A Details of the Intervention Procedure

In Algorithm 1 we describe the standard process of performing interventions in concept-based models using an intervention policy.

Algorithm 1 Intervention Algorithm

Inputs:
    T (total number of interventions)
    π (intervention policy, which takes the concepts as input)
    $\hat{c}$ (concepts predicted by the concept encoder)
$\tilde{c} \leftarrow \hat{c}$ ▷ output of the concept encoder, g
for $t \in \{0, \ldots, T-1\}$ **do**
    $i \leftarrow \pi(\tilde{c})$ ▷ i is the concept that we want the user to intervene on
    $\tilde{c}_i \leftarrow c_i$ ▷ replace the ith concept in $\tilde{c}$ with its ground truth value c_i
end for
return $\tilde{y} = f(\tilde{c})$ ▷ updated class prediction after all interventions have been performed

In Algorithm 2 we describe the procedure used in our setup, which realigns unintervened concepts following an intervention step. We use this algorithm to compute the loss for training the realignment model.

Algorithm 2 Realignment Model Training Loss

Inputs:
    T (total number of interventions)
    π (intervention policy, which takes the concepts as input)
    $\hat{c}$ (concepts predicted by the concept encoder)
$\tilde{c} \leftarrow \hat{c}$ ▷ output of the concept encoder, g
$\kappa_{-1} \leftarrow \hat{c}$ ▷ initialize realigned concepts
$\mathcal{L} \leftarrow 0$ ▷ initialize loss
for $t \in \{0, \ldots, T-1\}$ **do**
    $i \leftarrow \pi(\kappa_{t-1})$ ▷ i is the concept that we want the user to intervene on
    $\tilde{c}_i \leftarrow c_i$ ▷ replace the ith concept in $\tilde{c}$ with its ground truth value c_i
    $\kappa_t \leftarrow u(\tilde{c})$ ▷ output of realignment model
    $\mathcal{L} \leftarrow \mathcal{L} + \mathrm{CE}(\kappa_t, c)$ ▷ aggregate loss
end for
return $\mathcal{L}/T$ ▷ average loss across all intervention steps

B Comparison Between Random and UCP Policies

In this section, we compare the classification accuracies achieved by following the random and UCP intervention policies on the three datasets, respectively.

In Fig. 10 we show that the UCP policy is superior across all datasets, and is therefore our default policy across all experiments in this study.

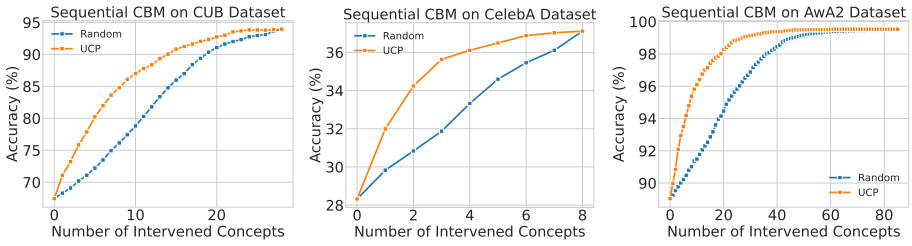

Fig. 10: Comparison between accuracy under UCP and Random intervention policies. UCP is superior in all three datasets.

C Additional Results on IntCEMs

In this section, we report the performance of posthoc concept realignment on the intervention-aware CEMs (IntCEMs) on the CelebA and AwA2 datasets to supplement the results in Section 4.3. In Fig. 11 we show that concept realignment improves the performance of the SoTA approach in both datasets.

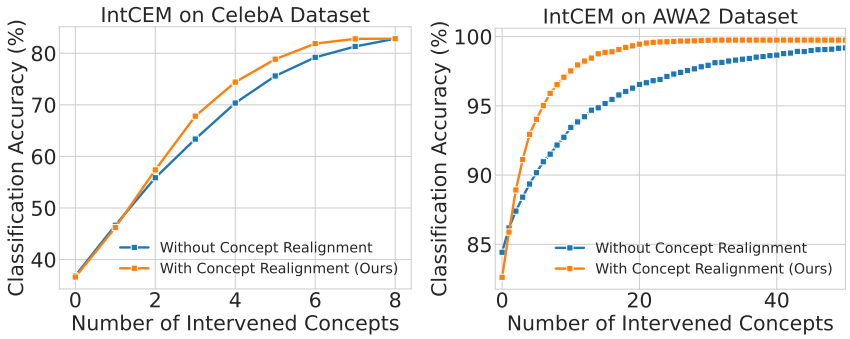

Fig. 11: Classification accuracy with and without posthoc concept realignment in intervention-aware CEMs. In both cases, concept realignment improves performance of the base IntCEM model.

D Additional Results

Here, we report the area under the curve of concept prediction loss and classification accuracy using three different random seeds. It can be seen in Tables 2 and 3 that concept realignment consistently improves performance on both metrics.

Table 2: Area Under Curve (AUC) of Concept Prediction Loss and Classification Accuracy with/without CIRM for three random seeds on the CUB dataset. We use the same backbone for sequential and independent CBMs. CIRM improves performance across all models and runs.

Base Model	**Realigned**	**Concept Loss AUC ↓**			**Accuracy AUC ↑**		
		Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
Sequential CBM	×	6.72	7.11	6.77	2460.86	2394.1	2444.08
	✓	**3.16**	**3.16**	**3.24**	**2510.48**	**2460.41**	**2501.08**
Independent CBM	×	6.72	7.11	6.77	2653.37	2652.75	2652.47
	✓	**3.16**	**3.16**	**3.24**	**2678.09**	**2675.04**	**2675.48**
Joint CBM	×	5.93	5.84	5.89	2580.28	2533.56	2591.32
	✓	**3.67**	**3.49**	**3.58**	**2608.89**	**2559.93**	**2622.53**
CEM	×	5.99	13.19	6.50	2521.31	1681.97	2579.84
	✓	**3.21**	**6.66**	**3.43**	**2558.07**	**1762.58**	**2617.25**

Table 3: Area Under Curve (AUC) of Concept Prediction Loss and Classification Accuracy with/without CIRM for three random seeds on the CelebA dataset. We use the same backbone for sequential and independent CBMs. CIRM improves performance across all models and runs.

Base Model	**Realigned**	**Concept Loss AUC ↓**			**Accuracy AUC ↑**		
		Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
Sequential CBM	×	1.59	1.64	1.65	281.09	279.64	279.47
	✓	**1.51**	**1.53**	**1.55**	**284.76**	**284.00**	**284.21**
Independent CBM	×	1.59	1.64	1.65	280.86	308.38	310.57
	✓	**1.51**	**1.53**	**1.55**	**282.48**	**312.72**	**316.45**
Joint CBM	×	2.88	3.23	3.10	273.06	236.22	296.80
	✓	**1.75**	**1.77**	**1.74**	**273.76**	**246.09**	**303.76**
CEM	×	1.65	1.90	1.83	396.70	366.87	361.60
	✓	**1.49**	**1.66**	**1.58**	**401.84**	**370.88**	**363.57**