Appendix: 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

In this appendix, we provide comprehensive implementation details and more analysis experiments. Towards reproducible research, we will release our complete codebase and pretrained weights. The appendix is organized as follows:

- Section A: Limitations
- Section B: Implementation Details
  - Datasets (Sec. B.1)
  - Architecture details (Sec. B.2)
  - Training details (Sec. B.3)
  - Human evaluation details (Sec. B.4, Fig. 1)
  - Miscellaneous details (Sec. B.5)
- Section C: Experiments
  - Extended comparisons (Sec. C.1, Table 1)
  - Extended analysis (Sec. C.2, Table 2-4, Fig. 2-5)
  - Efficiency analysis (Sec. C.3, Table 5, Table 6)

A. Limitations

While the steerability metric quantifies whether the target concept is obtained in the intervened image, it does not quantify if other concepts (outside the known concepts) have changed. For example, an intervention from "not smiling" to "smiling" may lead to a smiling image with different hair color. This cannot be easily identified with an automated metric, and it is challenging and expensive to design an unbiased human evaluation given its subjective nature. It will be interesting to address this in future work.

B. Implementation Details

B.1. Datasets

For the CelebA dataset, we follow CBGM [3] and use 8 balanced concepts for the balanced concept regime. We determine these concepts based on the fraction of number of images that contain a particular concept w.r.t. number of images that do not contain that concept. The 8 concepts for CelebA are "smiling", "male", "heavy makeup", "mouth open", "attractive", "wearing lipstick", "high cheekbones", and "wavy hair". For CelebA-HQ, we have the same 8 concepts with the exception of "wavy hair", which is replaced by "arched eyebrows". For CUB dataset, we use the 10 most balanced concepts: "small size (5 to 9 inches)", "perching-like shape", "solid breast pattern", "black bill color", "bill length shorter than head", "black wing color", "solid belly pattern", "all purpose bill shape", "black upperparts color", and "white underparts color", following CBGM [3]. For the steerability metric, we consider 16 and 20 target concepts for CelebA and CUB respectively since they are binary concepts.

B.2. Architecture details

For base generative models with vector latents or small spatial latents like StyleGAN2 or DDPM, we use a 4-layer MLP (with batch norm and leaky ReLU) for both CB-AE encoder E and decoder D. For models with larger spatial latents like GAN or PGAN, we use 4 convolution (and transposed convolution) layers with batch norm and leaky ReLU for the CB-AE encoder E (and decoder D). CC has the same architecture as the CB-AE encoder E.

B.3. Training details

For GANs, we use the training procedure as detailed in the main paper. For the DDPM diffusion model, we use saved generated images instead of generating the images at training time since DDPM generation is relatively slower than GANs. Further, we follow the diffusion model noising procedure where, at each training iteration, we choose a random timestep t and add the corresponding level of noise to the generated image before passing it through first part of the generative model g_1 (UNet encoder for DDPM). Since the CB-AE/CC would be used at different steps of denoising, it is trained using noised latents (instead of only clean latents from clean images). For GANs, g_2 produces an image while DDPM's g_2 predicts the estimated noise. So, we use the initial clean image to obtain the pseudo-label from M instead of the output of g_2. Apart from this, we follow the same training procedure as discussed in the main paper. While we use the noising techniques from DDPM, the training losses

*Equal contribution

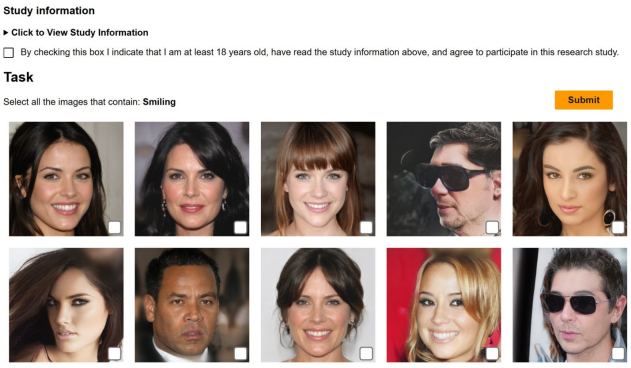

Figure 1. User interface shown to Amazon Mechanical Turk users. We ask users to click on images which match the displayed concept.

Table 1. **Per-concept steerability comparison** on CelebA dataset. Results for baseline intervention methods are from CBGM [3]. Note that average results in the main paper are over 16 target concepts, but here we compare with the available CBGM results.

Concept	High Cheekbones	Male	Mouth Open	Smiling	Wavy Hair
Baseline Intervention Methods					
CGAN [4]	5.8	6.0	6.1	3.6	13.5
ACGAN [5]	11.8	9.3	13.5	14.3	8.4
CB-GAN [3]	9.8	53.7	8.2	25.8	30.5
Our Methods					
CB-AE-GAN	48.1	35.0	51.3	64.5	27.6
CB-AE-GAN+opt-int	**66.0**	**72.3**	**81.3**	**67.3**	**38.1**
CC-GAN+opt-int	50.9	54.8	78.5	53.8	23.4
Baseline Intervention Methods					
CF-DDPM [2]	8.3	10.2	7.2	7.1	3.8
CB-DDPM [3]	11.7	14.8	13.9	15.1	10.3
Our Methods					
CB-AE-DDPM	15.7	39.6	34.9	29.8	21.0
CB-AE-DDPM+opt-int	51.3	**51.4**	**73.5**	58.8	45.4
CC-DDPM+opt-int	**61.9**	42.6	63.3	**64.0**	**65.9**

of DDPM are not used and only the CB-AE/CC is trained with our proposed losses.

While in original DDPM training, t is chosen from 0 (clean image) to 999 (complete noise), we restrict the choice of t from 0 to 400, similar to [1]. This is because the CB-AE has to predict the concepts and in practice, the generated images are very noisy at $t > 400$.

Based on this, at inference time, we use the CB-AE only for $t < 400$ and use the base model for $t > 400$. We also use the 50-step DDIM sampler [7] at inference time instead of the DDPM sampler since it is much faster with similar image quality. Note that DDIM converts the 1000 steps into 50 steps but retains the range of t from 0 to 999.

B.4. Human evaluation details

For our user study on Amazon Mechanical Turk to validate the automated evaluation of concept accuracy and steerability, we display 10 images at a time and ask the user to click on images that match a displayed concept c_i^+, as shown in Fig. 1. To ensure the quality of user responses, we require users to be in the United States, have $> 98\%$ approval rate, and > 10000 previously approved responses. For each set of 10 images, a user is paid \$0.05.

Table 2. **Ablation study on CB-AE training objectives** for the supervised classifier pseudo-label setting for CelebA-HQ pretrained StyleGAN2. Concept loss $\mathcal{L}_c$ and latent reconstruction loss $\mathcal{L}_{r_1}$ are not ablated since they are essential to concept prediction and AE reconstruction. $\mathcal{L}_{r_2}, \mathcal{L}_{i_1}, \mathcal{L}_{i_2}$ indicate image reconstruction loss, intervened concept loss, and intervened cyclic loss respectively from Eq. 1, 3 (main paper).

Row #	$\mathcal{L}_{r_2}$	$\mathcal{L}_{i_1}$	$\mathcal{L}_{i_2}$	Trained with M = Supervised classifiers		
				Conc. Acc. (%)	Steerability (%)	FID (↓)
1	✗	✗	✗	85.36	33.41	15.18
2	✓	✗	✗	83.40	38.68	11.27
3	✓	✓	✗	83.16	38.84	12.72
4	✓	✗	✓	86.52	36.95	18.57
5	✓	✓	✓	86.04	40.27	9.52

Table 3. **Steerability comparison when scaling image resolution** for our methods with PGAN and CelebA-HQ dataset.

Image Resolution	CB-AE	CB-AE+opt-int	CC+opt-int
256×256	29.31	32.10	47.29
512×512	26.48	34.92	36.87

B.5. Miscellaneous details

We implement our framework in PyTorch [6]. For all experiments, we use 10 CPU cores, 90 GB RAM, and a single Nvidia Tesla V100 GPU with 32 GB VRAM.

C. Experiments

C.1. Extended comparisons

We present extended per-concept steerability comparisons with CBGM [3] and other baseline intervention methods in Table 1. We compare the steerability on CelebA for the 5 concepts (out of 8) which are provided in the CBGM paper and find consistent improvements across all concepts.

C.2. Extended analysis

Ablation study. In the main paper, we performed the ablation study on CB-AE training objectives for the more challenging CLIP-zero-shot pseudo-label setting. In Table 2, we perform the same ablation study when using supervised classifiers as the pseudo-label source M. Similar to the results in the main paper, using the image reconstruction loss $\mathcal{L}_{r_2}$ leads to lower concept accuracy, higher steerability and better image quality (row #2 vs. #1, Table 2). Additionally using the intervened concept loss $\mathcal{L}_{i_1}$ improves the steerability and image quality but reduces the concept accuracy (row #3 vs. #1, Table 2). Whereas using the intervened cyclic

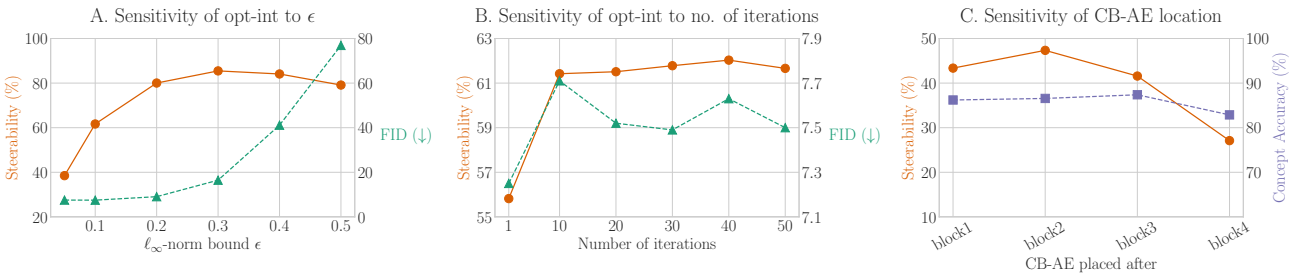

Figure 2. **A, B.** Sensitivity analysis of optimization-based interventions with CB-AE for CelebA-HQ, StyleGAN2 w.r.t. ℓ_∞-norm bound ϵ and number of iterations used in the optimization. **C.** Sensitivity analysis of CB-AE location in GAN. Note that orange circles represent steerability, green triangles represent FID, and purple squares represent concept accuracy.

Table 4. **Sensitivity of CB-AE to number of concepts** for CelebA-HQ-StyleGAN2 using TIP-few-shot for pseudo-labels. Evaluation is done only on 8 shared concepts for a fair comparison.

Trained with M = TIP-fs-128	Conc. Acc. (%)	Steerability (%)	
		CB-AE	CB-AE w/ opt-int
8 concepts	76.08	21.51	38.73
40 concepts	75.75	22.17	39.94

loss $\mathcal{L}_{i_2}$ with the image reconstruction loss $\mathcal{L}_{r_2}$ improves the concept accuracy at the expense of image quality and steerability (row #4 vs. #1, Table 2). Finally, using both of the intervention losses achieves a better tradeoff between the three metrics (row #5 vs. #3, #4, Table 2).

Scaling image resolution. Based on Table 2 and 4 (main paper), our methods achieve good performance on PGAN and DDPM when the image resolution is scaled from 64×64 to 256×256. We further validate this with CelebA-HQ PGAN trained at 512×512 in Table 3. While the steerability is relatively lower than at 256×256, we still achieve fairly good steerability, *i.e.* successful interventions with the same training time.

Sensitivity to intervention hyperparameters. We analyze the sensitivity to optimization-based intervention hyperparameters in Fig. 2A, B. Since we used the iterative randomized fast gradient sign method [8], the two hyperparameters involved are the number of iterations and the ℓ_∞-norm bound ϵ (maximum allowable perturbation). We find that as ϵ is increased, the steerability also increases but with a drop in image quality since the FID increases. Hence, we choose a small $\epsilon = 0.1$ for most of our experiments such that we obtain a good tradeoff between image quality and steerability. Further, we observe that steerability and image quality remain similar when the number of iterations are reduced from 50 to 10 iterations. However, we use 50 iterations in our experiments to allow the optimization to converge for samples that are more difficult to intervene.

Sensitivity to CB-AE location. We vary the CB-AE location in CelebA-pretrained DCGAN and report the steerability and concept accuracy in Fig. 2C. We observed that CB-AE closer to generator output hurts steerability (decreased to 27.1%) as modified latent has less influence on the output, but increased steerability up to 47.3% near the middle. On the other hand, concept accuracy remains reasonable across all locations.

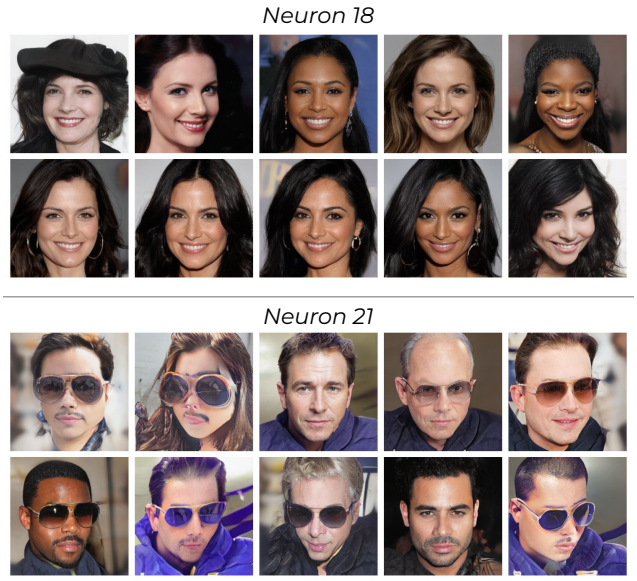

Figure 3. Top-10 images activating a particular neuron from the unsupervised concept embedding for CelebA-HQ StyleGAN2 CB-AE. We observe 'earrings' (top) and 'sunglasses' (bottom) concepts which were not present in the predefined concept set.

Unsupervised concept embedding analysis. For CB-AE trained with CelebA-HQ-pretrained StyleGAN2, we generated 5k images and collected top-10 images for each dimension in the unsupervised concept embedding being highly activated. Based on the common attributes in the top-10 images, we identified 'sunglasses' and 'earrings' (not in predefined concepts) as shown in Fig. 3.

Figure 4. Concept intervention examples for CB-AE and CB-AE with optimization-based interventions (opt-int) for CelebA-HQ-pretrained DDPM. Some cases where either or both of our methods failed are highlighted in purple.

Table 5. **Inference time analysis** for CB-AE with CelebA-HQ-pretrained StyleGAN2. Here, opt-int-k indicates optimization-based interventions with k iterations. Inference time (in milliseconds) is computed with batch size 64 on a single V100 GPU, repeated 1000 times for mean and standard deviation.

	Inference Time (ms)
Base model	170.02 ± 0.45
CB-AE reconstr.	170.70 ± 0.53
CB-AE interv.	170.27 ± 2.26
CB-AE+opt-int-10	181.68 ± 2.98
CB-AE+opt-int-50	226.01 ± 1.05

Qualitative evaluation. In Fig. 4, we show concept intervention examples of our CB-AE and CB-AE with optimization-based interventions for a CelebA-HQ-pretrained DDPM diffusion model. Unlike with StyleGAN examples (Fig. 4, 5, main paper), we find that optimization-based interventions produce relatively lower quality images compared to CB-AE interventions. We also highlight some cases where either or both of our methods failed. In these cases, we find that some other concepts like hair style change marginally or the desired concept does not change enough.

Concept interpolation. To demonstrate that our training objectives incorporate meaningful knowledge in the CB-AE, we generate images using interpolation (and extrapolation) between predicted and intervened concept vectors, as shown in Fig. 5. Concretely, for a randomly sampled noise vector z, we can compute the concept vector $c = E(g_1(z))$ using the CB-AE encoder E and the first part of the generator g_1. Then, given a target concept, we compute an intervened concept vector $c_{\text{intervened}}$ as described in the CB-AE Objective 3 (Sec. 3.1, main paper). The interpolated concept vector can be computed as $\hat{c}_{\text{intervened}} = (1-\alpha)c + \alpha c_{\text{intervened}}$ where $\alpha \in [0, 1]$ (and extrapolation for $\alpha > 1$). Then, an image can be generated using the interpolated concept vector as $\hat{x}_{\text{intervened}} = g_2(D(\hat{c}_{\text{intervened}}))$ using the CB-AE decoder D and g_2.

Overall, we observe that the CB-AE can produce smooth transitions in the image space from the original to intervened concept vectors as well as extrapolate further. However, in some of the extrapolation cases, we find changes in other concepts like hair color or skin color apart from the target concept. While it is generally undesirable for concept inter-

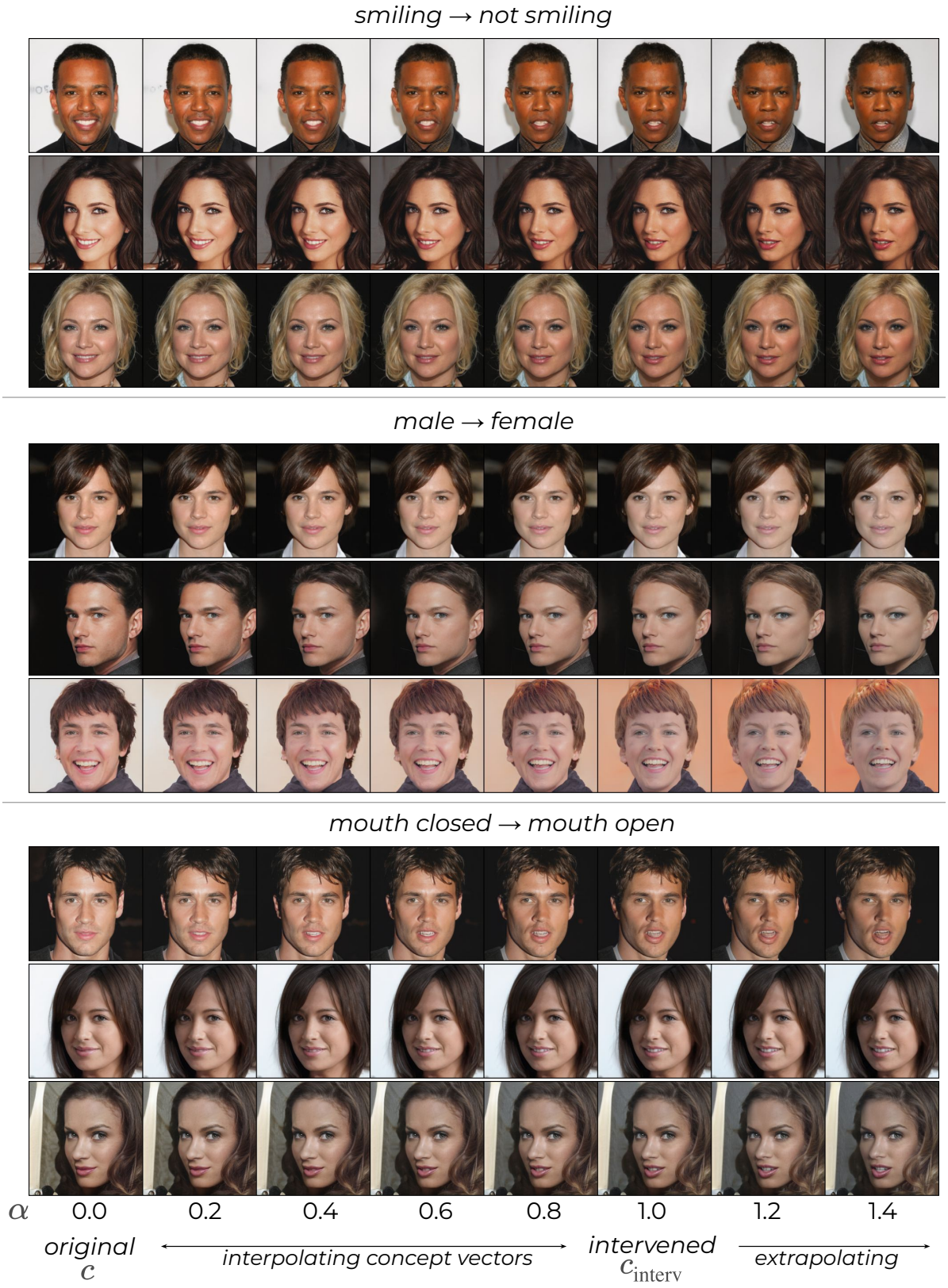

Figure 5. **Concept vector interpolation.** We interpolate between the concept vector c from the CB-AE and the intervened concept vector $c_{\text{intervened}}$ for generating images, *i.e.* $\hat{c}_{\text{intervened}} = (1-\alpha)c + \alpha c_{\text{intervened}}$ where $\alpha \in [0, 1]$. The interpolated vector $\hat{c}_{\text{intervened}}$ is passed through the CB-AE decoder D and the remaining generator g_2 to obtain the displayed images. We also show examples with extrapolation for $\alpha = 1.2, 1.4$.

Table 6. **Trainable parameters analysis** for CB-AE and CC with CelebA-HQ-pretrained StyleGAN2 w.r.t. CBGM. Reduction indicates % reduction in trainable parameters compared to CBGM.

Method	Trainable Parameters	Reduction (%)
CBGM [3]	24.77M	-
CB-AE (*Ours*)	1.64M	93.37
CC (*Ours*)	0.79M	96.77

ventions, this can be a potential tool for dataset creators or generative model developers to identify potential biases or spurious correlations between concepts.

C.3. Efficiency analysis

In Table 5, we compare the inference time of our methods with the base model. We compute the inference times for CB-AE trained with CelebA-HQ-pretrained StyleGAN2 using batch size 64 on a single V100 GPU. We repeat the inference 1000 times and report the mean and standard deviation, and find that using the CB-AE with the base model (in reconstruction mode, without interventions) and for concept interventions only causes a marginal increase in inference time. Given the number of iterations involved in optimization-based interventions, there is a relatively larger increase in inference time. However, as shown in Fig. 2, our method is effective even with 10 iterations, which adds only $\sim$11 milliseconds of inference time to that of the base model.

We also compare the number of trainable parameters in our CB-AE and CC compared to CBGM in Table 6. Due to our efficient and novel autoencoder setup, we find 93.37% and 96.77% reduction in trainable parameters for StyleGAN2 compared to CBGM [3].

References

[1] 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

[2] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop*, 2021. 2

[3] Aya Abdelsalam Ismail, Julius Adebayo, Hector Corrada Bravo, Stephen Ra, and Kyunghyun Cho. Concept bottleneck generative models. In *ICLR*, 2024. 1, 2, 6

[4] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[5] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In *ICML*, 2017. 2

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 2

[7] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 2

[8] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020. 3