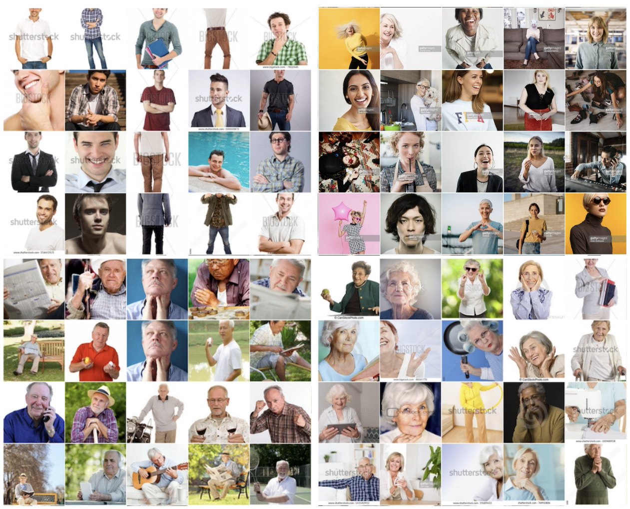
FairDeDup: Detecting and Mitigating Vision-Language Fairness Disparities in Semantic Dataset Deduplication

Supplementary Material

1. Bias Constrained Clusters

As described in Sec. 6, clustering may limit the availability of lower represented samples for biasing sensitive concept representation. We show several demonstrative clusters in Fig. 1 alongside descriptions of possible limitations.



(a) Stock close-ups bifurcated across gender presentation and age.

Figure 1. Sample Clusters where selecting for certain underrepresented concepts may be difficult due to them being split into an entirely different cluster.

2. Hyperparameters

Parameter	Value
Model	CLIP
Image Encoder	Vision Transformer Base/16
Text Encoder	Text Transformer
Epochs	16
Batch Size	33,820
Learning Rate	5×10^{-4}
LR Warmup	Linear / 2,000 Batches
LR Schedule	Cosine Annealing
Optimizer	AdamW
$\quad Decay$	0.2
$\quad \beta_1$	0.90
$\quad \beta_2$	0.98
$\quad \epsilon$	1×10^{-6}
Precision	AMP BFloat16

Table 1. Hyperparameters used to train all models.

3. Choosing Sensitive Concepts

We derive sensitive concepts based on commonly protected groups in law. For example, Title VII of the US Civil Rights Act prohibits employment discrimination based on groups like *race* and *religion*. We take the intersection of commonly protected groups and those which are annotated in VL fairness datasets (*e.g.*, only *race* from above) to inform our concept design. In this way, we are able to consider concepts relevant to real-world practice while retaining our ability to evaluate the effectiveness of our mitigation strategy in a lab setting. To represent intersectional identities, we specifically choose sensitive concepts which simultaneously capture many of these protected groups.

4. Sensitive Concept Prototypes

```
person
woman
man
black person
black woman
black man
white person
white woman
white man
indian person
indian woman
indian man
latino person
latino woman
latino man
east asian person
east asian woman
east asian man
middle eastern person
middle eastern woman
middle eastern man
southeast asian person
southeast asian woman
southeast asian man
old person
old woman
old man
old black person
old black woman
old black man
old white person
old white woman
old white man
old indian person
old indian woman
old indian man
old latino person
old latino woman
old latino man
old east asian person
old east asian woman
old east asian man
old middle eastern person
old middle eastern woman
old middle eastern man
old southeast asian person
old southeast asian woman
old southeast asian man
young person
young woman
young man
young black person
young black woman
young black man
young white person
young white woman
young white man
young indian person
young indian woman
young indian man
young latino person
young latino woman
young latino man
young east asian person
young east asian woman
young east asian man
young middle eastern person
young middle eastern woman
young middle eastern man
young southeast asian person
young southeast asian woman
young southeast asian man
child
black child
white child
indian child
latino child
east asian child
middle eastern child
southeast asian child
baby
black baby
white baby
indian baby
latino baby
east asian baby
middle eastern baby
southeast asian baby
boy
girl
black boy
black girl
white boy
white girl
indian boy
indian girl
latino boy
latino girl
east asian boy
east asian girl
middle eastern boy
middle eastern girl
southeast asian boy
southeast asian girl
person with dark skin
person with light skin
old person with dark skin
old person with light skin
young person with dark skin
young person with light skin
```

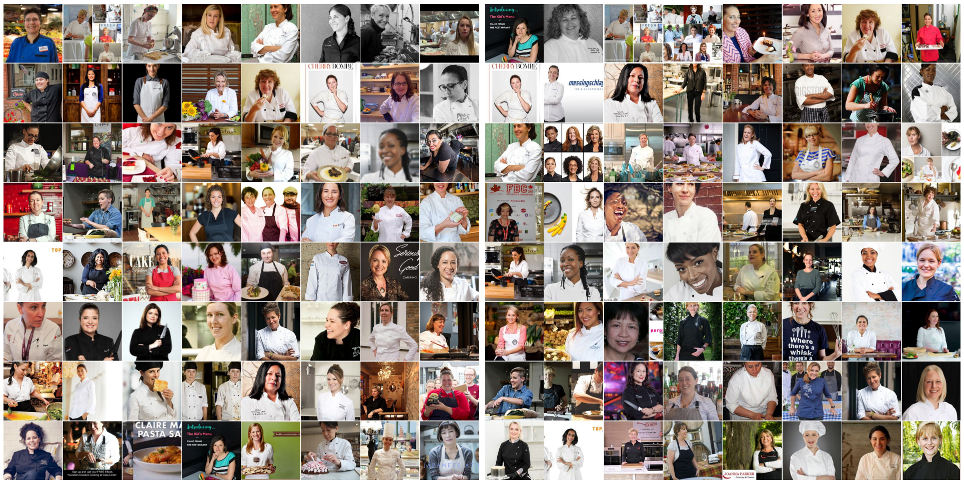
(a) Sensitive Concepts

```
A photo of a {concept}
This is a photo of a {concept}
A {concept}
```

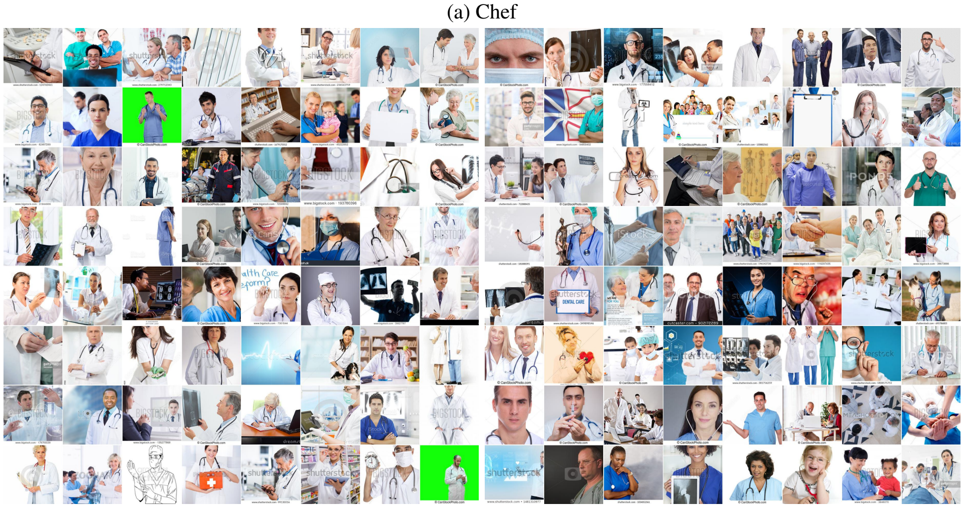
(b) Text Templates

Figure 2. Sensitive concepts and templates used to generate text concept prototypes for FairDeDup in Sec 3.

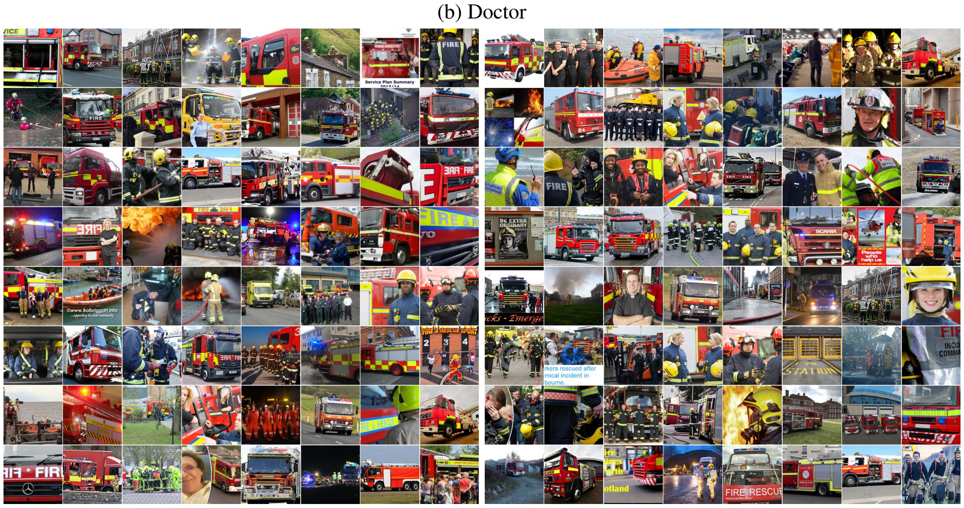
5. Deduplicated Cluster Examples



(a) Chef



(b) Doctor



(c) Firefighter

Figure 3. Additional samples from person-related clusters after applying **SemDeDup (left)** and **FairDeDup (right)**. Samples are randomly selected from clusters manually identified to include people. We annotate each cluster with our identification of its semantic contents.

6. Additional Datasets and Metrics

		Full Data (100%)	SemDeDup (50%)	FairDeDup (50%)
-a	Acc@1	.2947	.3096	.2988
	Acc@5	.6247	.6359	.6271
	MPCR	.3163	.3228	.3126
-o	Acc@1	.5060	.5365	.5360
	Acc@5	.8235	.8420	.8385
	MPCR	.5213	.5501	.5443
-r	Acc@1	.7646	.7652	.7534
	Acc@5	.9249	.9245	.9223
	MPCR	.7512	.7506	.7386
1k	Acc@1	.6640	.6579	.6535
	Acc@5	.8993	.8974	.8970
	MPCR	.6639	.6578	.6536
sketch	Acc@1	.5142	.5036	.4989
	Acc@5	.7803	.7774	.7724
	MPCR	.5144	.5040	.4993
v2	Acc@1	.5821	.5776	.5747
	Acc@5	.8453	.8413	.8368
	MPCR	.5822	.5774	.5747

Table 2. Zero-shot classification performance on ImageNet variants from CLIP Benchmark. Mean Per Class Recall abbreviated as MPCR. Higher is better for all metrics.

		Full Data (100%)	SemDeDup (50%)	FairDeDup (50%)
flickr30k	Image R@5	.8728	.8736	.8714
	Text R@5	.9640	.9670	.9620
flickr8k	Image R@5	.8570	.8592	.8574
	Text R@5	.9450	.9400	.9300
coco	Image R@5	.6325	.6318	.6255
	Text R@5	.7852	.7882	.7810

Table 3. Image-text (Image R@5) and text-image (Text R@5) retrieval Recall@5 on Flickr and COCO from CLIP Benchmark. Higher is better for all metrics.

		Full Data (100%)	SemDeDup (50%)	FairDeDup (50%)
cars	Acc@1	.8526	.8346	.8429
	Acc@5	.9923	.9909	.9922
	MPCR	.8541	.8344	.8429
country 211	Acc@1	.1791	.1742	.1685
	Acc@5	.3978	.3950	.3889
	MPCR	.1789	.1740	.1683
fer 2013	Acc@1	.3976	.3969	.4809
	Acc@5	.9358	.9115	.9413
	MPCR	.4057	.3702	.4152
fgvc aircraft	Acc@1	.1563	.1497	.1665
	Acc@5	.4164	.4146	.4212
	MPCR	.1545	.1495	.1667
gtsrb	Acc@1	.4159	.4101	.3383
	Acc@5	.7352	.6814	.7062
	MPCR	.3884	.3806	.3571
mnist mnist	Acc@1	.3896	.5474	.5890
	Acc@5	.8064	.8022	.9058
	MPCR	.3938	.5531	.5911
objectnet	Acc@1	.4722	.4754	.4781
	Acc@5	.7296	.7283	.7311
	MPCR	.4614	.4681	.4665
render sst2	Acc@1	.5371	.5157	.5041
	MPCR	.5368	.5155	.5036
stl10	Acc@1	.9659	.9705	.9686
	Acc@5	.9996	.9998	.9995
	MPCR	.9659	.9708	.9684
sun397	Acc@1	.6856	.6786	.6838
	Acc@5	.9344	.9351	.9363
	MPCR	.6730	.6613	.6662
voc2007	Acc@1	.7432	.7617	.7584
	Acc@5	.9498	.9518	.9515
	MPCR	.8197	.8297	.8286

Table 4. Additional zero-shot classification results from CLIP Benchmark. Mean Per Class Recall abbreivated as MPCR. Higher is better for all metrics.

		Full Data (100%)	SemDeDup (50%)	FairDeDup (50%)
caltech 101	Acc@1	.8304	.8230	.8309
	Acc@5	.9399	.9394	.9578
	MPCR	.9193	.9035	.9061
cifar 10	Acc@1	.9198	.9255	.9203
	Acc@5	.9986	.9984	.9992
	MPCR	.9198	.9254	.9204
cifar 100	Acc@1	.7234	.7291	.7299
	Acc@5	.9342	.9342	.9386
	MPCR	.7231	.7289	.7303
clevr obj dist	Acc@1	.2033	.2101	.2232
	Acc@5	.9187	.9187	.9187
	MPCR	.1686	.1633	.1673
clevr count all	Acc@1	.1421	.2317	.1916
	Acc@5	.6474	.8174	.6397
	MPCR	.1397	.2264	.1873
diabetic	Acc@1	.0646	.3209	.0666
	Acc@5	1.0000	1.0000	1.0000
	MPCR	.2178	.1998	.2029
dmlab	Acc@1	.1949	.1692	.1960
	Acc@5	.8430	.8236	.8270
	MPCR	.1677	.1869	.1569
dsprites label orient.	Acc@1	.0226	.0272	.0244
	Acc@5	.1259	.1264	.1302
	MPCR	.0231	.0276	.0249
dsprites label xpos	Acc@1	.0305	.0315	.0305
	Acc@5	.1600	.1568	.1625
	MPCR	.0313	.0321	.0312
dsprites label ypos	Acc@1	.0315	.0317	.0317
	Acc@5	.1553	.1559	.1596
	MPCR	.0311	.0312	.0312

Table 5. Zero-shot classification performance on Visual Task Adaptation Benchmark (VTAB) datasets from CLIP Benchmark. Mean Per Class Recall abbreviated as MPCR. Higher is better for all metrics.

		Full Data (100%)	SemDeDup (50%)	FairDeDup (50%)
dtd	Acc@1	.5330	.5021	.5074
	Acc@5	.8293	.7883	.8202
	MPCR	.5330	.5011	.5080
eurosat	Acc@1	.4904	.5220	.5246
	Acc@5	.9335	.9289	.8987
	MPCR	.5117	.5288	.5404
flowers	Acc@1	.6723	.6536	.6878
	Acc@5	.8533	.8442	.8374
	MPCR	.6657	.6384	.6509
kitti dist.	Acc@1	.1589	.1505	.2363
	MPCR	.2134	.1707	.2134
pcam	Acc@1	.5521	.4720	.5910
	MPCR	.5521	.4719	.5909
pets	Acc@1	.8929	.8577	.8812
	Acc@5	.9951	.9940	.9937
	MPCR	.8917	.8570	.8802
resisc 45	Acc@1	.5694	.5892	.5763
	Acc@5	.8889	.8938	.8914
	MPCR	.5751	.5955	.5853
smallnorb label azimuth	Acc@1	.0539	.0502	.0550
	Acc@5	.2796	.2781	.2738
	MPCR	.0547	.0509	.0560
smallnorb label elevation	Acc@1	.0927	.1125	.1114
	Acc@5	.5319	.5770	.5544
	MPCR	.0922	.1116	.1106
svhn	Acc@1	.3973	.3657	.4008
	Acc@5	.7782	.7694	.8107
	MPCR	.3727	.3670	.3238

Table 6. Additional results for VTAB extending Tab. 5.

7. Extended Pseudo-Code

```
def semdedup(embs, eps):
    # Sort by distance to the cluster centroid.
    sort_by_dist_to_centroid(embs, desc=True)

    # Compute the pairwise cosine similarity
    pair_sims = embs @ embs.T
    triu_sims = torch.triu(pair_sims, diagonal=1)
    M = torch.max(triu_sim_matrix, dim=0)[0]

    # Keep if the max similarity <= threshold
    points = M <= 1 - epsilon
    log_and_keep(points)

def fairdedup(embs, prototypes, eps):
    # Get similarity with concept prototypes
    proto = embs @ prototypes.T

    balance = AverageMeter(prototype.shape[0])
    tovisit = torch.ones(embs.shape[0])
    while tovist.any():
        # Find an unvisited neighborhood
        node = torch.where(tovisit)[0][0]
        sims = embs[node] @ embs.T
        neighbors = torch.where(sims > 1 - eps)
        neighbors = neighbors[0]

        # Maximize least represented concept
        c = balance.get_min_concept()
        point = proto[neigbors][:, c].argmax()
        balance.update(point)

        log_and_keep(point)
        tovisit[neighbors] = 0

# Input: embedding_model, dataset, eps, prototypes
# Embed and cluster the dataset
embeddings = embedding_model(dataset)
per_cluster_embeddings = kmeans(embeddings)
for cluster in per_cluster_embeddings:
    # Choose selection method
    semdedup(cluster)
    fairdedup(cluster)
```

Figure 4. Extended PyTorch-style pseudo-code for SemDeDup and FairDeDup selection given concept `prototypes`, an `embedding_model`, target `dataset` to deduplicate, and an `eps` similarity threshold for determining duplicates. SemDeDup pseudo-code modified from the original paper.