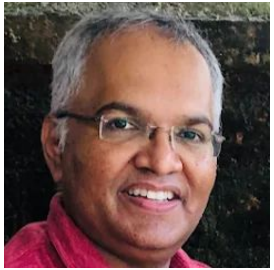
Improving Generalization Via Meta-Learning on Hard Samples



Nishant Jain

Arun S. Suggala

Pradeep Shenoy

Google Research

CVPR

JUNE 17-21, 2024

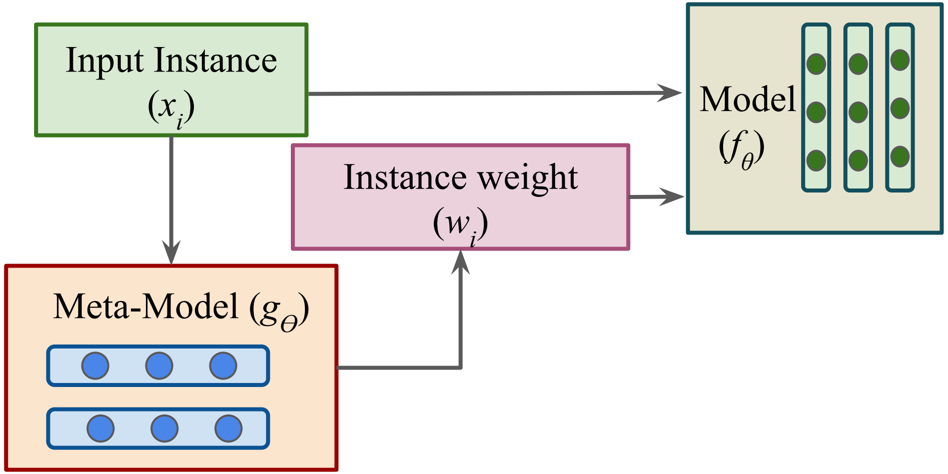
SEATTLE, WA

Introduction

- Neural Networks often struggle to generalize well on test distributions including both in-domain and out-of-domain cases.
- The standard ERM optimization has usually been seen as a sub-optimal solution to the exact distributional approximation.
- In this work, we instead formulate a learned re-weighting based formulation for training models, which exploits the fact that models focusing more on the hard examples usually tend to perform better on test distributions.

Instance Conditional Learned Re-weighting

- *Train Set*: $\{x_i, y_i\}$
- *Special Validation Set*: (x_j^v, y_j^v)
- $f_\theta(\cdot)$: *classifier*
- $g_\phi(\cdot)$: *meta-network*



Objective:

$$\phi^* = \arg\min_{\phi} \frac{1}{M} \sum_{j=1}^{M} l(y_j^v, f_{\theta^*(\phi)}(x_j^v))$$

$$s.t.\ \theta^*(\phi) = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} g_\phi(x_i) \cdot l(y_i, f_\theta(x_i))$$

Idea: Can we improve generalization by using *hard samples* as target set?

Meta **O**ptimization of **Le**arned **Re**weighting

A tri-level formulation given overall data $S = \{(x, y)\}_{i=1}^{N+M}$, a splitting function Θ diving S into a target set $\Theta(S)$ and a training set $\Theta(S)^c$:

$$\Theta^* = \arg\max_{\Theta} \sum_{(x,y)\in\Theta(S)} \ell(y, f_{\theta^*(\phi^*(\Theta),\Theta)}(x))$$

$$\texttt{where } \phi^*(\Theta) = \arg\min_{\phi} \sum_{(x,y)\in\Theta(S)} \ell(y, f_{\theta^*(\phi,\Theta)}(x))$$

$$\texttt{s.t.} \theta^*(\phi, \Theta) = \arg\min_{\theta} \sum_{(x,y)\in\Theta(S)^c} \phi(x)\ell(y, f_\theta(x)).$$

MOLERE(2): A heuristic and an optimization

Train twice Heuristic

- Train an ERM classifier on supplied train-val split
- Use model margin to rank-order (train+val) pool
- Pick hardest samples as new val set (LRW-Hard)
    - Additional controls: easy & random validation sets (LRW-Easy, LRW-Random)

E2E Optimization

$$\Theta^*, \phi^* = \arg\max_{\Theta} \min_{\phi} \sum_{k=1}^{N+M} \mathbb{1}\{\Theta[k] = 0\} \left(l_{val}(y_k, f_{\theta^*(\Theta,\phi)}(x_k)) - \mathcal{L}_{split} \right)$$

$$where \quad \theta^*(\Theta, \phi) = \arg\min_{\theta} \sum_{i=1}^{N} \mathbb{1}\{\Theta[k] = 1\} w_i l_{tr}(y_i, f_\theta(x_i))$$

$$\mathcal{L}_{split} = CE(\mathbb{P}_{splitter}(z_i|x_i, y_i), \mathbb{I}_{y_i}(\hat{y}))$$

$$\texttt{where} \quad \hat{y} = \arg\max \mathbb{P}_{predictor}(y|x_i)$$

- Inner objective of bilevel optimization minimizes loss on current validation set
- Outer objective is a *minimax optimization*
    - Propose a <train,val> split using a splitter $g_\Theta(x)$
    - Propose an instance weight using a scorer $g_\phi(x)$
    - Splitter aims to maximize validation loss, while reweighter aims to minimize it

MOLERE: A robust optimization objective

- Given total number of samples $N + M \to \infty$, and $\lim_{N,M\to\infty} \frac{M}{N+M} = \delta$, the objective of MOLERE is equivalent to:

$$\max_{S:|S|=\delta(N+M)} \min_{\theta} \sum_{(x,y)\in S} \ell(y, f_\theta(x)).$$

- Thus, in the limit of infinite samples, MOLERE identifies hardest examples and learns a classifier that minimizes error on these samples.
- Also, the above objective is the **dual** of the popular Distributionally Robust Optimization:

$$\min_{\theta} \max_{S:|S|=\delta(N+M)} \sum_{(x,y)\in S} \ell(y, f_\theta(x)).$$

MOLERE Algorithm

Algorithm 1 LRWOpt: The Overall One-Shot Algorithm.

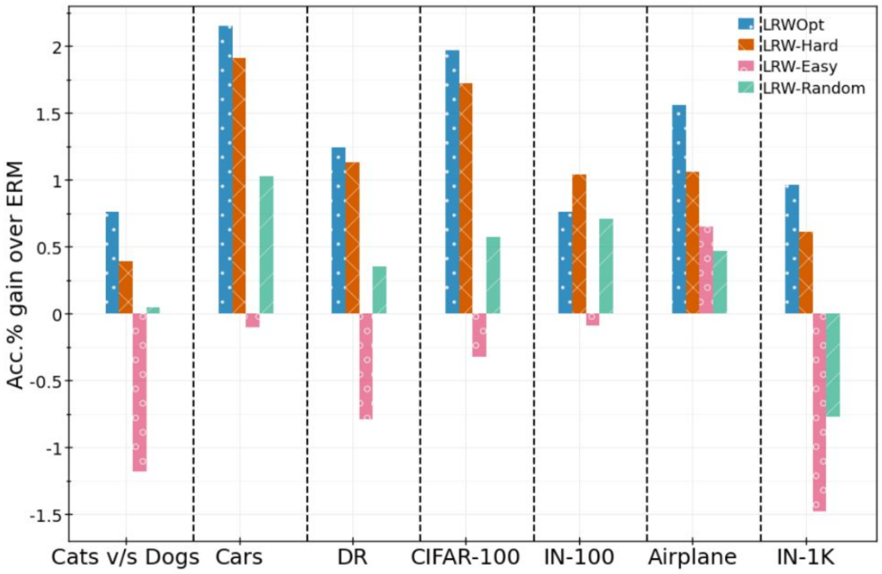
Require: θ, Θ, ϕ, learning rates $(\beta_1, \beta_2, \beta_3)$, $\mathcal{S}$, N, M.
Ensure: Robustly trained classifier parameters θ.

Randomly initialize θ, Θ and ϕ;
initialize ge = 0; ▷ Difference b/w train and val error
for e=1 **to** MaxEpochs **do**
    $\mathcal{S}_{tr}, \mathcal{S}_{val} =$ GenerateSplit($\mathcal{D}, \Theta$)
    for $b = 1$ **to** M//m **do** ▷ m is the batch size
        $\{(x_i^v, y_i^v)\}_{i=1}^m$ = SampleMiniBatch($\mathcal{S}_{val}, m$);
        $\Theta \leftarrow \Theta - \beta_1 \nabla_\Theta \sum (\mathcal{L}_{split} - \ell(y_i^v, f_\theta(x_i^v)))$
        $\phi \leftarrow \phi - \beta_2 \nabla_\phi \sum (\ell(y_i^v, f_\theta(x_i^v)) - \mathcal{L}_{split})$
        for $j = 1$ **to** Q **do**
            $\{(x_i, y_i)\}_{i=1}^n$ SampleMiniBatch($\mathcal{D}_t, n$);
            $\theta \leftarrow \theta - \beta_3 \nabla_\theta \sum g_\phi(x_i) \ell(f_\theta(x_i), y_i)$;
        end for
    end for
    if $\sum \ell(y_i^v, f_\theta(x_i^v)) - \sum \ell(y_i, f_\theta(x_i))$ <ge **then**
        break;
    end if
    ge = $\sum \frac{1}{M} \ell(y_i^v, f_\theta(x_i^v)) - \frac{1}{N} \sum \ell(y_i, f_\theta(x_i))$
end for

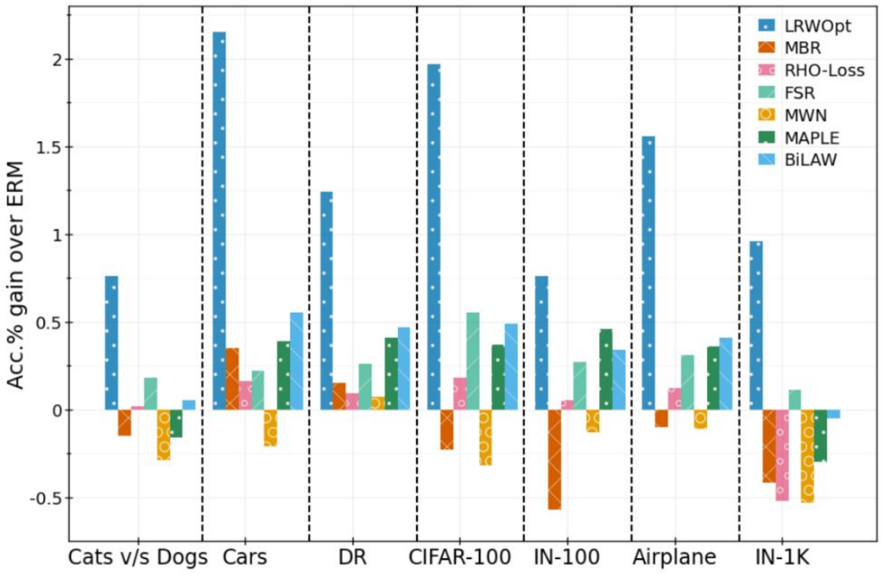
Results and Analysis

Robustness on the Benchmark Datasets: In-Distribution

MOLERE v/s the Heuristics

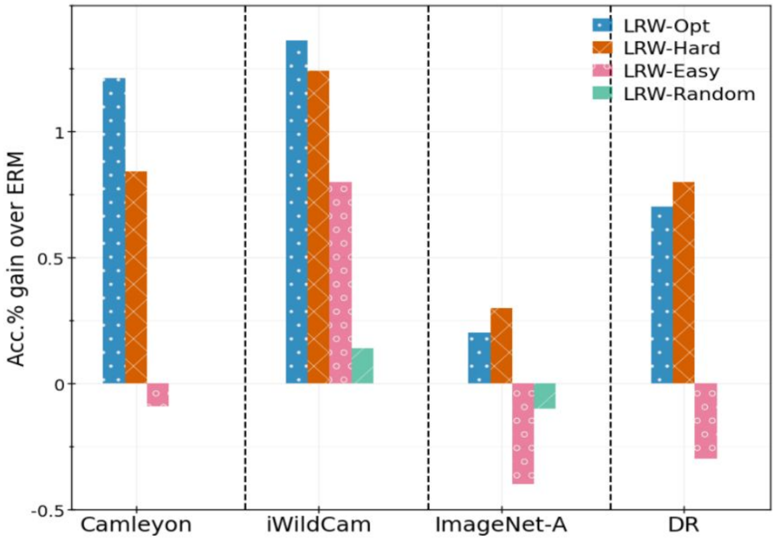


MOLERE v/s existing re weighing methods

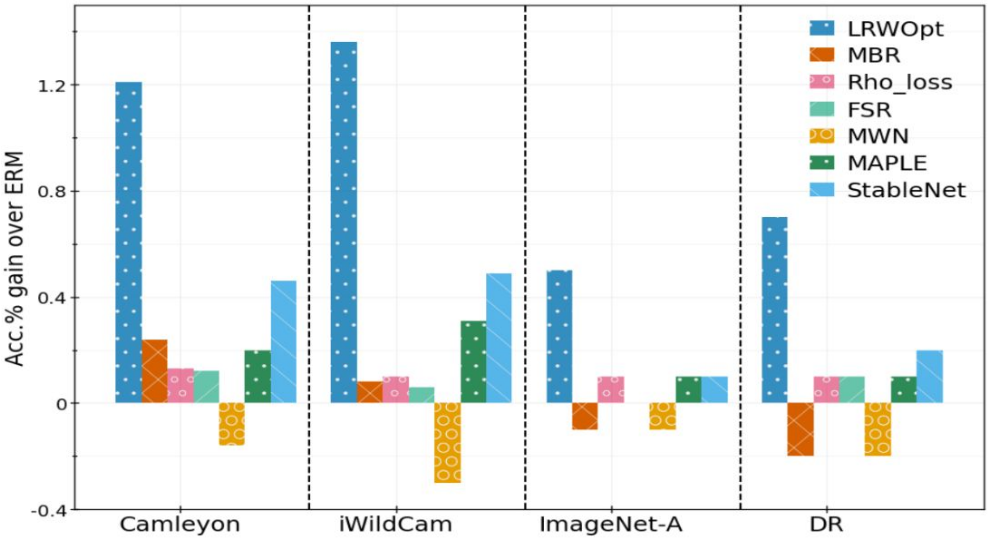


Robustness on the Benchmark Datasets: OOD

MOLERE v/s the Heuristics



MOLERE v/s existing re weighing methods



Practical Label Noise Settings and Skewed Label setups

Instance dependent noise:

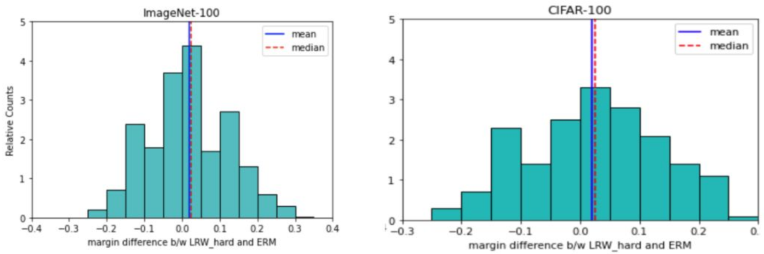
	MWN	FSR	L2R	MAPLE	GDW	Ours
Inst. C-10	65.89	67.12	70.21	70.34	69.12	**71.87**
Clothing-1M	72.79	72.07	72.22	71.67	73.12	**73.97**

Skewed Label Scenario on CIFAR-100:

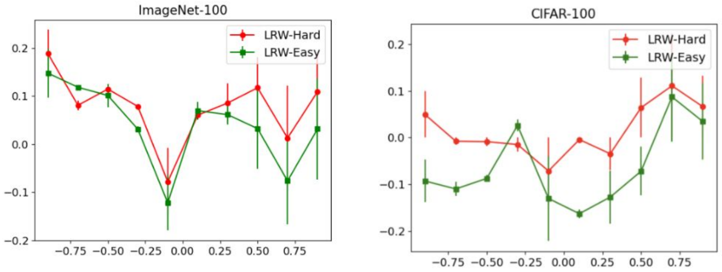
class skew	200	50	10	1
MWN [34]	40.11 ± 0.9	48.67 ± 0.7	61.32 ± 0.6	74.23 ± 0.3
FSR [43]	38.04 ± 0.8	45.12 ± 0.9	58.38 ± 0.6	74.68 ± 0.2
GDW [6]	40.36 ± 1.0	48.89 ± 0.8	61.67 ± 0.5	74.41 ± 0.4
LRWOpt	$\mathbf{42.33} \pm 0.8$	$\mathbf{50.77} \pm 0.7$	$\mathbf{63.28} \pm 0.8$	$\mathbf{75.12} \pm 0.3$

Margin Maximization Via Meta-Learning

Pairwise Margin Delta between MOLERE and ERM: right-skewed with mean/median > 0



Better Margin gain over ERM of the LRW-hard heuristic as compared to the LRM-Easy (as a function of ERM margin)



Thank You!