Compositional Risk Minimization

Divyat Mahajan[1,2,⋆] **Mohammad Pezeshki**[1] **Ioannis Mitliagkas**[2,3]

Kartik Ahuja[1,†] **Pascal Vincent**[1,2,⋆,†]

Abstract

In this work, we tackle a challenging and extreme form of subpopulation shift, which is termed *compositional shift*. Under compositional shifts, some combinations of attributes are totally absent from the training distribution but present in the test distribution. We model the data with flexible additive energy distributions, where each energy term represents an attribute, and derive a simple alternative to empirical risk minimization termed *compositional risk minimization (CRM)*. We provide an extensive theoretical analysis of CRM, where we show that our proposal extrapolates to special affine hulls of seen attribute combinations. Empirical evaluations on benchmark datasets confirms the improved robustness of CRM compared to other popular methods designed to tackle various forms of subpopulation shifts.

1 Introduction

The ability to make sense of the rich complexity of the sensory world by decomposing it into sets of elementary factors and recomposing these factors in new ways is a hallmark of human intelligence. This capability is typically grouped under the umbrella term compositionality [Fodor and Pylyshyn, 1988, Montague, 1970]. Compositionality underlies both the semantic understanding and the imaginative prowess of humans, enabling robust generalization and extrapolation. For instance, human language allows us to imagine situations we have never seen before, such as "a blue elephant riding a bicycle on the Moon." While most works on compositionality have focused on its generative aspect, i.e., imagination, as seen in diffusion models [Yang et al., 2023a], compositionality is equally important in discriminative tasks.

In this work, we dive into this less-explored realm of compositionality for discriminative tasks, specifically in the context of multi-attribute data, where each input is associated with multiple labeled categorical attributes. During training, we observe inputs from only a subset of all possible combinations of individual attributes, but the test data contains novel combinations of attributes never seen during training. Following Liu et al. [2023], we refer to this distribution shift as *compositional shift*, which can also be viewed as an extreme case of subpopulation shift [Yang et al., 2023b]. We develop Compositional Risk Minimization (CRM), an adaptation of the Empirical Risk Minimization (ERM) tailored for multi-attribute data under compositional shifts. CRM is built on additive energy distributions, which were previously studied for generative compositionality [Liu et al., 2022a].

Contributions. a) Theory of discriminative compositional shifts: For the family of additive energy distributions, we prove that additive energy classifiers generalize compositionally to novel combinations of attributes represented by a special mathematical object, which we call *discrete affine hull*. Our characterization of extrapolation is sharp, i.e., we show that it is not possible to generalize beyond *discrete affine hull*. **b) A practical method:** CRM is a simple algorithm for

1: Meta FAIR. 2: Mila, Université de Montréal, DIRO. 3 : Archimedes Research Unit, Athena Res. Center
⋆: Work done at Meta. †: Joint last author.

NeurIPS 2024 Workshop on Compositional Learning: Perspectives, Methods, and Paths Forward.

training classifiers, which first trains an additive energy classifier and then adjusts the trained classifier for tackling compositional shifts. We empirically validate the superiority of the CRM algorithm to other prior methods proposed for robustness to various forms of subpopulation shifts.

2 Problem Setting

2.1 Generalizing under Compositional Distribution Shift

Consider an input x (e.g., image) which belongs to a group that is characterized by an attribute vector $z = (z_1, \dots, z_m)$ (e.g., class label, background label). There are m attributes and each attribute z_i can take d possible values, i.e., $z_i \in \{1, \dots, d\}$. Let $p(x,z) = p(z)p(x|z)$ denote the train distribution, and $q(x,z) = q(z)q(x|z)$ the test distribution. We denote the support of each attribute component z_i under training distribution as $\mathcal{Z}_i^{\text{train}}$ and the support of z under training distribution as $\mathcal{Z}^{\text{train}}$. The corresponding supports for the test distribution are denoted as $\mathcal{Z}_i^{\text{test}}$ and $\mathcal{Z}^{\text{test}}$. We define the Cartesian product of marginal support under training as $\mathcal{Z}^{\times} := \mathcal{Z}_1^{\text{train}} \times \mathcal{Z}_2^{\text{train}} \times \cdots \mathcal{Z}_m^{\text{train}}$.

In this work, we study *compositional shifts* that are characterized by:

1. $p(x|z) = q(x|z), \forall z \in \mathcal{Z}^{\times}$.
2. $\mathcal{Z}^{\text{test}} \not\subseteq \mathcal{Z}^{\text{train}}$ but $\mathcal{Z}^{\text{test}} \subseteq \mathcal{Z}^{\times}$.

The first point states that the conditional density of inputs remains invariant, i.e., the data generation mechanism from attributes to the inputs remains invariant. The only change from train to test is the shift in the prior probabilities of attributes from $p(z)$ to $q(z)$, which is specified by the second point as the difference in their support. Specifically, at test we observe novel combinations of individual attributes but not a completely new individual attribute. To illustrate the setup, we use the Waterbirds dataset [Sagawa et al., 2019] as the running example. Each image x has two attributes, $z = (y, a)$, where y tells the class of the bird – Waterbird (WB) or Landbird (LB), and a tells the background – Water (W) or Land (L). Training distribution consists of data from three groups – (WB, W), (LB, L), (LB, W). However, the test distribution also consists of data from the novel group (WB, L) as well.

The task of compositional generalization is then to build classifiers that are robust to such compositional distribution shifts. Our setup differs from the standard subpopulation shifts [Yang et al., 2023b], where we observe data from all the groups but some groups present much more data than the others.

2.2 Additive Energy Distribution

We assume that $p(x|z)$ is of the form of an *additive energy distribution* (AED):

$$p(x|z) = \frac{1}{\mathbb{Z}(z)} \exp\Big(-\sum_{i=1}^{m} E_i(x, z_i)\Big) \tag{1}$$

where $\mathbb{Z}(z) := \int \exp\Big(-\sum_{i=1}^{m} E_i(x, z_i)\Big) dx$ is the partition function. We do not make assumptions[1] on the E_i except $\mathbb{Z}(z) < \infty$, leaving the resulting $p(x|z)$ very flexible. This form is a natural choice to model inputs that must satisfy a *conjunction* of characteristics (such as being a natural image of a landbird *AND* having a water background), corresponding to our attributes. Our choice of AED is inspired by two lines of work. Firstly, these distributions were used to enhance compositionality in generative tasks Liu et al. [2022a] but they have not been used in discriminative compositionality. Secondly, they can interpreted through the independent mechanisms principle [Janzing and Schölkopf, 2010, Parascandolo et al., 2018], where the energy functions E_i are (algorithmically) independent.

We provide an alternate way to express equation 1 using dot-products. For all $z \in \mathcal{Z}$ with $\mathcal{Z} = \{1, \dots, d\}^m$, denote $\sigma(z) \in \mathbb{R}^{m*d}$ as a concatenation of m one-hot vectors, i.e. $\sigma(z) = [\text{onehot}(z_1), \dots, \text{onehot}(z_m)]^{\top}$. We also define a vector valued map $E(x) = [E_1(x,1), \dots, E_1(x,d), \dots, E_m(x,1), \dots, E_m(x,d)]^{\top}$ where $E_i(x, z_i)$ is the energy term for i^{th} attribute taking the value z_i. Hence, equation 1 can be written as follows:

$$p(x|z) = \frac{1}{\mathbb{Z}(z)} \exp\Big(-\langle \sigma(z), E(x)\rangle\Big), \tag{2}$$

[1]The support of $p(x|z)$ is assume to be $\mathbb{R}^n$, $\forall z \in \mathcal{Z}^{\times}$.

3 Provable Compositional Generalization

Our goal is to learn a distribution $\hat{q}(z|x)$ that matches the test distribution $q(z|x)$ and predict the attributes at test time in a Bayes optimal manner. Towards this, we propose CRM where we first train an additive energy classifier to predict all the attributes jointly, and then we adjust this classifier for compositional shifts. We first introduce the notion of *Discrete Affine Hull* of a set of attributes that will be used to characterize what new combinations of attributes we can extrapolate to. The *discrete affine hull* of a set of attribute vectors $\mathcal{A} = \{z^{(1)}, \ldots, z^{(k)}\}$ where $z^{(i)} \in \mathcal{Z}$, is defined as:

$$\mathsf{DAff}(\mathcal{A}) = \Big\{ z \in \mathcal{Z} \mid \exists\, \alpha \in \mathbb{R}^k, \sigma(z) = \sum_{i=1}^{k} \alpha_i \sigma(z^{(i)}), \sum_{i=1}^{k} \alpha_i = 1 \Big\}$$

We now give a simple example to illustrate discrete affine hull. Let us revisit the Waterbirds dataset. Suppose we observe data from three out of the four groups. In one-hot encoding, we represent WB as $[1, 0]$ and LB as $[0, 1]$. We represent Water as $[1, 0]$ and Land as $[0, 1]$. Below we show that the attribute vector WB on L represented as $[1\ 0\ 0\ 1]$ can be expressed as an affine combination of the remaining three attribute vectors. Based on this, we can conclude that the discrete affine hull of three one-hot concatenated vectors contains all the four possible one-hot concatenations.

$$(+1) \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \;+\; (-1) \cdot \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \;+\; (+1) \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{3}$$

In Section C.4, we generalize the above finding and develop a mathematical characterization of discrete affine hulls that leads to an easy recipe to visualize these sets. In the remainder whenever we use affine hull it means discrete affine hull.

We now describe the first step in the proposed approach CRM, where we aim to train a perfect estimator of training distribution $p(z|x)$. Observe that if we apply Bayes rule to AED $p(x|z)$, we have $p(z|x) = \frac{\exp\Big(-\langle \sigma(z), E(x)\rangle + \log p(z) - \log \mathbb{Z}(z)\Big)}{\sum_{z' \in \mathcal{Z}^{\mathsf{train}}} \exp\Big(-\langle \sigma(z'), E(x)\rangle + \log p(z') - \log \mathbb{Z}(z')\Big)}$. To guarantee that we can model $p(z|x)$, we define our additive energy classifier with the same *form*, as follows:

$$\tilde{p}(z|x) = \frac{\exp\Big(- \langle \sigma(z), \tilde{E}(x)\rangle + \log \hat{p}(z) - \tilde{B}(z)\Big)}{\sum_{z' \in \mathcal{Z}^{\mathsf{train}}} \exp\Big(- \langle \sigma(z'), \tilde{E}(x)\rangle + \log \hat{p}(z') - \tilde{B}(z')\Big)}, \tag{4}$$

where $\hat{p}(z)$ is the empirical estimate of prior $p(z)$, $\tilde{E} : \mathbb{R}^n \to \mathbb{R}^{md}$ is a function to be learned, $\tilde{B}$ is a lookup table containing a learnable offset for each combination of attributes. Given a data point (x, z), cross-entropy loss $\ell(z, \tilde{p}(\cdot|x)) = -\log \tilde{p}(z|x)$ measures the prediction performance of $\tilde{p}(\cdot|x)$. Hence, we learn the optimal parameters as follows:

$$\hat{E}, \hat{B} \in \arg\min_{\tilde{E}, \tilde{B}} R(\tilde{p}). \tag{5}$$

If the minimization is over arbitrary functions, then we have a perfect estimator $\hat{p}(\cdot|x) = p(\cdot|x), \forall x \in \mathbb{R}^n$. Now, in the second step of CRM, we compute our final predictor $\hat{q}(z|x)$ based on the learned $\hat{p}(z|x)$ designed specifically to extrapolate to novel combinations at test time. Let $\hat{q}(z)$ be an estimate of the marginal distribution over the attributes $q(z)$ with support $\hat{\mathcal{Z}}^{\mathsf{test}}$. For each $z \in \mathcal{Z}^{\mathsf{test}}$, define

$$\hat{q}(z|x) = \frac{\exp\Big(- \langle \sigma(z), \hat{E}(x)\rangle + \log \hat{q}(z) - \log B^{\star}(z)\Big)}{\sum_{z' \in \hat{\mathcal{Z}}^{\mathsf{test}}} \exp\Big(- \langle \sigma(z'), \hat{E}(x)\rangle + \log \hat{q}(z') - \log B^{\star}(z')\Big)}, \tag{6}$$

where $\hat{E}$, $\hat{B}$ correspond to the learned estimator $\hat{p}(z|x)$ (4) and $B^{\star}(z)$ is defined as follows:

$$B^{\star}(z) = \mathbb{E}_{x \sim p(x)} \left[\frac{\exp\Big(- \langle \sigma(z), \hat{E}(x)\rangle \Big)}{\sum_{z' \in \mathcal{Z}^{\mathsf{train}}} \exp\Big(- \langle \sigma(z'), \hat{E}(x)\rangle + \log p(z') - \hat{B}(z')\Big)} \right] \tag{7}$$

We now state our main result on CRM's extrapolation (proof in Appendix C.2). For more intuition regarding the proof, please first check Appendix B.1, where we discuss extrapolation of $p(x|z)$.

Theorem 1. *Consider the setting where $p(.|z)$ follows AED $\forall z \in \mathcal{Z}^{\times}$, the test distribution q satisfies compositional shift characterization and $\mathcal{Z}^{\text{test}} \subseteq \mathsf{DAff}(\mathcal{Z}^{\text{train}})$. If $\hat{p}(z|x) = p(z|x), \forall z \in \mathcal{Z}^{\text{train}}, \forall x \in \mathbb{R}^n$ and $\hat{q}(z) = q(z), \forall z \in \mathcal{Z}^{\text{test}}$, then the output of CRM (equation 6) matches the test distribution, i.e., $\hat{q}(z|x) = q(z|x), \forall z \in \mathcal{Z}^{\text{test}}, \forall x \in \mathbb{R}^n$.*

Each of the above steps are easy to operationalize, as explained in Appendix B.3. Observe that $\hat{p}(\cdot|x) = p(\cdot|x)$ is a condition that even a model trained via ERM can satisfy (with sufficient capacity and data) but it cannot match the true $q(\cdot|x)$. In contrast, CRM optimally adjusts the additive-energy classifier (4) for compositional shifts.

So far we have made a crucial assumption that the attribute combinations in the test distribution are in the affine hull. Is this also a *necessary* condition? We show this is not possible in Appendix C.5, hence we cannot generalize to attributes outside the affine hull. Further, our results have restricted the support of test distribution as $\mathcal{Z}^{\text{test}} \subseteq \mathsf{DAff}(\mathcal{Z}^{\text{train}}) \subseteq \mathcal{Z}^{\times}$, while the compositional shifts only implied $\mathcal{Z}^{\text{test}} \subseteq \mathcal{Z}^{\times}$. This leads to a natural question, how fast does the affine hull grow to capture the cartesian product set? In Appendix B.2, we show for $m = 2$ and d values per attribute, if the number of randomly sampled $z \in \mathcal{Z}^{\text{train}}$ exceeds $8cdlog(d)$, then $\mathsf{DAff}(\mathcal{Z}^{\text{train}}) = \mathcal{Z}^{\times}$ with a high probability (Theorem 3).

4 Experiments

4.1 Setup

We evaluate CRM on widely recognized benchmarks for subpopulation shifts [Yang et al., 2023b], that have 2 attributes $z = (y, a)$, where y denotes the class label and a denotes the spurious attribute (y and a are correlated). However, the standard split between train and test data mandated in these benchmarks does not actually evaluate robustness to compositional shifts, because both train and test datasets contain all the groups ($\mathcal{Z}^{\text{train}} = \mathcal{Z}^{\text{test}} = \mathcal{Z}^{\times}$). Therefore, we repurpose these benchmarks for compositional shifts by discarding samples from one of the groups (z) in the train (and validation) dataset; but we don't change the test dataset, i.e., $z \notin \mathcal{Z}^{\text{train}}$ but $z \in \mathcal{Z}^{\text{test}}$. Let us denote the data splits from the standard benchmarks as $(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}, \mathcal{D}_{\text{test}})$. Then we generate multiple variants of compositional shifts $\{(\mathcal{D}^{\neg z}_{\text{train}}, \mathcal{D}^{\neg z}_{\text{val}}, \mathcal{D}_{\text{test}}) \mid z \in \mathcal{Z}^{\times}\}$, where $\mathcal{D}^{\neg z}_{\text{train}}$ and $\mathcal{D}^{\neg z}_{\text{val}}$ are generated by discarding samples from $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$ that belong to the group z.

Following this procedure, we adapted Waterbirds [Wah et al., 2011], CelebA [Liu et al., 2015], MetaShift [Liang and Zou, 2022], MultiNLI [Williams et al., 2017], and CivilComments Borkan et al. [2019] for experiments. We also experiment with the NICO++ dataset [Zhang et al., 2023], where we already have $\mathcal{Z}^{\text{train}} \subsetneq \mathcal{Z}^{\text{test}} = \mathcal{Z}^{\times}$ as some groups were not present in the train dataset. However, these groups are still present in the validation dataset ($\mathcal{Z}^{\text{val}} = \mathcal{Z}^{\times}$). Hence, the only transformation we apply to NICO++ is to drop samples from the validation dataset such that $\mathcal{Z}^{\text{train}} = \mathcal{Z}^{\text{val}}$. For baselines, we train classifiers via Empirical Risk Minimization (ERM), GroupDRO [Sagawa et al., 2019], Logit Correction (LC) [Liu et al., 2022b], and supervised logit adjustment (sLA) [Tsirigotis et al., 2024]. In all cases we employ a pretrained architecture as the representation network ϕ, followed by a linear layer W to get class predictions, and fine-tune them jointly (see Appendix D.3 for details). For evaluation metrics, we report the average accuracy, group balanced accuracy, and worst-group accuracy on the test dataset. Due to imbalances in group distribution, a method can obtain good average accuracy despite having bad worst-group accuracy. Therefore, the worst-group accuracy is a more indicative metric of robustness to spurious correlations (more details in Appendix D.2).

4.2 Results

Table 1 shows the results of our experiment. For each dataset, we report the *average* accuracy over its various compositional shift scenarios $\{(\mathcal{D}^{\neg z}_{\text{train}}, \mathcal{D}^{\neg z}_{\text{val}}, \mathcal{D}_{\text{test}}) \mid z \in \mathcal{Z}^{\times}\}$ (detailed results for all scenarios are in Appendix E.1). In all cases, CRM either outperforms or is competitive with the baselines in terms of worst group accuracy (WGA). Further, for Waterbirds and MultiNLI, while the logit adjustment baselines appear competitive with CRM on average, if we look more closely at the worst case compositional shift scenario, we find that logit adjustment baselines fare much worse than CRM. For Waterbirds, LC obtains 69.0% worst group accuracy while CRM obtains 73.0% worst

Dataset	Method	Average Acc	WGA	WGA (No Groups Dropped)
Waterbirds	ERM	77.9 (0.1)	43.0 (0.1)	62.3 (1.2)
	G-DRO	77.9 (0.6)	42.3 (2.5)	87.3 (0.3
	LC	88.3 (0.7)	75.5 (0.8)	88.7 (0.3)
	sLA	89.3 (0.4)	77.3 (0.5)	89.7 (0.3)
	CRM	87.1 (0.7)	78.7 (1.6)	86.0 (0.6)
CelebA	ERM	85.8 (0.3)	39.0 (0.6)	52.0 (1.0)
	G-DRO	89.2 (0.5)	67.7 (1.3)	91.0 (0.6)
	LC	91.1 (0.2)	57.4 (0.6)	90.0 (0.6)
	sLA	90.9 (0.1)	57.4 (0.3)	86.7 (1.9)
	CRM	91.1 (0.2)	81.8 (1.2)	89.0 (0.6)
MetaShift	ERM	85.7 (0.4)	60.5 (0.6)	63.0 (0.0)
	G-DRO	86.0 (0.4)	63.8 (0.6)	80.7 (1.3)
	LC	88.5 (0.0)	68.2 (0.5)	80.0 (1.2)
	sLA	88.4 (0.1)	63.0 (0.5)	80.0 (1.2)
	CRM	87.6 (0.2)	73.4 (0.7)	74.7 (1.5)
MultiNLI	ERM	69.1 (0.7)	7.2 (0.6)	68.0 (1.7)
	G-DRO	70.4 (0.1)	34.3 (0.5)	57.0 (2.3)
	LC	75.9 (0.1)	54.3 (0.5)	74.3 (1.2)
	sLA	76.4 (0.5)	55.0 (1.8)	71.7 (0.3)
	CRM	74.6 (0.5)	57.7 (3.0)	74.7 (1.3)
CivilComments	ERM	80.4 (0.1)	55.8 (0.4)	61.0 (2.5)
	G-DRO	80.1 (0.2)	61.6 (0.4)	64.7 (1.5)
	LC	80.7 (0.1)	65.7 (0.5)	67.3 (0.3)
	sLA	80.6 (0.1)	65.6 (0.1)	66.3 (0.9)
	CRM	83.7 (0.1)	68.1 (0.5)	70.0 (0.6)
NICO++	ERM	85.0 (0.0)	35.3 (2.3)	35.3 (2.3)
	G-DRO	84.0 (0.0)	36.7 (0.7)	33.7 (1.2)
	LC	85.0 (0.0)	35.3 (2.3)	35.3 (2.3)
	sLA	85.0 (0.0)	33.0 (0.0)	35.3 (2.3)
	CRM	84.7 (0.3)	40.3 (4.3)	39.0 (3.2)

Table 1: **Robustness under compositional shift.** We report test Average Accuracy and Worst Group Accuracy (WGA), averaged as a group is dropped from training and validation sets. Last column is WGA under the dataset's standard subpopulation shift benchmark, i.e. with no group dropped. All methods have a harder time to generalize when groups are absent from training, but CRM appears consistently more robust (standard error based on 3 random seeds).

group accuracy for the worst case scenario of dropping the group $(0, 1)$ (Table 5). Similarly, for the MultiNLI benchmark, sLA obtains 19.7% worst group accuracy while CRM obtains 31.0% worst group accuracy for the worst case scenario of dropping the group $(0, 0)$ (Table 8).

We also report the worst group accuracy (other metrics in Table 11) for the original benchmark $(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}, \mathcal{D}_{\text{train}})$ which was not transformed for compositional shifts, denoted WGA (No Groups Dropped).This can be interpreted as the "oracle" performance for each benchmark, and we can compare methods based on the performance drop in WGA due to discarding groups in compositional shifts. ERM and GroupDRO appear the most sensitive to compositional shifts, and the logit adjustment baselines also show a sharp drop for CelebA; while CRM is more robust to compositional shifts.

Importance of extrapolating the bias. We conduct an ablation study for CRM where we test a variant that uses the learned bias $\hat{B}$ (e.q. 5) instead of the extrapolated bias $B^\star$ (e.q. 7). Results are presented in Table 10. They show a significant drop in worst-group accuracy if we use the learned bias instead of the extrapolated one. Hence, our theoretically grounded bias extrapolation step is crucial to generalize under compositional shifts.

Acknowledgements

We thank Yash Sharma for his contribution to early exploration of compositional generalization from a generative perspective, and Vinayak Tantia for having many years ago helped shed light on the challenges posed by compositionality in discriminative training. This research was entirely funded by Meta, in the context of Meta's AI Mentorship program with Mila. Ioannis Mitliagkas in his role as Divyat Mahajan's academic advisor, acknowledges support by an NSERC Discovery grant (RGPIN-2019-06512), and a Canada CIFAR AI chair. Pascal Vincent is a CIFAR Associate Fellow in the Learning in Machines & Brains program. Divyat Mahajan acknowledges support via FRQNT doctoral training scholarship for his graduate studies.

References

Kartik Ahuja, Jun Wang, Amit Dhurandhar, Karthikeyan Shanmugam, and Kush R Varshney. Empirical or invariant risk minimization? a sample complexity perspective. *arXiv preprint arXiv:2010.16412*, 2020.

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

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pages 491–500, 2019.

Jack Brady, Roland S Zimmermann, Yash Sharma, Bernhard Schölkopf, Julius Von Kügelgen, and Wieland Brendel. Provably learning object-centric representations. In *International Conference on Machine Learning*, pages 3038–3062. PMLR, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Kefan Dong and Tengyu Ma. First steps toward understanding the extrapolation of nonlinear models to unseen domains. *arXiv preprint arXiv:2211.11719*, 2022.

Jerry A Fodor and Zenon W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, 1988.

Jonathan Gordon, David Lopez-Paz, Marco Baroni, and Diane Bouchacourt. Permutation equivariant models for compositional generalization in language. In *International Conference on Learning Representations*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Geoffrey E Hinton. Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence*, 46(1-2):47–75, 1990.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020.

Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, pages 336–351. PMLR, 2022.

Dominik Janzing and Bernhard Schölkopf. Causal inference using the algorithmic markov condition. *IEEE Transactions on Information Theory*, 56(10):5168–5194, 2010.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

Najoung Kim and Tal Linzen. Cogs: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)*, pages 9087–9105, 2020.

Sébastien Lachapelle, Divyat Mahajan, Ioannis Mitliagkas, and Simon Lacoste-Julien. Additive decoders for latent variables identification and cartesian-product extrapolation. *Advances in Neural Information Processing Systems*, 36, 2024.

Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR, 2018.

Brenden M Lake and Marco Baroni. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623(7985):115–121, 2023.

Weixin Liang and James Zou. Metashift: A dataset of datasets for evaluating contextual distribution shifts and training conflicts. *arXiv preprint arXiv:2202.06523*, 2022.

Baihan Lin, Djallel Bouneffouf, and Irina Rish. A survey on compositional generalization in applications. *arXiv preprint arXiv:2302.01067*, 2023.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pages 423–439. Springer, 2022a.

Sheng Liu, Xu Zhang, Nitesh Sekhar, Yue Wu, Prateek Singhal, and Carlos Fernandez-Granda. Avoiding spurious correlations via logit correction. *arXiv preprint arXiv:2212.01433*, 2022b.

Yuejiang Liu, Alexandre Alahi, Chris Russell, Max Horn, Dominik Zietlow, Bernhard Schölkopf, and Francesco Locatello. Causal triplet: An open challenge for intervention-centric causal representation learning. In *Conference on Causal Learning and Reasoning*, pages 553–573. PMLR, 2023.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

Richard Montague. Pragmatics and intensional logic. *Synthese*, 22(1):68–94, 1970.

Mitja Nikolaus, Mostafa Abdou, Matthew Lamm, Rahul Aralikatte, and Desmond Elliott. Compositional generalization in image captioning. *arXiv preprint arXiv:1909.04402*, 2019.

Giambattista Parascandolo, Niki Kilbertus, Mateo Rojas-Carulla, and Bernhard Schölkopf. Learning independent causal mechanisms. In *International Conference on Machine Learning*, pages 4036–4044. PMLR, 2018.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Mohammad Pezeshki, Diane Bouchacourt, Mark Ibrahim, Nicolas Ballas, Pascal Vincent, and David Lopez-Paz. Discovering environments with xrm. *arXiv preprint arXiv:2309.16748*, 2023.

Tony Plate et al. Holographic reduced representations: Convolution algebra for compositional distributed representations. In *IJCAI*, pages 30–35, 1991.

Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*, 2020.

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.

Simon Schug, Seijin Kobayashi, Yassir Akram, Maciej Wołczyk, Alexandra Proca, Johannes Von Oswald, Razvan Pascanu, João Sacramento, and Angelika Steger. Discovering modular solutions that generalize compositionally. *arXiv preprint arXiv:2312.15001*, 2023.

Sania Sinha, Tanawan Premsri, and Parisa Kordjamshidi. A survey on compositional learning of ai models: Theoretical and experimetnal practices. *arXiv preprint arXiv:2406.08787*, 2024.

Christos Tsirigotis, Joao Monteiro, Pau Rodriguez, David Vazquez, and Aaron C Courville. Group robust classification without any group information. *Advances in Neural Information Processing Systems*, 36, 2024.

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

Zihao Wang, Lin Gui, Jeffrey Negrea, and Victor Veitch. Concept algebra for (score-based) text-controlled generative models. *Advances in Neural Information Processing Systems*, 36, 2024.

Thaddäus Wiedemer, Jack Brady, Alexander Panfilov, Attila Juhos, Matthias Bethge, and Wieland Brendel. Provable compositional generalization for object-centric learning. *arXiv preprint arXiv:2310.05327*, 2023.

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional generalization from first principles. *Advances in Neural Information Processing Systems*, 36, 2024.

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023a.

Yuzhe Yang, Haoran Zhang, Dina Katabi, and Marzyeh Ghassemi. Change is hard: A closer look at subpopulation shift. In *International Conference on Machine Learning*, 2023b.

Xingxuan Zhang, Yue He, Renzhe Xu, Han Yu, Zheyan Shen, and Peng Cui. Nico++: Towards better benchmarking for domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16036–16047, 2023.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4396–4415, 2022.

Appendix

Contents

A Related Works

Compositional Generalization Compositionality has long been seen as an important capability on the path to building [Fodor and Pylyshyn, 1988, Hinton, 1990, Plate et al., 1991, Montague, 1970] human-level intelligence. The history of compositionality is very long to cover in detail here, refer to these surveys [Lin et al., 2023, Sinha et al., 2024] for more detail. Compositionality is associated with many different aspects, namely systematicity, productivity, substitutivity, localism, and overgeneralization [Hupkes et al., 2020]. In this work, we are primarily concerned with systematicity, which evaluates a model's capability to understand known parts or rules and combine them in new contexts. Over the years, several popular benchmarks have been proposed to evaluate this systematicity aspect of compositionality, Lake and Baroni [2018] proposed the SCAN dataset, Kim and Linzen [2020] proposed the COGS dataset. These works led to development of several insightful approaches to tackle the challenge of compositionality [Lake and Baroni, 2023, Gordon et al., 2019]. Most of these works on systematicity have largely focused on generative tasks, [Liu et al., 2022a, Lake and Baroni, 2023, Gordon et al., 2019, Wang et al., 2024], i.e., where the model needs to recombine individual distinct factors/concepts and generate the final output in the form of image or text. There has been lesser work on discriminative tasks [Nikolaus et al., 2019], i.e., where the model is given an input composed of a novel combination of factors and it has to predict the underlying novel combination. In this work, our focus is to build an approach that can provably solve these discriminative tasks.

On the theoretical side, recently, there has been a growing interest to build provable approaches for compositional generalization [Wiedemer et al., 2023, 2024, Brady et al., 2023, Dong and Ma, 2022, Lachapelle et al., 2024]. These works study models where the labeling function or the decoder is additive over individual features, and prove generalization guarantees over the Cartesian product of the support of individual features. The ability of a model to generalize to Cartesian products of the individual features is an important form of compositionality, which checks the model's capability to correctly predict in novel circumstances described as combination of contexts seen before. Dong and Ma [2022] developed results for additive models, i.e., labeling function is additive over individual features. While in Wiedemer et al. [2023], the authors considered a more general model class in comparison to Dong and Ma [2022]. The labeling function/decoder in [Wiedemer et al., 2023] takes the form $f(x_1, \cdots, x_n) = C(\psi_1(x_1), \cdots, \psi_n(x_n))$. However, they require a strong assumption, where the learner needs to know the function C that is used to generate the data. Lachapelle et al. [2024], Brady et al. [2023] extended the results from Dong and Ma [2022] to the unsupervised setting. Lachapelle et al. [2024], Brady et al. [2023] are inspired by the success of object-centric models and show additive decoders enable generative models (autoencoders) to achieve Cartesian product extrapolation. While these works take promising and insightful first steps for provable compositional guarantees, the assumption of additive deterministic decoders (labeling functions) may come as quite restrictive. In particular a given attribute combination can then only correspond to a *unique* observation, produced by a very limited interaction between generative factors, not to a rich distribution of observations. By contrast an additive energy model can associate an almost arbitrary distribution over observations to a given set of attributes. Hence, we take inspiration independent mechanisms principle [Janzing and Schölkopf, 2010, Parascandolo et al., 2018] for our setting based on additive energy models. In the spirit of this principle, we think of each factor impacting the final distribution through an independent function, where independence is in the algorithmic sense and not the statistical sense. Based on this more realistic assumption of additive energy, our goal is to develop an approach that provably enables *zero-shot compositional generalization in discriminative tasks*, where the model needs to robustly predict never seen before factor combinations that the input is composed of. These additive energy distributions have also been used in generative compositionality [Liu et al., 2022a] but not in discriminative compositionality.

Finally, in another line of work [Schug et al., 2023], the authors consider compositionality in the task space and develop an approach that achieves provable compositional guarantees over this task space and empirically outperforms meta-learning approaches such as MAML and ANIL. Specifically, they operate in a student-teacher framework, where each task has a latent code that specifies the weights for different modules that are active for that task.

Domain Generalization Generalization under subpopulation shifts, where certain groups or combinations of attributes are underrepresented in the training data, is a well-known challenge in machine learning. Group Distributionally Robust Optimization (GroupDRO) [Sagawa et al., 2019] is a prominent method that minimizes the worst-case group loss to improve robustness across groups. Invariant

Risk Minimization (IRM) Arjovsky et al. [2019] encourages the model to learn invariant representations that perform well across multiple environments. Perhaps the simplest methods are SUBG and RWG Idrissi et al. [2022], which focus on constructing a balanced subset or reweighting examples to minimize or eliminate spurious correlations. There are many other interesting approaches that were proposed, see the survey for details Zhou et al. [2022]. The theoretical guarantees developed for these approaches [Rosenfeld et al., 2020, Arjovsky et al., 2019, Ahuja et al., 2020] require a large diversity in terms of the environments seen at the training time. In our setting, we incorporate inductive biases based on additive energy distributions that help us arrive at provable generalization with limited diversity in the environments.

Closely related to our proposed method are the logit adjustment methods Kang et al. [2019], Menon et al. [2020], Ren et al. [2020] used in robust classification. Kang et al. [2019] introduced Label-Distribution-Aware Margin (LDAM) loss for long-tail learning, proposing a method that adjusts the logits of a classifier based on the class frequencies in the training set to counteract bias towards majority classes. Similarly, Menon et al. [2020] and Ren et al. [2020] (Balanced Softmax), modify the standard softmax cross-entropy loss to account for class imbalance by shifting the logits according to the prior distribution over the classes. Closest to our work are the Logit Correction (LC) [Liu et al., 2022b] and Supervised Logit Adjustment (sLA) [Tsirigotis et al., 2024] methods that use logit adjustment for group robustness. LC adjusts logits based on the joint distribution of environment and class label, reducing reliance on spurious features in imbalanced training sets. When environment annotations are unknown, a second network infers them. Supervised Logit Adjustment (sLA) adjusts logits according to the conditional distribution of classes given the environment. In the absence of environment annotations, Unsupervised Logit Adjustment (uLA) uses self-supervised learning (SSL) to pre-train a model for general feature representations, then derives a biased network from this pre-trained model to infer the missing environment annotations.

B Compositional Risk Minimization: Additional Details

B.1 Extrapolation of Conditional Density

We learn a set of distributions $\hat{p}(x|z) = \frac{1}{\hat{\mathbb{Z}}(z)} \exp\Big(- \langle \sigma(z), \hat{E}(x) \rangle \Big), \forall z \in \mathcal{Z}^{\mathsf{train}}$ by maximizing the likelihood over training distribution, where $\hat{E}$ denotes the estimated energy and $\hat{\mathbb{Z}}$ denotes the estimated partition function. Under perfect maximum likelihood maximization $\hat{p}(x|z) = p(x|z)$ for all the training groups $z \in \mathcal{Z}^{\mathsf{train}}$. We can define $\hat{p}(x|z)$ for all $z \in \mathcal{Z}^{\times}$ beyond $\mathcal{Z}^{\mathsf{train}}$ in a natural way as follows. For each $z \in \mathcal{Z}^{\times}$, we have estimated every individual component z_i we have estimated $\hat{E}_i(x, z_i)$. We set $\hat{\mathbb{Z}}(z) = \int \exp\Big(- \langle \sigma(z), \hat{E}(x) \rangle \Big) dx$ and the density for each $z \in \mathcal{Z}^{\times}$, $\hat{p}(x|z) = \frac{1}{\hat{\mathbb{Z}}(z)} \exp\Big(- \langle \sigma(z), \hat{E}(x) \rangle \Big)$.

Theorem 2. *If the true and learned distribution ($p(\cdot|z)$ and $\hat{p}(\cdot|z)$) are additive energy distributions, then $\hat{p}(\cdot|z) = p(\cdot|z), \forall z \in \mathcal{Z}^{\mathsf{train}} \implies \hat{p}(\cdot|z') = p(\cdot|z'), \forall z' \in \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$.*

The result above argues that so long as the group z' is in the discrete affine hull of $\mathcal{Z}^{\mathsf{train}}$, the estimated density extrapolates to it. We provide a proof sketch ahead, with the full proof in Appendix C.1

Proof sketch: Under perfect maximum likelihood maximization $\hat{p}(x|z) = p(x|z), \forall z \in \mathcal{Z}^{\mathsf{train}}$. Replacing these densities by their expressions and taking their $\log$ we obtain

$$\langle \sigma(z), \hat{E}(x) \rangle = \langle \sigma(z), E(x) \rangle + C(z), \forall z \in \mathcal{Z}^{\mathsf{train}} \tag{8}$$

where $C(z) = \log\big(\mathbb{Z}(z)/\hat{\mathbb{Z}}(z)\big)$.

For any $z' \in \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$, by definition there exists α such that $\sigma(z') = \sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z \sigma(z)$. Thus $\langle \sigma(z'), \hat{E}(x) \rangle = \sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z \langle \sigma(z), \hat{E}(x) \rangle$, by linearity of the dot product. Substituting the expression for $\langle \sigma(z), \hat{E}(x) \rangle$ from equation 8, this becomes

$$\langle \sigma(z'), \hat{E}(x) \rangle = \sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z \Big(\langle \sigma(z), E(x) \rangle + C(z) \Big) = \langle \sigma(z'), E(x) \rangle + \sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z C(z), \tag{9}$$

From equation 9, we can conclude that $\langle \sigma(z'), \hat{E}(x) \rangle$ estimates $\langle \sigma(z'), E(x) \rangle$ perfectly up to a constant error that does not depend on x. This difference of constant is absorbed by the partition function and hence the conditional densities match: $\hat{p}(x|z') = p(x|z')$.

Using extrapolation of conditional density for compositional generalization of classification. If, on data from training distribution p, we were able to train a good conditional density estimate $\hat{p}(x|z), \forall z \in \mathcal{Z}^{\mathsf{train}}$, then Theorem 2 implies that $\hat{p}(x|z')$ will also be a good estimate of $p(x|z')$ for *new unseen* attributes $z' \in \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$. Provided $\mathcal{Z}^{\mathsf{test}} \subseteq \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$, it is then straightforward to obtain a classifier that generalizes to compositionally-shifted test distribution q. Indeed, we have

$$q(z'|x) = \frac{q(x|z')q(z')}{\sum_{z'' \in \mathcal{Z}^{\mathsf{test}}} q(x|z'')q(z'')} = \frac{p(x|z')q(z')}{\sum_{z'' \in \mathcal{Z}^{\mathsf{test}}} p(x|z'')q(z'')} \approx \frac{\hat{p}(x|z')q(z')}{\sum_{z'' \in \mathcal{Z}^{\mathsf{test}}} \hat{p}(x|z'')q(z'')}$$

where we used the property of compositional shifts $q(x|z) = p(x|z)$. If we know test group prior $q(z')$ (or e.g. assume it to be uniform), we can directly use this expression to compute test group probabilities, even those for attribute combinations never seen at training.

In principle, we can obtain a classifier that generalizes under compositional shift, by first training conditional probability density models $\hat{p}(x|z)$. But high dimensional probability density modeling remains very challenging, and involves dealing with intractable partition functions. It is typically deemed much simpler to learn a discriminative classifier. Therefore, we chose to focus on CRM, which is a completely discriminative approach.

B.2 When does Discrete Affine Hull equal the Cartesian Product Extension?

Under the assumption of compositional shifts, we know that the support of $q(z)$, $\mathcal{Z}^{\mathsf{test}}$ is only restricted to be a subset of the Cartesian product set $\mathcal{Z}^{\times}$, but our theoretical result (Theorem 1) required us to restrict the support further $\mathcal{Z}^{\mathsf{test}} \subseteq \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}}) \subseteq \mathcal{Z}^{\times}$. This leads us to a natural question. If the training attributes that form $\mathcal{Z}^{\mathsf{train}}$ are drawn at random, then how many attribute combinations do we need so that $\mathsf{DAff}(\mathcal{Z}^{\mathsf{train}}) = \mathcal{Z}^{\times}$, at which point CRM can achieve Cartesian Product Extrapolation. Another way to think about this would be to say, we want to understand how fast does the affine hull grow and capture the Cartesian product set.

Consider the the setting with $m = 2$ attribute dimensions, where each attribute takes d possible values. In such a case, we have d^2 possible attribute combinations. Suppose we sample s attribute vectors z that comprise the support $\mathcal{Z}^{\mathsf{train}}$ uniformly at random (with replacement) from these d^2 possibilites. In the next theorem, we show that if the number of sampled attribute vectors exceeds $8cd\log(d)$, then the affine hull of $\mathcal{Z}^{\mathsf{train}}$ contains all the possible d^2 combinations with a high probability and as a result CRM achieves CPE (proof in Appendix C.3).

Theorem 3. *Consider the setting where $p(.|z)$ follows AED $\forall z \in \mathcal{Z}^{\times}$, $\mathcal{Z}^{\mathsf{train}}$ comprises of s attribute vectors z drawn uniformly at random from $\mathcal{Z}^{\times}$, and the test distribution q satisfies compositional shift characterization. If $s \geq 8cd\log(d/2)$, where d is sufficiently large, $\hat{p}(z|x) = p(z|x), \forall z \in \mathcal{Z}^{\mathsf{train}}, \forall x \in \mathbb{R}^n$, $\hat{q}(z) = q(z), \forall z \in \mathcal{Z}^{\mathsf{test}}$, then the output of CRM (equation 6) matches the test distribution, i.e., $\hat{q}(z|x) = q(z|x)$, $\forall z \in \mathcal{Z}^{\mathsf{test}}, \forall x \in \mathbb{R}^n$, with probability greater than $1 - \frac{1}{c}$.*

$(m = 5, d = 5)$	$(m = 10, d = 10)$	$(m = 20, d = 20)$
1.0	1.0	0.986

Table 2: Numerical experiments to check the probability that the affine hull of random $\mathcal{O}(\mathsf{poly}(m*d))$ one-hot concatenations span the entire set $\mathcal{Z}$. We sample random $3*m*d$ one-hot vectors and report the frequency of times out of 1000 runs a random one-hot concatenation is in the affine hull of the selected set of vectors.

For the more general setting of m attributes, we conjecture that a polynomial growth in md, i.e., $\mathcal{O}(\mathsf{poly}(md))$, groups suffice to generalize to distributions whose support span d^m groups. To support this conjecture, we conduct numerical experiments described in Table 2, where we show that a random $z' \in \mathcal{Z}^{\times}$ is in the affine span of a random set of $\mathcal{O}(md)$ training groups z with a high probability. To summarize, these results point to a surprising fact that, we need to see data from a much smaller number of groups to achieve extrapolation to an exponentially large set.

B.3 Implementation of the Proposed Approach CRM

In a nutshell, CRM consists of: a) training a model of the form of equation 4 by maximum likelihood (equation 5) for trainset group prediction; b) compute extrapolated biases (equation 7); c) infer group probabilities on compositionally shifted test distribution using equation 6. For the case where we have 2 attributes $z = (y, a)$, Figure 1 illustrates a basic architecture using a deep network backbone $\phi(x;\theta)$ followed by a linear mapping (matrix W), and Algorithm 1 provides the associated pseudo-code. The figure's architecture computes the logits $F_{L,B}(x)$ as implemented in the pseudocode. Alternatively to a single linear head whose output we split, we could use separate arbitrary (non-linear) heads to obtain the components for each attribute. Architecture and code can easily be generalized to handle more than 2 attributes.

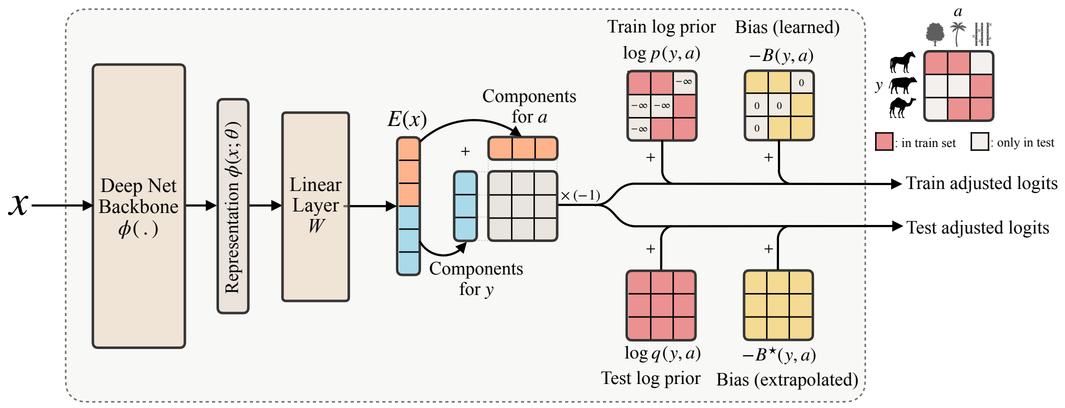


Figure 1: The additive energy classifier trained in CRM computes the logits for each group $z = (y, a)$ by adding the energy components of each attribute via boradcasting. For the train logits, we add the log of the prior probabilities and a learned bias $B(y, a)$ for the groups present in train data. At test time, the log prior term is replaced with the log of the test prior (if available, otherwise assumed to be uniform), and the biases for novel test groups, $B^\star(y, a)$, are extrapolated using Eq.7. Finally, we obtain $p(y, a|x)$ by applying softmax function on the adjusted logits. This adaptation from train to test is possible because of the additive energy distribution $p(x|y, a)$, which allows the model to factorize the distribution into distinct components associated with each attribute.

Algorithm 1: Compositional Risk Minimization (CRM)

Input: training set $\mathcal{D}^{\text{train}}$ with examples (x, y, a), where y is the class to predict and a is an attribute spuriously correlated with y

Output: classifier parameters θ, W, $B^\star$.

- Let $L, B \in \mathbb{R}^{d_y \times d_a}$ be the log prior and the bias terms.
- Define logits: $F_{L,B}(x) := -((W \cdot \phi(x;\theta))_{1:d_y} + (W \cdot \phi(x;\theta))^\top_{d_y+1:d_y+d_a}) + L - B$
- Define log probabilities: $\log p(y, a|x; \theta, W, L, B) := (F_{L,B}(x) - \text{logsumexp}(F_{L,B}(x)))_{y,a}$

Training:

- Estimate log prior L^{train} from $\mathcal{D}^{\text{train}}$; $L^{\text{train}}_{y,a} \leftarrow -\infty$ if (y, a) absent from $\mathcal{D}^{\text{train}}$.
- Optimize θ, W, and B to maximize the log-likelihood over $\mathcal{D}^{\text{train}}$:
  $\theta, W, B \leftarrow \arg\max_{\theta,W,B} \sum_{(x,y,a)\in\mathcal{D}^{\text{train}}} \log p(y, a|x; \theta, W, L^{\text{train}}, B)$
- Extrapolate bias: $B^\star \leftarrow \log\left(\frac{1}{n}\sum_{x\in\mathcal{D}^{\text{train}}} \exp(F_{0,0}(x) - \text{logsumexp}(F_{L^{\text{train}},B}(x)))\right)$

Inference on test point x:

- Compute group probabilities, using $B^\star$, and $L^{\text{unif}} = \log\frac{1}{d_y d_a}$ aiming for shift to uniform prior:
  $q(y, a|x) \leftarrow \exp(\log p(y, a|x; \theta, W, L^{\text{unif}}, B^\star))$
- Marginalize over a to get class probabilities: $q(y|x) \leftarrow \sum_a q(y, a|x)$

C Proofs

Remark on proofs We want to emphasize that the proofs developed here are quite different from related works on compositionality [Dong and Ma, 2022, Wiedemer et al., 2023]. The foundation of proofs is built on a new mathematical object, discrete affine hull. The proof of Theorem 1 cleverly exploits properties of softmax and discrete affine hulls to show how we can learn the correct distribution without involving the intractable partition function in learning. The proof of Theorem 3, uses fundamental ideas from randomized algorithms to arrive at the probabilistic extrapolation guarantees.

The proofs section is structured in the following manner, we first provide the proof for generative extrapolation (Theorem 2), and then provide proof for the main theoretical result of extrapolation of CRM (Theorem 1). This is done to provide intuition to the reader behind extrapolation of the energies via the simpler proof for Theorem 2, which will be useful for the proof of the main Theorem 1. Then we describe the proof for extrapolation of CRM with randomly sampled groups during training (Theorem 3) in Appendix C.3, where we also introduce several important notations that will be useful for proving Theorem 6 and Theorem 7 as well.

Before diving into these proofs, we make an observation that would be handy.

Lemma 1. *If* $z' \in \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$*, i.e.,* $\sigma(z') = \sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z \sigma(z)$*, where* $\langle 1, \alpha_z\rangle = 1$*, then* $\langle \sigma(z'), E(x)\rangle = \sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z \langle \sigma(z), E(x)\rangle$*.*

Proof. $\langle \sigma(z'), E(x)\rangle = \langle \sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z\sigma(z), E(x)\rangle = \sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z \langle \sigma(z), E(x)\rangle$.

□

C.1 Proof for Theorem 2: Extrapolation of Conditional Density

Theorem 2. *If the true and learned distribution ($p(\cdot|z)$ and $\hat{p}(\cdot|z)$) are additive energy distributions, then* $\hat{p}(\cdot|z) = p(\cdot|z), \forall z \in \mathcal{Z}^{\mathsf{train}} \implies \hat{p}(\cdot|z') = p(\cdot|z'), \forall z' \in \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$.

Proof. We start by expanding the expressions for true and estimated log densities below

$$\begin{aligned} -\log\Big[p(x|z)\Big] &= \langle \sigma(z), \mathbb{E}(x)\rangle + \log(\mathbb{Z}(z)),\\ -\log\Big[\hat{p}(x|z)\Big] &= \langle \sigma(z), \hat{E}(x)\rangle + \log(\hat{\mathbb{Z}}(z)). \end{aligned} \tag{10}$$

We equate these densities for the training attributes $z \in \mathcal{Z}^{\mathsf{train}}$. For a fixed $z \in \mathcal{Z}^{\mathsf{train}}$, we obtain that for all $x \in \mathbb{R}^n$

$$\langle \sigma(z), \hat{E}(x)\rangle = \langle \sigma(z), E(x)\rangle + C(z), \tag{11}$$

where $C(z) = \log\big(\mathbb{Z}(z)/\hat{\mathbb{Z}}(z)\big)$. Since $z' \in \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$, we can write $z' = \sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z z$, $\langle 1, \alpha_z\rangle = 1$. From Lemma 1, we know that $\langle \sigma(z'), E(x)\rangle = \sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z \langle \sigma(z), \hat{E}(x)\rangle$.

We use this decomposition and equation 11 to arrive at the key identity below. For all $x \in \mathbb{R}^n$

$$\begin{aligned} \langle \sigma(z'), \hat{E}(x)\rangle &= \sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z \langle \sigma(z), \hat{E}(x)\rangle \\ &= \sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z(\langle \sigma(z), E(x)\rangle + C(z)) \\ &= \Big(\sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z(\langle \sigma(z), E(x)\rangle\Big) + \Big(\sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z C(z)\Big) \\ &= \langle \sigma(z'), E(x)\rangle + \sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z C(z) \end{aligned} \tag{12}$$

From this we can infer that

$$\begin{aligned}\hat{p}(x|z') &= \frac{1}{\hat{\mathbb{Z}}(z')} \exp\Big(- \langle \sigma(z'), \hat{E}(x) \rangle \Big) \\ &= \frac{1}{\hat{\mathbb{Z}}(z')} \exp\Big(- \langle \sigma(z'), E(x) \rangle - \sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z C(z) \Big)\end{aligned} \tag{13}$$

We now use the fact that density integrates to one for continuous random variables (or alternatively the probability sums to one for discrete random variables). Thus

$$\begin{aligned}\int \hat{p}(x|z') dx &= 1 \\ \int \frac{1}{\hat{\mathbb{Z}}(z')} \exp\Big(- \langle \sigma(z'), E(x) \rangle - \sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z C(z) \Big) dx &= 1 \\ \frac{1}{\hat{\mathbb{Z}}(z')} \exp\Big(- \sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z C(z) \Big) \int \exp\Big(- \langle \sigma(z'), E(x) \rangle \Big) dx &= 1 \\ \frac{1}{\hat{\mathbb{Z}}(z')} \exp\Big(- \sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z C(z) \Big) \mathbb{Z}(z') &= 1 \\ \frac{1}{\hat{\mathbb{Z}}(z')} \exp\Big(- \sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z C(z) \Big) &= \frac{1}{\mathbb{Z}(z')}\end{aligned} \tag{14}$$

We substitute equation 14 into equation 13 to obtain

$$\hat{p}(x|z') = \frac{1}{\mathbb{Z}(z')} \exp\Big(- \langle \sigma(z'), E(x) \rangle \Big) = p(x|z'), \forall x \in \mathbb{R}^n \tag{15}$$

□

C.2 Proof for Thorem 1: Extrapolation of CRM

Theorem 1. *Consider the setting where $p(.|z)$ follows AED $\forall z \in \mathcal{Z}^{\times}$, the test distribution q satisfies compositional shift characterization and $\mathcal{Z}^{\mathsf{test}} \subseteq \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$. If $\hat{p}(z|x) = p(z|x), \forall z \in \mathcal{Z}^{\mathsf{train}}, \forall x \in \mathbb{R}^n$ and $\hat{q}(z) = q(z), \forall z \in \mathcal{Z}^{\mathsf{test}}$, then the output of CRM (equation 6) matches the test distribution, i.e., $\hat{q}(z|x) = q(z|x), \forall z \in \mathcal{Z}^{\mathsf{test}}, \forall x \in \mathbb{R}^n$.*

Proof. Since q follows compositional shifts,

$$\log q(x|z) = \log p(x|z) = - \langle \sigma(z), E(x) \rangle - \log \mathbb{Z}(z) \tag{16}$$

We can write it as $- \langle \sigma(z), E(x) \rangle = \log p(x|z) + \log \mathbb{Z}(z)$.

Consider $z' \in \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$. We can express z' as $\sigma(z') = \sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z \sigma(z)$, where $\langle 1, \alpha_z \rangle = 1$.

We use equation 16 and show that the partition function at z' can be expressed as affine combination of partition of the individual points and a correction term. We obtain the following condition. $\forall z' \in \mathcal{Z}^{\mathsf{test}}$, where recall $\mathcal{Z}^{\mathsf{test}} \subseteq \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$,

$$\begin{aligned}\log\big(\mathbb{Z}(z')\big) &= \log\Big(\mathbb{E}_x \Big[\exp\big(- \langle \sigma(z'), E(x) \rangle \big) \Big] \Big), \\ &= \log\Big(\mathbb{E}_x \Big[\exp\Big(- \sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z \langle \sigma(z), E(x) \rangle \Big) \Big] \Big), \\ &= \log\Big(\mathbb{E}_x \Big[\exp\Big(\sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z \big(\log p(x|z) + \log \mathbb{Z}(z) \big) \Big) \Big] \Big) \\ &= \sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z \log \mathbb{Z}(z) + \log \Bigg(\mathbb{E}_x \Big[\exp\Big(\sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z \log p(x|z) \Big) \Big] \Bigg),\end{aligned} \tag{17}$$

where $\mathbb{E}_x[f] = \int_{\tilde{x}\in\mathbb{R}^n} f(\tilde{x})d\tilde{x}$.

Denote the latter term in the above expression as

$$R(\{\alpha_z\}_{z\in\mathcal{Z}^{\text{train}}}) = \log\Big(\mathbb{E}_x\Big[\exp\Big(\sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z \log p(x|z)\Big)\Big]\Big) \tag{18}$$

We now simplify $\log\big(q(x|z')\big)$ using the property of partition function from equation 17 below. $\forall z' \in \mathcal{Z}^{\text{test}}$,

$$\begin{aligned}
&\log\big(q(x|z')\big) = -\langle\sigma(z'), E(x)\rangle - \log\mathbb{Z}(z') \\
&= \sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\Big(\log p(x|z) + \log\mathbb{Z}(z)\Big) - \log\mathbb{Z}(z') \\
&= \sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z \log p(x|z) + \sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\log\mathbb{Z}(z) - \sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\log\mathbb{Z}(z) - R\big(\{\alpha_z\}_{z\in\mathcal{Z}^{\text{train}}}\big) \\
&= \sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\log p(x|z) - R\big(\{\alpha_z\}_{z\in\mathcal{Z}^{\text{train}}}\big)
\end{aligned} \tag{19}$$

We now simplify the first term in the above expression, i.e., $\sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\log p(x|z)$, in terms of $p(z|x)$.

$$\begin{aligned}
\sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\big(\log\big(p(x|z)\big) &= \sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\log\left(\frac{p(z|x)p(x)}{p(z)}\right) \\
&= \sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\Big(\log p(z|x) - \log p(z)\Big) + \log p(x)
\end{aligned} \tag{20}$$

Similarly, $R(\{\alpha_z\}_{z\in\mathcal{Z}^{\text{train}}})$ can be phrased in terms of $p(z|x)$ as follows.

$$\begin{aligned}
R\big(\{\alpha_z\}_{z\in\mathcal{Z}^{\text{train}}}\big) &= \log\Big(\mathbb{E}_x\Big[\exp\Big(\sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\log p(x|z)\Big)\Big]\Big) \\
&= -\sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\log p(z) + \log\Big(\mathbb{E}_{x\sim p(x)}\Big[\exp\Big(\sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\log p(z|x)\Big)\Big]\Big) \\
&= -\sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\log p(z) + S\big(\{\alpha_z\}_{z\in\mathcal{Z}^{\text{train}}}\big),
\end{aligned} \tag{21}$$

where $S\big(\{\alpha_z\}_{z\in\mathcal{Z}^{\text{train}}}\big) = \log\Big(\mathbb{E}_{x\sim p(x)}\Big[\exp\Big(\sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\log p(z|x)\Big)\Big]\Big)$ and $\mathbb{E}_{x\sim p(x)}$ is the expectation w.r.t distribution $p(x)$. We use equation 20, equation 21 to simplify equation 19 as follows.$\forall z' \in \mathcal{Z}^{\text{test}}$,

$$\begin{aligned}
\log q(x|z') &= \sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\log p(z|x) - S\big(\{\alpha_z\}_{z\in\mathcal{Z}^{\text{train}}}\big) + \log p(x) \\
\log\left(\frac{q(z'|x)q(x)}{q(z')}\right) &= \sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\log p(z|x) - S\big(\{\alpha_z\}_{z\in\mathcal{Z}^{\text{train}}}\big) + \log p(x) \\
\log\big(q(z'|x)\big) &= \sum_{z\in\mathcal{Z}^{\text{train}}}\alpha_z\Big(q(z') + \log p(z|x)\Big) - S\big(\{\alpha_z\}_{z\in\mathcal{Z}^{\text{train}}}\big) + \log\left(\frac{p(x)}{q(x)}\right)
\end{aligned} \tag{22}$$

We use translation invariance of softmax to obtain

$$q(z'|x) = \mathsf{Softmax}\Big(\log q(z') + \sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z \log p(z|x) - S\big(\{\alpha_z\}_{z\in\mathcal{Z}^{\mathsf{train}}}\big)\Big)$$
$$q(z'|x) = \mathsf{Softmax}\Big(\log q(z') + \sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z \log p(z|x) - \log\Big(\mathbb{E}_{x\sim p(x)}\Big[\exp\Big(\sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z \log p(z|x)\Big)\Big]\Big)\Big) \tag{23}$$

To avoid cumbersome notation, we took the liberty to show only one input to softmax, other inputs bear the same parametrization, they are computed at other z's. From the above equation it is clear that if the learner knows the marginal distribution over the groups at test time, i.e., $q(z)$ and estimates $p(z|x)$ for all z's in the training distribution's support, i.e., $\mathcal{Z}^{\mathsf{train}}$, then the learner can successfully extrapolate to $q(z'|x)$.

Let us now use the *additive energy classifier* of the form we defined in equation 4 and whose energy $\hat{E}$ and bias $\hat{B}$ we optimized (equation 5) to match $p(z|x)$, so that:

$$p(z|x) = \frac{\exp\Big(-\langle\sigma(z), \hat{E}(x)\rangle + \log\hat{p}(z) - \hat{B}(z)\Big)}{\sum_{\tilde{z}\in\mathcal{Z}^{\mathsf{train}}} \exp\Big(-\langle\sigma(\tilde{z}), \hat{E}(x)\rangle + \log\hat{p}(\tilde{z}) - \hat{B}(\tilde{z})\Big)},$$

Consequently

$$\begin{aligned}&\sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z \log p(z|x) \\ &= \Big(\sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z \Big(-\langle\sigma(z), \hat{E}(x)\rangle + \log p(z) - \hat{B}(z)\Big)\Big) - \log\Big(\sum_{\tilde{z} \in \mathcal{Z}^{\mathsf{train}}} \exp\Big(-\langle\sigma(\tilde{z}), \hat{E}(x)\rangle + \log p(\tilde{z}) - \hat{B}(\tilde{z})\Big)\Big)\end{aligned} \tag{24}$$

where we used the property that $\langle 1, \alpha_z\rangle = 1$.

Let us use this to simplify the last term of equation 23:

$$\begin{aligned}
&\log\Big(\mathbb{E}_{x\sim p(x)}\Big[\exp\Big(\sum_{z \in \mathcal{Z}^{\mathsf{train}}} \alpha_z \log p(z|x)\Big)\Big]\Big) \\
&= \log\left(\mathbb{E}_{x\sim p(x)}\left[\frac{\exp\Big(\sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z\Big(-\langle\sigma(z), \hat{E}(x)\rangle + \log p(z) - \hat{B}(z)\Big)\Big)}{\Big(\sum_{\tilde{z}\in\mathcal{Z}^{\mathsf{train}}} \exp\Big(-\langle\sigma(\tilde{z}), \hat{E}(x)\rangle + \log p(\tilde{z}) - \hat{B}(\tilde{z})\Big)}\right]\right) \\
&= \log\left(\mathbb{E}_{x\sim p(x)}\left[\frac{\exp\Big(\sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z\Big(-\langle\sigma(z), \hat{E}(x)\rangle\Big)}{\Big(\sum_{\tilde{z}\in\mathcal{Z}^{\mathsf{train}}} \exp\Big(-\langle\sigma(\tilde{z}), \hat{E}(x)\rangle + \log p(\tilde{z}) - \hat{B}(\tilde{z})\Big)}\right]\exp\Big(\sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z\Big(\log p(z) - \hat{B}(z)\Big)\Big)\right) \\
&= \log\left(\mathbb{E}_{x\sim p(x)}\left[\frac{\exp\Big(\sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z\Big(-\langle\sigma(z), \hat{E}(x)\rangle\Big)}{\Big(\sum_{\tilde{z}\in\mathcal{Z}^{\mathsf{train}}} \exp\Big(-\langle\sigma(\tilde{z}), \hat{E}(x)\rangle + \log p(\tilde{z}) - \hat{B}(\tilde{z})\Big)}\right]\right) + \sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z\Big(\log p(z) - \hat{B}(z)\Big) \\
&= \log\left(\mathbb{E}_{x\sim p(x)}\left[\frac{\exp\Big(-\langle\sigma(z'), \hat{E}(x)\rangle\Big)}{\Big(\sum_{\tilde{z}\in\mathcal{Z}^{\mathsf{train}}} \exp\Big(-\langle\sigma(\tilde{z}), \hat{E}(x)\rangle + \log p(\tilde{z}) - \hat{B}(\tilde{z})\Big)}\right]\right) + \sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z\Big(\log p(z) - \hat{B}(z)\Big) \\
&= B^\star(z') + \sum_{z\in\mathcal{Z}^{\mathsf{train}}} \alpha_z\Big(\log p(z) - \hat{B}(z)\Big)
\end{aligned} \tag{25}$$

where we used Lemma 1, and $B^\star$ is as defined in equation 7.

Let us also define $c(x) = \log \Big(\sum_{\tilde{z} \in \mathcal{Z}^{\textsf{train}}} \exp \Big(-\langle \sigma(\tilde{z}), \hat{E}(x) \rangle + \log p(\tilde{z}) - \hat{B}(\tilde{z}) \Big) \Big)$ so that we can reexpress equation 24 as:

$$\sum_{z \in \mathcal{Z}^{\textsf{train}}} \alpha_z \log p(z|x) = \left(\sum_{z \in \mathcal{Z}^{\textsf{train}}} \alpha_z \Big(-\langle \sigma(z), \hat{E}(x) \rangle + \log p(z) - \hat{B}(z) \Big) \right) - c(x) \tag{26}$$

Subtracting equation 25 from equation 26 we get:

$$\begin{aligned}
&\sum_{z \in \mathcal{Z}^{\textsf{train}}} \alpha_z \log p(z|x) - \log \left(\mathbb{E}_{x \sim p(x)} \Big[\exp \Big(\sum_{z \in \mathcal{Z}^{\textsf{train}}} \alpha_z \log p(z|x) \Big) \Big] \right) \\
&= \sum_{z \in \mathcal{Z}^{\textsf{train}}} \alpha_z \Big(-\langle \sigma(z), \hat{E}(x) \rangle + \log p(z) - \hat{B}(z) \Big) - c(x) \\
&\quad - B^\star(z') - \sum_{z \in \mathcal{Z}^{\textsf{train}}} \alpha_z \Big(\log p(z) - \hat{B}(z) \Big) \\
&= \sum_{z \in \mathcal{Z}^{\textsf{train}}} \alpha_z \Big(-\langle \sigma(z), \hat{E}(x) \rangle \Big) - c(x) - B^\star(z') \\
&= -\langle \sigma(z'), \hat{E}(x) \rangle - c(x) - B^\star(z')
\end{aligned} \tag{27}$$

Substituting this inside equation 23 yields

$$\begin{aligned}
q(z'|x) &= \textsf{Softmax}\Big(\log q(z') - \langle \sigma(z'), \hat{E}(x) \rangle - c(x) - B^\star(z') \Big) \\
&= \textsf{Softmax}\Big(-\langle \sigma(z'), \hat{E}(x) \rangle + \log q(z') - B^\star(z') \Big)
\end{aligned} \tag{28}$$

where we removed the $c(x)$ term as softmax is invariant to addition of terms that do not depend on z'. If $\hat{q}(z') = q(z'), \forall z' \in \mathcal{Z}^{\textsf{test}}$, then the expression in RHS corresponds to $\hat{q}(z'|x)$, as we had defined it in equation 6, before stating our theorem. Thus $q(z'|x) = \hat{q}(z'|x)$. This completes the proof.

$\square$

C.3 Proof for Theorem 3: Extrapolation from a Small Set of Attribute Combinations to All Attribute Combinations

Theorem 3. *Consider the setting where $p(.|z)$ follows AED $\forall z \in \mathcal{Z}^{\times}$, $\mathcal{Z}^{\mathsf{train}}$ comprises of s attribute vectors z drawn uniformly at random from $\mathcal{Z}^{\times}$, and the test distribution q satisfies compositional shift characterization. If $s \geq 8cd\log(d/2)$, where d is sufficiently large, $\hat{p}(z|x) = p(z|x), \forall z \in \mathcal{Z}^{\mathsf{train}}, \forall x \in \mathbb{R}^n$, $\hat{q}(z) = q(z), \forall z \in \mathcal{Z}^{\mathsf{test}}$, then the output of CRM (equation 6) matches the test distribution, i.e., $\hat{q}(z|x) = q(z|x)$, $\forall z \in \mathcal{Z}^{\mathsf{test}}, \forall x \in \mathbb{R}^n$, with probability greater than $1 - \frac{1}{c}$.*

In order to prove Theorem 3 we first establish some basic lemmas. In the first lemma below, we consider a setting with two attributes, where each attribute takes two possible values, i.e., $m = 2$ and $d = 2$. In this setting there are four possible one-hot vectors z^1, z^2, z^3, z^4. We first show that each z^i can be expressed as an affine combination of the remaining three.

Lemma 2. *If $m = 2, d = 2$, then there are four possible concatenated one-hot vectors z denoted z^1, z^2, z^3, z^4. Each z^i can be expressed as an affine combination of the remaining.*

Proof. Below we explicitly show how each z^i can be expressed in terms of other z^j's.

$$(+1) \cdot \begin{bmatrix}0\\1\\0\\1\end{bmatrix} + (-1) \cdot \begin{bmatrix}0\\1\\1\\0\end{bmatrix} + (+1) \cdot \begin{bmatrix}1\\0\\1\\0\end{bmatrix} = \begin{bmatrix}1\\0\\0\\1\end{bmatrix} \tag{29}$$

$$(-1) \cdot \begin{bmatrix}0\\1\\0\\1\end{bmatrix} + (+1) \cdot \begin{bmatrix}0\\1\\1\\0\end{bmatrix} + (+1) \cdot \begin{bmatrix}1\\0\\0\\1\end{bmatrix} = \begin{bmatrix}1\\0\\1\\0\end{bmatrix} \tag{30}$$

$$(+1) \cdot \begin{bmatrix}1\\0\\1\\0\end{bmatrix} + (+1) \cdot \begin{bmatrix}0\\1\\0\\1\end{bmatrix} + (-1) \cdot \begin{bmatrix}1\\0\\0\\1\end{bmatrix} = \begin{bmatrix}0\\1\\1\\0\end{bmatrix} \tag{31}$$

$$(-1) \cdot \begin{bmatrix}1\\0\\1\\0\end{bmatrix} + (+1) \cdot \begin{bmatrix}0\\1\\1\\0\end{bmatrix} + (+1) \cdot \begin{bmatrix}1\\0\\0\\1\end{bmatrix} = \begin{bmatrix}0\\1\\0\\1\end{bmatrix} \tag{32}$$

□

We illustrate the setting of Lemma 2 in Figure 2. We now understand an implication of Lemma 2. Let us consider the setting where $m = 2$ and $d > 2$. Consider a subset of four groups $\{(i,j),(i',j),(i,j'),(i',j')\}$. Under one-hot concatenations these groups are denoted as $z^1 = [\underbrace{0,\cdots,1_i,\cdots 0}_{\text{first attribute}}, \underbrace{0,\cdots,1_j,\cdots 0}_{\text{second attribute}}]$, $z^2 = [0,\cdots,1_{i'},\cdots 0, 0,\cdots,1_j,\cdots 0]$, $z^3 = [0,\cdots,1_i,\cdots 0, 0,\cdots,1_{j'},\cdots 0]$, and $z^4 = [0,\cdots,1_{i'},\cdots 0, 0,\cdots,1_{j'},\cdots 0]$. Observe that using Lemma 2, we get $z^4 = z^2 + z^3 - z^1$. Similarly, we can express every other z^i in terms of rest of z^j's in the the set $\{(i,j),(i',j),(i,j'),(i',j')\}$.

In the setting when $m = 2$ and $d \geq 2$, the total number of possible values z takes is d^2. Each group recall is associated with attribute vector $z = [z_1, z_2]$, where $z_1 \in \{1,\cdots,d\}$ and $z_2 \in \{1,\cdots d\}$. The set of all possible values of z be visualized as $d \times d$ grid in this notation. We call this $d \times d$ grid as G. We will first describe a specific approach of selecting observed groups z for training, which shows that with just $2d - 1$ it is possible to affine span all the possible d^2 groups in the grid G. We leverage the insights from this approach and show that with a randomized approach of selecting groups, we can continue to affine span d^2 groups with $\mathcal{O}(d\log(d))$ groups.

Denote the set of observed groups as N. Suppose that their affine hull contains all the points in a subgrid $S \subseteq G$ of size $m \times n$. Let the subgrid $S = \{x_1,\cdots,x_m\} \times \{y_1,\cdots,y_n\}$. Without loss of generality, we can permute the points and make the subgrid contiguous as follows $S =$

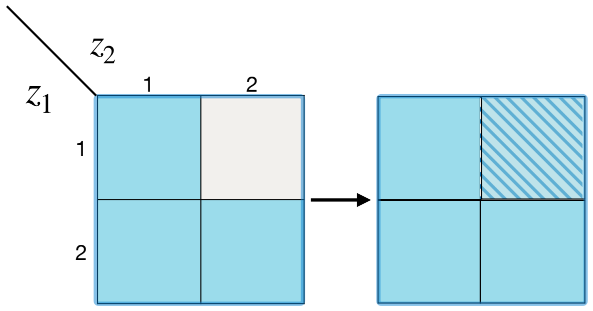


Figure 2: Setting of two binary attributes. Illustration of extrapolation from three groups to the remaining fourth group. Three dark colored groups indicate the observed groups and the light colored shaded group indicates the group that is the affine combination of the three observed groups.

$\{1, \cdots, m\} \times \{1, \cdots, n\}$. Next, we add a new point $g = (g_x, g_y) \in G$ but $g \notin S$. We argue that if $g_x \in \{1, \cdots, m\}$, then the affine hull of $N \cup \{g\}$ contains a larger subgrid of size $m \times (n + 1)$. Similarly, we want to argue that if $g_y \in \{1, \cdots, n\}$, then the affine hull of $N \cup \{g\}$ contains a larger subgrid of size $(m + 1) \times n$. Define C_x as the Cartesian product of $\{g_x\}$ with $\{1, \cdots, n\}$, i.e., $C_x = \{(g_x, 1), (g_x, 2), \cdots, (g_x, n)\}$. Define C_y as the Cartesian product of $\{1, \cdots, m\}$ with $\{g_y\}$, i.e., $C_y = \{(1, g_y), (2, g_y), \cdots, (m, g_y)\}$.

Theorem 4. *Suppose the affine hull of the observed set N contains a subgrid S of size $m \times n$. If the new point $g = (g_x, g_y)$ shares the x-coordinate with a point in S, and $g \notin S$, then the the affine hull of $N \cup \{g\}$ contains $S \cup C_y$.*

Proof. We write the set of observed groups N as $N = \{z^{\theta_j}\}_j$. The affine hull of N contains $S = \{1, \cdots, m\} \times \{1, \cdots, n\}$. We observe a new group $g \notin S$, which shares its x coordinate with one of the points in S. Without loss of generality let this point be $g = (1, n + 1)$ (if this were not the case, then we can always permute the columns and rows to achieve such a configuration). Consider the triplet – $(z^1, z^2, z^3) = \big((1, n), (2, n), (1, n+1)\big)$. Observe that z^1, z^2, z^3, z^4 form a 2×2 subgrid, where $z^4 = (2, n + 1)$. We use Lemma 2 to infer that the fourth point $z^4 = (2, n + 1)$ on this 2×2 subgrid can be obtained as an affine combination of this triplet, i.e., $z^4 = \alpha z^1 + \beta z^2 + \gamma z^3$. Since z^1, z^2 are in the affine hull of N, they can be written as an affine combination of seen points in N as follows $z^1 = \sum_{k\in N} a_k z^{\theta_k}$, $z^2 = \sum_{k\in N} b_k z^{\theta_k}$. As a result, we obtain

$$\begin{aligned} z^4 &= \alpha z^1 + \beta z^2 + \gamma z^3 = \alpha\Big(\sum a_k z^{\theta_k}\Big) + \beta\Big(\sum b_k z^{\theta_k}\Big) + \gamma z^3 \\ &= \sum_{k\in N} \Big(\alpha a_k + \beta b_k\Big) z^{\theta_k} + \gamma z^3 \end{aligned} \tag{33}$$

Observe that $\sum_k(\alpha a_k + \beta b_k) = (\alpha \sum_k a_k + \beta \sum_k b_k) = \alpha + \beta$. Since $\alpha + \beta + \gamma = 1$, z^4 is an affine combination of points in $N \cup \{g\}$. Thus we have shown the claim for the point $(2, n + 1)$. We can repeat this claim for point $(3, n + 1)$ and so on until we reach $(m, n + 1)$ beyond which there would be no points in S that are expressed as affine combination of N. We can make this argument formal through induction. We have already shown the base case above. Suppose all the points $(j, n + 1)$ in $j \le i < m$ are in the affine hull of $N \cup \{g\}$. Consider the point $z^4 = (i + 1, n + 1)$. Construct the triplet $(z^1, z^2, z^3) = \big((i, n), (i, n + 1), (i + 1, n)\big)$. Again from Lemma 2, it follows that $z^4 = \alpha z^1 + \beta z^2 + \gamma z^3$. We substitute z^1, z^2 and z^3 with their corresponding affine combinations. $z^4 = \alpha \sum_{k\in N\cup\{g\}} a_k z^{\theta_k} + \beta \sum_{k\in N\cup\{g\}} b_k z^{\theta_k} + \gamma \sum_{k\in N\cup\{g\}} c_k z^{\theta_k}$. Since $\sum_{k\in N\cup\{g\}} \alpha a_k + \beta b_k + \gamma c_k = 1$, it follows that z^4 is an affine combination of z^1, z^2 and z^3. This completes the proof.

□

We now describe a simple deterministic procedure that helps us understand how many groups we need to see before we are guaranteed that the affine hull of seen points span the whole grid $G = \{1, \cdots, d\} \times \{1, \cdots, d\}$.

- We start with a base set of three points – $B = \{(1,1),(1,2),(2,1)\}$. From Lemma 2, the affine hull contains $(2,2)$.
- For each $i \in \{2, \cdots, d-1\}$ add the points $(1, i+1), (i+1, 1)$ to the set B. From Theorem 4, it follows that affine hull of $B \cup \{(1, i+1)\} \cup \{(i+1, 1)\}$ contains $(i+1 \times i+1)$ subgrid $\{1, \cdots, i+1\} \times \{1, \cdots, i+1\}$ (here we apply Theorem 4 in two steps once for the addition of $(1, i+1)$ and then for the addition of $(i+1, 1)$.

At the end of the above procedure B contains $2d-1$ points and its affine hull contains the grid G. We illustrate this procedure in Figure 3.

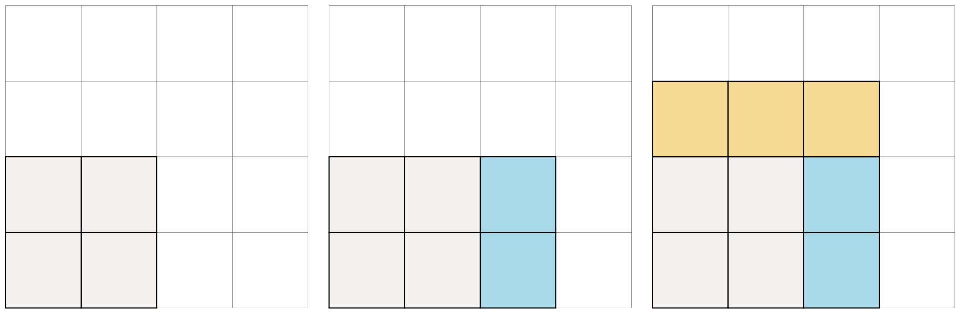

(a) Step 1 (b) Step 2 (c) Step 3

Figure 3: Illustration of steps of the deterministic sampling procedure for a 4×4 grid. (a) shows the base set, (b) add a group in blue and the affine hull extends to include all the blue cells, (c) add a group in yellow and the affine hull extends to include all yellow cells.

We now discuss a randomized procedure that also allows us to span the entire grid G with $\mathcal{O}(d \log(d))$ groups. The idea of the procedure is to start with a base set of groups and construct their affine hull S. Then we wait to sample a group g that is outside this affine hull. If this sampled group shares the x coordinate with affine hull of B denoted as S, then we expand the subgrid by one along with y coordinate. Similarly, we also wait for a point that shares a y coordinate and then we expand the subgrid by one along the x coordinate.

We use S_x to denote the distinct set of x-coordinates that appear in S and same goes for S_y. We write $g = (g_x, g_y)$. The procedure goes as follows.

Set $S = \emptyset, B = \emptyset$ and $\mathsf{Flag} = x$.

- Sample a group g from G uniform at random. Update $B = B \cup \{g\}, S = S \cup \{g\}$.
- While $S \neq G$, sample a group g from G uniform at random.
  - If $\mathsf{Flag} == x$, $g_x \in S_x$, $g \notin S$, then update $B = B \cup \{g\}$, $S = S \cup (S_x \times \{g_y\})$ and $\mathsf{Flag} = y$.
  - If $\mathsf{Flag} == y$, $g_y \in S_y$, $g \notin S$, then update $B = B \cup \{g\}$, $S = S \cup (\{g_x\} \times S_y)$ and $\mathsf{Flag} = x$.

In the above procedure, in every step in the while loop a group g is sampled. Whenever the Flag flips from x to y, then following Theorem 4, the updated set S belongs to the affine hull of B. We can say the same when Flag flips from y to x. In the next theorem, we will show that the while loop terminates after $8cd\log(d)$ steps with a high probability and the affine hull of B contains the entire grid G. We follow this strategy. We count the time it takes for Flag to flip from x to y (from y to x) as it grows the size of S from a $k \times k$ subgrid to $k \times (k+1)$ ($k \times (k+1)$ subgrid to $(k+1) \times (k+1)$) subgrid.

Theorem 5. *Suppose we sample the groups based on the randomized procedure described above. If the number of sampled groups is greater than $8cd\log(d)$, then $G \subseteq \mathsf{DAff}(B)$ with a probability greater than equal to $1 - \frac{1}{c}$.*

Proof. We take the first group g that is sampled. Without loss of generality, we say this group is $(1,1)$.

Suppose the Flag is set to x. Define an event A_1^k: newly sampled $g = (g_x, g_y)$ shares x-coordinate with a point in S (size $k \times k$), $g \notin S$. Under these conditions Flag flips from x to y. To compute the probability of this event let us count the number of scenarios in which this happens. If g_x takes one of the k values in S_x and g_y takes one of the remaining $(d-k)$, then the event happens. As a result, the probability of this event is $P(A_1^k) = \frac{(k)(d-k)}{d^2}$.

Suppose the Flag is set to y. Define an event A_2^k: newly sampled $g = (g_x, g_y)$ shares y-coordinate with a point in S (size $k \times (k+1)$) and $g \notin S$. Under these conditions Flag flips from y to x. The probability of this event is$P(A_2^k) = \frac{(k+1)(d-k)}{d^2}$.

Define T_1^k as the number of groups that need to be sampled before A_1^k occurs. Define T_2^k as the number of groups that need to be sampled before A_2^k occurs. Observe that after $T_1^k + T_2^k$ number of sampled groups the size of the current subgrid S, which is in the affine hull of B, grows to $(k+1) \times (k+1)$.

Define $T_{\mathsf{sum}} = \sum_{k=1}^{d-1}(T_1^k + T_2^k)$. T_{sum} is the total number of groups sampled before the affine span of the observed groups B contains the grid G.

We compute

$$\begin{aligned}
&\mathbb{E}[T_{\mathsf{sum}}] = \sum_{k=1}^{d-1}(\mathbb{E}[T_1^k] + \mathbb{E}[T_2^k]) \\
&\sum_{k=1}^{d}\mathbb{E}[T_1^k] = \sum_{k=1}^{d-1} d^2/(k(d-k)) = 2\sum_{k=1}^{(d-1)/2} d^2/(k(d-k)) \\
&2\sum_{k=1}^{(d-1)/2} d^2/(k(d-k)) = 2d\sum_{k=1}^{(d-1)/2}\left[\frac{1}{k} + \frac{1}{d-k}\right] \approx 4d\log((d-1)/2)
\end{aligned} \tag{34}$$

Similarly, we obtain a similar bound for $\sum_{k=1}^{d-1}\mathbb{E}[T_2^k]$.

$$\begin{aligned}
&\sum_{k=1}^{d}(\mathbb{E}[T_2^k] = \sum_{k=1}^{d-1} d^2/((k+1)(d-k)) = 2\sum_{k=1}^{(d-1)/2} d^2/((k+1)(d-k)) \\
&2\sum_{k=1}^{(d-1)/2} d^2/((k+1)(d-k)) \le 2d\sum_{k=1}^{(d-1)/2}\left[\frac{1}{k+1} + \frac{1}{d-k}\right] \approx 4d\log((d-1)/2)
\end{aligned} \tag{35}$$

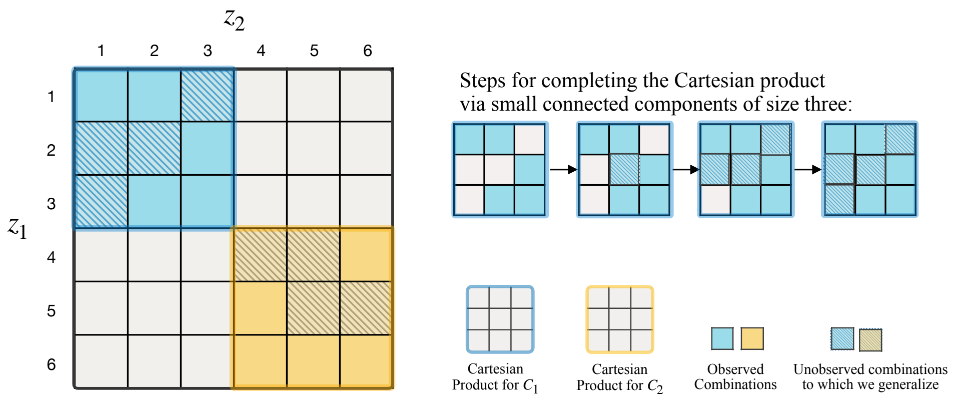
Overall $\mathbb{E}[T_{\mathsf{sum}}] \approx 8d\log(d/2)$. From Markov inequality, it immediately follows that $P(T_{\mathsf{sum}} \le 8cd\log(d/2)) \ge 1 - \frac{1}{c}$. In the above approximations, we use $\sum_{i=1}^{d}\frac{1}{i} \approx \log d + \gamma$, where γ is Euler's constant. We drop γ as its a constant, which can always be absorbed by adapting the constant c.

$\square$

Finally, we use the results proved above to complete the proof for Theorem 3. Suppose the support of training distribution $p(z)$ contains s groups. We know that these s groups are drawn uniformly at random. From Theorem 5, it is clear that if s grows as $\mathcal{O}(d\log d)$, then with a high probability the entire grid of d^2 combinations is contained in the affine span of these observed groups. This can be equivalently stated as $\mathcal{Z}^{\times} \subseteq \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$ with a probability greater than equal to $1 - \frac{1}{c}$. If $\mathcal{Z}^{\times} \subseteq \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$, then from the assumption of compositional shifts, it follows that $\mathcal{Z}^{\mathsf{test}} \subseteq \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$. We can now use Theorem 1 and arrive at our result. This completes the proof.

C.4 Discrete Affine Hull: A Closer Look

In the next result, we aim to give a characterization of discrete affine hull that helps us give a two-dimensional visualization of $\mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$. Before we even state the result, we illustrate discrete affine hull of a 6×6 grid. Consider the 6×6 grid shown in Figure 4. The attribute combinations



z_2
1
2
3
4
5
6
z_1
Steps for completing the Cartesian product via small connected components of size three:
Cartesian Product for C_1
Cartesian Product for C_2
Observed Combinations
Unobserved combinations to which we generalize

Figure 4: Illustration of the discrete affine hull. Each cell in the 6×6 grid represents an attribute combination, where observed combinations are solid-colored. The elements in blue form one connected component, C_1, and the elements in yellow form another connected component, C_2. Extrapolation is possible for unobserved combinations, represented by the crosshatched cells, as long as the test distribution samples from the Cartesian products of the connected components. The steps for completing the Cartesian product visually shows the intuition behind the extrapolation process.

corresponding to the observed groups are shown as solid colored cells (blue and yellow). The light shaded elements (blue and yellow) denote the set of groups that belong to the affine hull of the solid colored groups. We now build the characterization that helps explain this visualization.

We introduce a graph on the attribute vectors observed. Each vertex corresponds to the attribute vector, i.e., $[z_1, z_2]$. There is an edge between two vertices if the Hamming distance between the attribute vectors is one. A connected component is a subgraph in which all vertices are connected, i.e., between every pair in the subgraph there exists a path. Let us start by making an observation about the connected components in this graph.

We consider a partition of observed groups into K maximally connected components, $\{C_1, \cdots, C_K\}$. Define C_{ij} as the set of values the j^{th} component takes in the i^{th} connected component. Observe that $C_{ij} \cap C_{lj} = \emptyset$ for $i \neq l$. Suppose this was not that case and $C_{ij} \cap C_{lj} \neq \emptyset$. In such a case, there exists a point in C_i and another point in C_l that share the j^{th} component. As a result, the two points are connected by an edge and hence that would connect C_i and C_j. This contradicts the fact that C_i and C_j are maximally connected, i.e., we cannot add another vertex to the graph while maintaining that there is a path between any two points in the component. In what follows, we will show that the afine hull of C_j is $C_{j1} \times C_{j2}$, which is the Cartesian product extension of set C_j. Next, we give some definitions and make a simple observation that allows us to think of sets $C_{j1} \times C_{j2}$ as subgrids, which are easier to visualize.

Definition 1. ***Contiguous connected component:*** *For each coordinate $j \in \{1, 2\}$, consider the smallest value and the largest value assumed by it in the connected component C and call it $\min_j$ and $\max_j$. We say that the connected component C is contigous if each value in the set $\{\min_j, \min_j + 1, \cdots, \max_j - 1, \max_j\}$ is assumed by some point in C for all $j \in \{1, 2\}$.*

Smallest subgrid containing a contigous connected component C: The range of values assumed by j^{th} coordinate in C, where $j \in \{1, 2\}$, are $\{\min_j, \cdots, \max_j\}$. The subgrid $\{\min_1, \cdots \max_1\} \times \{\min_2, \cdots \max_2\}$ is the smallest subgrid containing C. Observe that this subgrid is the smallest grid containing C because if we drop any column or row, then some point taking that value in C will not be in the subgrid anymore.

The groups observed at training time can be divided into K maximally connected components $\{C_1, \cdots, C_K\}$. We argue that without any loss of generality each of these components are contiguous. Suppose some of the components in $\{C_1, \cdots, C_K\}$ are not contiguous. We relabel the first coordinate as $\pi(c_{i1}^r) = \sum_{j<i} |C_{j1}| + r$, where c_{i1}^r is the r^{th} point in C_{i1}. We can similarly relabel the second coordinate as well. Under the relabeled coordinates, each component is maximally connected and contiguous. Also, under this relabeling the Cartesian products $C_{j1} \times C_{j2}$ correspond to the smallest

subgrid containing C_j. Let us go back to the setting of Figure 4. The sets of observed groups shown in solid blue and solid yellow form two connected components C_1 and C_2 respectively. Their Cartesian product extensions are shown as well in the Figure 4. Since the connected components were contiguous the Cartesian product extensions correspond to smallest subgrids containing the respective connected component.

Theorem 6. *Given the partition of training support as $\mathcal{Z}^{\mathsf{train}} = \{C_1, \cdots, C_K\}$, we have:*

- *The affine span of a contiguous connected component C is the smallest subgrid that contains that connected component C.*
- *The affine span of the union over disjoint contiguous connected components is the union of the smallest subgrids that contain the respective connected components.*

Proof. C denotes the connected component under consideration and the smallest subgrid containing it is S. Denote the affine span of C as A. We first show that the subgrid $S \subseteq A$.

We start with a target point $t = (t_1, t_2)$ inside S. We want show that the one-hot concatention of this point t can be expressed as an affine combination of the points in C.

Firstly, if t is already in C, then the point is trivially in the affine span. If that is not the case, then let us proceed to more involved cases. Consider the shortest path joining a point of the form $(t_1, s_2) \in C$, where $s_2 \neq t_2$, and a point of the form $(s_1, t_2) \in C$, where $s_1 \neq t_1$. If such points do not exist, then t cannot be in S, which is a contradiction.

We assign a weight of $(+1)$ to the concatenation of one-hot encodings of the point (t_1, s_2). We then traverse the path until we encounter a point where s_2 changes, note that such a point has to occur because of existence of (s_1, t_2) on the path. We call this point $v = (\tilde{s}_1^{'}, s_2^{'})$. The point before v on the path is $w = (\tilde{s}_1^{'}, s_2)$. We assign a weight of (-1) to w. We summarize the path seen so far below. We also write the weights assigned to the points

$$\begin{aligned} s &= (t_1, s_2) && (+1) \\ u &= (s_1^{'}, s_2) && \\ &\vdots && \\ w &= (\tilde{s}_1^{'}, s_2) && (-1) \\ v &= (\tilde{s}_1^{'}, s_2^{'}) && \end{aligned} \tag{36}$$

After w, we have a weight of $+1$ assigned to t_1, -1 assigned to $\tilde{s}_1^{'}$ (note that $\tilde{s}_1^{'}$ cannot be t_1, this follows from the fact that we are on shortest path between points of the form (t_1, s_2) and (s_1, t_2)). We call this state S_1. After w, we wait for a point on the path where $\tilde{s}_1^{'}$ changes or we reach the terminal state (s_1, t_2). The latter can happen if $\tilde{s}_1^{'} = s_1$. In the latter case, we assign a weight $(+1)$ to the terminal state and thus the final weights are $(+1)$ for t_1 and t_2 and zero for everything else. This leads to the desired affine combination. We call this state T_1, corresponding to terminal state.

Now suppose we were in a situation where we reach a point $q = (s_1^{+}, \tilde{s}_2^{'})$. The point before q is $r = (\tilde{s}_1^{'}, \tilde{s}_2^{'})$. We assign a weight of $(+1)$ to r. We summarize the path seen after encountering w below.

$$\begin{aligned} v &= (\tilde{s}_1^{'}, s_2^{'}) && \\ &\vdots && \\ r &= (\tilde{s}_1^{'}, \tilde{s}_2^{'}) && (+1) \\ q &= (s_1^{+}, \tilde{s}_2^{'}) && \end{aligned} \tag{37}$$

After r, we have a weight of $+1$ assigned to t_1 and a weight of $+1$ assigned to $\tilde{s}_2^{'}$. We call this state S_2. After r, we wait for a point where $\tilde{s}_2^{'}$ changes. It could be that $\tilde{s}_2^{'}$ changes to t_2. The state before it is say $u = (s_1, \tilde{s}_2^{'})$ and last state $e = (s_1, t_2)$. Assign a weight of -1 to u and assign a weight of $+1$ to e. Thus we achieve the target as affine combination of points on the path. We call this state T_2, corresponding to the terminal state.

Now let us consider the other possibility that the terminal state has not been reached. We call such a point $m = (\tilde{s}_1^+, \tilde{s}_2^+)$. The point that occurs before this point is $l = (\tilde{s}_1^+, \tilde{s}_2^{'})$. We assign a weight of (-1) to l. We summarize the path taken below.

$$\begin{aligned} q &= (s_1^+, \tilde{s}_2^{'}) \\ &\vdots \\ l &= (\tilde{s}_1^+, \tilde{s}_2^{'}) && (-1) \\ m &= (\tilde{s}_1^+, \tilde{s}_2^+) \end{aligned} \tag{38}$$

After l, t_1 is assigned a weight of $+1$ and $\tilde{s}_1^+$ is assigned a weight of -1. We reach the state S_1 again. From this point on, the same steps repeat. We keep cycling between S_1 and S_2 until we reach the terminal state from either S_1 or S_2 at which point we achieve the desired affine combination. The cycling of states only goes on for a finite number of steps as the entire path we are concerned with has a finite length. We show the process in Figure 5. Thus $S \subseteq A$.

We now make an observation about the set A, which is the affine hull of set C. Suppose the first coordinate takes values between $\{\min_1, \cdots, \max_1\}$. The corresponding one-hot encodings of the first coordinate are written as $\{\mathrm{onehot}(\min_1), \cdots, \mathrm{onehot}(\max_1)\}$. Now consider a value c which is not in $\{\min_1, \cdots, \max_1\}$. We claim that no affine combination of vectors in $\{\mathrm{onehot}(\min_1), \cdots, \mathrm{onehot}(\max_1)\}$ can lead to $\mathrm{onehot}(c)$. We justify this claim as follows. Observe that no vector in $\{\mathrm{onehot}(\min_1), \cdots, \mathrm{onehot}(\max_1)\}$ has a non-zero entry in the same coordinate where $\mathrm{onehot}(c)$ is also non-zero. Hence, any affine combination of vectors in $\{\mathrm{onehot}(\min_1), \cdots, \mathrm{onehot}(\max_1)\}$ will always have a zero weight in the entry where $\mathrm{onehot}(c)$ is non-zero. It is now clear that the first component of affine hull of A is always between $\{\min_1, \cdots, \max_1\}$. Similarly, the second component of affine hull of A is always between $\{\min_2, \cdots, \max_2\}$. Therefore, $A \subseteq S$. As a result, $A = S$. Another way to say this is that $\mathsf{DAff}(C_j) = C_{j1} \times C_{j2}$.

We now move to the second part of the theorem. We have already shown that $\mathsf{DAff}(C_j) = C_{j1} \times C_{j2}$. We now want to show that

$$\mathsf{DAff}\big(\bigcup_{j=1}^{K} C_j\big) = \bigcup_{j=1}^{K} \Big(C_{j1} \times C_{j2}\Big)$$

Observe that $\mathsf{DAff}(A) \subseteq \mathsf{DAff}(A \cup B)$ and $\mathsf{DAff}(B) \subseteq \mathsf{DAff}(A \cup B)$, which implies $\mathsf{DAff}(A) \cup \mathsf{DAff}(B) \subseteq \mathsf{DAff}(A \cup B)$. Therefore, from the first part and this observation it follows that $\bigcup_{j=1}^{K} \Big(C_{j1} \times C_{j2}\Big) \subseteq \mathsf{DAff}\big(\bigcup_{j=1}^{K} C_j\big)$. We now show $\mathsf{DAff}\big(\bigcup_{j=1}^{K} C_j\big) \subseteq \bigcup_{j=1}^{K} \Big(C_{j1} \times C_{j2}\Big)$.

Take the K maximally connected components $\{C_1, \cdots, C_K\}$ and let the set of respective smallest subgrids containing them be $\{S_1, \cdots, S_K\}$. Define a point z' as the affine combination of points across these components as $z' = \sum_{i=1}^{K} \sum_{j=1}^{N_i} \alpha_{ij} z_{ij}$, where z_{ij} is the j^{th} point in C_i, which contains N_i points. We can also write z' as $z' = \sum_{i=1}^{K} \Big(\sum_{j=1}^{N_i} \alpha_{ij}\Big) \sum_{j=1}^{N_i} \frac{\alpha_{ij}}{\sum_{j=1}^{N_i} \alpha_{ij}} z_{ij}$. Define $z_i' = \sum_{j=1}^{N_i} \frac{\alpha_{ij}}{\sum_{j=1}^{N_i} \alpha_{ij}} z_{ij}$. Observe that z_i' is in the affine combination of points in C_i and hence z_i' is a point in S_i. Let $\tilde{\alpha}_i = \sum_{j=1}^{N_i} \alpha_{ij}$. In this notation, we can see z' is an affine combination of z_i''s denoted as $\sum_{i=1}^{K} \tilde{\alpha}_i z_i'$. In this representation, there is at most one point per S_i in the affine combination. There are two cases to consider. In the first case, exactly one component $\tilde{\alpha}_i$ is non-zero and rest all components are zero. In the second case, at least two components $\tilde{\alpha}_i$'s are non-zero. In this setting, we can only keep the non-zero $\tilde{\alpha}_i$'s in the sum denoted as $\sum_i \tilde{\alpha}_i z_i'$. Suppose $z_i' = (e_p, e_q)$ (without loss of generality), where e_p is one-hot vector that is one on the p^{th} coordinate. Observe that no other point in the sum $\sum_i \tilde{\alpha}_i z_i'$ will have a non-zero contribution on the p^{th} coordinate. As

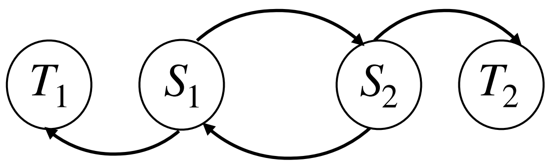

$$\begin{tikzcd} T_1 & S_1 \arrow[l, bend left] \arrow[r, bend left] & S_2 \arrow[l, bend left] \arrow[r, bend left] & T_2 \end{tikzcd}$$

Figure 5: Illustration of state transition in proof of Theorem 6.

a result, in the final vector the p^{th} coordinate of the first attribute will take the value $0 < \tilde{\alpha}_i < 1$. This point is not a valid point in the set of all possible one-hot concatenations $\mathcal{Z}$ and hence it does not belong to the affine hull $\mathsf{DAff}\big(\bigcup_{j=1}^{K} C_j\big)$. Thus we are left with the first case. Observe that in the first case, we will always generate a point in one of the $\mathsf{DAff}(C_j)$, where $j \in \{1, \cdots, K\}$. Thus $\mathsf{DAff}\big(\bigcup_{j=1}^{K} C_j\big) \subseteq \bigcup_{j=1}^{K} \mathsf{DAff}(C_j)$, which implies $\mathsf{DAff}\big(\bigcup_{j=1}^{K} C_j\big) \subseteq \bigcup_{j=1}^{K} C_{j1} \times C_{j2}$. This completes the proof.

□

C.5 No Extrapolation beyond Discrete Affine Hull

In this section, we rely on the characterization of discrete affine hulls shown in the previous section in Theorem 6. Suppose we learn an additive energy model to estimate $\hat{p}(x|z)$ and estimate the density $p(x|z)$ for all training groups using maximum likelihood. In this case, we know that $\hat{p}(x|z) = p(x|z)$ for all $z \in \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$. In the next theorem, we show that such densities that satisfy $\hat{p}(x|z) = p(x|z)$ for all $z \in \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}})$ may not match the true density outside the affine hull. In the next result, we assume that $\forall z \in \mathcal{Z}^{\times}, p(\cdot|z)$ is not uniform. As a corollary, CRM does not extrapolate to points outside the affine hull as well.

Theorem 7. *Suppose we learn an additive energy model to estimate $\hat{p}(x|z)$ and estimate the density $p(x|z)$ for all training groups. There exist densities that maximize likelihood and exactly match the training distributions but do not extrapolate to distributions outside the affine hull of $\mathcal{Z}^{\mathsf{train}}$, i.e., $\exists z \in \mathcal{Z}^{\times}$, where $\hat{p}(\cdot|z) \neq p(\cdot|z)$.*

Proof. We first take $\mathcal{Z}^{\mathsf{train}}$ and partition the groups into K maximally connected components denoted $\{C_1, \cdots, C_K\}$. From Theorem 6, we know that the affine hull of $\mathcal{Z}^{\mathsf{train}}$ is the union of subgrids $\{S_1, \cdots, S_K\}$, where each subgrid S_j is the Cartesian product $C_{j1} \times C_{j2}$.

Let us consider all points $(\tilde{z}_1, \tilde{z}_2)$ in some subgrid S_k. For each such $(\tilde{z}_1, \tilde{z}_2) \in S_k$, define $\hat{E}_1(x, \tilde{z}_1) = E_1(x, \tilde{z}_1) + \alpha_k(x)$, $\hat{E}_2(x, \tilde{z}_2) = E(x, \tilde{z}_2) - \alpha_k(x)$. Note that regardless of choice of α_k the density, $\hat{p}(x|z) = \frac{1}{\mathbb{Z}(z)} e^{-\langle \sigma(z), \hat{E}(x) \rangle}$ matches the true density $p(x|z)$ for all groups z in $\bigcup_{i=1}^{K} S_i$.

Select any group $z_{\mathsf{ref}} = (z_1, z_2)$ that is not in the union of subgrids. From the definition of $\mathcal{Z}^{\times}$, it follows that there are points of the form (z_1, z_2') in one of the subgrid S_j and points of the form (z_1', z_2) are in some subgrid S_r. Let $\alpha_j(x) = -\frac{E_1(x,z_1)+E_2(x,z_2)}{2}$ and $\alpha_r(x) = \frac{E_1(x,z_1)+E_2(x,z_2)}{2}$. Observe that $\hat{E}_1(x, z_1) + \hat{E}_2(x, z_2) = E_1(x, z_1) + E_2(x, z_2) + \alpha_j(x) - \alpha_r(x) = 0$. Thus this choice of $\alpha_j(x) - \alpha_r(x)$ ensures that $\hat{p}(x|z_1, z_2)$ is uniform and hence cannot match the true $p(x|z_1, z_2)$.

This completes the proof.

□

Based on the above proof, we now argue that there exist solutions to CRM that do not extrapolate outside the affine hull. Let us consider solutions to CRM denoted $\hat{E}, \hat{B}$, which satisfies the property that $\langle \sigma(z), \hat{E}(x) \rangle = \langle \sigma(z), E(x) \rangle, \hat{B}(z) = B(z) \forall z \in \mathcal{Z}^{\mathsf{train}}$. Following the proof above, we can choose $\hat{E}'s$ in such a way that the sum of energies at a certain reference point outside the affine hull is zero and at all points inside the affine hull the sum of energies achieve a perfect match. For the group $z_{\mathsf{ref}} = (z_1, z_2)$ not in the affine hull of $\mathcal{Z}^{\mathsf{train}}$, we set $\hat{E}_1(x, z_1) + \hat{E}_2(x, z_2) = \langle \sigma(z), \hat{E}(x) \rangle = 0$.

Suppose $\hat{q}(z|x) = q(z|x), \forall z \in \mathsf{DAff}(\mathcal{Z}^{\mathsf{train}}) \bigcup \{z_{\mathsf{ref}}\}$. We now compute the likelihood ratio at z_{ref} and a point $z \in \mathcal{Z}^{\mathsf{train}}$. We obtain

$$\begin{aligned}
&\frac{\hat{q}(z_{\mathsf{ref}}|x)}{\hat{q}(z|x)} = \frac{q(z_{\mathsf{ref}}|x)}{q(z|x)} \implies \\
&-\log\Big(\frac{\hat{q}(z_{\mathsf{ref}}|x)}{\hat{q}(z|x)}\Big) = -\log\Big(\frac{q(z_{\mathsf{ref}}|x)}{q(z|x)}\Big) \implies \\
&\langle \sigma(z_{\mathsf{ref}}), \hat{E}(x) \rangle - \langle \sigma(z), \hat{E}(x) \rangle = \langle \sigma(z_{\mathsf{ref}}), E(x) \rangle - \langle \sigma(z), E(x) \rangle - (\theta(z) - \theta(z_{\mathsf{ref}}))
\end{aligned} \tag{39}$$

where $\theta(z)$ corresponds to collection of all terms that only depend on z. We already know that $\langle \sigma(z), \hat{E}(x) \rangle = \langle \sigma(z), E(x) \rangle$ and $\langle \sigma(z_{\mathsf{ref}}), \hat{E}(x) \rangle = 0$. Substituting these into the above expression we obtain

$$\langle \sigma(z_{\mathsf{ref}}), E(x) \rangle = \theta(z) - \theta(z_{\mathsf{ref}}) \tag{40}$$

From the above condition, it follows that $q(x|z_{\mathsf{ref}})$ is uniform. This implies that $p(x|z_{\mathsf{ref}})$ is also uniform, which contradicts the condition that $p(x|z_{\mathsf{ref}})$ is not uniform. Therefore, $\hat{q}(z|x) = q(z|x), \forall z \in \mathcal{Z}^{\mathsf{train}} \bigcup \{z_{\mathsf{ref}}\}$ cannot be true.

D Experiments Setup

D.1 Dataset Details

Waterbirds [Wah et al., 2011]. The task is to classify land birds ($y = 0$) from water birds ($y = 1$), where the spurious attributes are land background ($a = 0$) and water background ($a = 1$). Hence, we have a total of 4 groups $z = (y, a)$ in the dataset.

CelebA [Liu et al., 2015]. The task is to classify blond hair ($y = 1$) from non-blond hair ($y = 0$), where the spurious attribute is gender, female ($a = 0$) and male ($a = 1$). Hence, we have a total of 4 groups $z = (y, a)$ in the dataset.

MetaShift [Liang and Zou, 2022]. The task is to classify cats ($y = 0$) from dogs ($y = 1$), where the spurious attribute is background, indoor ($a = 0$) and outdoor ($a = 1$). Hence, we have a total of 4 groups $z = (y, a)$ in the dataset.

MultiNLI [Williams et al., 2017]. The task is to classify the relationship between the premise and hypothesis in a text document into one of the 3 classes: netural ($y = 0$), contradiction ($y = 1$), and entailment ($y = 2$). The spurious attribute are words like negation (binary attribute a), which are correlated with the contradiction class. Hence, we have a total of 6 groups $z = (y, a)$ in the dataset.

CivilComments [Borkan et al., 2019]. The task is to classify whether a text document contains toxic language ($y = 0$) versus it doesn't contain toxic language ($y = 1$), where the spurious attribute a corresponds to 8 different demographic identities (Male, Female, LGBTQ, Christian, Muslim, Other Religions, Black, and White). Hence, we have a total of 16 groups $z = (y, a)$ in the dataset.

NICO++ [Zhang et al., 2023]. This is a a large scale (60 classes with 6 spurious attributes) domain generalization benchmark, and we follow the procedure in Yang et al. [2023b] where all the groups with less than 75 samples were dropped from training. This leaves us with 337 groups during training, however, the validation set still has samples from all the 360 groups. Hence, we additionally discard these groups from the validation set as well to design the compositional shift version.

Dataset	Total Classes	Total Groups	Train Size	Val Size	Test Size
Waterbirds	2	4	4795	1199	5794
CelebA	2	4	162770	19867	19962
MetaShift	2	4	2276	349	874
MultiNLI	3	6	206175	82462	123712
CivilComments	2	16	148304	24278	71854
NICO++	60	360	62657	8726	17483

Table 3: Statitics for the different benchmarks used in our experiments.

D.2 Metric Details

Given the test distributions $z = (y, a) \sim q(z)$ and $x \sim q(x|z)$, lets denote the corresponding class predictions as $\hat{y} = \hat{M}(x)$ as per the method $\hat{M}$. Then average accuracy is defined as follows:

$$\text{Average Acc} := \mathbb{E}_{(y,a)}\mathbb{E}_{x\sim q(x|z)}\left[\mathbb{1}[y == \hat{M}(x)]\right]$$

Hence, this denotes the mean accuracy with groups drawn as per the test distribution $q(z)$. However, if certain (majority) groups have a higher probability of being sampled than others (minority groups) as per the distribution $q(z|x)$, then the average accuracy metric is more sensitive to mis-classifications in majority groups as compared to the minority groups. Hence, a method can achieve high average accuracy even though its accuracy for the minority groups might be low.

Therefore, we use the worst group accuracy metric, defined as follows.

$$\text{Worst Group Acc} := min_{(y,a)\in\mathcal{Z}^{\text{test}}}\mathbb{E}_{x\sim q(x|z)}\left[\mathbb{1}[y == \hat{M}(x)]\right]$$

Essentially we compute the accuracy for each group $(y, a) \sim q(z)$ as $\mathbb{E}_{x\sim q(x|z)}\left[\mathbb{1}[y == \hat{M}(x)]|\right]$ and then report the worst performance over all the groups. This metrics has been widely used for evaluating methods for subpopulation shifts [Sagawa et al., 2019, Yang et al., 2023b].

Similarly, we define the group balanced accuracy [Tsirigotis et al., 2024] as follows, where we compute the average of all per-group accuracy $\mathbb{E}_{x\sim q(x|z)}\left[\mathbb{1}[y == \hat{M}(x)]\right]$.

$$\text{Group Balanced Acc} := \frac{1}{|\mathcal{Z}^{\text{test}}|} \sum_{(y,a)\in\mathcal{Z}^{\text{test}}} \mathbb{E}_{x\sim q(x|z)}\left[\mathbb{1}[y == \hat{M}(x)]\right]$$

D.3 Method Details

For all the methods we have a *pre-trained* representation network backbone with linear classifier heads. We use ResNet-50 [He et al., 2016] for the vision datasets (Waterbirds, CelebA, MetaShift, NICO++) and BERT [Devlin et al., 2018] for the text datasets (MultiNLI, CivilComments). The parameters of both the representation network and linear classifier are updated with the same learning rate, and do not employ any special fine-tuning strategy for the representation network. For vision datasets we use the SGD optimizer (default values for momemtum 0.9), while for the text datasets we use the AdamW optimizer [Paszke et al., 2017] (default values for beta (0.9, 0.999)).

Hyperparameter Selection. We rely on the group balanced accuracy on the validation set to determine the optimal hyperparameters. We specify the grids for each hyperparameter in Table 4, and train each method with 5 randomly drawn hyperparameters. The grid sizes for hyperparameter selection were designed following Pezeshki et al. [2023].

Dataset	Learning Rate	Weight Decay	Batch Size	Total Epochs
Waterbirds	$10^{\text{Uniform}(-5,-3)}$	$10^{\text{Uniform}(-6,-3)}$	$2^{\text{Uniform}(5,7)}$	5000
CelebA	$10^{\text{Uniform}(-5,-3)}$	$10^{\text{Uniform}(-6,-3)}$	$2^{\text{Uniform}(5,7)}$	10000
MetaShift	$10^{\text{Uniform}(-5,-3)}$	$10^{\text{Uniform}(-6,-3)}$	$2^{\text{Uniform}(5,7)}$	5000
MulitNLI	$10^{\text{Uniform}(-6,-4)}$	$10^{\text{Uniform}(-6,-3)}$	$2^{\text{Uniform}(4,6)}$	10000
CivilComments	$10^{\text{Uniform}(-6,-4)}$	$10^{\text{Uniform}(-6,-3)}$	$2^{\text{Uniform}(4,6)}$	10000
NICO++	$10^{\text{Uniform}(-5,-3)}$	$10^{\text{Uniform}(-6,-3)}$	$2^{\text{Uniform}(5,7)}$	10000

Table 4: Details about the grids for hyperparameter selection. The choices for grid sizes were taken from Pezeshki et al. [2023].

E Additional Results

E.1 Results for all the Compositional Shift Scenarios

Table 5, Table 6, Table 7, Table 8, and Table 9 present the results for the Waterbirds, CelebA, MetaShift, MultiNLI, and CivilComments benchmark respectively. Here we do not aggregate over the multiple compositional shift scenarios of a benchmark, and provide a more detailed analysis with results for each scenario. For each method, we further highlight the worst case scenario for it, i.e, the scenario with the lowest worst group accuracy amongst all the compositional shift scenarios. This helps us easily compare the performance of methods for the respective worst case compositional shift scenario, as opposed to the average over all scenarios in Table 1. An interesting finding is that CRM outperforms all the baselines in the respective worst case compositional shift scenarios.

Discarded Group (y, a)	Method	Average Acc	Balanced Acc	Worst Group Acc
$(0, 0)$	ERM	74.0 (0.0)	82.3 (0.3)	67.0 (0.0)
	G-DRO	77.3 (0.7)	83.0 (0.6)	59.7 (1.9)
	LC	85.7 (0.3)	88.7 (0.3)	82.0 (0.6)
	sLA	86.0 (0.0)	89.0 (0.0)	82.3 (0.3)
	CRM	86.7 (0.9)	88.7 (0.3)	83.0 (1.5)
$(0, 1)$	ERM	67.3 (0.3)	71.7 (0.3)	28.0 (1.2)
	G-DRO	58.3 (3.2)	70.7 (2.0)	11.7 (4.6)
	LC	82.7 (3.2)	86.0 (1.7)	72.0 (5.8)
	sLA	86.3 (1.7)	88.0 (1.0)	78.7 (3.3)
	CRM	86.0 (2.1)	86.7 (0.7)	73.0 (4.2)
$(1, 0)$	ERM	84.0 (0.0)	78.0 (0.0)	38.3 (0.3)
	G-DRO	90.0 (0.0)	86.0 (0.6)	67.0 (3.6)
	LC	93.0 (0.0)	89.0 (0.6)	79.0 (1.2)
	sLA	93.0 (0.0)	89.0 (0.6)	79.3 (1.5)
	CRM	86.7 (0.3)	89.0 (0.0)	83.7 (0.3)
$(1, 1)$	ERM	86.3 (0.3)	69.3 (0.3)	38.7 (0.7)
	G-DRO	86.0 (0.6)	75.7 (2.2)	31.0 (9.2)
	LC	92.0 (0.0)	84.0 (0.6)	69.0 (1.5)
	sLA	92.0 (0.0)	84.0 (0.6)	69.0 (1.5)
	CRM	89.0 (0.6)	86.7 (0.7)	75.0 (3.2)

Table 5: Results for the various compositional shift scenarios for the **Waterbirds** benchmark. For each metric, report the mean (standard error) over 3 random seeds on the test dataset. We highlight the worst case compositional shift scenario for each method, i.e, the scenario with the lowest worst group accuracy amongst all the compositional shift scenarios. CRM outperforms all the baselines in the respective worst case compositional shift scenarios.

Discarded Group (y, a)	Method	Average Acc	Balanced Acc	Worst Group Acc
$(0, 0)$	ERM	68.7 (0.3)	74.0 (0.0)	37.7 (0.3)
	G-DRO	85.0 (0.6)	88.0 (0.0)	75.0 (1.2)
	LC	88.0 (0.0)	90.3 (0.3)	82.3 (0.3)
	sLA	87.7 (0.3)	90.3 (0.3)	82.3 (0.7)
	CRM	91.7 (0.3)	89.3 (0.3)	81.0 (2.0)
$(0, 1)$	ERM	91.3 (0.9)	91.0 (0.6)	86.7 (1.3)
	G-DRO	85.0 (1.5)	88.7 (0.7)	72.7 (3.7)
	LC	93.0 (0.6)	87.7 (0.9)	71.0 (1.7)
	sLA	92.7 (0.3)	88.0 (0.0)	71.3 (0.9)
	CRM	88.3 (0.9)	91.0 (0.6)	85.0 (2.0)
$(1, 0)$	ERM	87.0 (0.0)	59.3 (0.3)	4.0 (0.0)
	G-DRO	91.7 (0.3)	86.3 (0.7)	71.7 (0.9)
	LC	88.3 (0.3)	70.7 (0.7)	21.0 (2.1)
	sLA	88.3 (0.3)	71.0 (0.6)	21.3 (1.9)
	CRM	93.0 (0.0)	85.7 (0.3)	73.3 (1.8)
$(1, 1)$	ERM	96.0 (0.0)	78.0 (0.6)	27.7 (2.0)
	G-DRO	95.0 (0.0)	84.3 (0.3)	51.7 (1.2)
	LC	95.0 (0.0)	85.3 (0.3)	55.3 (1.9)
	sLA	95.0 (0.0)	85.0 (0.6)	54.7 (2.3)
	CRM	91.3 (0.3)	91.0 (0.0)	88.0 (0.6)

Table 6: Results for the various compositional shift scenarios for the **CelebA** benchmark. For each metric, report the mean (standard error) over 3 random seeds on the test dataset. We highlight the worst case compositional shift scenario for each method, i.e, the scenario with the lowest worst group accuracy amongst all the compositional shift scenarios. CRM outperforms all the baselines in the respective worst case compositional shift scenarios.

Discarded Group (y, a)	Method	Average Acc	Balanced Acc	Worst Group Acc
$(0, 0)$	ERM	84.3 (0.3)	84.0 (0.6)	80.3 (0.9)
	G-DRO	84.0 (0.6)	83.3 (0.7)	78.0 (0.6)
	LC	89.0 (0.0)	85.7 (0.3)	74.3 (1.8)
	sLA	90.0 (0.0)	85.0 (0.0)	67.3 (1.9)
	CRM	87.3 (0.3)	84.3 (0.3)	73.3 (0.7)
$(0, 1)$	ERM	85.0 (0.0)	79.0 (0.0)	49.0 (0.0)
	G-DRO	86.0 (1.0)	81.7 (0.3)	55.3 (3.2)
	LC	86.0 (0.0)	84.0 (0.0)	63.7 (0.3)
	sLA	86.0 (0.0)	84.0 (0.0)	64.0 (0.6)
	CRM	88.3 (0.3)	85.7 (0.3)	78.0 (1.0)
$(1, 0)$	ERM	90.0 (0.0)	82.0 (0.0)	48.3 (0.3)
	G-DRO	90.3 (0.3)	82.7 (0.9)	52.7 (2.3)
	LC	90.0 (0.0)	84.3 (0.3)	62.0 (0.0)
	sLA	88.7 (0.3)	81.0 (0.0)	46.7 (0.7)
	CRM	87.0 (1.2)	83.3 (0.7)	70.0 (1.0)
$(1, 1)$	ERM	83.3 (1.2)	81.7 (0.9)	64.3 (1.2)
	G-DRO	83.7 (0.9)	82.7 (0.9)	69.3 (2.0)
	LC	89.0 (0.0)	86.0 (0.0)	72.7 (0.7)
	sLA	89.0 (0.0)	86.0 (0.0)	74.0 (0.0)
	CRM	87.7 (0.3)	85.3 (0.3)	72.3 (1.7)

Table 7: Results for the various compositional shift scenarios for the **MetaShift** benchmark. For each metric, report the mean (standard error) over 3 random seeds on the test dataset. We highlight the worst case compositional shift scenario for each method, i.e, the scenario with the lowest worst group accuracy amongst all the compositional shift scenarios. CRM outperforms all the baselines in the respective worst case compositional shift scenarios.

Discarded Group (y, a)	Method	Average Acc	Balanced Acc	Worst Group Acc
$(0, 0)$	ERM	62.7 (0.3)	66.7 (0.3)	0.7 (0.3)
	G-DRO	63.3 (0.3)	68.0 (0.0)	1.7 (0.7)
	LC	68.0 (0.0)	72.0 (0.0)	20.0 (0.0)
	sLA	67.7 (0.3)	72.0 (0.0)	19.7 (1.5)
	CRM	64.7 (0.9)	70.7 (0.9)	31.0 (5.6)
$(0, 1)$	ERM	77.7 (0.3)	71.7 (0.3)	14.0 (1.0)
	G-DRO	80.7 (0.7)	80.7 (0.7)	74.0 (1.0)
	LC	81.0 (0.0)	81.0 (0.0)	75.3 (0.3)
	sLA	81.3 (0.3)	80.7 (0.3)	69.0 (0.6)
	CRM	80.0 (0.6)	78.0 (1.2)	62.3 (8.2)
$(1, 0)$	ERM	58.0 (0.0)	67.0 (0.0)	0.0 (0.0)
	G-DRO	57.7 (0.3)	67.7 (0.3)	0.0 (0.0)
	LC	70.7 (0.9)	74.3 (0.3)	47.3 (4.3)
	sLA	73.3 (2.7)	76.3 (1.7)	58.3 (9.7)
	CRM	69.5 (0.5)	74.0 (0.0)	63.5 (0.5)
$(1, 1)$	ERM	82.0 (0.2)	73.0 (0.2)	20.0 (1.2)
	G-DRO	80.3 (0.3)	79.3 (0.3)	72.7 (0.9)
	LC	81.7 (0.3)	81.3 (0.3)	74.3 (1.5)
	sLA	82.0 (0.0)	81.0 (0.0)	75.3 (0.7)
	CRM	81.3 (0.3)	80.7 (0.3)	71.3 (1.8)
$(2, 0)$	ERM	62.0 (0.0)	68.3 (0.3)	0.0 (0.0)
	G-DRO	60.0 (0.0)	67.7 (0.3)	0.0 (0.0)
	LC	72.3 (0.3)	74.7 (0.3)	48.7 (0.7)
	sLA	72.7 (0.7)	74.3 (0.3)	48.3 (0.9)
	CRM	68.7 (0.3)	72.7 (0.3)	50.0 (0.6)
$(2, 1)$	ERM	81.3 (0.3)	74.3 (0.3)	17.3 (2.4)
	G-DRO	80.7 (0.3)	79.0 (0.0)	57.3 (2.2)
	LC	82.0 (0.0)	80.7 (0.3)	60.0 (1.2)
	sLA	81.7 (0.3)	80.3 (0.3)	59.3 (0.9)
	CRM	81.3 (0.3)	80.0 (0.6)	72.7 (0.9)

Table 8: Results for the various compositional shift scenarios for the **MultiNLI** benchmark. For each metric, report the mean (standard error) over 3 random seeds on the test dataset. We highlight the worst case compositional shift scenario for each method, i.e, the scenario with the lowest worst group accuracy amongst all the compositional shift scenarios. CRM outperforms all the baselines in the respective worst case compositional shift scenarios.

Discarded Group (y, a)	Method	Average Acc	Balanced Acc	Worst Group Acc
$(0, 0)$	ERM	79.0 (0.6)	78.7 (0.3)	61.3 (1.5)
	G-DRO	79.3 (1.2)	79.0 (0.0)	64.7 (3.0)
	LC	79.7 (0.3)	79.0 (0.0)	64.3 (0.9)
	sLA	79.7 (0.3)	79.3 (0.3)	66.7 (1.8)
	CRM	84.0 (0.0)	78.7 (0.3)	67.0 (2.5)
$(0, 1)$	ERM	78.0 (0.6)	78.3 (0.3)	64.3 (1.2)
	G-DRO	78.0 (0.6)	78.7 (0.3)	64.3 (1.5)
	LC	79.3 (0.3)	79.0 (0.0)	64.3 (0.9)
	sLA	79.7 (0.3)	79.0 (0.0)	65.3 (0.3)
	CRM	83.3 (0.7)	78.7 (0.3)	71.0 (1.5)
$(0, 2)$	ERM	78.3 (0.3)	77.7 (0.3)	38.0 (1.0)
	G-DRO	79.0 (0.6)	78.3 (0.3)	43.7 (0.3)
	LC	79.0 (0.6)	79.0 (0.0)	53.7 (2.3)
	sLA	79.3 (0.3)	79.0 (0.0)	55.0 (2.1)
	CRM	83.3 (0.3)	78.7 (0.3)	68.0 (1.0)
$(0, 3)$	ERM	80.5 (0.5)	79.0 (0.0)	66.0 (2.0)
	G-DRO	80.0 (0.6)	79.0 (0.0)	67.3 (2.7)
	LC	81.3 (0.3)	79.0 (0.0)	69.0 (1.2)
	sLA	80.7 (0.7)	79.0 (0.0)	66.7 (2.7)
	CRM	83.7 (0.3)	78.7 (0.3)	69.7 (0.3)
$(0, 4)$	ERM	78.0 (0.0)	77.7 (0.3)	38.0 (0.6)
	G-DRO	78.7 (0.9)	78.7 (0.3)	52.0 (3.2)
	LC	79.0 (0.0)	79.0 (0.0)	60.7 (1.5)
	sLA	78.3 (0.3)	79.0 (0.0)	62.0 (1.0)
	CRM	83.7 (0.3)	79.0 (0.0)	69.7 (1.9)
$(0, 5)$	ERM	80.0 (0.0)	79.0 (0.0)	61.0 (0.6)
	G-DRO	80.0 (0.6)	79.0 (0.0)	67.3 (1.8)
	LC	79.3 (0.9)	79.0 (0.0)	65.7 (2.3)
	sLA	80.0 (0.0)	79.7 (0.3)	66.7 (0.3)
	CRM	84.0 (0.0)	78.7 (0.3)	71.0 (1.0)
$(0, 6)$	ERM	78.7 (0.3)	78.0 (0.0)	36.3 (1.2)
	G-DRO	78.3 (0.3)	78.3 (0.3)	46.3 (1.2)
	LC	80.7 (0.3)	79.0 (0.0)	58.7 (2.3)
	sLA	79.7 (0.9)	79.0 (0.0)	57.0 (3.1)
	CRM	83.3 (0.7)	78.7 (0.3)	70.0 (1.0)
$(0, 7)$	ERM	79.0 (0.0)	77.7 (0.3)	40.0 (1.2)
	G-DRO	77.7 (0.3)	78.7 (0.3)	49.7 (0.3)
	LC	79.7 (0.3)	79.0 (0.0)	60.0 (2.3)
	sLA	78.7 (0.3)	79.0 (0.0)	56.3 (1.3)
	CRM	83.3 (0.3)	78.3 (0.3)	64.0 (1.2)
$(1, 0)$	ERM	81.3 (0.3)	79.0 (0.0)	60.3 (0.3)
	G-DRO	82.3 (0.7)	79.0 (0.0)	69.7 (1.3)
	LC	81.3 (0.3)	79.0 (0.0)	71.0 (0.6)
	sLA	81.3 (0.9)	79.0 (0.0)	70.0 (1.2)
	CRM	84.0 (0.0)	78.0 (0.0)	68.3 (0.9)
$(1, 1)$	ERM	81.7 (0.3)	77.7 (0.3)	60.3 (1.2)
	G-DRO	82.0 (0.6)	79.0 (0.0)	67.3 (0.9)
	LC	80.7 (0.3)	79.0 (0.0)	69.3 (0.9)
	sLA	81.3 (0.3)	79.0 (0.0)	71.0 (1.2)
	CRM	84.0 (0.0)	78.3 (0.3)	70.0 (0.6)
$(1, 2)$	ERM	81.3 (0.3)	78.7 (0.3)	61.3 (0.7)
	G-DRO	80.7 (0.3)	79.0 (0.0)	63.7 (2.4)
	LC	82.0 (0.6)	79.0 (0.0)	70.0 (2.1)
	sLA	82.0 (0.6)	79.0 (0.0)	69.7 (1.8)
	CRM	83.7 (0.3)	78.3 (0.3)	63.7 (3.2)
$(1, 3)$	ERM	82.3 (0.9)	78.0 (0.0)	59.0 (1.5)
	G-DRO	81.0 (0.6)	79.0 (0.0)	67.3 (2.6)
	LC	82.0 (0.0)	79.0 (0.0)	70.0 (1.5)
	sLA	82.7 (0.9)	79.3 (0.3)	69.0 (1.5)
	CRM	83.7 (0.3)	78.0 (0.0)	71.0 (1.5)
$(1, 4)$	ERM	82.3 (0.3)	78.7 (0.3)	58.3 (1.8)
	G-DRO	80.3 (0.3)	79.0 (0.0)	68.0 (0.6)
	LC	82.0 (0.0)	79.3 (0.3)	70.7 (0.3)
	sLA	82.0 (0.6)	79.3 (0.3)	70.0 (0.6)
	CRM	83.7 (0.3)	78.3 (0.3)	60.0 (1.5)
$(1, 5)$	ERM	82.0 (0.0)	78.7 (0.3)	63.7 (0.3)
	G-DRO	81.7 (0.3)	79.0 (0.0)	64.7 (1.3)
	LC	81.3 (0.3)	79.3 (0.3)	68.3 (0.7)
	sLA	82.0 (0.6)	79.0 (0.0)	71.3 (0.9)
	CRM	83.7 (0.3)	78.3 (0.3)	70.0 (1.0)
$(1, 6)$	ERM	82.0 (0.6)	79.0 (0.0)	65.3 (2.4)
	G-DRO	81.0 (0.0)	79.3 (0.3)	66.0 (1.2)
	LC	81.7 (0.7)	79.0 (0.0)	69.7 (2.3)
	sLA	80.7 (0.3)	79.0 (0.0)	66.7 (0.3)
	CRM	84.0 (0.0)	78.3 (0.3)	70.0 (1.5)
$(1, 7)$	ERM	82.0 (1.2)	78.7 (0.3)	63.3 (1.8)
	G-DRO	81.0 (0.0)	79.0 (0.0)	64.3 (0.3)
	LC	81.7 (0.3)	79.0 (0.0)	66.0 (1.5)
	sLA	82.3 (0.3)	79.0 (0.0)	67.0 (1.5)
	CRM	84.0 (0.3)	77.0 (0.0)	65.0 (1.0)

Table 9: Results for the various compositional shift scenarios for the **CivilComments** benchmark.

E.2 Results for Ablations with CRM

In the implementation of CRM in Algorithm 1, we have the following two choices; 1) we use the extrapolated bias $B^\star$ (equation 7); 2) we set $\hat{q}(z)$ as the uniform distribution, i.e, $\hat{q}(z = (y, a)) = \frac{1}{d_y \times d_a}$. We now conduct ablation studies by varying these components as follows.

- *Bias $B^\star$ + Emp Prior:* We still use the extrapolated bias $B^\star$ but instead of uniform $\hat{q}(z)$, we use test dataset to obtain the counts of each group, denoted as the empirical prior. Note that this approach assumes the knowledge of test distribution of groups, hence we expect this to improve the average accuracy but not the necessarily the worst group accuracy.
- *Bias $\hat{B}$ + Unf Prior:* We still use the uniform prior for $\hat{q}(z)$ but instead of the extrapolated bias $B^\star$, we use the learned bias $\hat{B}$ (equation 5). This ablation helps us to understand whether extrapolated bias $B^\star$ are crucial for CRM to generalize to compositional shifts.
- *Bias $\hat{B}$+ Emp Prior:* Here we change both aspects of CRM as we use the learned bias $\hat{B}$ and empirical prior from the test dataset for $\hat{q}(z)$.

Table 10 presents the results of the ablation study. We find that extrapolated bias is crucial for CRM as the worst group accuracy with learned bias is much worse! Further, using empirical prior instead of the uniform prior leads to improvement in average accuracy at the cost of worst group accuracy.

Dataset	Ablation	Average Acc	Balanced Acc	Worst Group Acc
Waterbirds	CRM	87.1 (0.7)	87.8 (0.1)	78.7 (1.6)
	Bias $B^\star$ + Emp Prior	91.6 (0.2)	87.4 (0.3)	75.2 (1.3)
	Bias $\hat{B}$ + Unf Prior	81.2 (0.6)	82.7 (0.2)	55.7 (1.0)
	Bias $\hat{B}$ + Emp Prior	84.3 (0.6)	81.6 (0.3)	51.3 (1.0)
CelebA	CRM	91.1 (0.2)	89.2 (0.3)	81.8 (1.2)
	Bias $B^\star$ + Emp Prior	94.3 (0.1)	75.8 (0.4)	34.1 (1.0)
	Bias $\hat{B}$ + Unf Prior	83.6 (0.1)	84.7 (0.2)	58.9 (0.4)
	Bias $\hat{B}$ + Emp Prior	90.9 (0.1)	77.2 (0.3)	35.4 (0.7)
MetaShift	CRM	87.6 (0.2)	84.7 (0.1)	73.4 (0.7)
	Bias $B^\star$ + Emp Prior	89.2 (0.2)	84.0 (0.4)	65.1 (1.4)
	Bias $\hat{B}$ + Unf Prior	87.2 (0.3)	82.9 (0.4)	58.7 (0.6)
	Bias $\hat{B}$ + Emp Prior	88.1 (0.1)	82.1 (0.1)	56.1 (0.4)
MultiNLI	CRM	74.6 (0.5)	76.1 (0.4)	57.7 (3.0)
	Bias $B^\star$ + Emp Prior	75.0 (0.5)	72.2 (0.4)	39.7 (3.2)
	Bias $\hat{B}$ + Unf Prior	72.9 (0.9)	74.0 (0.4)	28.9 (2.1)
	Bias $\hat{B}$ + Emp Prior	73.6 (0.9)	70.8 (0.4)	20.2 (0.2)
CivilComments	CRM	83.7 (0.1)	78.4 (0.1)	68.1 (0.5)
	Bias $B^\star$ + Emp Prior	87.0 (0.0)	74.1 (0.3)	48.0 (1.2)
	Bias $\hat{B}$ + Unf Prior	76.8 (0.2)	77.8 (0.0)	51.9 (1.0)
	Bias $\hat{B}$ + Emp Prior	83.5 (0.1)	78.0 (0.0)	62.2 (0.6)
NICO++	CRM	84.7 (0.3)	84.7 (0.3)	40.3 (4.3)
	Bias $B^\star$ + Emp Prior	85.0 (0.0)	85.0 (0.0)	41.0 (4.9)
	Bias $\hat{B}$ + Unf Prior	85.0 (0.0)	85.0 (0.0)	31.0 (1.0)
	Bias $\hat{B}$ + Emp Prior	85.0 (0.0)	85.0 (0.0)	27.7 (3.9)

Table 10: **Ablation study with CRM.** We consider the average performance over the different compositional shift scenarios for each benchmark, and report the mean (standard error) over 3 random seeds on the test dataset. CRM corresponds to the usual implementation with extrapolated bias $B^\star$ and uniform prior for $\hat{q}(z)$. CRM obtains better worst group accuracy than all the ablations, highlighting the importance of both extrapolated bias and uniform prior! Extrapolated bias is critical for generalization to compositional shifts as the performance with learned bias is much worse.

E.3 Results for the Original Benchmarks

We present results for the original benchmarks $(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}, \mathcal{D}_{\text{train}})$ in Table 11, which corresponds to the standard subpopulation shift case for these benchmarks. For Waterbirds, CelebA, MetaShift, and MultiNLI, subpopulation shift implies all the groups $z = (y, a)$ are present in both the train and test dataset ($\mathcal{Z}^{\text{train}} = \mathcal{Z}^{\text{test}} = \mathcal{Z}^{\times}$), however, the groups sizes change from train to test, inducing a spurious correaltion between class labels y and attributes a. For the NICO++ dataset, we have a total of 360 groups in the test dataset but only 337 of them are present in the train dataset. But still this is not a compositional shift as the validation dataset contains all the 360 groups. We find that CRM is still competitive to the baselines for the standard subpopulation shift scenario of each benchmark!

Dataset	Method	Average Acc	Balanced Acc	Worst Group Acc
Waterbirds	ERM	87.3 (0.3)	84.0 (0.0)	62.3 (1.2)
	G-DRO	91.7 (0.3)	91.0 (0.0)	87.3 (0.3)
	LC	92.0 (0.0)	91.0 (0.0)	88.7 (0.3)
	sLA	92.3 (0.3)	91.0 (0.0)	89.7 (0.3)
	CRM	91.3 (0.9)	91.0 (0.0)	86.0 (0.6)
CelebA	ERM	95.7 (0.3)	84.0 (0.0)	52.0 (1.0)
	G-DRO	92.0 (0.6)	93.0 (0.0)	91.0 (0.6)
	LC	92.0 (0.6)	92.0 (0.0)	90.0 (0.6)
	sLA	92.3 (0.3)	91.7 (0.3)	86.7 (1.9)
	CRM	93.0 (0.0)	92.0 (0.0)	89.0 (0.6)
MetaShift	ERM	90.0 (0.0)	84.0 (0.0)	63.0 (0.0)
	G-DRO	90.3 (0.3)	88.3 (0.3)	80.7 (1.3)
	LC	89.7 (0.3)	87.7 (0.3)	80.0 (1.2)
	sLA	90.0 (0.6)	87.7 (0.3)	80.0 (1.2)
	CRM	88.3 (0.7)	85.7 (0.3)	74.7 (1.5)
MultiNLI	ERM	81.7 (0.3)	80.7 (0.3)	68.0 (1.7)
	G-DRO	80.7 (0.3)	78.0 (0.0)	57.0 (2.3)
	LC	82.0 (0.0)	82.0 (0.0)	74.3 (1.2)
	sLA	82.0 (0.0)	82.0 (0.0)	71.7 (0.3)
	CRM	81.7 (0.3)	81.7 (0.3)	74.7 (1.3)
CivilComments	ERM	80.3 (0.3)	79.0 (0.0)	61.0 (2.5)
	G-DRO	79.7 (0.3)	79.0 (0.0)	64.7 (1.5)
	LC	80.7 (0.3)	79.7 (0.3)	67.3 (0.3)
	sLA	80.3 (0.3)	79.0 (0.0)	66.3 (0.9)
	CRM	83.3 (0.3)	78.0 (0.0)	70.0 (0.6)
NICO++	ERM	85.3 (0.3)	85.0 (0.0)	35.3 (2.3)
	G-DRO	83.7 (0.3)	83.3 (0.3)	33.7 (1.2)
	LC	85.0 (0.0)	85.0 (0.0)	35.3 (2.3)
	sLA	85.0 (0.0)	85.0 (0.0)	35.3 (2.3)
	CRM	85.0 (0.0)	84.7 (0.3)	39.0 (3.2)

Table 11: Results for the standard subpopulation shift case for each benchmark. Here we do not transform the datasets for compositional shifts, hence all the groups are present in both the train and the test dataset (except the NICO++ benchmark). CRM is still competitive with the baselines for this scenario where no groups were discarded additionally.