Interpretable Generative Models through Post-hoc Concept Bottlenecks

Akshay Kulkarni, Ge Yan*, Chung-En Sun*, Tuomas Oikarinen, and Tsui-Wei Weng
University of California San Diego
{a2kulkarni, lweng}@ucsd.edu

Abstract

Concept bottleneck models (CBM) aim to produce inherently interpretable models that rely on human-understandable concepts for their predictions. However, existing approaches to design interpretable generative models based on CBMs are not yet efficient and scalable, as they require expensive generative model training from scratch as well as real images with labor-intensive concept supervision. To address these challenges, we present two novel and low-cost methods to build interpretable generative models through post-hoc techniques and we name our approaches: concept-bottleneck autoencoder (CB-AE) and concept controller (CC). Our proposed approaches enable efficient and scalable training without the need of real data and require only minimal to no concept supervision. Additionally, our methods generalize across modern generative model families including generative adversarial networks and diffusion models. We demonstrate the superior interpretability and steerability of our methods on numerous standard datasets like CelebA, CelebA-HQ, and CUB with large improvements (average ∼25%) over the prior work, while being 4-15× faster to train. Finally, a large-scale user study is performed to validate the interpretability and steerability of our methods.

1. Introduction

Deep generative models [36–38] have become increasingly powerful and widely used in many high-stakes domains and applications, including realistic data generation [11], simulating hypothetical scenarios or environments [19], and scientific discovery [1]. It is therefore important to ensure that the generation process is interpretable, which will allow us to understand and audit the generation, and further mitigate potential biases and harms (*e.g.* content moderation).

Unfortunately, most of the advances in deep learning utilize complex, black-box neural network architectures that are difficult to interpret and understand. This leads to user mistrust of model predictions due to the absence of explanations.

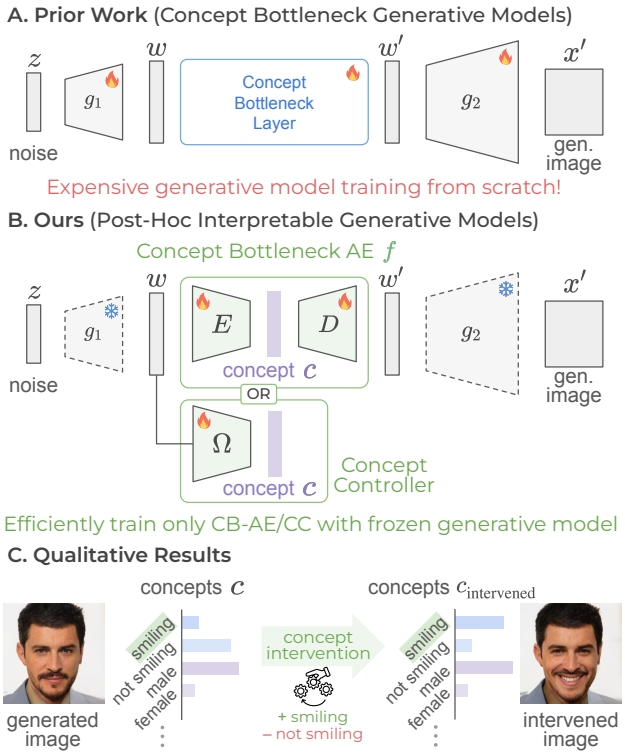

Figure 1. **A.** Prior work on interpretable generative models requires expensive generative model training from scratch. **B.** Our CB-AE and CC can be trained efficiently for post-hoc interpretability in a pretrained, frozen generative model $g_2 \circ g_1$. **C.** Example concept intervention with CB-AE and corresponding concept vectors.

To address this, there has been work on developing inherently interpretable deep vision models using concept bottlenecks [20, 32, 39, 43–45]. These approaches train a concept bottleneck layer after the feature extractor (backbone) to embed a set of human-understandable concepts, followed by an interpretable sparse linear layer for the final classification based on the concept prediction. However, current development of CBMs is primarily focused on classification tasks, and only one prior work, CBGM [16], has extended it to image generation, indicating this area is under-explored.

The key idea of the seminal work CBGM [16] is to represent concepts as learnable embeddings [46] at an inter-

*Equal contribution
[1]Code: github.com/Trustworthy-ML-Lab/posthoc-generative-cbm

Table 1. Comparison of our CB-AE and CC with prior work CBGM [16] (green indicates desirable properties). CC trades-off inherent interpretability for better steerability, image quality, and faster training. We could not reproduce CBGM [16] to evaluate concept accuracy.

Method	Post-hoc training	Training without concept-labeled real images	Inherently interpretable model	Concept Acc. (%)	Steerability (%) (↑)	FID (↓)	Train time (V100-hrs) (↓)	
							StyleGAN2	DDPM
CBGM [16]	✗	✗	✓	-	25.60	9.10	50	240
CB-AE (*Ours*)	✓	✓	✓	86.56	47.34 (+21.74)	9.52	14 (3.5× faster)	29.5 (8.1× faster)
CC (*Ours*)	✓	✓	✗	**87.65**	**51.14** (+25.54)	**7.65**	**6** (8.3× faster)	**8.3** (28.9× faster)

mediate location in the generative model, and combine the embeddings to compute the generative model latent. However, the CBGM-based generative model has to be trained from scratch using concept-labeled real images, which could be difficult to scale and computationally intensive (*e.g.* 240 V100-hours for DDPM-256×256 [15]) as shown in Fig. 1A. To address these limitations, in this work, we develop an *efficient* and *scalable* post-hoc concept bottleneck to *transform* any pretrained generative model into an interpretable model. In contrast to CBGM [16], our approach works well by only training a few layers with minimal concept supervision.

Specifically, we propose a novel concept-bottleneck autoencoder (CB-AE) f that can be inserted into the intermediate layers of a pretrained generative model $g = g_2 \circ g_1$ as shown in Fig. 1B (top). The CB-AE input and output are the generator latent w and its reconstruction w' respectively, and the overall generative model now becomes $g_2 \circ f \circ g_1$. The CB-AE latent space is the concept space c which is used to reconstruct the generator latent space. In our framework, given a pretrained generative model $g = g_2 \circ g_1$, only the encoder E and decoder D in the CB-AE $f = D \circ E$ need to be trained, while g_1, g_2 are frozen pretrained weights, *i.e.* CB-AE uses post-hoc training. The benefit of the proposed CB-AE is that it allows us to debug the model easily with concept-level control by modifying the CB-AE concept latent during image generation (Fig. 1C). Further, our training requires only concept pseudo-labels achievable with minimal concept supervision (*e.g.* zero-shot CLIP classifier [35]).

We also propose novel optimization-based concept interventions to achieve higher success rate and intervention quality. Based on the CB-AE, we propose an even more efficient post-hoc concept controller (CC) method (Fig. 1B, bottom) with simplified training, that can provide concept predictions and leverage optimization-based concept interventions. Note that while CB-AE is part of the new interpretable generative model, CC is a post-hoc control method that is not part of the generative model. Finally, we evaluate our CB-AE and CC on various generative models, including GANs and diffusion models, for standard datasets including CelebA, CelebA-HQ, and CUB. We show that CB-AE and CC significantly outperform prior work CBGM [16] w.r.t. steerability (or intervention success rate) on CelebA (average +23%) while also having 4-15× faster training (Table 1).

Our contributions can be summarized as follows:

- We are the first to propose a *post-hoc* concept bottleneck autoencoder (CB-AE) for interpretable generative models. CB-AE can be trained efficiently with a frozen pretrained generative model, without real concept-labeled images.
- We also propose a novel and efficient optimization-based concept intervention method with improved steerability (avg. +19%) and higher image quality (avg. 32% better).
- We validate the effectiveness of our methods for GANs and diffusion models (avg. +31% and +28% steerability w.r.t. prior state-of-the-art) across varying image resolutions, while being 4-15× faster to train on average.

2. Related Work

Concept Bottleneck Models. Early work on CBMs [20, 25] relied on concept-labeled images to train a concept bottleneck layer with each neuron as a human-understandable concept, followed by a linear layer based on the concepts for the final classification. Post-hoc CBMs [45] extended this idea to convert a pretrained backbone into a CBM. More recent works like LF-CBM [32], LM4CV [43], LaBo [44], and VLG-CBM [39] use interpretability tools [31], vision-language models like CLIP [35], large language models, or open-set object detection [23] to eliminate the need for expensive concept-labeled data. Independently from our work, [21] proposed a concept-based intervention without a concept bottleneck, similar to our CC, but for classification. All these works are specifically designed for classification, while in this paper, our focus is on the image generation task.

Interpretability for Generative Models. A line of work [4, 7, 13] on learning disentangled concepts in variational autoencoders enables controllable generation, but they train from scratch and are not applicable to other generative models like GANs. Other works focus on identifying and manipulating structural rules or concepts in GANs [2, 3] and large language models [26, 28, 29, 40] by editing the model weights. In contrast, we focus on training inherently interpretable models, and the closest prior work is the recent CBGM [16], which also aims to build interpretable generative models, but requires expensive concept labels and training from scratch, limiting its scalability. In contrast, our proposed CB-AE can be trained efficiently with a frozen pretrained generative model with minimal concept supervision.

Image Editing in Generative Models. Some works focus on conditional generation [37, 38] and image editing in generative models by modifying model weights [8, 33]. In

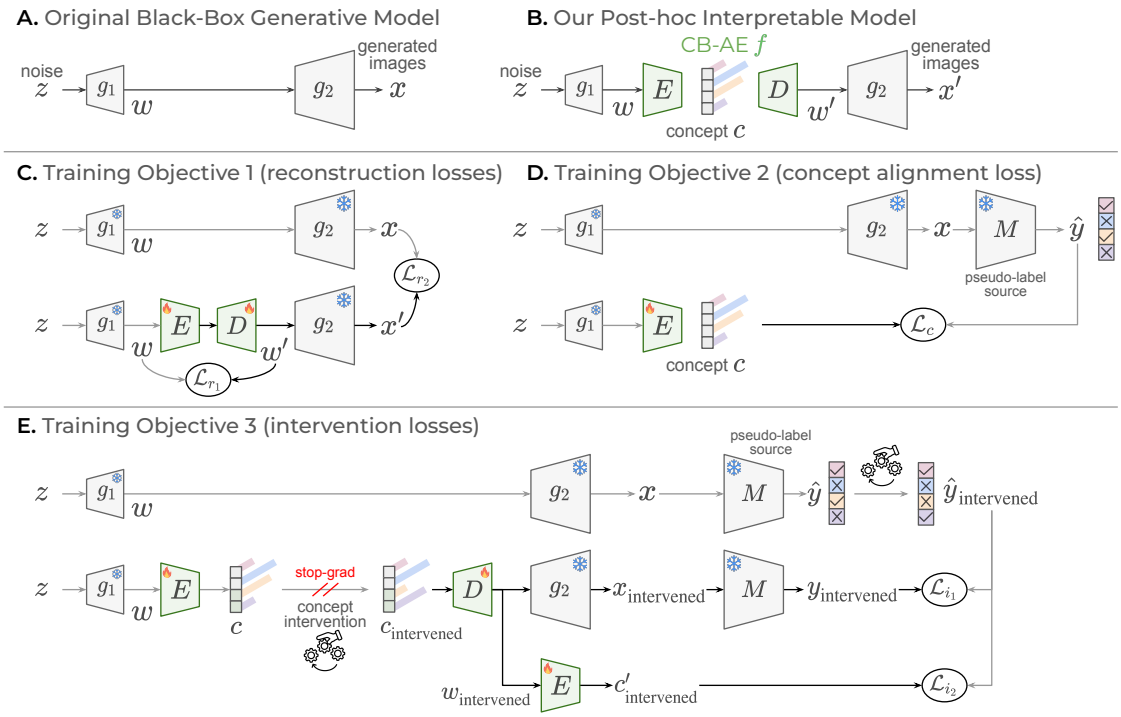

Figure 2. **Post-hoc CB-AE training** for reconstruction, concept alignment, and intervention with a frozen pretrained generator $g_2 \circ g_1$. Note that $\mathcal{L}_c$, $\mathcal{L}_i$ indicate cross-entropy loss and $\mathcal{L}_r$ indicates mean-squared-error loss. The darker lines indicate gradient flow during training.

contrast, our work focuses on developing inherently interpretable generative models by introducing additional concept bottleneck layers, where the capability of editing or intervention is naturally a by-product of the interpretability.

3. Proposed Methods

We propose a novel and low-cost concept bottleneck autoencoder (CB-AE) method in Sec. 3.1 to incorporate post-hoc interpretability in pretrained generative models. In Sec. 3.2, we present an optimization-based intervention method for concept-based steerability in the generative model. Finally, based on our insights from CB-AE and optimization-based interventions, we propose an even lower-cost post-hoc control method, concept controller (CC) in Sec. 3.3.

Preliminaries. Consider a generative model $g : \mathcal{Z} \to \mathcal{X}$ that maps from random noise $z \in \mathcal{Z}$ to an image $x \in \mathcal{X}$. To make the generative model g become inherently interpretable, the goal of CBMs is to insert and train a concept bottleneck (say) f at an intermediate location in g. Let $g = g_2 \circ g_1$, *i.e.* g can be divided into two parts, g_1 and g_2 (*e.g.* for a DCGAN [34], g_1 is the first 2 layers of g and g_2 is the remaining layers). The input to the concept bottleneck f will be the output of g_1 (Fig. 1). At the bottleneck of f, we obtain the concept prediction $c \in \mathcal{C}$ where $\mathcal{C}$ is the set of pre-defined concepts. In our setup, the concept vector c has two logits for each binary concept c_i (*e.g.* "smiling" would have two logits, for smiling and not smiling) and N logits for each categorical concept c_j with N classes (*e.g.* blonde/black/white/gray hair color would have four logits). For example, suppose we have one binary concept c_i and one categorical concept c_j, then we define $c = [c_i^+, c_i^-, c_j^{(1)}, c_j^{(2)}, \ldots, c_j^{(N)}]^\top$ where $c_i = [c_i^+, c_i^-]^\top \in \mathbb{R}^2$ and $c_j = [c_j^{(1)}, c_j^{(2)}, \ldots, c_j^{(N)}]^\top \in \mathbb{R}^N$.

3.1. Post-hoc Concept Bottleneck Autoencoder

We propose a concept bottleneck autoencoder (CB-AE), $f = D \circ E$ (see Fig. 2B). The latent space of the CB-AE encoder E is the concept prediction $c = E(g_1(z))$. Apart from the predefined concepts, c also contains an unsupervised concept embedding, learned using autoencoder reconstruction and intervention objectives, to encode other concepts absent from the predefined set (similar to CBGM [16]). The decoder D reconstructs the features from g_1 based on the concept prediction c, outputting $w' = D(c)$. Considering the original output of g_1 to be $w = g_1(z)$, we can generate the original image $x = g_2(w)$ as well as a reconstructed image $x' = g_2(w') = g_2(D \circ E(w))$ generated using the CB-AE.

Training. There are 3 goals for the CB-AE. First, the generator's performance should be preserved even if the CB-AE output w' is used instead of w. Second, CB-AE should provide interpretability for the generated images x' through the corresponding concepts c. Lastly, CB-AE should allow accurate steering on image generation via concept interven-

tions. Based on these 3 goals, we formulate **Objective 1-3** for CB-AE training below and illustrated in Fig. 2C-E.

Objective 1 (reconstruction losses $\mathcal{L}_{r_1}, \mathcal{L}_{r_2}$). At each training iteration, we sample a latent w by passing uniform noise z to g_1. Then, the latent w is passed through g_2 to obtain a generated image $x = g_2(w)$ from the original model $g = g_2 \circ g_1$, without using our CB-AE. We also reconstruct $w' = D \circ E(w)$ using our CB-AE and the same latent w, and obtain another generated image $x' = g_2(w')$. Since our first goal is to preserve the generator's performance when using the CB-AE, we apply reconstruction losses (mean-squared error loss $\mathcal{L}_r$) between the latents w, w' and between the generated images x, x', as shown in Fig. 2C:

$$\min_{E,D}[\mathcal{L}_{r_1}(w, w') + \mathcal{L}_{r_2}(x, x')], \tag{1}$$

where $w' = D \circ E(w)$ and $x' = g_2(w') = g_2 \circ D \circ E(w)$ and only the CB-AE parameters (*i.e.* E, D) are trainable.

Objective 2 (concept alignment loss $\mathcal{L}_c$). To ensure interpretability, we first obtain a concept pseudo-label $\hat{y} = M(x)$ for a generated image x from a pseudo-label source M (Fig. 2D). The source M can be either an off-the-shelf supervised model or a zero-shot prediction pipeline from (say) CLIP with concept names $\mathcal{C}$ as the text inputs. With this approach, we avoid the requirement of any real images for training as well as the requirement of concept labels, unlike CBGM [16]. Since our second goal is to provide interpretability for the generated images through the concepts c, we apply a cross-entropy loss $\mathcal{L}_c$ between the CB-AE encoder output $c = E(w)$ and the concept pseudo-label $\hat{y} = M(x)$:

$$\min_{E}[\mathcal{L}_c(\hat{y}, c)]. \tag{2}$$

The losses in Eq. (1), (2) are simultaneously optimized to learn only the CB-AE parameters E, D.

Objective 3 (intervention losses $\mathcal{L}_{i_1}, \mathcal{L}_{i_2}$). Interventions are an important feature of concept-bottleneck models [20], allowing users to control the model output by modifying the concepts c. However, for our CB-AE decoder D, reconstruction and concept alignment losses in Objective 1 and 2 do not provide guidance on how the reconstructed latent w' should change when concepts c are manually modified. Hence, for steerability, we design Objective 3 (Fig. 2E) that encourages the CB-AE decoder D to produce an appropriately changed and realistic latent w' when the concepts in c are modified. We first describe **(a)** how interventions are performed in our CB-AE, followed by designing **(b)** an intervened concept alignment loss $\mathcal{L}_{i_1}$ and **(c)** a cyclic intervened concept loss $\mathcal{L}_{i_2}$ to encourage steerability, *i.e.* intervention success.

a) Intervening concepts. At each training iteration, we choose a random logit for a random concept to intervene, and modify only the chosen concept based on the chosen logit to get an intervened concept vector $c_{\text{intervened}}$ (Fig. 2E). For example, for a desired binary concept $i \in \mathcal{C}$, the new concept vector $c_{\text{intervened}}$ is computed by swapping the two logits, *i.e.* $c_{\text{intervened}} = [\ldots, c_i^-, c_i^+, \ldots]$ from $c = [\ldots, c_i^+, c_i^-, \ldots]$. The same can be done for a categorical concept $i \in \mathcal{C}$ as well by swapping the desired logits (say) $c_i^{(k)}$ with the highest logits $c_i^{(\ell)}$ where $\ell = \arg\max_j c_i^{(j)}$. Concretely, $c = [\ldots, c_i^{(1)}, \ldots, c_i^{(\ell)}, \ldots, c_i^{(k)}, \ldots, c_i^{(N)}, \ldots]$ is modified to $c_{\text{intervened}} = [\ldots, c_i^{(1)}, \ldots, c_i^{(k)}, \ldots c_i^{(\ell)}, \ldots, c_i^{(N)}, \ldots]$. We can also intervene on multiple concepts simultaneously.

b) Designing $\mathcal{L}_{i_1}$. Using the intervened concepts $c_{\text{intervened}}$, we reconstruct an intervened latent $w_{\text{intervened}} = D(c_{\text{intervened}})$ and obtain an intervened generated image $x_{\text{intervened}} = g_2(w_{\text{intervened}})$. Using the pseudo-label source, we obtain an intervened concept prediction $y_{\text{intervened}} = M(x_{\text{intervened}})$. Since we already have the concept pseudo-label $\hat{y} = M(x)$ for the original image x, we modify $\hat{y}$ to $\hat{y}_{\text{intervened}}$ by changing only the earlier chosen concept $i \in \mathcal{C}$ in $\hat{y}$ to the chosen value (based on earlier chosen logit) as shown in Fig. 2E. In other words, $\hat{y}_{\text{intervened}}$ is the intervened concept pseudo-label. Now, to align the concepts in the intervened image $x_{\text{intervened}}$ with the predicted concepts from M, we use a cross-entropy loss $\mathcal{L}_{i_1}$ between $\hat{y}_{\text{intervened}}$ and $y_{\text{intervened}}$:

$$\min_{E,D}[\mathcal{L}_{i_1}(\hat{y}_{\text{intervened}}, y_{\text{intervened}})]. \tag{3}$$

c) Designing $\mathcal{L}_{i_2}$. For cyclic consistency, we pass the intervened latent $w_{\text{intervened}}$ through the CB-AE encoder E to obtain a concept prediction $c'_{\text{intervened}}$ (Fig. 2E) and apply a cross-entropy loss $\mathcal{L}_{i_2}$ w.r.t. $\hat{y}_{\text{intervened}}$ to align the encoder's prediction with that of the pseudo-label source M:

$$\min_{E,D}[\mathcal{L}_{i_2}(\hat{y}_{\text{intervened}}, c'_{\text{intervened}})]. \tag{4}$$

We use $\mathcal{L}_{i_1}, \mathcal{L}_{i_2}$ instead of $\mathcal{L}_c$ for cross-entropy loss to differentiate the intervention losses from the concept loss in Objective 2, and only CB-AE parameters E, D are trainable. Finally, recall that the concept vector c contains an unsupervised concept embedding. Objective 3 implicitly encourages this embedding to not encode known concepts, since the unsupervised embedding is not modified during the intervention from c to $c_{\text{intervened}}$.

Test-Time Intervention. Similar to training-time interventions, we can perform test-time interventions by modifying the value of any chosen concept (by swapping the logits) in the predicted concept vector c to $c_{\text{intervened}}$. The intervened image $x_{\text{intervened}} = g_2(w_{\text{intervened}})$ can be obtained where $w_{\text{intervened}} = D(c_{\text{intervened}})$. Note that swapping the logits ensures that the range of values in $c_{\text{intervened}}$ are similar to c, and also avoids the requirement of any estimation of how much to change the desired concept's logit. This makes it more accessible to users as they need not worry about the actual values being changed during an intervention.

3.2. Optimization-based interventions

For an alternative intervention method, we draw inspiration from adversarial attacks [10] to perform test-time interventions using gradient-based optimization. Specifically, we use the iterative randomized fast gradient sign method (I-RFGSM) [42] on the CB-AE encoder prediction.

Consider a generated image $x = g_2(w)$ with concept prediction $c = E(w)$. To intervene in the generation process to obtain modified concepts c^*, we solve the following objective using gradient ascent,

$$w^* = w + \arg\max_{\delta \in \Delta} [-\mathcal{L}_c(E(w+\delta), c^*)] \tag{5}$$

where $\Delta = \{\delta : \|\delta\|_\infty \leq \epsilon\}$ is the ℓ_∞-norm bound on δ with a hyperparameter $\epsilon > 0$. Intuitively, we optimize a small perturbation δ such that $w^* = w + \delta$ leads to the desired concepts c^*. Then, the generated image $x^* = g_2(w^*)$ is very similar to x but contains concepts c^*.

3.3. Post-hoc Concept Controller (CC) for Steering

While optimization-based interventions work well empirically, it is interesting to note that the CB-AE decoder D is not involved in the process. This leads us to question whether the CB-AE decoder can be removed if a user only plans to perform optimization-based interventions. While such a removal would not result in a CBM, it would be sufficient for the purpose of steering image generation. For this particular use case, it would be even more efficient than training the CB-AE since reconstruction and intervention losses are no longer required. Hence, we propose a post-hoc concept controller (CC), denoted as Ω, that predicts the concepts, *i.e.* $c = \Omega(g_1(z))$ to efficiently steer image generation (Fig. 1B).

Training. For CC, the training is simply Objective 2 of the CB-AE training, *i.e.* the cross-entropy loss $\mathcal{L}_c$ w.r.t. the concept pseudo-labels $\hat{y} = M(x)$ from the pseudo-label source M. Formally, the objective is $\min_\Omega[\mathcal{L}_c(\hat{y}, c)]$ where $c = \Omega(g_1(z))$ are the predicted concepts, and the loss encourages the concept controller Ω to align with the pseudo-label source M. Similar to the CB-AE training, we avoid the requirement of any real images for training as well as the requirement of concept labels.

4. Experiments

We detail the experimental setup and comprehensively evaluate our proposed methods with respect to the state-of-the-art prior work as well as other baselines.

4.1. Experimental setup

Base generative models and datasets. We evaluate our CB-AE and CC methods on diverse generative models including GAN [9], Progressive GAN [17], StyleGAN2 [18], and DDPM [15]. We use models pretrained on standard datasets of varying image resolution (64×64 to 512×512) like CelebA [24], CelebA-HQ [22], and CUB [41]. Following CBGM [16], we evaluate on the small balanced concept regime with the 8 most balanced concepts and the large unbalanced concept regime with all 40 concepts from the dataset. For CUB, we use 10 balanced concepts as per CBGM. Please refer to the Appendix for more details.

CB-AE and CC. We use a 4-layer MLP or 4 convolution (and transposed convolutional) layers for CB-AE encoder or CC (and decoder) depending on the dimensions of latent w. We use unsupervised concept embedding $\in \mathbb{R}^{40}$ in CB-AE (ensuring bottleneck is much smaller than the latent). As in CBGM [16], we train CB-AE/CC for 50 epochs with batch size 64. For optimization-based interventions, we use 50-step I-RFGSM [42] with ℓ_∞-norm bound $\epsilon = 0.1$. Please refer to the Appendix for complete implementation details.

Pseudo-label source M. We consider three variants for M with varying levels of concept supervision. First, we use off-the-shelf supervised (ResNet18-based) concept classifiers. Second, with no concept supervision, we use CLIP zero-shot classifier [35] with only the concept names to obtain concept pseudo-labels. Third, as a compromise between the above two, we use TIP [48] which is a few-shot-labeled version of CLIP zero-shot classifier, utilizing 128 concept-labeled real images. Unless otherwise mentioned, our experiments use the supervised classifiers for a fair comparison with the prior work CBGM [16] that utilizes concept labels.

Automated evaluation. Following CBGM [16], we train concept classifiers (ViT-L-16-based) on real images and concept labels with high accuracy on a held-out test set. Note that these classifiers are separate and have higher accuracy than those used for pseudo-labels. We evaluate our method using three automated metrics:

- **Concept Accuracy** is computed over 5k generated images as the average agreement between the supervised classifiers and our proposed CB-AE or CC.
- **Steerability** [16]: For each target concept, we find 5k latents that do not produce the target concept (*i.e.* probability of target concept < 0.5 from the supervised concept classifier). For these latents, we perform the concept intervention using either the baselines or our methods, and compute the steerability as the percentage of intervened images that are classified to have the target concept.
- **Generation Quality** is evaluated using the standard Fréchet Inception Distance (FID) [12].

Intuitively, the concept accuracy, FID, and steerability metrics measure how well the concept, reconstruction, and intervention objectives, respectively, are satisfied.

Human evaluation. We conduct a large-scale user study on Amazon Mechanical Turk to validate the automated evaluation of concept accuracy and steerability. For both metrics,

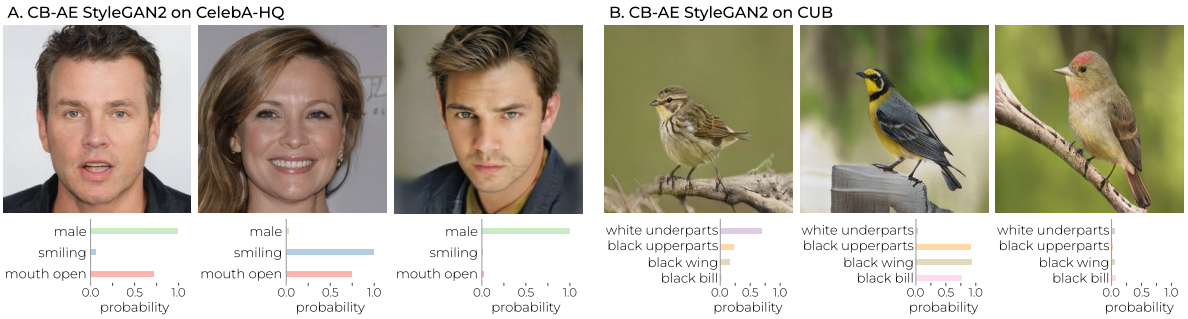

Figure 3. Samples generated using CB-AE with CelebA-HQ and CUB pretrained StyleGAN2 models along with concept probabilities.

Figure 4. Concept intervention examples for CB-AE with CelebA-HQ pretrained StyleGAN2.

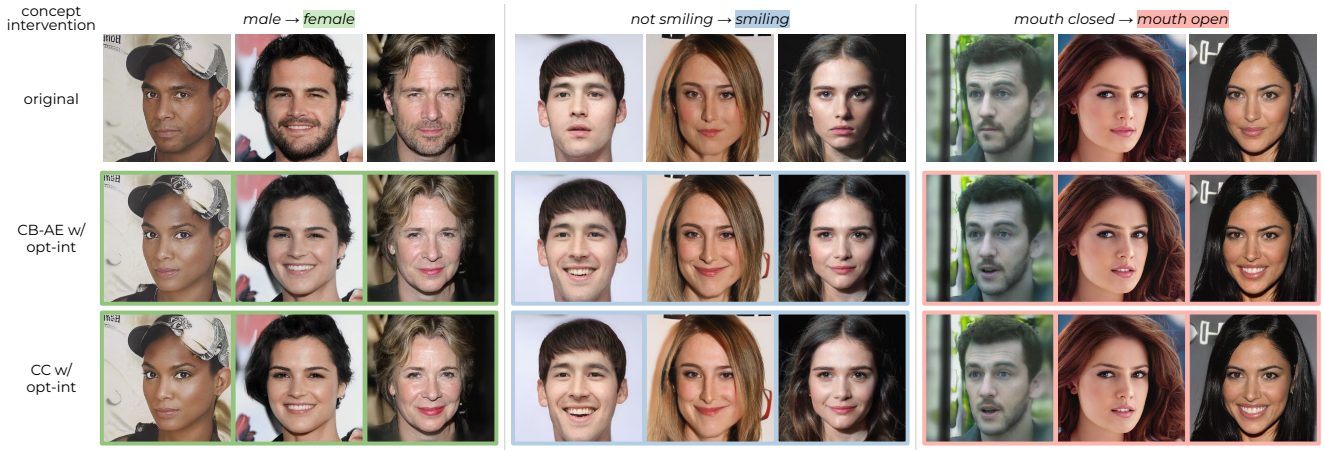

Figure 5. Optimization-based concept intervention (opt-int) examples for CB-AE and CC with CelebA-HQ pretrained StyleGAN2.

we display 10 images at a time and ask the user to click on images that match a displayed concept c_i^+ (see Appendix for more details). The images are collected as follows:

- **Concept Accuracy**: We collect and shuffle generated images and save their CB-AE and CC concept predictions (c_i^+ or c_i^-). Based on user responses, we compute human agreement rate w.r.t. CB-AE/CC predictions.
- **Steerability**: We collect and shuffle generated images of concept c_i^- with intervened images of concept c_i^+. Based on user responses, we compute human agreement rate on whether intervened images actually contain concept c_i^+. Shuffling the original and intervened images ensures that users are not biased towards clicking all images.

We evaluate using approximately 100 images per concept and per method for two concepts from CelebA-HQ: "smiling" and "male/female". Each set of 10 images is evaluated by 3 different users. Refer to the Appendix for more details.

4.2. Evaluation

Qualitative evaluation. In Fig. 3, we present CB-AE-StyleGAN2 generated images for CelebA-HQ and CUB datasets with corresponding concept predictions. We also visualize CB-AE interventions in Fig. 4 and optimization-based interventions in Fig. 5. Interestingly, with the same latent w, both CB-AE and CC lead to very similar optimization-based interventions, which is reasonable since the same pseudo-label source M was used in both cases.

Table 2. **Concept accuracy evaluation** on 5k samples with 8 concepts for CelebA, CelebA-HQ, and 10 concepts for CUB.

Conc. Acc. (%)	CelebA (64×64)			CelebA-HQ (256×256)			CUB (64×64)		CUB (256×256)
	GAN	PGAN	DDPM	DDPM	StyGAN2	PGAN	DDPM	GAN	StyGAN2
Ours (CB-AE)	86.56	87.87	84.98	89.79	**86.04**	82.68	72.35	74.41	**81.33**
Ours (CC)	**87.65**	**90.00**	**85.13**	**89.82**	83.57	**83.94**	**72.87**	**75.60**	81.11

Table 3. **Extended steerability evaluation** on 5k samples with 8 concepts for CelebA, CelebA-HQ, and 10 concepts for CUB. †CBGM numbers are from their paper (1k samples) since their results are not reproducible using their released code. For CBGM training time, we used the time taken by their code for GAN and base model training times for other models (whose code was not available) with 1 V100 GPU.

Steerability (%)		CelebA (64×64)			CelebA-HQ (256×256)			CUB (64×64)		CUB (256×256)
		GAN	PGAN	DDPM	DDPM	StyGAN2	PGAN	DDPM	GAN	StyGAN2
CBGM [16]†		25.60	-	13.80	-	-	-	14.80	21.30	-
Ours (CB-AE)		47.34	40.31	23.72	23.50	40.27	29.31	25.64	20.89	10.52
Ours (CB-AE+opt-int)		**61.14**	41.73	38.09	50.49	61.66	32.10	36.94	46.03	**65.11**
Ours (CC+opt-int)		51.14	**58.94**	**41.45**	**56.70**	**67.95**	**47.29**	**44.47**	**48.91**	44.72
Train time reduction	CB-AE	3.7×	5.4×	4×	8.1×	3.5×	2×	3.7×	3.3×	3.1×
w.r.t. CBGM	CC	30.9×	20.3×	10×	28.9×	8.3×	6.7×	8.5×	21×	7.1×

Overall, we find that the optimization-based interventions are relatively higher quality and more orthogonal (*i.e.* less changes to other concepts) than the CB-AE interventions.

Concept Accuracy. In Table 2, we report the concept accuracies for our CB-AE and CC. We could not compare with CBGM since they do not evaluate concept accuracy, and since we could not reproduce their results. Overall, we find that both CB-AE and CC achieve good concept accuracies across various datasets, models and image resolutions.

In most scenarios, CC outperforms CB-AE since concept alignment is the sole objective optimized in CC, while CB-AE has additional objectives. However, we observe that CB-AE outperforms CC only for StyleGAN2. This is because the multiple objectives in CB-AE training may be easier to balance when dealing with clean latents (GANs) than with noisy latents (DDPM), leading to CB-AE outperforming CC for StyleGAN2. This is supported by average loss for CB-AE being lower for StyleGAN2 (0.68) than DDPM (0.92).

Steerability. We compare the concept steerability of our methods on 1k samples with GAN intervention methods like conditional GAN (CGAN) [27], auxiliary classifier GAN (ACGAN) [30], and CBGM (CB-GAN) [16], and with diffusion model intervention methods like classifier-guided (CG) [6] DDPM, classifier-free (CF) [14] DDPM, and concept bottleneck DDPM (CB-DDPM) [16] in Table 4. We observe significant gains over CBGM [16] for both GAN (average +14.2% for CB-AE, +31.1% for CB-AE with opt-int, and +26.6% for CC) and for DDPM (average +10.4% for CB-AE, +23.3% for CB-AE with opt-int, and +29.1% for CC).

Table 3 presents extended steerability evaluation on 5k samples with other types of GANs like Progressive GAN [17] and StyleGAN2 [18] for higher resolution (256×256) datasets. We could not obtain CBGM's results for these settings using their code. Overall, we find optimization-based interventions (opt-int) outperform the CB-AE intervention method, and CC generally outperforms CB-AE. Intuitively, opt-int involves instance-specific and iterative optimization, while CB-AE is applied in the same way to all samples. Although CB-AE is trained with intervention losses, its steerability tends to be worse than CC since CB-AE is more challenging to train, with multiple objectives to satisfy.

Table 4. **Steerability comparisons** with CBGM [16] and other baseline intervention methods computed on 1k samples (each experiment repeated three times for mean and standard deviation).

Concept Regime	Small balanced concepts		Large unbalanced concepts
Dataset	CUB (10 conc.)	CelebA (8 conc.)	CelebA (40 conc.)
Baseline Intervention Methods			
CGAN [27]	5.4 ± 0.4	8.7 ± 1.3	2.9 ± 0.0
ACGAN [30]	18.5 ± 0.4	9.2 ± 0.7	1.2 ± 0.1
CB-GAN [16]	21.3 ± 0.3	25.6 ± 0.5	23.1 ± 0.2
Our Methods			
CB-AE-GAN	19.9 ± 0.9	47.1 ± 0.5	45.5 ± 0.4
CB-AE-GAN+opt-int	45.1 ± 1.1	**59.9 ± 1.8**	**58.3 ± 1.3**
CC-GAN+opt-int	**49.3 ± 0.6**	50.8 ± 1.2	49.7 ± 0.9
Baseline Intervention Methods			
CF-DDPM [14]	2.7 ± 1.9	7.2 ± 3.8	5.1 ± 2.4
CG-DDPM [6]	2.1 ± 1.4	6.8 ± 1.1	5.4 ± 2.6
CB-DDPM [16]	14.8 ± 6.2	13.8 ± 2.7	12.6 ± 1.7
Our Methods			
CB-AE-DDPM	25.8 ± 1.1	23.1 ± 0.9	23.6 ± 1.0
CB-AE-DDPM+opt-int	37.3 ± 1.5	37.5 ± 1.3	36.2 ± 1.1
CC-DDPM+opt-int	**45.4 ± 2.2**	**41.8 ± 1.8**	**41.3 ± 1.5**

Generation quality. We compare the generation quality of CBGM [16] with CB-AE and CC in Table 5. For CB-AE, we observe a relatively higher drop in image quality than CBGM, but our methods are trained 3.5-8× faster, and do not require training from scratch. For a high-quality StyleGAN2 model, our CB-AE and CC with optimization-based interventions can produce almost the same quality

Table 5. **Generation quality and training time comparisons** for CelebA-HQ with StyleGAN2. †CBGM results are from their paper. Training time is in V100 GPU-hours.

FID (↓)	CBGM† [16]	CB-AE (*Ours*)	CC (*Ours*)
Base model	9.0	7.66	7.66
CB model	9.1	9.52	-
CB interv.	-	9.65	-
Opt-interv.	-	7.67	7.65
Train time (hrs)	50	14	6

Table 6. **Human evaluation results** for CB-AE and CC trained with CelebA-HQ pretrained StyleGAN2.

Conc. Acc. (%)	Smiling		Male/Female	
	Automated	Human	Automated	Human
CB-AE (*Ours*)	92.38	86.35	100.0	94.06
CC (*Ours*)	89.47	80.35	96.96	96.30

Steerability (%)	Smiling		Female	
	Automated	Human	Automated	Human
CB-AE (*Ours*)	65.90	77.27	17.02	17.73
CB-AE w/ opt-int (*Ours*)	76.59	78.72	42.85	41.50
CC w/ opt-int (*Ours*)	77.36	77.36	26.92	19.87

of images as the base model while having high steerability (61.66% and 67.95% respectively, from Table 3).

Intuitively, since CB-AE focuses on concept and intervention losses, we obtain better steerability, while CBGM has better FID since they include generative model losses. Hence, there is a tradeoff between image quality and interpretability which can be improved in future work.

Human evaluation. In Table 6, we compare human agreement rate with automated evaluation of concept accuracy and steerability. We chose easily recognizable concepts, "gender" and "smiling" as representative of low and high steerability respectively. Overall, our automated evaluation is similar to human agreement (with room for improving the classifiers), validating the usefulness of automated evaluation.

4.3. Analysis

Ablation study. We analyze the contribution of each CB-AE training loss. However, we do not ablate concept loss $\mathcal{L}_c$ or latent reconstruction loss $\mathcal{L}_{r_1}$ since our evaluation metrics would be meaningless if the CB-AE cannot predict the concepts or cannot reconstruct the generator latent w'. Hence, Table 7 ablates image reconstruction loss $\mathcal{L}_{r_2}$ from Eq. (1), intervened concept loss $\mathcal{L}_{i_1}$ from Eq. (3), and cyclic intervened concept loss $\mathcal{L}_{i_2}$ from Eq. (4). The ablations are for the most challenging pseudo-label setting of M (CLIP-zero-shot), since it is most affected by loss ablations.

In Table 7, we observe that using the image reconstruction loss $\mathcal{L}_{r_2}$ improves the generation quality FID (row #2 vs. #1), which is intuitive since this loss directly encourages the images to be closer to the original ones. Next, using only the intervened concept loss or only the intervened cyclic loss improves the steerability while trading-off generation quality (row #3 vs. #2 or row #4 vs. #2). Finally, using both intervened losses significantly improves both concept accuracy and steerability (row #5 vs. #3 and #4), while generation quality remains similar. Overall, both intervention losses are crucial to ensure good concept accuracy and steerability, while image reconstruction loss improves generation quality.

Table 7. **Ablation study on CB-AE training objectives** for the most challenging CLIP-zero-shot pseudo-label setting for CelebA-HQ pretrained StyleGAN2. $\mathcal{L}_{r_2}, \mathcal{L}_{i_1}, \mathcal{L}_{i_2}$ indicate image reconstruction loss, intervened concept loss, and intervened cyclic loss respectively from Eq. (1), (3), (4).

Row #	$\mathcal{L}_{r_2}$	$\mathcal{L}_{i_1}$	$\mathcal{L}_{i_2}$	Trained with M = CLIP-zero-shot		
				Conc. Acc. (%)	Steerability (%)	FID (↓)
1	✗	✗	✗	58.87	6.01	11.06
2	✓	✗	✗	59.32	6.51	**9.88**
3	✓	✓	✗	57.75	9.98	12.42
4	✓	✗	✓	58.33	13.03	13.62
5	✓	✓	✓	**67.26**	**20.61**	12.82

Table 8. **Sensitivity to pseudo-label source M.** For CB-AE trained with CelebA-HQ pretrained StyleGAN2, we compare concept accuracy and steerability with different M: CLIP-zero-shot [35], TIP-few-shot [48], or supervised classifiers.

Pseudo-label source M	Conc. Acc. (%)	Steerability (%)	
		CB-AE	CB-AE w/ opt-int
CLIP-zs	67.26	20.61	29.23
TIP-fs-128	76.08	21.51	38.73
Supervised-clsf	**86.04**	**40.27**	**61.66**

Sensitivity to pseudo-label source M. Table 8 compares concept accuracy and steerability when varying pseudo-label source M. Concept accuracy improves as pseudo-label quality improves from CLIP-zero-shot to supervised classifiers. While TIP few-shot does not improve CB-AE interventions, it significantly improves optimization-based interventions and highlights the usefulness of even limited labeled data. Further, our method can use newer CLIP models like SigLIP [47], OpenCLIP [5] to further improve performance.

5. Conclusion

In this work, we proposed two novel and low-cost methods, concept-bottleneck autoencoder (CB-AE) and concept controller (CC), to efficiently build interpretable generative models from pretrained models. Compared to the prior approach that struggles with efficiency and scalability, our methods achieve 4-15× faster training, require minimal to no concept supervision, and generalize across modern generative model families including GANs and diffusion models with 25% improved steerability on average.

Acknowledgements

This work is supported in part by National Science Foundation (NSF) awards CNS-1730158, ACI-1540112, ACI-1541349, OAC-1826967, OAC-2112167, CNS-2100237, CNS-2120019, the University of California Office of the President, and the University of California San Diego's California Institute for Telecommunications and Information Technology/Qualcomm Institute. This work used Delta CPU, GPU and Storage through allocation CIS230153 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support program, which is supported by National Science Foundation grants 2138259, 2138286, 2138307, 2137603, and 2138296. The authors are partially supported by National Science Foundation under Grant No. 2107189, 2313105, 2430539, Hellman Fellowship, and Intel Rising Star Faculty Award. The authors would also like to thank anonymous reviewers for valuable feedback to improve the manuscript.

References

[1] Dylan M Anstine and Olexandr Isayev. Generative models as an emerging paradigm in the chemical sciences. *Journal of the American Chemical Society*, 145(16):8736–8750, 2023. 1

[2] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. GAN Dissection: Visualizing and understanding generative adversarial networks. In *ICLR*, 2019. 2

[3] David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. Rewriting a deep generative model. In *ECCV*, 2020. 2

[4] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *NeurIPS*, 2018. 2

[5] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023. 8

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021. 7

[7] Zheng Ding, Yifan Xu, Weijian Xu, Gaurav Parmar, Yang Yang, Max Welling, and Zhuowen Tu. Guided variational autoencoder for disentanglement learning. In *CVPR*, 2020. 2

[8] Rohit Gandikota, Joanna Materzyńska, Tingrui Zhou, Antonio Torralba, and David Bau. Concept sliders: Lora adaptors for precise control in diffusion models. In *ECCV*, 2024. 2

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 5

[10] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 5

[11] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? In *ICLR*, 2023. 1

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 5

[13] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017. 2

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop*, 2021. 7

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2, 5

[16] Aya Abdelsalam Ismail, Julius Adebayo, Hector Corrada Bravo, Stephen Ra, and Kyunghyun Cho. Concept bottleneck generative models. In *ICLR*, 2024. 1, 2, 3, 4, 5, 7, 8

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 5, 7

[18] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 5, 7

[19] Pushkal Katara, Zhou Xian, and Katerina Fragkiadaki. Gen2Sim: Scaling up robot learning in simulation with generative models. In *ICRA*, 2024. 1

[20] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *ICML*, 2020. 1, 2, 4

[21] Sonia Laguna, Ričards Marcinkevičs, Moritz Vandenhirtz, and Julia E. Vogt. Beyond concept bottleneck models: How to make black boxes intervenable? In *NeurIPS*, 2024. 2

[22] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *CVPR*, 2020. 5

[23] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *ECCV*, 2024. 2

[24] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 5

[25] Emanuele Marconato, Andrea Passerini, and Stefano Teso. GlanceNets: Interpretable, leak-proof concept-based models. In *NeurIPS*, 2022. 2

[26] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In *NeurIPS*, 2022. 2

[27] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 7

[28] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *ICLR*, 2022. 2

[29] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Memory-based model editing at scale. In *ICML*, 2022. 2

[30] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In *ICML*, 2017. 7

[31] Tuomas Oikarinen and Tsui-Wei Weng. CLIP-Dissect: Automatic description of neuron representations in deep vision networks. In *ICLR*, 2023. 2

[32] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *ICLR*, 2023. 1, 2

[33] Rishubh Parihar, VS Sachidanand, Sabariswaran Mani, Tejan Karmali, and R Venkatesh Babu. PreciseControl: Enhancing text-to-image diffusion models with fine-grained attribute control. In *ECCV*, 2024. 2

[34] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016. 3

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 5, 8

[36] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021. 1

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2

[38] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. 1, 2

[39] Divyansh Srivastava, Ge Yan, and Tsui-Wei Weng. VLG-CBM: Training concept bottleneck models with vision-language guidance. In *NeurIPS*, 2024. 1, 2

[40] Chung-En Sun, Tuomas Oikarinen, Berk Ustun, and Tsui-Wei Weng. Concept bottleneck large language models. In *ICLR*, 2025. 2

[41] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5

[42] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020. 5

[43] An Yan, Yu Wang, Yiwu Zhong, Chengyu Dong, Zexue He, Yujie Lu, William Yang Wang, Jingbo Shang, and Julian McAuley. Learning concise and descriptive attributes for visual recognition. In *ICCV*, 2023. 1, 2

[44] 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 *CVPR*, 2023. 2

[45] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *ICLR*, 2023. 1, 2

[46] Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lio, and Mateja Jamnik. Concept embedding models: Beyond the accuracy-explainability trade-off. In *NeurIPS*, 2022. 1

[47] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 8

[48] Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. In *ECCV*, 2022. 5, 8