Causal Omnivore: Fusing Noisy Estimates of Spurious Correlations

Dyah Adila[†] Sonia Cromp[†] Sicheng Mo[⋆] Frederic Sala[†]

[†]University of Wisconsin-Madison
{adila, cromp}@wisc.edu
{fredsala}@cs.wisc.edu
[⋆]University California, Los Angeles
{smo3}@ucla.edu

December 5, 2022

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. Theoretically, we show that our technique can recover causal structures under certain conditions. 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]

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 issue is not new. A classic (likely apocryphal) piece of lore describes military

[1]Our code is available at https://github.com/SprocketLab/comnivore

researchers training a classifier for tank detection that instead learns to predict weather patterns—as cloudy-day training images lack any positive labels. 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 [3, 20, 28]. These techniques are promising but suffer from multiple flaws. For example, invariant risk minimization (IRM) [3] 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 [33]. Experimentally, none of these techniques are known to work in all cases [13]. 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. Such a technique must have two key properties:

1. **Fusing Noisy Causal Estimates.** Causal feature estimation approaches rely on different assumptions and are affected by noise and variation differently. This leads to noisy and contradictory estimates of causal features (or causal structures among the features) that need to be reconciled.
2. **Obtaining Inexpensive Sources of Causal Signal.** Humans often have an easy time determining causal information. An ideal system can either integrate human-based specifications of causal features, or extract what are likely human-like signals that exist inside knowledge bases, pretrained models, or other resources.

s

We propose COMNIVORE a system that takes a step towards satisfying these properties. First, it enables the use of multiple sources to generate potential candidate feature sets. In particular, it allows for simple ways for human 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 [32], 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 the detected features with vanilla empirical risk minimization. We validate COMNIVORE empirically, showing that it outperforms competing end-to-end baselines

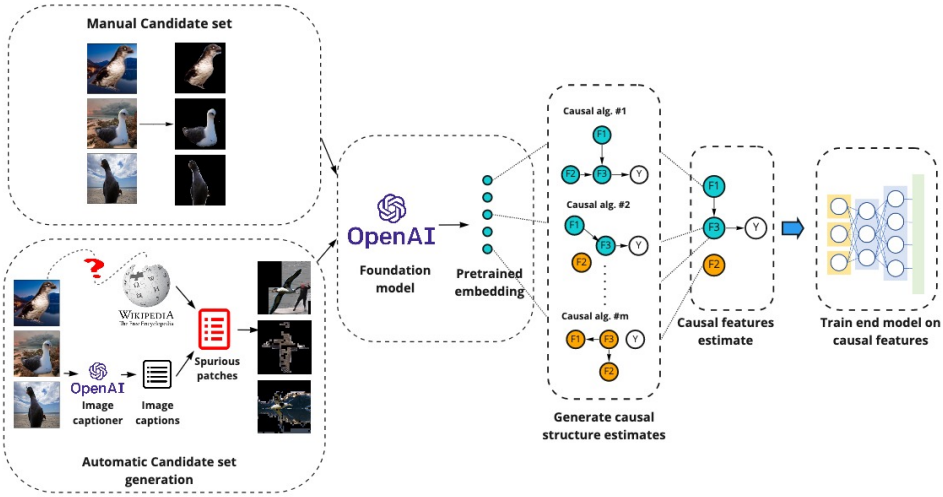

Figure 1: COMNIVORE seeks to find causal features through a two-step process. It flexibly pulls together multiple sources of candidate feature sets (left), such as pretrained model embeddings, features from hand-crafted transformations, or automatically-learned augmentations. It runs these candidate 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).

such as IRM, while improving on ERM by 37.9% on three benchmark datasets (Table 2).

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. We briefly describe two sets of approaches. 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, PC [38], FCI [38], and Greedy Equivalence Search [9] assume the absence of certain conditions on the latent confounders between features. Grow-Shrink (GS) [23], Incremental Association Markov Blanket (IAMB) [40], Interleaved IAMB [45], and Exact Search [37] require the underlying model to have a certain Bayesian structure. More recent optimization-based methods [47] [48] are limited by optimization constraints. These assumptions thus limit their accuracy when applied to complex and high-dimensional data.

A second set of techniques attempt to use multiple distinct training environments to

obtain causal features. The idea is that a good learned representation should be invariant to changes in environment [3, 28]. On the other hand, the resulting optimization problem is difficult, leading to the need to use approximations that may not perform well in practice. In fact, even theoretically a huge number of environments may be needed to ensure that one popular approach, invariant risk minimization [3], outperforms ERM [33] .

Weak Supervision Weak supervision is a set of techniques that are used to construct labeled training sets [16, 30, 32] 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 environemnts.

Our goal is to learn a model $f(x)$ that performs well in new scenarios. Typically this means that it is capable of domain generalization, i.e., it generalizes well to test distributions $\mathcal{D}_{test}$ that are disjoint from the training distributions $\mathcal{D}_{train}$. 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

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 some combination of the above. 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 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 [42] [49]. 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 Section 4, we will 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. An example is shown in Figure 3.

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 $\mathcal{G}$. Our approach treats the causal inference algorithms [9, 23, 37, 38, 40, 45, 47, 48] like labeling functions in weak supervision as they output noisy estimates of causal structures. Formally, given m causal algorithms, the output of each algorithm λ^a is described by

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

One challenge is that such algorithms often have high complexity, sometimes superexponential in the number of features. 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. At the end of this step, we have m DAGs $\{G^1, \ldots, G^m\}$ per candidate set and environment.

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 ∈ C do
    for λ_i ∈ λ do
        Generate causal structure estimates G^i
    for G^i ∈ 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 (2) to get Ĝ
            collect all z_i ∈ 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 Ŷ_i ± 1
```

Result: Singleton set of configurations .

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 [36], 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 $\{1, \ldots, m\}$:

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

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 $w_a = \frac{1}{\|r(G^a) - r(G^*)\|^2})$.

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

$$\hat{G} = \arg\min_{G \in \mathcal{G}} \frac{1}{m} \sum_{j=1}^{m} w_j d_H(G, G^j) \tag{2}$$

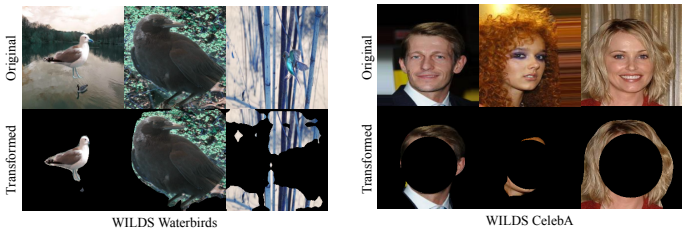

Figure 2: Manual transformation candidate set examples. Humans can easily observe that background is not causally linked to bird species (left) and that gender is not linked to hair (right). These insights can be translated into simple augmentations that help remove potentially spurious features: segmentation to remove background (left) and facial features (right).

where d_H denotes the Hamming distance. Note that we compute the mean in the original DAG space, not in the embedding space. We use simulated annealing [17], 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*.

Inspired by [16], 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., [16, 31, 32].

4 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 [22] 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 [34]. To discover patterns in the training images, we generate a caption for each image using a CLIP-based captioner

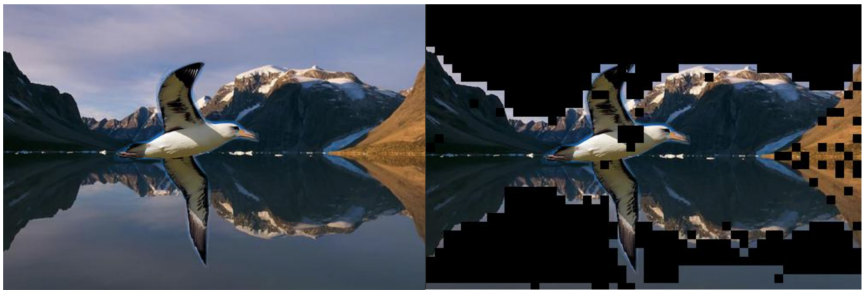

Figure 3: Automated transformation candidate set. Left, an image from Waterbirds [34] with its caption's keywords in bold. Right, a 10×10 patch is covered if zero-shot CLIP [29] predicts with confidence greater than 0.6 the presence of any word in *{tree, branch, forest, beach, rock, woman, ocean, field, man, background}*. These words were discovered to associate with non-causal information and therefore signal that the corresponding patch should be masked out.

[24], then extract captions' keywords. We search each label (*waterbird* and *landbird*) on Wikipedia [43] 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 [29] 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.

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.

5 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 [36]:

$$P_\theta(\lambda^1, \ldots, \lambda^m | D^*) = \frac{1}{Z} \exp\left(- \sum_{a=1}^{m} \theta_a d_H(\lambda^a, D^*) \right), \tag{3}$$

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 (3). 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 [27]. The advantage of the exponential family model above is that it is tractable without requiring significant specifications on the underlying causal model.

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.

Suppose $\lambda^a, \ldots, \lambda^m$ are distributed according to (3) 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.

6 Experiments

This section validates the following claims about COMNIVORE:

- **Performance (Section 6.1):** We compare COMNIVORE against two automated causal discovery techniques: IRM and REx [20]. We show that COMNIVORE outperforms baseline end-to-end approaches on unseen environment $\mathcal{D}_{new}$ with comparable performance on the original environment $\mathcal{D}_{orig}$ on both subpopulation shift and domain generalization datasets.

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-CLIP		ERM-CLIP(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.

- **AutoLF (Section 6.2):** We demonstrate for Waterbirds [34] that COMNIVORE with AutoLF performs identically to human-based LFs on $\mathcal{D}_{orig}$ and outperforms prior works on $\mathcal{D}_{new}$, without any human supervision.
- **Theory (Section 6.3):** Aligned with the theoretical analysis presented in section 5, we show empirical results on synthetic data that training vanilla ERM model with causal features from COMNIVORE strictly outperforms vanilla ERM with all features on $\mathcal{D}_{new}$, and is equal on $\mathcal{D}_{orig}$ given sufficiently high-quality estimation functions, as reflected by θ_a.
- **Ablations (Section 6.4):** COMNIVORE's significant lift on $\mathcal{D}_{new}$ while retaining good performance on $\mathcal{D}_{orig}$ is produced by our careful selection of causal features, rather than simply removing spurious parts in $\{C_1, C_2, \ldots, C_b\}$ and using foundation model embeddings. We show this by comparing COMNIVORE with training vanilla ERM and baseline approaches using the foundation model embeddings of $\{C_1, C_2, \ldots, C_b\}$.

Datasets We evaluate COMNIVORE on three datasets in the WILDS benchmark [18]. In subpopulation shift, **Waterbirds** combines bird images from the Caltech-UCSD Birds-200-2011 (CUB) dataset [42] with backgrounds from the Places dataset [49], with spurious correlation occurs between label $Y = \{\textit{landbirds}, \textit{waterbirds}\}$ and background

	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 3: 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.

attribute $\mathcal{A} = \{land, water\}$. There are $n = 4,795$ training examples and 56 in the smallest group (waterbirds on land); **CelebA** celebrity faces dataset [22] has spurious correlation between the hair color label $Y = \{blond, dark\}$ and the gender attribute $\mathcal{A} = \{male, female\}$. There are $n = 162,770$ training examples with 1387 in the smallest group (blond-haired males); In domain generalization, **Camelyon17** [6]'s task is to identify tumor in medical images. The domain shift is the different hospitals where training and test samples are collected. There are $n = 302,436$ training samples and $n = 85,054$ test samples. In addition, we also evaluate on **ColorMNIST**, where spurious correlations between digits and color are artificially created, similar to the task used in [3] and [20]. **Pre-trained embeddings** We use pre-trained CLIP embeddings [29].

6.1 Performance Comparisons

We compare COMNIVORE with baseline approaches (IRM [3] and REx [20]), 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 [18]. 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.

6.2 Automated Candidate Set Transformation

Next, we evaluate COMNIVORE when using the candidate set built from automated transformations described in Section 4. We use a patch size p of 75, spurious word list length m of 10 and threshold τ of 0.6 on Waterbirds [34]. As shown in Table 3, this configuration yields a similar performance to COMNIVORE with human-supervised

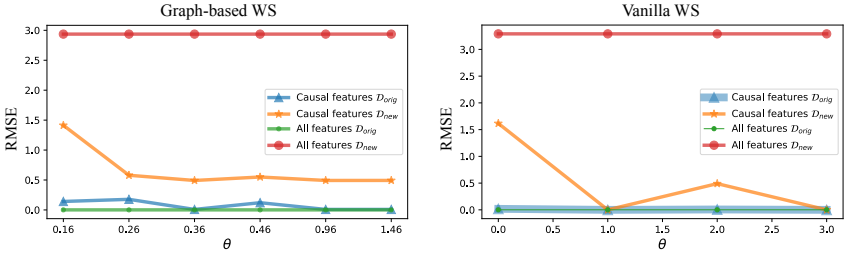

Figure 4: Synthetic Experiments. Errors on $\mathcal{D}_{orig}$ and $\mathcal{D}_{new}$ when using only causal features converges to lower bound (error on $\mathcal{D}_{orig}$ using all features) with increasing θ.

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 3.

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.

6.3 Synthetics and Theoretical Characterization

We evaluate a key claim from our theoretical characterization in Section 5. 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.

6.4 Ablations

We investigate the source of COMNIVORE's performance lift. 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 features selection.

7 Conclusion

We introduced COMNIVORE, a system for efficiently discovering causal features for downstream model training. It operates by flexibly integrating multiple sources of potential causal signal and fusing together noisy estimates of causal structures. We showed how to acquire candidate feature sets through a variety of means—via pretrained model embeddings, hand-crafted augmentations that encode human insights into causal relationships, or automated transformations that take advantage of preexisting signal in ontologies and foundation models. The causal feature sets are further refined by running suites of causal feature estimation methods and fusing their outputs. We validated COMNIVORE empirically, showing how that it can outperform end-to-end methods like IRM.

References

[1] Faruk Ahmed, Yoshua Bengio, Harm van Seijen, and Aaron Courville. Systematic generalisation with group invariant predictions. In *International Conference on Learning Representations*, 2020.

[2] Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *International Conference on Machine Learning*, pages 145–155. PMLR, 2020.

[3] Martin Arjovsky, Leon Bottou, and David Lopez-Paz Ishaan Gulrajani. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[4] Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

[5] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the

scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[6] Péter Bándi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, Quanzheng Li, Farhad Ghazvinian Zanjani, Svitlana Zinger, Keisuke Fukuta, Daisuke Komura, Vlado Ovtcharov, Shenghua Cheng, Shaoqun Zeng, Jeppe Thagaard, Anders B. Dahl, Huangjing Lin, Hao Chen, Ludwig Jacobsson, Martin Hedlund, Melih Çetin, Eren Halıcı, Hunter Jackson, Richard Chen, Fabian Both, Jörg Franke, Heidi Küsters-Vandevelde, Willem Vreuls, Peter Bult, Bram van Ginneken, Jeroen van der Laak, and Geert Litjens. 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.

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017.

[8] David Maxwell Chickering. Learning bayesian networks is np-complete. In *Learning from data*, pages 121–130. Springer, 1996.

[9] David Maxwell Chickering. 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`.

[10] Max Chickering, David Heckerman, and Chris Meek. Large-sample learning of bayesian networks is np-hard. *Journal of Machine Learning Research*, 5: 1287–1330, 2004.

[11] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021.

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

[13] Yana Dranker, He He, and Yonatan Belinkov. 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.

[14] John Duchi, Peter Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv preprint arXiv:1610.03425*, 2016.

[15] Karl Pearson F.R.S. 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.

[16] Daniel Y. Fu, Mayee F. Chen, Frederic Sala, Sarah M. Hooper, Kayvon Fatahalian, and Christopher Ré. Fast and three-rious: Speeding up weak supervision with triplet methods. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, 2020.

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

[18] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. *CoRR*, abs/2012.07421, 2020. URL `https://arxiv.org/abs/2012.07421`.

[19] Masanori Koyama and Shoichiro Yamaguchi. Out-of-distribution generalization with maximal invariant predictor. 2020.

[20] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation. In *Proc. of the 38th International Conference on Machine Learning (ICML)*, 2021.

[21] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021.

[22] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015.

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

[24] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.

[25] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33:20673–20684, 2020.

[26] Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*, 2020.

[27] Jonas Peters and Peter Bühlmann. Structural intervention distance for evaluating causal graphs. *Neural Computation*, 27(3), 2015.

[28] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by

using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society Series B*, 78(5), 2016.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL `https://arxiv.org/abs/2103.00020`.

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

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

[32] Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 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.

[33] Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

[34] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[35] Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

[36] Changho Shin, Winfred Li, Harit Vishwakarma, Nicholas Carl Roberts, and Frederic Sala. Universalizing weak supervision. In *International Conference on Learning Representations (ICLR)*, 2022.

[37] Tomi Silander and Petri Myllymaki. A simple approach for finding the globally optimal bayesian network structure. *arXiv preprint arXiv:1206.6875*, 2012.

[38] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000.

[39] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. *Proceedings of the International KDD Workshop on Text Mining*, 06 2000.

[40] Ioannis Tsamardinos, Constantin F Aliferis, Alexander R Statnikov, and Er Statnikov. Algorithms for large scale markov blanket discovery. In *FLAIRS conference*, volume 2, pages 376–380. St. Augustine, FL, 2003.

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

[42] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

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

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

[45] Sandeep Yaramakala and Dimitris Margaritis. Speculative markov blanket discovery for optimal feature selection. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 4–pp. IEEE, 2005.

[46] Michael Zhang, Nimit S. Sohoni, Hongyang R. Zhang, Chelsea Finn, and Christopher Ré. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations, 2022. URL https://arxiv.org/abs/2203.01517.

[47] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. DAGs with NO TEARS: Continuous Optimization for Structure Learning. In *Advances in Neural Information Processing Systems*, 2018.

[48] Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Learning sparse nonparametric DAGs. In *International Conference on Artificial Intelligence and Statistics*, 2020.

[49] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 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, provide a glossary containing key terminology, introduce additional details into our algorithm and theoretical claims, then give extra experimental details and results.

A Related Work

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

Invariant learning methods such as IRM [3], REx [20], and a multitude of similar works [1, 2, 11, 26] 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) [11] and Predictive Group Invariance (PGI) [1] 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) [34], which divides the data into explicit groups and then trains them to directly minimize the worst group-level error among these groups. Similarly, Fish [35] and Inter-environment Gradient Alignment (IGA) [19] 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 [4, 14]. More recent methods [21, 25, 46] 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 [9, 23, 37, 38, 40, 45, 47, 47] 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 [8, 10]. 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 [16, 30, 32]. This technique is vastly explored for binary classification problems. Recently, [36] 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 4 below.

Symbol	Used for
x	Input data point $x \in X$.
y	Ground truth label $y \in Y$.
$\mathcal{D}$	Data distribution. 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 off-the-shelf pretrained or foundation models.
C	Candidate sets.
k	number of environments
z	Features $z = \{z_1, \ldots z_d\}$, where each z_i is feature vector component.
n	Number of data points.
d	Number of features (i.e., dimension of feature vector).
λ	Causal inference algorithms $\lambda = \{\lambda_1, \ldots, \lambda_m\}$ that vote on each set of features $\{(\{z_1^1, \ldots, z_d^1\}, y^1), \ldots, (\{z_1^n, \ldots, z_d^n\}, y^n)\}$.
m	Number of causal inference estimate algorihtms.
G	DAG where each $G^m \in \mathcal{G}$ represents 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 parameter of λ, where θ_m is accuracy of λ_m

Table 4: Glossary of variables and symbols used in this paper.

C Additional Algorithmic and Theory Details

Next we give some additional algorithmic and theory details.

C.1 Projection to Low-Dimensional Space

We use sklearn's [5] implementation of feature agglomeration, an unsupervised dimensionality reduction technique that uses agglomerative clustering to group together features that look very similar [39]. Our method also works with any dimensionality reduction technique like PCA [15]. 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.

C.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 $(d+1) \times (d+1)$ adjacency matrix representation of G^j from bth environment, and the label node is the $d+1$th node in

G^j, each entry of L is defined by:

$$L_{bj} = \begin{cases} 1, & \text{if} \quad A_{id}^{(b,j)} = 1 \\ 0, & \text{if} \quad A_{id}^{(b,j)} = 0 \\ -1, & \text{if} \quad A_{id}^{(b,j)} = -1 \end{cases}, \quad b \in 1 \dots k, \quad j \in 1 \dots m, \quad i \in 1, \dots d-1 \quad (4)$$

C.3 Theoretical Details

The proof of Theorem 5 follows directly by applying the results in [16] (Theorem 1).

D Extended Experimental Details

D.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 [18].

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.

As an additional point of comparision, we note that EIIL [11] achieves a 69.7% accuracy on $\mathcal{D}_{orig}$ and 78.8% $\mathcal{D}_{new}$ in Waterbirds dataset.

D.2 ColorMNIST

We construct our implementation of synthetic colored version of the MNIST dataset [12]. 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.

D.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 [7]. For Camelyon17, the candidate set generated is the gaussian blurred version of the original images, generated using PIL's Gaussian Blur filter [41]. 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. v

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).

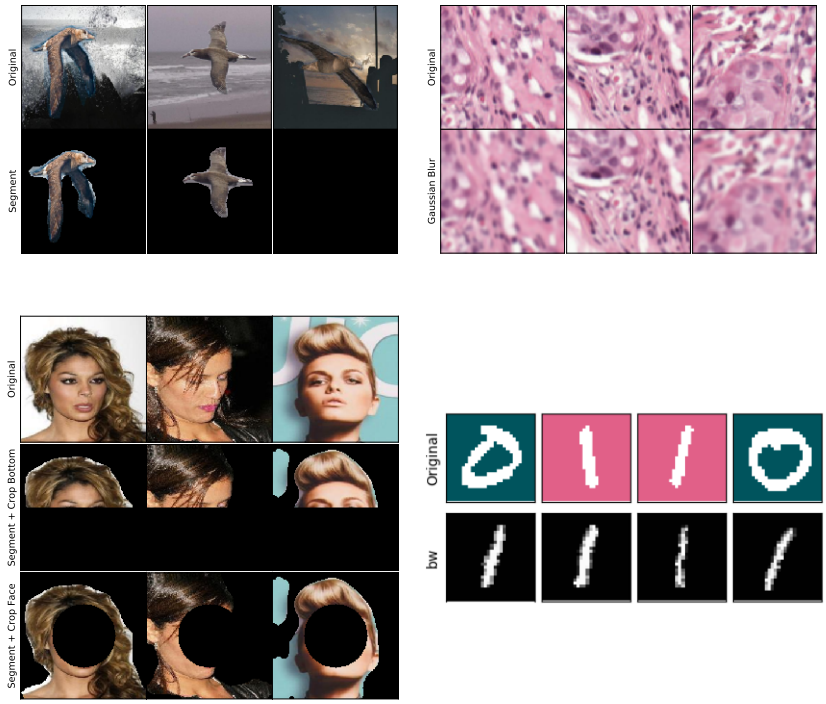

Figure 5: Candidate Sets

D.4 Hyperparameters and Model Selection

D.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	$dim(z)$	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.

D.4.2 Automated Candidate Set Generation

We choose the word count m, patch size p and threshold τ hyperparameters yielding highest average dev set accuracy, then report the performance on the test set averaged across three runs.

D.5 Baseline Implementations

IRM A ResNet50 is trained using the IRM implementation from the WILDS benchmark [18]. 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.

We do not report the hyperparameters for Camelyon, because we report IRM result on Cameyon based on the WILDS leaderboard [44].

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

Table 8: Best hyperparameters for IRM.

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
ColoredMNIST	$3e-5$	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[20].