A DISCUSSION

In this section, we elaborate on SPIRE's strengths, its weaknesses, and suggested directions for future work. While there are many ways to improve SPIRE, we have demonstrated that it is a clear step forwards for the problem of addressing SPs.

Generating Counterfactual Images. SPIRE relies on the ability to produce counterfactual images. As a result, finding ways to produce similar counterfactuals with fewer assumptions (e.g., being able to add/remove objects without relying on having an annotated dataset) or to produce different types of counterfactuals (e.g., changing attributes such as "color") are both directions for future work. The former would improve the general applicability of SPIRE while the later would increase the scope of the types of SPs SPIRE could address.

Identification. SPIRE's strategy for identification can be summarized as "measure the probability that the model's prediction changes when we take an image from Group X and apply Counterfactual Transformation Y." Intuitively, this strategy is effective because the original and counterfactual versions of an image differ only in terms of the effect of the counterfactual transformation while, if we were to compare natural images in one group to another group, there are probably going to be additional differences. Because SPIRE uses X = Both, it may not be as effective as possible for identifying negative SPs because this split is likely to be very small for negatively correlated objects. As a result, future work could increase the scope of the types of SPs SPIRE could identify by considering different definitions of X (e.g., X is the set of images that have objects $1, \ldots, m$ and do not have objects $m + 1, \ldots, n$; X is the set of images where objects 1 and 2 appear near to/far from each other) or Y (e.g., Y removes objects 1 and 2; Y changes the location of object 1).

Interestingly, we find that a strong correlation is neither sufficient (Figure 6 shows that the model can ignore a strong correlation) nor necessary (Table 2 shows that some SPs are between objects that are almost uncorrelated) for a model to learn to use a SP, which is consistent with prior findings (Shah et al., 2020; Nagarajan et al., 2020). These result demonstrates SPIRE's advantage over identification methods that only consider the training distribution (e.g., Wang et al., 2020).

Mitigation. To begin with, it is worth noting that mitigating a SP may not always be worthwhile (e.g., when one is certain that the distribution will not shift).

At a high level, SPIRE's strategy for mitigation works by removing the statistical incentive for the model to rely on the SP, while trying not to add new SPs; this strategy may be less effective for SPs that do not arise from correlations in the training distribution. Previous augmentation-based mitigation methods might be less effective because they are intuitive rather than statistical (e.g., it makes intuitive sense that removing people should lessen the model's reliance on people to detect tennis rackets, but this intuition does not carry over to the dataset statistics). Previous regularization-based mitigation methods might be less effective because they may interfere with the learning process (e.g., cause the model to become stuck in a local minimum) or they may have effects that are too local to matter (e.g., changing the model's gradient at a point may not change its predictions very far away from that point). In particular, the Feature Splitting (FS) method from (Singh et al., 2020) assumes that one half of the features learned by the model are relevant for detecting objects "in context" and that the other half are relevant for objects "out of context;" while plausible for a single SP, this assumption becomes more tenuous as the number of SPs being mitigated increases.

While SPIRE's mitigation strategy is defined by two high-level goals, it is not always successful at realizing those goals (e.g., for $p < 0.5$ in Section 5.1, SPIRE introduces the potential for new SPs) and the way those goals are realized depends on the problem setting. As a result, future work could improve the general applicability of SPIRE by finding a unified strategy that does not depend on the problem setting, generalizing that strategy to work for more general SPs, and extending that strategy to problems other than image classification. Additionally, future work could develop a theoretical framework to help understand the effects of augmentation-based mitigation strategies.

B METHOD DETAILS

B.1 GENERATING COUNTERFACTUAL IMAGES

Similar to prior work, SPIRE generates counterfactual images by adding objects to or removing objects from the original image (Shetty et al., 2019; Teney et al., 2020; Xiao et al., 2021; Chen et al., 2020a; Liang et al., 2020; Agarwal et al., 2020). In this work, use the pixel-wise object-annotations that are part of various datasets such as COCO to generate the counterfactual images. Figure 5 shows examples. Orthogonally, there is prior work that generates fundamentally different types of counterfactual images (Neto, 2020; Zhang & Sang, 2020; Goel et al., 2021; Sauer & Geiger, 2021).

Removing an Object. We consider two different ways to define which region of the image we are going to replace (pixel-wise or bounding-box) and two different ways to in-fill that region (using constant grey color or in-painting with the model from (Nazeri et al., 2019)). When we say that we "remove" an object, we mean that we found its bounding-box region and in-filled it with grey. When we say that we "in-paint" an object, we mean that we found its pixel-wise region and in-painted it. In order to minimize label noise, we make sure we do not include Main in the region that is going to be removed when we are removing Spurious.

Adding an Object. To add an object to an image, we find the pixel-wise region for that object in a different image and then replace that region's counterpart in the original image with it. In order to minimize label noise, we make sure that we do not cover Main when we add Spurious.

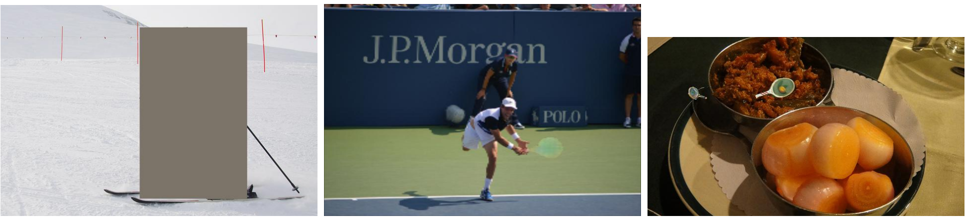

Figure 5: Example counterfactual images for the tennis racket example. **(Left)** An example of moving an image from Just Spurious to Neither by Removing Spurious. **(Center)** An example of moving an image from Both to Just Spurious by In-Painting Main. **(Right)** An example of moving an image from Neither to Just Main by Adding Main.

B.2 WHAT DOES IT MEAN TO INTRODUCE NEW POTENTIAL SPS?

We try to minimize the potential for new SPs by ensuring that P(Main | Artifact) = 0.5, where the Artifact could be "Grey Box" from removing objects from an image or objects with "Unusual Placement" from adding objects to an image. However, it is not clear whether 0.5 or P(Main) is the "correct" choice for this value. One one hand, using P(Main | Artifact) = 0.5 maximizes the loss that the model will receive if it relies on Artifact. On the other hand, setting P(Main | Artifact) = P(Main) means that Main is independent of Artifact and that there is no statistical incentive for the model to rely on Artifact. Because we will not be evaluating the model (in terms of accuracy) on images with Artifact, we chose 0.5 because it actively discourages using Artifact rather than simply not encouraging it.

B.3 SETTING 1: WORKING THROUGH SPIRE'S AUGMENTATION STRATEGY

In this setting, {Both, Neither} each have size $0.5p$ while {Just Main, Just Spurious} have size $0.5(1-p)$.

For $p > 0.5$, SPIRE removes {Main, Spurious} from Both with probability $\frac{2p-1}{2p}$ and, as a result, P(Main | Grey Box) = 0.5. Similarly, SPIRE adds {Main, Spurious} to Neither with the same probability and, as a result, P(Main | Unusual Placement) = 0.5. As a result, {Just Main, Just Spurious} each receive $0.25(2p-1)$ images from each of {Both, Neither} and have an augmented size of $0.5p$. So SPIRE produces the balanced distribution without creating the potential for new SPs.

For $p < 0.5$, SPIRE adds Main to Just Spurious and adds Spurious to Just Main with probability $\frac{p-0.5}{p-1}$ and, as a result, P(Main | Unusual Placement) = 1. Similarly, SPIRE removes Spurious from Just Spurious and removes Main from Just Main with the same probability and, as a result, P(Main | Grey Box) = 0. As a result, $\{$Both, Neither$\}$ each receive $0.5(0.5 - p)$ images from each of $\{$Just Main, Just Spurious$\}$ and have an augmented size of $0.5(1 - p)$. So SPIRE produces the balanced distribution while creating the potential for new SPs.

C EVALUATION DETAILS

C.1 WHY NOT SET P(SPURIOUS | MAIN) = P(SPURIOUS | NOT MAIN) = P(SPURIOUS) FOR THE BALANCED DISTRIBUTION?

Using P(Spurious) instead of 0.5 may be an intuitive choice because it would mean that the main statistical difference between the original and balanced distributions is that Main and Spurious are now independent. However, doing so can have dramatic and unexpected effects on which splits are more important for evaluation. To see this, consider Main = "fork" and Spurious = "dining table". For the original distribution, we have P(Spurious | Main) $= 0.76$ which means we have, roughly, a 3:1 ratio of images in Both to Just Main. For the balanced distribution, using $\lambda =$ P(Spurious) $= 0.1$ would change this ratio to 1:9. Not only would this choice change which split is more important for evaluation (from Both to Just Main) but it would also would increase the degree to which that split is more important (from a factor of 3 to a factor of 9). Without domain knowledge telling us that such a dramatic shift is warranted, using 0.5 is the more conservative option because assigning equal importance to images with and without Spurious never flips which splits are more important for evaluation.

C.2 WHY DO SMALL, IN ABSOLUTE TERMS, HALLUCINATION GAPS MATTER?

To understand this, consider the per split accuracies for the tennis racket example (Figure 3 Left) where we observe that the Hallucination gap is "only" 0.5% and may be tempted to conclude that it is not significant. However, when we look at where the model's errors come from on the original distribution, we find that roughly 40% of them come from Just Spurious, despite the model's 99.5% accuracy on this split. This means that the model's performance is sensitive to both small changes to its accuracy on Just Spurious and Neither and distribution shifts that move weight between Just Spurious and Neither.

As a result, small, in absolute terms, changes to the Hallucination gap can have large impacts on the model's robustness to distribution shift. In general, we adjust for this by measuring changes in the gap metrics relative to their original value (e.g., if the new model had a hallucination gap of 0.25% we would say that it "reduced the hallucination gap by a factor of 50%").

C.3 WHY CAN THE GAP METRICS CHANGE MUCH MORE THAN PERFORMANCE ON THE BALANCED DISTRIBUTION?

In general, mitigation methods shrink the gap metrics by sacrificing accuracy on the splits where relying on the SP is helpful in order to gain accuracy on the other splits; whether or not this trade-off improves performance on the balanced distribution depends on how much accuracy is sacrificed and gained. As a result, the size of the gap metrics and performance on the balanced distribution are not necessarily closely connected.

C.4 COUNTERFACTUAL EVALUATION

While the evaluations described in Section 4 are all based on the natural images, we also run a *counterfactual evaluation*. Unlike in SPIRE's identification step, where we only measure the probability that "removing Spurious from an image from Both" changes the model's prediction, this evaluation measures the probability that the model's prediction changes when we move an image from one split to another for each pairs of splits that differs by one object. This acts as an additional sanity check that a mitigation strategy has reduced the model's reliance on a SP, but we consider it to be less important than the model's performance on the balanced distribution and the gap metrics because its results depend on the specific definition of the counterfactuals used (e.g., it is easy to do well on this evaluation for a specific type of counterfactual by training the model on that same type of counterfactual).

D MODEL TRAINING DETAILS

Many of our experiments are based on the COCO dataset (Lin et al., 2014). Because the test set for this dataset is not publicly available, we used its validation set as our test set and divided its training set into 90-10 training and validation splits.

All of our experiments started with the pretrained ResNet18 (He et al., 2016) that is available from PyTorch (Paszke et al., 2019). For each task, the classification layer was replaced with one of the appropriate dimension and then trained via transfer learning (i.e., only the classification layer had its weights updated). The resulting model was then fine-tuned (i.e., all of its weights were updated) to produce what we call the Baseline Model throughout this work.

Optimization. We minimized the binary cross entropy loss using Adam (Kingma & Ba, 2014) with a batch size of 64. For transfer-learning, we used a learning rate of 0.001 and, for fine-tuning, we used a learning rate of 0.0001; we explored other options during early experiments, but found there was no benefit to doing so. If the training loss failed to decrease sufficiently after some number of epochs, we lowered the learning rate.

Model selection. During the training process, we selected the best model weights using their performance on the validation set. For the Benchmark Experiments, we measured performance using Accuracy and, for the Full Experiment, No Object Annotation Experiment, Scene Identification Experiment, and ISIC Experiment, we used F1. If the validation performance failed to increase sufficiently after some number of epochs, we stopped training.

Benchmark Experiments: Hyper-parameter selection. For this experiment, we tuned the hyper-parameters using balanced accuracy on the bottle-person pair with $p = 0.95$. For all methods, we considered both transfer-learning and fine-tuning, as applicable. For SPIRE, we considered both removing objects by covering them with a grey box and by in-painting them; we found that transfer-learning while covering objects with a grey box was the most effective. RRR, CDEP, and GS all have regularization weights that can be tuned. FS has a minimum weight for images of objects “out of context” that can be tuned. For these methods, we considered values that are powers of 10 ranging from 0.1 to 10,000; no method chose one of the extreme values.

Full Experiment: Hyper-parameter selection. For this experiment, we tuned the hyper-parameters using the mean, across SPs, Average Precision on the balanced distribution for a model trained on 50% of the training dataset and then evaluated on the remaining 50% of the training dataset; we used such large chunk of the dataset for evaluation in order to be able to estimate the per split accuracies, which are required to calculate Average Precision on the balanced distribution.

For SPIRE, we still used transfer-learning while covering objects with a grey box. However, we tune the weight of the augmentation by scaling δ from Setting 2 in Section 3.1 by a factor of $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$; intuitively, this is to prevent us from adding too many counterfactual images. Note that the weight for each SP is tuned independently and each weight is tuned by training a linear classifier.

For FS, the configuration chosen by this procedure yielded poor results and, consequently, we used the default value of 3 for our results (Singh et al., 2020).

E ADDITIONAL RESULTS: BENCHMARK EXPERIMENTS - SECTION 5.1

Creating the benchmark. While varying p allows us to control the strength of the correlation between Main and Spurious (i.e., p near 0 indicates a strong negative correlation while p near 1 indicates a strong positive correlation), it does not guarantee that the model actually relies on the intended SP. Indeed, when we plot the models' balanced accuracy as p varies (Figure 6), we observe that 5 out of the 13 pairs show little to no loss in balanced accuracy as p approaches 1 (dashed lines). Consequently, subsequent evaluation considers the other 8 pairs (solid lines). For these pairs, the model's reliance on the SP increases as p approaches 0 or 1 as evidenced by the increasing loss of balanced accuracy and confirmed via counterfactual evaluation (Figure 8).

Counterfactual Evaluation. For models that are trained on a dataset augmented with a specific type of counterfactual images, the results of this evaluation for that type of counterfactual are often skewed and, consequently, we exclude those results. Specifically, this means that: SPIRE is only evaluated on In-Painting counterfactuals, QCEC is not evaluated on In-Painting counterfactuals, and GS is not evaluated on counterfactuals that In-Paint Main.

Figure 8 shows the results (averaged across the chosen object pairs and eight trials per pair). The first thing to note is that all of the counterfactual evaluations show that the Baseline model is relying on the intended SP because their results get worse as P(Main | Spurious) approaches 0 or 1 (i.e., there is a strong negative or positive correlation between Main and Spurious in the training dataset). Observe that SPIRE improves all of evaluations based on In-Painting with the exception of "Just Spurious and In-Paint Spurious" for $0.05 < p < 0.5$. In contrast, the other mitigation methods have clear and consistent failures (e.g., RRR, CDEP, GS, and FS all make the evaluation worse for "Both and Remove Main", QCEC makes the evaluation worse for "Neither and Add Main").

Per split analysis. By looking at the models' accuracy on each split (Figure 7, averaged across the chosen object pairs and eight trials per pair), we see that SPIRE exhibits all of the expected signs of a method that is reducing a model's reliance on a SP:

- It sacrifices accuracy on splits where relying on the SP is helpful (e.g., Both for $p > 0.5$ and Just Main for $p < 0.5$) in order to gain accuracy on the splits where the SP is not helpful (e.g., Just Main for $p > 0.5$ and Both for $p < 0.5$).
- It substantially flattens the per split accuracy curves for images with Spurious and, to a lesser extent, flattens them for images without Spurious. This indicates that it produces a model that is less sensitive to the original training distribution.

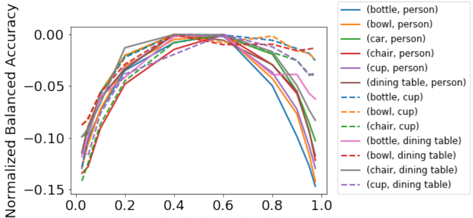

Figure 6: For each pair of objects, we plot the models' balanced accuracy as we vary p for the training set. The y-axis is normalized so that we can easily compare the curvature of the plots. We either accept (solid line) or reject (dashed line) pairs based on whether or not we see a significant drop in balanced accuracy both as p approaches 0 and as it approaches 1. The rejected pairs show an insufficient drop as p approached 1.

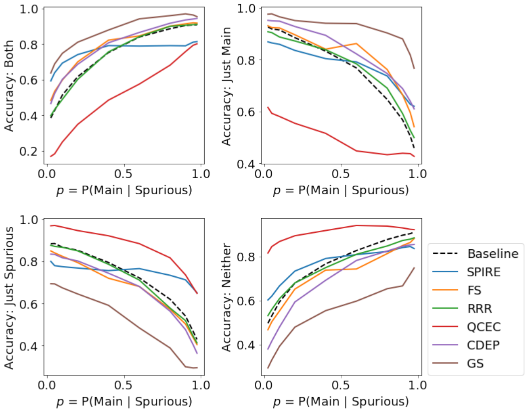

Figure 7: The models' accuracies on each split.

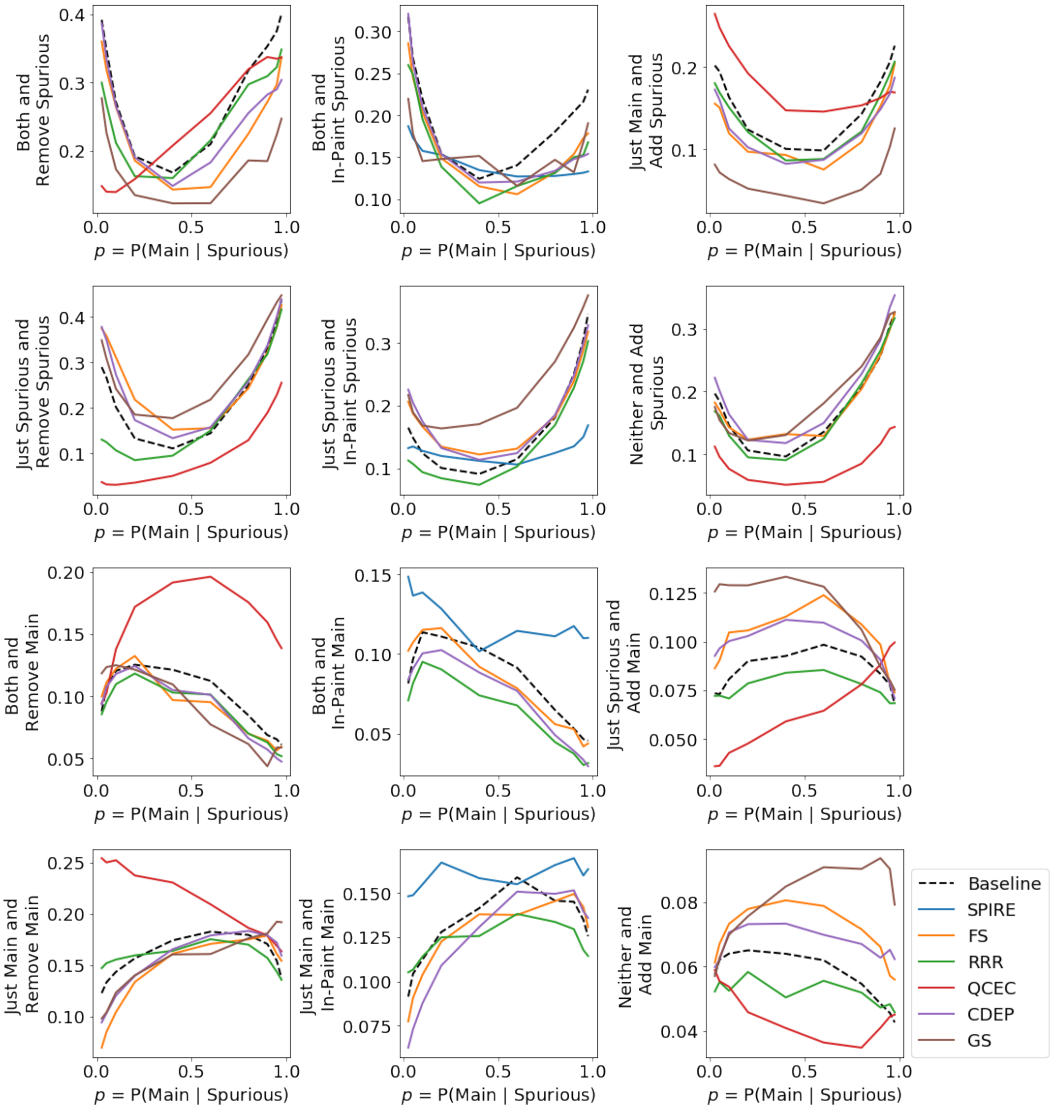

Figure 8: The columns correspond to Removing, In-Painting, and Adding an object. The first two rows do that to Spurious and, as a result, a lower value is better. The last two rows do that to Main and, as a result, a higher value is better. Methods that train on an augmented dataset that contains a certain type of counterfactual are excluded from its evaluation because their results are usually skewed.

F ADDITIONAL RESULTS: FULL EXPERIMENT- SECTION 5.2

Validating the Identified SPs. In Figure 9, we verify that the model has large recall and hallucination gaps, indicating that it is relying on these SPs.

SPIRE's effect on each SP. Figures 10, 11, and 12 show SPIRE's effect on the Balanced Average Precision, the Average Recall Gap, and Average Hallucination Gap respectively. SPIRE improved Balanced Average Precision by an average of 1.1% with a positive change for 21 of the SPs. SPIRE decreased the Average Recall/Hallucination Gaps by an average factor of 14.2%/28.1% for 24/29 of the SPs. Overall, these results indicate that SPIRE consistently reduces the model's reliance the identified SPs.

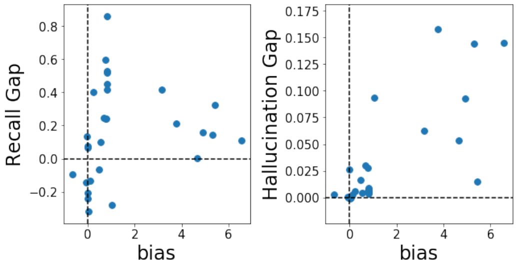

Figure 9: We expect positive gaps for SPs with positive bias and negative gaps for negative SPs. In general, this is what we find. **(Left)** A comparison of the Recall Gap to the bias of the dataset for the SP. **(Right)** A comparison of the Hallucination Gap to the bias.

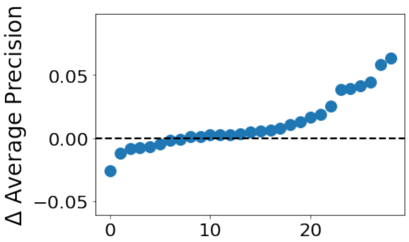

Figure 10: The Average Precision on the balanced distribution for SPIRE compared to the baseline for each SP.

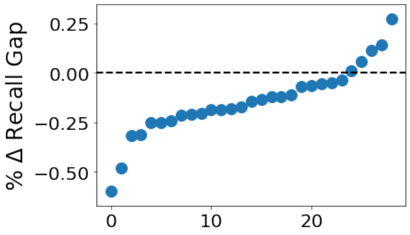

Figure 11: The percent change of the Average Recall Gap for SPIRE compared to the baseline model for each SP.

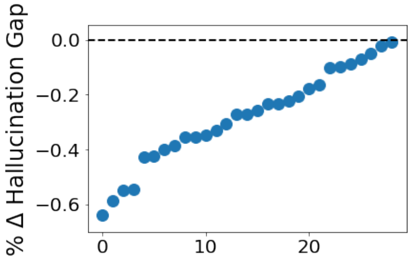

Figure 12: The percent change of the Average Hallucination Gap for SPIRE compared to the baseline model for each SP.

Tennis Racket Example: Full Version. Here, we walk through the evaluation used in Section 5.2 for the tennis racket example. Figure 13 shows the results. The numbers in the legends are "mean (standard deviation)" across 8 trials for the metric measured in that plot.

Top Left: Average Precision. This panel shows the model's Precision vs Recall curve, for the balanced distribution, which we use to calculate Average Precision by finding its Area Under the Curve (AUC). SPIRE improves Average Precision on the balanced distribution for this SP by 0.6%.

Top Middle: Average Recall Gap. This panel shows the model's recall gap (the absolute value of the difference of the model's accuracy on Both and Just Main) vs its Recall on the balanced distribution. We calculate this metric by finding the AUC. SPIRE decreases this metric by 31.4% which means that it produces a model that is more robust to distribution shifts that move probability between Both and Just Main.

Top Right: Average Hallucination Gap. This panel shows the model's hallucination gap (the absolute value of the difference of the model's accuracy on Just Spurious and Neither) vs its Recall on the balanced distribution. We calculate this metric by finding the AUC. SPIRE decreases this metric by 25.0% which means that it produces a model that is more robust to distribution shifts that move probability between Just Spurious and Neither.

Center Row: Accuracy on Both and Just Main. These panels plot the model's accuracy on Both/Just Main vs its Recall on the balanced distribution. The value shown is the AUC of this curve. Because the baseline model uses the presence of a person to help detect a tennis racket, we expect a model that does not rely on this SP to lose accuracy on Both and gain it on Just Main. SPIRE does this.

Bottom Row: Accuracy on Just Spurious and Neither. These panels plot the log of the model's accuracy on Just Spurious/Neither vs its Recall on the balanced distribution. The value shown is the AUC of this curve (before taking the log). Because the baseline model uses the presence of a person to help detect tennis rackets, we expect a model that does not rely on this SP to lose accuracy on Neither and gain it on Just Spurious. Because SPIRE improved AP, we do not see this because it's accuracy on these splits is higher for most levels of recall.

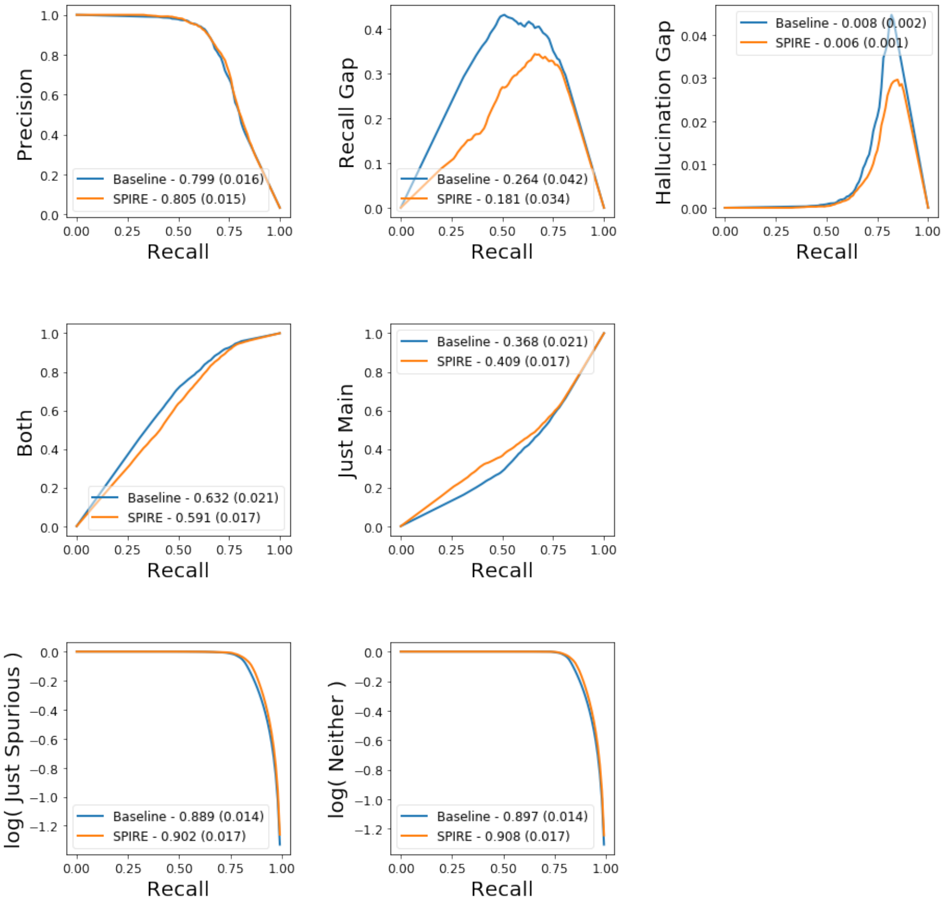

Figure 13: The results of our evaluation for the tennis racket example. The numbers in the legends are "mean (standard deviation)" across 8 trials. SPIRE improved Average Precision on the balanced distribution by 0.6%, decreased the average recall gap by 31.4%, and decreased the average hallucination gap by 25.0%. Further, it had the expected effect of decreasing accuracy on Both and increasing it on Just Main. As a result, we conclude that it reduced the model's reliance on this SP.

G ADDITIONAL RESULTS: GENERALIZATION EXPERIMENTS- SECTION 5.3

G.1 ISIC EXPERIMENT- PIPELINE FOR CREATING COUNTERFACTUALS

Our pipeline, which is based on clustering image segments (i.e., super-pixels), is constructed as follows:

- We use an image segmentation algorithm to extract segments from an image and represent each segment using its mean RGB value.
- We run a hierarchical clustering on those RGB values to produce nine clusters. Then, we manually inspect several randomly sampled images from each cluster and label those clusters based on whether or not they represent stickers.
- Finally, we use a K-NearestNeighbor classifier to predict which of those nine clusters an image segment belongs to.

Overall, this pipeline produces a per-image map of which pixels belong to a sticker and identifies stickers with 86.7% recall and 99.0% precision. We use this map to produce counterfactual images.