Confidence-Based Model Selection: 
When to Take Shortcuts in Spurious Settings

Annie S. Chen\textsuperscript{1}, Yoonho Lee\textsuperscript{1}, Amrith Setlur\textsuperscript{2}, Sergey Levine\textsuperscript{3}, Chelsea Finn\textsuperscript{1} 
Stanford University\textsuperscript{1}, Carnegie Mellon University\textsuperscript{2}, UC Berkeley\textsuperscript{3} 
asc8@stanford.edu

Abstract

Effective machine learning models learn both robust features that directly determine the outcome of interest (e.g., an object with wheels is more likely to be a car), and shortcut features (e.g., an object on a road is more likely to be a car). The prevailing sentiment in the robustness literature is to avoid such correlative shortcut features and learn robust predictors. However, while robust predictors perform better on worst-case distributional shifts, they often sacrifice accuracy on majority subpopulations. In this paper, we argue that shortcut features should not be entirely discarded. Instead, if we can identify the subpopulation to which an input belongs, we can adaptively choose among models with different strengths to achieve high performance on both majority and minority subpopulations. We propose COnfidence-baSed MOdel Selection (COSMOS), where we observe that model confidence can effectively guide model selection. Notably, COSMOS does not require any target labels or group annotations, either of which may be difficult to obtain or unavailable. We evaluate COSMOS on four datasets with spurious correlations, each with multiple test sets with varying levels of distribution shift. We find that COSMOS achieves 2-5\% lower average regret across all subpopulations, compared to using only robust predictors or other model aggregation methods.

1 Introduction

Datasets often exhibit spurious correlations, where a classifier based on a shortcut feature that is predictive on the training data can be misled when faced with a distribution shift in inputs [19]. For example, consider the task of classifying cows or camels, where most cows in the source distribution have grass backgrounds while most camels have sand backgrounds. Standard models trained with empirical risk minimization (ERM) optimize for average performance and may learn a predictor that relies on the background of the image for this task. This reliance on shortcut features can result in subpar performance when the model is tested on data distributions with a larger representation from regions of minority subpopulations.

The robustness literature has traditionally entirely discarded shortcut features, and the term carries a negative connotation. In particular, recent works have proposed various debiasing methods to counteract the issues arising from shortcut features [54, 47, 41]. The aim of these debiasing methods is to learn an “invariant predictor”—a function that is invariant to changes in features that bear no causal relationship with the label. However, these methods often result in lower average accuracy compared to models that use shortcut features, as they necessarily sacrifice accuracy on majority subpopulations. In the cow/camel task, while we do want our predictors to focus on the animal, an invariant predictor may entirely discard the background information. This information can still be valuable, particularly since the animal itself may be difficult to classify in some images.

Figure 1: **Confidence-based Model Selection (CosMoS)**. We start with multiple base models with diverse performance characteristics. After clustering test embeddings, CosMoS routes each cluster to the base model that has highest average predictive confidence for that cluster. The final aggregated classifier leverages the strengths of each model, achieving higher performance.

As they both have strengths and weaknesses on different subpopulations, we argue for viewing shortcut and invariant classifiers on equal footing, where both are experts but on different regions of the input space. This shift in mindset may lead to a more optimal strategy for these distribution shifts – provided we can discern when to include or exclude these features in our predictions, we can exploit the advantages of both types of models in a complementary way. Prior work has studied this phenomenon in human decision-making [62, 59, 20]: sometimes, adopting informal mental shortcuts is more effective than making a complete logical analysis based on all available information. In this paper, we demonstrate the benefits of leveraging the strengths of both shortcut and debiased classifiers in achieving high performance on both majority and minority subpopulations.

Instead of combining models as done in typical ensemble methods, we use different models for different inputs. We hypothesize that selectively employing an appropriate classifier for different inputs maintains high performance more effectively on both majority and minority subpopulations compared to relying on a single predictor. In practice, we find that simply using model confidence can effectively determine which model to choose for different inputs, and we propose COSMOS (ConfiDence-based MOdel Selection) which, given multiple base classifiers, selectively employs an appropriate classifier for each input in the test set. Since model confidence can be noisy on a single data point, we propose clustering inputs before model assignment as a means of variance reduction, using average model confidence on that cluster to determine which classifier to use. Our approach is general in that it may be applied to any set of base classifiers, including ones in different loss basins, and it does not require access to the weights of the base classifiers or to any group annotations. Furthermore, many prior works that study robustness to spurious correlations subtly require access to labeled target distribution data when tuning hyperparameters, whereas our approach does not require any access to any labeled data from the target domain. A summary is given in Figure 1.

We evaluate our method on four datasets with spurious correlations, each with many test sets with various levels of data distribution shift. We consider many test sets because when using models in the real world, we do not know which test set the model will be evaluated on. Thus, we want models to perform reliably on a wide range of subpopulations, including those that are over- and under-represented during training. In other words, we aim to satisfy multiple desirable objectives – achieving high accuracy on unseen data from both the source distribution and the worst-group target distributions. Our results show that our method achieves 2-5% lower average regret across the subgroups of the input space compared to using a single predictor or methods that aggregate multiple classifiers. In particular, our approach can achieve high accuracy on minority subpopulations without sacrificing performance on majority subpopulations. Furthermore, we show that our method can also be used for other tasks where we need to choose the best model for a test distribution. For example, by taking the candidate that is chosen the most by CosMoS, we can use CosMoS to do hyperparameter tuning without any labels from the desired test distribution, and to our knowledge, we are the first method to do so.

2 **Confidence-Based Model Selection (CosMoS)**

In this section, we describe CosMoS, a framework for leveraging multiple models that adaptively chooses an appropriate model for each input. In the previous section, we described our problem setting, where we consider spurious settings, where there exist shifts in the frequencies of data subpopulations.
There are multiple good classifiers that perform well on \( p_S(x, y) \) but may perform differently on different subpopulations determined by \( z \) and therefore also on unknown test distributions \( p_T(x, y) \). Hence, to leverage the strengths of these classifiers, if we can determine which \( z \) each test input belongs to, we can just choose the corresponding classifier that performs best for that subpopulation. However, we do not have any labels on \( z \) and do not even know how many different groups there are. To avoid needing to explicitly classify the \( z \) for each datapoint, we instead estimate which datapoints have similar \( z \) and use confidences of the classifiers to implicitly indicate a subpopulation identity.

We are given base classifiers \( f_1, f_2, \ldots, f_K \) that are trained using the source data \( D_S \). We start with a simple calibration procedure and then cluster the test inputs so that we can select one of the base classifiers to use for each cluster of examples.

**Calibration.** We will use the softmax probability as our measure of confidence, so each model needs to first be calibrated so that the probabilities outputted for different inputs match the expected accuracy on those inputs. We can use the Expected Calibration Error (ECE) [13, 46] to calibrate each model, with the following procedure: We calculate the ECE on held-out data from the source distribution using the model’s logits temperature scaled by different values of \( \alpha \). Formally, ECE\((f_i) = \sum_{j=1}^{10} P(j) \cdot |o_j - e_j| \), where \( o_j \) is the true fraction of positive instances in bin \( j \), \( e_j \) is the mean of the post-calibrated probabilities for the instances in bin \( j \), and \( P(j) \) is the fraction of all instances that fall into bin \( j \). Let \( \alpha^* \) be the \( \alpha \) that gives the lowest ECE \( e^*_j \) for classifier \( f_i \). We take the model with the highest \( e^* \) and then choose \( \alpha_i \) for every other model that gives an ECE closest to that \( e^* \).

**Selecting a classifier for an input.** Now consider our test set, for which we do not have any labels. We calculate the test logits scaled by the chosen \( \alpha_i \) for each classifier. For each input \( x \) in the test set and each classifier \( f_i \), we calculate the confidence of that classifier’s prediction as the softmax probability, i.e. \( C(x, f_i) = \arg \max_y p(y|f_i(x)/\alpha_i) \). Our goal is to find the most appropriate classifier for each input. However, using the softmax probability as the confidence for each classifier can be noisy for individual inputs. Prior works have shown that clustering can recover meaningful subpopulations with various correlations [60]. Thus, as long as the embedding space maps inputs from the same population closer together, clustering can alleviate the noise in input-level confidence estimates. Given a test set, we cluster the logits using \( K \)-means into \( k \) clusters \( c_1, c_2, ..., c_k \), where \( k = \lceil \frac{|D_T|}{N} \rceil \). We use \( N = 50 \) and do not tune \( N \) in our experiments but do show the effect of using different values in Section F.4. For each cluster, we then select the classifier with the highest average confidence on the points in the cluster, i.e. \( f^* = \arg \max_{f_i} \frac{1}{N} \sum_{x \in c_j} C(x_k, f_i) \). In some cases, the embedding space may not be learned well enough so that clustering may not be able to recover the subpopulations, so we also consider a variant of our method that does not use clustering, which we call CosMoS (input-dep). We provide a summary of CosMoS (cluster) in Algorithm 1.

### 3 Experiments

We seek to empirically answer the following questions: (1) Can CosMoS achieve higher average accuracy than using individual classifiers or other model aggregation methods on a wide range of subpopulations without any labeled data from the target domain? (2) Can we use CosMoS for other use cases such as hyperparameter tuning? (3) How sensitive is CosMoS to design choices such as cluster size? Below and in Appendix F, we describe our experimental setup along with experiments to answer the above questions.

**Datasets.** We run experiments on the following four datasets: (1) Waterbirds [54], (2) CelebA [43], (3) MultiNLI [65], and (4) MetaShift [40]. We provide descriptions of each dataset in Appendix E. For all settings, the base classifiers are trained on the original source datasets, and we use held-out data from the source distribution in order to calibrate the models. For each dataset, we construct multiple target distributions for evaluation representative of a range of potential test distributions, consisting of different subpopulations from either (a) mixing majority and minority groups or (b) each individual group. More specifically, we evaluate on the following subsets of the original test dataset: subsets where majority samples make up \( m \in \{0, 10, 30, 50, 70, 90, 100\} \) percent of the samples with minority samples constituting the remaining samples.

**Base Models.** We use base classifiers trained using ERM, JTT [41], and LISA [69] using their released codebases. For the image datasets, we use a ResNet-50 backbone pre-trained on ImageNet, and we use a pre-trained BERT model for MultiNLI. We take the hyperparameters described in the
We perform a comprehensive experimental evaluation on the four datasets. Along with ERM and a shortcut predictor, there is a subpopulation for which it is the worst classifier, so it be a shortcut or invariant predictor, there is a subpopulation for which it is the worst classifier, but not effectively cluster inputs in the same subpopulations together, but clustering does so on the language domain. For each dataset, for each individual classifier—whether it be a shortcut or invariant predictor, the strategy adopted by prior works by default, by using shortcut features when appropriate. In particular, these results show that we can achieve better performance than using a single invariant model or the ensembling approaches on average accuracy across all mixtures and individual groups.

Thus, the goal in our problem setting is to achieve high accuracy on every potential subgroup. We find that on the 3 image domains, CosMOS (cluster) achieves significantly higher accuracy and lower regret than any of the individual classifiers or model aggregation through ensembling. On the text domain, CosMOS (input-dep) matches Ensemble as the best-performing method.

### Evaluation Metrics
In practice, we may want to use models that have been trained on some source distribution on a variety of potential test distributions with unknown subpopulation composition. Thus, the goal in our problem setting is to achieve high accuracy on *every* potential subgroup. The latter measures the average difference between our method’s accuracy and the accuracy of the best-performing base classifier across each given individual subgroup.

### Table 1: Main results
On each dataset, we evaluate accuracy on a wide range of representative test sets to reflect the wide range potential subpopulations that a model may need to be used for. We find that on the 3 image domains, CosMOS (cluster) achieves significantly higher accuracy and lower regret than any of the individual classifiers or model aggregation through ensembling. On the text domain, CosMOS (input-dep) matches Ensemble as the best-performing method.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Waterbirds</th>
<th>CelebA</th>
<th>MultiNLI</th>
<th>MetaShift</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg Acc</td>
<td>Avg Regret</td>
<td>Avg Acc</td>
<td>Avg Regret</td>
</tr>
<tr>
<td>ERM</td>
<td>87.22 (0.15)</td>
<td>-5.94 (0.37)</td>
<td>79.38 (0.44)</td>
<td>-16.00 (0.53)</td>
</tr>
<tr>
<td>JTT [41]</td>
<td>90.71 (0.22)</td>
<td>-2.5 (0.08)</td>
<td>84.91 (0.11)</td>
<td>-9.92 (0.87)</td>
</tr>
<tr>
<td>LISA [69]</td>
<td>87.22 (0.14)</td>
<td>-5.94 (0.45)</td>
<td>89.88 (0.63)</td>
<td>-6.1 (2.4)</td>
</tr>
<tr>
<td>Ensemble (logits)</td>
<td>90.36 (0.11)</td>
<td>-2.81 (0.23)</td>
<td>87.15 (0.33)</td>
<td>-7.89 (0.60)</td>
</tr>
<tr>
<td>Ensemble (weights)</td>
<td>90.00 (0.0)</td>
<td>-43.17 (0.20)</td>
<td>50 (0)</td>
<td>-45.22 (0.65)</td>
</tr>
<tr>
<td>CosMOS (input-dep)</td>
<td>90.36 (0.15)</td>
<td>-2.81 (0.34)</td>
<td>88.07 (0.32)</td>
<td>-7.03 (0.66)</td>
</tr>
<tr>
<td>CosMOS (clusters)</td>
<td>91.72 (0.19)</td>
<td>-0.74 (0.22)</td>
<td>90.96 (0.18)</td>
<td>-1.5 (0.95)</td>
</tr>
</tbody>
</table>

Table 1: Main results. On each dataset, we evaluate accuracy on a wide range of representative test sets to reflect the wide range potential subpopulations that a model may need to be used for. We find that on the 3 image domains, CosMOS (cluster) achieves significantly higher accuracy and lower regret than any of the individual classifiers or model aggregation through ensembling. On the text domain, CosMOS (input-dep) matches Ensemble as the best-performing method. We do not have access to any target domain labels. We calibrate each model using $\alpha$ between 0.25 and 15 in 0.25 increments. CosMOS uses cluster size $k = \left\lfloor \frac{|D_i|}{N} \right\rfloor$ for each target dataset $D_i$, but we do not tune $N$ and instead just use $N = 50$ for each dataset.

### 3.1 Maintaining low average regret across each subgroup
We investigate whether CosMOS can consistently choose the best classifier for different subpopulations, and how this approach compares only using one model or other model aggregation methods. We perform a comprehensive experimental evaluation on the four datasets. Along with ERM and a robustness approach (either JTT or LISA), we evaluate four methods: (1) Ensemble (logits), which averages the logits of the three models, (2) Ensemble (weights), which averages the weights of the three models, similar to model soups [66], (3) CosMOS (input-dep), which selects different models based on confidence for different inputs but does not cluster the inputs before model selection, and (4) CosMOS (cluster), which clusters the inputs before model selection. Experiments in Table 1 indicate that on our three image datasets, CosMOS (cluster) significantly outperforms using any individual model or the ensembling approaches on average accuracy across all mixtures and individual groups as well as average regret over the groups, and CosMOS (input-dep) matches the best other method, Ensemble (logits), on the language domain. Clustering on the language domain does not improve performance because it does not effectively cluster inputs in the same subpopulations together, but CosMOS without clustering still performs well because it can select the best model for each input. In particular, these results show that we can achieve better performance than using a single invariant predictor, the strategy adopted by prior works by default, by using shortcut features when appropriate. This is because those shortcut features are valuable and are actually better than invariant predictors on some subpopulations, as shown in Table 2.

In Figure 2, we show the accuracy of each method on test mixtures of majority and minority groups on each dataset, and in Figure 3, we show the accuracy of each method on each individual group of the datasets. We observe that on the image domains, CosMOS consistently chooses the classifier that gives the best accuracy on the individual groups and on the mixtures of groups, and CosMOS without clustering does so on the language domain. For each dataset, for each individual classifier—whether it be a shortcut or invariant predictor, there is a subpopulation for which it is the worst classifier, and CosMOS is able to alleviate this issue, as there is no subpopulation on which it significantly underperforms. We also observe that CosMOS (cluster) is even able to achieve higher accuracy than the best individual classifier on some mixtures of groups—in these cases, it is best to use both classifiers on different inputs in the test set.
Figure 2: **Accuracies on mixture distributions.** We show the accuracy of each method on a range of test sets containing mixtures of majority and minority groups weighted differently. While each individual classifier has a subpopulation that it performs poorly on, CosMoS is able to leverage the strengths of each classifier to mitigate worst performance on a mixture.

Figure 3: **Accuracies on individual groups.** We show that CosMoS is able to selectively assign different classifiers based on which is best for different groups.

4 **Limitations and Conclusion**

In this paper, we study the issue of spurious correlations in datasets and propose a framework called COnfidence-baSed MOdel Selection (CosMoS), to selectively employ appropriate classifiers for different inputs. Our method is based on the observation that different models have unique strengths on different regions of the input space, and using a single predictor may result in suboptimal performance. While prior methods typically focus on the average-case or worst-case subpopulation, CosMoS can achieve high accuracy across a range of subpopulations without sacrificing performance on any specific subpopulation. Our method does not require access to any labeled data from the target domain, any group annotations, or any model weights. We evaluate our method on four spurious correlation datasets each with multiple test distributions and find that CosMoS outperforms existing approaches, achieving 2-5% lower average regret across subgroups of the input space. Furthermore, our approach can be used for other use cases where it may be desirable to choose the best model for an unlabeled test set, including hyperparameter tuning. Despite the strengths of our framework, limitations remain. First, the performance relies on the capabilities of the base classifiers, and will not provide improvements if the base classifiers are all similar to one another. In addition, there are scenarios in which it is not be appropriate to use shortcut features; for example, in some applications, shortcut features may correspond to protected attributes, e.g. race, gender, age, etc. CosMoS should not be applied in such settings. Nevertheless, CosMoS is useful in other settings where we aim to achieve both high average accuracy and worst-group accuracy, and our results demonstrate the benefits of viewing shortcut and invariant classifiers on equal footing and selectively employing appropriate models for different inputs.
References


A Related Work

Robustness and Adaptation Using Unlabeled Target Data. Many prior works aim to improve robustness to various distribution shifts [63, 17, 1, 54, 47, 9, 41, 69, 75, 70]. Additionally, prior works have studied how to adapt pre-trained features to a target distribution via fine-tuning [48, 71, 58]. Such fine-tuning works frame robustness to distribution shift as a zero-shot generalization problem [29, 73, 67, 31], where the model is fine-tuned on source data and evaluated on data from the target distribution. Several recent works have also studied how to adapt to a target distribution using unlabeled target data at test time [33, 17, 64, 74]. In contrast to these single-model methods, this paper presents a simple and novel approach that capitalizes on multiple models to address the distribution shift caused by spurious correlations. The model selection strategy of CosMoS is orthogonal to the specific methods above, and can in principle leverage models trained with any of those techniques.

Diverse Classifiers. Neural networks, by their nature, often exhibit a bias towards learning simple functions that rely on shortcut features [2, 21, 57, 19, 53, 38, 44]. To better handle novel distributions, prior works consider the entire set of functions that are predictive on the training data [16, 56, 68]. Recent diversification methods show how to discover such a set [61, 50, 36, 7] and show that there are multiple predictors that perform well on the source domain but differently on different test domains. CosMoS assumes access to a diverse set of classifiers that were trained with different strategies, with a focus on the selection stage. We choose an appropriate classifier for each input region in an unsupervised way, whereas most existing works for choosing among diverse classifiers do a form of adaptation using labels.

Estimating Confidence. Prior works have studied various metrics for estimating model confidence, such as softmax probability [51], ensemble uncertainty [32], or prediction entropy [52]. These metrics are often studied in the context of uncertainty quantification for out-of-distribution detection [24, 34, 49, 5]. CosMoS first calibrates each classifier via temperature scaling, and then uses the softmax probability as an estimate of the confidence of each model. Future advancements in calibration and confidence estimation methods [22, 39] can be incorporated into the CosMoS framework to further improve performance.

Model Selection and Ensemble Methods. Ensemble methods improve performance by integrating predictions from several models [14, 4, 23, 32, 25, 66]. Recognizing that different models may be best suited for different inputs, this paper investigates optimal model selection in an input-specific fashion. Previous works aggregate the information in different models through feature selection [12, 42, 6, 37] or model selection by estimating the accuracy of each model on unlabeled target data [18, 3]. Our method is an example of dynamic selection (DS) from multiple classifiers [10, 55, 11, 15]; we propose a distinct selection criteria, combining both clustering and confidence strategies to enhance performance on subpopulation shifts. Finally, there is an extensive body of work on mixture of experts (MoE) methods, which train a set of specialized experts and a mechanism to route datapoints to experts [26, 27, 72, 45]. While CosMoS is similar in spirit to MoE methods, it is much simpler and does not require any additional training.

B Problem Setting: Spurious Correlations

We now describe our problem setting, where the goal is to provide an accurate decision boundary under a wide range of target distributions for datasets with spurious correlations. In spurious settings, the source and target distributions are mixtures of a common set of subpopulations, but the relative proportions of these subpopulations may differ. Because of the different subpopulation proportions, certain groups may be over- or under-represented in the (source) training data. Importantly, we do not assume access to target labels, or any form of prior knowledge of the target distributions. We also do not assume access to group labels of any kind; i.e. we do not know the subpopulation that a datapoint belongs to, nor do we even know the number of underlying groups.

More formally, we consider a source distribution \( p_S(x, y) \) and multiple target distributions \( p_T^1(x, y), p_T^2(x, y), \ldots \), each corresponding to a different distribution over subpopulations. The source dataset \( D_S \in (X \times Y)^N \) is sampled according to the source distribution \( p_S \). We evaluate performance on each target distribution \( p_T^i \) on test datapoints \( D_T^i \). Let \( z \) be a hidden index variable that indicates the
subpopulation of a datapoint. We assume that for every target distribution $T_i$, the following invariances hold: $p_S(y) = p_{T_i}(y)$, $p_S(y|x, z) = p_{T_i}(y|x, z)$, and $p_S(x|y, z) = p_{T_i}^1(x|y, z)$. The main difference across distributions is the relative proportions of subpopulations $p(z)$. Our goal is to find the optimal predictor $p_T^1(y|x_i)$ for any desired subpopulation $p_T^1$.

We assume access to diverse classifiers $f_1, f_2, \ldots, f_K$ that are trained on the source data $D_S$, potentially using different objectives or pre-trained backbones, etc. Classifiers trained in different ways are known to have different inductive biases, making different classifiers perform better on different subpopulations. In other words, if we consider two subpopulations denoted $z = 0$ or $z = 1$, a classifier $f_i$ may achieve higher accuracy than another $f_j$ on $p(y|x, z = 0)$, while a different classifier may be best on $p(y|x, z = 1)$. We do not make any specific assumptions on the training procedure of each classifier aside from leveraging the fact that they have complementary strengths.

We note that our setting differs from the setting studied in some prior works on spurious correlations [54], which evaluate the model’s performance only on the hardest target distribution (i.e., worst-group accuracy). Our setting also differs from those that use a small amount of target data to fine-tune [35, 7]. It may be expensive to acquire any target labels, and we desire a method that can be applied to any target distribution without requiring additional information from the domain.

C Full Algorithm

Algorithm 1 Confidence-Based Model Selection

1: Input: \{f_1, ..., f_K\} base classifiers.
2: for each classifier $f_i$ do
3: Calibrate with ECE to obtain temperature $\alpha_i$.
4: for test input $x$ do
5: for each classifier $f_i$ do
6: Compute confidence $C(x, f_i) = \arg \max_y p(y|f_i(x))/\alpha_i$
7: Cluster test embeddings with K-means into $c_1, c_2, ..., c_k$.
8: for test input $x$ in each cluster $c_j$ do
9: Select classifier $f^* = \arg \max_{f_i} \frac{1}{N} \sum_{x_j \in c_j} C(x_j, f_i)$.
10: Predict label $f^*(x)$
11: return predicted labels

D Formal Intuition for Method

CosMOS is based on the intuition that a well-calibrated classifier can provide a signal on its suitability for a given input. The classifier with the highest confidence on an input is likely the best choice for the subpopulation $z$ which contains that input. We acknowledge that calibration is a difficult problem but find that simple temperature scaling suffices for our experiments. If all classifiers are perfectly calibrated for each test input, we can solely rely on the predictive confidence, as indicated by softmax probabilities, to select the best classifier for each input. However, due to mismatches between training and test distributions, classifiers may not be perfectly calibrated on test data, so the confidence estimates may be noisy. If the inputs are mapped to a embedding space where inputs from the same population are closer together, we can mitigate the variance in these estimates by first clustering the test inputs, as one cluster will likely contain inputs from the same subpopulation. Thus, as long as it successfully groups inputs from the same population together, clustering may give a clearer signal of which implicit $z$ an input belongs to, and we should choose the appropriate classifier according to which has the highest average confidence on each cluster of inputs. We find that this approach effectively reduces noise in the confidence estimates through a smoothing effect.

We now provide some more formal intuition on our method. While we make some stronger assumptions in this section, we do not strictly require these assumptions in practice. The purpose of the assumptions here is to provide clear intuition for why our method works by establishing a connection to entropy minimization. Consider the graphical model presented in Section B, where
\( p(x, z, y) \) factorizes as \( p(x|z, y)p(z)p(y) \). Assume each model \( f_i \) in our class of base classifiers \( \{f_1, f_2, \ldots, f_K\} \) corresponds to some subpopulation \( z_i \) on which the model performs the best among the set of models. Let us denote the set \( \{z_1, \ldots, z_k\} \) as \( Z_k \). Now, if we restrict the support of \( p(z|x) \) from our graphical model above to the set \( Z_k \), then if we have knowledge of \( p(z|x) \) on every datapoint, we can then choose to use the model (from our class) that has the highest probability of being correct on that datapoint. Consequently, given some unlabeled data points from the target distribution \( x_1, x_2, \ldots, x_m \), we attempt to solve an inference problem over \( p(z|x) \). But, issues arise because we are only given unlabeled data from the target domain, and without any knowledge of true labels \( y \) on the set of target datapoints \( x_1, \ldots, x_m \), or access to a hold out set with information on \( z \) (group/subpopulation information), the problem remains a bit ill-posed.

One way to make the problem more identifiable, is to first estimate the label \( \hat{y}_i \), on each test point \( x_i \), and then solve the maximum likelihood estimation problem:

\[
\arg \max_{\{p(z|x_i)\}_{i=1}^m} \sum_{i=1}^m \log p(\hat{y}_i \mid x_i) = \arg \max_{\{p(z|x_i)\}_{i=1}^m} \sum_{i=1}^m \log \int z p(\hat{y}_i \mid x_i, z) \cdot p(z \mid x_i) \, dz
\]

Now, under the assumption that \( p(z \mid x) \) is smooth in some metric (, for two data points \( x_1, x_2 \) that are close to each other in some metric space, \( p(z \mid x_1) \approx p(z \mid x_2) \)), we can solve the above inference problem after we cluster \( x_1, x_2, \ldots, x_m \) into disjoint clusters under the metric space over which the smoothness assumption holds. If all data points in the cluster share the same \( p(z \mid x) \), we can write the solution of the above likelihood maximization problem as:

\[
p(z \mid x_i) = \delta_{z(x_i)} \quad \text{where,} \quad z(x_i) = \arg \max_z \sum_{x' \in C(x_i)} p(\hat{y}_i \mid z, x'),
\]

where \( \delta_z \) is a Dirac delta function on some point \( z \) in our set \( Z_k \) and \( C(x_i) \) is the cluster assignment of the point \( x_i \). The better the pseudolabels are, the more the assignment \( z(x_i) \) corresponds to a specific subpopulation in set \( Z_k \).

Additionally, prior works have shown in different problem settings how pseudolabeling can be motivated through entropy minimization [8]. For our setting, if the calibration of each model gives a reliable signal, and for each subpopulation we expect one model in particular to be the best, then the model assignment that achieves the lowest entropy will achieve highest performance. Furthermore, the predictive entropy of single instances can have high variance, and we can mitigate this by the smoothed estimate of average confidence within each cluster. Hence, we can also interpret the inference problem as smoothed entropy minimization, with the following objective:

\[
\min_{z, i} \sum_i H(p(y|x_i)) + \sum_{j,k} \text{dist}(p(z_i|x_j)||p(z_i|x_k))\lambda_{jk},
\]

where \( \lambda_{jk} \rightarrow \infty \) for points in the same cluster and 0 otherwise. Our method can therefore be derived as performing MLE estimation on top of pseudo-labels, which is equivalent to minimizing the entropy of predictions with a smoothing regularizer term. This motivates our use of confidence in choosing which classifier to use, as models that have low entropy correspond to models with high confidence.

### E Dataset Details

Below, we describe our four datasets. For each, we calibrate the classifiers using held-out data from the source domain and evaluate on multiple different target distributions from either (a) mixing majority and minority groups or (b) each individual group. Our aim with constructing many test sets is to evaluate performance on a wide range of potential subpopulation shifts.

- **Waterbirds** [54]. This dataset tasks the model with classifying images of birds as either a waterbird or landbird. The label is spurious correlated with the image background, which is either water or land. There are four individual groups; in our evaluation, we consider the two groups where the bird and background are correlated as majority groups and the others as minority groups.

- **CelebA** [43]. The task is to classify the hair color of the person as blond or not blond, and this label is spuriously correlated with gender. There are four total groups; in our evaluation, we consider the two groups (blond, female) and (not blond, male) as majority groups and the two others as minority groups, and we additionally evaluate on all four groups individually.
We are given 6 JTT [41] models trained on Waterbirds with different (learning rate, weight decay) hyperparameter combinations. We find that for each subpopulation, the classifier chosen for the most points by CosMoS corresponds to the classifier with the highest (or nearly highest) accuracy on that subpopulation. Thus, CosMoS is able to hyperparameter tuning for different target domains without any target labels.

- **MultiNLI** [65]. The task is to classify whether the second sentence is entailed by, neutral with, or contradicts the first sentence in a pair of sentences. The label is spuriously correlated with the presence of negation words in the second sentence. There are 6 individual groups; we take (no neg, entailment), (neg, contradiction), (no neg, neutral) as majority groups and (neg, entailment), (no neg, contradiction), and (neg, neutral) as minority groups.

- **MetaShift** [40]. This dataset is derived using the real-world images and natural heterogeneity of Visual Genome [30]. The model is tasked with identifying an image as a dog(shelf) or cat(shelf), given training data of cat(sofa), cat(bed), dog(cabinet), and dog(bed) domains. There are 2 individual groups, and we take the dog(shelf) images as one group and cat(shelf) as another to construct test domains. This task requires generalizing to novel data subpopulations not observed during training.

### F Additional Experimental Results

#### F.1 Using CosMoS for hyperparameter tuning

Although our main motivation is to show how CosMoS can use multiple models improve performance across subpopulations, it can be used for other tasks where we want to choose the best model from a given set of models. In particular, we show that we can use CosMoS in order to do hyperparameter tuning without any target domain labels. For hyperparameter tuning, the typical practice is to use a target validation set. Such a set is implicitly assumed by prior works that study robustness or adaptation [54, 41, 28], and such prior works have not effectively done hyperparameter tuning for a target domain without additional domain-specific information. In this experiment, shown in Figure 4, we take six JTT checkpoints for Waterbirds as our given models, which are trained using different hyperparameters (learning rate and weight decay), and we use CosMoS to choose the best model for each input. For each desired test distribution, the model that is most commonly chosen by CosMoS corresponds to the model with the highest accuracy on that test distribution, so we can take the model most commonly chosen as the model with the best hyperparameters for that target set. Thus, we can do hyperparameter tuning without any additional information from the desired test distribution, and to our knowledge, CosMoS is the first method that does so.

#### F.2 Models trained on source data often perform differently on different test distributions

We first aim to demonstrate how two different classifiers that are predictive on source data can perform differently on different target distributions. On the Waterbirds dataset, consider a robust feature learned by JTT [41] vs. an ERM classifier. In Table 2, as expected, the robust feature achieves the best worst-group accuracy. However, we find that the shortcut feature outperforms the robust feature on the two majority groups, indicating that this feature would achieve higher performance in a distribution skewed towards majority groups. In other words, there is no one best feature, and different features can be best for different target distributions. These observations justify CosMoS: it
Table 2: Different classifiers can be best for different target distributions. There is an inherent tradeoff between the robust and shortcut classifiers: ERM has higher total average accuracy, but JTT has higher worst-group accuracy.

<table>
<thead>
<tr>
<th></th>
<th>Majority Groups</th>
<th>Minority Groups</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LB+L</td>
<td>WB+W</td>
<td>LB+W</td>
<td>WB+L</td>
</tr>
<tr>
<td>Robust Classifier (JTT) [41]</td>
<td>92.99 (0.06)</td>
<td>95.36 (0.39)</td>
<td>86.77 (1.46)</td>
<td>87.55 (0.08)</td>
</tr>
<tr>
<td>Shortcut Classifier (ERM)</td>
<td>96.74 (0.06)</td>
<td>99.24 (0.21)</td>
<td>76.09 (1.20)</td>
<td>76.82 (0.08)</td>
</tr>
</tbody>
</table>

can be beneficial to apply different features to different inputs, based on uncertainty. Extract a diverse set of features that cover both causal and shortcut features, and adapt to different target distributions by interpolating between these learned features.

F.3 Qualitative Clustering

We cluster the test embeddings obtained using a ResNet50 pre-trained on ImageNet and a pre-trained BERT model for MultiNLI. Below, we show how clustering can allow us to estimate which datapoints have similar $z$, after which we can assign classifiers to the points using confidence with less noise. In Figure 5, on the Waterbirds dataset, although clustering does not recover the groups perfectly, it generally groups together datapoints in the same subgroup, and as pre-trained embeddings improve in the future, such clustering may also improve. We can leverage the strength of these pre-trained embeddings to better choose the appropriate classifier for different inputs.

![Figure 5: Clustering recovers subgroups.](image)

In Figure 5, we plot the PCA projections on 2 dimensions of the test embeddings of the Waterbirds dataset, colored by cluster on the left and by group on the right. The points in a cluster chosen by K-means generally correspond to those in a single subgroup.

F.4 Ablation on cluster size

In this subsection, we ablate on the size of clusters used for CosMoS (cluster) on the image datasets. In our above experiments, we cluster the test inputs into size(test set) / $N$ number of clusters, where $N = 50$. $N$ is the only additional hyperparameter that is used in CosMoS (cluster), but we do not tune $N$ above, and we find the value of $N = 50$ works for each image domain where clustering helps. In Table 3, we find that CosMoS (cluster) is not sensitive to the exact cluster size. However, there are small dips in performance with very small and very large cluster sizes, particularly in average regret. For the former, CosMoS (cluster) will become similar to CosMoS (input-dep) and the average confidence per cluster may be less reliable, and if the cluster size is too large, it may not be able to capture different mixtures of subpopulations that are in a test set and assign different classifiers to those different subpopulations.

<table>
<thead>
<tr>
<th>Cluster Size</th>
<th>Avg Acc</th>
<th>Avg Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90.36</td>
<td>-2.81</td>
</tr>
<tr>
<td>5</td>
<td>91.55</td>
<td>-1.48</td>
</tr>
<tr>
<td>10</td>
<td>91.65</td>
<td>-1.6</td>
</tr>
<tr>
<td>20</td>
<td>91.97</td>
<td>-0.66</td>
</tr>
<tr>
<td>50</td>
<td>91.72</td>
<td>-0.74</td>
</tr>
<tr>
<td>100</td>
<td>91.98</td>
<td>-0.39</td>
</tr>
<tr>
<td>500</td>
<td>91.78</td>
<td>-0.82</td>
</tr>
</tbody>
</table>

Table 3: Ablation on cluster size.