Learning Recourse on Instance Environment to Enhance Prediction Accuracy

Lokesh N

Sai Koushik

Abir De

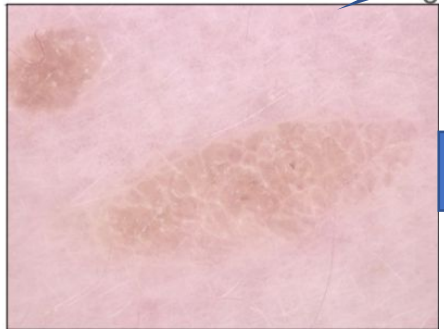
Sunita Sarawagi

Problem Statement

- ML models make incorrect predictions on input instances obtained from poor environments.
- Should we settle with incorrect predictions?
- No, we design a recourse module that seeks instances under alternative settings.
    - The instances generated under these settings are hopefully amenable to correct predictions.

Skin-Lesion Example

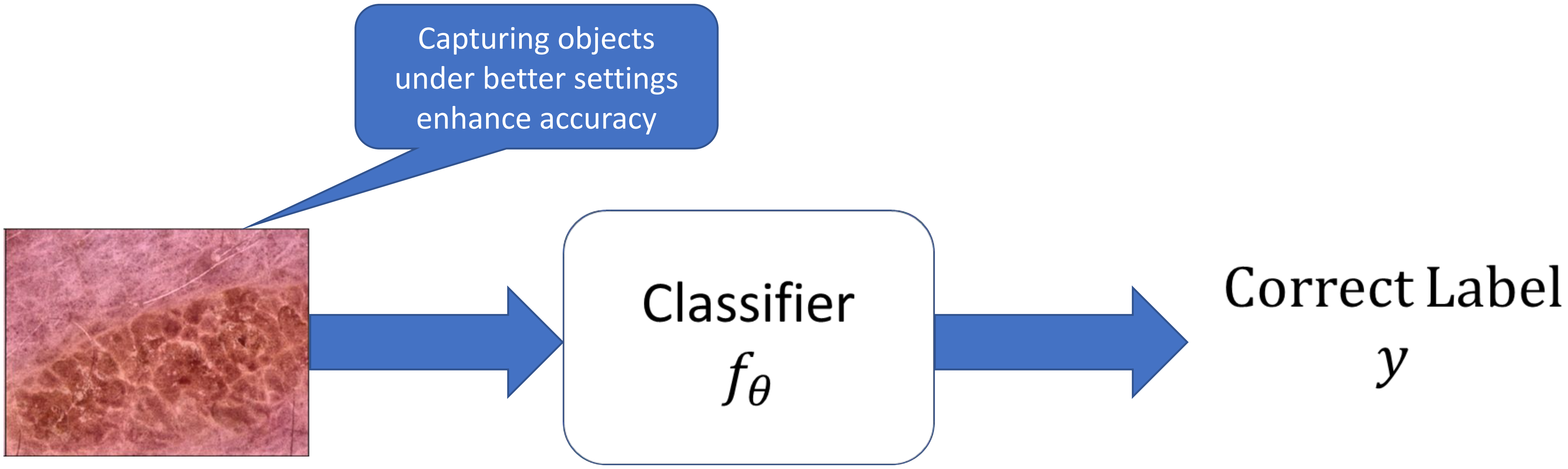
This image is poor because it is captured under extreme lighting



Classifier

f_θ

Skin-Lesion Example



A 3D object recognition task

- We consider the shapenet Dataset that consists of 3D models of many kinds of objects
- These objects can be rendered into 2D images under various settings.
- The settings used to render the object affects the classifier performance.

A Chair object under 9 settings

Training Dataset

$$D = \left\{ \{x_{ij}, \beta_{ij}\}_{j=1}^{B_i}, y_i \right\}_{i=1}^{N}$$

Recourse Architecture

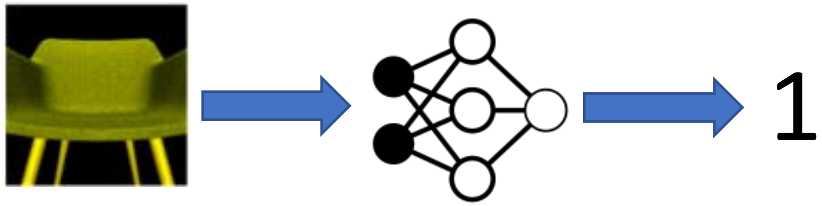


Image x under poor setting β

Recourse Trigger $\Pi(x, \beta)$

Recourse Architecture

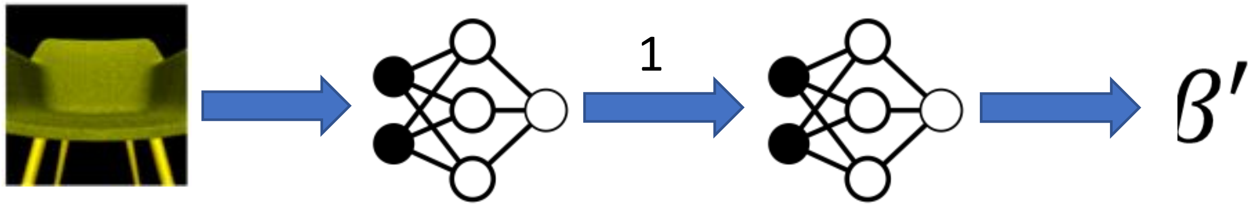
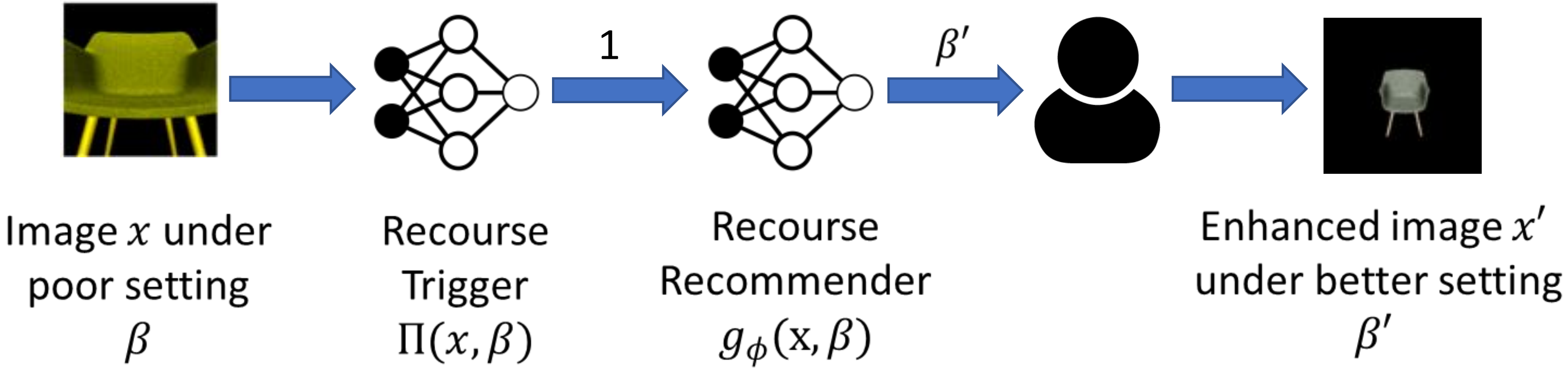


Image x under poor setting β

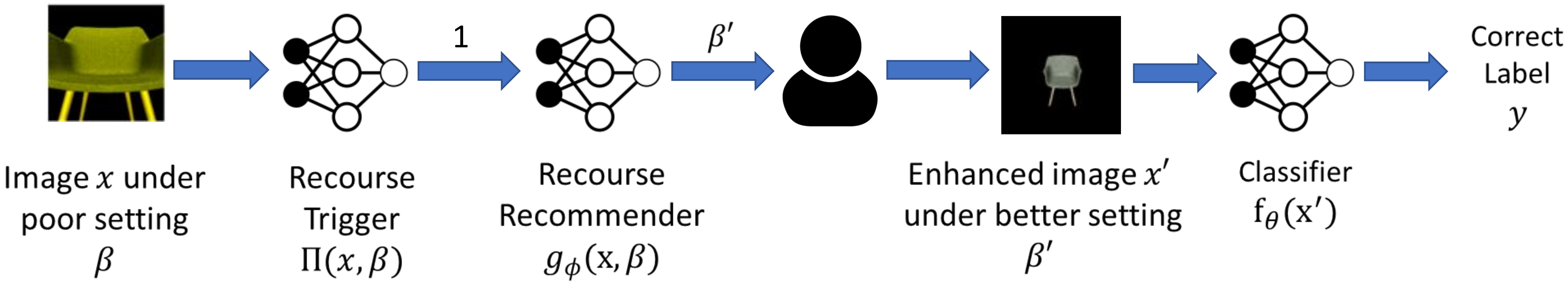
Recourse Trigger $\Pi(x, \beta)$

Recourse Recommender $g_\phi(\mathrm{x}, \beta)$

Recourse Architecture



Recourse Architecture



Training Objective

For instances that do not need recourse

$$\max_{\theta,\phi,\pi} \sum_{\substack{i \in D \\ j \in B}} \log \Big[\boxed{\left(1-\pi(\mathbf{x}_{ij},\boldsymbol{\beta}_{ij})\right) f_\theta(y_i \mid \mathbf{x}_{ij})} \\ + \pi(\mathbf{x}_{ij},\boldsymbol{\beta}_{ij}) f_\theta(y_i \mid Z(z_i, \operatorname{argmax}_{\boldsymbol{\beta}} g_\phi(\boldsymbol{\beta} \mid \mathbf{x}_{ij},\boldsymbol{\beta}_{ij}))) \Big]$$

Training Objective

For instances that need recourse

$$\max_{\theta,\phi,\pi} \sum_{\substack{i \in D \\ j \in B}} \log \Big[\left(1 - \pi(\mathbf{x}_{ij}, \boldsymbol{\beta}_{ij})\right) f_\theta(y_i \mid \mathbf{x}_{ij}) + \boxed{\pi(\mathbf{x}_{ij}, \boldsymbol{\beta}_{ij}) f_\theta(y_i \mid Z(z_i, \operatorname{argmax}_{\boldsymbol{\beta}} g_\phi(\boldsymbol{\beta} \mid \mathbf{x}_{ij}, \boldsymbol{\beta}_{ij})))} \Big]$$

Training Objective

$$\max_{\theta,\phi,\pi} \sum_{\substack{i\in D \\ j\in B}} \log \Big[\big(1-\pi(\mathbf{x}_{ij},\boldsymbol{\beta}_{ij})\big)\, f_\theta(y_i \,|\, \mathbf{x}_{ij}) + \pi(\mathbf{x}_{ij},\boldsymbol{\beta}_{ij}) f_\theta\big(y_i \,|\, \boxed{Z(z_i, \mathrm{argmax}_{\boldsymbol{\beta}}\, g_\phi(\boldsymbol{\beta} \,|\, \mathbf{x}_{ij},\boldsymbol{\beta}_{ij}))}\big) \Big]$$

Proxy for human
We do not model Z

Training Objective

$$\max_{\theta,\phi,\pi} \sum_{\substack{i\in D\\ j\in B}} \log \Big[\big(1-\pi(\mathbf{x}_{ij},\boldsymbol{\beta}_{ij})\big)\, f_\theta(y_i \mid \mathbf{x}_{ij}) \\ + \pi(\mathbf{x}_{ij},\boldsymbol{\beta}_{ij}) f_\theta(y_i \mid Z(z_i, \mathrm{argmax}_{\boldsymbol{\beta}}\, g_\phi(\boldsymbol{\beta} \mid \mathbf{x}_{ij},\boldsymbol{\beta}_{ij}))) \Big]$$

$$\text{subject to,} \sum_{i\in D, j\in B} \pi(\mathbf{x}_{ij}) \le b,$$

$$\pi(\mathbf{x}_{ij},\boldsymbol{\beta}_{ij}) \in \{0,1\}$$

Classifier Training ($f_\theta(y|x)$)

- Training the classifier on entire training data may be suboptimal especially when some poor instances will be recoursed at test time.
- Thus, we should focus training f_θ on instances after recourse.

We use an iterative greedy algorithm to identify instances likely to be recoursed

Algorithm 1: GREEDY
for training f_θ

Require: Data $T = \{y_i, \{$

$V = \cup_{i \in D} \{\{i\} \times B_i\}$
$R \leftarrow \emptyset, \theta^0(\emptyset) \leftarrow \text{TRAIN}(F(\bullet, \emptyset))$
for $k \in [b]$ **do**
  for $(i, j) \in V \backslash R$ **do**
    $\mathcal{L}[(i,j)] = F(\theta^k(R \cup \{(i,j)\}), R \cup \{(i,j)\})$
  $(i^*, j^*) \leftarrow \text{argmax}_{(i,j) \in V \backslash R} \mathcal{L}[(i,j)]$
  $R \leftarrow R \cup \{(i^*, j^*)\}$
  $\theta^{k+1}(R) \leftarrow \text{TRAIN}(F(\bullet, R))$
Return $\theta^{k+1}(R)$

Classifier Training ($f_\theta(y|x)$)

- Training the classifier on entire training data may be suboptimal especially when some poor instances will be recoursed at test time.
- Thus, we should focus training f_θ on instances after recourse.

Algorithm 1: GREEDY
for training f_θ

Require: Data $T = \{y_i, \{$

$V = \cup_{i \in D}\{\{i\} \times B_i\}$

$R \leftarrow \emptyset,\ \theta^0(\emptyset) \leftarrow$ TR $(F(\bullet, \emptyset))$

for $k \in [b]$ **do**

  for $(i, j) \in V \backslash R$ **do**

    $\mathcal{L}[(i,j)] = F(\theta^k(R \cup \{(i,j)\}), R \cup \{(i,j)\})$

  $(i^*, j^*) \leftarrow \text{argmax}_{(i,j) \in V \backslash R}\, \mathcal{L}[(i,j)]$

  $R \leftarrow R \cup \{(i^*, j^*)\}$

  $\theta^{k+1}(R) \leftarrow$ TRAIN$(F(\bullet, R))$

Return $\theta^{k+1}(R)$

We first estimate the improvement in accuracy if an instance is recoursed

Classifier Training ($f_\theta(y|x)$)

- Training the classifier on entire training data may be suboptimal especially when some poor instances will be recoursed at test time.
- Thus, we should focus training f_θ on instances after recourse.

Algorithm 1: GREEDYALGORITHM for training f_θ

Require: Data $T =$

$V = \cup_{i \in D}\{\{i\}$

$R \leftarrow \emptyset,\ \theta^0(\emptyset) \leftarrow$

for $k \in [b]$ **do**

  for $(i, j) \in V \backslash R$

    $\mathcal{L}[(i, j)] =$

    $F(\theta^k(R \cup \{(i, j)\}), R \cup \{(i, j)\})$

  $(i^*, j^*) \leftarrow \mathrm{argmax}_{(i,j) \in V \backslash R}\, \mathcal{L}[(i, j)]$

  $R \leftarrow R \cup \{(i^*, j^*)\}$

  $\theta^{k+1}(R) \leftarrow \textsc{Train}(F(\bullet, R))$

Return $\theta^{k+1}(R)$

We drop the instance that is
(a) Poor
(b) Amenable to recourse

Classifier Training ($f_\theta(y|x)$)

- Training the classifier on entire training data may be suboptimal especially when some poor instances will be recoursed at test time.
- Thus, we should focus training f_θ on instances after recourse.

Algorithm 1: GREEDYALGORITHM for training f_θ

Require: Data $T = \{y_i, \{\mathbf{x}_{ij}, \boldsymbol{\beta}_{ij}\}_{j \in B_i}\}, b$

$V = \cup_{i \in D}\{\{i\}$

$R \leftarrow \emptyset,\ \theta^0(\emptyset) \leftarrow$

for $k \in [b]$ **do**

  for $(i, j) \in V$

    $\mathcal{L}[(i, j)] =$

    $F(\theta^k(R \cup \{($ $), R \cup \{(i, j)\})$

  $(i^*, j^*) \leftarrow \arg$ $\text{x}_{(i,j) \in V \setminus R}\, \mathcal{L}[(i, j)]$

  $R \leftarrow R \cup \{($ $, j^*)\}$

  $\theta^{k+1}(R) \leftarrow \text{TRAIN}(F(\bullet, R))$

Return $\theta^{k+1}(R)$

We train the classifier by iteratively dropping such instances

Recourse Recommender Training (g_ϕ)

- Given an instance (x, β), g_ϕ outputs alternate setting β' to render objects.
- But we do not have supervision for such good settings.
- Thus, we can make g_ϕ emit settings that produce better classifier accuracy.

It is enough if g_ϕ predicts good settings for instances that need recourse

$$\underset{\phi}{\operatorname{argmax}} \sum_{\substack{i \in D, j \in B_i \\ \boxed{\pi(\mathbf{x}_{ij})=1}}} \max_{\boldsymbol{\beta}} \log \left[f_\theta(y_i \mid Z(z_i, \boldsymbol{\beta}))\, g_\phi(\boldsymbol{\beta} \mid \mathbf{x}_{ij}, \boldsymbol{\beta}_{ij}) \right]$$

Recourse Recommender Training (g_ϕ)

- Given an instance (x, β), g_ϕ outputs alternate setting β' to render objects.
- But we do not have supervision for such good settings.
- Thus, we can make g_ϕ emit settings that accuracy.

But Z is unavailable

$$\operatorname*{argmax}_{\phi} \sum_{\substack{i\in D, j\in B_i \\ \pi(\mathbf{x}_{ij})=1}} \max_{\boldsymbol{\beta}} \log \left[f_\theta(y_i \mid \boxed{Z(z_i, \boldsymbol{\beta})})\, g_\phi(\boldsymbol{\beta} \mid \mathbf{x}_{ij}, \boldsymbol{\beta}_{ij})\right]$$

Recourse Recommender objective

We borrow β' labels from within the training data from instances that have better accuracy

$$\max_{\phi} \sum_{\substack{i \in D_\delta \\ j \in B_i}} \boxed{\max_{r \in B_i}} \log \left[f_\theta(y_i \mid \mathbf{x}_{ir}) \boxed{g_\phi(\boldsymbol{\beta}_{ir} \mid \mathbf{x}_{ij}, \boldsymbol{\beta}_{ij})} \right]$$

Recourse Recommender objective

For groups that have atleast one good instance, we borrow β' from them

For groups that have all bad instances we set β' that corresponds to the best estimated counterfactual accuracy

$$\max_{\phi} \sum_{\substack{i \in D_\delta \\ j \in B_i}} \max_{r \in B_i} \log \left[f_\theta(y_i \mid \mathbf{x}_{ir})\, g_\phi(\boldsymbol{\beta}_{ir} \mid \mathbf{x}_{ij}, \boldsymbol{\beta}_{ij}) \right] + \sum_{\substack{i \notin D_\delta \\ j \in B_i}} \log g_\phi \left(\mathrm{argmax}_{\boldsymbol{\beta}}\, f^{\mathrm{CF}}(y_i \mid \mathbf{x}_{ij}, \boldsymbol{\beta}) \mid \mathbf{x}_{ij}, \boldsymbol{\beta}_{ij} \right)$$

Recourse Recommender objective

Counterfactual accuracy estimated for x_{ij} under alternate setting β

$$\max_{\phi} \sum_{\substack{i \in D_\delta \\ j \in B_i}} \max_{r \in B_i} \log \left[f_\theta(y_i \,|\, \mathbf{x}_{ir})\, g_\phi(\boldsymbol{\beta}_{ir} \,|\, \mathbf{x}_{ij}, \boldsymbol{\beta}_{ij}) \right] + \sum_{\substack{i \notin D_\delta \\ j \in B_i}} \log g_\phi \left(\mathrm{argmax}_{\boldsymbol{\beta}} \boxed{f^{\mathrm{CF}}(y_i \,|\, \mathbf{x}_{ij}, \boldsymbol{\beta})} \,|\, \mathbf{x}_{ij}, \boldsymbol{\beta}_{ij} \right)$$

$$f^{\mathrm{CF}}(y \,|\, \mathbf{x}, \boldsymbol{\beta}) = \frac{\sum_{(i,j) \in V} \mathbb{I}[y_i = y, \boldsymbol{\beta}_{ij} = \boldsymbol{\beta}] f_{\hat{\theta}}(y_i = y \,|\, \mathbf{x}_{ij})}{\sum_{(i,j) \in V} \mathbb{I}[y_i = y, \boldsymbol{\beta}_{ij} = \boldsymbol{\beta}]}$$

Recourse Trigger (π)

- Recourse Trigger module does not contain any parameters.
- During inference we trigger reco factual accuracy under the recourse set acy than the original instance.

Since we do not have ground truth labels during inference, we use $y_{max} = argmax_y f_\theta(y|x_{ij})$

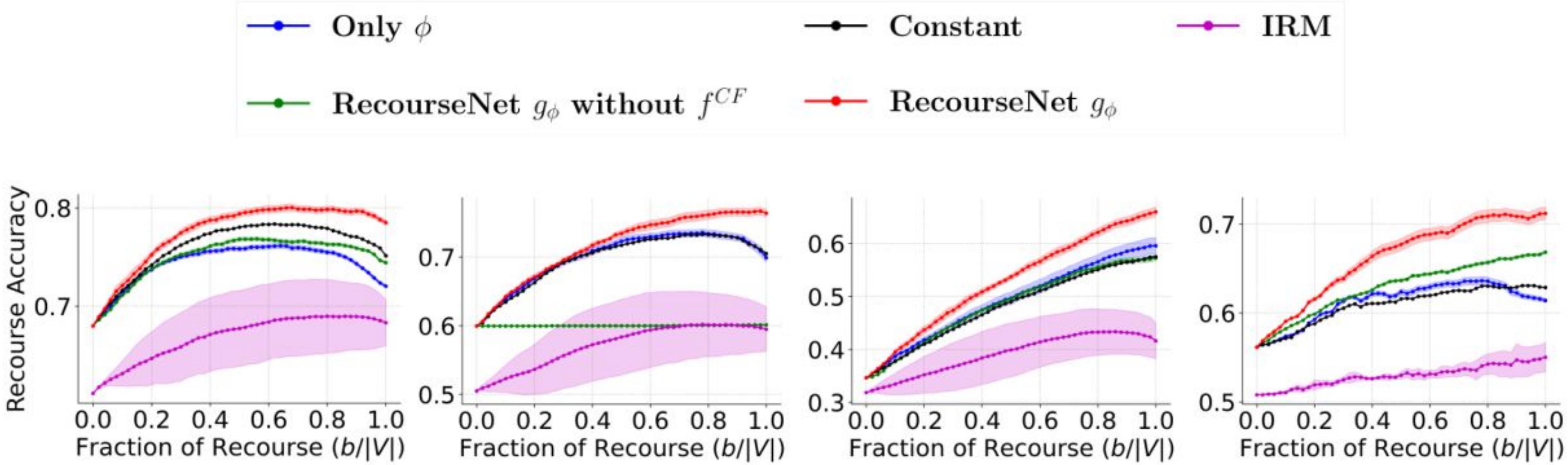
$$\pi(\mathbf{x}_{ij}, \boldsymbol{\beta}_{ij}) = \mathbb{I}[f^{\mathrm{CF}}(\boxed{y_{\max}} \mid \mathbf{x}_{ij}, \boldsymbol{\beta}'_{ij}) > f_{\hat{\theta}}(y_{\max} \mid \mathbf{x}_{ij})]$$

$$\boldsymbol{\beta}'_{ij} = \mathrm{argmax}_{\boldsymbol{\beta}}\, g_{\hat{\phi}}(\boldsymbol{\beta} \mid \mathbf{x}_{ij}, \boldsymbol{\beta}_{ij}))$$

Classifier Performance

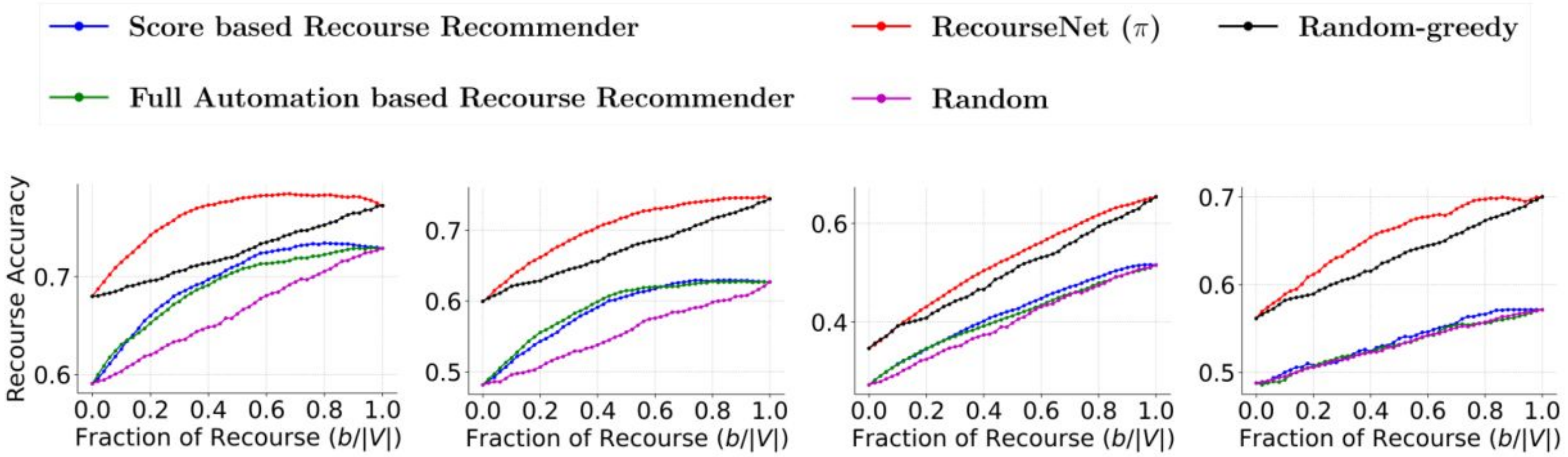
Training Data	Shapenet-Large	Shapenet-Small	Speech-Commands	Skin-Lesion
Full-data (Baseline)	71.93 ± 0.63	62.97 ± 0.80	51.85 ± 1.08	56.42 ± 0.80
One-shot subsetting	72.63 ± 0.54	65.55 ± 1.11	54.66 ± 1.2	60.89 ± 1.11
Iterative greedy (Ours)	$\mathbf{77.14 \pm 0.63}$	$\mathbf{74.13 \pm 1.10}$	$\mathbf{65.76 \pm 1.44}$	$\mathbf{68.62 \pm 0.90}$

Recourse Recommender Performance



(a) Shapenet-Large. (b) Shapenet-Small. (c) Speech Commands. (d) Skin-Lesion.

Recourse Trigger Performance



(a) Shapenet-Large (b) Shapenet-Small (c) Speech Commands (d) Skin-Lesion

Thank You!