Personalized Generative Models for Contextual Debiasing

Xinran Liang[1], Esin Tureci[1], Prachi Sinha[1], Ye Zhu[12], Vikram V. Ramaswamy[1], Olga Russakovsky[1]
[1]Department of Computer Science, Princeton University [2]LIX, CNRS, École Polytechnique

Abstract

Many vision datasets sample contextual patterns unevenly, causing uncommon contextual patterns to be underrepresented relative to common ones. Models trained on these datasets thus often fail to generalize to uncommon scenarios. While collecting such images in the real world may be difficult, we explore whether generating images with less frequent contexts can serve as effective training augmentation. A key challenge is guiding generations to remain close to the original dataset distribution while creating diverse images with uncommon contexts. To address this issue, we introduce ***Decoupl****ing Cont****e****xtual Patterns with* ***Gen****erations (****DecoupleGen****), a method that personalizes text-to-image diffusion models to facilitate coherent synthesis of images with rare contexts while preserving original visual details. The generated images contain semantically meaningful content and remain visually aligned with the original datasets. We further apply verification constraints to ensure relevance of the augmented data. We evaluate our approach on object classification and recognition tasks on complex scene datasets. Our experiments demonstrate consistent improvements over previous approaches, and our analyses identify factors underlying these improvements.*

1. Introduction

Improvements in computer vision models are fueled, in part, by advances in data collection methods, resulting in ever-larger datasets. However, despite diverse curation strategies, these datasets still contain multiple forms of bias [9, 10, 51, 58, 64]. One way these biases can manifest is through common co-occurring patterns; for example, skis frequently appear with people, while images of skis alone are rare. These *contextual biases* often cause models to fail to generalize to uncommon scenarios; going back to our example, these models often have low accuracy on images of skis without people [53]. While recent general-purpose vision-language models may exhibit greater robustness to such biases due to large-scale pretraining [7, 52], many real-world and industrial applications still utilize smaller specialized models given compute and memory constraints [12, 36], motivating the need to improve generalization of compact classifiers.

(a) Original (b) Txt2Img [47] (c) Inpaint [47] (d) DecoupleGen

Figure 1. **An example illustrating challenges in contextual debiasing.** Our goal is to generate an image of skis without a person. Using common Txt2Img models typically produces images with visual appearances that differ substantially from those in the original dataset (Figure 1b). Applying existing local image editing methods, e.g. inpainting, often generates less plausible results (Figure 1c). Modifying other elements related to person is necessary to ensure visual coherence of the entire image (Figure 1d).

One way to address these biases is through training objectives, such as reweighting, that penalize models for over-relying on contextual cues [8, 14, 17, 53]. However, these methods are limited to using only the original datasets and can often result in poorer performance on in-context samples. An alternative way to tackle these biases is to leverage recent advances in generative models [24, 25, 41, 45, 47] to synthesize images with diverse content. These models can be used to edit images to remove common contexts, or generate images with difficult contexts. Recent works explore the potential to incorporate these generations as data augmentation to improve recognition performance. However, they usually focus on classifying a single object in an image [6, 15, 21, 23], thus are not well equipped to handle interactions between multiple objects in real world scenarios, limiting their potential scope and applicability.

Constructing images that contain objects in uncommon contexts using generative models can be challenging. Since objects and scenes usually interact with each other in real images, manipulating contextual patterns requires handling their complex relationships in order to generate natural looking images. Furthermore, in order for these samples to improve model performance, the resulting generations should ideally remain close to the distribution of the original dataset. Consider the COCO [11] dataset and, in particular, images containing skis, the person within the image usually interacts with the skis in some way (Figure 1a). When

generating images with rare contexts such as skis without people, pre-trained text-to-image (Txt2Img) models often produce images with viewpoints and backgrounds backgrounds that differ substantially from the original dataset (Figure 1b), which may not be relevant to downstream tasks. On the other hand, local editing methods, e.g. inpainting [9, 47], which takes mask as input to explicitly remove or replace the context, are unable to deal with interactions among objects. Removing the person in this manner results in the skis floating (Figure 1c). Instead, generating a plausible version of the image without the person requires repositioning the skis as well, as shown in Figure 1d.

To address this issue, we introduce **Decoupl**ing Cont**e**xtual Patterns with **Gen**erations (**DecoupleGen**) (Figure 2). We adapt pre-trained diffusion models to downstream datasets by fine-tuning them on images containing common contextual patterns, and then generate samples containing uncommon contextual patterns. Specifically, inspired by personalization methods of diffusion models for subject-driven generations [18, 48], we learn new word tokens that capture visual details from original images. We then prompt the fine-tuned diffusion models with the learned text embeddings to manipulate contextual patterns. The resulting generations are natural looking and visually similar to images from the original datasets. To obtain generations more relevant to downstream recognition tasks, we further select synthesized images that do not contain common contextual patterns, ensuring that the additional training samples capture underrepresented groups in the original datasets and supplement out-of-context samples.

We evaluate our method on visual recognition tasks using real-world object and scene datasets [11, 22]. Our experiments show that DecoupleGen outperforms existing approaches that modify loss functions or leverage generative models for data augmentation by 1.5%, improving over standard classifier by 14.2% on the NICO dataset and by 5.4% on the COCO dataset. Our analyses show two distinctive advantages of DecoupleGen: first, synthesized images from our method remain closer to the downstream dataset, indicating better alignment; and second, our method is better accounts for complex relationships between objects and scenes depicted in natural images. [1]

2. Related Work

Addressing model bias via dataset manipulations. Prior work has shown that existing datasets are not fully representative of the real world [31, 56], causing models trained on them to struggle to generalize to uncommon scenarios. We focus specifically on failures where vision models cannot recognize objects without their co-occurring contexts, a limitation observed in both deep neural network classifiers [13, 53] and recent large vision-language models [34, 35]. One approach to address these limitations is to rebalance training datasets through methods such as reweighting [14, 29], which assigns higher loss weights to underrepresented samples, and over-sampling [8, 17], which duplicates such samples during training. While these methods manipulate the effective training distribution using real images, we instead use synthetic samples generated by diffusion models to achieve similar goals. Unlike Ramaswamy et al. [44], which generates additional samples for facial recognition, our method focuses on manipulating contextual patterns and is therefore more broadly applicable to real-world scene images.

Leveraging generated data for downstream tasks. With recent advances in generative models, using synthetic data for model training or evaluation is not entirely new. Previous works [16, 27, 37, 54, 55] develop self-supervised representation learning methods using generations from GANs [19] or diffusion models [25]. Synthetic images and labels from generative models have been used effectively in providing diverse augmentations and improving model performance in classification tasks [6, 15, 21, 23, 57]. Generated images can be used to create test sets for classifier evaluations [43, 62]. Our work falls into this category of approaches using generated data for model training. While these works perform generation or editing at the image level for classification or self-supervised tasks, our method instead considers relationships between multiple objects and scenes captured in real images during generation.

Generative models for local image editing. A natural approach for manipulating contextual patterns is to provide object masks to generative models such as inpainting [42]. The task of local image editing has recently received significant attention and seen substantial progress [1, 3, 38, 47]. Consider an image of a person jumping on a skateboard in midair. While such models can feasibly modify local content – for example, the jumping person – they cannot adequately adjust related information in the remaining regions, such as the skateboard suspended in the air, due to the complex relationships between objects and scenes. In contrast, our method can generate more realistic images with physically plausible scenes (as discussed in Section 4.2).

Finetuning diffusion models for personalization. Recent works, e.g., Textual Inversion [18], DreamBooth [48], and Custom Diffusion [32], explore fine-tuning text-to-image diffusion models for personalization using a small set of images and learning new word embeddings. These methods reconstruct visual details of a single subject – such as a person, animal, or object – from one or a few training images and enable diverse generations through different text prompts. Unlike these approaches, which reconstruct the primary subject from training images, our goal is to pre-

[1] In this work, "personalization" refers to subject-driven visual generation techniques that extract and reproduce specific visual concepts from one or a few reference samples.

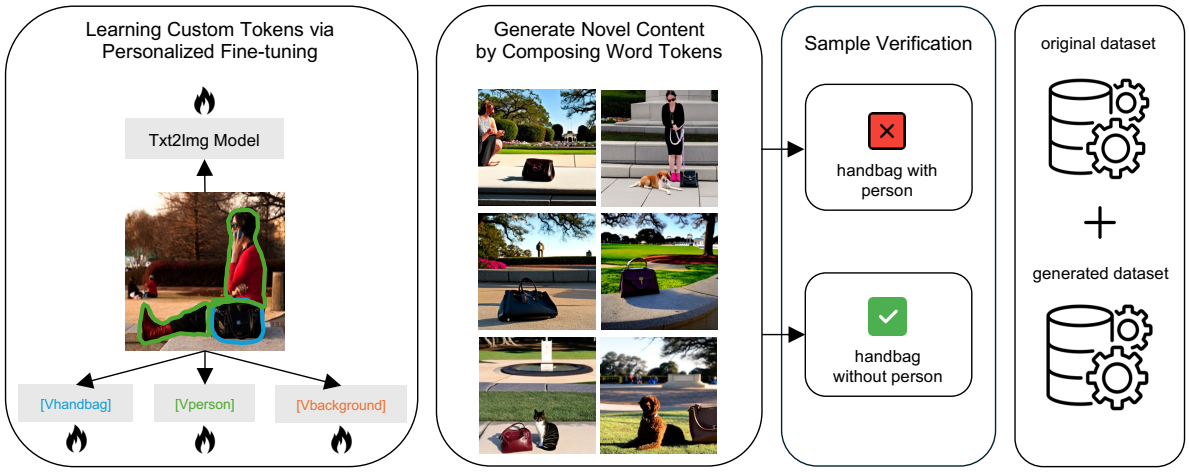


Figure 2. **Overview of DecoupleGen.** We first fine-tune text-to-image diffusion models on each co-occurring image by learning new word tokens to describe visual details. We then synthesize images by combining these learned text embeddings to manipulate contextual patterns. Our generations are visually coherent and maintain visual information similar to original image. Finally, we select relevant and meaningful generations via a verification process and incorporate them in classifier training.

serve visual details from regions unrelated to the manipulated class categories and use them to generate new object-context combinations. Following Avrahami et al. [4], our method learns new word tokens to describe corresponding regions in the training image and reconstruct similar content during generation (as discussed in Section 3.2).

3. Contextual Debiasing via Generative Model

We aim to augment the original dataset with generations that do not contain common contextual patterns, while encouraging these generations to be visually coherent and close to the original dataset distribution. To achieve this goal, we perform text-to-image generations while maintaining visual information by adapting personalization methods for diffusion models on individual samples.

We propose **Decoupl**ing Cont**e**xtual Patterns with **Gen**erations (**DecoupleGen**) and provide an overview of the approach in Figure 2. Our method consists of three steps: Starting from real images with common contextual patterns, we first fine-tune text-to-image diffusion models to learn new word tokens that describe different image components (biased object, contextual object, and background). We then guide generation process to create images without common contextual patterns by composing these learned word tokens. Finally, we filter the synthesized images using a pretrained object detector to remove samples that may still contain common contextual patterns. We augment the real dataset with generated images for classifier training.

3.1. Problem Formulation: Image Generation

We aim to teach the model to separate biased object classes `b` (e.g., skateboard) from their contextual categories `c` (e.g., person) by generating images that remove common co-occurrence patterns. Specifically, we formulate contextual debiasing as editing images containing `(b, c)` by either removing `c` or replacing it with alternative semantic categories while preserving the remaining visual content.

What is hard about this problem? Images encode complex object–scene relationships, making direct edits prone to producing implausible results (e.g. a floating skateboard). Meanwhile, off-the-shelf Txt2Img models often fail to preserve viewpoint and other image-specific details. Our goal is therefore to generate images that reposition objects while maintaining the original visual characteristics.

3.2. Adapting Personalized Diffusion Models

We take inspiration from recent personalization methods [18, 32, 48] that fine-tune special tokens to reconstruct subjects. Specifically, we learn a new "background" token that captures visual details of the remaining pixels.

At training time, given an image with masks of `b` and `c`, we learn a new `[Vbackground]` token to reconstruct visual details specified by remaining pixels not belonging to either `b` or `c`. We use the masks of `b` and `c` as input, while the background mask is defined as all remaining pixels outside the two classes. The word tokens and the diffusion model are optimized to reconstruct the image from the text prompt "a photo of `[Vhandbag]` and `[Vperson]` and `[Vbackground]`". Following Avrahami et al. [4], we use the training objective $\mathcal{L} = \mathcal{L}_{\text{maskrec}} + \lambda \mathcal{L}_{\text{crossattn}}$, where

$$\mathcal{L}_{\text{maskrec}} = \mathbf{E}_{z,m,\epsilon,t} \left[\|\epsilon \odot M_m - \epsilon_\theta(z, t, \mathrm{V}_m) \odot M_m\|^2 \right]$$

is the variational lower bound loss between the predicted noise ϵ_θ and the added noise $\epsilon \sim \mathcal{N}(0, 1)$, where z denotes the noisy latent at diffusion timestep t. This objective encourages faithful reconstruction of each masked region M_m

from the corresponding text token $\texttt{V}_m$. In addition,

$$\mathcal{L}_{\text{crossattn}} = \mathbf{E}_{z,m,t}\left[\|\text{CrossAttn}_\theta(z, t, \texttt{V}_m) - M_m\|^2\right]$$

is the cross attention loss that encourages each word token $\texttt{V}_m$ to attend only to its corresponding masked region M_m and not to other regions.

At inference time, we generate images from the finetuned diffusion model by composing the learned background token with standard text tokens describing object categories. The custom token `[Vbackground]` captures contextual visual details and scene appearance, while object categories are represented using regular text tokens. For example, to remove the common contextual class `person` from handbag images, we use the prompt "a photo of handbag at `[Vbackground]`". To replace the contextual class with another category, we use prompts such as "a photo of handbag and dog at `[Vbackground]`", where `dog` can be substituted with other alternative classes.

A verification step on synthesized samples is helpful, since our approach for manipulating contextual patterns does not guarantee successful generations in all cases. After generation, we annotate each image using an off-the-shelf pre-trained semantic segmentation model [28] to predict class labels. We discard images that do not have the expected labels during sample selection. For example, given an image with both skateboard and person, successful generations containing skateboard but not person will be included. This verification process ensures that generated samples contain information useful for downstream tasks.

Similarly, our pipeline can be applied to image classification tasks in which each image contains an object from a single class. Our goal in this case is to generate images that preserve the visual appearance of objects to the original image but place them in diverse background scenes. Accordingly, our finetuning process learns two new word tokens `[Vclass]` and `[Vcontext]`. The word tokens and diffusion models are optimized to reconstruct the image from the text prompt "a photo of `[Vclass]` and `[Vcontext]`" with the same loss function as above. At inference time, we use prompts of the form "a photo of `[Vclass]` in `context-name`" for generation. We use this technique in our experiments, although our pipeline can be integrated with other approaches that learns to reconstruct visual details from masked regions of a single image.

4. Experiments

We aim to generate synthetic images to increase sampling of the underrepresented regions of the dataset, while keeping generations as close to the original training data distribution as possible. Our goal is to improve model accuracy on underrepresented groups while preserving the accuracy on well-represented groups. In this section, we first demonstrate that our method outperforms existing approaches that modify training objectives on original datasets, as well as approaches that leverage generative models to synthesize additional training samples (Section 4.1). We then analyze factors contributing to the advantages of our method over alternative generation-based approaches (Section 4.2).

Implementation Details. For image generation, we use Stable-Diffusion-v2 [47] as the base model with default hyperparameters. For classifier training, we initialize a ResNet50 [20] with ImageNet [49] pre-trained weights and perform end-to-end training on each downstream dataset. For all classification models, we tune learning rate and weight decay hyperparameters based on validation performance and report the best results.

4.1. Quantitative Comparisons

4.1.1. Image Classification

Experimental setup. We use NICO dataset [22] for single-object image classification. The dataset contains 60 object classes across 6 contexts, with a highly imbalanced distribution of object-context combinations (which we treat as groups). Our goal is to improve performance on underrepresented groups. We apply our pipeline independently to each training image, learning image-specific `[Vclass]` and `[Vcontext]` tokens during fine-tuning. At inference time, we preserve the learned `[Vclass]` token while replacing `[Vcontext]` with alternative textual context descriptions to generate new object-context combinations.

Following previous work Hemmat et al. [23], we synthesize images such that the combined dataset is balanced across groups, resulting in a total of 229k synthetic samples consistently used across different methods. We do not further apply verification constraints, since we found that samples generated by all methods considered in our experiments are visually plausible for single-object generation.

We compare our method against classifiers trained with different categories of learning algorithm [60]: (1) subgroup robust methods: **GroupDRO** [50], **LISA** [61]; (2) domain invariant learning: **IRM** [2]; (3) imbalanced learning: **BSoftmax** [46], **CRT** [30]; and (4) data augmentation: **Mixup** [63]. We use benchmarking results reported in previous work [60]. We also compare to models that utilize generative models to obtain additional training samples: **Txt2Img** with captions formatted as "a photo of `class-name` in `context-name`" as text prompt, and **Feedback Guidance** [23] that uses the same captions as Txt2Img and additional entropy guidance from a standard classifier. We implemented both methods and our reproduced results match those reported in previous work [23].

To evaluate model performance, we use overall accuracy (**Acc**) and worst group accuracy (**WGA**) following previous work [23, 60]. Overall accuracy is computed over all samples, while worst-group accuracy is computed over all subgroup accuracies.

Experimental results. Our results in Table 1 show that DecoupleGen either matches or outperforms existing approaches on NICO dataset, while also improving upon the overall accuracy of the standard classifier, which previous methods failed to achieve. DecoupleGen achieves an overall accuracy of $85.8\% \pm 0.1$, compared to $85.3\% \pm 0.3$ for the standard classifier, while matching the state-of-the-art worst-group accuracy of 49.2%, previously achieved by the feedback-guided generation approach [23].

Method	Num Generations	Acc	WGA
Standard	0	85.3 ± 0.3	35.0 ± 4.1
Mixup [63]	0	84.0 ± 0.6	42.7 ± 1.4
GroupDRO [50]	0	83.2 ± 0.4	37.8 ± 1.8
LISA [61]	0	84.7 ± 0.3	42.7 ± 2.2
IRM [2]	0	84.4 ± 0.7	40.0 ± 0.0
BSoftmax [46]	0	84.0 ± 0.5	40.4 ± 0.3
CRT [30]	0	85.2 ± 0.3	43.3 ± 2.7
Txt2Img [47]	$229k$	85.2 ± 0.1	33.5 ± 1.1
Feedback Guidance [23]	$229k$	85.3 ± 0.3	$\mathbf{49.2} \pm 1.0$
DecoupleGen (ours)	$229k$	$\mathbf{85.8} \pm 0.1$	$\mathbf{49.2} \pm 0.7$

Table 1. **Quantitative comparisons on NICO dataset.** Our method achieves competitive or better performance in WGA compared to previous work, without sacrificing in accuracy.

4.1.2. Object Recognition

Experimental setup. We use COCO-Stuff [11] dataset, containing complex real-world images, for multi-object recognition. Singh et al. [53] defined a bias score to identify biased categories and their co-occurring context; we use the same set of biased classes for evaluation. Our goal is to improve model performance on images in which biased classes appear without their common contextual labels.

To generate samples that manipulate contextual patterns, we follow the semantic label hierarchy of the original dataset to identify sets of similar categories within the same superclass for replacement. We generate 1 output from each replacement or removal prompt to avoid duplicates and encourage diversity in generated semantic content. We then consider a generation successful if its annotation labels contain only $\texttt{b}$ (target class) without $\texttt{c}$ (common context label), and retain only the successful generations. This process results in 96k generated samples in total.

We evaluate classifier performance using average precision (AP) for each class. We consider several evaluation metrics following previous work [53]: (1) **Exclusive mAP** computed over 20 biased classes on images where $\texttt{b}$ *never* co-occurs with $\texttt{c}$; (2) **Cooccur mAP** computed over 20 biased classes on images where $\texttt{b}$ *always* co-occurs with $\texttt{c}$; (3) **Unbiased mAP** computed over the remaining 60 object classes in COCO; and (4) **All mAP** is computed over all 171 object and context classes.

Comparison with loss-function based approaches. In addition to the **Standard** classifier that is trained on the original dataset with supervised loss, we compare our method against two sets of previous work. The first set includes approaches that modify training objectives on original datasets in order to improve performance on underrepresented groups. Following Singh et al. [53], we consider **Class balancing loss**, **Negative penalty**, **Weighted loss**, **Remove cooccur labels**, **Remove coocccur images**, **CAM**, which uses class activation map [65] as weak localization supervision during classifier training, and **Feature split**, which enforces classifier to learn separate feature spaces for objects and their contexts.

When comparing against these approaches utilizing alternative loss functions, we use all successful generations containing uncommon contextual patterns (96k) to augment the original training set in classifier training and report the results in Table 2 (top half). Our method achieves better performance on all evaluation metrics compared to these loss-function based approaches. In particular, we observe a 1.3% improvement in exclusive mAP, increasing from 28.21% with the re-weighting objective to 29.52% with our method, as well as a 1.6% improvement in cooccur mAP, increasing from 64.80% for the standard classifier to 66.43% with our method. While previous approaches improve performance on exclusive images at the expense of cooccur, unbiased, and all mAP, our method achieves consistent improvements across all evaluation metrics.

Comparison with other generation-based approaches. Since we generates additional training samples for downstream tasks, we compare against several alternative generative approaches. We first consider **Txt2Img** with three variants, where captions describing the target biased classes (**Txt2Img Target Class**), co-occurring images (**Txt2Img Cooccur**), or exclusive images (**Txt2Img Exclusive**) are used to generate synthetic images [33, 47]. We also compare against **Feedback Guidance** [23], which uses the same captions as **Txt2Img Target Class** together with guidance from a standard classifier, since classifier entropy reduces to classification loss in the multi-label classification setting. In addition, we compare against **Stable Diffusion Inpainting**, an inpainting model trained to reconstruct masked regions from the remaining unmasked areas [47], and **Blended Latent Diffusion** [5], where editing masks are applied in the down-sampled latent space and masked and unmasked regions are blended throughout the diffusion process.

To compare the effectiveness of different generation-based methods, we fix the number of generated samples added to each target biased class across all baselines, matching the number of co-occurring images in the original dataset (18k in total). This is motivated by the difficulty of successfully removing or replacing contextual classes using local image editing methods (discussed further in Section 4.2). We apply a consistent filtering procedure across all methods considered in our experiments, retaining only successful generations.

Method	Num Generations	Exclusive mAP	Cooccur mAP	Unbiased mAP	All mAP
Standard	0	24.10 ± 0.13	**64.80** ± 0.08	**72.54** ± 0.04	**55.97** ± 0.02
Real Cooccur	0	22.69 ± 0.53	64.00 ± 0.57	71.28 ± 0.24	54.82 ± 0.31
Class balancing loss [14]	0	25.03 ± 0.06	64.67 ± 0.05	72.35 ± 0.04	55.79 ± 0.02
Negative penalty [53]	0	23.71 ± 0.03	64.61 ± 0.07	72.19 ± 0.05	55.68 ± 0.02
Weighted loss [53]	0	**28.21** ± 0.15	59.89 ± 0.16	72.13 ± 0.01	55.44 ± 0.00
Remove cooccur labels [53]	0	24.17 ± 0.06	64.59 ± 0.06	71.59 ± 0.04	55.30 ± 0.01
Remove cooccur images [53]	0	28.09 ± 0.34	60.09 ± 0.08	72.25 ± 0.04	55.50 ± 0.04
CAM [53]	0	25.09 ± 0.23	64.45 ± 0.04	72.45 ± 0.04	55.87 ± 0.01
Feature split [53]	0	26.76 ± 0.52	64.10 ± 0.01	72.39 ± 0.07	55.74 ± 0.02
Txt2Img Cooccur [47]	$18k$	24.66 ± 0.36	64.41 ± 0.08	72.87 ± 0.14	56.11 ± 0.20
Txt2Img Exclusive [47]	$18k$	25.01 ± 1.04	65.26 ± 0.09	72.66 ± 0.02	55.96 ± 0.04
Txt2Img Target Class [47]	$18k$	23.90 ± 0.40	64.87 ± 0.15	72.44 ± 0.02	55.92 ± 0.02
SD Inpainting [47]	$18k$	25.34 ± 1.13	64.77 ± 0.09	72.61 ± 0.01	55.85 ± 0.05
Blended LD [3]	$18k$	24.87 ± 0.57	64.77 ± 0.05	72.71 ± 0.01	55.94 ± 0.01
Feedback Guidance [23]	$18k$	25.55 ± 0.01	65.01 ± 0.06	72.68 ± 0.01	56.11 ± 0.01
DecoupleGen (ours)	$18k$	**27.20** ± 0.32	**65.56** ± 0.04	**73.09** ± 0.03	**56.37** ± 0.04
DecoupleGen (ours)	$96k$	**29.52** ± 0.21	**66.43** ± 0.04	**73.52** ± 0.05	**57.00** ± 0.06

Table 2. **Quantitative comparisons on COCO dataset.** We compare the performance of DecoupleGen against methods that modify training objectives on the original dataset, where we include all successful generations from our method, totaling 96k for classifier training (top). We also compare DecoupleGen performance to different generation-based approaches, where we keep the number of additional samples for each target biased class consistent across methods, totaling 18k generations (bottom). DecoupleGen achieves the best performance across all evaluation metrics compared to existing approaches in both settings. The best result within each block is bolded.

Table 2 shows that DecoupleGen improves over the standard method with 18k additional images, and outperforms other generative data augmentation approaches across all evaluation metrics. The largest improvement is observed in exclusive mAP: DecoupleGen improves over Feedback Guidance by 1.6%, achieving an mAP of 27.2% compared to 25.6%. Thi improvement in exclusive mAP is especially noteworthy, indicating that our generations provide useful information for underrepresented groups in the dataset.

4.2. Why are DecoupleGen Images Better?

Our goal in this section is to gain insight into the advantages provided by the personalized fine-tuning approach compared to alternative generative methods. Through qualitative analysis, we first illustrate with examples that common local image editing models perform poorly on the contextual debiasing problem for complex object and scene images. We then analyze limitations of popular pre-trained Txt2Img models in ensuring that generations remain close to the downstream dataset distribution.

Limitations of local image-editing models. We found that a major limitation of local image editing models is their low success rate. Through qualitative inspection, we observe that many generated images either contain visual artifacts caused by local modifications, resulting in lower image quality, or fail to produce semantically reasonable content that reflects natural scenes. Consequently, these generations are intuitively less helpful for downstream classifier performance. Figure 4 shows several such examples.

When a person interacts with objects in the original image, modifying the person often requires repositioning the objects as well. For example, local image editing models may generate images in which a wine glass appears to float in the air (Figure 4b), whereas our approach can place the wine glass onto different surfaces, producing physically more plausible images (Figure 4d). Similarly, when the generation process involves changing the background scene, local image editing models often fail to maintain global visual consistency. In contrast, our approach can modify contexts such as changing a road into mud while preserving image quality and avoiding the visual artifacts observed in alternative approaches.

Pretrained Txt2Img models fail to adapt to downstream dataset distribution. We analyze synthetic images generated by DecoupleGen and by pre-trained Txt2Img models, including Txt2Img Target Class and Feedback Guidance. We compare their distributions to real samples from the validation split of the original dataset using UMAP [39].

In Figure 3, we show UMAP projections for two representative object classes, `wine glass` and `microwave`, which have high co-occurrence with `person` and `cabinet`, respectively, in the real dataset. The UMAP distributions of the real dataset show that, for `wine glass`, exclusive and co-occurring images form relatively distinct clusters. For `microwave`, however, the exclusive and co-occurring images are less clearly separated.

In both cases, compared to DecoupleGen generations, Target Class and Feedback-Guided generations based on

pre-trained Txt2Img models produce images that lie significantly farther from the real data distribution. This supports our intuition that, without constraining the generation process, images synthesized by pre-trained models can exhibit a substantial distribution gap from the real dataset in our downstream task. These results highlight the importance of the personalization-based fine-tuning process in DecoupleGen, which learns new word tokens to preserve visual details from real samples and thereby better adapts generative models to downstream datasets.

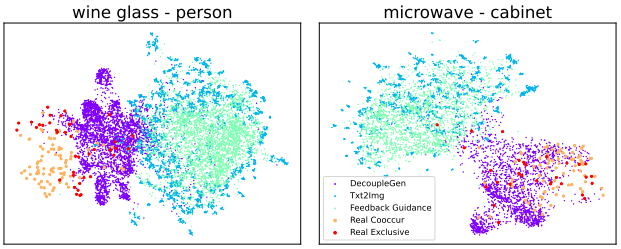


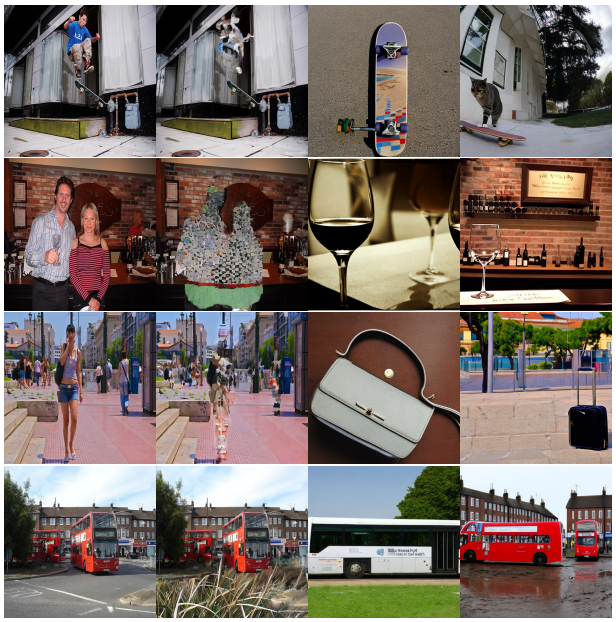
Figure 3. **UMAP visualizations of real samples and generations.** We plot real exclusive and cooccur images from validation split, and synthetic samples from different generation approaches. Generations from DecoupleGen overall stay closer to real data distribution than those from other methods, thus providing information more relevant to downstream tasks.

Figure 4 additionally shows qualitative examples. From visual inspection, we observe that DecoupleGen produces images containing objects in complex scenes that better mimic real-world scenarios (Figure 4d). In contrast, Feedback-Guided generations based on Txt2Img models depict the target objects more prominently within simpler scenes and exhibit visual appearances that differ substantially from the real dataset (Figure 4c).

We note that, while images synthesized by DecoupleGen generally remain closer to real samples, they also expand beyond the original data distribution, with some samples lying farther from real exclusive images. One contributing factor may be limitations of our fine-tuning process in accurately learning new word tokens that capture fine-grained visual details, especially when target objects are very small (we discuss failure cases in the Supplementary Material). Another reason is that our method can generate novel combinations of objects and context classes that do not exist in the real dataset, enabling expansion beyond the original samples. For example, a cat or dog sitting near a handbag in Figure 2 represents semantic combinations that may rarely appear in real samples.

4.3. Ablation and Analysis

Effect of selecting useful generations To study the effect of the verification step for selecting informative generations, we compare using only successful generations that manipulate contextual patterns against randomly selecting



(a) Original (b) Blended LD (c) Txt2Img FG (d) DecoupleGen

Figure 4. **Qualitative comparisons.** Local image editing models, e.g. Blended LD [5], often fail to keep generate reasonable and coherent images of complex scenes. Feedback-guided generations [23] fail to preserve similar visual appearance in natural images. Instead, DecoupleGen can address issues when person interacts with objects in the image and when generation involves transforming into a different background scene.

from all generated samples. We conduct experiments on the COCO dataset and fix the number of additional samples for each class to match the number of co-occurring images. Table 3 shows that filtering out synthetic samples containing common contextual patterns provides more useful information for model performance, resulting in a 1.47% improvement in accuracy on exclusive images.

	Exclusive mAP	Cooccur mAP	Unbiased mAP	All mAP
w/o selection	25.73 ± 0.04	65.18 ± 0.15	72.72 ± 0.06	56.16 ± 0.05
w/ selection	**27.20** ± 0.32	**65.56** ± 0.04	**73.09** ± 0.03	**56.37** ± 0.04

Table 3. **Effect of selecting useful generations.** Excluding unsuccessful manipulations of contextual patterns improves training.

Decomposing model accuracy and bias score In addition to mAP as an evaluation metric, Singh et al. [53] proposed a bias score computed for each class, defined as the relative ratio between classifier-predicted probabilities for images containing an object with and without its co-occurring context. Intuitively, a higher bias score indicates a larger performance drop when the object appears without its common context. We analyze the bias scores for the top biased pairs identified by Singh et al. [53] and find that our method improves the bias score for only 3 out of 14 classes, while the remaining classes show little change. Interestingly, the feature split method improves the bias scores for

8 out of 14 classes. We note, however, that feature split method often reduces overall accuracy, with a 0.29% drop in exclusive mAP and a 0.71% drop in cooccur mAP compared to the standard model, which can lead to improved bias scores. In contrast, our method improves both cooccur mAP by 2.24% and exclusive mAP by 3.68% relative to the standard model, resulting in unchanged bias scores in some cases (more details in the Supplementary Material).

In addition to the bias scores of object-context pairs targeted by our method, we also track the bias scores of the second most co-occurring context class. Specifically, for each object class, we consider the context class with the second-highest bias score, excluding the highest-scoring context already addressed by augmentation. Figure 5 compares these bias scores before and after our pipeline, i.e., between the standard classifier and DecoupleGen.

We found that these bias scores either improve or remain similar after applying DecoupleGen (lower bias score values indicate less bias), with improvements observed in 9 out of 13 classes. This demonstrates that DecoupleGen does not negatively affect, and can even improve, bias scores without explicitly targeting these co-occurring biases.

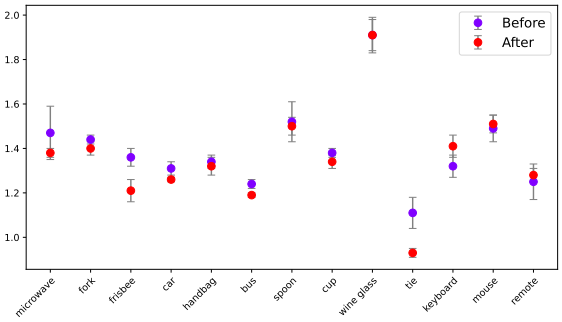


Figure 5. **Bias scores for biased object classes and their second-highest co-occurring context classes.** Even without explicitly targeting these contextual patterns, DecoupleGen maintains or slightly improves the corresponding bias scores.

Newly introduced common contextual patterns. Since our approach targets only the context label with the highest bias score for each object class, we further investigate whether including generated samples negatively affects other contextual patterns or introduces additional contextual bias. Figure 6 shows the conditional probability distribution of observing co-occurring contexts (x-axis) for each biased object category (y-axis) in both the original dataset and the synthesized samples.

We observe that many original co-occurrence patterns are reduced to substantially lower probabilities, for example the co-occurrence of `person` with `handbag`, `snowboard`, and `baseball glove`. However, newly introduced co-occurrence patterns also emerge, such as `dining table` with `wine glass` and `apple` with `dining table`, possibly due to implicit biases in pre-trained generative models. While we do not address these secondary biases in this work, they could be mitigated in future work through approaches such as tracking the class distribution of the augmented dataset during the image generation process or employing iterative generation strategies to synthesize additional images. Despite the presence of these newly introduced biases, DecoupleGen improves classifier performance compared to existing approaches.

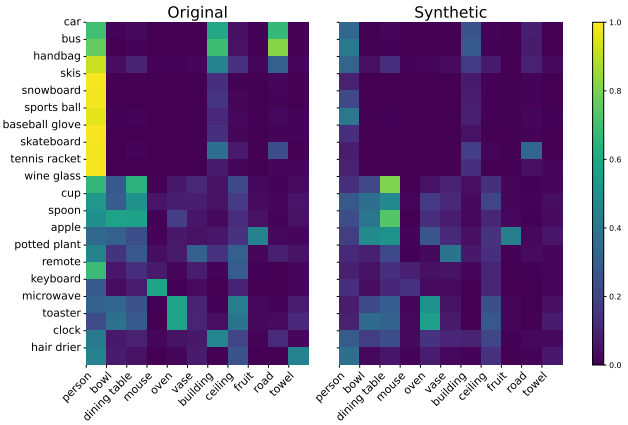


Figure 6. **Conditional distribution of biased object classes (y-axis) and cooccurring contexts (x-axis).** Common contextual patterns in original datasets are reduced, while generations also introduce new cooccurring patterns.

5. Discussions

In this work, we introduce **DecoupleGen**, a contextual debiasing pipeline that adapts diffusion model personalization to learn new word tokens for visual details and generate images without common contextual patterns. Our experiments show that incorporating generated samples from our method achieves competitive or superior performance on downstream image classification and object recognition tasks. Through qualitative analysis, we further demonstrate two key advantages of DecoupleGen over alternative generation approaches: it produces visually consistent and semantically meaningful images, and its generations better align with the downstream dataset distribution.

Limitations and future directions. DecoupleGen finetunes the diffusion model on each image to capture visual details, which can be time-consuming for large-scale datasets. A promising future direction is to improve the scalability of such approaches without degrading generation quality or downstream model performance. Finally, although our experimental results show that DecoupleGen significantly improves over previous work, there remains a substantial gap in classification accuracy between uncommon and common samples, motivating further investigation into how to better reduce this discrepancy.

Acknowledgment

This work is supported by NSF Career Award #2145198 to O.R.. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors' and do not necessarily reflect the views of the National Science Foundation. We would like to thank Princeton VisualAI Lab for discussions and feedback.

References

[1] Alex Andonian, Sabrina Osmany, Audrey Cui, YeonHwan Park, Ali Jahanian, Antonio Torralba, and David Bau. Paint by word. *arXiv preprint arXiv:2103.10951*, 2021. 2

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 4, 5

[3] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022. 2, 6

[4] Omri Avrahami, Kfir Aberman, Ohad Fried, Daniel Cohen-Or, and Dani Lischinski. Break-a-scene: Extracting multiple concepts from a single image. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–12, 2023. 3, 1

[5] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Transactions on Graphics (TOG)*, 42 (4):1–11, 2023. 5, 7, 3

[6] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*, 2023. 1, 2

[7] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025. 1

[8] Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10(9), 2009. 1, 2

[9] Abeba Birhane and Vinay Uday Prabhu. Large image datasets: A pyrrhic win for computer vision? In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1536–1546. IEEE, 2021. 1, 2

[10] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018. 1

[11] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018. 1, 2, 5

[12] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 1

[13] Myung Jin Choi, Antonio Torralba, and Alan S Willsky. Context models and out-of-context objects. *Pattern Recognition Letters*, 33(7):853–862, 2012. 2

[14] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019. 1, 2, 6

[15] Lisa Dunlap, Alyssa Umino, Han Zhang, Jiezhi Yang, Joseph E Gonzalez, and Trevor Darrell. Diversify your vision datasets with automatic diffusion-based augmentation. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2

[16] Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. Scaling laws of synthetic images for model training... for now. *arXiv preprint arXiv:2312.04567*, 2023. 2

[17] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, Francisco Herrera, Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, et al. Cost-sensitive learning. *Learning from imbalanced data sets*, pages 63–78, 2018. 1, 2

[18] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 2, 3

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[21] 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? *arXiv preprint arXiv:2210.07574*, 2022. 1, 2

[22] Yue He, Zheyan Shen, and Peng Cui. Towards non-iid image classification: A dataset and baselines. *Pattern Recognition*, 110:107383, 2021. 2, 4

[23] Reyhane Askari Hemmat, Mohammad Pezeshki, Florian Bordes, Michal Drozdzal, and Adriana Romero-Soriano. Feedback-guided data synthesis for imbalanced classification. *arXiv preprint arXiv:2310.00158*, 2023. 1, 2, 4, 5, 6, 7, 3

[24] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 1

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2

[26] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 1

[27] Ali Jahanian, Xavier Puig, Yonglong Tian, and Phillip Isola. Generative models as a data source for multiview representation learning. *arXiv preprint arXiv:2106.05258*, 2021. 2

[28] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2989–2998, 2023. 4

[29] Heinrich Jiang and Ofir Nachum. Identifying and correcting label bias in machine learning. In *International conference on artificial intelligence and statistics*, pages 702–712. PMLR, 2020. 2

[30] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019. 4, 5, 1

[31] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I 12*, pages 158–171. Springer, 2012. 2

[32] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023. 2, 3

[33] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 5

[34] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 292–305, 2023. 2

[35] Zhaoyang Li, Zhan Ling, Yuchen Zhou, Litian Gong, Erdem Bıyık, and Hao Su. Oric: Benchmarking object recognition under contextual incongruity in large vision-language models. *arXiv preprint arXiv:2509.15695*, 2025. 2

[36] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022. 1

[37] Hao Liu, Tom Zahavy, Volodymyr Mnih, and Satinder Singh. Palm up: Playing in the latent manifold for unsupervised pretraining. *Advances in Neural Information Processing Systems*, 35:35880–35893, 2022. 2

[38] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022. 2

[39] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 6

[40] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 2

[41] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 1

[42] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 2

[43] Viraj Prabhu, Sriram Yenamandra, Prithvijit Chattopadhyay, and Judy Hoffman. Lance: Stress-testing visual models by generating language-guided counterfactual images. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[44] Vikram V Ramaswamy, Sunnie SY Kim, and Olga Russakovsky. Fair attribute classification through latent space de-biasing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9301–9310, 2021. 2

[45] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021. 1

[46] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020. 4, 5, 1

[47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2, 4, 5, 6

[48] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 2, 3

[49] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 4

[50] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019. 4, 5, 1

[51] Shreya Shankar, Yoni Halpern, Eric Breck, James Atwood, Jimbo Wilson, and D Sculley. No classification without representation: Assessing geodiversity issues in open data sets

for the developing world. *arXiv preprint arXiv:1711.08536*, 2017. 1

[52] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025. 1

[53] Krishna Kumar Singh, Dhruv Mahajan, Kristen Grauman, Yong Jae Lee, Matt Feiszli, and Deepti Ghadiyaram. Don't judge an object by its context: Learning to overcome contextual bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11070–11078, 2020. 1, 2, 5, 6, 7, 3

[54] Yonglong Tian, Lijie Fan, Kaifeng Chen, Dina Katabi, Dilip Krishnan, and Phillip Isola. Learning vision from models rivals learning vision from data. *arXiv preprint arXiv:2312.17742*, 2023. 2

[55] Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. Stablerep: Synthetic images from text-to-image models make strong visual representation learners. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[56] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. A deeper look at dataset bias. *Domain adaptation in computer vision applications*, pages 37–55, 2017. 2

[57] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*, 2023. 2

[58] Emiel Van Miltenburg. Stereotyping and bias in the flickr30k dataset. *arXiv preprint arXiv:1605.06083*, 2016. 1

[59] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019. 1

[60] Yuzhe Yang, Haoran Zhang, Dina Katabi, and Marzyeh Ghassemi. Change is hard: A closer look at subpopulation shift. *arXiv preprint arXiv:2302.12254*, 2023. 4, 1

[61] Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. In *International Conference on Machine Learning*, pages 25407–25437. PMLR, 2022. 4, 5, 1

[62] Chenshuang Zhang, Fei Pan, Junmo Kim, In So Kweon, and Chengzhi Mao. Imagenet-d: Benchmarking neural network robustness on diffusion synthetic object. *arXiv preprint arXiv:2403.18775*, 2024. 2

[63] Hongyi Zhang. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 4, 5, 1

[64] Dora Zhao, Angelina Wang, and Olga Russakovsky. Understanding and evaluating racial biases in image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14830–14840, 2021. 1

[65] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5, 1

Personalized Generative Models for Contextual Debiasing

Supplementary Material

6. Experimental Details

6.1. Descriptions of Previous Approaches

We provide more detailed descriptions of existing approaches that we compared against for each downstream task below:

Image Classification We compare DecoupleGen to a set of approaches mentioned in [23, 60]. **GroupDRO** [50] minimizes the worst-case group loss by reweighting training examples based on their group representations. **Mixup** [63] performs data augmentation by linearly interpolating pairs of training samples and their labels, resulting in smoothier decision boundaries. **LISA** [61] learns invariant predictors through strategically augmenting training data with interpolations between samples. **Balanced Softmax** [46] adjusts standard cross entropy loss by reweighting logits inversely proportional to class frequencies in training dataset. **Classifier Retraining** [30] decouples representation and classifier learning, which finetunes classification layer using class balanced sampling with a fixed visual feature.

Object Recognition We compare DecoupleGen to set of approaches proposed and mentioned in [53]. **Class balancing loss** [14] treats scenarios where biased categories occur exclusively as long-tail classes. **Negative penalty** assigns high penalty when model predicts the context category in cases where a biased category occurs exclusively. **Weighted loss** applies 10 times higher weights to loss values when biased categories occur exclusively. **Remove co-occurring labels** modifies true labels of co-occurring images by removing context categories c for each b in training datasets. **Remove co-occurring images** modifies training datasets by removing all instances where biased categories co-occur with its context classes. **CAM** modifies training objective by using class activation map [65] as location supervision. **Feature split** separates feature space of classifier by enforcing them to attend to objects and their context.

6.2. Implementation Details

Image Generation We use publicly available Stable Diffusion v2 in Huggingface [59], with default hyperparameters as the base generative model. In our image generation step, we finetune text encoder and diffusion model unet with learning rate of word tokens 10^{-4} and learning rate of diffusion models 10^{-6}. We follow implementation details and experimental configurations in Avrahami et al. [4].

NICO dataset We finetune Stable Diffusion v2 for 200 iterations in both finetuning stages. We search for these hyperparameters across $\{10^{-4}, 5 \times 10^{-4}\}$ and $\{10^{-6}, 5 \times 10^{-6}, 10^{-5}\}$ for both learning rate and $\{200, 300, 400\}$ for training iterations. We adopt LoRA [26] as parameter-efficient fine-tuning technique.

For image classification model, following previous work [23, 60], we search over 10 hyperparameter configurations and select the best based on validation performance. We train classifiers for 50 epochs and report test performance.

COCO dataset We finetune Stable Diffusion v2 for 300 iterations in both finetuning stages. We search for best hyperparameters among the same set as in NICO dataset.

For multi-lable classification model, following previous work [53], we train classifiers with batch size of 200, learning rate 0.1, and cosine scheduler for a total of 100 epochs. We search across different values of batch size $\{128, 200, 256\}$, learning rate schedules at [60] or [50, 75] epochs, and pick final values based on validation performance. In both cases, we utilizes standard cross entropy loss without modifying supervised learning objectives.

PASCAL dataset The PASCAL dataset [40] contains 59 classes, including 20 object classes. Following prior work [53], we identify 10 biased classes and their most cooccurring context categories based on bias scores (Table 4). Since the dataset does not provide a semantic hierarchy of labels, following COCO we manually construct sets of semantically similar classes for replacement (Table 5). We use same set of hyperparameters for model training in both image generation stage and classification stage.

Biased Class	Cooccur Class	Bias Score
car	road	1.18 ± 0.01
aeroplane	sky	1.30 ± 0.06
tvmonitor	wall	1.28 ± 0.08
cow	grass	1.35 ± 0.05
table	wall	1.40 ± 0.04
bus	building	1.64 ± 0.06
chair	floor	1.77 ± 0.06
train	ground	1.85 ± 0.07
sofa	wall	2.02 ± 0.08
boat	water	35.39 ± 4.41

Table 4. Top 10 biased and cooccurring pairs among 20 object classes identified on PASCAL dataset [40].

road	snow, grass, water, ground, track
sky	building, ceiling, wall (indoor)
wall	ceiling, curtain, door, fence, window
grass	sidewalk, track, ground, road, snow
building	bench, chair, fence, sign
floor	ground, platform, road, sidewalk, track
ground	grass, platform, road, sidewalk, track, water
water	grass, platform, road, rock, wood, track

Table 5. List of original cooccurring classes (left) and their semantically similar classes used for replacement (right).

7. Additional Experimental Results

7.1. Image classification

Decompose group-wise model accuracy on NICO dataset. In NICO dataset, in addition to overall accuracy and worst group accuracy, we take a closer look at classifier accuracies on object-context combinations, after removing groups that contain few data points in validation split (less than 25 samples in our experiments), since model evaluations may not generalize well for these groups. As shown in Figure 7, we observe similar performance between different generation approaches, all of which improves from standard classifier. Overall DecoupleGen achieves similar performance compared to other approaches.

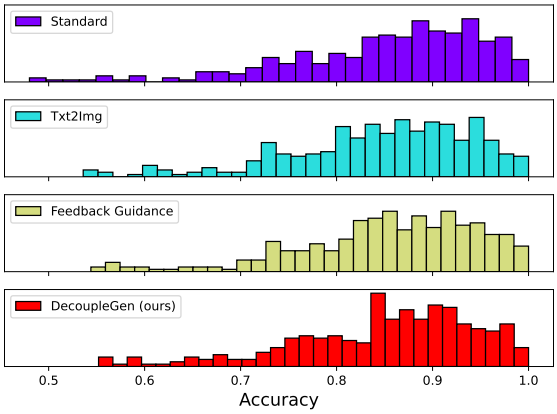


Figure 7. **Histogram of group-wise accuracies on NICO dataset.** All generation approaches achieve similar performance, all of which improve from standard classifier.

7.2. Object Recognition

Decompose class-wise model accuracy and bias score on COCO dataset. In COCO dataset, we further look at performance of accuracy metric and bias score metric on each class with initial bias score at least 1.5 on average from standard classifier, after removing classes that contain few data points in validation split (less than 30 samples in our experiments). These classes mostly overlap with top biased classes identified in [53]. We report results in Table 7 and Table 8 respectively.

As shown in Table 7, DecoupleGen shows significant improvements in 13 out of 14 classes on both exclusive and cooccur mAP, while feature split only improves in 4 classes on exclusive mAP and results in performance drop in 4 classes on both exclusive and cooccur mAP. However, as shown in Table 8, feature split shows significant improvements in 8 out of 14 classes in bias scores, while our method only improves 3 classes. This demonstrates that advantages of DecoupleGen mainly comes in better recognition performance in terms of model accuracy instead of bias score.

In particular, bias score metric that measures relative accuracy drop between cooccur and exclusive images may not capture complete information about classifier performance. While feature split results in better bias score, it penalizes model accuracy on cooccur, i.e. overrepresented, images more than exclusive images, resulting in no improvement in its accuracy from standard classifier. Instead, DecoupleGen improves 3.68% in exclusive mAP and 2.24% in cooccur mAP compared to standard classifier, with more significant benefits on exclusive, i.e. underrepresented, images.

Additionally, our results in Table 7 show that DecoupleGen achieves much better performance in exclusive and cooccur mAP compared to Feedback Guidance approach [23] based on Txt2Img models, which only improves 2 out of 14 classes in exclusive mAP and 4 classes in cooccur mAP. As discussed in the main paper, such discrepancy is possibly due to that Txt2Img models fail to generate samples that maintain similar to original datasets compared to our approach, therefore providing limited information helpful to downstream tasks.

Quantitative Comparisons on PASCAL dataset [40]. We report experimental results in Table 6. Using the same number of synthetic samples (2k), DecoupleGen improves over other generation-based baselines by 3.01%, increasing exclusive mAP from 58.48% to 61.49%. By further including more generations (8k), DecoupleGen improves over the best reweighting-based baseline by 1.26%, increasing exclusive mAP from 63.00% to 64.26%. DecoupleGen outperforms existing approches while maintaining comparable cooccur mAP. Together with results on COCO dataset, this shows that DecoupleGen is consistently beneficial on common real world scene datasets.

Method	Num Gens	Exclusive mAP	Cooccur mAP
Standard	0	57.56 ± 0.68	86.65 ± 0.08
Real Cooccur	0	57.30 ± 0.58	85.66 ± 0.48
Class balancing loss	0	60.08 ± 0.44	86.74 ± 0.11
Negative penalty	0	57.56 ± 0.55	86.77 ± 0.14
Weighted loss	0	**63.00** ± 0.39	86.23 ± 0.11
Remove cooccur labels	0	57.68 ± 0.32	**86.84** ± 0.14
Remove cooccur images	0	61.47 ± 0.41	86.41 ± 0.16
CAM	0	60.55 ± 0.27	86.64 ± 0.15
Feature split	0	60.08 ± 0.27	86.68 ± 0.16
Txt2Img Cooccur	$2k$	58.14 ± 2.05	85.11 ± 0.34
Txt2Img Exclusive	$2k$	57.78 ± 0.71	85.32 ± 0.84
Txt2Img Target Class	$2k$	57.33 ± 0.96	84.90 ± 1.44
SD Inpainting	$2k$	57.69 ± 1.87	85.36 ± 0.43
Blended LD	$2k$	57.51 ± 1.54	84.78 ± 0.38
Feedback Guidance	$2k$	58.48 ± 1.33	85.59 ± 0.44
DecoupleGen (ours)	$2k$	**61.49** ± 0.90	**85.60** ± 0.47
DecoupleGen (ours)	$8k$	**64.26** ± 0.75	85.08 ± 0.32

Table 6. **Quantitative comparisons on PASCAL dataset.** DecoupleGen achieves better performance in exclusive mAP and maintain similar performance in cooccur mAP compared to existing approaches.

Bias class	Cooccur class	Feature split [53]		Feedback guidance [23]		DecoupleGen (ours)	
		Exclusive mAP	Cooccur mAP	Exclusive mAP	Cooccur mAP	Exclusive mAP	Cooccur mAP
microwave	cabinet	34.53 ± 1.50 (+3.58)	58.30 ± 0.73 (-0.95)	30.02 ± 0.38 (-0.93)	62.75 ± 1.43 (+3.51)	**37.66** ± 1.32 (+6.71)	**68.35** ± 0.16 (+9.11)
potted plant	wall	21.51 ± 0.97 (+2.00)	46.95 ± 0.76 (-4.77)	20.18 ± 0.17 (+0.67)	54.13 ± 0.58 (+2.41)	**24.01** ± 0.24 (+4.50)	**57.53** ± 0.35 (+5.81)
fork	dining table	12.00 ± 0.46 (-0.61)	46.07 ± 0.24 (+0.67)	11.91 ± 0.39 (-0.70)	45.19 ± 0.58 (-0.21)	**16.05** ± 0.47 (+3.44)	**49.29** ± 0.52 (+3.89)
frisbee	grass	50.28 ± 1.16 (-3.22)	84.64 ± 0.20 (-0.45)	52.03 ± 0.20 (-1.47)	85.61 ± 0.38 (+0.52)	**60.06** ± 0.36 (+6.56)	**86.51** ±0.23 (+1.42)
car	road	38.17 ± 0.30 (+1.83)	78.49 ± 0.10 (-0.67)	36.06 ± 0.26 (-0.28)	79.35 ± 0.18 (+0.20)	**39.73** ±0.36 (+3.40)	**81.21** ± 0.25 (+2.06)
handbag	person	2.78 ± 0.67 (-0.03)	41.06 ± 0.23 (-0.18)	2.56 ± 0.18 (-0.25)	41.22 ± 0.18 (-0.02)	**3.87** ± 0.41 (+1.06)	**43.15** ± 0.45 (+1.91)
bus	road	44.72 ± 0.63 (+3.56)	83.95 ± 0.21 (-1.61)	40.93 ± 0.49 (-0.23)	85.18 ± 0.18 (-0.37)	44.88 ± 1.07 (+3.72)	**86.71** ± 0.16 (+1.15)
spoon	bowl	12.01 ± 0.50 (-1.99)	34.11 ± 0.46 (-1.34)	13.52 ± 0.36 (-0.48)	36.22 ± 1.34 (+0.77)	**16.03** ± 0.22 (+2.03)	**37.05** ± 1.20 (+1.60)
cup	dining table	25.41 ± 0.29 (-4.28)	63.06 ± 0.20 (+1.00)	29.71 ± 0.23 (+0.02)	62.90 ± 0.47 (+0.84)	**32.72** ± 0.08 (+3.03)	**65.22** ± 0.28 (+3.15)
wine glass	person	36.86 ± 0.14 (0.43)	55.94 ± 0.78 (-1.15)	36.34 ± 0.34 (-0.08)	56.63 ± 0.31 (-0.45)	**42.06** ± 0.78 (+5.64)	**59.18** ± 0.39 (+2.10)
tie	clothes	10.06 ± 0.41 (-0.09)	36.60 ± 0.38 (+0.20)	9.86 ± 0.86 (-0.29)	38.50 ± 0.41 (+2.10)	**11.29** ± 0.46 (+1.14)	39.35 ± 1.29 (+2.95)
keyboard	mouse	46.30 ± 0.31 (-0.43)	83.73 ± 0.54 (-0.67)	47.59 ± 0.43 (+0.85)	85.32 ± 0.19 (+0.92)	**49.90** ± 0.17 (+3.17)	**85.72** ± 0.15 (+1.32)
mouse	keyboard	24.88 ± 0.58 (0.54)	79.20 ± 0.18 (-0.25)	23.51 ± 0.26 (-0.83)	80.45 ± 0.20 (+0.99)	**31.10** ± 0.57 (+6.76)	**81.04** ± 0.50 (+1.59)
remote	person	19.36 ± 0.47 (-3.98)	70.44 ± 0.52 (-0.03)	22.39 ± 0.34 (-0.95)	71.18 ± 0.68 (+0.71)	**24.32** ± 0.76 (+0.98)	70.63 ± 0.73 (+0.17)
Mean		26.49 ± 0.29 (-0.29)	61.86 ± 0.12 (-0.71)	26.66 ± 0.18 (-0.13)	63.22 ± 0.30 (+0.65)	**30.46** ± 0.11 (+3.68)	**64.81** ± 0.26 (+2.24)

Table 7. **Comparisons of model accuracy on each class.** We report both exclusive and cooccur mAP for each class, and change in mAP values compared to standard classifier trained on original dataset, where blue indicates improvement, red indicates drop, and gray indicates similar performance within error bars. Bolded values indicate significantly better performance for each class among different approaches. DecoupleGen achieves better performance and more significant improvements in model accuracy in most cases.

Bias class	Cooccur class	Feature split [53]	DecoupleGen (ours)
microwave	cabinet	**1.36** ± 0.02 (-0.13)	1.60 ± 0.02 (+0.11)
potted plant	wall	**1.25** ± 0.03 (-0.34)	1.58 ± 0.04 (-0.01)
fork	dining table	1.75 ± 0.08 (+0.08)	**1.60** ± 0.06 (-0.06)
frisbee	grass	1.73 ± 0.06 (+0.03)	**1.64** ± 0.04 (-0.06)
car	road	**1.37** ± 0.04 (-0.38)	1.61 ± 0.02 (-0.14)
handbag	person	1.63 ± 0.13 (-0.13)	1.72 ± 0.12 (-0.05)
bus	road	**1.55** ± 0.04 (-0.21)	1.78 ± 0.03 (+0.02)
spoon	bowl	1.58 ± 0.08 (-0.28)	1.60 ± 0.03 (-0.27)
cup	dining table	**1.62** ± 0.11 (-0.28)	1.87 ± 0.05 (-0.02)
wine glass	person	1.75 ± 0.02 (-0.15)	**1.63** ± 0.01 (-0.27)
tie	clothes	1.91 ± 0.09 (+0.00)	1.91 ± 0.05 (-0.01)
keyboard	mouse	**1.72** ± 0.03 (-0.24)	1.92 ± 0.05 (-0.04)
mouse	keyboard	2.04 ± 0.09 (+0.05)	**1.87** ± 0.07 (-0.13)
remote	person	2.13 ± 0.13 (+0.10)	1.86 ± 0.12 (-0.17)

Table 8. **Bias score of each class and its cooccurring context.** We report bias scores for each class, and changes in bias score values compared to standard classifier trained on original dataset, where blue indicates improvements, red indicates drop, and gray indicates similar performance within error bars. Bolded values indicate significantly better performance for each class among different approaches. Feature split shows more improvements in bias scores compared to our method.

7.3. Additional Qualitative Results

Additional qualitative examples We provide additional qualitative comparisons in Figure 11. Compared to local image editing models (Figure 11b), i.e. Blended Latent Diffusion [5], DecoupleGen can handle complex relationships between objects and scenes, for example when objects interact with each other or when transforming images into a different contextual background (Figure 11d). Compared to Txt2Img models with Feedback Guidance [23] (Figure 11c), our method is able to preserve similar visual information as in original samples (Figure 11d).

Additional UMAP visualizations To demonstrate that generated samples from DecoupleGen are able to better adapt to downstream datasets than common Txt2Img models, we provide UMAP visualizations for more classes in Figure 12. We extract visual features from ResNet50 pretrained on ImageNet and use them as input to UMAP projections. We observe that real exclusive and cooccur images are more separable from each other for classes `handbag` and `remote`, while for classes `keyboard` and `spoon` there are not significantly distinct clusters. In both cases our method is able to generate samples more similar to original datasets in their UMAP projections, which demonstrates effectiveness of our adapted finetuning process.

How good is the capability of preserving visual details? Our approach preserves visual details by learning a new special token `[Vbackground]` and use this word as part of text input at inference time. Hence, the background reconstruction ability could indicate whether generations would stay similar to original images. Figure 8 shows generations from fine-tuned models with text input "A photo of `[Vbackground]`". This further justifies that our adapted personalized finetuning process is able to preserve visual details from original datasets.

Figure 8. **Background reconstruction capability.** We provide background mask (orange) at training time and generate from text input "A photo of `[Vbackground]`" at inference time. Our fine-tuned token can capture most visual details from original images.

Failure cases and limitations The personalization finetuning approach in every image is not perfect, as shown in

Figure 9. Specifically, it is not very good at scenarios where objects are very small, where such thin masks usually do not have enough learning signals, and our new word tokens fail to learn semantic information (Figure 9a). For example, prompted fine-tuned models to generate "A photo of `[Vremote]`", the new word token fails to learn semantic information corresponding to the remote object (Figure 9b). Therefore, it always fails to include the remote object in generations. Additionally, when the original image contains very complex visual details, for example a cooccur image of handbag and person containing cake and dining table (Figure 9c), generated samples usually cannot fully replicate all of contents. For example, such generations may fail to reconstruct the table and cake while removing person from the image (Figure 9d).

(a) Original (b) `[Vremote]` (c) Original (d) Miss details

Figure 9. **Examples of failure cases.** Limitations of our approach include when mask regions of object class, e.g. remote, are very small, and when original image, e.g. handbag and person, contains very complex visual information, generations fail to replicate some details, such as cake and dining table.

7.4. Further Analysis of DecoupleGen

Impact on contextual patterns. As shown in the main paper, while DecoupleGen is able to reduce contextual patterns that are targeted to improve, it may introduce new common cooccurring object-context pairs even though not specified. In our experiments, such examples include `wine glass - dining table` and `spoon - cup`. Figure 10 shows some of these scenarios.

In some cases, such cooccurring patterns can come from original datasets, for example dining table exists as well in the cooccur image of wine glass and person (Figure 10a). Generated samples from DecoupleGen remove person successfully but leave dining table preserved (Figure 10b), further strengthening additional biases in original datasets. In other cases, such cooccurring patterns can be implicitly introduced by Txt2Img models, for example when handling a pair of cup and dining table (Figure 10c), the generative model automatically includes a spoon together with the cup (Figure 10d), even though spoons do not exist in original image or is not specified as part of text input to generative model. Therefore, although our experiments demonstrate improvements in target biased cooccurring pairs, we cannot completely avoid introducing new contextual patterns.

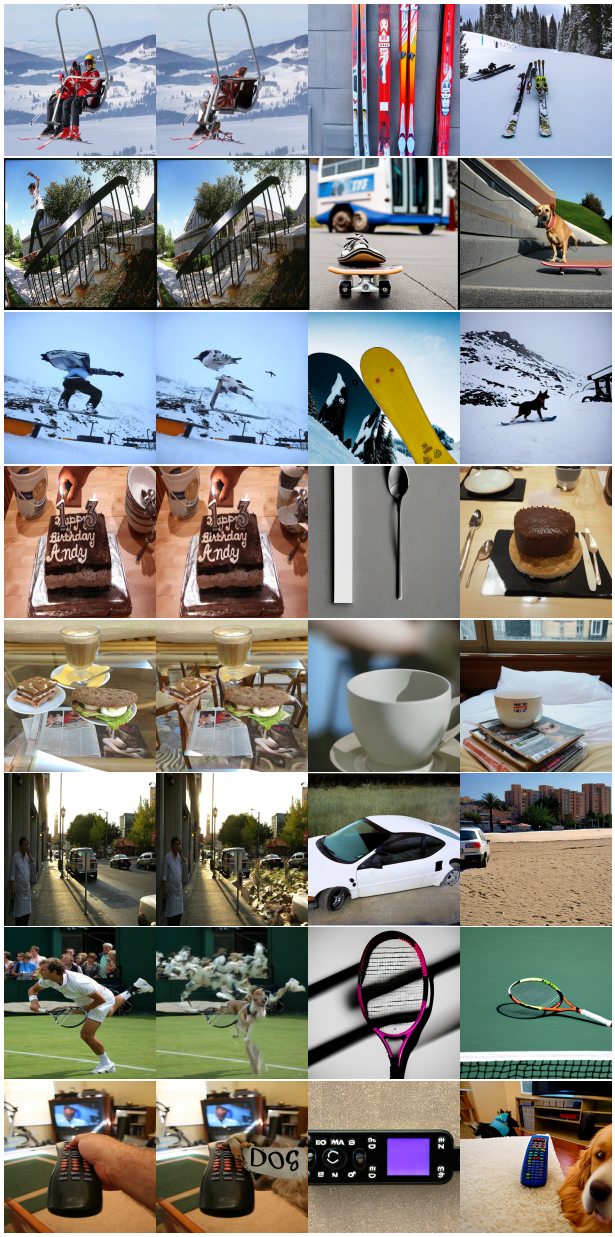
(a) Wine glass and Person (b) Person removed (c) Cup and Dining table (d) Spoon included

Figure 10. **Examples of newly introduced common contextual patterns.** We observe that such cooccurring patterns can either come from original images and become strengthened, or are implicitly introduced by generative models.

Model accuracy on unbiased classes also improves. We report AP values on each unbiased classes from models trained in our approach in Table 9. Our results show that model accuracy on unbiased classes improves from standard classifier by 1.3% increase in mAP as well. This is a smaller improvements compared to biased classes, where we have 3.68% and 2.24% improvement for exclusive and cooccur images respectively. Among 56 unbiased classes, after ignoring those containing very few data points, 1 class decreases, 15 classes stayed within error bar, and 40 classes significantly improved.

Unbiased class	AP value	Unbiased class	AP value
person	98.04 ± 0.00 (+0.31)	knife	46.92 ± 0.48 (+1.59)
bicycle	62.52 ± 0.51 (+3.58)	bowl	61.81 ± 0.40 (+1.99)
motorcycle	87.41 ± 0.19 (+0.84)	banana	73.26 ± 0.11 (+0.58)
airplane	95.09 ± 0.22 (+0.36)	apple	53.34 ± 0.28 (+1.28)
train	95.06 ± 0.10 (+0.78)	sandwich	61.59 ± 0.29 (+1.43)
truck	66.33 ± 0.20 (+3.29)	orange	64.61 ± 0.33 (+0.77)
boat	83.94 ± 0.29 (+2.09)	broccoli	80.32 ± 0.34 (+0.03)
traffic light	74.90 ± 0.17 (+1.90)	carrot	60.38 ± 0.36 (+2.15)
fire hydrant	72.84 ± 0.67 (+0.16)	hot dog	60.71 ± 0.69 (-0.72)
stop sign	67.85 ± 0.43 (-0.26)	pizza	84.60 ± 0.22 (+0.31)
bench	59.70 ± 0.44 (+3.64)	donut	70.79 ± 0.13 (+0.78)
bird	70.55 ± 0.30 (+1.76)	cake	71.25 ± 0.30 (+0.90)
cat	88.62 ± 0.09 (+0.50)	chair	71.10 ± 0.08 (+3.30)
dog	77.16 ± 0.32 (+1.24)	couch	77.12 ± 0.23 (+3.58)
horse	88.18 ± 0.31 (+0.05)	bed	83.49 ± 0.14 (+2.18)
sheep	90.75 ± 0.16 (+0.65)	dining table	75.15 ± 0.14 (+1.65)
cow	83.35 ± 0.46 (-0.04)	toilet	93.58 ± 0.10 (+0.39)
elephant	96.82 ± 0.12 (-0.20)	tv	80.43 ± 0.21 (+2.07)
bear	94.52 ± 0.42 (+0.23)	laptop	81.45 ± 0.32 (+1.23)
zebra	97.74 ± 0.15 (-0.05)	cell phone	42.47 ± 0.28 (+1.62)
giraffe	98.57 ± 0.07 (+0.04)	oven	80.42 ± 0.67 (+3.85)
backpack	38.29 ± 0.62 (+2.89)	sink	85.79 ± 0.33 (+1.90)
umbrella	77.06 ± 0.11 (+1.66)	refrigerator	70.36 ± 0.27 (+1.82)
suitcase	62.87 ± 0.63 (+2.57)	book	57.69 ± 0.18 (+2.52)
kite	92.27 ± 0.05 (+0.67)	clock	75.02 ± 0.21 (+1.27)
baseball bat	88.22 ± 0.42 (+0.23)	vase	71.48 ± 0.43 (+2.33)
surfboard	93.21 ± 0.03 (+0.64)	scissors	37.59 ± 1.02 (-2.82)
bottle	58.81 ± 0.14 (+3.51)	teddy bear	76.43 ± 0.70 (+0.61)

Table 9. **Model accuracy on each unbiased class.** We report AP values on each unbiased class from DecoupleGen, and change in AP values compared to standard classifier trained on original dataset, where blue indicates improvement, red indicates drop, and gray indicates similar performance within error bars. Model performance on most classes improve from standard classifier, even though these classes are not explicitly addressed in our approach.



(a) Original (b) Blended LD (c) Txt2Img FG (d) DecoupleGen

Figure 11. **More examples of qualitative comparisons.** Local image editing models, e.g. Blended LD [5], often fail to keep generate reasonable and coherent images of complex scenes. Feedback-guided generations [23] fail to preserve similar visual appearance in natural images. Instead, DecoupleGen can handle complex relationship between objects and scene, while capable of maintaining similar visual details from original images.

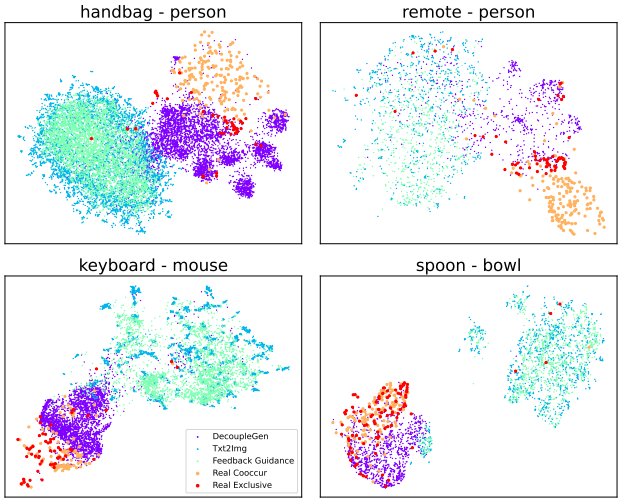


Figure 12. **UMAP visualizations of real samples and generations.** We plot real exclusive and cooccur images from validation split, together with synthetic samples from different generation approaches. Generations from DecoupleGen overall stay closer to real data distribution than those from other methods, thus providing information more relevant to downstream tasks.