Correct-N-Contrast: a Contrastive Approach for Improving Robustness to Spurious Correlations

Michael Zhang [1] **Nimit S. Sohoni** [1] **Hongyang R. Zhang** [2] **Chelsea Finn** [1] **Christopher Ré** [1]

Abstract

Neural networks trained with empirical risk minimization (ERM) may learn spurious correlations, resulting in poor performance on data groups without these correlations. To tackle this issue when group labels are unknown at training time, we propose Correct-N-Contrast (CNC), a two-stage contrastive learning method to learn representations only dependent on ground-truth labels. By first using the outputs of a trained ERM model to sample contrastive batches where anchors and positives have the same class label, but different ERM model outputs, we infer pairs of same-class samples with different spurious attributes. Then with contrastive learning, we train a new model to learn similar representations for these samples. Theoretically, we support CNC by showing that worst-group loss is upper bounded by the average loss and a term that depends on the representation metrics CNC explicitly minimizes. We validate CNC on popular image and language datasets, where CNC obtains state-of-the-art worst-group accuracy (on average 22.6% absolute lift over ERM). CNC also outperforms the "oracle" GDRO approach that requires group labels by 1.5%.

1. Introduction

For many tasks, deep neural networks are negatively affected by spurious correlations—dependencies between observed features and ground-truth labels that hold for some, but not all, subsets or "groups" of the data. For example, imagine classifying images of cows or camels, where 90% of cow images depict grassy backgrounds. A model may learn to predict the "cow" class based on the spurious background attribute, and incorrectly classify cow images with non-grass backgrounds as camels (Ribeiro et al., 2016; Beery et al., 2018; Kaufman et al., 2012). This illustrates the issue where training with empirical risk minimization (ERM) can result in low test error on certain groups, yet high error on others (Blodgett et al., 2016; Buolamwini & Gebru, 2018; Hashimoto et al., 2018; Duchi et al., 2019). Prior work has shown that this problem is increasingly aggravated as correlations between ground-truth and spurious attribute become stronger (Sagawa et al., 2020) and "easier" to learn (Arpit et al., 2017; Hermann & Lampinen, 2020). Unfortunately, such correlations exist in many safety-critical settings, motivating our goal to improve performance on all groups.

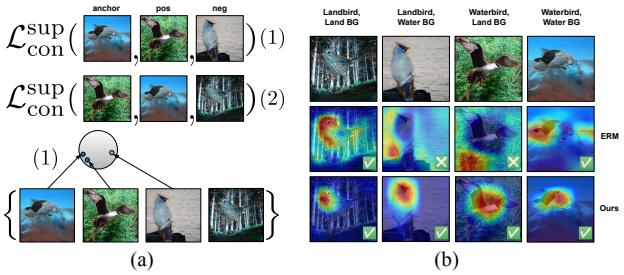


Figure 1. (a) In CNC, we use contrastive learning to learn similar representations for datapoints with the same ground-truth but different ERM predictions, while keeping apart those with different labels and the same ERM predictions. (b) Models then ignore the spurious attributes and achieve greater robustness to spurious correlations, visualized with GradCAM (Selvaraju et al., 2017).

How can we obtain models robust to spurious correlations? If group labels are known, one option is to directly minimize the worst-group loss, e.g. with Group DRO (GDRO) (Sagawa et al., 2019). However, such labels may be expensive to obtain. This presents the additional challenge to improve worst-group accuracy when group labels are unknown. To tackle this, a recent approach is to infer the groups first from a trained ERM model, before training another model with an inferred-group-reweighted loss (Sohoni et al., 2020; Nam et al., 2020; Liu et al., 2021). While they achieve best worst-group accuracy among methods not requiring training data group labels, they do not perform as well as methods *with* group labels on popular benchmarks.

We thus aim to improve the worst-group accuracy of methods not requiring group labels. Our core motivating observation is that a neural network's worst-group accuracy correlates with how well its data representations–i.e. the outputs from its last hidden layer–exhibit dependence *only*

[1]Department of Computer Science, Stanford University [2]Khoury College of Computer Sciences, Northeastern University. Correspondence to: Michael Zhang <mzhang@cs.stanford.edu>.

Presented at the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning.,

on the labels, and not spurious attributes. We quantify this property through both estimated mutual information and our own notion of representation *alignment* (Wang & Isola, 2020), which measures how close representations of samples with the same class but different spurious attribute labels are in Euclidean distance. From this motivation, we theoretically establish that alignment indeed plays a key role in bounding a model's worst-group vs. average error gap. However, current approaches do not explicitly optimize for alignment, suggesting one direction to improve worst-group performance by learning better-aligned representations.

We therefore propose Correct-N-Contrast (CNC), a two-stage robust training framework using contrastive learning to directly maximize alignment and learn representations with only ground-truth dependence. First we train an initial ERM model to predict ground-truth labels from the data. We then train a second model to learn representations that "pull together" points with the same label but different ERM model outputs (*anchors* and *positives*), and "push apart" points with different labels but the same ERM outputs (anchors and *negatives*). Intuitively, this encourages the model to ignore ERM-learned spurious correlations, and maximize class-consistent information instead.

Empirically, we find that CNC closes the gap on popular benchmarks in worst-group accuracy between robust training methods that do and do not assume group labels, achieving 2.5% absolute improvement over GDRO on Waterbirds (Wah et al., 2011) and CelebA (Liu et al., 2015), while falling short of GDRO on CivilComments (Borkan et al., 2019) by 0.7%. Furthermore, among methods that do not assume group labels, for worst-group accuracy CNC obtains 22.6% absolute lift over the ERM baseline (from 59.7% to 82.3%), and 3.5% over the prior state-of-the-art. These gains also help validate our theory, where CNC achieves more than twice as high alignment than ERM, and consistently pairs highest alignment and worst-group accuracy.

2. Background

Problem setup. We consider spurious correlations in the context of classification (Sagawa et al., 2020). For training dataset $X = (x_1, \ldots, x_n), Y = (y_1, \ldots, y_n)$, each datapoint has observed features $x_i \in \mathcal{X}$, label $y_i \in \mathcal{Y}$, and *unobserved* spurious attribute $a_i \in \mathcal{A}$. The set of groups $\mathcal{G}$ is defined as the set of all combinations of (label, spurious attribute) pairs, i.e. $\mathcal{G} = \mathcal{Y} \times \mathcal{A}$. We denote $k = |\mathcal{G}|$, $C = |\mathcal{Y}|$. Each example (x_i, y_i, a_i) is sampled from the joint distribution P, a mixture of the k per-group distributions P_g. We assume that at least one sample from each group is observed in the training data.

To model spurious correlations, we let p_{corr} be the fraction of datapoints that belong to groups with a strong correlation between a and y (e.g. cow-grass images). Typically, spurious correlations are problematic when p_{corr} is very large. Given a model $f_\theta : \mathcal{X} \mapsto \mathbb{R}^C$ and a convex loss $\ell : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$, we wish to minimize the worst-group loss:

$$\mathcal{L}_{\text{wg}}(f_\theta) := \max_{g \in \mathcal{G}} \mathbb{E}_{(x,y,a)\sim P_g}[\ell(f_\theta(x), y)] \tag{1}$$

In contrast, standard ERM minimizes the average training loss as a surrogate for the expected population loss $\mathcal{L}_{\text{avg}}$:

$$\mathcal{L}_{\text{avg}}(f_\theta) := \mathbb{E}_{(x,y,a)\sim P}[\ell(f_\theta(x), y)] \tag{2}$$

The core problem is that spurious correlations may cause minimizers of Eq. (2) to obtain low average loss over P but high error on minority groups, as high p_{corr} may encourage learning undesired dependencies on a.

Representation metrics. To quantify this spurious correlations behavior, we use two metrics to capture the learned representation dependence on ground-truth vs. spurious attributes. First, we compute an alignment loss $\mathcal{L}_{\text{align}}$:

$$\mathcal{L}_{\text{align}}(f_{\text{enc}}) := \mathop{\mathbb{E}}_{x,x':y=y',a\neq a'} \left[\|f_{\text{enc}}(x) - f_{\text{enc}}(x')\|_2^2\right] \tag{3}$$

Here x and x' are the features of any two samples with the same label y but different spurious attributes a and a'; lower $\mathcal{L}_{\text{align}}$ denotes higher alignment. We also quantify representation dependence by estimating the mutual information (MI) of a model's learned representations with the ground-truth, i.e. $\hat{I}(Y; Z)$ and the spurious attributes $\hat{I}(A; Z)$. We defer computational details to Appendix D.

Contrastive learning. Our work is also inspired by contrastive learning, a simple yet powerful framework for representation learning (Chen et al., 2020; Oord et al., 2018; Tian et al., 2019; Song & Ermon, 2020; Sermanet et al., 2018; Hassani & Khasahmadi, 2020; Robinson et al., 2021; Khosla et al., 2020; Gunel et al., 2021). Here, an encoder learns data representations by maximizing the representation similarity of *anchor* and *positive* datapoints designed to be similar to each other, e.g. as distinct "views" of the same source / input (Chen et al., 2020), while minimizing that of anchors and *negatives* depicting a different source. We model the encoder $f_{\text{enc}} : \mathcal{X} \mapsto \mathbb{R}^d$ as the "feature representation layers"of a neural network f_θ. The final (classification) layer of f_θ [denoted $f_{\text{cls}} : \mathbb{R}^d \mapsto \mathbb{R}^C$] maps these feature representations from the encoder to a one-hot label prediction. In our work, we train f_{enc} with the *supervised contrastive loss* $\mathcal{L}_{\text{con}}^{\text{sup}}$ proposed in Khosla et al. (2020). In this setup, for each anchor x, M positives $\{x_i^+\}_{i=1}^M$ and N negatives $\{x_i^-\}_{i=1}^N$ are sampled. We let $y, \{y_i^+\}_{i=1}^M, \{y_i^-\}_{i=1}^N$ be the labels and $z, \{z_i^+\}_{i=1}^M, \{z_i^-\}_{i=1}^N$ be the normalized outputs of $f_{\text{enc}}(x)$ for the anchor, positives, and negatives respectively. With input x mapped to z, we aim to minimize $\mathcal{L}_{\text{con}}^{\text{sup}}(x; f_{\text{enc}}) =$

$$\mathop{\mathbb{E}}_{z^+,z^-}\left[-\log \frac{\exp(z^\top z^+/\tau)}{\sum_{m=1}^M \exp(z^\top z_m^+/\tau) + \sum_{n=1}^N \exp(z^\top z_n^+/\tau)}\right] \tag{4}$$

where $\tau > 0$ is a scalar temperature hyperparameter.

3. Improving representations & combating spurious correlations with CNC

We now detail CNC in Sec. 3.1 and summarize our theoretical analysis to further motivate the method in Sec. 3.2. Appendix A includes the full theorems and analysis.

3.1. Method

Stage 1: ERM training. First, we train an initial model $f_{\hat{\theta}}$ on the training dataset $\{(x_i, y_i)\}_{i=1}^n$ with standard ERM, and save its predictions $\{\hat{y}_i\}_{i=1}^n$ on the training datapoints (where $\hat{y}_i := f_{\hat{\theta}}(x_i)$). This sets up our next stage, where we use these outputs to train a more robust model.

Stage 2: Correct-N-Contrast. Next, we use these initial ERM predictions to train a robust model with contrastive learning. This distinguishes CNC from prior work (Sohoni et al., 2020; Nam et al., 2020; Liu et al., 2021), and as shown in Sec. 3.2 uniquely lets us optimize for representation alignment directly. While inspired by simple contrastive learning frameworks (Chen et al., 2020; Khosla et al., 2020), CNC also introduces its own unique "contrastive batch" sampling procedure and optimization objective to improve robustness.

Contrastive batch sampling. As described in Sec. 2, contrastive learning requires anchors, positives, and negatives $\{x\}, \{x^+\}, \{x^-\}$. Here we aim to sample points such that by maximizing the similarity between anchors and positives (and keeping anchors and negatives apart), a model "ignores" spurious similarities while learning class-consistent dependencies. With our prediction set $\{\hat{y}_i\}_{i=1}^n$, we then treat pairs of datapoints with the same y_i and different $\hat{y}_i$ as positives, and pairs of datapoints with different y_i and the same $\hat{y}_i$ as negatives. For each batch we sample random anchor $x_i \in X$ (with label y_i and ERM prediction $\hat{y}_i$), along with:

$$M \text{ positives } \{x_m^+\}_{m=1}^M \sim \{x_m^+ \in X : \hat{y}_m^+ \neq \hat{y}_i,\ y_m^+ = y_i\}$$

$$N \text{ negatives } \{x_n^-\}_{n=1}^N \sim \{x_n^- \in X : \hat{y}_n^- = \hat{y}_i,\ y_n^- \neq y_i\}$$

To achieve greater separation between negatives and both anchors and positives, we can double the pairwise comparisons in a training batch by switching the anchor and positive roles, as in the full *NT-Xent* loss (Chen et al., 2020) with an additional sampling step detailed in Appendix **??**.

To prevent the initial ERM model $f_{\hat{\theta}}$ from "memorizing" the training data and not outputting any different y_i and $\hat{y}_i$'s, we train $f_{\hat{\theta}}$ either with high ℓ_2 regularization or for a small number of epochs. More details are in Appendix D.

Optimization objective and updating procedure. For each batch, while our core objective is to learned aligned representations via contrastive learning, we also wish to train the full model to classify datapoints correctly. As we have access to the training labels, we jointly update both its encoder layers f_{enc} with a standard contrastive loss, and the full model f_θ with a cross-entropy loss. Using a "one-sided" batch $x_i, \{x_m^+\}_{m=1}^M, \{x_n^-\}_{n=1}^N$ as an example, we first forward propagate the features through f_{enc} and normalize them to obtain representation vectors $z_i, \{z_m^+\}_{m=1}^M, \{z_n^-\}_{n=1}^N$. To learn closely aligned z_i and z^+ for all $\{z_m^+\}_{m=1}^M$, we update f_{enc} with the $\mathcal{L}_{\text{out}}^{\text{sup}}(x; f_{\text{enc}})$ loss (4) (Khosla et al., 2020). Finally, we also pass the unnormalized outputs of the encoder f_{enc} to the classifier layers, and compute a batch-wise cross-entropy loss $\hat{\mathcal{L}}_{\text{cross}}(f_\theta)$ using the labels corresponding to each batch sample and the full model's outputs. Letting $\hat{\mathcal{L}}_{\text{con}}^{\text{sup}}(f_{\text{enc}})$ denote the batch-wise supervised contrastive loss and λ be a balancing hyperparameter, we then update the entire network jointly with both loss components:

$$\hat{\mathcal{L}}(f_\theta) = \lambda \hat{\mathcal{L}}_{\text{con}}^{\text{sup}}(f_{\text{enc}}) + (1-\lambda)\hat{\mathcal{L}}_{\text{cross}}(f_\theta) \tag{5}$$

3.2. Analysis

We outline our theoretical analysis, providing full theorems and proofs in Appendix A. Our theory suggests that if a model's feature representations of samples with the same class labels but from different groups are aligned, then the model's worst-group error $\mathcal{L}_{\text{wg}}(f_\theta)$ will be close to its average error $\mathcal{L}_{\text{avg}}(f_\theta)$. Training models robust to group differences then reduces to aligning the representations of different groups (e.g. by minimizing our contrastive loss) while keeping the average classification loss small. In Theorem A.1, we show that if $\mathcal{L}_{align}(f_{\text{enc}})$ is small, then $\mathcal{L}_{\text{wg}}(f_\theta)$ is close to $\mathcal{L}_{\text{avg}}(f_\theta)$. In Theorem A.2, we show that provided the full loss (5) can be minimized efficiently, the average error of its minimizer will be small plus a generalization error term that scales down with the number of training samples. Next in Sec. 4, we help validate this theory by finding that CNC minimizes $\mathcal{L}_{\text{align}}$ substantially compared to other methods while improving worst-group accuracy.

4. Experiments

We empirically aim to address if: (1) CNC improves worst-group performance over prior state-of-the-art methods on datasets with spurious correlations, and (2) if CNC actually encourages learning hidden layer representations with greater ground-truth-only dependence. We briefly describe the benchmark datasets used to answer these questions below; more details on datasets, models, and experimental hyperparameters can be found in Appendix D.

Colored MNIST: We classify MNIST digits from 5 classes $\mathcal{Y} = \{(0, 1), (2, 3), (4, 5), (6, 7), (8, 9)\}$, and treat color as the spurious attribute. For training data, we color 0.995 of each class's datapoints with color a, and color the rest randomly. Validation and test images are colored randomly.

Waterbirds (Sagawa et al., 2019): We classify $\mathcal{Y} = \{\text{waterbird, landbird}\}$ with background $\mathcal{A} = \{\text{water, land}\}$. 95% of images have the same bird and background type.

CelebA (Liu et al., 2015): We classify celebrities' hair color

Table 1. Worst-group and average accuracies. For worst-group accuracy, CNC obtains best results by a significant margin on image datasets, and near-SOTA on CivilComments. GEORGE results from (Sohoni et al., 2020). Other non-CNC results from (Liu et al., 2021).

Method	CMNIST		Waterbirds		CelebA		CivilComments	
Accuracy (%)	Worst-group	Avg.	Worst-group	Avg.	Worst-group	Avg.	Worst-group	Avg.
ERM	0.0	20.1	72.6	85.9	47.2	95.6	59.4	92.6
Joint DRO	0.0	20.1	69.5	88.5	74.4	82.4	56.6	92.5
LfF	0.0	21.2	75.2	91.6	70.6	86.0	58.8	92.5
GEORGE	0.2	22.2	83.8	90.5	54.9	94.5	-	-
JTT	74.5	90.2	86.0	90.3	81.1	88.0	**69.3**	91.1
CNC (Ours)	**77.4**	90.9	**89.0**	90.4	**88.8**	89.9	69.2	82.1
Group DRO	0.0	21.4	85.7	89.5	87.2	93.0	69.9	88.9

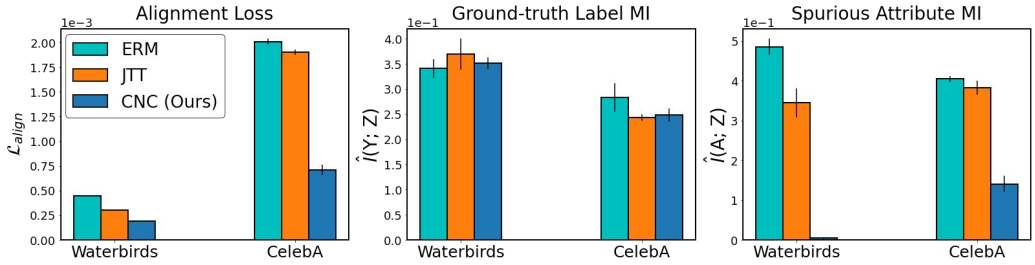


Figure 2. Comparing alignment loss and mutual information from models trained with ERM, JTT, and CNC on the Waterbirds and CelebA datasets. CNC again most effectively removes dependence on the spurious attribute.

$\mathcal{Y} = \{\text{blond, not blond}\}$ with $\mathcal{A} = \{\text{male, female}\}$. Only 6% of blond celebrities in the dataset are male.

CivilComments (Borkan et al., 2019): Given an online comment, we classify $\mathcal{Y} = \{\text{toxic, not toxic}\}$. $\mathcal{A}$ denotes whether the comment mentions one of eight demographic identities, as described in (Koh et al., 2021).

4.1. Main results

To study (1), we evaluate CNC on image classification and NLP datasets with spurious correlations. As baselines, we use standard ERM and an oracle GDRO approach that assumes access to the group labels. We also compare against recent methods that tackle spurious correlations without requiring group labels: joint DRO (Levy et al., 2020), GEORGE (Levy et al., 2020), Learning from Failure (LfF) (Levy et al., 2020), and Just Train Twice (JTT) (Liu et al., 2021). Results are reported in Table 1; CNC achieves highest worst-group accuracy among *all* methods on CMNIST, Waterbirds and CelebA, while also obtaining competitive (near-SOTA) worst-group accuracy on CivilComments.

While LFF, GEORGE, and JTT similarly rely on using trained ERM models to estimate group labels or spurious attribute values, CNC uniquely uses these ERM predictions to learn similar *representations* via contrastive learning. This may provide additional signal to ignore spurious attributes beyond the upsampling or group reweighting procedures in prior approaches. This comparative benefit may also hold even when subgroups are perfectly known—CNC outperforms the "oracle" GDRO method by 1.5% on average.

4.2. Effect of CNC on representation metrics

To study (2) and shed light on CNC's worst-group accuracy gains, we investigate if models trained with CNC actually learn representations with higher alignment. Compared to ERM and JTT (the next-best performing method that does not require subgroup labels), CNC achieves a significantly lower alignment loss and representations with lower mutual information to spurious attributes (while having comparable mutual information with the label), as depicted in Fig. 2.

We find that CNC representations exhibit the lowest alignment loss consistently for these datasets; this also corresponds to CNC models achieving the highest worst-group accuracy. Furthermore, while all methods result in representations that exhibit high mutual information with the ground-truth, suggesting that all models learn ground-truth dependencies, only CNC results in representations that drastically reduce mutual information with spurious attributes.

5. Conclusion

We present CNC, a two-stage approach to improve model robustness to spurious correlations using contrastive learning. We show that CNC learns representations that better depend only on ground-truth labels, and achieves SOTA or near-SOTA worst-group performance on several benchmarks. We also theoretically support CNC by relating worst-group generalization performance to its optimization objective.

References

Ahuja, K., Shanmugam, K., Varshney, K., and Dhurandhar, A. Invariant risk minimization games. In *International Conference on Machine Learning*, pp. 145–155. PMLR, 2020.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pp. 233–242. PMLR, 2017.

Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Balashankar, A., Lees, A., Welty, C., and Subramanian, L. What is fair? exploring Pareto-efficiency for fairness constrained classifiers. *arXiv preprint arXiv:1910.14120*, 2019.

Beery, S., Van Horn, G., and Perona, P. Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 456–473, 2018.

Ben-Tal, A., Den Hertog, D., De Waegenaere, A., Melenberg, B., and Rennen, G. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

Blodgett, S. L., Green, L., and O'Connor, B. Demographic dialectal variation in social media: A case study of african-american english. *arXiv preprint arXiv:1608.08868*, 2016.

Borkan, D., Dixon, L., Sorensen, J., Thain, N., and Vasserman, L. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pp. 491–500, 2019.

Buolamwini, J. and Gebru, T. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pp. 77–91. PMLR, 2018.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A., and Jegelka, S. Debiased contrastive learning. In *Advances in Neural Information Processing Systems*, volume abs/2007.00224, 2020.

Combes, R. T. d., Pezeshki, M., Shabanian, S., Courville, A., and Bengio, Y. On the learning dynamics of deep neural networks. *arXiv preprint arXiv:1809.06848*, 2018.

Creager, E., Jacobsen, J.-H., and Zemel, R. Environment inference for invariant learning. In *ICML Workshop on Uncertainty and Robustness*, 2020.

Curi, S., Levy, K. Y., Jegelka, S., and Krause, A. Adaptive sampling for stochastic risk-averse learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1036–1047, 2020.

Duchi, J. and Namkoong, H. Variance-based regularization with convex objectives. *The Journal of Machine Learning Research*, 20(1):2450–2504, 2019.

Duchi, J. C., Hashimoto, T., and Namkoong, H. Distributionally robust losses against mixture covariate shifts. *Operations Research*, 2019.

Goel, K., Gu, A., Li, Y., and Ré, C. Model patching: Closing the subgroup performance gap with data augmentation. In *International Conference on Learning Representations*, 2020.

Gunel, B., Du, J., Conneau, A., and Stoyanov, V. Supervised contrastive learning for pre-trained language model fine-tuning. In *International Conference on Learning Representations*, 2021.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010.

Hashimoto, T., Srivastava, M., Namkoong, H., and Liang, P. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pp. 1929–1938. PMLR, 2018.

Hassani, K. and Khasahmadi, A. H. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pp. 4116–4126. PMLR, 2020.

Hermann, K. L. and Lampinen, A. K. What shapes feature representations? exploring datasets, architectures, and training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Kaufman, S., Rosset, S., Perlich, C., and Stitelman, O. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):1–21, 2012.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673, 2020.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Beery, S., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. WILDS: A benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2021.07421*, 2021.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Priol, R. L., and Courville, A. Out-of-distribution generalization via risk extrapolation (rex). *arXiv preprint arXiv:2003.00688*, 2020.

Levy, D., Carmon, Y., Duchi, J. C., and Sidford, A. Large-scale methods for distributionally robust optimization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 8847–8860, 2020.

Liang, P. Cs229t/stat231: Statistical learning theory (winter 2016), 2016.

Liu, E., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. Just train twice: Improving group robustness without training group information. 2021.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Martinez, N., Bertran, M., and Sapiro, G. Minimax pareto fairness: A multi objective perspective. In *International Conference on Machine Learning (ICML)*, 2020.

Mnih, A. and Kavukcuoglu, K. Learning word embeddings efficiently with noise-contrastive estimation. In *NIPS*, 2013.

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. Learning from failure: De-biasing classifier from biased classifier. In *Advances in Neural Information Processing Systems*, volume 33, pp. 20673–20684, 2020.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Oren, Y., Sagawa, S., Hashimoto, T. B., and Liang, P. Distributionally robust language modeling. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Parascandolo, G., Neitz, A., Orvieto, A., Gresele, L., and Schölkopf, B. Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*, 2020.

Pezeshki, M., Kaba, S.-O., Bengio, Y., Courville, A., Precup, D., and Lajoie, G. Gradient starvation: A learning proclivity in neural networks. *arXiv preprint arXiv:2011.09468*, 2020.

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Robinson, J., Chuang, C.-Y., Sra, S., and Jegelka, S. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2019.

Sagawa, S., Raghunathan, A., Koh, P. W., and Liang, P. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pp. 8346–8356. PMLR, 2020.

Saunshi, N., Plevrakis, O., Arora, S., Khodak, M., and Khandeparkar, H. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, pp. 5628–5637. PMLR, 2019.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., and Levine, S. Time-contrastive networks: Self-supervised learning from video. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1134–1141, 2018.

Sohoni, N., Dunnmon, J., Angus, G., Gu, A., and Ré, C. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19339–19352, 2020.

Song, J. and Ermon, S. Multi-label contrastive predictive coding. In *Advances in Neural Information Processing Systems*, volume 33, pp. 8161–8173, 2020.

Telgarsky, M. Deep learning theory lecture notes, 2020.

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Wiesemann, W., Kuhn, D., and Sim, M. Distributionally robust convex optimization. *Operations Research*, 62(6): 1358–1376, 2014.

Wu, M., Mosse, M., Zhuang, C., Yamins, D., and Goodman, N. D. Conditional negative sampling for contrastive learning of visual representations. In *International Conference on Learning Representations*, 2021.

Zhang, Z. and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

A. Theorems and proofs

First, we show that if $\mathcal{L}_{align}(f_{\text{enc}})$ is small, then $\mathcal{L}_{\text{wg}}(f_\theta)$ is close to $\mathcal{L}_{\text{avg}}(f_\theta)$. We consider loss functions $\ell(\cdot)$ that are 1-Lipschitz such as the hinge loss and logistic loss (which we use in practice). The analysis is based on standard concentration bounds.

Theorem A.1. *In the setting described above, suppose that the model parameters θ satisfy the condition that for any two different groups $g_1 \in \mathcal{G}$ and $g_2 \in \mathcal{G}$ that have the same groundtruth label, the alignment loss is less than ε (cf. equation (3)): $\mathcal{L}_{align}(f_{enc}) \leq \varepsilon$. Suppose the linear classification layer W has bounded operator norm: $\|W\|_2 \leq B$. Let n_g denote the size of the group $g \in \mathcal{G}$ in the training data, and k denote $|\mathcal{G}|$ as before. Then, with probability at least $1-\delta$,*

$$\mathcal{L}_{wg}(f_\theta) \leq \mathcal{L}_{avg}(f_\theta) + B \cdot \varepsilon + \max_{g \in \mathcal{G}} \sqrt{\frac{8 \log(k/\delta)}{n_g}}. \tag{6}$$

Proof of Theorem A.1. Recall that our main assumption is that the feature representations between every pair of subgroups that have the same ground truth labels are approximately aligned (cf. Eq. (3)). Let $g_1 = (y, a_1)$ and $g_2 = (y, a_2)$ be any two different subgroups where $y \in \mathcal{Y}$ denotes a ground truth label and $a_1 \neq a_2$ denote spurious attributes. Let G_1 and G_2 denote the two subgroups in the training data that belong to subgroup g_1 and g_2, respectively. Let n_{g_1} and n_{g_2} denote the size of these two subgroups, respectively. Let $\mathcal{L}(\cdot)$ denote the population loss. Recall that f_{enc} represents the encoder layers of a full model f_θ. Because the classifier f_{cls} is often just a linear layer, for brevity in this section we denote it as W. Our alignment assumption implies that

$$\frac{1}{n_{g_1}} \frac{1}{n_{g_2}} \sum_{x \in G_1} \sum_{x' \in G_2} \|f_{\text{enc}}(x) - f_{\text{enc}}(x')\|_2^2 \leq \varepsilon. \tag{7}$$

Next, we bound the difference between the population loss of g_1 and the population loss of g_2:

$$\Delta(g_1, g_2) = \left| \mathop{\mathbb{E}}_{(x,y,a_1) \sim \mathcal{P}_{g_1}} [\mathcal{L}(W f_{\text{enc}}(x), y] - \mathop{\mathbb{E}}_{(x,y,a_2) \sim \mathcal{P}_{g_2}} [\mathcal{L}(W f_{\text{enc}}(x), y)] \right|.$$

It is not hard to verify that

$$\mathcal{L}_{\text{wg}}(f_\theta) \leq \mathcal{L}_{\text{avg}}(f_\theta) + \max_{g_1, g_2 \in \mathcal{G}: g_1 \neq g_2} \Delta(g_1, g_2).$$

Hence, we focus on the function $\Delta(g_1, g_2)$ for the rest of the proof. By standard concentration bounds, the following result holds with probability at least $1-\delta$ for all $\binom{k}{2}$ pairs of subgroups $g_1 \in \mathcal{G}$ and $g_2 \in \mathcal{G}$,

$$\left| \mathop{\mathbb{E}}_{(x,y,a_1) \sim \mathcal{P}_{g_1}} [\mathcal{L}(W f_{\text{enc}}(x), y)] - \frac{1}{n_{g_1}} \sum_{(x,y,a_1) \in G_1} \mathcal{L}(W f_{\text{enc}}(x), y) \right| \lesssim \sqrt{\frac{2 \log(k/\delta)}{n_{g_1}}}. \tag{8}$$

Thus, with probability at least $1-\delta$, the following holds:

$$\begin{aligned} \Delta(g_1, g_2) \leq & \left| \frac{1}{n_{g_1}} \sum_{(x,y,a_1) \in G_1} \mathcal{L}(W f_{\text{enc}}(x), y) - \frac{1}{n_{g_2}} \sum_{(x,y,a_2) \in G_2} \mathcal{L}(W f_{\text{enc}}(x), y) \right| \\ & + \left(\sqrt{\frac{2 \log(k/\delta)}{n_{g_1}}} + \sqrt{\frac{2 \log(k/\delta)}{n_{g_2}}} \right). \end{aligned} \tag{9}$$

Next, we focus on the RHS of equation (9). First, equation (9) is also equal to the following:

$$\left| \frac{1}{n_{g_1}} \frac{1}{n_{g_2}} \sum_{(x,y,a_1) \in G_1} \sum_{(x',y,a_2) \in G_2} \mathcal{L}(W f_{\text{enc}}(x), y)) - \frac{1}{n_{g_1}} \frac{1}{n_{g_2}} \sum_{(x,y,a_1) \in G_1} \sum_{(x',y,a_2) \in G_2} \mathcal{L}(W f_{\text{enc}}(x'), y)) \right|.$$

Since $\mathcal{L}(\cdot)$ is 1-Lipschitz, the above is at most:

$$\begin{aligned}
&\left|\frac{1}{n_{g_1}n_{g_2}}\sum_{(x,y,a_1)\in G_1}\sum_{(x,y,a_2)\in G_2}\|Wf_{\text{enc}}(x)-Wf_{\text{enc}}(x')\|_2\right| \\
\leq&\frac{B}{n_{g_1}n_{g_2}}\sum_{(x,y,a_1)\in G_1}\sum_{(x',y,a_2)\in G_2}\|f_{\text{enc}}(x)-f_{\text{enc}}(x')\|_2 \\
\leq&B\varepsilon,
\end{aligned}$$

where the last step is because of equation (7). Thus, we have shown that

$$\Delta(g_1,g_2)\leq B\varepsilon+\left(\sqrt{\frac{2\log(k/\delta)}{n_{g_1}}}+\sqrt{\frac{2\log(k/\delta)}{n_{g_2}}}\right).$$

Since the above result holds any two different subgroups in $\mathcal{G}$, we conclude that

$$\mathcal{L}_{\text{wg}}(f_\theta)\leq\mathcal{L}_{\text{avg}}(f_\theta)+B\cdot\varepsilon+\max_{g\in\mathcal{G}}\sqrt{\frac{8\log(k/\delta)}{n_g}}.$$

□

Second, we show that provided the full loss of equation (5) can be minimized efficiently, the average error of its minimizer will be small plus a generalization error term that scales down with the number of training samples. Let $\hat{f}$ be the minimizer of the full (training) loss $\hat{\mathcal{L}}$ within a function class $\mathcal{F}$. Let $\mathcal{R}(\mathcal{F})$ denote the (unnormalized) Rademacher complexity of $\mathcal{F}$ (cf. (Telgarsky, 2020; Liang, 2016)).

Theorem A.2. *Let $\hat{f}$ be the minimizer of the empirical full loss $\hat{\mathcal{L}}$ within a function class $\mathcal{F}$. Let $p_{\min}$ denote the minimum over any class of $\mathcal{Y}$ that a sample from $\mathcal{P}$ belongs to the group. Then, with probability at least $1-\delta$ over the randomness of the training samples, the following holds:*

$$\mathcal{L}_{avg}(\hat{f})\leq C\cdot\mathcal{L}(\hat{f})+E, \tag{10}$$

where $C=\left(1-|\mathcal{Y}|\cdot(1-p_{\min})^N\right)^{-1}$, $E=O\left(\frac{\mathcal{R}(\mathcal{F})}{n}+\sqrt{\frac{\log\delta^{-1}}{n}}\right)$, and $\mathcal{L}$ is the population version of equation (5).

Note that C decreases to one and E decrease to zero as the number of negative samples and training samples both increase.

Proof of Theorem A.2. We first recall the following notations. Let $f_{\hat{\theta}}$ be the minimizer of the empirical loss $\hat{\mathcal{L}}(f)$ in a function class $\mathcal{F}$. Let n be the size of the training dataset. $\mathcal{Y}$ is the set of all possible class labels. The first step is to bound the generalization error between the empirical loss $\hat{\mathcal{L}}(f_{\hat{\theta}})$ and the population loss $\mathcal{L}(f_{\hat{\theta}})$. Using standard uniform convergence techniques, we show that for any $f\in\mathcal{F}$,

$$\left|\hat{\mathcal{L}}(f)-\mathcal{L}(f)\right|\lesssim\frac{\mathcal{R}(\mathcal{F})}{n}+\sqrt{\frac{\log\frac{1}{\delta}}{n}}. \tag{11}$$

In particular, equation (11) implies that the generalization error between $\hat{\mathcal{L}}(\hat{f})$ and $\mathcal{L}(f)$ is small. The idea for showing equation (11) is based on the Lipschitz composition property of Rademacher complexity. Let $\ell(\cdot)$ denote the contrastive loss. Let $\mathcal{L}_{\text{con}}$ denote the population loss that corresponds to the contrastive estimation loss in $\mathcal{L}$. It is not hard to verify that $\ell(\cdot)$ is bounded from above by $C_1=\log(1+\exp(\tau)N)$. By uniform convergence, we have that with probability at least $1-\delta$, the following holds (e.g. Theorem 9 in Liang (2016))

$$\left|\hat{\mathcal{L}}_{\text{con}}(f)-\mathcal{L}_{\text{con}}(f)\right|\leq\frac{2\mathcal{R}(\ell\circ\mathcal{F})}{n}+C_1\sqrt{\frac{2\log\frac{2}{\delta}}{n}},\text{ for any }f\in\mathcal{F}.$$

Next, we deal with the composition of $\ell(\cdot)$ over $\mathcal{F}$ above. Since $\log(z+x)$ is 1-Lipschitz for any $z > 0$, one can expand out $\mathcal{R}(\ell \circ \mathcal{F})$ over every negative sample. Then, since $\exp(\cdot)$ is $N\exp(\tau)$-Lipschitz for any positive x^+ and negative samples $\{x_{i,j}^-\}_{j=1}^N$, we have that $\mathcal{R}(\ell \circ \mathcal{F}) \leq N\exp(\tau)\mathcal{R}_M(\mathcal{F})$. One can similarly apply Rademacher complexity over the cross-entropy loss portion of $\mathcal{L}$, since the operator norm of the linear classification layer is bounded from above by a fixed constant. We therefore conclude that equation (11) holds.

The next step is to show an upper bound on the average loss of $f_{\hat{\theta}}$ using the contrastive loss. The idea for showing this step is based on the technique of Saunshi et al. (Saunshi et al., 2019). For simplicity, we present the analysis for the case when $M = 1$ to show the following result:

$$\mathcal{L}_{\text{avg}}(f_{\hat{\theta}}) \leq C\mathcal{L}_{\text{con}}(f_{\hat{\theta}}). \tag{12}$$

Here $f_{\hat{\theta}}$ is the minimizer of the empirical contrastive loss, i.e. an encoder function. $\mathcal{L}_{\text{avg}}(f_{\hat{\theta}})$ is the average population loss computed with the fine-tuned full model. Provided with this result, we then have that $\mathcal{L}(f_{\hat{\theta}}) \geq \mathcal{L}_{\text{avg}}(f_{\hat{\theta}})$, since the cross-entropy loss of $\hat{f}$ is equal to the average loss of $f_{\hat{\theta}}$. Therefore, we conclude that

$$\mathcal{L}_{\text{avg}}(f_{\hat{\theta}}) \leq C\mathcal{L}(f_{\hat{\theta}}) + O\left(\frac{\mathcal{R}(\mathcal{F})}{n} + \sqrt{\frac{\log(1/\delta)}{n}}\right).$$

Thus, for the rest of the proof, we focus on showing equation (12). Consider one set of contrastive sample $S = (x, x^+, \{x_j\}_{j=1}^N)$. We can write the contrastive loss as

$$\mathcal{L}_{\text{con}}(f_{\hat{\theta}}) = \mathop{\mathbb{E}}_{S}\left[\log\left(1 + \sum_{j=1}^{N} \exp\left(f_{\hat{\theta}}(x)^\top\left(f_{\hat{\theta}}(x_j^-) - f_{\hat{\theta}}(x^+)\right)\right)\right)\right].$$

Let $C(x)$ denote the true label of any datapoint x. Since $\log(\cdot)$ is a non-decreasing function, we can thus eliminate any negative sample that may have the same label as the anchor:

$$\mathcal{L}_{\text{con}}(f_{\hat{\theta}}) \geq \mathop{\mathbb{E}}_{S}\left[\log\left(1 + \sum_{C(x_j^-)\neq C(x)} \exp\left(f_{\hat{\theta}}(x)^\top\left(f_{\hat{\theta}}(x_j^-) - f_{\hat{\theta}}(x^+)\right)\right)\right)\right].$$

An equivalent way to calculate the above expectation over S is by first conditioning on the labels of every x_j^- as Y_j, and then sample x_j^- conditional on having label Y_j. Let $Y = [Y_1, Y_2, \ldots, Y_N]$ denote the label vector. Since the logistic loss is convex, we can use Jensen's inequality to push the expectation over x_j^- (conditional on having label Y_j) inside the expectation:

$$\mathcal{L}_{\text{con}}(f_{\hat{\theta}}) \geq \mathop{\mathbb{E}}_{x,Y}\left[\log\left(1 + \sum_{Y_i\neq C(x)} \exp\left(f_{\hat{\theta}}(x)^\top\left(\mathbb{E}\left[f_{\hat{\theta}}(x_j^-)\right] - \mathbb{E}\left[f_{\hat{\theta}}(x^+)\right]\right)\right)\right)\right].$$

Let $\mu_i = \mathbb{E}\left[f_{\hat{\theta}}(x)\right]$ for x sampled from class $i \in \mathcal{Y}$. When Y contains the entire class $\mathcal{Y}$, the above is at least

$$\mathop{\mathbb{E}}_{x}\left[\log\left(1 + \sum_{c\neq C(x)} \exp\left(f_{\hat{\theta}}(x)^\top\left(\mu_{C(x)} - \mu_c\right)\right)\right)\right] \geq \mathcal{L}_{\text{avg}}(f_{\hat{\theta}}).$$

The probability that Y contains the entire class is at least:

$$1 - \sum_{i=1}^{|\mathcal{Y}|}(1-p_i)^N \geq 1 - |\mathcal{Y}| \cdot (1-p_{\min})^N.$$

The above is equal to C^{-1}. Thus, we conclude that equation (12) is true. □

B. Contrastive algorithm design details

In this section, we provide further details on the training setup and contrastive batch sampling, pseudocode, and additional properties related to CNC's implementation.

B.1. Training setup

In Fig. 3, we illustrate the two training stages of Correct-N-Contrast described in Sec. 3.1. In Stage 1, we first train an ERM model with a cross-entropy loss. For consistency with Stage 2, we depict the output as a composition of the encoder and linear classifier layers. Then in Stage 2, we train a new model with the same architecture using contrastive batches sampled with the Stage 1 ERM model and a supervised contrastive loss (4) (which we compute after the depicted representations are first normalized) to update the encoder layers. Note that unlike prior work in contrastive learning (Chen et al., 2020; Khosla et al., 2020), as we have the ground-truth labels of the anchors, positives, and negatives, we also continue forward-passing the unnormalized representations (encoder layer outputs) and compute a cross-entropy loss to update the classifier layers while jointly training the encoder layers as well.

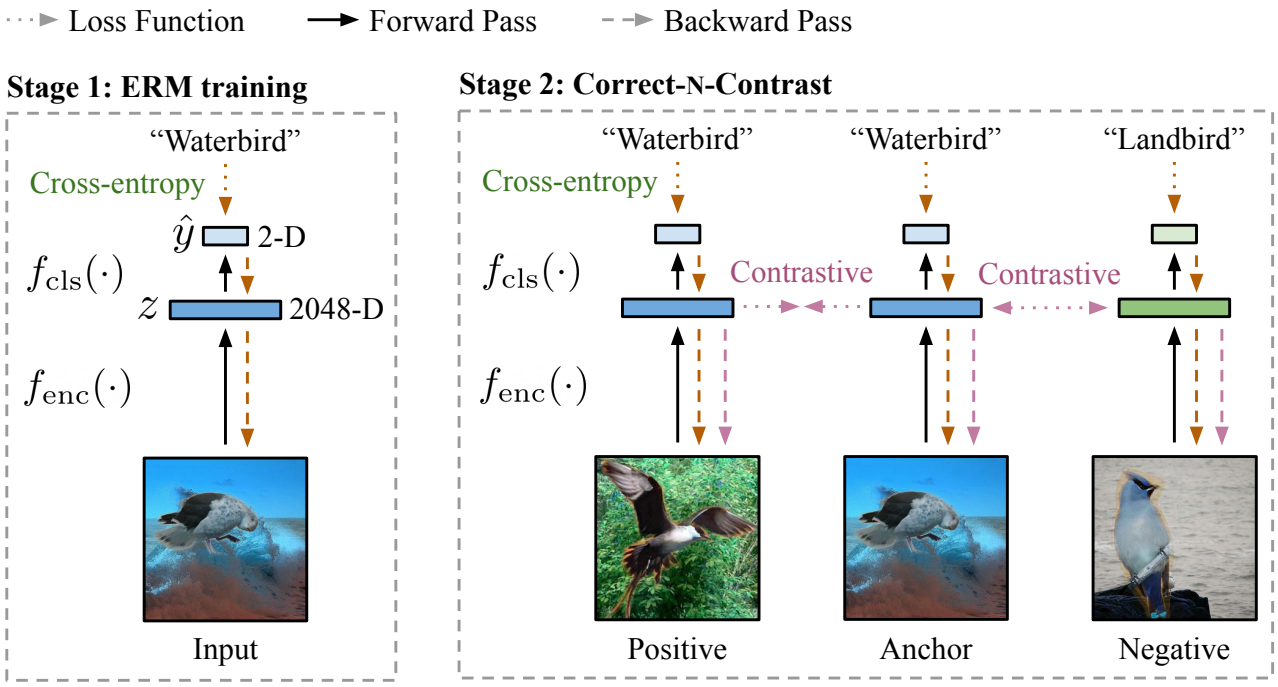


Figure 3. The two stages of Correct-N-Contrast. In Stage 1, we train a model with standard ERM and a cross-entropy loss. Then in Stage 2, we train a new model with the same architecture, but specifically learn spurious-attribute-invariant representations with a contrastive loss (4) and batches of anchors, positives, and negatives sampled with the ERM model's predictions. We also update the full model jointly with a cross-entropy loss on the classifier layer output and the input ground-truth labels. Dimensions for ResNet-50 and Waterbirds.

We also note that unlike prior work, we wish to learn invariances between anchors and positives that maximally reduce the presence of features not needed for classification. We thus do not pass the representations through an additional *projection network* (Chen et al., 2020). Instead, we use Eq. 4 to compute the supervised contrastive loss directly on the encoder outputs $z = f_{\text{enc}}(x)$. In Appendix E.3.2, we study ablations with both design choices.

B.2. Two-sided contrastive batch implementation

We provide more details on our default contrastive batch sampling approach described in Sec. 3.1. To recall, we can incorporate additional anchor-positive and anchor-negative comparisons as in prior contrastive learning work (Chen et al., 2020) by switching the role of the anchor and first positive sampled in a contrastive batch. However, with "one-sided" batch $\{x_i\}, \{x_m^+\}_{m=1}^M, \{x_n^-\}_{n=1}^N$, naively doing so results in the batch $\{x_1^+\}, \{x_i\}, \{x_n^-\}_{n=1}^N$. This introduces "easy" negative

comparisons where x_1^+ and x_n^- differ in both spurious attribute and ground-truth, which may encourage learning spurious correlations. To avoid this, given original anchor x_i and a randomly sampled positive x_1^+, we augment our batch by sampling $M-1$ positives from $\{x_j \in X : \hat{y}_j \neq \hat{y}_i, y_j = y_i\}$ and N negatives from $\{x_{n'} \in X : \hat{y}_{n'}^- = \hat{y}_1^+, y_{n'}^- \neq y_1^+\}$. In other words, we sample additional positives and negatives using the same guidelines as before, but adjust for the "new" anchor. We call this "two-sided" in contrast to the "one-sided" comparisons with just the original anchor, positives, and negatives.

Implementing this sampling procedure in practice is simple. First, recall our initial setup with trained ERM model $f_{\hat{\theta}}$, its predictions $\{\hat{y}_i\}_{i=1}^n$ on training data $\{(x_i, y_i)\}_{i=1}^n$ (where $\hat{y}_i = f_{\hat{\theta}}(x_i)$), and number of positives and negatives to sample M and N. We then sample batches with Algorithm 1.

Algorithm 1 Sampling two-sided contrastive batches

Require: Number of positives M and number of negatives N to sample for each batch.

Initialize set of contrastive batches $B = \{\}$
for each $x_i \in \{x_i \in X : \hat{y}_i = y_i\}$ **do**
  Sample $M-1$ additional "anchors" to obtain $\{x_i\}_{i=1}^M$ from $\{x_i \in X : \hat{y}_i = y_i\}$
  Sample M positives $\{x_m^+\}_{m=1}^M$ from $\{x_m^- \in X : \hat{y}_m^- = \hat{y}_i,\ y_m^- \neq y_i\}$
  Sample N negatives $\{x_n^-\}_{n=1}^N$ from $\{x_n^- \in X : \hat{y}_n^- = \hat{y}_i,\ y_n^- \neq y_i\}$
  Sample N negatives $\{x_n'^-\}_{n=1}^N$ from $\{x_n'^- \in X : \hat{y}_n'^- = \hat{y}_1^+, y_n'^- \neq y_1^+\}$
  Update contrastive batch set: $B \leftarrow B \cup \left(\{x_i\}_{i=1}^M, \{x_m^+\}_{m=1}^M, \{x_n^-\}_{n=1}^N, \{x_n'^-\}_{n=1}^N\right)$
end for

Because the initial anchors are then datapoints that the ERM model gets correct, under our heuristic we infer $\{x_i\}_{i=1}^M$ as samples from the majority group. Similarly the M positives $\{x_m^+\}_{m=1}^M$ and N negatives $\{x_n^-\}_{n=1}^N$ that it gets incorrect are inferred to belong to minority groups.

For one batch, we then compute the full contrastive loss with

$$\hat{\mathcal{L}}_{\text{con}}^{\text{sup}}(f_{\text{enc}}) = \hat{\mathcal{L}}_{\text{con}}^{\text{sup}}\left(x_1, \{x_m^+\}_{m=1}^M, \{x_n^-\}_{n=1}^N; f_{\text{enc}}\right) + \hat{\mathcal{L}}_{\text{con}}^{\text{sup}}\left(x_1^+, \{x_i\}_{i=1}^M, \{x_n'^-\}_{n=1}^N; f_{\text{enc}}\right) \tag{13}$$

where e.g. $\hat{\mathcal{L}}_{\text{con}}^{\text{sup}}\left(x_1, \{x_m^+\}_{m=1}^M, \{x_n^-\}_{n=1}^N; f_{\text{enc}}\right)$ is given by:

$$-\frac{1}{M}\sum_{m=1}^{M} \log \frac{\exp(z_1^\top z_m^+/\tau)}{\sum_{m=1}^M \exp(z_1^\top z_m^+/\tau) + \sum_{n=1}^N \exp(z_1^\top z_n^+/\tau)} \tag{14}$$

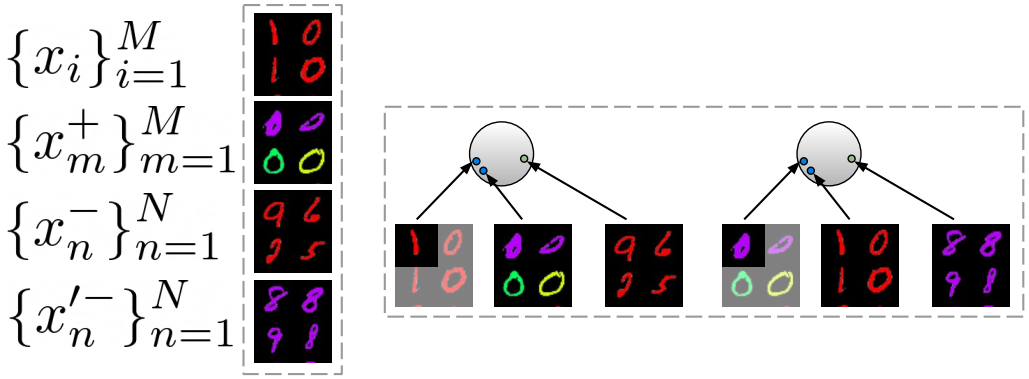
and again let z be the normalized output $f_{\text{enc}}(x)$ for corresponding x. We compute the cross-entropy component of the full loss for each x in the two-sided batch with its corresponding label y.

B.3. Summary of CNC design choices and properties

We summarize CNC's design choices, additional properties, and differences from standard supervised contrastive learning below. In Appendix E.3, we empirically validate each component.

No projection network. As we wish to learn data representations that maximize the alignment between anchor and positive datapoints, we do not compute the contrastive loss with the outputs of an additional nonlinear projection network. This is inspired by the logic justifying a projection head in prior contrastive learning, e.g. SimCLR (Chen et al., 2020), where the head is included because the contrastive loss trains representations to be "invariant to data transformation" and may encourage removing information "such as the color or orientation of objects". In our case, we view inferred datapoints with the same class but different spurious attributes as "transformations" of each other, and we hypothesize that maximally removing these differences can help us improve worst-group performance.

Two-sided contrastive sampling. To incorporate additional comparisons between datapoints that only differ in spurious attribute during training, we employ "two-sided" contrastive batch sampling. This lets us equally incorporate instances where the second contrastive model in CNC treats datapoints that the initial ERM model got incorrect and correct as anchors.



(a) Contrastive Batch (b) Two-sided Update

Figure 4. Illustration of two-sided contrastive batch sampling with Colored MNIST as an example. From a single batch (a), we can train a contrastive model with two anchor-positive-negative pairings (b). Aside from increasing the total number of "hard negatives" considered for each anchor-positive pair, this intuitively "pushes" together anchors and positives from two different directions for greater class separation.

Additional intrinsic hard positive/negative mining. Because the new model corrects for potentially learned spurious correlations by only comparing and contrasting datapoints that differ in ground-truth or spurious attribute, but not both (as dictated by the initial ERM model's outputs), the contrastive batches naturally carry "hard" positives and negatives. Thus, our approach provides a natural form of hard negative mining (in addition to the intrinsic hard positive / negative mining at the gradient level with InfoNCE-style contrastive losses (Chen et al., 2020; Khosla et al., 2020)) while avoiding class collisions, two nontrivial challenges in standard self-supervised contrastive learning (Robinson et al., 2021; Wu et al., 2021; Chuang et al., 2020).

C. Further related work discussion

We provide additional discussion of related work and connections to our work below.

C.1. Improving robustness to spurious correlations

Our core objective is to improve model robustness to group or subpopulation distribution shifts that arise from the presence of spurious correlations, specifically for classification tasks. Because these learnable correlations hold for some but not all samples in a dataset, standard training with ERM may result in highly variable performance: a model that learns to classify datapoints based on these spurious correlations does well for some subsets or "groups" of the data but not others. To improve model robustness and avoid learning spurious correlations, prior work introduces the goal to maximize the worst-group accuracy (Sagawa et al., 2019). Related works broadly fall under two categories:

Improving robustness with group information. If information such as spurious attribute labels is provided, one can divide the data into explicit groups as defined in Sec. 2, and then train to directly minimize the worst group-level error among these groups. This is done in group DRO (GDRO) (Sagawa et al., 2019), where the authors propose an online training algorithm that focuses training updates over datapoints from higher-loss groups. Goel et al. (2020) also adopt this approach with their method CycleGAN Augmented Model Patching (CAMEL). However, similar to our motivation, they argue that a stronger modeling goal should be placed on preventing a model from learning group-specific features. Their approach involves first training a CycleCAN (Zhu et al., 2017) to learn the data transformations from datapoints in one group to another that share the same ground-truth label. They then apply these transformations as data augmentations to different samples, intuitively generating new versions of the original samples that take on group-specific features. Finally they train a new model with a consistency regularization objective to learn invariant features between transformed samples and their sources. Unlike their consistency loss, we accomplish a similar objective to learn group-invariant features with contrastive learning. Our first training stage is also less expensive. Instead of training a CycleGAN and then using it to augment datapoints, we train a relatively simple standard ERM classification model, sometimes with only a few number of epochs, and use its predictions to identify pairs of datapoints to serve a similar purpose. Finally, unlike both CAMEL and GDRO, we do not require spurious attribute or group labels for each training datapoints. We can then apply CNC in less restrictive settings

where such information is not known.

Related to GDRO are methods that aim to optimize a "Pareto-fair" objective, more general than simply the worst-case group performance. Notable examples are the works of Balashankar et al. (2019) and Martinez et al. (2020). However, these approaches similarly do not directly optimize for good representation alignment (unlike our work).

Improving robustness without training group information. More similar to our approach are methods that do not assume group information at training time, and only require validation set spurious attribute labels for fine-tuning. As validation sets are typically much smaller in size than training sets, an advantage of CNC and comparable methods is that we can improve the accessibility of robust training methods to a wider set of problems. One popular line of work is distributionally robust optimization (DRO), which trains models to minimize the worst loss within a ball centered around the observed distribution (Ben-Tal et al., 2013; Wiesemann et al., 2014; Duchi & Namkoong, 2019; Levy et al., 2020; Curi et al., 2020; Oren et al., 2019). This includes the joint DRO (Levy et al., 2020) method we evaluate against. However, prior work has shown that these approaches may be too pessimistic, optimizing not just for worst-group accuracy but worst possible accuracy within the distribution balls (Sagawa et al., 2019), or too undirected, optimizing for too many subpopulations, e.g. by first upweighting minority points but then upweighting majority points in later stages of training (Liu et al., 2021). Pezeshki et al. (2020) instead suggest that *gradient starvation* (GS), where neural networks only learn to capture statistically dominant features in the data (Combes et al., 2018), is the main culprit behind learning spurious correlations, and introduce a "spectral decoupling" regularizer to alleviate GS. However this does not prevent models from learning spurious correlations.

Most similar to our approach are methods that first train an initial ERM model with ground-truth labels as a way to identify datapoints belonging to minority groups, and subsequently train an additional model with greater emphasis on the estimated minority groups. Sohoni et al. (2020) demonstrate that even when only trained on the ground-truth labels, neural networks learn feature representations that can be clustered into groups of data exhibiting different spurious attributes, and thus estimate the subgroup labels before running GDRO. Meanwhile, Nam et al. (2020) train a pair of models, where one model minimizes a generalized cross-entropy loss (Zhang & Sabuncu, 2018), such that the datapoints this model classifies incorrectly largely correspond to those in the minority group. They then train the other model on the same data but upweight the minority-group-estimated points. While they interweave training of the biased and robust model, Liu et al. (2021) instead train one model first with a shortened training time but no modified cross-entropy objective, and show that then upsampling the incorrect datapoints and training another model with ERM is sufficient for higher worst-group accuracy.

We extend this line of work by demonstrating that the ERM model's predictions can be leveraged to not only estimate groups and train a new model with supervised learning but with different weightings. Instead, we can specifically identify pairs of points that a contrastive model can then learn invariant features between. Our core contribution comes from rethinking the objective with a contrastive loss that more directly reduces the model's ability to learning spurious correlations.

C.2. Contrastive learning

Our method also uses contrastive learning, a simple yet powerful framework for both self-supervised (Chen et al., 2020; Oord et al., 2018; Tian et al., 2019; Song & Ermon, 2020; Sermanet et al., 2018; Hassani & Khasahmadi, 2020; Robinson et al., 2021) and supervised (Khosla et al., 2020; Gunel et al., 2021) representation learning. The core idea is to learn data representations that maximize the similarity between a given input "anchor" and distinct views depicting the input again in some way ("positives"). Frequently this also involves *contrasting* positives with "negative" inputs sampled from the data, but without any assumed relation to the anchor (Bachman et al., 2019). Core components then include some way to source multiple views, e.g. with data transformations (Chen et al., 2020), and training objectives similar to noise contrastive estimation (Gutmann & Hyvärinen, 2010; Mnih & Kavukcuoglu, 2013).

Two particularly relevant criteria that benefit contrastive learning are gathering appropriate positives and negatives. For sampling positives, Chen et al. (2020) show that certain data augmentations (e.g. crops and cutouts) may be more beneficial than others (e.g. gaussian noise and sobel filtering) when generating anchors and positives for unsupervised contrastive learning. Tian et al. (2020) further study what makes good views for contrastive learning. They propose an "InfoMin principle", where anchors and positives should share the least information necessary for the contrastive model to do well on the downstream task. For sampling negatives, Robinson et al. (2021) show that contrastive learning also benefits from using "hard" negatives, which are both actually a different class from the anchor (which they approximate in the unsupervised setting) and embed closest to the anchor under the contrastive model's current data representation. Both of these approaches capture the principle that if positives are always too similar to the anchor and negatives are always too different, then contrastive learning may be inefficient at learning generalizable representations of the underlying classes.

In our work, we incorporate this principle by sampling datapoints with the same ground-truth but different ERM predictions–presumably because of spurious attribute differences–as anchor and positive views, while sampling negatives from datapoints with different ground-truth but the same ERM prediction as the anchor. The anchors and positives are different enough that a trained ERM model predicted them differently, while the anchors and negatives are similar enough that the trained ERM model predicted them the same. Contrasting the above then allows us to exploit both "hard" positive and negative criteria for our downstream classification task. In Appendix E.3.1, we show that removing this ERM-guided sampling and only sampling positives and negatives based on class labels leads to substantially lower worst-group accuracy.

C.3. Learning invariant representations

Finally, our work is also similar in motivation to Invariant Risk Minimization (IRM) (Arjovsky et al., 2019) and other related works in domain-invariant learning (Krueger et al., 2020; Parascandolo et al., 2020; Ahuja et al., 2020; Creager et al., 2020). These methods aim to train models that learn a single invariant representation that is consistently optimal (e.g. with respect to classifying data) across different domains or environments. These environments can be thought of as data groups, and while traditionally methods such as IRM require that environment labels are known, recent approaches such as Environment Inference for Invariant Learning (EIIL) (Creager et al., 2020) similarly aim to infer environments with an initial ERM model. However, their main goal for learning these invariances is to extrapolate to out-of-domain distribution shifts not seen during training, as opposed to improving worst-group performance with group-shifts exacerbated by rare groups in our setting. Thus they may not perform as well as the previously introduced methods for improving worst-group accuracy. In Appendix E.4, we report the worst-group accuracy obtained with EIIL on Waterbirds (Creager et al., 2020). The results support this conjecture, especially when only a small number of examples exists in an underperforming minority subgroup.

D. Additional experimental details

For all methods, we report the test set worst-group and average accuracies from models selected through hyperparameter tuning for the best validation set worst-group accuracy.

D.1. Dataset details

Colored MNIST. We evaluate with a version of the Colored MNIST dataset proposed in Arjovsky et al. (2019). The goal is to classify MNIST digits belonging to one of 5 classes $\mathcal{Y} = \{(0, 1), (2, 3), (4, 5), (6, 7), (8, 9)\}$, and treat color as the spurious attribute. In the training data, we color p_{corr} of each class's datapoints with an associated color a, and color the rest randomly. If p_{corr} is high, trained ERM models fail to classify digits that are not the associated color. We pick a from uniformly interspersed intervals of the `hsv` colormap, e.g. 0 and 1 digits may be spurious correlated with the color red (`#ff0000`), while 8 and 9 digits may be spuriously correlated with purple (`#ff0018`). The full set of colors in class order are $\mathcal{A} = \{$`#ff0000`, `#85ff00`, `#00fff3`, `#6e00ff`, `#ff0018`$\}$ (see Fig. **??**). For validation and test data, we color each datapoint randomly with a color $a \in \mathcal{A}$. We use the default test set from MNIST, and allocate 80%-20% of the default MNIST training set to the training and validation sets. For main results, we set $p_{\text{corr}} = 0.995$.

Waterbirds. We evaluate with the Waterbirds dataset, which was introduced as a standard spurious correlations benchmark in Sagawa et al. (2019). In this dataset, masked out images of birds from the CUB dataset (Wah et al., 2011) are pasted on backgrounds from the Places dataset (Zhou et al., 2017). Bird images are labeled either as waterbirds or landbirds, while the background either depicts water or land. From CUB, waterbirds consist of seabirds (ablatross, auklet, cormorant, frigatebird, fulmar, gull, jaeger, kittiwake, pelican, puffin, tern) and waterfowl (gadwell, grebe, mallard, merganser, guillemot, Pacific loon). All other birds are labeled as landbirds. From Places, water backgrounds consist of ocean and natural lake classes, while land backgrounds consist of bamboo forest and broadleaf forest classes.

The goal is to classify the foreground bird as $\mathcal{Y} = \{$waterbird, landbird$\}$, where there is spurious background attribute $\mathcal{A} = \{$water background, land background$\}$. We use the default training, validation, and test splits (Sagawa et al., 2019), where in the training data 95% of waterbirds appear with water backgrounds and 95% of landbirds appear with land backgrounds. Trained ERM models then have trouble classifying waterbirds with land backgrounds and landbirds with water backgrounds. For validation and test sets, water and land backgrounds are evenly split among landbirds and waterbirds.

CelebA. We evaluate with the CelebA spurious correlations benchmark introduced in Sagawa et al. (2019). The goal is to classify celebrities' hair color $\mathcal{Y} = \{$blond, not blond$\}$, which is spuriously correlated with the celebrity's identified gender $\mathcal{A} = \{$male, female$\}$. We use the same training, validation, test splits as in Sagawa et al. (2019). Only 6% of blond

celebrities are male; trained ERM models perform poorly on this group.

CivilComments. We evaluate with the CivilComments-WILDS dataset from Koh et al. (2021), derived from the Jigsaw dataset from Borkan et al. (2019). Each datapoint is a real online comment curated from the Civil Comments platform, a commenting plugin for independent news sites. For classes, each comment is labeled as either toxic or not toxic. For spurious attributes, each comment is also labeled with the demographic identities {male, female, LGBTQ, Christian, Muslim, other religions, Black, White} mentioned; multiple identities may be mentioned per comment.

The goal is to classify the comment $\mathcal{Y} = \{\text{toxic, not toxic}\}$. As in Koh et al. (2021), we evaluate with $\mathcal{A} = \{\text{male, female, LGBTQ, Christian, Muslim, other religions, Black, White}\}$. There are then 16 total groups corresponding to (toxic, identity) and (not toxic, identity) for each identity. Groups may overlap; a datapoint falls in a group if it mentions the identity. We use the default data splits (Koh et al., 2021). In Table 2, we list the percentage of toxic comments for each identity based on the groups. Trained ERM models in particular perform less well on the rarer toxic groups.

Table 2. Percent of toxic comments for each identity in the CivilComments-WILDS training set.

Identity	male	female	LGBTQ	Christian	Muslim	other religions	Black	White
% toxic	14.9	13.7	26.9	9.1	22.4	15.3	31.4	28.0

D.2. Methods details

D.2.1. Reported metrics

Main results. For the Colored MNIST, Waterbirds, and CelebA datasets, we run CNC with three different seeds, and report the average worst-group accuracy over these three trials in Table 1. As we use the same baselines and comparable methods as Liu et al. (2021), we referenced their main results for the reported numbers, which did not have standard deviations or error bars reported. For CivilComments-WILDS, due to time and compute constraints we only reported one run.

Estimated mutual information. We give further details for calculating the representation metric introduced in Sec. **??**. As a reminder, we report both alignment and estimated mutual information metrics to quantify how dependent a model's learned representations are on the ground-truth versus the spurious attributes, and compute both metrics on the representations $Z = \{f_{\text{enc}}(x)\}$ over all test set datapoints x. Then to supplement the alignment loss calculation in Sec. **??**, we also estimate $I(Y; Z)$ and $I(A; Z)$, the mutual information between the model's data representations and the ground-truth labels and spurious attribute labels respectively.

To first estimate mutual information with Y, we first approximate $p(y \mid z)$ by fitting a multinomial logistic regression model over all representations Z to classify y. With the empirical ground-truth label distribution $p(y)$, we compute:

$$\hat{I}(Y; Z) = \frac{1}{|Z|} \sum_{z \in Z} \sum_{y \in Y} p(y \mid z) \log \frac{p(y \mid z)}{p(y)} \tag{15}$$

We do the same but substitute the spurious attributes a for y to compute $\hat{I}(A; Z)$.

D.2.2. Stage 1 ERM training details

We describe the model selection criterion, architecture, and training hyperparameters for the initial ERM model in our method. To select this model, recall that we first train an ERM model to predict ground-truth labels, as the model may also learn dependencies on the spurious attributes. Because we then use the model's predictions on the training data to infer samples with different spurious attribute values but the same ground-truth label, we prefer an initial ERM model that better learns this spurious dependency, and importantly also does not overfit to the training data. Inspired by the results in prior work (Sohoni et al., 2020; Liu et al., 2021), we then explored using either a standard ERM model, one with high ℓ-2 regularization, or one only trained on a few number of epochs. To select among these, because the validation data has both ground-truth and spurious attribute labels, we choose the model with the largest gap between worst-group and average accuracy on the validation set. We found high regularization and training with few epochs generally sufficient for their task of estimating different groups. Doing so was also preferable to another alternative of holding out additional training sets for ERM training versus prediction, as we could use more training datapoints for both ERM and contrastive model training. For each dataset, we detail the ERM architecture and hyperparameters below:

Colored MNIST. We use the LeNet-5 CNN architecture in the `pytorch` image classification tutorial. We train with SGD, few epochs $E = 5$, SGD, learning rate 1e-3, batch size 32, default weight decay 5e-4, and momentum 0.9.

Waterbirds. We use the `torchvision` implementation of ResNet-50 with pretrained weights from ImageNet as in Sagawa et al. (2019). Also as in (Sagawa et al., 2019), we train with SGD, default epochs $E = 300$, learning rate 1e-3, batch size 128, and momentum 0.9. However we use high weight decay 1.0.

CelebA. We also use the `torchvision` ImageNet-pretrained ResNet-50 and default hyperparameters from Sagawa et al. (2019) but with high weight decay: we train with SGD, default epochs $E = 50$, learning rate 1e-4, batch size 128, momentum 0.9, and high weight decay 0.1.

CivilComments-WILDS. We use the HuggingFace (`pytorch-transformers`) implementation of BERT with pre-trained weights and number of tokens capped at 300 as in Koh et al. (2021). As in Liu et al. (2021), with other hyperparameters set to their defaults (Koh et al., 2021) we tune between using the AdamW optimizer with learning rate 1e-5 and SGD with learning rate 1e-5, momentum 0.9, and the PyTorch `ReduceLROnPlateau` learning rate scheduler. Based on our criterion, we use SGD, few number of epochs $E = 2$, learning rate 1e-5, batch size 16, default weight decay 1e-2, and momentum 0.9.

D.2.3. Contrastive batch sampling details

We provide further details related to collecting predictions from the trained ERM models, and the number of positives and negatives that determine the contrastive batch size.

ERM model prediction. To collect trained ERM model predictions on the training data, we explored two approaches: (1) using the actual predictions, i.e. the argmax for each classifier layer output vector, and (2) clustering the representations, or the last hidden-layer outputs, and assigning a cluster-specific label to each datapoint in one cluster. This latter approach is inspired by Sohoni et al. (2020), and we similarly note that ERM models trained to predict ground-truth labels in spuriously correlated data may learn data representations that are clusterable by spurious attribute. As a viable alternative to collecting the "actual" predictions of the trained ERM model on the training data, with C ground-truth classes, we can then cluster these representations into C clusters, assign the same class label only to each datapoint in the same cluster, and choose the label-cluster assignment that leads to the highest accuracy on the training data. We also follow their procedure to first apply UMAP dimensionality reduction to 2 UMAP components, before clustering with K-means or GMM (Sohoni et al., 2020). To choose between all approaches, we selected the procedure that lead to highest worst-group accuracy on the validation data after the second-stage of training. While this cluster-based prediction approach was chosen as a computationally efficient heuristic, we found that in practice it either lead to comparable or better final worst-group accuracy on the validation set. To better understand this, as a preliminary result we found that when visualizing the validation set predictions with the Waterbirds dataset, the cluster-based predictions captured the actual spurious attributes better than the classifier layer predictions (Fig. 5). We defer additional discussion to Sohoni et al. (2020) and leave further analysis to future work.

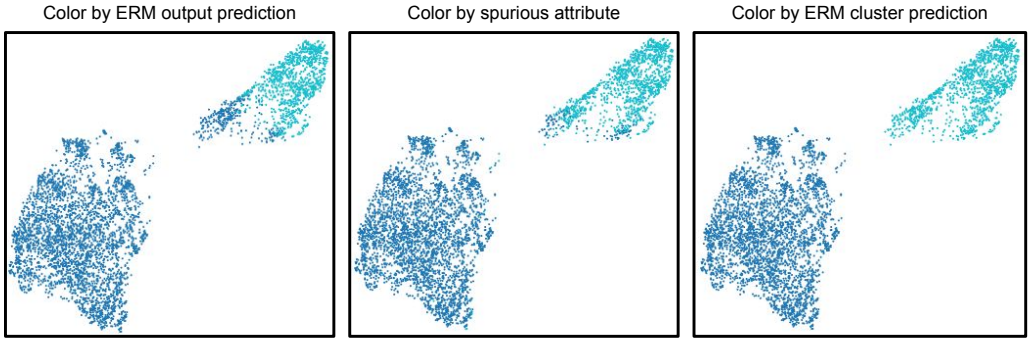


Figure 5. UMAP visualization of ERM data representations for the Waterbirds training data. We visualize the last hidden layer outputs for a trained ERM ResNet-50 model given training samples from Waterbirds, coloring by either the ERM model's "standard" predictions, the actual spurious attribute values (included here just for analysis), and predictions computed by clustering the representations as described above. Clustering-based predictions more closely align with the actual spurious attributes than the ERM model outputs.

Number of positives and negatives per batch. One additional difference between our work and prior contrastive learning methods (Chen et al., 2020; Khosla et al., 2020) is that we specifically construct our contrastive batches by sampling anchors,

positives, and negatives first. This is different from the standard procedure of randomly dividing the training data into batches first, and then assigning the anchor, positive, and negative roles to each datapoint in a given batch. As a result, we introduce the number of positives M and the number of negatives N as two hyperparameters that primarily influence the size of each contrastive batch (with number of additional anchors and negatives also following M and N with two-sided batches). To maximize the number of positive and negative comparisons, as a default we set M and N to be the maximum number of positives and negatives that fit the sampling criteria specified under Algorithm 1 that also can fit in memory. In Appendix D.2.4, for each dataset we detail the ERM prediction method and number of positives and negatives per batch.

D.2.4. Stage 2 contrastive model training details

In this section we describe the model architectures and training hyperparameters used for training the second model of our procedure, corresponding the reported worst-group and average test set results in Table 1. In this second stage, we train a new model with the same architecture as the initial ERM model, but now with a contrastive loss and batches sampled based on the initial ERM model's predictions. We report test set worst-group and average accuracies from models selected with hyperparameter tuning and early stopping based on the highest validation set worst-group accuracy. For all datasets, we sample contrastive batches using the clustering-based predictions of the initial ERM model. Each batch size specified here is also a direct function of the number of positives and negatives: $2M + 2N$.

Colored MNIST. We train a LeNet-5 CNN. For CNC, we use $M = 32$, $N = 32$, batch size 128, temperature $\tau = 0.05$, contrastive weight $\lambda = 0.75$, SGD optimizer, learning rate 1e-3, momentum 0.9, and weight decay 1e-4. We train for 3 epochs, and use gradient accumulation to update model parameters every 32 batches.

Waterbirds. We train a ResNet-50 CNN with pretrained ImageNet weights. For CNC, we use $M = 17$, $N = 17$, batch size 68, temperature $\tau = 0.1$, contrastive weight $\lambda = 0.75$, SGD optimizer, learning rate 1e-4, momentum 0.9, weight decay 1e-3. We train for 5 epochs, and use gradient accumulation to update model parameters every 32 batches.

CelebA. We train a ResNet-50 CNN with pretrained ImageNet weights. For CNC, we use $M = 64$, $N = 64$, batch size 256, temperature $\tau = 0.05$, contrastive weight $\lambda = 0.75$, SGD optimizer, learning rate 1e-5, momentum 0.9, and weight decay 1e-1. We train for 15 epochs, and use gradient accumulation to update model parameters every 32 batches.

CivilComments-WILDS. We train a BERT model with pretrained weights and max number of tokens 300. For CNC, we use $M = 16$, $N = 16$, batch size 64, temperature $\tau = 0.1$, contrastive weight $\lambda = 0.75$, AdamW optimizer, learning rate 1e-4, weight decay 1e-2, and clipped gradient norms. We train for 10 epochs, and use gradient accumulation to update weights every 128 batches.

D.2.5. Comparison method training details

As reported in the main results (Table 1) we compare CNC with the ERM and Group DRO baselines, as well as robust training methods that do not require spurious attribute labels for the training data: joint DRO (Levy et al., 2020), GEORGE (Levy et al., 2020), Learning from Failure (LfF) (Levy et al., 2020), and Just Train Twice (JTT) (Liu et al., 2021). For each dataset, we use the same model architecture for all methods. For the Waterbirds, CelebA, and CivilComments-WILDS datasets, we report the worst-group and average accuracies reported in Liu et al. (2021) for all comparison methods except GEORGE. For GEORGE, we report the accuracies reported in Sohoni et al. (2020). For these hyperparameters, we defer to the respective papers. For Colored MNIST, we run implementations for GEORGE and all other methods using code from the GEORGE (Sohoni et al., 2020) and JTT (Liu et al., 2021) authors respectively. We include training details below:

Colored MNIST. For JTT, we train with SGD, learning rate 1e-3, momentum 0.9, weight decay 5e-4, batch size 32, and report best worst-group accuracy after 20 epochs. We use the same initial ERM model as CNC, with hyperparameters described in Appendix D.2.2. For upsampling we first tried constant factors $\{10, 50, 100, 200, 1000\}$. We also tried a resampling strategy where for all the datapoints with the same initial ERM model prediction, we upsample the incorrect points such that they equal the correct points in frequency, and found this worked the best. With $p_{\text{corr}} = 0.995$, this upsamples each incorrect point by roughly 1100. We also use this approach for the results in Fig. 7.

For Group DRO we use the same training hyperparameters as JTT, but without the upsampling and instead set group adjustment parameter $C = 0$. For LfF, we use the same hyperparameters as JTT, but instead of upsampling gridsearched the q parameter $\in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, using $q = 0.7$. For joint DRO we do the same but use hyperparameter $\alpha = 0.1$. Finally for GEORGE we also train with SGD, learning rate 1e-3, momentum 0.9, weight decay 5e-4, and 20 epochs.

Table 3. CNC Average total training time for first and second stages of CNC

Dataset	CMNIST $p = 0.995$	Waterbirds	CelebA	CivilComments-WILDS
Stage 1 ERM train time	2 min.	1.5 hrs	3 hrs	3.1 hrs
Stage 2 CNC train time	1.2 hrs	1.8 hrs	32.2 hrs	37.6 hrs

Comparison limitations. One limitation of our comparison is that because for each dataset we sample new contrastive batches which could repeat certain datapoints, the number of total batches per epoch changes. For example, 50 epochs training the second model in CNC does not necessarily lead to the same total number of training batches as 50 epochs training with ERM, even if they use the same batch size. However, we note that the numbers we compare against from Liu et al. (2021) are reported with early stopping. In this sense we are comparing the best possible worst-group accuracies obtained by the methods, not the highest worst-group accuracy achieved within a limited number of training batches. We also found that although in general the time to complete one epoch takes much longer with CNC, CNC requires fewer overall training epochs for all but the CivilComments-WILDS dataset to obtain the highest reported accuracy.

D.3. CNC compute resources and training time

All experiments for Colored MNIST, Waterbirds, and CelebA were run on a machine with 14 CPU cores and a single NVIDIA Tesla P100 GPU. Experiments for CivilComments-WILDS were run on an Amazon EC2 instance with eight CPUs and one NVIDIA Tesla V100 GPU.

Regarding runtime, one limitation with the current implementation of CNC is its comparatively longer training time compared to methods such as standard ERM. This is both a result of training an initial ERM model in the first stage, and training another model with contrastive learning in the second stage. In Table 3 we report both how long it takes to train the initial ERM model and long it takes to complete one contrastive training epoch on each dataset. We observe that while in some cases training the initial ERM model is negligible, especially if we employ training with only a few epochs to prevent memorization (for Colored MNIST it takes roughly two minutes to obtain a sufficient initial ERM model), it takes roughly 1.5 and 3 hours to train the high regularization initial models used for Waterbirds and CelebA. While these hurdles are shared by all methods that train an initial ERM model, we find that the second stage of CNC occupies the bulk of training time. Prior work has shown that contrastive learning typically requires longer training times and converges more slowly than supervised learning (Chen et al., 2020). We also observe this in our work.

We note however that because we sample batches based on the ERM model's predictions, the contrastive training duration is limited by how many datapoints the initial ERM model predicts incorrectly. In moderately sized datasets with very few datapoints in minority groups, (e.g. Waterbirds, which has roughly 4794 training points and only 56 datapoints in its smallest group), the total time it takes to train CNC is on par with ERM. Additionally, other methods such as additional hard negative mining (Robinson et al., 2021) have been shown to improve the efficiency of contrastive learning, and we can incorporate these components to speed up training time as well.

E. Additional evaluations and ablations

E.1. Visualization of learned data representations

In Fig. 6, we visualize and compare the learned representations of test set samples from models trained with ERM, JTT, and CNC. Compared to ERM models, both JTT and CNC models learn representations that better depict dependencies on the ground-truth classes. However, especially with the Waterbirds and CelebA datasets, CNC model representations more clearly depict dependencies only on the ground-truth class, as opposed to JTT models which also show some organization by the spurious attribute still.

E.2. Representation metrics vs. worst-group accuracy with increasing spurious correlations strength

While Table 1 and Fig. 2 show that CNC's superior worst-group accuracy goes hand-in-hand with lower alignment loss and lower representation mutual information with spurious attributes, we also study how this relation between representation metrics and worst-group accuracy scales with the strength of the spurious correlations. We compute the same metrics, but now with CNC, ERM, and JTT models trained on increasingly spurious (increasing p_{corr}) Colored MNIST datasets, and report the resulting alignment and estimated mutual information metrics in Fig. 7.

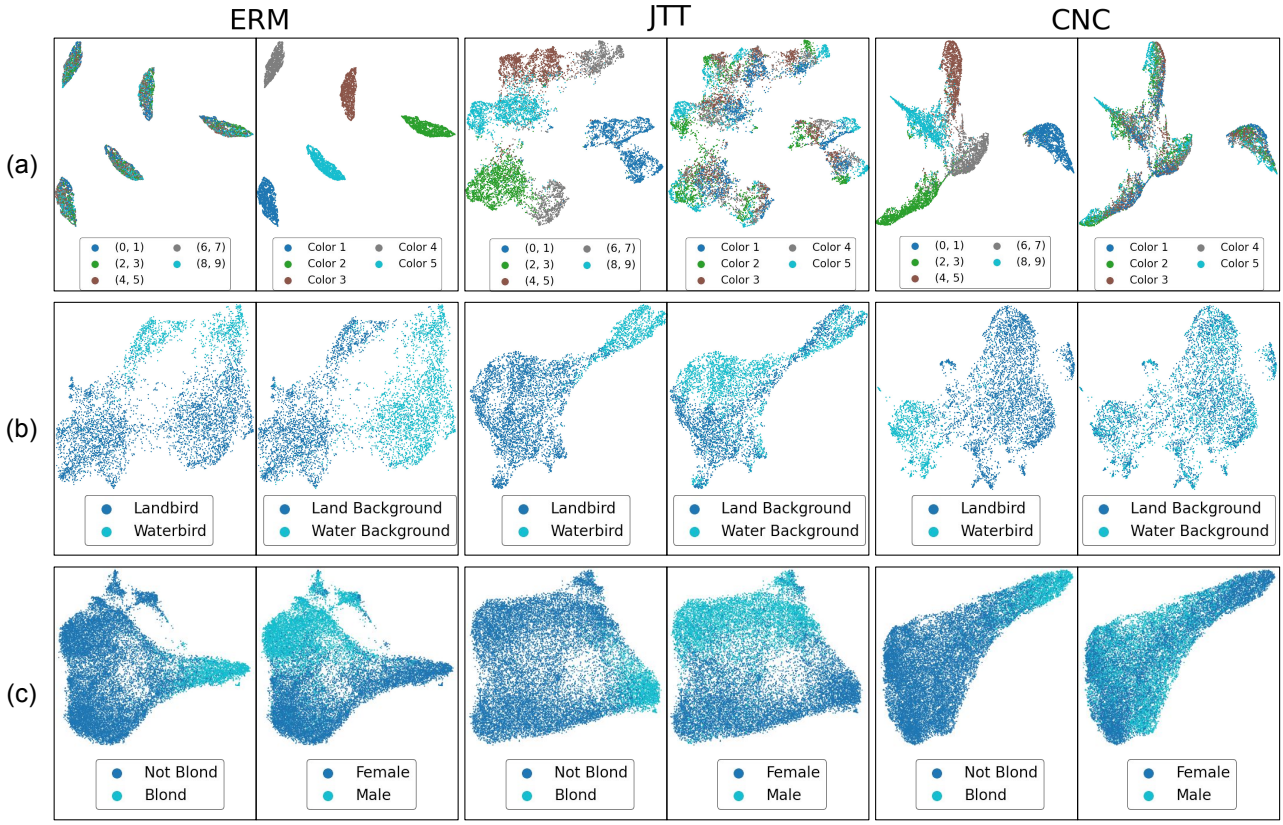


Figure 6. UMAP visualizations of learned representations for Colored MNIST (a), Waterbirds (b), and CelebA (c). We color datapoints based on the ground-truth (left) and spurious attribute (right). Most consistently across datasets, CNC representations exhibit dependence and separability by the ground-truth but not the spurious attribute, suggesting that they best learn features which only help classify ground-truth labels.

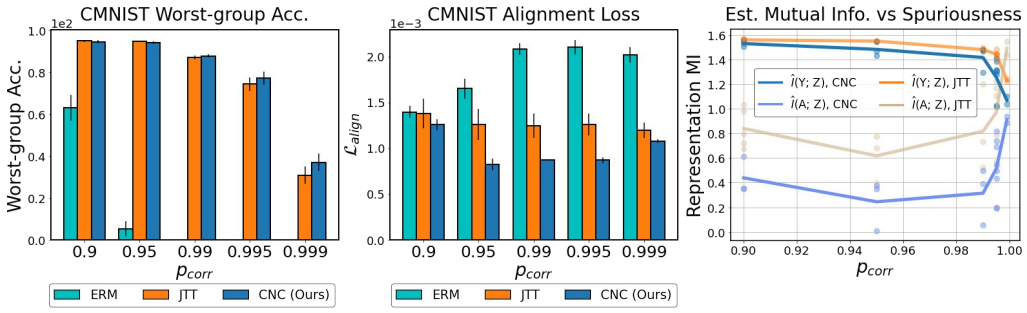
We observe that with high spurious correlations, ERM fails to classify digits in the minority classes, while CNC and JTT comparably maintain high worst-group accuracy. However, CNC performs marginally higher in more spurious settings with $p_{\text{corr}} > 0.95$. Both of these improvements over ERM are reflected by drops in alignment loss, but CNC also consistently achieves lowest $\mathcal{L}_{\text{align}}$ as a result of training with a contrastive loss. Fig. 7c shows that CNC's learned representations maintain a more favorable balance of mutual information between the ground-truth and spurious attribute than JTT. While JTT models exhibit slightly higher estimated $I(Y; Z)$ than CNC models, CNC models exhibit much lower dependence on the spurious attribute.

E.3. Empirical validation of CNC components

We validate the design choices of CNC through various ablations studying the effects of the individual components of our method on worst-group accuracy.

E.3.1. Importance of ERM-guided contrastive sampling

Although CNC relies on an initial trained ERM model's predictions, can we still improve worst-group accuracy without this step and with supervised contrastive learning alone, i.e. by sampling positives uniform randomly from *all* datapoints with the same label as the anchor? In Table 4, we show that this "vanilla" contrastive learning implementation competes with ERM on Waterbirds and outperforms ERM on CelebA in worst-group accuracy, while achieving similar average accuracy (standard deviation with three seeds recorded in parenthesis). However, we still achieve substantially higher worst-group accuracy with our full method. This supports that using initial ERM predictions to sample "hard comparisons" helps obtain best observed worst-group accuracy. We conjecture this is the case because positive samples with the ERM model are not



(a) (b) (c)

Figure 7. Alignment loss and mutual information representation metrics with worst-group accuracy on increasingly spurious CMNIST. CNC highest worst-group accuracy (a) coincides with learning representations with better alignment (b) and ratio of mutual information dependence on the labels vs the spurious attribute (c).

only the same class as anchors, but different enough such that an initial trained ERM model classified them differently. Comparing these against negatives which are also presumably more similar to the anchors than in the no-ERM setup lends additional contrastive learning signal to not learn dependencies on spurious differences.

Table 4. CNC accuracies without initial ERM model for contrastive sampling

Method	Waterbirds		CelebA	
Accuracy (%)	Worst-group	Average	Worst-group	Average
ERM	72.6	85.9	47.2	95.6
CNC (no ERM)	71.0 (1.92)	85.9 (0.8)	60.8 (0.8)	84.9 (0.4)
CNC (Ours)	89.0 (0.6)	90.4 (0.4)	88.8 (0.9)	89.85 (0.5)

Table 5. Ablation over CNC algorithmic components on Waterbirds. Default choices achieve highest worst-group and average accuracy.

Method	CNC (Default)	Projection Head	One-sided Contrasting	Train + Finetune
WG Acc. (%)	**89.0** (0.6)	82.4 (1.8)	85.2 (3.6)	84.0 (1.7)
Avg. Acc. (%)	**90.4** (0.4)	88.7 (0.6)	90.1 (1.6)	87.7 (1.1)

E.3.2. Additional design choice ablations

To validate the additional algorithmic components of CNC, we report how CNC performs on the Waterbirds dataset when modifying the individual design components. We use the same hyperparameters as in the main results, and report accuracies as the average over three training runs for the following ablations. Table 5 summarizes that across these design ablations, default CNC as presented consistently outperforms these alternative implementations.

No projection head. We incorporate a nonlinear projection head as is typical in prior contrastive learning works (Chen et al., 2020), that maps the encoder output to lower-dimensional representations (from 2048 to 128 in our case). We then update the encoder layers and the projection head jointly by computing the contrastive loss on the projection head's output, still passing the encoder layer's direct outputs to the classifier to compute the cross-entropy loss. We note that using the projection head decreases worst-group accuracy substantially. We reason that as previously discussed, while using the projection head in prior work can allow the model to retain more information in its actual hidden layers (Chen et al., 2020), in our case to remove dependencies on spurious attributes we actually want to encourage learning invariant representations when we model the differences between anchor and positive datapoints as due to spurious attributes.

Two-sided contrastive batches. Instead of "two-sided" contrasting where we allow both sampled anchors and positives to take on the anchor role, for each batch we only compute contrastive updates by comparing original positives and negatives with the original anchor. When keeping everything else the same, we find that just doing these one-sided comparisons also leads to a drop in performance for worst-group accuracy. This suggests that the increased number of comparisons and training setup where we swap the roles of anchors and positives of the two-sided batches introduces greater contrastive

learning signal.

Joint training of encoder and classifier layers. Instead of training the full model jointly, we first only train the encoder layers with the contrastive loss in CNC, before freezing these layers and finetuning the classifier layers with the cross-entropy loss. With this implementation, we also obtain noticeable drop in performance. While we leave further analysis for the joint cross-entropy and contrastive optimization for future work, one conjecture is that the cross-entropy loss may aid in learning separable representations while also training the full model to keep the average error small. From our theory, the contrastive loss can help bound the gap between worst-group and average error, and so to improve worst-group performance it may make sense to also try to minimize average error in the same parameter update.

This also follows prior work, where updating the entire model and finetuning all model parameters instead of freezing the encoder layers leads to higher accuracy (Chen et al., 2020). However, we found that with an initial encoder-only training stage, if we did not freeze the trained layers the fine-tuning on a dataset with spurious correlations would "revert" the contrastive training, resulting in a large gap between worst-group and average error similar to ERM.

E.4. Comparison to domain-invariant learning

We also study how EIIL, an invariance-learning method that similarly aims to use an initial ERM model to infer groups and learn an optimal classifier across all of them, compares with previously introduced methods for improving worst-group accuracy. In Table 6, we compare the worst-group and average accuracy obtained by EIIL on Waterbirds with the rest of our main results, reporting numbers from Creager et al. (2020). All methods use the same `torchvision` ResNet-50 implementation. We find that EIIL substantially outperforms the ERM baseline and joint DRO in worst-group accuracy, and also outperforms LfF, another approach that uses an ERM model's predictions to guide training a mroe robust model. However, it's worst-group accuracy falls below the others studied in this work. While EIIL has a similar objective with CNC to learn features that reliably predict the true class regardless of the domain, we conjecture that the contrastive learning component in CNC more directly encourage this invariance at the representation level, especially with respect to rarer minority groups that its initial ERM model frequently misclassifies. GEORGE and JTT may also outperform EIIL in worst-group accuracy as they actually focus training on the inferred minority groups, as opposed to invariant-learning mechanism in EIIL.

Table 6. Worst-group (WG) and average (Avg) accuracies for the Waterbirds dataset, with emphasis on comparing EIIL, a domain-invariant learning method, to others previously introduced.

Method	ERM	Joint DRO	LfF	EIIL*	GEORGE	JTT	CNC (Ours)	GDRO
WG Acc. (%)	72.6	69.5	75.2	78.7*	83.8	86.0	**89.0**	85.7
Avg Acc. (%)	85.9	88.5	91.6	96.9*	90.5	90.3	90.4	89.5

E.5. Additional GradCAM visualizations

We include additional GradCAM visualizations depicting saliency maps for samples from each group in the Waterbirds and CelebA datasets. Warmer colors denote higher saliency, suggesting that the model considered these pixels more important in making the final classification as measured by gradient activations. For both datasets, we compare maps from models trained with ERM, the next most competitive method for worst-group accuracy JTT, and CNC. CNC models most consistently measure highest saliency with pixels directly associated with ground-truth and not spurious attributes.

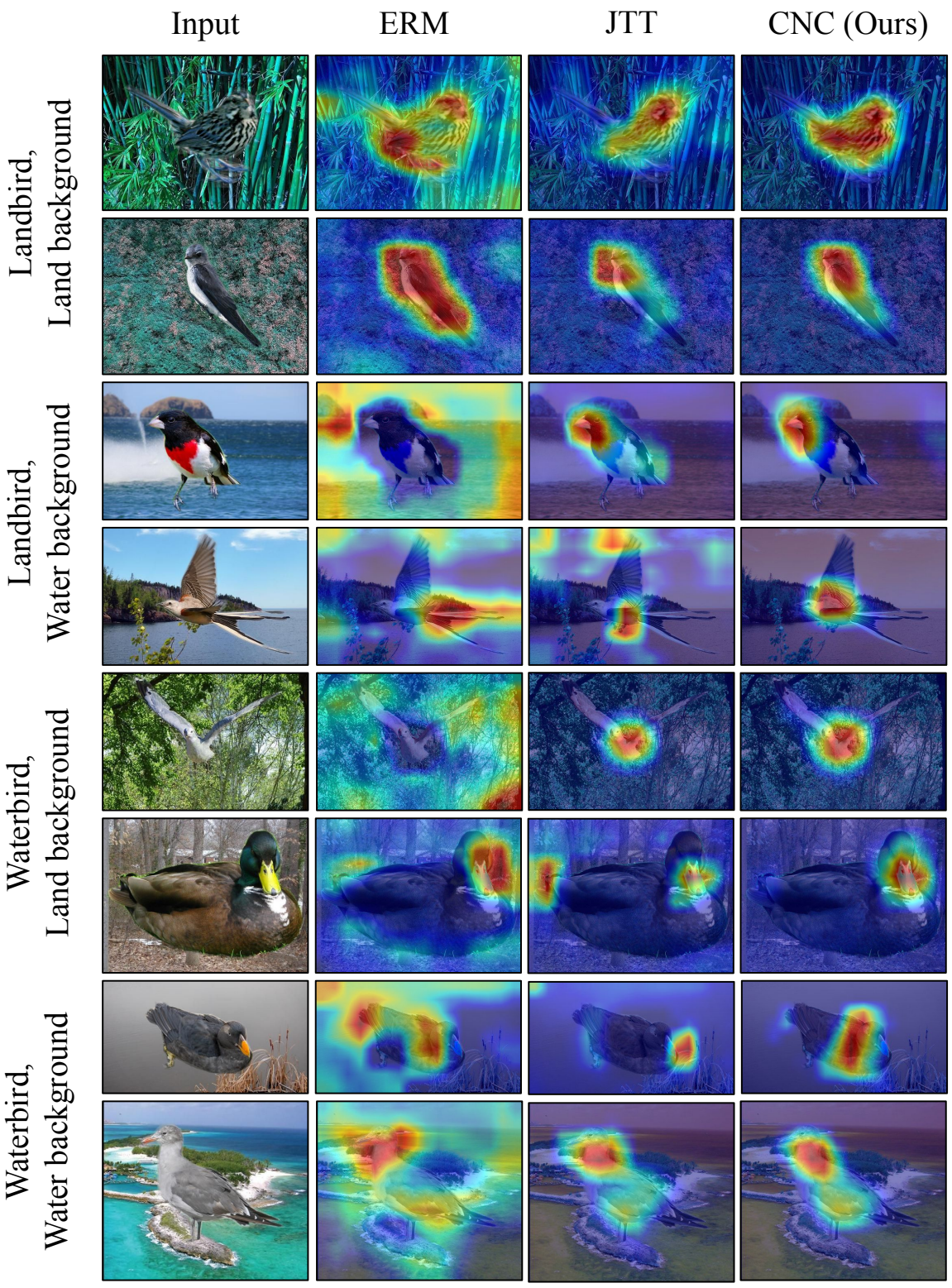


Figure 8. Additional GradCAM visualizations for the Waterbirds dataset. We use GradCAM to visualize the "salient" observed features used to classify images by bird type for models trained with ERM, JTT, and CNC. ERM models output higher salience for spurious background attribute pixels, sometimes almost exclusively. JTT and CNC models correct for this, with CNC better exclusively focusing on bird pixels.

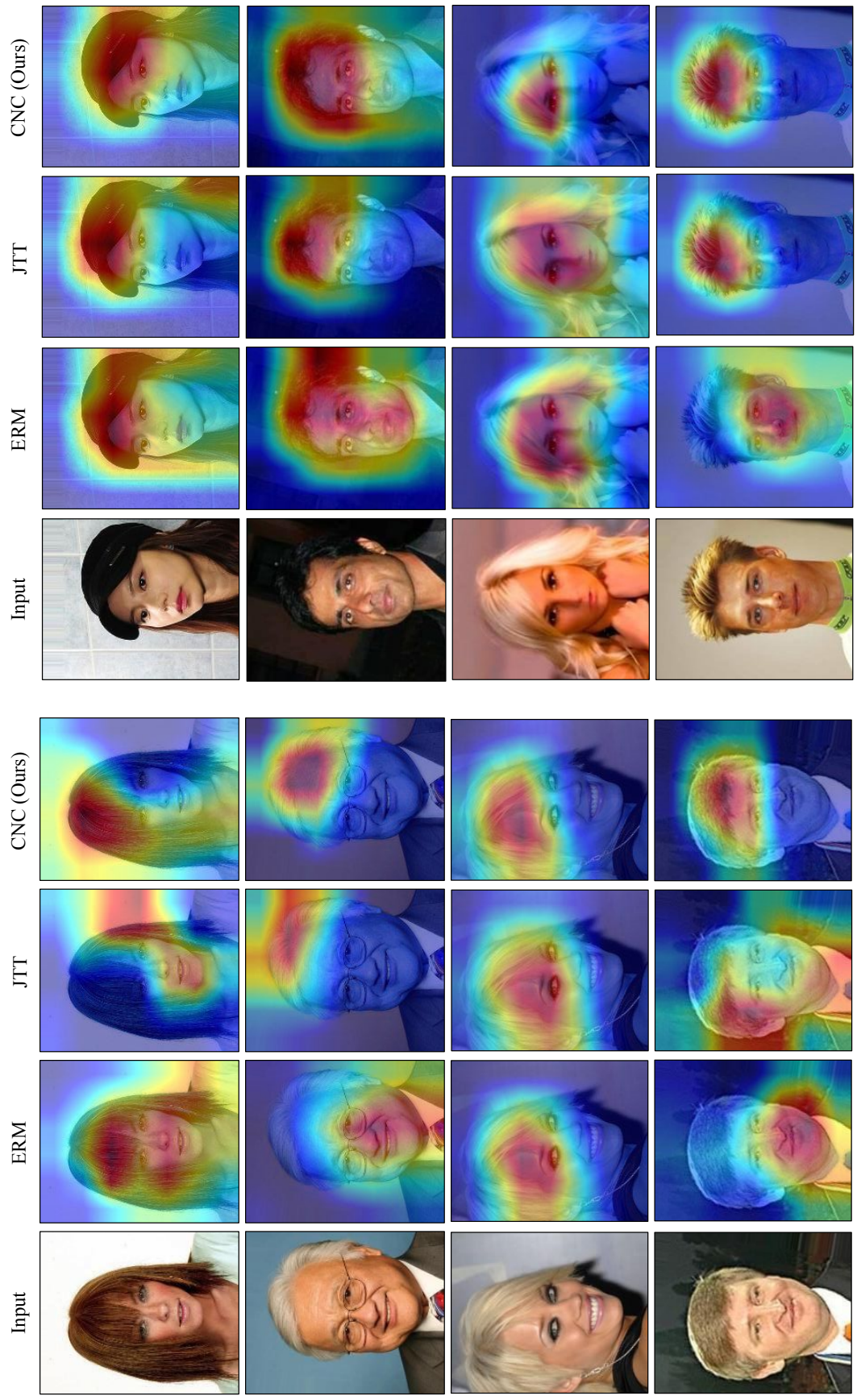


Figure 9. Additional GradCAM visualizations for the CelebA dataset. For models trained with ERM, JTT, and CNC, we use GradCAM to visualize the "salient" observed features used to classify whether a celebrity has blond(e) hair. ERM models interesting also "ignore" the actual hair pixels in favor of other pixels, presumably associated with the spurious gender attribute. In contrast, GradCAMs for JTT and CNC models usually depict higher salience for regions that at least include hair pixels. CNC models most consistently do so.