Reinforced Sample Re-weighting for Pedestrian Attribute Recognition

Yuan Liu
School of EEE
Nanyang Technological University
Singapore
yliu050@e.ntu.edu.sg

Zhiping Lin
School of EEE
Nanyang Technological University
Singapore
ezplin@ntu.edu.sg

***Abstract*—Pedestrian Attribute Recognition aims to recognise person attributes including age, gender, clothing and accessories in a given image. It is challenging due to the high variance in sample quality and attribute content. Many network structures have been proposed to better capture the fine-grained details in an image. However, how to learn from each training sample based on their importance to the model still remains to be addressed. In this paper, we propose Reinforced Sample Re-weighting (RSR), a novel approach to re-weight samples in a batch during back-propagation through reinforcement learning. RSR agents are proposed to assign sample weights based on both the sample itself and the recognition model status. The agent learns in an on-policy manner, where it learns together with the attribute recognition model and no additional training is required. The proposed approach achieves state-of-the-art performance against other existing methods on three large scale pedestrian attribute datasets PETA, PA-100K and RAP, which demonstrates the effectiveness of our method.**

***Index Terms*—Pedestrian Attribute Recognition, Reinforcement Learning, Computer Vision**

I. Introduction

Pedestrian Attribute Recognition aims to accurately identify person attributes including age, gender, clothing and accessories from a given image sample. It has attracted significant research interest due to its potential application in surveillance and behavioural analysis. With the recent advancement of Deep Learning and the release of large scale datasets such as PETA [1], PA-100K [2], RAP [3] and PARSE27K [4], many approaches have been proposed including DeepMAR [5], HP-Net [2] and RCRA [6] to tackle this task.

As shown in Figure 1, pedestrian attribute datasets normally contain high variance in image quality. In addition, the number of attributes in each image also differs significantly. Therefore, the knowledge a model can learn from each image sample is different. Intuitively, difficult cases with occlusion or rare attributes and high quality samples with rich attribute content might be more important than normal images as they could help the model to improve its robustness and better understand the feature detail. However, existing methods typically adopt an averaging strategy when calculating the loss from a batch of samples, where each sample is treated equally regardless of the information it carries.

In this work, we propose a novel approach called "Reinforced Sample Re-weighting" (RSR), where RSR agents assign loss weights for samples in a batch during back-propagation. As a result, the attribute recognition model is able to focus on more informative samples and improve its performance. The RSR agent updates itself in an on-policy manner, where it can learn to make the best re-weighting action according to both the image samples and the recognition model status. No additional training for the agent is needed prior to the attribute recognition task. To the best of our knowledge, this is the first method that re-weights pedestrian image samples based on their importance to the recognition model.

Fig. 1. High sample quality variance in pedestrian attribute dataset. Corner-cases, normal quality and high quality from left to right.

The main contribution of our work is three-fold. Firstly, we propose Reinforced Sample Re-weighting (RSR) with Reinforcement Learning agent that re-weights image samples during training and maximises the benefit of the model from these samples. Secondly, we introduce an on-policy agent update method that enables the agent to learn efficiently without additional training iterations. Thirdly, we conduct extensive experiments on three large-scale datasets PETA, PA-100K and RAP to illustrate the advantage of the proposed RSR.

II. Related Work

A. Pedestrian Attribute Recognition

Pedestrian Attribute Recognition has been studied extensively over the past few years. Earlier researches [7] [8] [9] mainly focused on hand-crafted features such as HOG and colour histograms with traditional machine learning algorithms including Support Vector Machine and AdaBoost to perform recognition.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

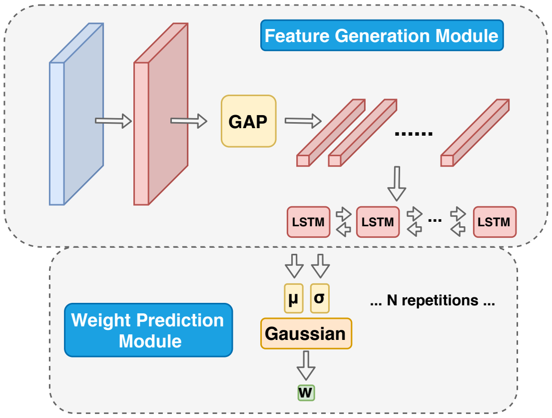

Fig. 2. Structure of the RSR agent. The feature map in blue represents the backbone stage 3 output. GAP represents global average pooling. The weight score w sampled from the predicted Gaussian is obtained for all N samples in a batch.

With the development of Deep Learning, features extracted using Deep Convolutional Neural Networks (CNN) have been commonly adopted to learn better image representations. DeepMAR [5] applied CaffeNet and introduced a new loss function to tackle attribute unbalance problem. Spatial pyramid pooling was adopted in [10] to localise and recognise attribute simultaneously. In addition, many part-based models [11] [12] [13] [14] [15] have been introduced to mine the fine-grained details in local regions via parsing, attention mechanism and feature map partitioning. Prior knowledge has also been explored in [16] via a multi-branch network to model attribute dependencies. To better capture the relations among attributes, Recurrent Neural Network (RNN) models [17] [6] [18] have been adopted to predict attributes in sequence. When combining with attention mechanism, these RNN based models are able to focus on different regions to jointly predict attributes. However, these RNN based methods' performances heavily rely on the prediction sequence, which is difficult to optimise. Recently, some graphical methods are introduced to better explore the attribute correlations without the constraint of sequence. The work in [19] calculated the pair-wise conditional probabilities and predicted attribute confidence with Support Vector Machine. An And-Or structured graph was also proposed in [20] to mine the attribute dependencies. The work in VSGR [21] applied Graph Convolution Network to mine spatial and semantic information simultaneously. However, the above-mentioned models only focused on better representing the information within each input sample, while the relative importance of each sample for model's learning was not considered. Our proposed Reinforced Sample Re-weighting is able to assist recognition model to focus on the more informative and beneficial samples, which improves its performance in a simple yet effective manner.

B. *Reinforcement Learning and Sample Re-weighting*

Reinforcement Learning (RL) enables an agent to perform the best action given a state in an environment. It has attracted significant amount of research attention due to the potential applications such as gaming, robotics and Industrial 4.0 [22]. During each training step, the agent observes the current state of the environment, predicts an action according to a learned policy and receives the reward value for the action. The training objective is to optimise its policy such that the reward can be maximised.

One effective approach to train the RL agent is Policy Gradient, in which the REINFORCE algorithm [23] is a typical example. It teaches the agent to perform the best trajectory over a sequence of actions. In our work, the weights of each samples in a batch are modeled as a trajectory and Policy Gradient based method is adopted to train the RL agent.

Re-weighting each samples during training has been studied previously on image classification tasks. Methods that emphasise hard examples [24] [25] or easy example [26] have been proposed to improve model's robustness. However, for applications with high sample quality variance such as pedestrian attribute recognition, solely emphasising hard or easy examples might reduce model's overall learning stability. The work in [27] proposed to re-weight samples based on gradient directions when minimising loss on a high-quality validation set. However, its performance might be constrained when encountering applications with limited number of validation examples. Notably, the above-mentioned methods are not able to adjust their re-weighting strategy according to the application scenario, such as high precision or high recall. In our proposed RSR, no additional validation set is needed and the re-weighting policy can be adjusted flexibly based on the actual application needs.

III. Proposed Method

A. *Reinforced Sample Re-weighting Agent*

The structure of the RSR agent is shown in Figure 2. It consists of two main components, namely the Feature Generation Module (FGM) and the Weight Prediction Module (WPM). The FGM generates features for weight prediction of each sample in a batch. The WPM takes these features and predicts their weight scores accordingly. Let N be the batch size and l_i represents the loss from the i^{th} sample in a batch. For normal training of recognition model without the agent, losses from individual samples are averaged as shown in (1).

$$L_{normal} = \frac{1}{N}\sum_{i}^{N} l_i, \tag{1}$$

When training with the agent, the weight score w are used to perform weighted sum on the individual losses as shown in (2), which enables the recognition model to make the best use of each training image and improve performance.

$$\begin{cases} L_{re-weight} = \sum_i^N w_i l_i, \\ \sum_i^N w_i = 1, \end{cases} \tag{2}$$

where w_i is the weight score of the i^{th} sample.

To generate features with FGM, the raw feature maps from ResNet-50 backbone stage 3 are adopted to represent both the

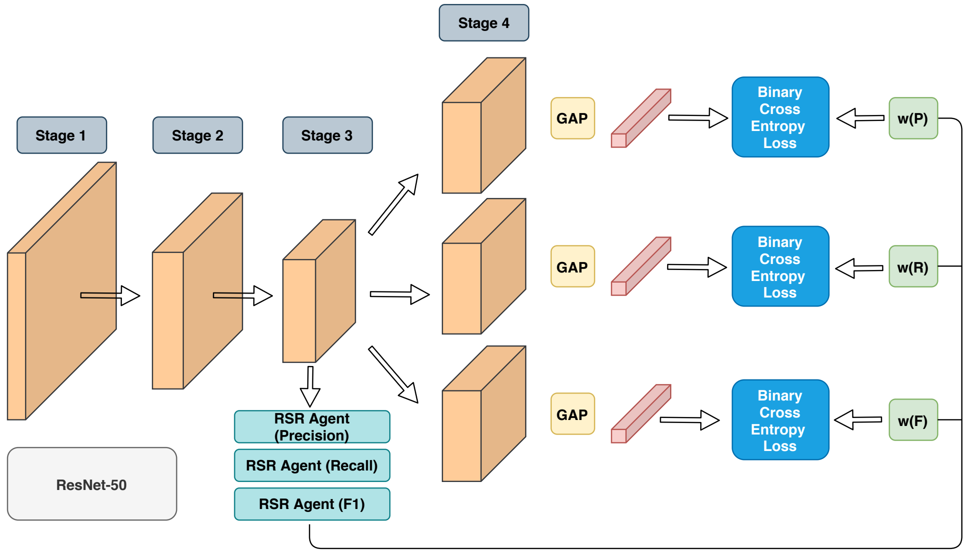

Fig. 3. The structure of our network. ResNet-50 is adopted as the backbone feature extractor, where its last stage is duplicated to form three branches with a RSR agent each. The three RSR agents are assigned with different learning objectives, P, R and F represents Precision, Recall and F1 score, respectively. GAP represents global average pooling and w represents the predicted weight scores for input samples.

individual image features and the current recognition model status. These feature maps are projected with convolution followed by pooling and reshaping to form feature vectors for all samples. To further model the "relative importance" of each sample in a batch, we feed these feature vectors into a bi-directional LSTM layer to produce the final features for weight prediction.

Directly predicting weight scores and updating WPM parameters by minimising weighted recognition loss might cause agent's bias towards simple examples. In addition, the definition of "informative sample" varies under different learning objectives such as high precision and high recall. Therefore, a more effective approach is needed to predict the weight scores based on the learning objective. Hence, we adopt Gaussian Policy Gradient [28] in WPM to predict continuous weight scores. Specifically, for a given feature, two fully-connected layers are adopted to predict the mean and log standard deviation which describe a Gaussian distribution that the weight score can be sampled from. The procedure is formulated as follows,

$$\mu = F_m(x; \theta_\mu), \tag{3}$$

$$\sigma = e^{F_{sd}(x; \theta_\sigma)}, \tag{4}$$

$$P(a) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{a-\mu}{\sigma}\right)^2}, \tag{5}$$

where F_m is a fully connected layer to predict mean with parameter θ_μ, F_{sd} is a fully connected layer to predict log standard deviation with parameter θ_σ. μ and σ represents the predicted mean and standard deviation, $P(a)$ is the probability density given a sampled action a.

For convenience, we denote the parameters of the entire RSR agent as θ. Hence, the probability of generating the batch weight scores for samples x is,

$$\pi(a_{1:N} | x, \theta) = \prod_{i=1}^{N} U(P(a_i)) \tag{6}$$

Note that we use the probability density to represent the actual probability as they are positively correlated. However, the density value can be larger than 1 and consecutive multiplication of these values might result in gradient explosion when back-propagating. Hence, we re-scale the value of $P(a)$ and constrain its upper bound by U, which is implemented as a Sigmoid operation that sets an upper bound of 1.

Through iteration of the agent with a given learning objective, the standard deviation σ will decrease and the value predicted for each sample will stabilise towards its mean μ. In this manner, the agent is able to learn its policy flexibly based on the learning objective such as high precision, high recall or high F1 score, which will be further illustrated in the next section. Note that a softmax operation is performed for weight scores of a batch to ensure they sum up to 1.

B. On-Policy Agent Update and Diversely Focused Attribute Modelling

A reward function is typically required in policy gradient methods to evaluate the agent's action as well as performing policy update. Intuitively, if the RSR agent's weight prediction is reasonable, the model should yield much better performance on the same batch samples after back-propagation. Hence, given a performance metric, the reward for RSR agent's prediction is designed as shown in (7).

$$R = M(A(x|\psi'), y) - M(A(x|\psi), y), \tag{7}$$

where M represents a performance metrics, x represents a batch of samples, y represents the ground truth attribute labels, $A(x|\psi)$ and $A(x|\psi^{'})$ represent the attribute recognition model with the parameters before back-propagation ψ and after back-propagation $\psi^{'}$. We take this reward as the RSR agent's learning objective. Maximising the reward forces the agent to adjust its policy and focus more on the important samples based on the learning objective.

To update the agent based on the reward, we adopt the REINFORCE algorithm [23] to maximise the expected reward $J(\theta)$. Its gradient with respective to the RSR agent parameter θ can be approximated via Monte Carlo method as follows,

$$\nabla_{\theta} J(\theta) = \frac{1}{E} \sum_{i=1}^{E} \nabla_{\theta} log \pi(a_{1:N}^{i}|x_t;\theta) R(x_t, a_{1:N}^{i}), \quad (8)$$

where E is the number of iterations before each agent update and $R(x_t, a_{1:N}^{i})$ represents the reward at t^{th} iteration.

The Precision, Recall and F1 in multi-label recognition are defined in [29] as follows,

$$Precision = \frac{1}{p} \sum_{i=1}^{p} \frac{|y_i \cap A(x_i)|}{|A(x_i)|}, \quad (9)$$

$$Recall = \frac{1}{p} \sum_{i=1}^{p} \frac{|y_i \cap A(x_i)|}{|y_i|}, \quad (10)$$

$$F1 = \frac{2 * Precision * Recall}{Precison + Recall}, \quad (11)$$

where p is the number of samples for evaluation, x_i and y_i are the i^{th} sample and its labels. In general scenarios, F1 score is more interesting as it finds a balance between Precision and Recall. However, as the agent will capture different details when updating under different learning objective, merely applying F1 score might constrain the model from exploring rich information during training. Therefore, we propose Diversely Focused Attribute Modelling, where the last stage of the ResNet-50 [30] backbone is duplicated twice to form three branches as shown in Figure 3. Precision, Recall and F1 scores are adopted with a RSR agent for each of the three branches, correspondingly. The overall loss for the attribute recognition model can be formulated as follows,

$$L_{total} = \sum_{j} \sum_{i}^{N} w_{ji} l_i, \quad (12)$$

where j represents one of the three branches and w_{ji} represents the weight score for i^{th} sample in a batch predicted by the RSR agent of the j^{th} branch.

C. Training and Inference

The whole model is trained in an end-to-end manner. As the attribute recognition model itself is relatively unstable at the beginning of training, we only apply RSR after T_{stage1} model iterations and continue RSR update until the convergence of recognition model at T_{stage2}. The training procedure can be summarised in Algorithm 1 and 2. During inference, we simply take the element wise maximum among the three branch predictions. Note that the RSR agents are not utilised during inference as they are only facilitating the training process.

Algorithm 1: Training Algorithm Stage 1

```
Training stage 1: Initialise attribute recognition model
  parameters ψ, learning rate α and stage 1 iterations
  T_stage1;
while Batch Index <= T_stage1 do
    Sample a batch of images;
    Calculate model loss L_normal via 1;
    Update ψ via: ψ = ψ + α∇_ψ L_normal;
end
```

Algorithm 2: Training Algorithm Stage 2

```
Training stage 2: Initialise RSR agent for each branch
  with parameter θ, agent learning rate β, agent update
  frequency E and stage 2 iterations T_stage2 ;
while Batch Index <= T_stage2 do
    Sample a batch of images;
    for each branch do
        Predict weight scores using RSR agent;
        Calculate model loss L_re-weight via 2;
        Store RSR agent weight scores probability
          π(a_1:N|x, θ);
    end
    Sum the individual branch loss to obtain L_total;
    Update ψ via: ψ = ψ + α∇_ψ L_total;
    if Batch Index is a multiple of E then
        for each RSR agent do
            Calculate gradient ∇_θ J(θ) with 8;
            Update θ via: θ = θ + β∇_θ∇_θ J(θ);
        end
    end
end
```

IV. Experiment

A. Datasets

PEdesTrian Attribute (PETA) contains 19000 images with 65 annotated attribute labels. Only 35 of the binary labels are selected to perform the attribute recognition task. Same as the widely adopted protocol, we split the dataset into three non-overlapping partitions for training, validation and testing, which contain 9500, 1900 and 7600 images, respectively.

PA-100K is the largest pedestrian attribute dataset currently with 100000 images. Each image in the dataset is annotated with 26 binary attribute labels. The dataset is split into 80000, 10000 and 10000 images for training, validation and testing respectively.

Richly Annotated Pedestrian (RAP) dataset contains 41,585 pedestrian images with 72 attributes. The dataset is split into two partitions with 33,268 images for training and 8,317 images for testing. 51 binary attributes are selected for

TABLE I
COMPARISON WITH STATE-OF-THE-ART METHODS ON PA-100K, RAP AND PETA DATASETS. THE TOP PERFORMANCE OF EACH METRIC IS IN **BOLD** AND THE SECOND BEST PERFORMANCE OF EACH METRIC IS UNDERLINED. PRECISION AND RECALL ARE LESS INDICATIVE AS THEY ARE NEGATIVELY CORRELATED.

	PA-100K					RAP					PETA				
Method	Acc	Pre	Rec	F1	mA	Acc	Pre	Rec	F1	mA	Acc	Pre	Rec	F1	mA
ACN	-	-	-	-	-	62.61	80.12	72.26	75.98	81.15	73.66	84.06	81.26	82.64	81.15
JRL	-	-	-	-	-	-	75.08	74.96	74.62	74.74	-	82.55	82.12	82.02	82.13
PG-Net	73.08	84.36	82.24	83.29	74.95	64.57	78.86	75.90	77.35	74.31	78.08	86.86	84.68	85.76	82.97
DeepMAR	70.39	82.24	80.42	81.32	72.70	-	-	-	-	-	75.07	83.68	83.14	83.41	82.60
GRL	-	-	-	-	-	-	77.70	80.90	79.29	81.20	-	84.34	88.82	86.51	86.70
HP-Net	72.19	82.97	82.09	82.53	74.21	65.39	77.33	78.79	78.05	76.12	76.13	84.92	83.24	84.07	81.77
VAA	-	-	-	-	-	-	-	-	-	-	78.56	86.79	86.12	86.46	84.59
LG-Net	75.55	86.99	83.17	85.04	76.96	68.00	80.36	79.82	80.09	78.68	-	-	-	-	
RCRA(RC)	-	-	-	-	-	-	82.67	76.65	79.54	78.47	-	85.42	88.02	86.70	85.78
RCRA(RA)	-	-	-	-	-	-	79.45	79.23	79.34	81.16	-	84.69	88.51	86.56	86.11
CoCNN	78.30	**89.49**	84.36	86.85	80.56	68.37	**81.04**	80.27	**80.65**	81.42	79.95	**87.58**	87.73	**87.65**	**86.97**
ALM	77.08	84.21	88.84	86.46	80.68	68.17	74.71	**86.48**	80.16	81.87	79.52	85.65	88.09	86.85	86.30
PAA	78.89	86.83	87.73	87.27	81.61	67.91	78.56	81.45	79.98	81.25	79.46	87.42	86.33	86.87	84.88
RSR(Ours)	**79.21**	86.23	**89.07**	**87.62**	**81.85**	**68.62**	77.39	84.20	**80.65**	**81.69**	**80.23**	86.13	**88.98**	87.53	85.92

the attribute recognition task aligning with the widely adopted protocol.

B. Evaluation Criteria

We use five performance metrics to evaluate the performance of the attribute recognition model accordingly to the widely adopted protocol. Accuracy (Acc), Precision (Pre), Recall (Rec) and F1 score: The average accuracy, precision, recall and F1 score over all testing samples, which are also named as instance-based metrics as they represents the performance on each individual testing image. Mean Accuracy (mA): The average of attribute-wise positive and negative classification accuracies, which is also named as label-based metric as it focuses on the accuracy of each attribute accross the entire testing dataset. Note that Precision and Recall are negatively correlated and are hence less indicative.

C. Implementation Details

We resize all input images to 224×448 pixels. Data augmentation techniques including horizontal flipping and random cropping are adopted to enrich training data variations. Adam optimiser [31] is applied for both the recognition model and the RSR agent with an initial learning rate of 0.0001. We firstly train the attribute recognition model without RSR agent for $T_{stage1} = 3000$ iterations, then add in the RSR agent and train the model until converge at T_{stage2}. Specifically, $T_{stage2_{PETA}} = 25000$, $T_{stage2_{PA100K}} = 20000$ and $T_{stage2_{RAP}} = 15000$. Learning rate decay of 0.5 is applied for both the model and the agent for every 5000 iterations. The training batch size is set as 32. The network is implemented with PyTorch framework and trained with Nvidia V100 graphic cards.

D. Comparison with State-of-the-Arts

We compare the proposed Reinforced Sample Re-weighting method against 13 competitors ACN [32], JRL [18], PG-Net [12], DeepMAR [5], GRL [17], HP-Net [2], VAA [33], LG-Net [13], RCRA (RC and RA) [6], CoCNN [16], ALM [14] and PAA [15] on the three datasets. As shown in Table I, the proposed RSR method outperforms all the previous state-of-the-art methods on PA-100K and RAP, while maintaining a highly competitive performance on PETA. Unlike most of the competitors that apply carefully designed network structures or prior knowledge such as Encoder-Decoder Networks, Attention Mechanism and Conditional Probabilities, our method can achieve high performance with only a fully-connected layer on top of each branch's feature vector. It is notable that the PETA dataset is relatively small and the identity-level attribute labels are inaccurate, which might limit RSR's performance to a certain extend.

TABLE II
EFFECT OF RSR ON PA-100K, RAP AND PETA

PA-100K

Method	Ins Acc	Ins Pre	Ins Rec	Ins F1	Label Acc
Baseline	78.73	85.56	**89.21**	87.35	81.55
RSR	**79.21**	**86.23**	89.07	**87.62**	**81.85**

RAP

Method	Ins Acc	Ins Pre	Ins Rec	Ins F1	Label Acc
Baseline	67.95	76.31	**84.48**	80.19	81.64
RSR	**68.62**	**77.39**	84.20	**80.65**	**81.69**

PETA

Method	Ins Acc	Ins Pre	Ins Rec	Ins F1	Label Acc
Baseline	79.62	85.29	**89.17**	87.18	85.55
RSR	**80.23**	**86.13**	88.98	**87.52**	**85.92**

E. Ablation Study

1) Effect of Reinforced Sample Re-weighting: In order to showcase the effectiveness of RSR quantitatively, we train a

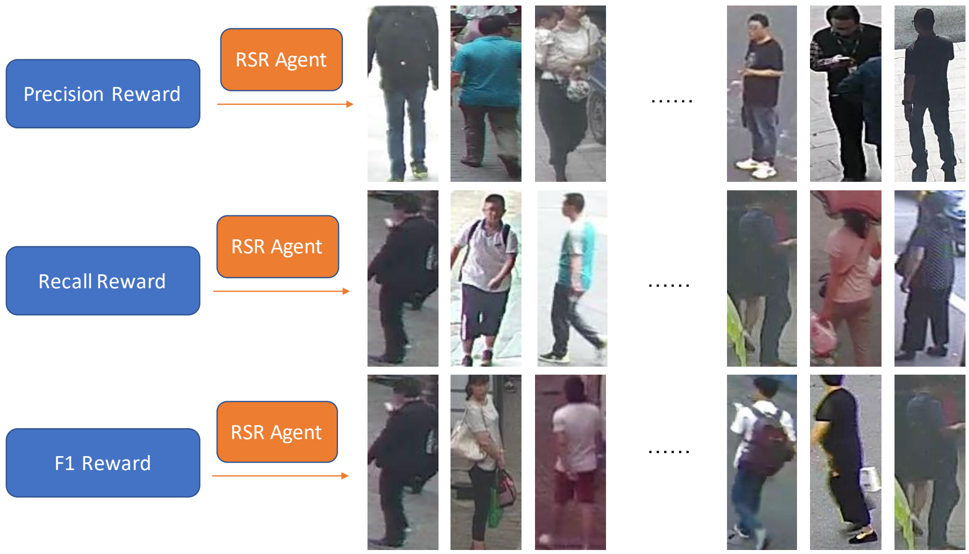

Fig. 4. A batch of samples ranked by the RSR agents trained with Precision, Recall and F1 score learning objectives.

separate three-branch structured model without RSR as the baseline. As shown in Table II, adding in RSR improves the baseline significantly on all three datasets, which demonstrates RSR's capability to produce more accurate recognition result. In addition, we also visualise the samples ranked by their weight scores for a batch of training inputs predicted by the three RSR agents. In Figure 4, the agent trained with precision objective tends to focus more on better quality samples while suppressing hard-cases, which improves recognition model's ability to precisely capture the attributes presented. On the other hand, the agent trained with recall objective allocates higher weights for samples with occlusion and poor lighting condition, which avoids the recognition model from missing attributes for corner-cases. The agent trained with F1 reward focuses on both good and poor quality samples, which helps the recognition model to maintain a balance between precision and recall.

2) Effect of Diversely Focused Attribute Modelling: To elaborate the advantage of Diversely Focused Attribute Modelling, we firstly train a recognition model with only F1 reward RSR agent as the benchmark. We then add the Precision and Recall RSR agents one by one on top of the benchmark to show the performance gain when incorporating each branch. As shown in Table III, the overall performance improves significantly when adding each RSR agent. One interesting observation is when adding more RSR agents, the overall Recall score increases, which is reasonable as the predictions from all branches are fused during inference. It is notable that although Precision and Recall are negatively correlated, the significant improvement in Recall does not result in a drastically decreased Precision which still yield a steady increment in F1 that shows the robustness of our method.

As Diversely Focused Attribute Modelling contains 3 sets of the ResNet-50 final stage, we perform a complexity study on its inference time. Specifically, we compare the time in

TABLE III
EFFECT OF DIVERSELY FOCUSED ATTRIBUTE MODELLING ON PA-100K, RAP AND PETA

PA-100K

Method	Ins Acc	Ins Pre	Ins Rec	Ins F1	Label Acc
F1	78.68	**87.38**	86.94	87.16	81.41
F1+Rec	78.84	86.29	88.46	87.36	81.26
RSR	**79.21**	86.23	**89.07**	**87.62**	**81.85**

RAP

Method	Ins Acc	Ins Pre	Ins Rec	Ins F1	Label Acc
F1	68.28	**80.17**	80.33	80.25	80.04
F1+Rec	68.28	78.28	82.50	80.33	80.94
RSR	**68.62**	77.39	**84.20**	**80.65**	**81.69**

PETA

Method	Ins Acc	Ins Pre	Ins Rec	Ins F1	Label Acc
F1	78.87	**88.05**	85.00	86.50	84.29
F1+Rec	79.32	86.56	87.12	86.84	84.97
RSR	**80.23**	86.13	**88.98**	**87.52**	**85.92**

seconds to inference a single image using the entire RSR network versus only one branch. We compute the inference time for 10 times and take the average, the results are presented in Table IV. RSR only introduces 0.67% additional inference time while providing a significant performance boost, which validates its advantage and efficiency for the Pedestrian Attribute Recognition task.

TABLE IV
COMPLEXITY STUDY ON INFERENCE TIME

Method	Average Inference Time
Single Branch	1.1350 s
RSR	1.1426 s

3) Comparison with Other Re-weighting Method: Sample re-weighting has been studied in image classification tasks, especially for biased dataset or noisy labels. However, most of these applications are designed for single label recognition datasets such as MNIST [34] and CIFAR [35]. For pedestrian attribute recognition where a large number of labels is presented in a given sample, the performance of those existing methods might be affected.

TABLE V
COMPARISON OF RSR AND GRADIENT-BASED METHOD ON PETA

Method	Ins Acc	Ins Pre	Ins Rec	Ins F1	Label Acc
Gradient-based	42.31	67.72	51.24	58.34	52.18
RSR	**75.67**	**80.69**	**87.86**	**84.12**	**83.11**

We re-implement the recently proposed gradient-based re-weighting method [27] and integrate the re-weighting module into our network. We keep the same three branch structure for the network for fair comparison with RSR. As the gradient-based method requires multiple forward and backward passes with a held-out clean validation set during training, using relatively large input size results in out-of-memory error. Hence, we resize all images to 64×64 to ensure a smooth training process. Note that we use 500 image samples as the held-out validation set and applied the same learning rate and decay as RSR. We present the results on PETA dataset in Table V. The gradient-based method fails to converge while RSR maintains a stable result. By manually inspecting the weights allocated by the gradient-based method, we find that most of the samples are considered unimportant as their weights are predicted to be negative which are then clamped to be 0. The underlining assumption of the gradient-based method requires useful samples to be similar to the held-out validation set, which could be difficult to hold for pedestrian attribute annotations where the variance is high among different samples.

V. CONCLUSION

In this paper, we propose a novel Reinforced Sample Re-weighting (RSR) method that maximises the gain from training samples for pedestrian attribute recognition model. RSR agents with diverse learning objectives are applied to predict loss weight scores for input images, which enables the recognition model to make the best use of each sample in the training data. The RSR agent learns on-policy while training the recognition model and no additional training iterations are required. The proposed RSR method achieves state-of-the-art performance on three large-scale PETA, PA-100K and RAP, which illustrates its advantages over existing methods.

REFERENCES

[1] Y. Deng, P. Luo, C. C. Loy, and X. Tang, "Pedestrian attribute recognition at far distance," *Proceedings of the ACM International Conference on Multimedia - MM 14*, 2014.

[2] X. Liu, H. Zhao, M. Tian, L. Sheng, J. Shao, S. Yi, J. Yan, and X. Wang, "Hydraplus-net: Attentive deep features for pedestrian analysis," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[3] D. Li, Z. Zhang, X. Chen, H. Ling, and K. Huang, "A richly annotated dataset for pedestrian attribute recognition," *ArXiv*, vol. abs/1603.07054, 2016.

[4] P. Sudowe, H. Spitzer, and B. Leibe, "Person Attribute Recognition with a Jointly-trained Holistic CNN Model," in *ICCV'15 ChaLearn Looking at People Workshop*, 2015.

[5] D. Li, X. Chen, and K. Huang, "Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios," *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, 2015.

[6] X. Zhao, L. Sang, G. Ding, J. Han, N. Di, and C. Yan, "Recurrent attention model for pedestrian attribute recognition," *AAAI*, 2019.

[7] J. Zhu, S. Liao, Z. Lei, D. Yi, and S. Z. Li, "Pedestrian attribute classification in surveillance: Database and evaluation," *2013 IEEE International Conference on Computer Vision Workshops*, 2013.

[8] R. Layne, T. M. Hospedales, and S. Gong, "Attributes-based re-identification," *Person Re-Identification*, p. 93–117, 2014.

[9] L. Bourdev, S. Maji, and J. Malik, "Describing people: A poselet-based approach to attribute classification," *2011 International Conference on Computer Vision*, 2011.

[10] Y. Zhou, K. Yu, B. Leng, Z. Zhang, D. Li, and K. Huang, "Weakly-supervised learning of mid-level features for pedestrian attribute recognition and localization," *Procedings of the British Machine Vision Conference 2017*, 2017.

[11] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. D. Bourdev, "Panda: Pose aligned networks for deep attribute modeling," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1637–1644, 2013.

[12] D. Li, X. Chen, Z. Zhang, and K. Huang, "Pose guided deep model for pedestrian attribute recognition in surveillance scenarios," *2018 IEEE International Conference on Multimedia and Expo (ICME)*, 2018.

[13] P. Liu, X. Liu, J. Yan, and J. Shao, "Localization guided learning for pedestrian attribute recognition," in *BMVC*, 2018.

[14] C. feng Tang, L. Sheng, Z. Zhang, and X. Hu, "Improving pedestrian attribute recognition with weakly-supervised multi-scale attribute-specific localization," *ArXiv*, vol. abs/1910.04562, 2019.

[15] Z. Tan, Y. Yang, J. Wan, H. Hang, G. Guo, and S. Z. Li, "Attention-based pedestrian attribute analysis," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 6126–6140, Dec 2019.

[16] K. Han, Y. Wang, H. Shu, C. Liu, C. Xu, and C. Xu, "Attribute aware pooling for pedestrian attribute recognition," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 2456–2462. [Online]. Available: https://doi.org/10.24963/ijcai.2019/341

[17] X. Zhao, L. Sang, G. Ding, Y. Guo, and X. Jin, "Grouping attribute recognition for pedestrian with joint recurrent learning," *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018.

[18] J. Wang, X. Zhu, S. Gong, and W. Li, "Attribute recognition by joint recurrent learning of context and correlation," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[19] H. Chen, A. Gallagher, and B. Girod, "Describing clothing by semantic attributes," *Computer Vision – ECCV 2012 Lecture Notes in Computer Science*, p. 609–623, 2012.

[20] S. Park, B. X. Nie, and S.-C. Zhu, "Attribute and-or grammar for joint parsing of human pose, parts and attributes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 7, p. 1555–1569, 2018.

[21] Q. Li, X. Zhao, R. He, and K. Huang, "Visual-semantic graph reasoning for pedestrian attribute recognition," *AAAI*, 2019.

[22] Y. Li, "Deep reinforcement learning: An overview," *ArXiv*, vol. abs/1701.07274, 2017.

[23] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3, pp. 229–256, May 1992.

[24] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 89–96. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2011.6126229

[25] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017.

[26] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 1189–1197. [Online]. Available: http://papers.nips.cc/paper/3923-self-paced-learning-for-latent-variable-models.pdf
[27] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," *ArXiv*, vol. abs/1803.09050, 2018.
[28] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html
[29] M. Zhang and Z. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, Aug 2014.
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.
[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.
[32] P. Sudowe, H. Spitzer, and B. Leibe, "Person attribute recognition with a jointly-trained holistic cnn model," *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015.
[33] N. Sarafianos, X. Xu, and I. A. Kakadiaris, "Deep imbalanced attribute classification using visual attention aggregation," in *ECCV*, 2018.
[34] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[35] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.