Appendix

A. Implementation Details of DIM

A.1. Preliminaries

In a standard image classification setting with subgroup information, inputs $\boldsymbol{x}$ in image space $\mathcal{X}$, labels y in class space $\mathcal{Y}$, and subgroups g in subgroup space $\mathcal{G}$ follows some certain distribution P. Specifically, training data, validation data, and test data are respectively drawn from distributions $\hat{P}_{train}$, $\hat{P}_{val}$, and $\hat{P}_{test}$. An image classifier trained on training data $(\boldsymbol{x}, y, g)$, where g is not available to the classifier, can be denoted as a function $\phi_\theta \colon \mathcal{X} \to \mathcal{Y}$ that predict the class of a given image. Assume there are L annotated classes in the dataset, making $\mathcal{Y} = \{1, 2, ..., L\}$. For each class, we hypothesize the inputs can be further divided into G subgroups. The subgroup is defined as a smaller, more specific category within a larger class, representing a partition based on certain characteristics or features, like the "dolphin" and "beaver" in the class "aquatic mammals".

In this work, for simplicity, we hypothesize that $G \in \mathbb{Z}^+$ is a constant value across classes. It's also supported by the fact that while data can be partitioned into subgroups in various ways through different standards, the partitioning that affects the model's output most is what we are most interested in. Due to under-representation or other difficulties, the image classifier exhibits low performance on certain subgroups within one class. We define bias subgroups as subgroups with lower classification accuracy than the median, which implies $k = \lfloor \frac{G}{2} \rfloor$ biased subgroups. The primary objective of this work is to discover multiple bias subgroups ($k \geq 2$), thereby implying $\lfloor \frac{G}{2} \rfloor \geq 2$ and then $G \geq 4$.

Either mean or median is reasonable in this context. We choose the median here for simplicity. If using mean, a constant number of subgroups G across classes does not guarantee a constant number of biased subgroups due to different cases of subgroup classification accuracy. For example, accuracies of $(0.1, 0.2, 0.8, 0.9)$ lead to 2 subgroups lower than the mean. However, accuracies like $(0.1, 0.7, 0.8, 0.9)$ results in only one bias subgroup.

A.2. Decomposition

In the decomposition stage of DIM, we apply the partial least squares (PLS) method to discover the embeddings of multiple unknown subgroups. Here, we give a detailed overview of the PLS.

PLS Details. We consider the input $\boldsymbol{x}$ and the response $\boldsymbol{z}$. PLS consists of the following steps iteratively repeated n times (for n components):

1. searching for paired directions that maximize covariance between the corresponding components in the input (observation) space and response (supervision) space.

$$\begin{aligned} &\max_{\boldsymbol{w}_i,\, \boldsymbol{h}_i} \mathbb{E}_{\hat{P}_l}(u_i v_i) = \max_{\boldsymbol{w}_i,\, \boldsymbol{h}_i} \mathbb{E}_{\hat{P}_l}\left((\boldsymbol{w}_i^T \widehat{\boldsymbol{x}}_i)(\boldsymbol{h}_i^T \boldsymbol{z}_{\boldsymbol{x},i})\right) \\ &\textit{s.t. } \|\boldsymbol{w}_i\| = 1,\ \|\boldsymbol{h}_i\| = 1 \end{aligned} \tag{4}$$

   where $\boldsymbol{w}_i \in \mathbb{R}^d$, $\boldsymbol{h}_i \in \mathbb{R}^M$ are the discovered directions and $u_i \coloneqq \boldsymbol{w}_i^T \widehat{\boldsymbol{x}}_i$, $v_i \coloneqq \boldsymbol{h}_i^T \boldsymbol{z}_{\boldsymbol{x},i}$ are the corresponding components (also called input score and response score). In the matrix notation, where the input $X \in \mathbb{R}^{N \times D}$ and the response $Z \in \mathbb{R}^{N \times M}$ are matrices (N is the number of samples), $\boldsymbol{w}_i$ and $\boldsymbol{h}_i$ are the first left and right singular vectors of the cross-covariance matrix $X^T Z$.
2. performing least squares regression on the scores for input and response.

$$\begin{cases} \widehat{\boldsymbol{x}}_i = u_i \boldsymbol{\alpha}_i + \boldsymbol{\epsilon}_i \\ \boldsymbol{z}_{\boldsymbol{x},i} = v_i \boldsymbol{\beta}_i + \boldsymbol{\eta}_i, \end{cases} \tag{5}$$

   where $\boldsymbol{\alpha}_i$, $\boldsymbol{\beta}_i$ are regression coefficients and $\boldsymbol{\epsilon}_i$, $\boldsymbol{\eta}_i$ are the remainders.
3. deflating the inputs and responses by subtracting the approximation modeled by the regression. $\widehat{\boldsymbol{x}}_{i+1} \coloneqq \widehat{\boldsymbol{x}}_i - u_i \boldsymbol{\alpha}_i = \boldsymbol{\epsilon}_i$ and $\boldsymbol{z}_{\boldsymbol{x},i+1} \coloneqq \boldsymbol{z}_{\boldsymbol{x},i} - v_i \boldsymbol{\beta}_i = \boldsymbol{\eta}_i$.

In our problem, we use the CLIP embedding of images as the $\boldsymbol{x}$ and the model supervision as $\boldsymbol{z}$. With such a design, we can derive the principal components of image features mostly aligned with the changes of the supervision, achieving the supervised decomposition.

A.3. Interpretation

To exploit existing coarse-grained knowledge, we employ text embeddings generated from class-specific text prompts, "a photo of {class}," as retrieval bases. For instance, within the "large man-made outdoor things" class, we first compute the text embedding from "a photo of large man-made outdoor things." We then add each discovered subgroup embedding to it and

proceed to retrieve images and corresponding metadata using the resultant normalized embeddings.

B. Bias Mitigation with Soft-label Strategy

In our paper, we have provided the model-centric strategy with Soft-DI in Sec. 4. The soft-label strategy can also be applied to the groupDRO method as follows,

Case study of model-centric strategy with soft gDRO. Consider the empirical distribution on the training data $\hat{P}$. For some assignment of weights $\boldsymbol{q} = (q_1, ..., q_N) \in \Delta_N$, where Δ_N is the $(m-1)$-dimensional probability simplex, the expected loss and the way to update $\boldsymbol{q}$ in original groupDRO [39] is:

$$\mathcal{L} = \mathbb{E}_{(\boldsymbol{x},y,g)\sim\hat{P}}\left(q_g \ell(\theta;(\boldsymbol{x},y))\right) \tag{6}$$

$$\boldsymbol{q}_n^{(t)} = \boldsymbol{q}_n^{(t-1)} \exp\left(\eta_q \mathbb{E}_{\hat{P}}(\ell(\theta;(\boldsymbol{x},y))|g=n)\right) \tag{7}$$

where $g \in \{1, 2, ..., N\}$ is the hard group label of data.

We define soft-label version groupDRO [39] expected loss and the way to update weights $\boldsymbol{q}$ as following:

$$\mathcal{L} = \mathbb{E}_{(\boldsymbol{x},y,\boldsymbol{g})\sim\hat{P}}\left(\boldsymbol{q}^T \boldsymbol{g} \ell(\theta;(\boldsymbol{x},y))\right) \tag{8}$$

$$\boldsymbol{q}_n^{(t)} = \boldsymbol{q}_n^{(t-1)} \exp\left(\eta_q \mathbb{E}_{\hat{P}}(g_n \ell(\theta;(\boldsymbol{x},y)))\right) \tag{9}$$

where $\boldsymbol{g} = (g_1, ..., g_N) \in \Delta_N$ is the soft label of data.

C. Experiment Details

C.1. The Selection of Supervision

In our experiments, we use three kinds of supervision: correctness, logit, and loss. All are in the form of training dynamics. For each image $\boldsymbol{x}$ and the corresponding label y, we define correctness as whether the image classifier ϕ_θ correctly outputs the label, $\mathbb{1}\{\phi_\theta(\boldsymbol{x}) = y\}$. The logit is the unnormalized (without soft-max activation) final score of the image classifier. For loss, we adopt the cross-entropy loss. Specifically, we train the image classifier from scratch and record the correctness, logit, and loss for each image in each epoch. Then, we concatenate all this information together to supervise the decomposition of image features. For training of t epochs, the final corresponding supervision z of each image is a vector of length $3t$.

The use of supervision may have a different impact on the decomposition of image features. The logit provides information on image features that the biased model learns. Correctness helps the decomposition align with the direction that affects the image classifier's performance. Loss is a fine-grained correctness.

C.2. Bias Mitigation

In our evaluation, we adopt two types of methods to mitigate the biased behavior of the model, namely the unsupervised and supervised methods. For the hyper-parameter settings, the details are presented as follows:

1. For JTT [26], by grid-search on the hyper-parameters, we set the number of epochs for first-time training T as 10, the up-sampling factor λ_{up} as $\frac{|\text{Training set}|}{|\text{Error set}|}$. We set the total number of training epochs as the same as the vanilla training.
2. For SubY [11], no hyper-parameter is required.
3. For LfF [31], we tune the hyper-parameter q by grid searching over $q \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$.
4. For EIIL [4], no hyperparameter is required.
5. For gDRO [39] , we set the group number as the number of subgroups in each class, *i.e.*, 5in CIFAR-100, 13 in Breeds, and set η as 0.1.
6. For DI, we set the group number as the number of subgroups in each class, *i.e.*, 5 in CIFAR-100, and 13 in Breeds.

Table 6. The classification accuracy (**including the error bar**) of ResNet-18 on the CIFAR-100 test set.

Type	Method	Worst subgroup accuracy		Acc.
		1st	2nd	
-	ERM	$24.8_{\pm 0.09}$	$33.6_{\pm 0.12}$	$44.4_{\pm 0.11}$
Model-centric (Unsupervised)	JTT [26]	$26.9_{\pm 0.24}$	$34.6_{\pm 0.30}$	$48.5_{\pm 0.25}$
	SubY [11]	$25.1_{\pm 0.16}$	$33.8_{\pm 0.16}$	$45.6_{\pm 0.13}$
	LfF [31]	$25.0_{\pm 0.22}$	$33.8_{\pm 0.18}$	$44.3_{\pm 0.15}$
	EIIL [4]	$25.9_{\pm 0.09}$	$34.8_{\pm 0.07}$	$47.2_{\pm 0.10}$
Model-centric (labeled by Jain *et al.* [12])	gDRO [39]	$26.7_{\pm 0.11}$	$34.5_{\pm 0.15}$	$46.9_{\pm 0.12}$
	DI [44]	$24.3_{\pm 0.06}$	$34.2_{\pm 0.14}$	$47.5_{\pm 0.10}$
Model-centric (labeled by Domino [7])	gDRO [39]	$25.9_{\pm 0.09}$	$34.8_{\pm 0.07}$	$47.2_{\pm 0.07}$
	DI [44]	$25.6_{\pm 0.18}$	$35.3_{\pm 0.22}$	$47.1_{\pm 0.19}$
Model-centric (labeled by **Ours**)	gDRO [39]	$27.2_{\pm 0.04}$	$35.8_{\pm 0.07}$	$48.3_{\pm 0.06}$
	Soft-gDRO	$\mathbf{27.2}_{\pm 0.08}$	$\mathbf{38.4}_{\pm 0.12}$	$\mathbf{49.8}_{\pm 0.09}$
	DI [44]	$26.5_{\pm 0.05}$	$36.7_{\pm 0.07}$	$48.7_{\pm 0.08}$
	Soft-DI	$26.8_{\pm 0.06}$	$37.1_{\pm 0.04}$	$48.9_{\pm 0.07}$
Data-centric	Jain *et al.* [12]	$33.7_{\pm 0.15}$	$41.9_{\pm 0.07}$	$53.1_{\pm 0.12}$
	DIM (Ours)	$\mathbf{35.6}_{\pm 0.10}$	$\mathbf{45.1}_{\pm 0.04}$	$\mathbf{54.7}_{\pm 0.11}$

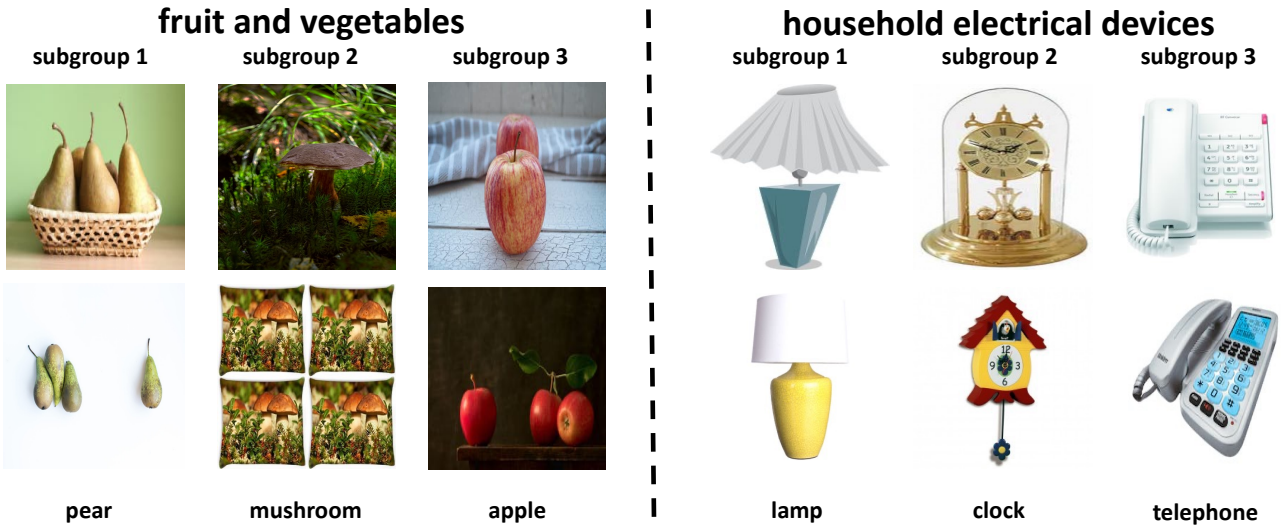

Figure 6. The CLIP Retrieval results of discovered subgroups in the class of "fruit and vegetables" and "household electrical devices" respectively. Images in each column come from the same identified subgroup.

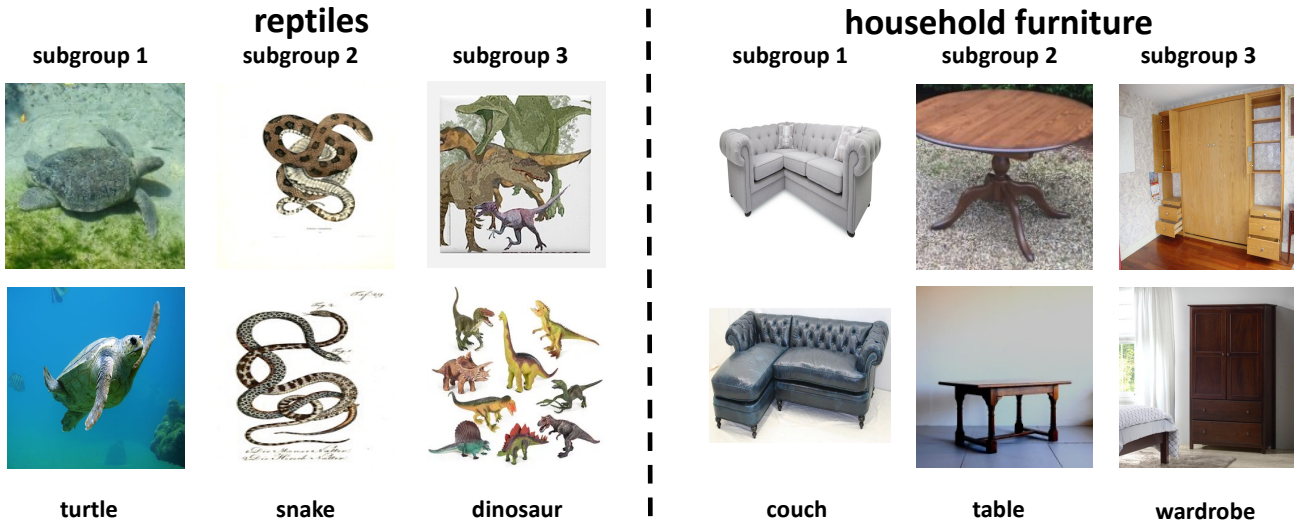

Figure 7. The CLIP Retrieval results of discovered subgroups in the class of "reptiles" and "household furniture " respectively. Images in each column come from the same identified subgroup.

For all of the baseline methods, we use the Adam optimizer to train the model. The grid search has fine-tuned hyperparameters to achieve the best performance.

Statistical significance As shown in Tab. 6, we train each model for 5 times and report the mean accuracy for a fair comparison. Specifically, for the most efficient data-centric mitigation of ResNet-18 on CIFAR-10, our method has an average accuracy on the worst two subgroups of $40.35 \pm 0.07\%$, while that of the runner-up method [12] is $37.80 \pm 0.11\%$.The performance gap without overlap indicates significant performance improvement.

D. Experiments on CIFAR-100

D.1. Experimental Details

We train ResNet-18 on the CIFAR-100 dataset from scratch. We use SGD as the optimizer. We set the learning rate as 0.1 as the learning rate and the batch size as 128. We set the number of training epochs as 50 and used early stop with validation loss

as the criterion to eliminate the overfitting problem.

D.2. Subgroup Interpretation

We show examples of detecting multiple subgroups within the CIFAR-100 dataset. As displayed in Fig. 6, our method can accurately capture multiple unknown subgroups, the "pear", "mushroom", and "apple" in the "fruit and vegetables" class, "lamp", "clock", and "telephone" in the "household electrical devices" class. "pear", "mushroom", "lamp", and "clock" are low-performance subgroups. "apple" and "telephone" are easy subgroups. In Fig. 7, our DIM successfully identified subgroups, "turtle", "snake", and "dinosaur", in the class "reptiles" and "household furniture", and "couch", "table", and "wardrobe" in the class "household funiture". "turtle", "snake", "couch", and "table" are low-performance subgroups. "dinosaur" and "wardrobe" are easy subgroups.

Ablation study on supervision. We conduct experiments on studying the impact of supervision in the decomposition of our proposed DIM. Without supervision, the use of the PLS at the decomposition stage degrades to the PCA. We replace the PLS with PCA in DIM. As shown in Fig. 8, in the "large man-made outdoor things" class of CIFAR-100, we can see that the PCA can not accurately discover the "bridge" subgroup, which is confused by the spurious correlation between bridge and water. For comparison in Fig. 3, the PLS can accurately discover the "bridge" subgroup.

It supports our argument that it is impossible to discover the multiple subgroups in the image classifier without the supervision of the studied model.

Figure 8. The CLIP-Retrieval results for interpreting the multiple subgroups within the "large man-made outdoor things" class discovered by DIM-PCA on the CIFAR-100 dataset.

D.3. Bias Mitigation

Ablation study on number of subgroups. We conduct experiments to study the impact of the number of subgroups to be discovered, "n", on the classification accuracy in data-centric mitigation. When we reduce n from 5 to 3 on the CIFAR-100 dataset, the mean accuracy of the worst two subgroups is reduced from 40.35% to 39.22%, which is still higher than Jain *et al*. Jain *et al.* [12] by 1.42%. We refrain from reducing n to 2 as this would lead DIM to address only the single bias issue, failing to showcase our motivation and how DIM differs from Jain *et al*. Jain *et al.* [12].

E. Experiments on Breeds

E.1. Experimental Details

We train ResNet-34 on the Breeds dataset from scratch. We use Adam as the optimizer. We set the learning rate as 0.01 as the learning rate and the batch size as 32. We set the number of training epochs as 100 and used early stop with validation loss as the criterion to eliminate the overfitting problem.

E.2. Subgroup Interpretation

We present several examples of discovering multiple subgroups in ResNet-34 on the Breeds dataset. As shown in Fig. 9, our method can accurately capture multiple unknown subgroups, namely the "shopping car", "passenger car", and "unicycle" in the "vehicle" class, "cloak", "miniskirt", and "abaya" in the "garment" class. Another confidence is provided in Fig. 10. We can see that there are three[3] subgroups accurately discovered by our method, namely the "bassinet", "four-poster", and "mosquito net" in the "furniture" class, "steel drum", "corkscrew", and "scale" in the "instrument" class.

F. Experiments on Hard ImageNet

F.1. Experimental Details

We train ResNet-50 on the Hard ImageNet dataset. The ResNet-50 is pre-trained on the full ImageNet dataset and fine-tuned on the Hard ImageNet dataset. We use Adam as the optimizer. We set the learning rate as 0.01 as the learning rate and the batch size as 32. We set the number of training epochs as 100 and used early stop with validation loss as the criterion to eliminate the overfitting problem.

Figure 9. The CLIP Retrieval results of discovered subgroup embeddings in the class of "vehicle" and "garment" respectively. Images in each column come from the same identified subgroup.

Figure 10. The CLIP Retrieval results of discovered subgroup embeddings in the class of "furniture" and "instrument" respectively. Images in each column come from the same identified subgroup.

F.2. Subgroup Interpretation

In Fig. 5, we uncover the implicit subgroups of the "balance beam" and "dog sled" classes in the Hard ImageNet dataset. Here, we additionally present four examples of the implicit multiple subgroups, namely the "seat belt", "keyboard space bar", "hockey puck", and "gymnastic horizontal bar", in Fig. 11 and Fig. 12, respectively. We provide the analysis of the spurious correlations involved in the dataset as follows.

"balance beam". The presented subgroups are "a group of little kids playing", "the women's uneven bars event", and "'balance beam". The model learns two biases, namely, the population and the scene. For the population, it means that the balance beam often comes up with kids, indicating a spurious correlation. For the scene, the balance beam is used for competition, causing unintended bias.

"dog sled". The presented subgroups involve "some people are skiing on the snow", "there are a lot of dogs", and "there

[3]Here, we only present the results of the discovered three subgroups, while our method discovers 10 subgroups in the experiments.

Figure 11. Example of subgroup interpretation on Hard ImageNet. The first two rows are the retrieval images of identified biased subgroups and corresponding summary descriptions by ChatGPT based on metadata. The last row is from the high-performance subgroup.

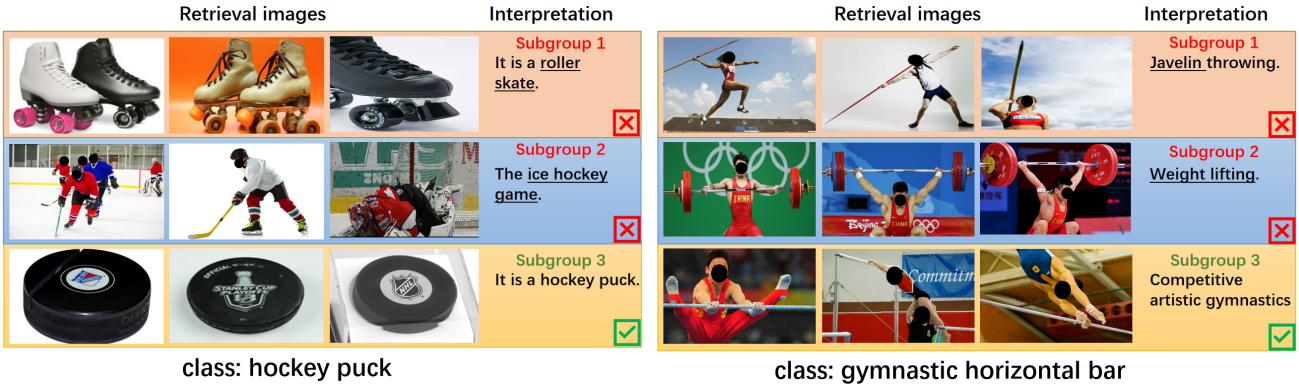

Figure 12. Example of subgroup interpretation on Hard ImageNet. The first two rows are the retrieval images of identified biased subgroups and corresponding summary descriptions by ChatGPT based on metadata. The last row is from the high-performance subgroup.

are dog sleds". Correspondingly, the results indicate the model unindently learns two biases. The first one comes from the background, where the dog sled usually appears in the snow. The second bias comes from the object, where the model is biased to the spurious correlation of dogs.

"seat belt". The results of "seat belt" indicate that the model is largely biased by the background and unrelated objects, namely the car and belt, respectively.

"keyboard space bar". The biases of the model on the "keyboard space bar" mainly derive from the objects, namely the laptop and the typewriter, which are the source of the keyboard.

"hockey puck". The biases of the model on the "hockey puck" mainly derive from the objects, namely the roller skate and ice hockey game, which usually come up with the hockey puck.

"gymnastic horizontal bar". It is interesting to see the results of the subgroup discovery in the "gymnastic horizontal bar". It can be seen that the model detects the horizontal bar by the shape, leading to the wrong classification of the javelin and barbell.