Causal Omnivore: Fusing Noisy Estimates of Spurious Correlations

Dyah Adila [1] **Sonia Cromp** [1] **Sicheng Mo** [1] **Frederic Sala** [1]

Abstract

Spurious correlations are one of the biggest pain points for users of modern machine learning. To handle this issue, many approaches attempt to learn features that are causally linked to the prediction variable. Such techniques, however, suffer from various flaws—they are often prohibitively complex or based on heuristics and strong assumptions that may fail in practice. There is no one-size-fits-all causal feature identification approach. To address this challenge, we propose a simple way to fuse multiple noisy estimates of causal features. Our approach treats the underlying causal structure as a latent variable and exploits recent developments in estimating latent structures without any access to ground truth. In addition, our approach omnivorously integrates any source of causal signal. We propose new sources, including an automated way to extract causal insights from existing ontologies or foundation models. On multiple benchmark environmental shift datasets, our discovered features can train a model via vanilla empirical risk minimization that outperforms multiple baselines, including automated causal feature discovery techniques such as invariant risk minimization on three benchmark datasets.

1. Introduction

Standard training pipelines struggle to differentiate between features that are causally linked to the prediction target and those that are merely associations. When measured in a new environment, such associations may no longer be predictive; they become spurious correlations. This leads to models that are brittle: they may perform well in environments identical to those they were trained on, but fail to generalize to others. The importance of this problem has spurred significant research in the hope of building tools to identify a set of causal features that transfer to any environment.

The holy grail is an algorithm that provably locates causal features from data without any additional signal. In general, this is hopeless. It is known that identifying causal relationships from observational data is not possible absent additional assumptions or knowledge. Recent works attempt to use information from data drawn from multiple environments to discover a common set of causal features (Peters et al., 2016; Arjovsky et al., 2019; Krueger et al., 2021). These techniques are promising but suffer from multiple flaws. For example, invariant risk minimization (IRM) (Arjovsky et al., 2019) requires a vast number of environments to be guaranteed to learn causal features—and may perform worse than vanilla empirical risk minimization when this fails to happen (Rosenfeld et al., 2021). Experimentally, none of these techniques are known to work in all cases (Dranker et al., 2021). Furthermore, by targeting a full end-to-end solution usable in any scenario, they ignore the presence of easily-accessible sources of causal knowledge in many specific scenarios.

Given the substantial challenge of a single technique that always finds spurious correlations, an alternative is to build an *omnivorous* method that can flexibly take advantage of any kind of causal signal.

We propose COMNIVORE , a system that takes a step towards satisfying this property by two key aspects. **First**, it enables the use of multiple sources to generate potential candidate feature sets. In particular, it allows for simple human-based specifications. When not available, it enables for simple ways to automate such specifications. **Second**, it extracts causal features from the resulting candidate feature sets by combining the outputs of multiple causal estimation approaches. Using principles similar to those in weak supervision (Ratner et al., 2018), it estimates the reliability of each causal estimate output, *without ground truth*. It then provides a higher-quality fused set of estimated features.

COMNIVORE is compatible with any pre-existing approach to causal feature estimation. It has the benefit of simplicity—not requiring any specialized loss functions or difficult bi-level optimization. Effectively, COMNIVORE simply asks as many sources of signal as possible for causal information, weights this information, and trains a downstream model on

[1]Department of Computer Science, University of Winconsin-Madison, Wisconsin, USA. Correspondence to: Dyah Adila <adila@wisc.edu>, Sonia Cromp <cromp@wisc.edu>.

Published at the ICML 2022 Workshop on Spurious Correlations, Invariance, and Stability. Baltimore, Maryland, USA. Copyright 2022 by the author(s).

the detected features with vanilla empirical risk minimization. We validate COMNIVORE empirically, showing that it outperforms competing end-to-end baselines like IRM, while improving on ERM by 37.9% on three benchmark datasets.

2. Background and Problem Setting

We first describe some of the tools we will use and then detail the problem setting.

Identifying Causal Features Discovering causal features is an active area of research under which approach can take many forms. We briefly describe the approach we use. First, an important problem in causal inference is learning causal structures from observational data, interventions, structural assumptions, or heuristics. Naturally, such approaches do not work in every setting; violations of their underlying assumptions can be thought of as noise. For instance, Peter-Clark (PC) (Spirtes et al., 2000), FCI (Spirtes et al., 2000), and Greedy Equivalence Search (Chickering, 2003) assume the absence of certain conditions on the latent confounders between features. Grow-Shrink (GS) (Margaritis, 2003), Incremental Association Markov Blanket (IAMB) (Tsamardinos et al., 2003), Interleaved IAMB (Yaramakala & Margaritis, 2005), and Exact Search (Silander & Myllymaki, 2012) require the underlying model to have a certain Bayesian structure. More recent optimization-based methods (Zheng et al., 2018) (Zheng et al., 2020) are limited by optimization constraints. These assumptions thus limit their accuracy when applied to complex and high-dimensional data.

Weak Supervision Weak supervision is a set of techniques that are used to construct labeled training sets (Ratner et al., 2016; 2018; Fu et al., 2020) from unlabeled training data. The idea is that even though no labels are available, multiple noisy estimates of each label are observed. These are the outputs of labeling functions $\lambda_1, \ldots, \lambda_m$. The challenge is to determine the reliability of these functions and to use this information to fuse their outputs into a pseudolabel of higher quality than each of their individual votes. We use similar principles to fuse noisy causal estimates.

Problem Setting We have access to a dataset $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ of samples drawn from some distribution D. Potentially, we have access to additional training distribution sets from $D_2, D_3, \ldots, D_k$. We refer to these as the k training environments. Different environments represent the different data collection settings (e.g., hospital, time periods, countries, demographics).

Our goal is to learn a model $f(x)$ that performs well in new scenarios. Typically this means that it generalizes well to test distributions $\mathcal{D}_{test}$, where $\mathcal{D}_{test}$ and $\mathcal{D}_{train}$ are non-overlapping (i.e., $\mathcal{D}_{test} \cup \mathcal{D}_{train} = \emptyset$). However, we might also be interested in $f(x)$ performing well in subpopulation shift scenarios. In this case, our goal is to maximize $f(x)$'s performance across all domains seen during training (i.e., $\mathcal{D}_{test} \subseteq \mathcal{D}_{train}$), but the proportions of samples from each domain can change.

We also assume we have access to pretrained model or foundation model (FM) embeddings. These embeddings are the outputs of a mapping $g : X \to Z$ from input space to latent embedding space. This mapping is fixed and obtained from off-the-shelf models.

3. Approach

Algorithm 1 COMNIVORE

Input: Training dataset drawn from a distribution $\mathcal{D}_{orig} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, causal feature selection functions λ, embedding mapping f

Generate candidate sets C
for $C_i \in C$ **do**
  for $\lambda_i \in \lambda$ **do**
    Generate causal structure estimates G^i
  for $G^i \in \mathcal{G}$ **do**
    if *fusing method* == *Graph-based WS* **then**
      Compute λ_j's weight w_j
      while *Annealing iteration* **do**
        Minimize weighted objective as in (1) to get $\hat{G}$
      collect all $z_i \in z$ that has a causal edge to label node
    if *fusing method* == *Vanilla WS* **then**
      for z_i *in* z **do**
        Construct label matrix L
        Get causal predictions from WS system $\hat{Y}_i \pm 1$

Result: Singleton set of configurations .

We present our method to train models robust to spurious correlations: COMNIVORE. At high level, we break up the task into two parts. Our first goal is to obtain many sources of potentially causal features and group them into a set of distinct feature transformations. These might include the raw features, embeddings from pretrained models like a ResNet or foundation models like CLIP, the result from performing a manually-chosen transformation/augmentation on the dataset, or their combinations. We call the resulting sets of feature transformations the *candidate sets*.

Directly relying on the features in these candidate sets may not be sufficient, however—they may also be affected by spurious correlations. To further refine our estimated features, we run a suite of causal estimation approaches for each set. We refer to these as the *causal feature selection*

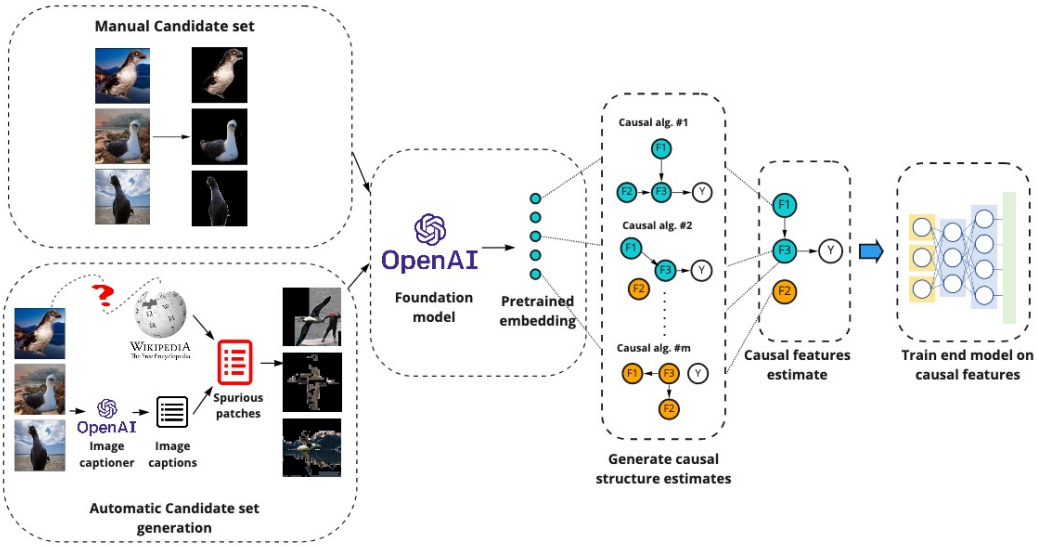

Figure 1: COMNIVORE seeks to find causal features through a two-step process. It flexibly pulls together multiple sources of candidate feature sets (left). It runs these candidate sets through a suite of causal feature estimation approaches and models and combines the resulting estimates (center). A conventional end model is trained on the discovered features (right).

functions. We will estimate the reliability of each of the selection functions and produce an improved overall estimate of the causal features. Finally, we combine the resulting estimated features from each of the candidate sets and train the end model.

Generating candidate sets Our first task is to generate candidate sets $\{C_1, C_2, \ldots, C_b\}$. These are transformed versions of the original features that ideally have some reduced spuriousness. Potential choices of these include,

- Embeddings from off-the-shelf models, such as foundation models like CLIP,
- Existing end-to-end invariant feature construction methods, like IRM, when suitable,
- Manually-selected transformations/augmentations,
- Automated transformations/augmentations.

We describe the latter two possibilities. First, we observe that humans can often identify causal features with ease. As a running example, consider the Waterbirds dataset (Wah et al., 2011) (Zhou et al., 2018). The goal is to classify birds as being water-based or terrestrial, and the background in the images of these birds (bodies of water versus land) acts as a spurious feature. This is challenging for training algorithms to discern, but nearly trivial for humans. It is easy to encode this human insight into transformations that can be built with off-the-shelf tools. In this case, running a standard segmentation algorithm to isolate the bird acts as such a transformation, as shown in Figure 2.

Manually-selected transformations help translate easily-acquired human insights into high-quality candidate sets. However, we do not always have access to such information. In Appendix C, we show how to automate the process of encoding human insights into causal versus spurious features. This will enable us to get the best-of-both worlds.

The term Candidate Set loosely applies to any result of transformations that can possibly reduce spuriousness in the original sample. When using manually-selected transformation, this term refers to the transformed samples. When using invariant feature construction like IRM, Candidate Set refers to the latent feature extracted from the penultimate layer of an IRM model. For the first case, we then get the latent representation using foundation model or any pretrained model $\theta : C \to \{z_1^1, \ldots, z_d^1\}, \ldots, \{z_1^n, \ldots, z_d^n\}$.

Generating causal feature selection functions Next, we use the suite of causal inference algorithms listed in Section 2 to obtain the estimated causal structures for each candidate set. These algorithms take the sets of features paired with the corresponding labels $\{(\{z_1^1, \ldots, z_d^1\}, y^1), \ldots, (\{z_1^n, \ldots, z_d^n\}, y^n)\}$ for each available training environment and output the estimates of causal structures that govern the relationships among individual features and with the label. These causal structures are represented in form of DAGs in $\mathcal{G}$. Formally, given m causal algorithms, the output of each algorithm λ^a is described by

$$\lambda^a : \{(\{z_1^1, \ldots, z_d^1\}, y^1), \ldots, (\{z_1^n, \ldots, z_d^n\}, y^n)\} \to \mathcal{G}, \quad a = 1, \ldots m$$

One challenge is that such algorithms often have high complexity, sometimes superexponential in the number of fea-

tures. We use a simple way to address this difficulty. We map the features into a lower-dimensional space, perform estimation in this space, and then return to the original space. We use Feature Agglomeration (Steinbach et al., 2000), but our method is compatible with any dimensionality reduction technique. At the end of this step, we have m DAGs $\{G^1, \ldots, G^m\}$ per candidate set and environment.

Our approach treats the causal inference algorithms like labeling functions in weak supervision (section 2). Since these algorithms assume certain conditions or heuristics, we expect their causal estimate outputs to be noisy. Weak supervision nicely complements our need to combine these noisy estimates since it fuses noisy labels to build a labeled dataset.

Fusing noisy causal estimates Our final task is to obtain a fused estimate $\mathcal{G}$ from the DAGs. Our goal is to obtain a better set of features compared to the noisy DAG estimates $\{G^1, \ldots, G^m\}$. We employ two weak supervision-based techniques to combine the G^a's into $\hat{G}$:

1. **Graph-based Weak Supervision**. With this approach, we learn weights w_a for each estimate G^a. These weights correspond to average distances to a true G^* which we do not observe. To estimate the weights, we use the algorithm in (Shin et al., 2022), described below. We embed the graphs into $\mathbb{R}^d$, producing an embedding $r(G^a)$. We set up the following system of equations for triplets (a, b, c) chosen from $\{1, \ldots, m\}$:

$$\|r(G^a) - r(G^b)\|^2 = \|r(G^a) - r(G^*)\|^2 + \|g(G^b) - r(G^*)\|^2$$
$$\|r(G^a) - r(G^c)\|^2 = \|r(G^a) - r(G^*)\|^2 + \|g(G^c) - r(G^*)\|^2$$
$$\|r(G^b) - r(G^c)\|^2 = \|r(G^b) - r(G^*)\|^2 + \|r(G^c) - r(G^*)\|^2.$$

To obtain $\|r(G^a) - r(G^*)\|^2$, we add the first two equations, subtract the third, and divide by two. This is an estimate of the average distance between our (embeddings of) graphs; the weights w are just the reciprocals, so that $w_a = 1/(\|r(G^a) - r(G^*)\|^2)$.

Once we have estimated $\hat{w}_a$, we perform the following optimization,

$$\hat{G} = \arg\min_{G \in \mathcal{G}} \frac{1}{m} \sum_{j=1}^{m} w_j d_H(G, G^j) \tag{1}$$

Note that we compute the mean in the original DAG space, not in the embedding space. We use simulated annealing (Kirkpatrick et al., 1983), an iterative global search optimization method, to obtain $\hat{G}$. Next we take all features z_i that have a causal path in $\hat{G}$ to the label node as the causal feature subset.

2. **Vanilla Weak Supervision.** Alternatively, instead of searching for the best overall causal structure, we can try to solve a perhaps more manageable problem: *is feature z_i causally related to the label y?* We treat this problem for each feature z_i as a simple classification problem $Y_i \in \{\pm 1\}$ where $+1$ means *a causal edge present between z_i and y* in DAG G^j, 0 indicates no relationship, and -1 *an anti-causal edge present between z_i and y*.

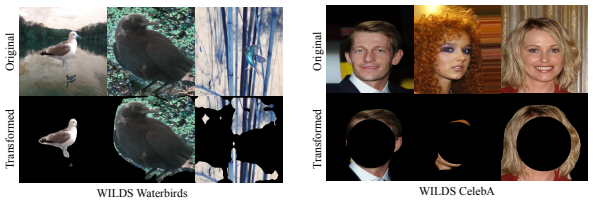

Figure 2: Manual candidate set examples. Humans can easily observe that background is not causally linked to bird species (left) and that gender is not linked to hair (right).

Inspired by (Fu et al., 2020), for each z_i, we first construct a $k \times m$ label matrix L, where k is the number of environments we have access to and m is the number of causal estimation functions. Note that L is constructed for each z_i in each candidate set separately. We encode the predictions output by each estimation algorithm into L and pass it as input to any weak supervision approach, e.g., (Ratner et al., 2018; 2019; Fu et al., 2020).

4. Experiments

This section validates the following claims about COMNIVORE:

- **Performance (Section 4.1):** We show that COMNIVORE outperforms baseline end-to-end approaches (IRM (Arjovsky et al., 2019) and REx (Krueger et al., 2021)) on unseen environment $\mathcal{D}_{new}$ with comparable performance on the original environment $\mathcal{D}_{orig}$ on both subpopulation shift and domain generalization datasets.

- **Ablations (Section 4.2):** The source of COMNIVORE's lift on $\mathcal{D}_{new}$ while retaining good performance on $\mathcal{D}_{orig}$ is **all** components of the approach. We show this by comparing COMNIVORE with training vanilla ERM and baseline approaches using the foundation model embeddings of the candidate sets.

Datasets We evaluate COMNIVORE on three datasets in the WILDS benchmark (Koh et al., 2020). In subpopulation shift, we use **Waterbirds**, which combines bird images from the CUB dataset (Wah et al., 2011) with backgrounds from the Places dataset (Zhou et al., 2018); and **CelebA** celebrity faces dataset (Liu et al., 2015); In domain generalization, we use **Camelyon17** (Bándi et al., 2019), which task is to identify tumor in medical images. We also evaluate on

Dataset	IRM		REx		COMNIVORE-G		COMNIVORE-V	
	$\mathcal{D}_{new}$	$\mathcal{D}_{orig}$	$\mathcal{D}_{new}$	$\mathcal{D}_{orig}$	$\mathcal{D}_{new}$	$\mathcal{D}_{orig}$	$\mathcal{D}_{new}$	$\mathcal{D}_{orig}$
Waterbirds	37.5	72.3	58.6	**95.5**	**71.0**	91.4	71.0	90.7
CelebA	63.3	88.5	61.6	85.1	60.4	88.6	**63.4**	**90.1**
Camelyon17	64.2*	82.6*	75.0	87.0	**87.2**	**91.6**	72.3	89.0
ColorMNIST	66.9*	70.8*	68.7*	71.5*	70.4	**99.7**	**80.0**	67.2

Table 1: COMNIVORE performance compared to baseline end-to-end approaches. All scores are accuracy. Best results for $\mathcal{D}_{new}$ are highlighted in blue and $\mathcal{D}_{orig}$ in red. COMNIVORE -G uses Graph-based WS as fusing method, COMNIVORE-V uses Vanilla WS. Results are average over three runs. Results marked by * are quoted from appropriate papers.

Dataset	ERM		ERM(Augment)		COMNIVORE-G	
	$\mathcal{D}_{new}$	$\mathcal{D}_{orig}$	$\mathcal{D}_{new}$	$\mathcal{D}_{orig}$	$\mathcal{D}_{new}$	$\mathcal{D}_{orig}$
Waterbirds	24.6	**96.0**	63.7	93.7	**71.0**	91.4
CelebA	2.20	**93.8**	52.0	90.0	**60.4**	88.6
Camelyon17	78.2	89.5	74.3	90.0	**87.2**	**91.6**
ColorMNIST	9.0	93.0	22.8	**100.0**	**70.4**	99.7

Table 2: COMNIVORE ablations. All scores are accuracy.

ColorMNIST, where spurious correlations between digits and color are artificially created, similar to the synthetic task used in IRM and REx. We list the complete dataset details in Appendix F.1. **Pre-trained embeddings** We use pre-trained CLIP embeddings (Radford et al., 2021).

4.1. Performance Comparisons

We compare COMNIVORE with IRM and REx, measuring accuracy on the original train distribution $\mathcal{D}_{orig}$ and the new test distribution $\mathcal{D}_{new}$. For COMNIVORE , we train a simple 2-layer MLP using ERM on the sets of causal features acquired using both graph-based WS and vanilla WS. For IRM and REx, we experiment with 2-layer MLPs using two choices of feature extractors: CLIP and ResNet50. The latter follows the choice of architecture used in the WILDS benchmark (Koh et al., 2020). We report the best of the two results.

Table 1 shows the results. COMNIVORE outperforms IRM and REx on $\mathcal{D}_{new}$ across all datasets. For $\mathcal{D}_{orig}$, COMNIVORE's performance is comparable to the best baseline on Waterbirds (by 4.6%) and achieved the best accuracies on ColorMNIST (tie with IRM), CelebA and Camelyon17. This reflects our method's ability to ingest and refine a large number of causal features.

We also note that we construct our implementation of ColorMNIST. In contrast with IRM and REx, we do not collapse the classes and directly use the digits 0 vs 1. In IRM's version of ColorMNIST, IRM achieves 70.8% accuracy on $\mathcal{D}_{orig}$ and 66.9% $\mathcal{D}_{new}$; and in REx's implementation, REx achieves 71.5% accuracy on $\mathcal{D}_{orig}$ and 68.7% on $\mathcal{D}_{new}$. Our main experimental table contains the values we obtained on our version of the dataset. As an additional point of comparison, we note that EIIL (Creager et al., 2021) achieves a 69.7% accuracy on $\mathcal{D}_{orig}$ and 78.8% $\mathcal{D}_{new}$ in Waterbirds dataset.

4.2. Ablations

We train models using vanilla ERM on the extracted features of the candidate sets and original images, without performing the causal estimate step. We report the results in table 2. On $\mathcal{D}_{new}$, COMNIVORE outperforms both vanilla ERM trained using original images and candidate sets. COMNIVORE's performance on $\mathcal{D}_{orig}$ is within relatively comparable accuracy with the best of vanilla ERM on Waterbirds, CelebA, and ColorMNIST (by 5.6%, 5.2%, and 0.3%) and performed the best on Camelyon17. This result demonstrates that COMNIVORE 's performance lift is produced by both candidate set generation and causal feature selection.

References

Ahmed, F., Bengio, Y., van Seijen, H., and Courville, A. Systematic generalisation with group invariant predictions. In *International Conference on Learning Representations*, 2020.

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., and Ishaan Gulrajani, D. L.-P. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

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.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

Bándi, P., Geessink, O., Manson, Q., Van Dijk, M., Balkenhol, M., Hermsen, M., Ehteshami Bejnordi, B., Lee, B.,

Paeng, K., Zhong, A., Li, Q., Zanjani, F. G., Zinger, S., Fukuta, K., Komura, D., Ovtcharov, V., Cheng, S., Zeng, S., Thagaard, J., Dahl, A. B., Lin, H., Chen, H., Jacobsson, L., Hedlund, M., Çetin, M., Halıcı, E., Jackson, H., Chen, R., Both, F., Franke, J., Küsters-Vandevelde, H., Vreuls, W., Bult, P., van Ginneken, B., van der Laak, J., and Litjens, G. From detection of individual metastases to classification of lymph node status at the patient level: The camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 38(2):550–560, 2019. doi: 10.1109/TMI.2018.2867350.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017.

Chickering, D. M. Learning bayesian networks is np-complete. In *Learning from data*, pp. 121–130. Springer, 1996.

Chickering, D. M. Optimal structure identification with greedy search. *J. Mach. Learn. Res.*, 3(null): 507–554, mar 2003. ISSN 1532-4435. doi: 10.1162/153244303321897717. URL https://doi.org/10.1162/153244303321897717.

Chickering, M., Heckerman, D., and Meek, C. Large-sample learning of bayesian networks is np-hard. *Journal of Machine Learning Research*, 5:1287–1330, 2004.

Creager, E., Jacobsen, J.-H., and Zemel, R. Environment inference for invariant learning. In *International Conference on Machine Learning*, pp. 2189–2200. PMLR, 2021.

Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Dranker, Y., He, H., and Belinkov, Y. Irm—when it works and when it doesn't: A test case of natural language inference. In *Proc. of the 35th Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Duchi, J., Glynn, P., and Namkoong, H. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv preprint arXiv:1610.03425*, 2016.

F.R.S., K. P. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11): 559–572, 1901. doi: 10.1080/14786440109462720.

Fu, D. Y., Chen, M. F., Sala, F., Hooper, S. M., Fatahalian, K., and Ré, C. Fast and three-rious: Speeding up weak supervision with triplet methods. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, 2020.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. Optimization by simulated annealing. *SCIENCE*, 220(4598): 671–680, 1983.

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. *CoRR*, abs/2012.07421, 2020. URL https://arxiv.org/abs/2012.07421.

Koyama, M. and Yamaguchi, S. Out-of-distribution generalization with maximal invariant predictor. 2020.

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. In *Proc. of the 38th International Conference on Machine Learning (ICML)*, 2021.

Liu, E. Z., 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. In *International Conference on Machine Learning*, pp. 6781–6792. PMLR, 2021.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2015.

Margaritis, D. Learning bayesian network model structure from data. Technical report, Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science, 2003.

Mokady, R., Hertz, A., and Bermano, A. H. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.

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

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

Peters, J. and Bühlmann, P. Structural intervention distance for evaluating causal graphs. *Neural Computation*, 27(3), 2015.

Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society Series B*, 78(5), 2016.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Ratner, A., Bach, S. H., Ehrenberg, H., Fries, J., Wu, S., and Ré, C. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the 44th International Conference on Very Large Data Bases (VLDB)*, Rio de Janeiro, Brazil, 2018.

Ratner, A. J., Sa, C. M. D., Wu, S., Selsam, D., and Ré, C. Data programming: Creating large training sets, quickly. In *Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS 2016)*, Barcelona, Spain, 2016.

Ratner, A. J., Hancock, B., Dunnmon, J., Sala, F., Pandey, S., and Ré, C. Training complex models with multi-task weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, 2019.

Rosenfeld, E., Ravikumar, P., and Risteski, A. The risks of invariant risk minimization. In *International Conference on Learning Representations (ICLR)*, 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. *arXiv preprint arXiv:1911.08731*, 2019.

Shi, Y., Seely, J., Torr, P. H., Siddharth, N., Hannun, A., Usunier, N., and Synnaeve, G. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

Shin, C., Li, W., Vishwakarma, H., Roberts, N. C., and Sala, F. Universalizing weak supervision. In *International Conference on Learning Representations (ICLR)*, 2022.

Silander, T. and Myllymaki, P. A simple approach for finding the globally optimal bayesian network structure. *arXiv preprint arXiv:1206.6875*, 2012.

Spirtes, P., Glymour, C. N., Scheines, R., and Heckerman, D. *Causation, prediction, and search*. MIT press, 2000.

Steinbach, M., Karypis, G., and Kumar, V. A comparison of document clustering techniques. *Proceedings of the International KDD Workshop on Text Mining*, 06 2000.

Tsamardinos, I., Aliferis, C. F., Statnikov, A. R., and Statnikov, E. Algorithms for large scale markov blanket discovery. In *FLAIRS conference*, volume 2, pp. 376–380. St. Augustine, FL, 2003.

Umesh, P. Image processing in python. *CSI Communications*, 23, 2012.

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

Wikipedia contributors. Wikipedia, the free encyclopedia. URL https://en.wikipedia.org/w/index.php?title=Plagiarism&oldid=5139350.

WILDS. Wilds leaderboard. URL https://wilds.stanford.edu/leaderboard/#without-unlabeled-data-2.

Yaramakala, S. and Margaritis, D. Speculative markov blanket discovery for optimal feature selection. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pp. 4–pp. IEEE, 2005.

Zhang, M., Sohoni, N. S., Zhang, H. R., Finn, C., and Ré, C. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations, 2022. URL https://arxiv.org/abs/2203.01517.

Zheng, X., Aragam, B., Ravikumar, P., and Xing, E. P. DAGs with NO TEARS: Continuous Optimization for Structure Learning. In *Advances in Neural Information Processing Systems*, 2018.

Zheng, X., Dan, C., Aragam, B., Ravikumar, P., and Xing, E. P. Learning sparse nonparametric DAGs. In *International Conference on Artificial Intelligence and Statistics*, 2020.

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, 2018. doi: 10.1109/TPAMI.2017.2723009.

Appendix

We discuss related work (Appendix A), provide a glossary containing key terminology (Appendix B) and algorithm table (Appendix **??**), introduce additional details into our algorithm claims (Appendix E), then give extra experimental details and results (Appendix F). We also provide theoretical analysis considering simple setups where estimates of causal features are useful (Appendix D). Finally, we show a simple approach to automate the process of encoding human insights into causal versus spurious features mentioned in section 3 (Appendix C).

A. Related Work

This section presents discussion of related work and connections to our work.

Invariant learning methods such as IRM (Arjovsky et al., 2019), REx (Krueger et al., 2021), and a multitude of similar works (Ahmed et al., 2020; Parascandolo et al., 2020; Ahuja et al., 2020; Creager et al., 2021) share a similar goal with our work. The aim is finding feature representations that are invariant across domains or environments. We can think of these invariant features as similar to our goal causal feature subset. This is achieved mainly by minimizing specialized loss functions. IRM and REx minimize the sum of loss terms across environments and thus require environment labels. Environment Inference for Invariant Learning (EIIL) (Creager et al., 2021) and Predictive Group Invariance (PGI) (Ahmed et al., 2020) train an initial ERM model to infer environment labels and later on train another model with invariant learning objectives. In contrast, COMNIVORE estimates the causal features *in-prior* to training a model and thus circumvents the need for any specialized loss functions.

Improving robustness to spurious correlations and distribution shift is an extensive line of work that can be divided into two aspects, based on whether access to group/domain information is given or not. In the line that requires group information a priori, one popular work is group distributionally robust optimization (GDRO) (Sagawa et al., 2019), which divides the data into explicit groups and then trains them to directly minimize the worst group-level error among these groups. Similarly, Fish (Shi et al., 2021) and Inter-environment Gradient Alignment (IGA) (Koyama & Yamaguchi, 2020) aim to improve domain generalization performance by maximizing inter-domain gradient terms in their loss functions.

More similar to our approach are methods that do not assume group information at training time. For instance, distributionally robust optimization (DRO) minimizes worst group loss within a ball centered around the training distribution (Ben-Tal et al., 2013; Duchi et al., 2016). More recent methods (Nam et al., 2020; Liu et al., 2021; Zhang et al., 2022) train two ERM models: the first one is to estimate which data points play a crucial role in their subsequent steps (e.g., which points belong to minority groups, which samples come from the same class but has different spurious features, etc.). Next, such methods train another ERM model with specialized objectives (e.g., to up-weight minority groups, using contrastive loss to learn invariant features, etc.). Note that all of these works are compatible with our approach as well.

Causal inference algorithms (Spirtes et al., 2000; Margaritis, 2003; Chickering, 2003; Tsamardinos et al., 2003; Yaramakala & Margaritis, 2005; Silander & Myllymaki, 2012; Zheng et al., 2018;?) seek to discover the structure that governs relationship between set of features in the data. Ideally, for our purpose, if we feed the sets of features and labels into these algorithms, we hope to be able to extract the features that have a causal link to the label. Unfortunately, this problem is statistically and computationally hard (Chickering, 1996; Chickering et al., 2004). As a result, these methods resort to local heuristics and assumptions, thus limiting their accuracy when applied to complex high-dimensional data. Our approach fuses these noisy estimates of causal structures to get the estimated set of causal features on which training an end model will be robust to spurious correlation and domain shift.

Weak supervision is a set of techniques that use noisy sources of labels to construct labeled training sets without access to ground truth labels (Ratner et al., 2016; 2018; Fu et al., 2020). This technique is vastly explored for binary classification problems. Recently, (Shin et al., 2022) enables weak supervision over broader sets of problems, which also serves as a basis for our graph-based weak supervision fusion method.

B. Glossary

The glossary is given in Table 3 below.

Symbol	Used for
x	Input data point $x \in X$.
y	Ground truth label $y \in Y = \{0, 1\}$.
$\mathcal{D}$	Data Distribution, where each $\mathcal{D}_i$ is a distribution where samples are drawn (e.g., $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$).
f	End classification model.
g	A fixed mapping from input space X to embedding space Z that is made available by the off-the-shelf foundation model.
C	Candidate sets.
z	Features $z = \{z_1, \dots z_d\}$, where each z_i is feature vector entry at index i.
n	Number of data points.
d	Number of features (i., dimension of feature vector).
λ	Causal inference algorithms $\lambda = \{\lambda_1, \dots, \lambda_m\}$ that votes on each $\{(\{z_1^1, \dots, z_d^1\}, y^1), \dots, (\{z_1^n, \dots, z_d^n\}, y^n)\}$.
m	Number of causal inference estimate algorihtms.
G	DAG (Directed Acyclic Graph), where each $G^m \in \mathcal{G}$ represent a noisy estimate of causal structure.
$\mathcal{G}$	Space of graphs.
$\hat{G}$	Combination of Gs.
G^*	True causal structure (not observable).
$r(G)$	Graph embedding.
L	Label matrix.
θ	Accuracy of λ, where θ_m is accuracy of λ_m

Table 3: Glossary of variables and symbols used in this paper.

C. Automating Candidate Set Transformations

There are many situations where a human user may not be aware of a spurious pattern in the data. Had CelebA (Liu et al., 2015) not contained the appropriate annotation, a machine learning practitioner wishing to predict hair colors may have overlooked this feature's spurious correlation with gender. More generally, it is not always certain that users may have sufficient domain expertise to design hand-crafted transformations for candidate feature sets.

We describe a simple method to fully automate the candidate set transformation. An illustrative example is provided for the Waterbirds dataset (Sagawa et al., 2019). To discover patterns in the training images, we generate a caption for each image using a CLIP-based captioner (Mokady et al., 2021), then extract captions' keywords. We search each label (*waterbird* and *landbird*) on Wikipedia (Wikipedia contributors) and extract the keywords from the first resulting article's introduction section.

Spurious words are considered to be the top m most common caption keywords that do not occur in the article keywords. We next break each training image into non-overlapping $p \times p$ patches. If zero-shot CLIP (Radford et al., 2021) predicts any of the spurious words in a given patch with confidence greater than τ, the patch is covered. A resulting image from this process is depicted in Figure 3.

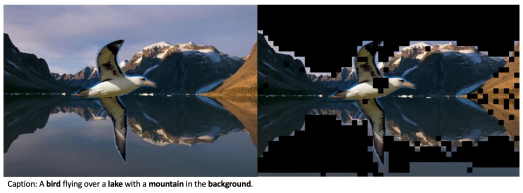

Figure 3: Automated transformation candidate set. Left, an image from Waterbirds (Sagawa et al., 2019) with its caption's keywords in bold. Right, a 10×10 patch is covered if zero-shot CLIP (Radford et al., 2021) predicts with confidence greater than 0.6 the presence of any word in *{tree, branch, forest, beach, rock, woman, ocean, field, man, background}*. These words were discovered to associate with non-causal information and therefore signal that the corresponding patch should be masked out.

We note that there are many potential ways to fully automate the candidate set transforms by taking advantage of ontologies and pretrained models. The proposed procedure requires only the label names and some form of task description, for instance that the dataset is comprised of images, allowing it to also be implemented in other settings outside the specific

example described above.

C.1. Result with Automated Candidate Set

We evaluate COMNIVORE when using the candidate set built from automated transformations described previously. We use a patch size p of 75, spurious word list length m of 10 and threshold τ of 0.6 on Waterbirds (Sagawa et al., 2019). As shown in Table 4, this configuration yields a similar performance to COMNIVORE with human-supervised LFs for $\mathcal{D}_{orig}$. AutoLF's score of 66.2 on $\mathcal{D}_{new}$ also improves by 7.6% on the baseline $\mathcal{D}_{new}$ Waterbirds results of Table 4.

We observed that the result is typically sensitive to the choices of threshold. We hypothesize that expanding the approach to larger ontologies will further close the gap to manual performance.

	REx	Manual Candidate Set	Automated Candidate Set
$\mathcal{D}_{new}$	58.6	71.0	66.2
$\mathcal{D}_{orig}$	95.5	90.7	93.1

Table 4: COMNIVORE-V with manually-built versus automated transformation-based candidate set and REx, the prior approach with best $\mathcal{D}_{new}$ performance. All scores are accuracy. The automated approach extracts human insights by combining the use of foundation models and ontologies, offering close-to-manual performance.

D. Theoretical Analysis

While the idea of fusing multiple noisy causal estimates is intuitively appealing, it is not clear whether we can expect this to work and under what conditions. This section is dedicated to showing how, in certain simple scenarios, the resulting estimates of the causal structure are useful.

Setup and Noise Distributions We will consider two scenarios. In the first, we have some candidate set of features $z_1, \ldots, z_k$, and we are interested in determining whether z_i is causal for output y. In other words, we are predicting a set $D \subseteq \{1, \ldots, k\}$. In the second scenario, we additionally take into account the causal structure, i.e., a directed acyclic graph G over the nodes $z_1, \ldots, z_k$.

We denote the causal recovery techniques by λ^a for $a = 1, \ldots, m$, so that $\lambda^a : \mathbb{R}^{k \times n} \to 2^{\{1,\ldots,k\}}$ in the first case, or $\lambda^a : \mathbb{R}^k \to \mathcal{G}$ in the second case. Recall that $\mathcal{G}$ is the set of DAGs on k nodes. Finally, we have a access to k environments, where for each environment, we observe n samples of the features $z_1, \ldots, z_k$.

Each causal estimation function λ^a may fail in a variety of ways; this may be because the underlying assumptions are not met, or because of noise, or for some other reason. The outcome of such noise is either a predicted set D not equal to the true D^*, or a predicted graph G not equal to G^*. We will model the noise in the estimation approaches with the following model inspired by (Shin et al., 2022):

$$P_\theta(\lambda^1, \ldots, \lambda^m | D^*) = \frac{1}{Z} \exp\left(-\sum_{a=1}^{m} \theta_a d_H(\lambda^a, D^*) \right), \tag{2}$$

where Z is the normalizing partition function, d_H is the Hamming distance, and $\theta = [\theta_1, \ldots, \theta_m]^T$ is a vector of parameters. For sets, d_H is simply the size of the symmetric diference. We can also operate in the second scenario by switching D^* to G^* on both sides of (2). In this case, the Hamming distance over graphs counts the number of differences in edges. Note how the model works: if θ_a is large, then the probability mass is significantly reduced even for a small distance between the prediction and the true causal model; this implies that the quality of the approach λ^a is high. If θ_a is small, then even a large distance does not significantly reduce the probability, so λ^a is low-quality.

Note that richer models are possible; for example, we could replace the graph Hamming distance with the *interventional* distance as in (Peters & Bühlmann, 2015). The advantage of the exponential family model above is that it is tractable without requiring significant specifications on the underlying causal model.

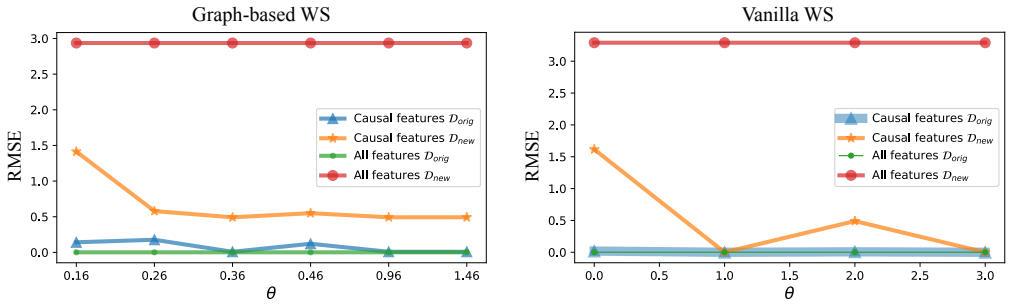

Figure 4: Synthetic Experiments. Errors on $\mathcal{D}_{orig}$ and $\mathcal{D}_{new}$ when using only causal features converges to lower bound (error on $\mathcal{D}_{orig}$ using all features) with increasing θ.

Estimating Qualities and Performing Fusion The main challenge is how to estimate $\theta_1, \ldots, \theta_m$. The two techniques in Algorithm 1 work for these two scenarios. We show that the second approach has consistent estimation of θ in terms of the number of environments k.

Theorem D.1. *Suppose $\lambda^a, \ldots, \lambda^m$ are distributed according to (2) and we have access to k training environments. Using vanilla weak supervision to estimate $\hat{\theta}$, we have that $\mathbb{E}[\|\hat{\theta} - \theta^*\|] \leq O(1/\sqrt{k})$.*

This implies that, given sufficiently many environments, the weights we learn for use in Algorithm 1 reflect the underlying quality of the causal estimation functions.

D.1. Theoretical Analysis Experiment

We evaluate a key claim from our theoretical characterization outlined above. We expect that as the values of the θ parameter vector are larger, the quality of the causal estimation functions improves, and that our resulting algorithm produces causal features that perform well in a new environment.

We validate this notion using a synthetic dataset reflecting a simple linear regression setup. In the original environment $\mathcal{D}_{orig}$, the label is a function of all of the features, while in the new environment $\mathcal{D}_{new}$, the label is a function of only a subset of features—and the remaining features have a significantly different distribution from their counterparts in $\mathcal{D}_{orig}$.

The results are shown in Fig. 4. We swept the average magnitude of θ, used our two approaches based on graph-based WS (left) and vanilla WS (right), trained a linear regression end model, and measured the root mean squared error (RMSE). As expected, using all features results in very good error in $\mathcal{D}_{orig}$ (green curve) and very poor error in $\mathcal{D}_{new}$ (red curve). Applying our causal approaches resulted in nearly-as-good $\mathcal{D}_{orig}$ performance (blue curve), and vastly improved $\mathcal{D}_{new}$ performance (yellow curve). As we hoped, the error of this curve generally decreases with improved quality estimates (i.e., larger θ). Additionally, we note that the vanilla WS approach, while slightly noisier, produces a smaller final error. This suggests a closer analysis of the two approaches would be useful.

E. Additional Algorithmic Details

E.1. Projection to Low-Dimensional Space

We use sklearn's (Buitinck et al., 2013) implementation of feature agglomeration, an unsupervised dimensionality reduction technique that uses agglomerative clustering to group together features that look very similar (Steinbach et al., 2000). Our method also works with any dimensionality reduction technique like PCA (F.R.S., 1901). We chose feature agglomeration because it provides an automatic mapping from higher to lower dimensional space, thus eliminating the need to manually set thresholds for the components.

E.2. L Matrix Computation

In Vanilla WS fusion method, for each z_i, we construct a $k \times m$ label matrix L, where k is the number of environments we have access to and m is the number of causal estimation functions. Formally, let $A^{(b,j)}$ be the $d \times d$ adjacency matrix

representation of G^j from bth environment, and the label node is the dth node in G^j, each entry of L is defined by:

$$L_{bj} = \begin{cases} 1, & \text{if } A_{id}^{(b,j)} = 1 \\ 0, & \text{if } A_{id}^{(b,j)} = 0 \\ -1, & \text{if } A_{id}^{(b,j)} = -1 \end{cases}, \quad b \in 1 \dots k, \quad j \in 1 \dots m, \quad i \in 1, \dots d-1 \tag{3}$$

F. Experimental Details

F.1. Dataset Details

Table 5 shows details on train/dev/test splits for each dataset, as well as the number of smallest group samples in distribution shift datasets. All splits are following the default provided by WILDS benchmark (Koh et al., 2020).

Dataset	N_{train}	N_{dev}	N_{test}	$N_{smallest}$
Waterbirds	4,795	1,199	5,794	56
CelebA	162,770	19,867	19,962	1,387
Camelyon17	302,436	33,560	85,054 ($\mathcal{D}_{new}$) + 34,904 ($\mathcal{D}_{orig}$)	N/A

Table 5: Details for each dataset. N_{train}: The size of the unlabeled training set. N_{dev}: The size of the labeled dev set. N_{test}: The size of the held-out test set. $N_{smallest}$: The size of the smallest group for subpopulation shift datasets.

We evaluate COMNIVORE on three datasets in the WILDS benchmark (Koh et al., 2020). In subpopulation shift, **Waterbirds**'s spurious correlation occurs between label $Y = \{landbirds, waterbirds\}$ and background attribute $\mathcal{A} = \{land, water\}$.; **CelebA** celebrity faces dataset (Liu et al., 2015) has spurious correlation between the hair color label $Y = \{blond, dark\}$ and the gender attribute $\mathcal{A} = \{male, female\}$; In domain generalization, **Camelyon17** (Bándi et al., 2019)'s domain shift is the different hospitals where training and test samples are collected.

F.2. ColorMNIST

We construct our implementation of synthetic colored version of the MNIST dataset (Deng, 2012). In contrast with IRM and REx, we do not collapse the classes (i.e., $y = 0$ for digits $0 - 4$ and $y = 1$ for digits $5 - 9$). Instead, we directly use the digits 0 vs 1. More specifically, we take MNIST subsets of digits 0 and 1, assign a color to each digit, and flip the color on $\mathcal{D}_{test}$. We use the default train/dev/test splits provided by MNIST.

We also note that in IRM's version of ColorMNIST, IRM achieves 70.8% accuracy on $\mathcal{D}_{orig}$ and 66.9% $\mathcal{D}_{new}$; and in REx's implementation, REx achieves 71.5% accuracy on $\mathcal{D}_{orig}$ and 68.7% on $\mathcal{D}_{new}$. Our main experimental table contains the values we obtained on our version of the dataset.

F.3. Manual Candidate Sets

This section details the construction of manual candidate sets used in our experiments. The original and transformed images are shown in figure 5. For Waterbirds and celebA, segmentation is done using Pytorch's off-the-shelf DeepLabV3 model (Chen et al., 2017). For Camelyon17, the candidate set generated is the gaussian blurred version of the original images, generated using PIL's Gaussian Blur filter (Umesh, 2012). For ColorMNIST, the candidate set used is the original images and the black and white version. Table 6 details the candidate sets used for best numbers reported.

F.4. Hyperparameters and Model Selection

F.4.1. END CLASSIFICATION MODEL

Experiments were done three times, and we reported an average of three runs. Models are selected based on the best performance on the dev set (and OOD dev set for Camelyon17). Experiments are conducted using two NVIDIA RTX A4000 GPUs. For all datasets, we train a 2-layer MLP with 512 hidden dimensions. Best hyperparameters are reported in table 7. All models are trained using 0.9 momentum and 0.1 $l2$ regularization penalty. Training epochs are set until 500, and we picked the checkpoint with the highest dev performance.

Dataset	Candidate Sets Used
Waterbirds	{Segmentation}
CelebA	{Original, Segment + Crop Bottom, Segment + Crop Face}
Camelyon17	{Original, Gaussian Blur}
ColorMNIST	{Original, bw}

Table 6: Candidate Set used for each dataset. Original images can also be a candidate set (e.g., in celebA and Camelyon17).

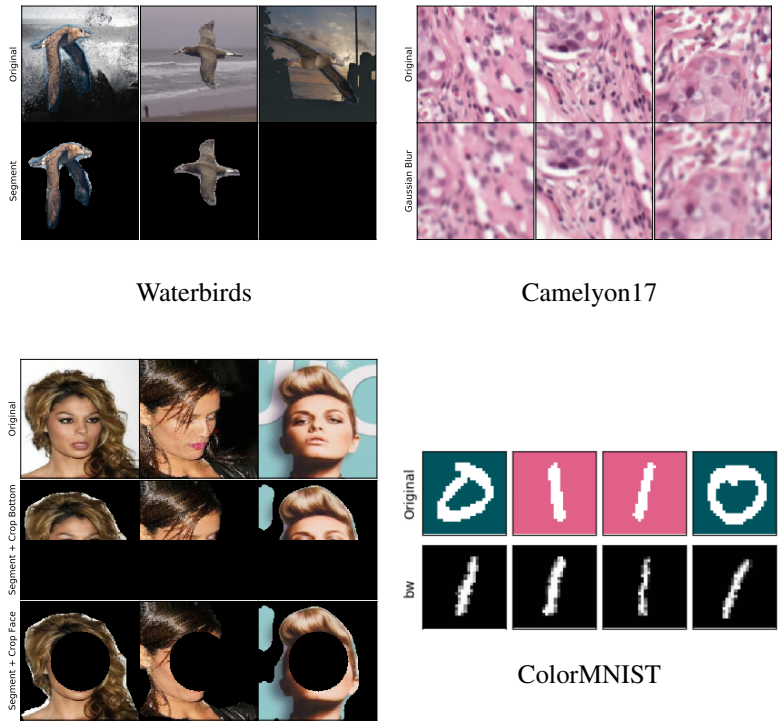

Figure 5: Candidate Sets

Dataset	$dim(z)$	Learning rate	Batch size
Waterbirds	5	$5e-4$	32
CelebA	3	$1e-4$	16
Camelyon17	3	$5e-4$	1280
ColorMNIST	10	$1e-4$	1280

Table 7: Best hyperparameters. $dim(z)$ is the lower dimension space used to project features.

F.5. Baseline Implementations

IRM A ResNet50 is trained using the IRM implementation from the WILDS benchmark (Koh et al., 2020). Reported results are averaged across three runs, using the hyperparameters yielding highest average accuracy on the dev set in any epoch. In real-world applications, the best strategy would often be to select a model that balances somewhere in between maximizing average and worst-group accuracy as determined by domain experts. In this work, however, we choose to report epochs that maximize average accuracy without regard to worst-group accuracy in order to establish a uniform, unbiased

method to select the “best" hyperparameters and performance metrics.

The maximum possible number of epochs is 200. Momentum of 0.9, IRM λ of 100 and penalty annealing iterations of 500 are used for all datasets. Learning rate and batch size are reported in Table 8.

Dataset	Learning rate	Batch size
Waterbirds	$1e-5$	128
CelebA	$1e-6$	96
ColorMNIST	$1e-7$	64

Table 8: Best hyperparameters for IRM.

We do not report the hyperparameters for Camelyon, because we report IRM result on Cameyon based on the WILDS leaderboard (WILDS).

REx We train a 2-layer MLP with 256 hidden dimensions using REx implementation for all datasets. Experiments were done three times, and we reported an average of three runs. The maximum possible number of epochs is 500, and we picked the checkpoint with the highest performance on dev set (and OOD dev set for Camelyon17). Penalty annealing iterations of 100 are used for all the datasets. Other best hyperparameters are reported in Table 9.

Dataset	Learning rate	Batch size	β
Waterbirds	$1e-3$	2000	10000
CelebA	$3e-3$	4000	100
Camelyon17	$3e-3$	32	100
ColorMNIST	$3e-5$	1000	10000

Table 9: Best hyperparameters for REx. β is assigned weight for variance of risks in REx risk function used to balance between reducing average risk and enforcing quality of risks(Krueger et al., 2021).