task accuracy than BCE-based models while better maintaining its intervention performance. Furthermore, we demonstrate how the $\mathcal{L}_{\text{CPO}}$ objective's prior can be leveraged to learn more efficiently from streaming data. Ultimately, $\mathcal{L}_{\text{CPO}}$ offers numerous benefits for CBM and CBM-like methods with minimal computational overhead.

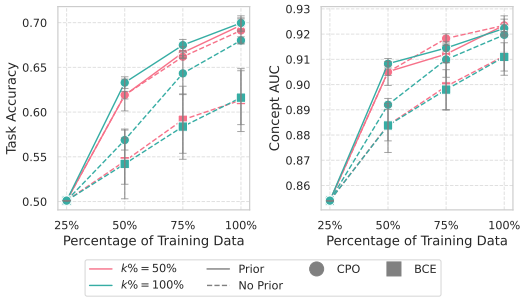


Figure 6: Updating a CBM with streaming concept labels (no task labels).

REFERENCES

Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 233–242. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/arpit17a.html.

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.

Viktor Bengs, Robert Busa-Fekete, Adil El Mesaoudi-Paul, and Eyke Hüllermeier. Preference-based online learning with dueling bandits: A survey, 2021. URL https://arxiv.org/abs/1807.11398.

Kushal Chauhan, Rishabh Tiwari, Jan Freyberg, Pradeep Shenoy, and Krishnamurthy Dvijotham. Interactive concept bottleneck models. *arXiv preprint arXiv:2212.07430*, 2022.

Carl Doersch. Tutorial on variational autoencoders, 2016.

Miroslav Dudík, Katja Hofmann, Robert E. Schapire, Aleksandrs Slivkins, and Masrour Zoghi. Contextual dueling bandits. In Peter Grünwald, Elad Hazan, and Satyen Kale (eds.), *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pp. 563–587, Paris, France, 03–06 Jul 2015. PMLR. URL https://proceedings.mlr.press/v40/Dudik15.html.

Mateo Espinosa Zarlenga, Barbiero Pietro, Ciravegna Gabriele, Marra Giuseppe, Francesco Giannini, Michelangelo Diligenti, Shams Zohreh, Precioso Frederic, Stefano Melacci, Weller Adrian, et al. Concept embedding models: Beyond the accuracy-explainability trade-off. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21400–21413. Curran Associates, Inc., 2022.

Mateo Espinosa Zarlenga, Katie Collins, Krishnamurthy Dvijotham, Adrian Weller, Zohreh Shams, and Mateja Jamnik. Learning to Receive Help: Intervention-Aware Concept Embedding Models. *Advances in Neural Information Processing Systems*, 36, 2023.

Benjamin Eysenbach and Sergey Levine. Maximum entropy rl (provably) solves some robust rl problems, 2022. URL https://arxiv.org/abs/2103.06257.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies, 2017. URL https://arxiv.org/abs/1702.08165.

Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. Addressing leakage in concept bottleneck models. In *Advances in Neural Information Processing Systems*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax, 2017. URL https://arxiv.org/abs/1611.01144.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback, 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.

Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. Probabilistic concept bottleneck models. In *International Conference on Machine Learning*, pp. 16521–16540. PMLR, 2023.

Sangwon Kim, Dasom Ahn, Byoung Chul Ko, In su Jang, and Kwang-Ju Kim. Eq-cbm: A probabilistic concept bottleneck with energy-based models and quantized vectors, 2024. URL https://arxiv.org/abs/2409.14630.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pp. 5338–5348. PMLR, 2020.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. ISBN 0262013193.

Aviral Kumar, Joey Hong, Anikait Singh, and Sergey Levine. When should we prefer offline reinforcement learning over behavioral cloning?, 2022. URL https://arxiv.org/abs/2204.05618.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review, 2018.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Alexandra Sasha Luccioni and David Rolnick. Bugs in the data: how imagenet misrepresents biodiversity. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i12.26682. URL https://doi.org/10.1609/aaai.v37i12.26682.

Ričards Marcinkevičs, Sonia Laguna, Moritz Vandenhirtz, and Julia E Vogt. Beyond concept bottleneck models: How to make black boxes intervenable? *arXiv preprint arXiv:2401.13544*, 2024.

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.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022a.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022b. URL https://arxiv.org/abs/2203.02155.

Eleonora Poeta, Gabriele Ciravegna, Eliana Pastor, Tania Cerquitelli, and Elena Baralis. Concept-based explainable artificial intelligence: A survey. *arXiv preprint arXiv:2312.12936*, 2023.

Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pp. 43–52, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781595939913. doi: 10.1145/1458082.1458092. URL https://doi.org/10.1145/1458082.1458092.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2023.

Naveen Raman, Mateo Espinosa Zarlenga, and Mateja Jamnik. Understanding Inter-Concept Relationships in Concept-Based Models. In *International Conference on Machine Learning*. PMLR, 2024.

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 11702–11716. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/60ce36723c17bbac504f2ef4c8a46995-Paper.pdf.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015. URL https://arxiv.org/abs/1409.0575.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Ivaxi Sheth and Samira Ebrahimi Kahou. Auxiliary losses for learning generalizable concept-based models. *Advances in Neural Information Processing Systems*, 36, 2024.

Lucy Xiaoyang Shi, Zheyuan Hu, Tony Z. Zhao, Archit Sharma, Karl Pertsch, Jianlan Luo, Sergey Levine, and Chelsea Finn. Yell at your robot: Improving on-the-fly from language corrections, 2024.

Sungbin Shin, Yohan Jo, Sungsoo Ahn, and Namhoon Lee. A closer look at the intervention procedure of concept bottleneck models. *arXiv preprint arXiv:2302.14260*, 2023.

Pannaga Shivaswamy and Thorsten Joachims. Online structured prediction via coactive learning, 2012. URL https://arxiv.org/abs/1205.4213.

David Steinmann, Wolfgang Stammer, Felix Friedrich, and Kristian Kersting. Learning to intervene on concept bottlenecks. *arXiv preprint arXiv:2308.13453*, 2023.

Moritz Stephan, Alexander Khazatsky, Eric Mitchell, Annie S Chen, Sheryl Hsu, Archit Sharma, and Chelsea Finn. Rlvf: Learning from verbal feedback without overgeneralization, 2024. URL https://arxiv.org/abs/2402.10893.

Aneeta Sylolypavan, Derek Sleeman, Honghan Wu, and Malcolm Sim. The impact of inconsistent human annotations on ai driven clinical decision making. *NPJ Digital Medicine*, 6(1):26, 2023.

Llama 3 team. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Moritz Vandenhirtz, Sonia Laguna, Ričards Marcinkevičs, and Julia E Vogt. Stochastic concept bottleneck models. *arXiv preprint arXiv:2406.19272*, 2024.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Caltech ucsd birds. Technical report, 2011.

Yishu Wei, Yu Deng, Cong Sun, Mingquan Lin, Hongmei Jiang, and Yifan Peng. Deep learning with noisy labels in medical prediction problems: a scoping review. *Journal of the American Medical Informatics Association*, pp. ocae108, 2024.

Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265, 2019. doi: 10.1109/TPAMI.2018.2857768.

Xinyue Xu, Yi Qin, Lu Mi, Hao Wang, and Xiaomeng Li. Energy-based concept bottleneck models: unifying prediction, concept intervention, and conditional interpretations. *arXiv preprint arXiv:2401.14142*, 2024.

Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19187–19197, 2023.

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.

Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 1201–1208, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553527. URL https://doi.org/10.1145/1553374.1553527.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *ICLR 2022 Workshop on PAIR2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2023.

A Implementation Details

A.1 Tuning

We employ a ResNet34 (He et al., 2015) as the backbone image encoder k_θ, pretrained on ImageNet-1k (Russakovsky et al., 2015). Following standard procedures, we apply random cropping and flipping to a portion of the images during training. This augmentation process may introduce non-zero noise levels, as some concepts could be removed from the images after transformation. We use a batch size of 512 for the Celeb dataset and 256 for for CUB and AwA2. We train all models using RTX8000 Nvidia-GPU. In all datasets we train for up to 200 epochs and early stop if the validation loss has not improved in 15 epochs. For fair evaluation across methods, we tune the learning rate for CEMs, CBMs, and ProbCBM. Specifically, for CUB and AwA2 datasets, we explore learning rates $\in \{0.1, 0.01\}$, while for CelebA, we expand the search to $\in \{0.1, 0.01, 0.05, 0.005\}$ due to the observed instability of CEMs at higher learning rates. Additionally, we set the hyper-parameter $\lambda \in \{1, 5, 10\}$ for all methods. For CEMs and models trained using $\mathcal{L}_{\text{DPO}}$, we found RandInt beneficial, which randomly intervenes on 25% of the concepts during training. ProbCBM introduce a few extra hyperparameters which in this work we did not tune and directly use the hyperparameters provided by the original authors. Similar to other models, ProbCBM employs RandInt at 50%, making it particularly sensitive to interventions, especially in concept-complete tasks such as AwA2 and CUB. The only model for which we tune additional hyper-parameters is Coop-CBM, where we adjust the weight parameter for the auxiliary loss we discuss more in detail in App B.1. All experiments are run using a forked version of the Github[2] repository used by Espinosa Zarlenga et al. (2022).

B Datasets

CUB Wah et al. (2011) In CUB we use the standard dataset used in Koh et al. (2020) made up of $k = 112$ concept annotations representing bird attributes (e.g., beak type, wing color) and use the bird class ($m = 200$) as the downstream task. Our only departure from Koh et al. (2020) is that we group the concepts into 28 semantic concept groups, following Espinosa Zarlenga et al. (2022). We use the same image processing as in (Koh et al., 2020) and by randomly flipping and cropping some images during training. The final dataset is composed of $\sim$ 6,000 RGB images of dimension $(3, 299, 299)$ and split into a standard 70%-10%-20% train-validation-test split.

AwA2 Xian et al. (2019) For AwA2 we use the same data processing as Xu et al. (2024). These are made up of where the $k = 85$ concepts correspond to visual animal attributes (e.g., has wings, has claws) which are grouped into 28 semantic concept groups. We apply standard rotation and cropping augmentations throughout training and use the standard 70%-10%-20% train-validation-test split.

CelebA Liu et al. (2015) For this dataset, we closely follow the data processing done by Espinosa Zarlenga et al. (2022), where they select the 8 most balanced attributes out of a total of 40 binary attributes. Where they generate $m = 256$ classes by assign them a value based on the base-10 representation of their attribute label. We construct the incomplete concept set using the same 6 attributes selected by Espinosa Zarlenga et al. (2022). We follow the same subsampling procedure as Espinosa Zarlenga et al. (2022) and randomly select $\frac{1}{12}$th of the images for training. This results in a final dataset composed of 16,900 RGB images where we use the same 70%-10%-20% train-validation-test split.

B.1 Coop CBM

Here, we briefly outline the training procedure for Coop-CBM, which we found to perform similarly to CBMs trained with $\mathcal{L}_{\text{BCE}}$. Similar to other methods, we tune the learning rate and the concept loss weight λ. However, Coop-CBM is the only model for which we conduct more extensive hyper-parameter tuning, as we observed minimal differences between it and a standard CBM trained with $\mathcal{L}_{\text{BCE}}$. Specifically, we tune the additional hyper-parameter $\gamma \in \{0.01, 1, 5, 10\}$, which controls the strength of the auxiliary head[3]. In our setup, the optimal values for γ were found to be $\gamma = 5$ for

[2] https://github.com/mateoespinosa/cem

[3] Referred to as β in their work, but we change the notation to avoid confusion with our β parameter.

CUB, $\gamma = 10$ for AwA2, and $\gamma = 0.01$ for CelebA. We observed negligible differences between Coop-CBM and standard CBMs in terms of base and intervention performance (see Figure 7), except for in AwA2 where it improves intervention performance but outperforms ProbCBM at higher number of interventions. As a result, we did not include Coop-CBM in the remaining experiments.

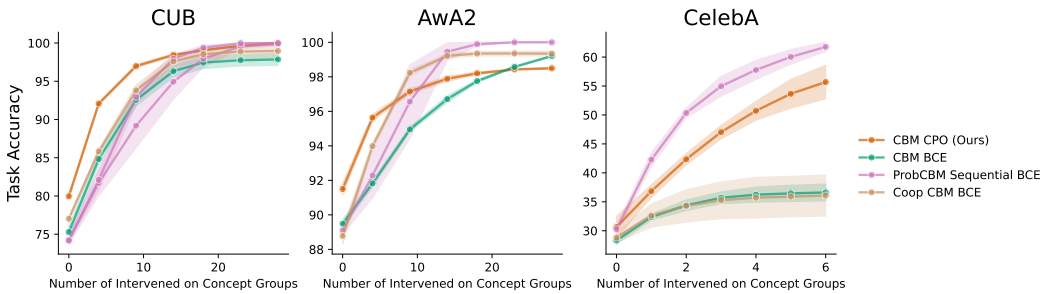


Figure 7: Intervention performance including Coop CBMs.

C Analysis

While prior work has shown that under mild conditions, offline RL performs (equivalent CPO) better than likelihood-based training under noisy labels (Kumar et al., 2022; Rashidinejad et al., 2021), it still does not fully answer the question as to why specifically CPO should perform better in our context.

C.1 Gradient Derivations

Assumptions: For all derivations, we assume binary labels. The conditional independence of the concepts i.e., $c_i \perp c_j | x \; \forall \; i \neq j$, and that π_0 follows a uniform distribution. Additionally, to reduce redundancy, in all gradient derivations, we drop the ∇k_θ as it does not affect the gradients of the loss functions.

Derivation of the CPO objective

$$\mathcal{L}_{\text{CPO}} = -\mathbb{E}_{c,x\sim D, c'\sim\pi_\theta(c|x)}[\log\sigma(\log\pi_\theta(c|x) - \log\pi_\theta(c'|x)] \tag{10}$$

$$= -\mathbb{E}_{c,x\sim D, c'\sim\pi_\theta(c|x)}[\log\sigma(\frac{\pi_\theta(c|x)}{\pi_\theta(c'|x)})] \tag{11}$$

$$= -\mathbb{E}_{c,x\sim D, c'\sim\pi_\theta(c|x)}[\log 1 - \log(1+\exp(-\log\frac{\pi_\theta(c|x)}{\pi_\theta(c'|x)}))] \tag{12}$$

$$= \mathbb{E}_{c,x\sim D, c'\sim\pi_\theta(c|x)}[\log(1+\exp(-\log(\frac{\pi_\theta(c|x)}{\pi_\theta(c'|x)}))] \tag{13}$$

$$= \mathbb{E}_{c,x\sim D, c'\sim\pi_\theta(c|x)}[\log(1+\frac{\pi_\theta(c'|x)}{\pi_\theta(c|x)})] \tag{14}$$

$$\tag{15}$$

Thus, in this case, we can see if c' is not equivalent to c, this loss reduces to cross-entropy.

$$\mathcal{L}_{\text{CPO}} = \mathbb{E}_{c,x\sim D, c'\neq c\sim\pi_\theta}[\log(\frac{\pi_\theta(c|x) + (1-\pi_\theta(c|x))}{\pi_\theta(c|x)})] \tag{16}$$

$$= \mathbb{E}_{c,x\sim D, c'\neq c\sim\pi_\theta}[\log(\frac{1}{\pi_\theta(c|x)})] \tag{17}$$

$$= -\mathbb{E}_{c,x\sim D, c'\neq c\sim\pi_\theta}[\log(\pi_\theta(c|x))] \tag{18}$$

$$\tag{19}$$

Otherwise, it reduces to a constant:

$$\mathcal{L}_{\text{CPO}} = \mathbb{E}_{c,x\sim D,c'=c\sim\pi_\theta}[\log(\frac{\pi_\theta(c|x)+\pi_\theta(c|x))}{\pi_\theta(c|x)})] \tag{20}$$

$$= -\log(\frac{1}{2}) \tag{21}$$

$$\tag{22}$$

CPO Gradient Derivation

$$\nabla_\theta\mathcal{L}_{\text{CPO}} = \mathbb{E}_{c,x\sim D,c'\neq c\sim\pi_\theta}[\nabla_\theta \log(\frac{\pi_\theta(c|x)+(1-\pi_\theta(c'|x))}{\pi_\theta(c|x)})] \tag{23}$$

$$\tag{24}$$

Due to the gradient being zero when $c = c'$, the expected gradient of the CPO objective simplifies to:

$$\nabla_\theta\mathcal{L}_{\text{CPO}} = \frac{1}{N}\sum_{(c,x)\sim\mu,c'\neq c\sim\pi_\theta(c|x)}(\pi_\theta(c|x)-1)\pi_\theta(c'|x) \tag{25}$$

$$= \frac{1}{N}\sum_{(c,x)\sim\mu,c'\neq c\sim\pi_\theta(c|x)}(\pi_\theta(c|x)-1)\left(1-\pi_\theta(c|x)\right) \tag{26}$$

That is, the CPO objective only takes a gradient step for sampled concepts that do not equal the empirical concepts.

C.2 Bounding the gradients

Proposition C.1. *The expected gradient given by $\mathcal{L}_{CPO}$ under binary labels is strictly less than or equal to the gradient of the $\mathcal{L}_{BCE}$*

Proof: This proof relies strictly on the notion that $1 - \pi(c|x) \leq 1$ thus:

$$\frac{1}{N}\sum_{(c,x)\sim\mu,c'\sim\pi_\theta(c|x)}(\pi_\theta(c|x)-1)(1-\pi_\theta(c|x)) \leq \frac{1}{N}\sum_{(c,x)\sim\mu}(\pi_\theta(c|x)-1) \tag{27}$$

Observe how the right-hand side is equivalent to the expected cross-entropy loss. The above proposition also takes into account the maximum gradient possible for the $\mathcal{L}_{\text{CPO}}$, which is when $c_i = c'_i$ for all i

C.3 Comparing the gradients under noise

Theorem C.2. *The expected squared difference between the optimal gradient and one computed under noisy labels for direct preference optimization is less than or equal to that for binary cross entropy.*

Proof: The optimal gradient to take under noisy labels is given by:

$$\mathbb{E}_{(c^*,x)\sim d}[\nabla_\theta\mathcal{L}] = \mathbb{E}_{(c,x)\sim\mu}[\frac{d(c,x)}{\mu(c,x)}\nabla_\theta\mathcal{L}(c,\pi_\theta(c|x)] \tag{28}$$

$$= \mathbb{E}_{(c,x)\sim\mu}[\frac{d(c,x)}{\mu(c,x)}\nabla_\theta\mathcal{L}(c,\pi_\theta(c|x)] \tag{29}$$

$$= \mathbb{E}_{(c^*,x)\sim\mu^+}[\nabla_\theta\mathcal{L}(c^*,\pi_\theta(c|x)] \tag{30}$$

$$\tag{31}$$

We observe that when we do not adjust for the importance weight, the gradient under noisy labels is:

$$\mathbb{E}_{(c,x)\sim\mu}[\nabla_\theta \mathcal{L}] = \mathbb{E}_{(c^*,x)\sim\mu^+}[\nabla_\theta \mathcal{L}(c^*, \pi_\theta(c^*|x)] + \mathbb{E}_{(c^-,x)\sim\mu^-}[\nabla_\theta \mathcal{L}(c^-, \pi_\theta(c^-|x)] \tag{32}$$

$$\tag{33}$$

Thus the difference in the expected value of the gradients is:

$$(\mathbb{E}_{(c^*,x)\sim d}[\nabla_\theta \mathcal{L}] - \mathbb{E}_{(c,x)\sim\mu}[\nabla_\theta \mathcal{L}])^2 = \mathbb{E}_{(c^-,x)\sim\mu^-}[\nabla_\theta \mathcal{L}(c^-, \pi_\theta(c^-|x)] \tag{34}$$

Therefore, using Proposition C.1 we can observe that :

$$\mathbb{E}_{(c^-,x)\sim\mu^-}[\nabla_\theta \mathcal{L}_{\text{CPO}}(c^-, \pi_\theta(c^-|x)] \leq \mathbb{E}_{(c^-,x)\sim\mu^-}[\nabla_\theta \mathcal{L}_{\text{BCE}}(c^-, \pi_\theta(c^-|x)] \tag{35}$$

And thus:

$$(\mathbb{E}_{(c^*,x)\sim d}[\nabla_\theta \mathcal{L}_{\text{CPO}}] - \mathbb{E}_{(c,x)\sim\mu}[\nabla_\theta \mathcal{L}_{\text{CPO}}])^2 \leq (\mathbb{E}_{(c^*,x)\sim d}[\nabla_\theta \mathcal{L}_{\text{BCE}}] - \mathbb{E}_{(c,x)\sim\mu}[\nabla_\theta \mathcal{L}_{\text{BCE}}])^2 \tag{36}$$

D GRADIENT VISUALIZATIONS:

We empirically verify the results posed in Theorem 4.3. For this, we train a standard CBM where the total loss function is $\mathcal{L}_{\text{total}} = \frac{1}{2}(\mathcal{L}_{\text{CPO}} + \mathcal{L}_{\text{BCE}})$ and *do not* optimize the task loss. We train this model over 100 gradient steps and visualize their gradients throughout training. Where the optimal gradient for each loss $\mathcal{L}^*$ is computed using the empirical concepts and the full loss $\mathcal{L}^-$ is computed over both noisy and non-noisy data points. To minimize the effects of noise on the labelled data and gain a better approximation, we explicitly *do not* augment the data in any way. Figure 8 visualizes these results for $p \in \{0.1, 0.3\}$, which empirically confirms the proposed theoretical results showing how even under low amounts of noise $p = 0.1$ $\mathcal{L}_{\text{CPO}}$ is a better approximation to its optimal gradient when compared to $\mathcal{L}_{\text{BCE}}$. We find that in higher noise settings $p = 0.3$, $\mathcal{L}^-_{\text{CPO}}$ deviates substantially less to $\mathcal{L}^*_{\text{CPO}}$ compared to $\mathcal{L}^-_{\text{BCE}}$ against $\mathcal{L}^*_{\text{BCE}}$. This difference is specifically evident early on in training. We hypothesize that providing better gradients early on in training potentially improves the generalization of the model being a possible explanation for the improved performance of $\mathcal{L}_{\text{CPO}}$ under noise seen in the empirical evaluation.

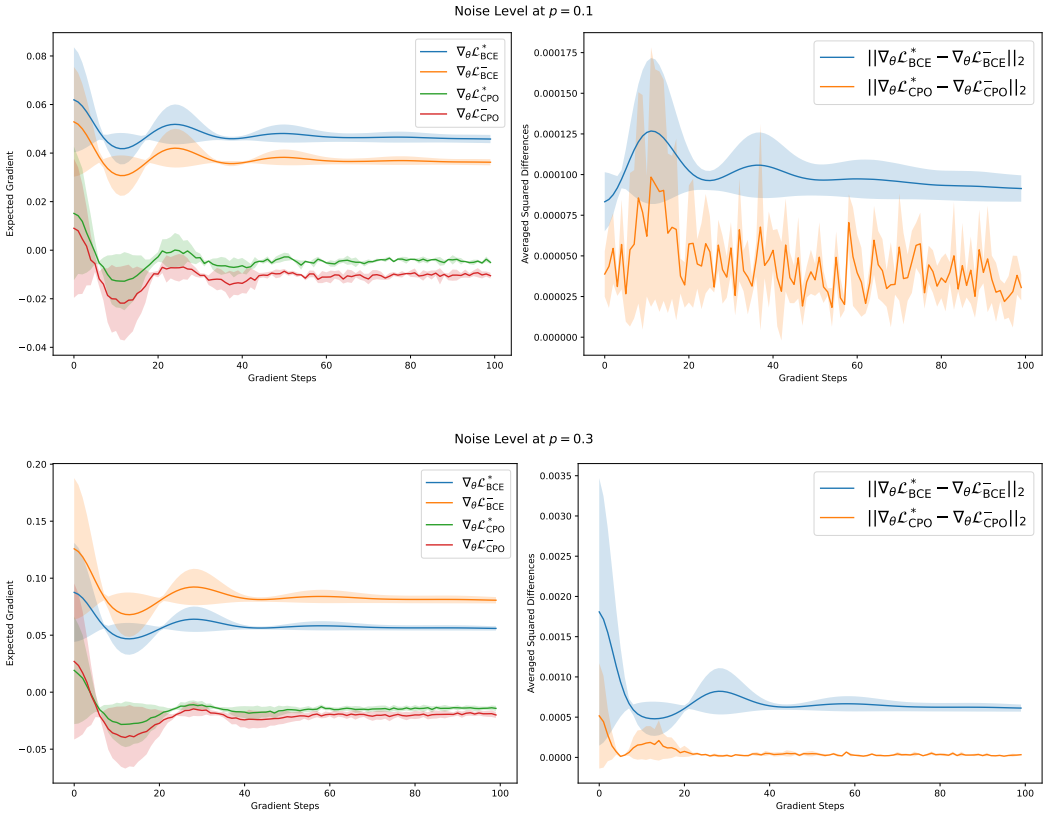
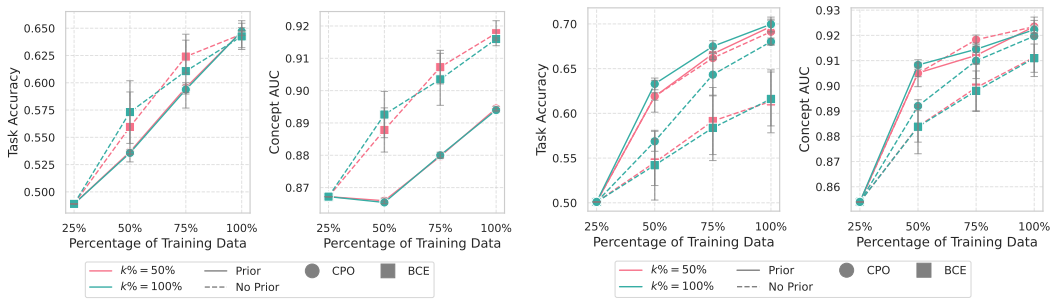


Figure 8: Visualization of noisy (indicated by a $^-$) and optimal (indicated with a *) gradients for $\nabla_\theta \mathcal{L}_{\text{CPO}}$ and $\nabla \mathcal{L}_{\text{BCE}}$. We can observe that even in low noise settings $p = 0.1$, $\mathcal{L}_{\text{CPO}}$ better approximates its optimal gradient, with this difference growing as the noise level increases, especially at the beginning of training. We note that while visually it may seem that the squared difference for $\mathcal{L}_{\text{CPO}}$ is smaller for $p = 0.3$ than that for $p = 0.1$, this is mainly due to scale.

E ADDITIONAL CONTINUAL EXPERIMENTS

Here, we examine the impact of using a model trained with $\mathcal{L}_{\text{BCE}}$ as the starting point for the experiments in § 6. Figure 9 compares the performance of CBMs trained on streaming data when initialized with $\mathcal{L}_{\text{BCE}}$ (left) versus $\mathcal{L}_{\text{DPO}}$ (right, same as Figure 6). Overall, we find that updating a $\mathcal{L}_{\text{BCE}}$-initialized model with $\mathcal{L}_{\text{BCE}}$ yields the best results for $\mathcal{L}_{\text{BCE}}$. However, while this improves performance, it still falls short of the results achieved when both initialization and training are done with $\mathcal{L}_{\text{DPO}}$. We note in the leftmost plot, we exclude the $\mathcal{L}_{\text{DPO}}$ model updated without a prior to improve the clarity of the plot, but note that we find it yields approximately equal results to training with a prior.



((a)) Initialized using a $\mathcal{L}_{\text{BCE}}$ model.

((b)) Initialized using a $\mathcal{L}_{\text{DPO}}$ model.

Figure 9: Task Accuracy/Concept AUC vs the percentage of data the model has been trained on. We find that updating models initialized with a $\mathcal{L}_{\text{BCE}}$ policy yields improved results for $\mathcal{L}_{\text{BCE}}$ with detrimental ones for $\mathcal{L}_{\text{CPO}}$ (A). In (B) we again visualize the result for updating the models using a $\mathcal{L}_{\text{DPO}}$- initialized policy (same as Figure 6). We find that the best result is given by using $\mathcal{L}_{\text{DPO}}$ to update a $\mathcal{L}_{\text{DPO}}$- initialized policy