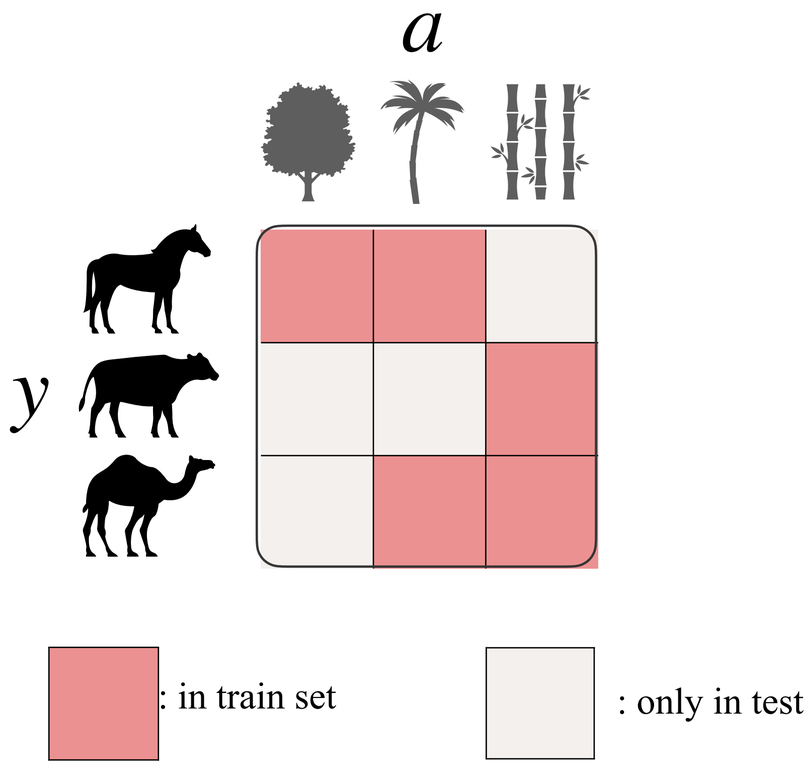
Compositional Risk Minimization

Divyat Mahajan[1,2⋆], **Mohammad Pezeshki**[1], **Charles Arnal**[1], **Ioannis Mitliagkas**[2], **Kartik Ahuja**[1,†], **Pascal Vincent**[1,2⋆,†]

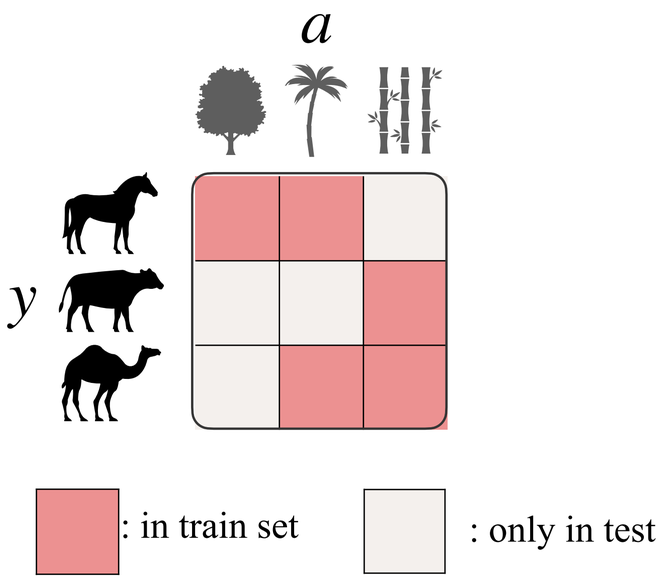
[1]Meta FAIR, [2]Mila, Université de Montréal, DIRO
[⋆]Work done at Meta, [†]Joint last author

Compositional Shifts



- Some combinations of attributes are totally absent from the training distribution but present in the test distribution

Compositional Shifts



Compositional Distribution Shifts

- Assumption 1: $p(x \mid z) = q(x \mid z)\ \forall z \in \mathcal{Z}^{\times}$
- Assumption 2: $\mathcal{Z}^{\text{test}} \not\subseteq \mathcal{Z}^{\text{train}}$ but $\mathcal{Z}^{\text{test}} \subseteq \mathcal{Z}^{\times}$

- Attribute Vector: $z = (z_1, \ldots, z_m)$ that characterizes the group for the input x
  - Each attribute z_i is categorical and can take d possible values.
- Train Distribution: $p(x, z) = p(z)p(x \mid z)$ with support of z as $\mathcal{Z}^{\text{train}}$
- Test Distribution: $q(x, z) = q(z)q(x \mid z)$ with support of z as $\mathcal{Z}^{\text{test}}$
- Cartesian Product: $\mathcal{Z}^{\times} = \mathcal{Z}_1^{\text{train}} \times \mathcal{Z}_2^{\text{train}} \times \cdots \mathcal{Z}_m^{\text{train}}$

Subpopulation Shifts

Subpopulation Shift: $p(x \mid z) = q(x \mid z)$ but $p(z) \neq q(z)$

Common Setup: **Imbalanced** distribution over group during **training** while **balanced** distribution over groups during **evaluation**

Compositional Shifts are an extreme version of Subpopulation shifts!

Contributions

Build classifiers that are robust to compositional distributions shifts!

Theory of 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*.

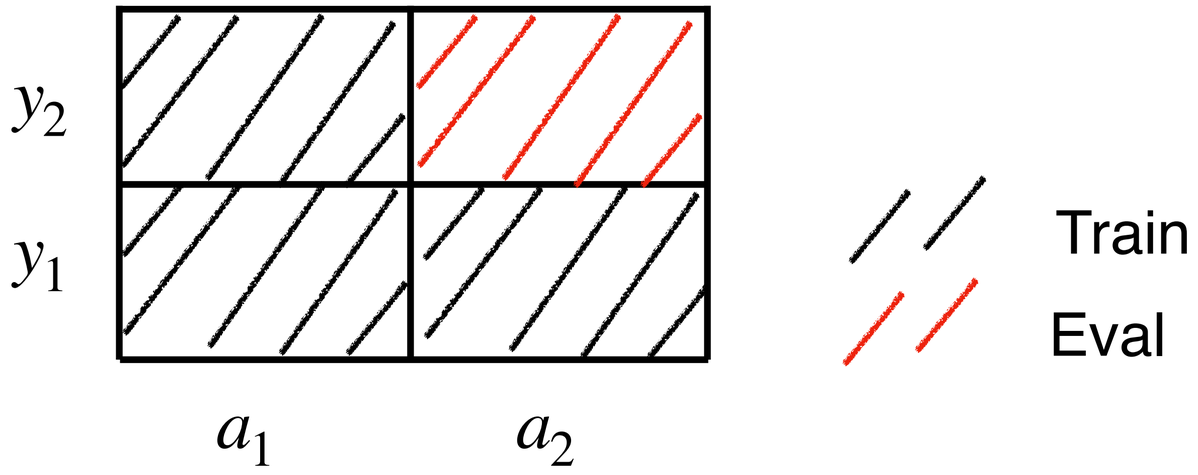
A Practical Method. We propose simple algorithm Compositional Risk Minimization (CRM), which first trains an additive energy classifier and then adjusts the trained classifier for tackling compositional shifts.

Generative Classification

$$p(z \mid x) = \frac{p(x \mid z)p(z)}{p(x)} \propto p(x \mid z)p(z)$$

If we can reliably estimate $p(x \mid y_2, a_2)$ then we can make predictions for the novel group at test time

Challenge: We never observe samples from group (y_2, a_2) during training

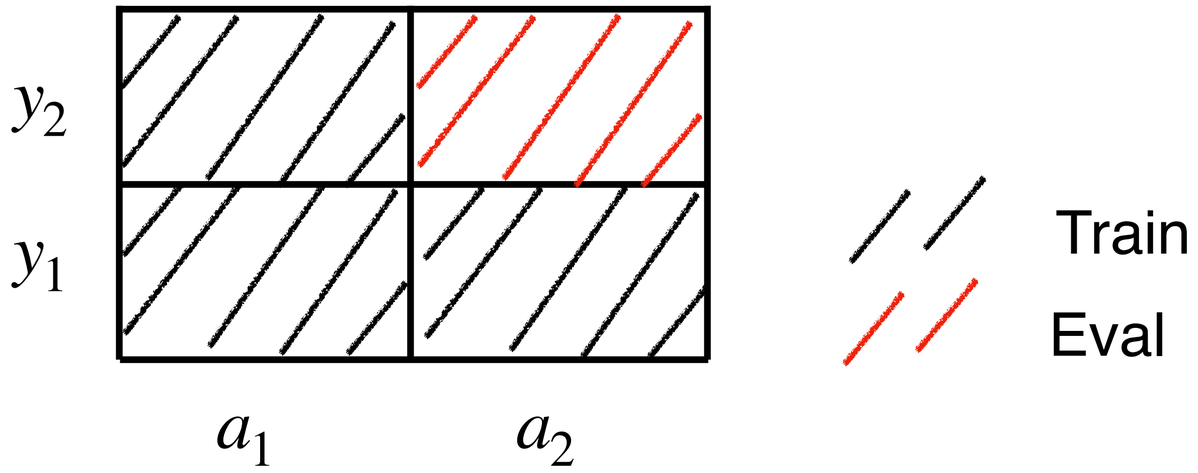


Cartesian Product Extrapolation (CPE)

Assume we have done density estimation perfectly for train groups,

$$p(x \mid y_1, a_1) = \hat{p}(x \mid y_1, a_1)$$
$$p(x \mid y_2, a_1) = \hat{p}(x \mid y_2, a_1)$$
$$p(x \mid y_1, a_2) = \hat{p}(x \mid y_1, a_2)$$

Does this imply $p(x \mid y_2, a_2) = \hat{p}(x \mid y_2, a_2)$?



Additive Decoders

Assume $p(x \mid y, a)$ as parameterized by an additive function
$p(x \mid y, a) = N(x; f(y, a), I)$ where $f(y, a) = f_y(y) + f_a(a)$

Then it can be proved that CPE is possible!

Additive Decoders for Latent Variables Identification and Cartesian-Product Extrapolation

Sébastien Lachapelle*, Divyat Mahajan*, Ioannis Mitliagkas & Simon Lacoste-Julien

Neural Information Processing Systems (*NeurIPS*) 2023 (*Oral*)

*Equal contribution

Additive Decoders

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z}) = \sum_{B \in \mathcal{B}} \boldsymbol{f}^{(B)}(\boldsymbol{z}_B)$$

Observation
e.g. an image

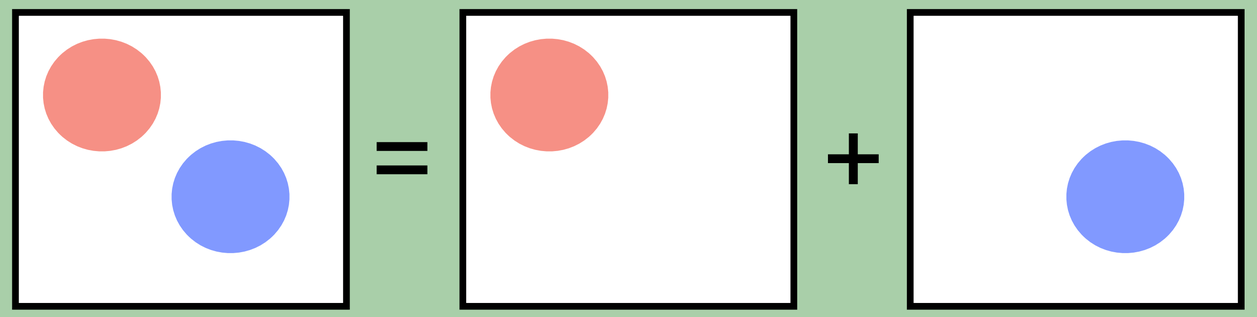
Latent
Factors

Partition of $\{1, \ldots, d_z\}$
e.g. $\mathcal{B} = \{\{1,2\}, \{3,4\}\}$

Sub-blocks of z

Example: Images of moving balls

$x = f(z)$ $\quad$ $f^{(B_1)}(z_1)$ $\quad$ $f^{(B_2)}(z_2)$



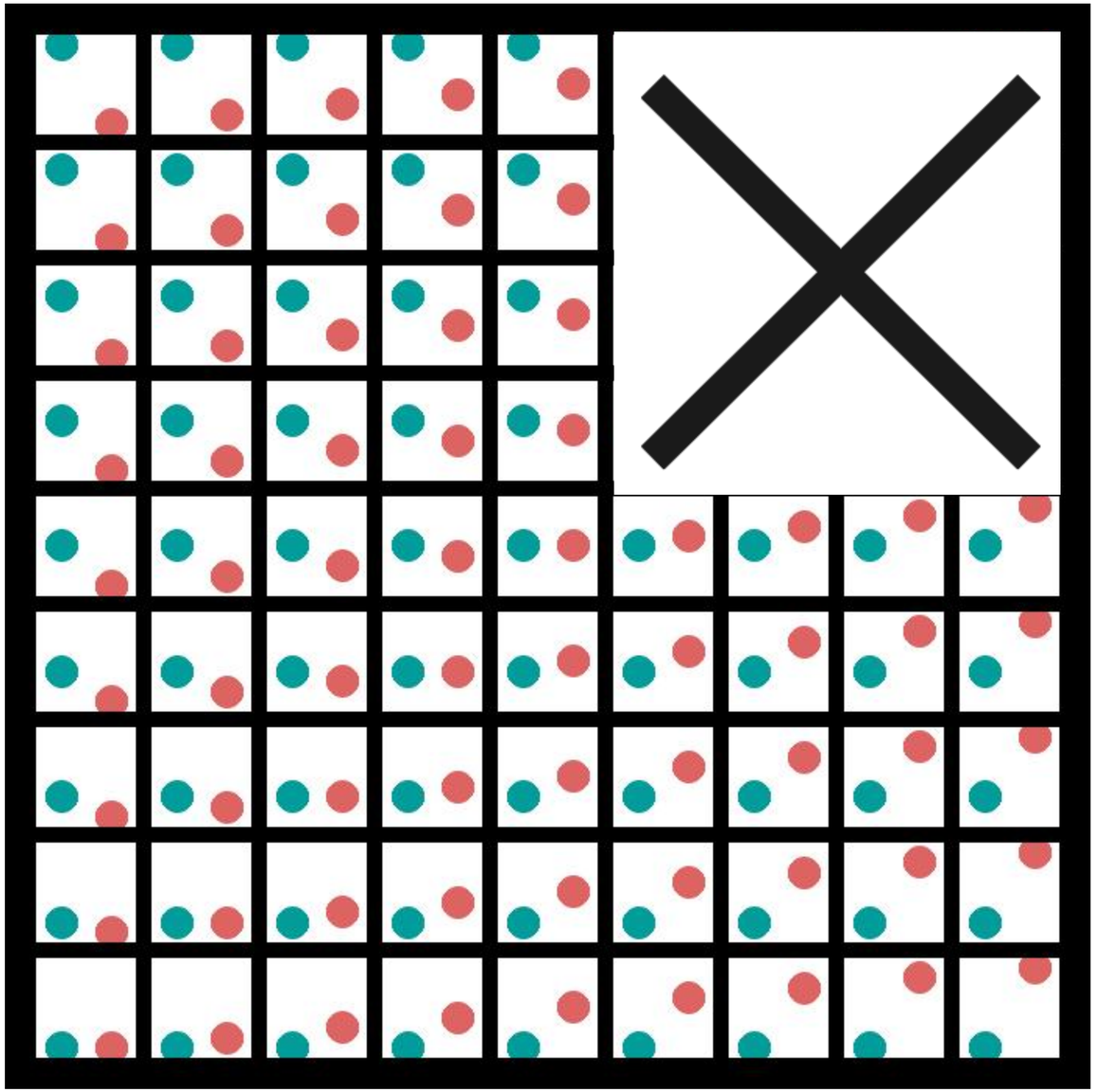
$\mathcal{B} = \{\{1,2\}, \{3,4\}\}$

$z_{B_1} = (z_1, z_2)$ Coordinates of

$z_{B_2} = (z_3, z_4)$ Coordinates of

$f^{(B)}$ Block-specific Decoder

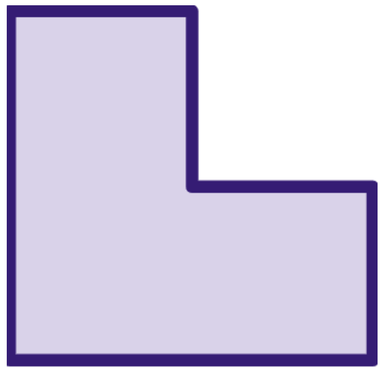
Extrapolation



Scalar Latent Dataset:

- Balls move only along y-axis
- Remove images where both balls have high y-coordinate to get L-shaped training support

$\hat{Z}_{train}$

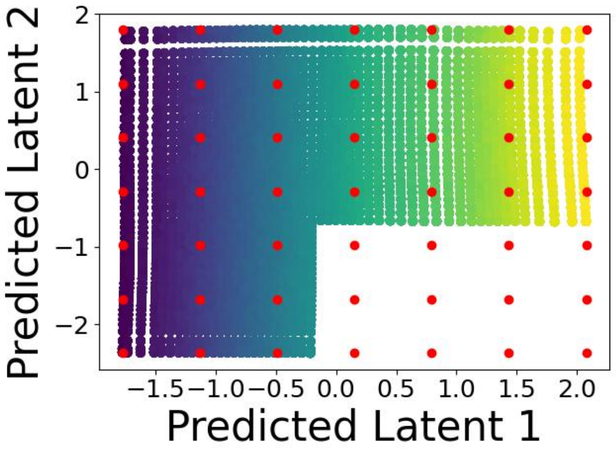


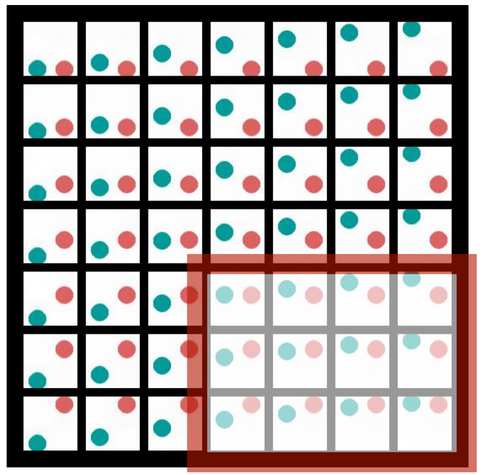
Extrapolation

Learned Latent Space

Generated Images

Additive Decoder

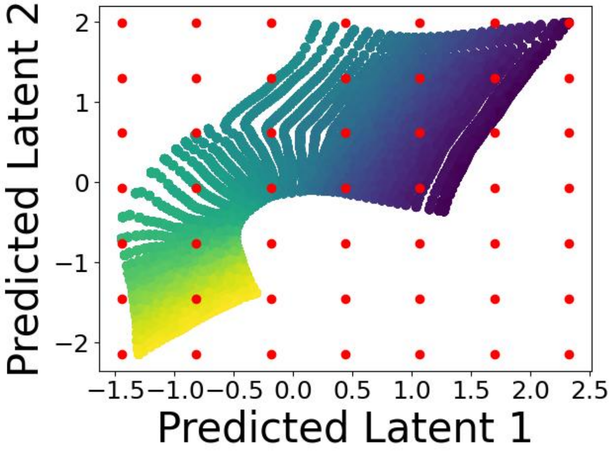


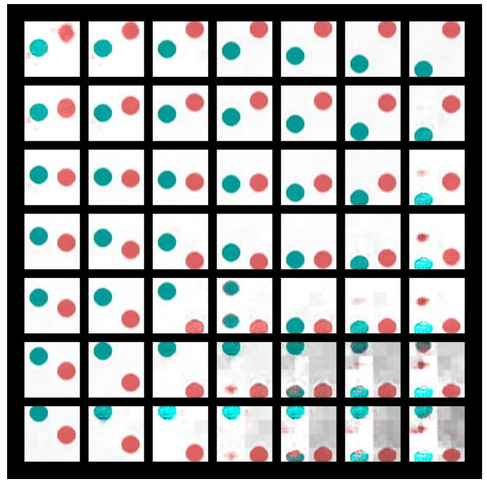


Disentangled

These samples were never seen during training

Non-Additive Decoder

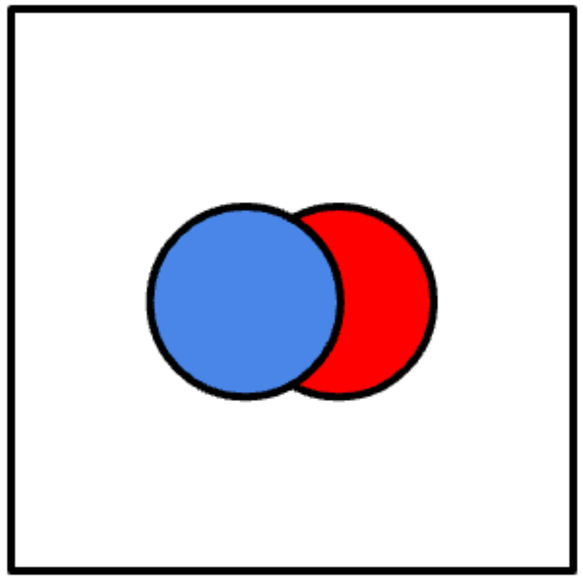




Entangled

Cannot generate unseen samples

Limitation: Additive Decoder



$$X = f(Z_1) + f(Z_2)$$

$$X = \mu_1(Z) \times f(Z_1) + \mu_2(Z) \times f(Z_2)$$

Does not work for images with occlusions!

Additive Energy Distribution (AED)

$$p(x \mid z) = \frac{1}{\mathbb{Z}(z)} \exp\Big(-1^T E(x, z) \Big) \quad \text{where} \quad 1^T E(x, z) = \sum_{i=1}^{m} E_i(x, z_i)$$

Conditional distribution of data given factors

Partition Function

Energy Function

Energy Function for each component

- **Assumption**: The energy function can be decomposed as addition of energies with different components of z
  - Natural choice to model inputs that satisfy a conjunction of characteristics
- More expressive than additive decoders; can model interaction between components of z via the partition function $\mathbb{Z}(z) = \int \exp\Big(-1^T \, E(x, z) \Big) dx$

Additive Energy Distribution (AED)

$$p(x \mid z) = \frac{1}{\mathbb{Z}(z)} \exp\Big(-1^T E(x, z) \Big) \quad \text{where} \quad 1^T E(x, z) = \sum_{i=1}^{m} E_i(x, z_i)$$

Conditional distribution of data given factors

Partition Function

Energy Function

Energy Function for each component

- AED expressed with inner product:

$$p(x \mid z) = \frac{1}{\mathbb{Z}(z)} \exp\Big(-\sigma(z)^T E(x) \Big)$$

where $\sigma(z) = [\text{onehot}(z_1), \ldots, \text{onehot}(z_m)]^\top$,

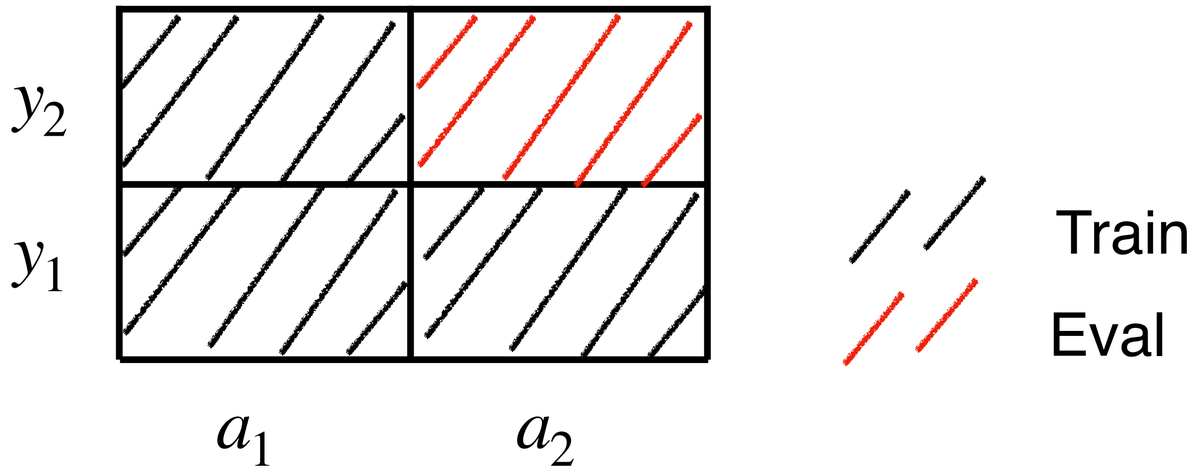
$$E(x) = [E_1(x,1), \ldots, E_1(x,d), \ldots, E_m(x,1), \ldots, E_m(x,d)]^\top$$

CPE with Additive Energy Distributions

Assume we have perfectly estimated energy functions for train groups

$$\sigma(y_1, a_1)^T E(x) = \sigma(y_1, a_1)^T \hat{E}(x)$$
$$\sigma(y_1, a_2)^T E(x) = \sigma(y_1, a_2)^T \hat{E}(x)$$
$$\sigma(y_2, a_1)^T E(x) = \sigma(y_2, a_1)^T \hat{E}(x)$$

Does this imply $\sigma(y_2, a_2)^T E(x) = \sigma(y_2, a_2)^T \hat{E}(x)$?



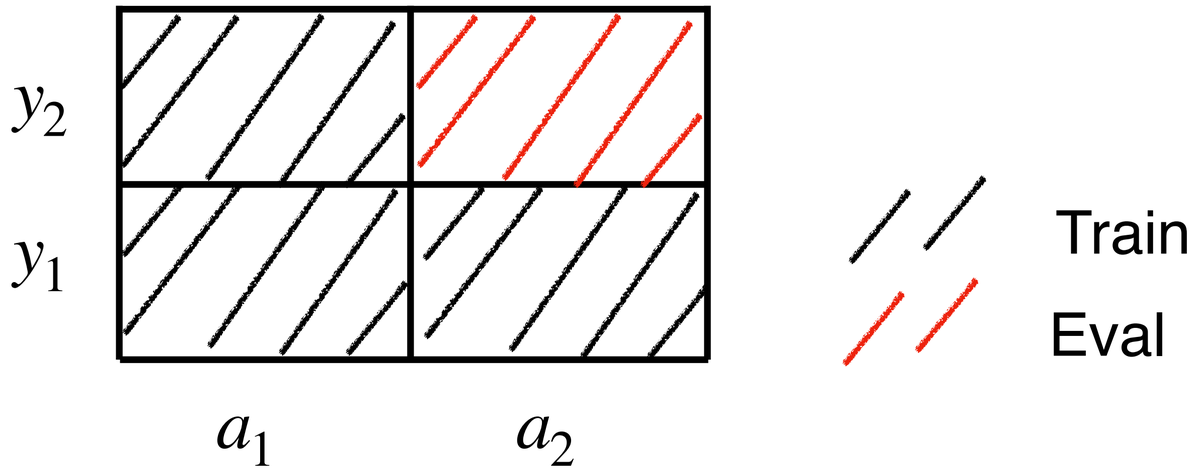
CPE via affine combination of train groups

$$\sigma(y_2, a_2)^T \hat{E}(x) = \sigma(y_2, a_1)^T \hat{E}(x) - \sigma(y_1, a_1)^T \hat{E}(x) + \sigma(y_1, a_2)^T \hat{E}(x)$$

$$\implies \sigma(y_2, a_1)^T E(x) - \sigma(y_1, a_1)^T E(x) + \sigma(y_1, a_2)^T E(x)$$

$$\implies \sigma(y_2, a_2)^T E(x)$$

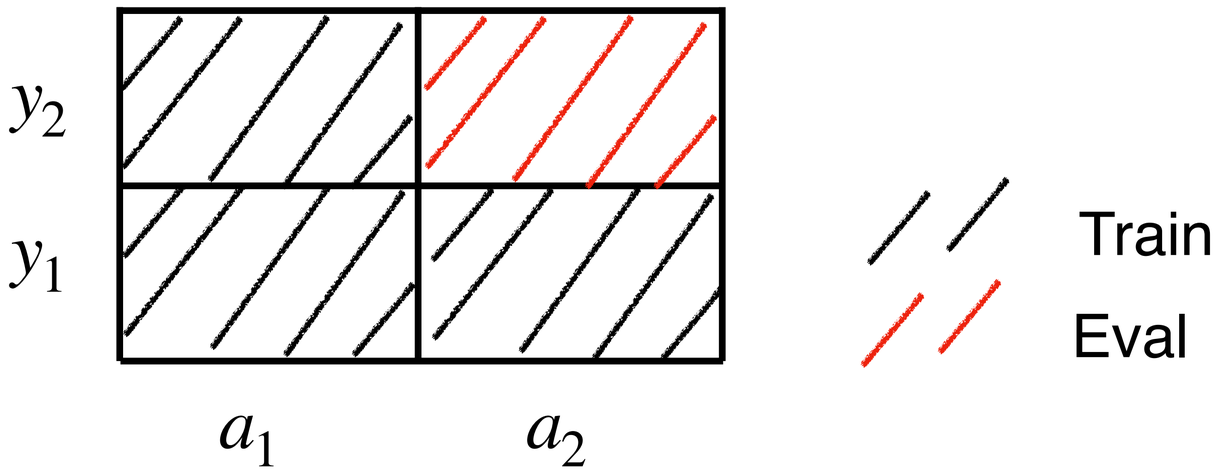
If the novel group can be expressed as an affine combination of train groups, then we can extrapolate the learned energies to the novel groups!



Discrete Affine Hull Extension

- $\mathrm{DAff}(\mathscr{A}) = \left\{ z \in \mathscr{Z} \mid \exists\, \alpha \in \mathbb{R}^k, \sigma(z) = \sum_{i=1}^{k} \alpha_i \sigma\left(z^{(i)}\right), \sum_{i=1}^{k} \alpha_i = 1 \right\}$

where $\mathscr{A} = \{z^{(1)}, \ldots, z^{(k)}\}$, $z^{(i)} \in \mathscr{Z}$



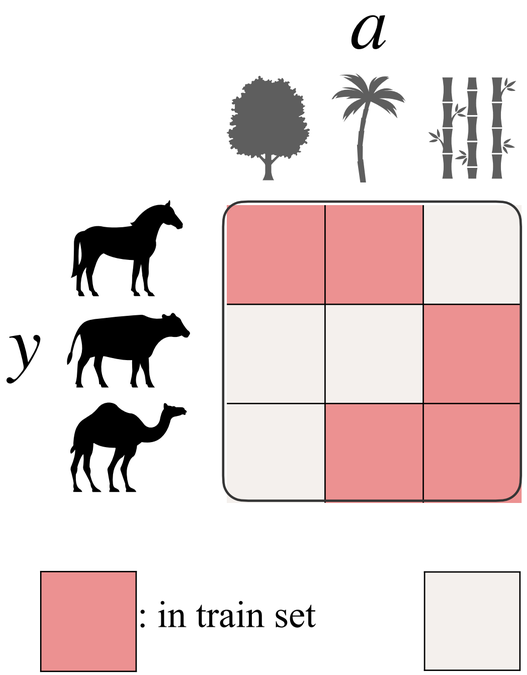
$$\sigma(y_2, a_2) = \sigma(y_2, a_1) - \sigma(y_1, a_1) + \sigma(y_1, a_2)$$

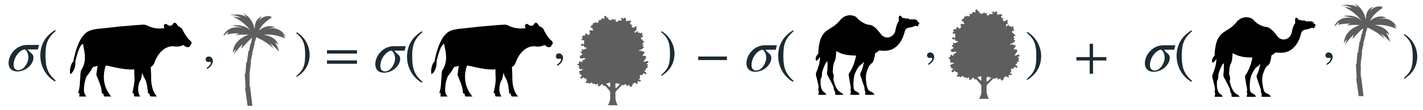
$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = (+1) \cdot \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} + (-1) \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + (+1) \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Discrete Affine Hull Extension

- $\mathrm{DAff}(\mathscr{A}) = \left\{ z \in \mathscr{Z} \mid \exists\, \alpha \in \mathbb{R}^k, \sigma(z) = \sum_{i=1}^{k} \alpha_i \sigma\left(z^{(i)}\right), \sum_{i=1}^{k} \alpha_i = 1 \right\}$

where $\mathscr{A} = \{z^{(1)}, \ldots, z^{(k)}\}$, $z^{(i)} \in \mathscr{Z}$





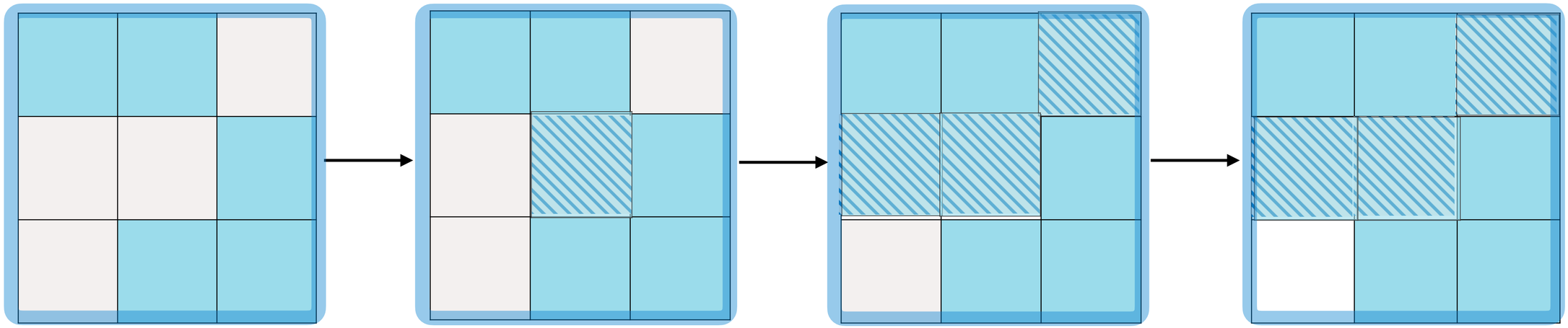
: in train set

: only in test

Discrete Affine Hull Extension

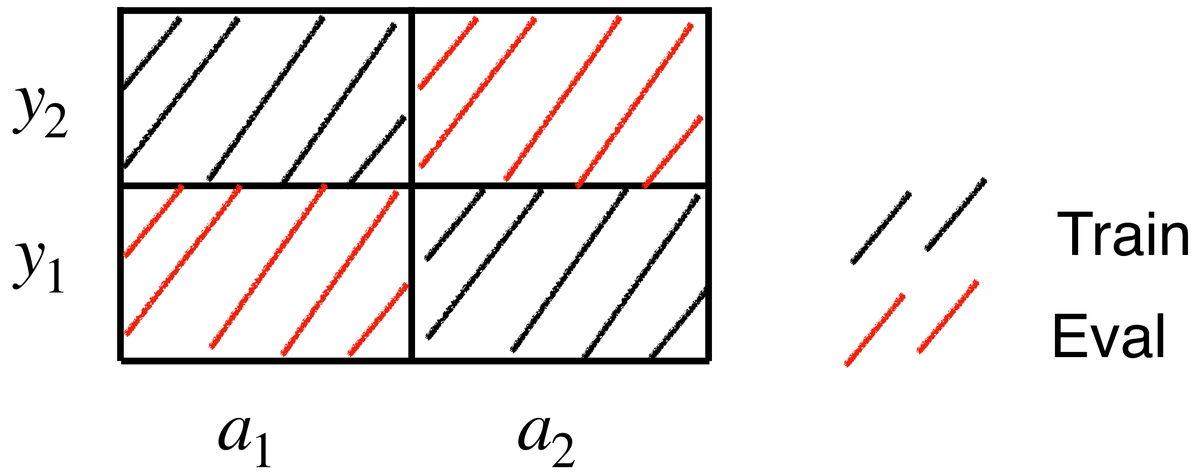
- $\mathrm{DAff}(\mathscr{A}) = \left\{ z \in \mathscr{Z} \mid \exists\, \alpha \in \mathbb{R}^k, \sigma(z) = \sum_{i=1}^{k} \alpha_i \sigma\left(z^{(i)}\right), \sum_{i=1}^{k} \alpha_i = 1 \right\}$

where $\mathscr{A} = \{z^{(1)}, \ldots, z^{(k)}\}$, $z^{(i)} \in \mathscr{Z}$



CPE is not always same as Discrete Affine Hull

Note that extrapolation to novel groups is dependent on the training groups



Extrapolation to Discrete Affine Hull

True Distribution: $$p(x \mid z) = \frac{1}{\mathbb{Z}(z)} \exp\Big(-\sigma(z)^T E(x) \Big)$$

Learned Distribution: $$\hat{p}(x \mid z) = \frac{1}{\hat{\mathbb{Z}}(z)} \exp\Big(-\sigma(z)^T \hat{E}(x) \Big)$$

Theorem: If $p(\,.\,|\,z) = \hat{p}(\,.\,|\,z)\ \forall z \in \mathcal{Z}^{\text{train}}$ then $p(\,.\,|\,z) = \hat{p}(\,.\,|\,z)\ \forall z \in DAff(\mathcal{Z}^{\text{train}})$

Generative Classification with AED

True Model:

$$p(z\,|\,x) = Softmax(\log p(x\,|\,z) + \log p(z)) \quad \text{where} \quad p(x\,|\,z) = \frac{1}{\mathbb{Z}(z)} \exp\Big(-\sigma(z)^T E(x) \Big)$$

Learned Model:

$$\hat{p}(z\,|\,x) = Softmax(\log \hat{p}(x\,|\,z) + \log p(z)) \quad \text{where} \quad \hat{p}(x\,|\,z) = \frac{1}{\hat{\mathbb{Z}}(z)} \exp\Big(-\sigma(z)^T \hat{E}(x) \Big)$$

Corollary: If $p(z\,|\,x) = \hat{p}(z\,|\,x)\ \forall z \in \mathcal{Z}^{\text{train}}$ then $p(z\,|\,x) = \hat{p}(z\,|\,x)\ \forall z \in DAff(\mathcal{Z}^{\text{train}})$

Generative Classification with AED

True Model:

$$p(z\,|\,x) = Softmax(\log p(x\,|\,z) + \log p(z)) \quad \text{where} \quad p(x\,|\,z) = \frac{1}{\mathbb{Z}(z)} \exp\Big(-\sigma(z)^T E(x) \Big)$$

Learned Model:

$$\hat{p}(z\,|\,x) = Softmax(\log \hat{p}(x\,|\,z) + \log p(z)) \quad \text{where} \quad \hat{p}(x\,|\,z) = \frac{1}{\hat{\mathbb{Z}}(z)} \exp\Big(-\sigma(z)^T \hat{E}(x) \Big)$$

Inferring partition function $\mathbb{Z}(z) = \int \exp\Big(-\sigma(z)^T E(x) \Big) dx$ is challenging!

Compositional Risk Minimization (CRM)

- **Additive Energy Classifier:** $\tilde{p}(z \mid x) = \dfrac{\exp\left(-\sigma(z)^T \tilde{E}(x) + \log \hat{p}(z) - \tilde{B}(z) \right)}{\sum_{z' \in \mathcal{Z}^{\text{train}}} \exp\left(-\sigma(z')^T \tilde{E}(x) + \log \hat{p}(z') - \tilde{B}(z') \right)}$

- **CRM First Step:**

  $\hat{E}, \hat{B} \in argmin_{\tilde{E},\tilde{B}} R(\tilde{p})$ where $R(\tilde{p}) = \mathbb{E}_{(x,z)\sim p}\left[-\log \tilde{p}(z \mid x) \right]$

Compositional Risk Minimization (CRM)

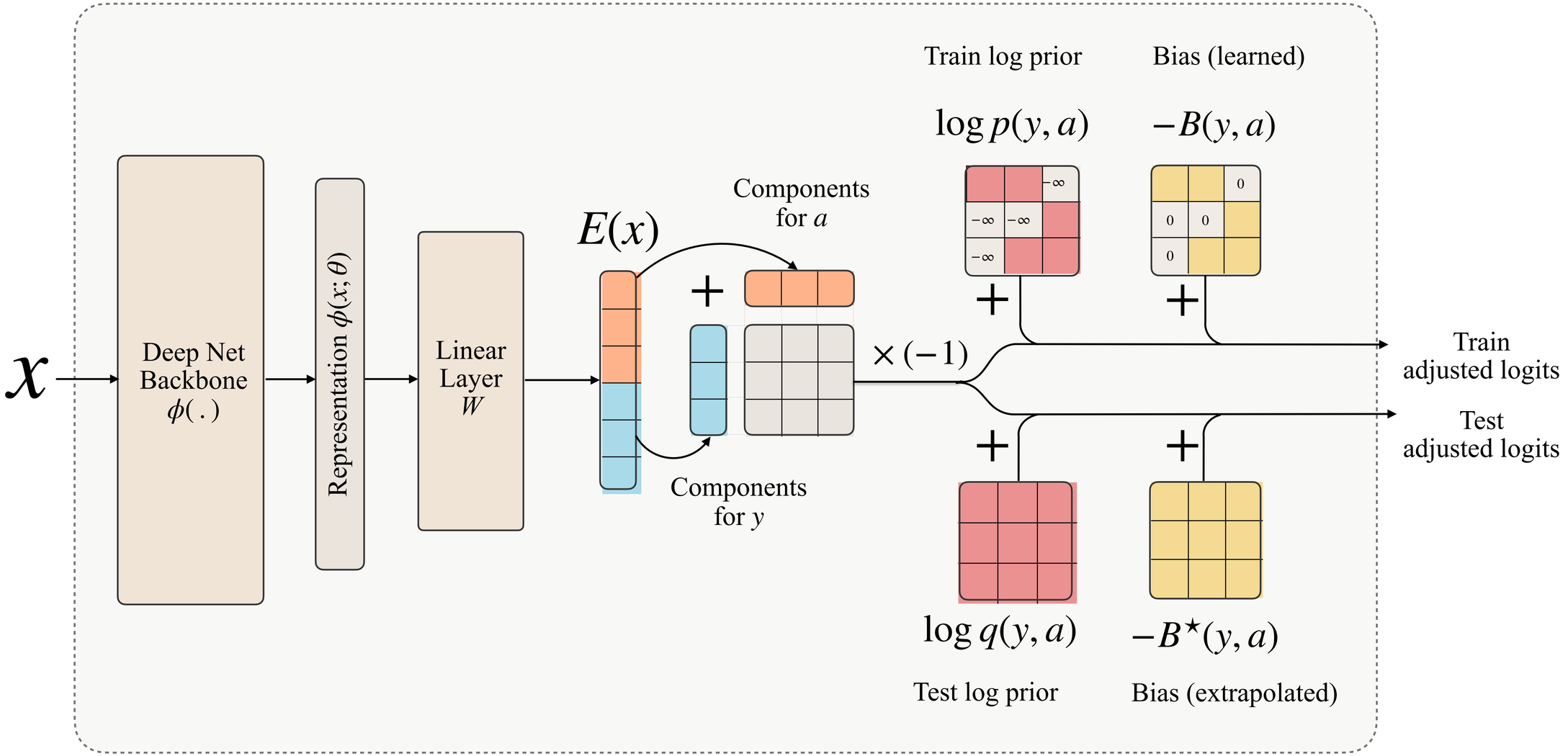
- **Additive Energy Classifier:** $\tilde{p}(z \mid x) = \dfrac{\exp\left(-\sigma(z)^T \tilde{E}(x) + \log \hat{p}(z) - \tilde{B}(z) \right)}{\sum_{z' \in \mathcal{Z}^{\text{train}}} \exp\left(-\sigma(z')^T \tilde{E}(x) + \log \hat{p}(z') - \tilde{B}(z') \right)}$

- **CRM Second Step:**

  Construct $\hat{q}(z \mid x)$ by replacing the prior $\hat{p}(z)$ with $\hat{q}(z)$ and learned bias $\hat{B}(z)$ with extrapolated bias $B^\star(z)$

$$B^\star(z) = \log\left(\mathbb{E}_{x \sim p(x)} \left[\frac{\exp\left(-\sigma(z)^T \hat{E}(x) \right)}{\sum_{\tilde{z} \in \mathcal{Z}^{\text{train}}} \exp\left(-\sigma(\tilde{z})^T \hat{E}(x) + \log p(\tilde{z}) - \hat{B}(\tilde{z}) \right)} \right] \right)$$

Compositional Risk Minimization (CRM)



Provable Extrapolation with CRM

True Model:

$$p(z\,|\,x) = Softmax(\log p(x\,|\,z) + \log p(z)) \quad \text{where} \quad p(x\,|\,z) = \frac{1}{\mathbb{Z}(z)} \exp\Big(-\sigma(z)^T E(x) \Big)$$

Learned Model (Train):

$$\hat{p}(z\,|\,x) = Softmax(\log \hat{p}(x\,|\,z) + \log p(z)) \quad \text{where} \quad \hat{p}(x\,|\,z) = \frac{1}{\hat{B}(z)} \exp\Big(-\sigma(z)^T \hat{E}(x) \Big)$$

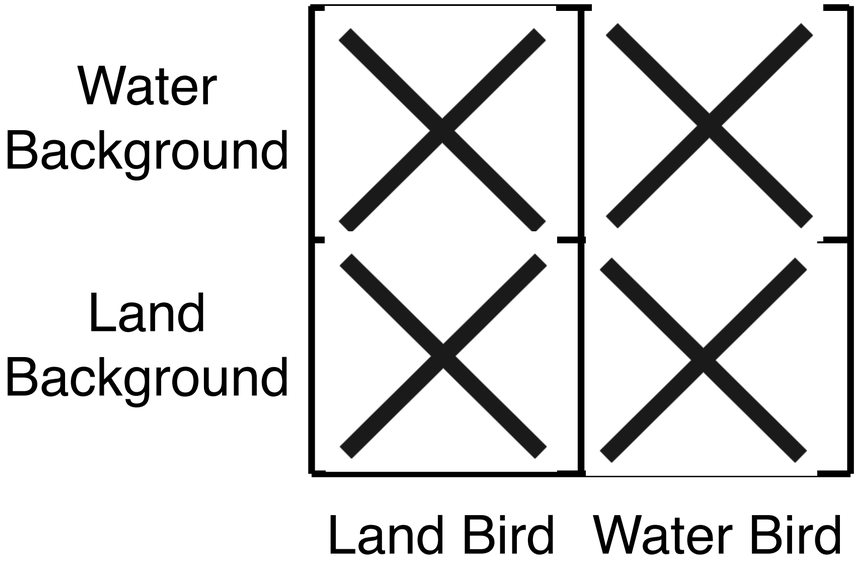
Learned Model (Eval):

$$\hat{q}(z\,|\,x) = Softmax(\log \hat{q}(x\,|\,z) + \log \hat{q}(z)) \quad \text{where} \quad \hat{q}(x\,|\,z) = \frac{1}{B^\star(z)} \exp\Big(-\sigma(z)^T \hat{E}(x) \Big)$$

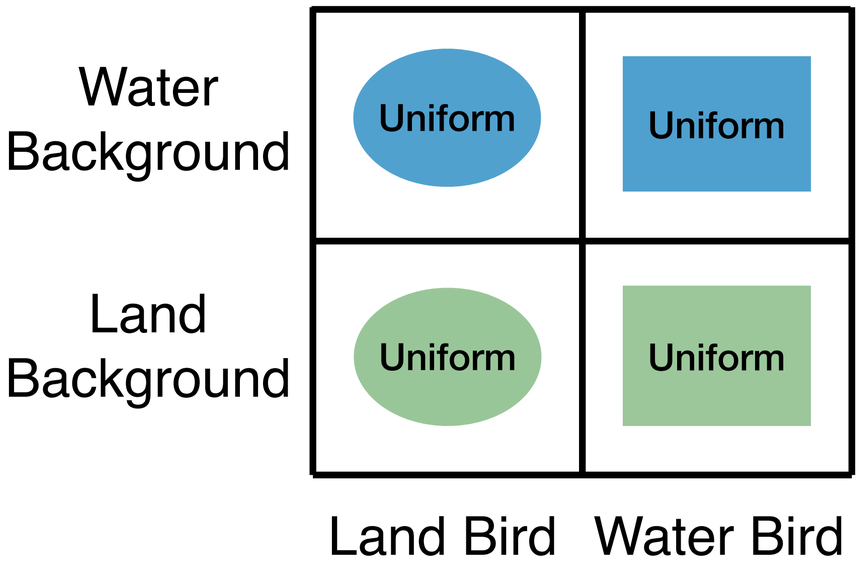
Theorem:

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 $\hat{q}(z\,|\,x) = q(z\,|\,x), \forall z \in \mathcal{Z}^{\text{test}}, \forall x \in \mathbb{R}^n$

Experiments: Setup



Train Distribution



Test Distribution

- Factors $z = (y, a)$ where y denotes the class label and a denotes the spurious attribute
- Compositional Shift: $\mathcal{Z}^{\text{train}} \neq \mathcal{Z}^{\text{test}}$ but $\mathcal{Z}^{\text{test}} = DAff(\mathcal{Z}^{\text{train}})$

Experiments: Results

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)

- 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

Experiments: Ablation

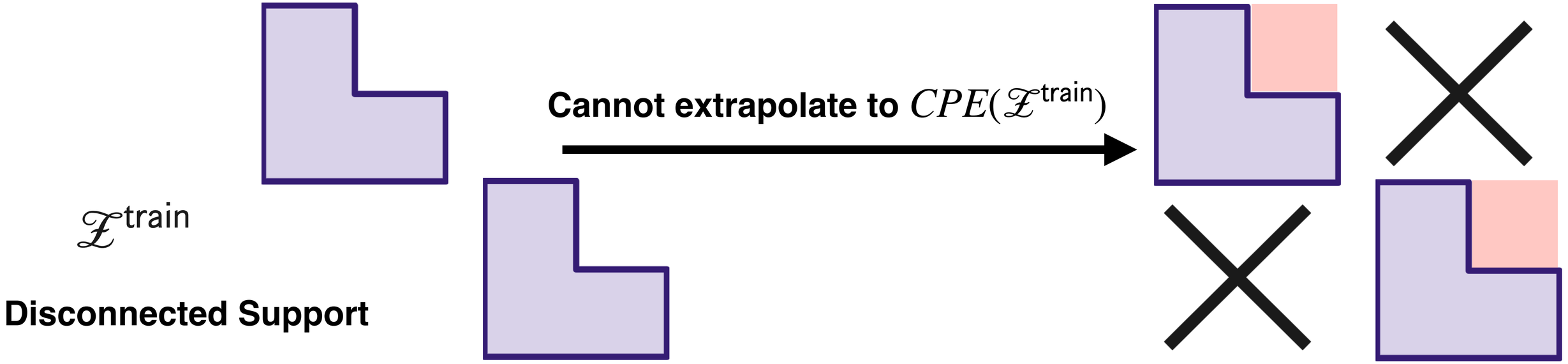
Method	Waterbirds	CelebA	MetaShift	MulitNLI	CivilComments	NICO++
CRM ($\hat{B}$)	55.7 (1.0)	58.9 (0.4)	58.7 (0.6)	29.2 (2.1)	51.9 (1.0)	31.0 (1.0)
CRM	78.7 (1.6)	81.8 (1.2)	73.4 0.7)	57.7 (3.0)	68.1 (0.5)	40.3 (4.3)

- We report Worst Group Accuracy, averaged as a group is dropped from training and validation sets
- CRM ($\hat{B}$) is an ablated version of CRM where we use the trained bias $\hat{B}$ instead of the extrapolated bias B^* mandated by our theory
- The extrapolation step appears crucial for robust compositional generalization. Merely adjusting logits based on shifting group prior probabilities does not suffice

Thank You!

Challenges with Disconnected Support

- Disconnected support makes it hard to extrapolate to $CPE(\mathcal{Z}^{\text{train}})$
- This is a fundamental challenge when the factors z are discrete!



How fast does Discrete Affine Hull grow?

- As we add more factors to $\mathcal{Z}^{\text{train}}$, then $DAff(\mathcal{Z}^{\text{train}})$ would increase as well
- Can we show after enough samples $DAff(\mathcal{Z}^{\text{train}})$ spans the full cartesian product $\mathcal{Z}^{\times}$?

Theorem: Assume $m = 2$, i.e, $z = (z_1, z_2)$ where each z_i has d possible values.

If $|\mathcal{Z}^{\text{train}}| > 8cd\log(d/2)$, then $DAff(\mathcal{Z}^{\text{train}}) = \mathcal{Z}^{\times}$ with probability $\geq 1 - \frac{1}{c}$

How fast does Discrete Affine Hull grow?

- As we add more factors to $\mathscr{Z}^{\text{train}}$, then $DAff(\mathscr{Z}^{\text{train}})$ would increase as well
- Can we show after enough samples $DAff(\mathscr{Z}^{\text{train}})$ spans the full cartesian product $\mathscr{Z}^{\times}$?

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

Table 12 Numerical experiments to check the probability that the affine hull of random $\mathcal{O}(\text{poly}(m * d))$ one-hot concatenations span the entire set $\mathscr{Z}$. 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.