Towards Robustness to Label Noise in Text Classification via Noise Modeling

Siddhant Garg*  
Amazon Alexa AI  
sidgarg@amazon.com

Goutham Ramakrishnan∗  
Health at Scale Corporation  
goutham7r@gmail.com

Varun Thumbe∗  
KLA Corporation  
thumbevarun@gmail.com

ABSTRACT

Large datasets in NLP tend to suffer from noisy labels due to erroneous automatic and human annotation procedures. We study the problem of text classification with label noise, and aim to capture this noise through an auxiliary noise model over the classifier. We first assign a probability score to each training sample of having a clean or noisy label, using a two-component beta mixture model fitted on the training losses at an early epoch. Using this, we jointly train the classifier and the noise model through a novel de-noising loss having two components: (i) cross-entropy of the noise model prediction with the input label, and (ii) cross-entropy of the classifier prediction with the input label, weighted by the probability of the sample having a clean label. Our empirical evaluation on two text classification tasks and two types of label noise: random and input-conditional, shows that our approach can improve classification accuracy, and prevent over-fitting to the noise.

CCS CONCEPTS

• Computing methodologies → Natural language processing

KEYWORDS

Label Noise; Noise Model; Robustness; Text Classification; NLP

ACM Reference Format:

Siddhant Garg, Goutham Ramakrishnan, and Varun Thumbe. 2021. Towards Robustness to Label Noise in Text Classification via Noise Modeling. In Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM ’21), November 1–5, 2021, Virtual Event, QLD, Australia. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459637.3482204

1 INTRODUCTION

Training modern ML models requires access to large accurately labeled datasets, which are difficult to obtain due to errors in automatic or human annotation techniques [20, 24]. Recent studies [22] have shown that neural models can over-fit on noisy labels and thereby not generalize well. Human annotations for language tasks have been popularly obtained from platforms like Amazon Mechanical Turk [8], resulting in noisy labels due to ambiguity of the correct label [21], annotation speed, human error, inexperience of annotator, etc. While learning with noisy labels has been extensively studied in computer vision [18, 19, 23], the corresponding progress in NLP has been limited. With the increasing size of NLP datasets, noisy labels are likely to affect several practical applications [1].

In this paper, we consider the problem of text classification, and capture the label noise through an auxiliary noise model (See Fig. 1). We leverage the finding of learning on clean labels being easier than on noisy labels [2], and first fit a two-component beta-mixture model (BMM) on the training losses from the classifier at an early epoch. Using this, we assign a probability score to every training sample of having a clean or noisy label. We then jointly train the classifier and the noise model by selectively guiding the former’s prediction for samples with high probability scores of having clean labels. More specifically, we propose a novel de-noising loss having two components: (i) cross-entropy of the noise model prediction with the input label and (ii) cross-entropy of the classifier prediction with the input label, weighted by the probability of the sample having a clean label. Our formulation constrains the noise model to learn the label noise, and the classifier to learn a good representation for the prediction task from the clean samples. At inference time, we remove the noise model and use the predictions from the classifier.

Most existing works on learning with noisy labels assume that the label noise is independent of the input and only conditional on the true label. Text annotation complexity has been shown to depend on the lexical, syntactic and semantic input features [12] and not solely on the true label. The noise model in our formulation can capture an arbitrary noise function, which may depend on both the input and the original label, taking as input a contextualized input representation from the classifier. While de-noising the classifier for sophisticated noise functions is a challenging problem, we take the first step towards capturing a real world setting.

We evaluate our approach on two popular datasets, for two different types of label noise: random and input-conditional; at different...
noise levels. Across two model architectures, our approach results in improved model accuracies over the baseline, while preventing over-fitting to the label noise.

2 RELATED WORK

There have been several research works that have studied the problem of combating label noise in computer vision [4, 9, 10] through techniques like bootstrapping [18], mixup [23], etc. Applying techniques like mixup (convex combinations of pairs of samples) for textual inputs is challenging due to the discrete nature of the input space and retaining overall semantics. In natural language processing, Agarwal et al. [1] study the effect of different kinds of noise on text classification, Ardehaly and Culotta [3] study social media text classification using label proportion (LLP) models, and Malik and Bhardwaj [16] automatically validate noisy labels using high-quality class labels. Jindal et al. [11] capture random label noise via a \(\ell_2\)-regularized matrix learned on the classifier logits. Our work differs from this as we i) use a neural network noise model over contextualized embeddings from the classifier, with ii) a new denoising loss to explicitly guide learning. It is difficult to draw a distinction between noisy labels, and outliers which are hard to learn from. While several works perform outlier detection [5, 14] to discard these samples while learning the classifier, we utilise the noisy data in addition to the clean data for improving performance.

3 METHODOLOGY

Problem Setting Let \((X, Y^{(c)}:\{(x_1, y_1^{(c)}), \ldots, (x_N, y_N^{(c)})\}\) denote clean training samples from a distribution \(D=X\times Y\). We assume a function \(F:X\times Y \rightarrow Y\) that introduces noise in labels \(Y^{(c)}\). We apply \(F\) on \((X, Y^{(c)})\) to obtain the noisy training data \((X, Y^{(n)}) = \{(x_1, y_1^{(n)})\}, \ldots, (x_N, y_N^{(n)})\}\). \((X, Y^{(n)})\) contains a combination of clean samples (whose original label is retained \(y_1^{(n)}=y_1^{(c)}\)) and noisy samples (whose original label is corrupted \(y_1^{(n)} \neq y_1^{(c)}\)). Let \((X_T, Y_T)\) be a test set sampled from the clean distribution \(D\). Our goal is to learn a classifier model \(M: X \rightarrow Y\) trained on the noisy data \((X, Y^{(n)})\), which generalizes well on \((X_T, Y_T)\). Note that we do not have access to the clean labels \(Y^{(c)}\) at any point during training.

Modeling Noise Function \(F\) We propose to capture \(F\) using an auxiliary noise model \(N_M\) on top of the classifier model \(M\), as shown in Fig. 1. For an input \(x\), a representation \(R_M(x)\), derived from \(M\), is fed to \(N_M\). \(R_M(x)\) can typically be the contextualized input embedding from the penultimate layer of \(M\). We denote the predictions from \(M\) and \(N_M\) to be \(\hat{y}^{(c)}\) (clean prediction) and \(\hat{y}^{(n)}\) (noisy prediction) respectively. The clean prediction \(\hat{y}^{(c)}\) is used for inference.

3.1 Estimating clean/noisy label using BMM

It has been empirically observed that classifiers that capture input semantics do not fit the noise before significantly learning from the clean samples [2]. For a classifier trained using a cross entropy loss \(L_{\text{CE}}\) on the noisy dataset, this can be exploited to cluster the input samples as being clean/noisy in an unsupervised manner. Initially the training loss on both clean and noisy samples is large, and after a few training epochs, the loss of majority of the clean samples reduces. Since the loss of the noisy samples is still large, this segregates the samples into two clusters with different loss values. On further training, the model over-fits on the noisy samples and the training loss on both samples reduces. We illustrate this in Fig. 2(a)–(c). We fit a 2-component Beta mixture model (BMM) over the normalized training losses \(L_{\text{CE}}(\hat{y}^{(c)}, y) \in [0, 1]\) obtained after training the model for some warmup epochs \(T\). Using a Beta mixture model works better than using a Gaussian mixture model as it allows for asymmetric distributions and can capture the short tails of the clean sample losses. For a sample \((x, y)\) with normalized loss \(L_{\text{CE}}(\hat{y}^{(c)}, y) = \ell\), the BMM is given by:

\[
p(\ell) = \lambda_c \cdot p(\ell|\text{clean}) + \lambda_n \cdot p(\ell|\text{noisy})
\]

where \(\Gamma\) denotes the gamma distribution and \(\alpha_{c/n}, \beta_{c/n}\) are the parameters corresponding to the individual clean/noisy Beta distributions. The mixture coefficients \(\lambda_c\) and \(\lambda_n\), and parameters \((\alpha_{c/n}, \beta_{c/n})\) are learnt using the EM algorithm. On fitting the BMM \(B\), for a given input \(x\) with a normalized loss \(L_{\text{CE}}(\hat{y}^{(c)}, y) = \ell\), we denote the posterior probability of \(x\) having a clean label by:

\[
B(x) = \frac{\lambda_c \cdot p(\ell|\text{clean})}{\lambda_c \cdot p(\ell|\text{clean}) + \lambda_n \cdot p(\ell|\text{noisy})}
\]
We aim to train an algorithm for training using best Table 1: Results from experiments using random noise. Here for A(B): A refers to the Best model accuracy while B refers to (Last-Best) accuracy. The models with highest Best accuracies are in bold. For each noise %, the least and most reductions in Last accuracy are highlighted in green and red. Baseline (0% noise) reported beside dataset.

**Table 1:** Results from experiments using random noise. Here for A(B): A refers to the Best model accuracy while B refers to (Last-Best) accuracy. The models with highest Best accuracies are in bold. For each noise %, the least and most reductions in Last accuracy are highlighted in green and red. Baseline (0% noise) reported beside dataset.

| Model       | TREC (word-LSTM: 93.8, word-CNN: 92.6) | AG-News (word-LSTM: 92.5, word-CNN: 91.5) |
|-------------|-------------------------------------|----------------------------------------|
| Noise %     | 10 20 30 40 50 | 10 20 30 40 50 |
| word LSTM   | 88.0 (0.0) 89.4 (0.0) 83.4 (0.0) 79.6 (24.8) 77.6 (27.2) | 91.9 (1.7) 91.3 (1.5) 90.5 (2.5) 89.0 (2.5) 87.6 (10.5) |
| LSTM        | 92.2 (0.0) 90.2 (0.2) 88.8 (0.0) 83.5 (3.6) 82.4 (0.0) | 91.5 (0.1) 90.6 (0.1) 90.8 (0.1) 90.3 (0.0) 89.0 (0.1) |
| word CNN    | 88.8 (1.4) 89.2 (1.8) 84.8 (0.0) 82.2 (15.0) 77.6 (16.0) | 90.9 (2.7) 90.6 (4.2) 89.3 (10.2) 92.9 (17.9) 87.4 (25.2) |
| LSTM        | 91.0 (2.3) 90.8 (2.2) 89.4 (1.0) 81.4 (0.0) 81.4 (4.8) | 91.3 (2.2) 91.0 (4.0) 90.3 (0.2) 88.3 (3.2) 86.6 (3.2) |
| LSTM        | 92.2 (1.4) 91.8 (2.0) 88.8 (2.8) 77.0 (2.4) 77.2 (7.9) | 90.9 (0.0) 90.4 (0.1) 88.7 (1.1) 86.6 (3.5) 84.5 (10.2) |

**Algorithm 1** Training using $L_{DN-H}$

**Input:** Train data $(x_i, y_i^{(n)})$ N, warmup epochs $T_0$, total epochs $T$, parameter $\beta$, classifier $M$, noise model $N_M$

**for** epoch in $1, \ldots, T_0$

$\hat{y}_i^{(c)} \leftarrow M(x_i) \forall i \in [N]$

Train M with $\sum_i L_{CE}(\hat{y}_i^{(c)}, y_i^{(n)})$

**end for**

**for** epoch in $T_0 + 1, \ldots, T$

$\hat{y}_i^{(c)} \leftarrow M(x_i)$, $\hat{y}_i^{(n)} \leftarrow N_M(x_i)$ \forall i \in [N]

Train M, $N_M$ with $L_{DN-H} = \sum (L_{CE}(\hat{y}_i^{(n)}, y_i^{(n)}) + \beta \cdot [-1 \cdot B(x) > 0.5] \cdot L_{CE}(\hat{y}_i^{(c)}, y_i^{(n)})$

**end for**

Return: Trained classifier model $M$

---

### 4 EVALUATION

**Datasets**

We experiment with two popular text classification datasets: (i) TREC question-type dataset [15], and (ii) AG-News dataset [6] (Table 2). We inject noise in the training and validation sets, while retaining the original clean test set for evaluation. Note that collecting real datasets with known patterns of label noise is a challenging task, and out of the scope of this work. We artificially inject noise in clean datasets, which enables easy and extensive experimentation.

**Models**

We conduct experiments on two popular model architectures: word-LSTM [7] and word-CNN [13]. For word-LSTM, we use a 2-layer BiLSTM with hidden dimension of 150. For word-CNN, we use 300 kernel filters each of size 3, 4 and 5. We use the pre-trained GloVe embeddings [17] for initializing the word embeddings for both models. We train models on TREC and AG-News for 100 and 30 epochs respectively. We use an Adam optimizer with a learning rate of $10^{-3}$ and a dropout of 0.3 during training. For the noise model $N_\lambda$, we use a simple 2-layer feedforward neural network, with the number of hidden units $n_{hidden} = 4 \cdot n_{input}$. We choose the inputs to the noise model $R_M(x)$ as per the class of label noise, which we describe in Section 4.1 and 4.2. We conduct hyper-parameter tuning for the number of warmup epochs $T_0$ and $\beta$ using grid search over the ranges of $[6, 10, 20]$ and $[2, 4, 6, 8, 10]$ respectively.

**Metrics and Baseline**

We evaluate the robustness of the model to label noise on two fronts: (i) how well it performs on clean data, and (ii) how much it over-fits the noisy data. For the former, we report the test set accuracy (denoted by Best) corresponding to the model with best validation accuracy. For the latter, we examine the gap in test accuracies between the Best, and the Last model (after last training epoch). We evaluate our approach against only training $M$ (as the baseline), for two types of noise: random and input-conditional, at different noise levels.

**4.1 Results: Random Noise**

For a specific Noise %, we randomly change the original labels of this percentage of samples. Since the noise function is independent of the input, we use logits from $M$ as the input $R(x)$ to $N_M$. We report the Best and (Last - Best) test accuracies in Table 1. From the experiments, we observe that:
We heuristically condition the noise function $F$ on lexical and syntactic input features. We are the first to study such label noise for text inputs, to our knowledge. For both the TREC and AG-News, we condition $F$ on syntactic features of the input: (i) The TREC dataset contains different types of questions. We selectively corrupt the labels of inputs that contain the question words ‘How’ or ‘What’ (chosen based on occurrence frequency). For texts starting with ‘How’ or ‘What’, we inject random label noise (at different levels). (ii) The AG-News dataset contains news articles from different news agency sources. We inject random label noise for inputs containing the token ‘AP’, ‘Reuters’ or either one of them. We concatenate the contextualised input embedding from the penultimate layer of $M$ and the logits corresponding to $g_i(x)$ as the input $R_M(x)$ to $N_M$. We present the results in Tables 3 and 4.

On TREC, our method outperforms the baseline for both the noise patterns we consider. For the question-length based noise, we observe the same trend of $\mathcal{L}_{DN-S}$ outperforming $\mathcal{L}_{DN-H}$ at high noise levels, and vice-versa. On AG-News, the noise % for inputs having the specific tokens ‘AP’ and ‘Reuters’ are relatively low, and our method performs at par or marginally improves over the baseline performance. Interestingly, the input–conditional noise we consider $F$ on the text length (a lexical feature). More specifically, we inject random label noise for the longest $x$% inputs in the dataset. (ii) The AG-News dataset contains news articles from different news agency sources. We inject random label noise for inputs containing the token ‘AP’, ‘Reuters’ or either one of them. We concatenate the contextualised input embedding from the penultimate layer of $M$ and the logits corresponding to $g_i(x)$ as the input $R_M(x)$ to $N_M$. We present the results in Tables 3 and 4.

4.2 Results: Input-Conditional Noise

We heuristically condition the noise function $F$ on lexical and syntactic input features. We are the first to study such label noise for text inputs, to our knowledge. For both the TREC and AG-News, we condition $F$ on syntactic features of the input: (i) The TREC dataset contains different types of questions. We selectively corrupt the labels of inputs that contain the question words ‘How’ or ‘What’ (chosen based on occurrence frequency). For texts starting with ‘How’ or ‘What’, we inject random label noise (at different levels). We also consider $F$ conditional on the text length (a lexical feature). More specifically, we inject random label noise for the longest $x$% inputs in the dataset. (ii) The AG-News dataset contains news articles from different news agency sources. We inject random label noise for inputs containing the token ‘AP’, ‘Reuters’ or either one of them. We concatenate the contextualised input embedding from the penultimate layer of $M$ and the logits corresponding to $g_i(x)$ as the input $R_M(x)$ to $N_M$. We present the results in Tables 3 and 4.

On TREC, our method outperforms the baseline for both the noise patterns we consider. For the question-length based noise, we observe the same trend of $\mathcal{L}_{DN-S}$ outperforming $\mathcal{L}_{DN-H}$ at high noise levels, and vice-versa. On AG-News, the noise % for inputs having the specific tokens ‘AP’ and ‘Reuters’ are relatively low, and our method performs at par or marginally improves over the baseline performance. Interestingly, the input–conditional noise we consider $F$ on the text length (a lexical feature). More specifically, we inject random label noise for the longest $x$% inputs in the dataset. (ii) The AG-News dataset contains news articles from different news agency sources. We inject random label noise for inputs containing the token ‘AP’, ‘Reuters’ or either one of them. We concatenate the contextualised input embedding from the penultimate layer of $M$ and the logits corresponding to $g_i(x)$ as the input $R_M(x)$ to $N_M$. We present the results in Tables 3 and 4.

5 CONCLUSION

We have presented an approach to improve text classification when learning from noisy labels by jointly training a classifier and a noise model using a de-noising loss. We have evaluated our approach on two text classification tasks. demonstrate its effectiveness through an extensive evaluation. Future work includes studying more complex $F$ for other NLP tasks like language inference and QA.
REFERENCES

[1] Sumeet Agarwal, Shantanu Godbole, Shourya Roy, and Diwakar Punjani. 2007. How much noise is too much: A study in automatic text classification. In In Proc. of ICDM.

[2] Erik Arazo, Diego Ortego, Paul Albert, Noel O’Connor, and Kevin McGannness. 2019. Unsupervised Label Noise Modeling and Loss Correction. In Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97), Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR. Long Beach, California, USA, 312–321. http://proceedings.mlr.press/v97/ara19a.html

[3] Ehsan Ardehaly and Aron Culotta. 2018. Learning from noisy label proportions for classifying online social data. Social Network Analysis and Mining 8 (2018). https://doi.org/10.1007/s13278-017-0478-6

[4] Benoit Frénay and Michel Verleysen. 2014. Classification in the Presence of Label Noise: A Survey. IEEE Transactions on Neural Networks and Learning Systems 25 (2014). 845–869.

[5] James Goodman, Andreas Vlachos, and Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for Abstract Meaning Representation parsing. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Berlin, Germany. 1–11. https://doi.org/10.18653/v1/P16-1001

[6] A. Gulli. 2005. The Anatomy of a News Search Engine. In Special Interest Tracks and Posters of the 14th International Conference on World Wide Web (Chiba, Japan) (WWW ’05). Association for Computing Machinery, New York, USA. 860–881.

[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. Neural Comput. 9, 8 (Nov. 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[8] Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. 2010. Quality Management on Amazon Mechanical Turk. In Proceedings of the ACM SIGKDD Workshop on Human Computation (Washington DC) (HCOMP ’10). Association for Computing Machinery, New York, NY, USA. 64–67. https://doi.org/10.1145/1837885.1837906

[9] Junjun Jiang, Jiayi Ma, Zheng Wang, Chen Chen, and Xuming Liu. 2019. Hyperspectral Image Classification in the Presence of Noisy Labels. IEEE Transactions on Geoscience and Remote Sensing 57 (2019). 851–865.

[10] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. On the Importance of Label Quality for Semantic Segmentation. In Proceedings of the 36th International Conference on Machine Learning, Jennifer Dy and Andreas Krause (Eds.). PMLR. http://proceedings.mlr.press/v80/jiang18c.html

[11] Ishan Jindal, Daniel Preszel, Brian Lester, and Matthew Nokleby. 2019. An Effective Label Noise Model for DNN Text Classification. In Proceedings of North American Chapter of the Association of Computational Linguistics 2019.

[12] Aditya Joshi, Abhijit Mishra, Nivedan Senthimalselvan, and Pushpak Bhatcharyya. 2014. Measuring Sentiment Annotation Complexity of Text. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, Baltimore, Maryland.

[13] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014. 1746–1751.

[14] Stefan Larson, Anish Mahendran, Andrew Lee, Jonathan K. Kummerfeld, Parker Hill, Michael A. Laureznano, Johann Hauswald, Lingjia Tang, and Jason Mars. 2019. Outlier Detection for Improved Data Quality and Diversity in Dialog Systems. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota. 517–527. https://doi.org/10.18653/v1/N19-1031

[15] Xin Li and Dan Roth. 2002. Learning Question Classifiers. In Proceedings of the 19th International Conference on Computational Linguistics - Volume 1 (Taipei, Taiwan) (COLING ’02). Association for Computational Linguistics, Stroudsburg, PA, USA. 1–7.

[16] H. H. Malik and V. S. Bhardwaj. 2011. Automatic Training Data Cleaning for Text Classification. In 2011 IEEE 11th International Conference on Data Mining Workshops. 442–449.

[17] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In Empirical Methods in Natural Language Processing (EMNLP). 1532–1543. http://www.aclweb.org/anthology/D14-1162

[18] Scott E. Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2015. Training Deep Neural Networks on Noisy Labels with Bootstrapping. In ICLR (Workshop). Yoshua Bengio and Yann LeCun (Eds.) http://dblp.uni-trier.de/db/conf/iclr/iclr2015w/html/ReedLASER14

[19] Sunil Thulasidasan, Tamnoy Bhattacharya, Jeff Bilmes, Gopinath Chemnupati, and Jamal Mohd-Yusof. 2019. Combating Label Noise in Deep Learning using Abstention. In Proceedings of the 36th International Conference on Machine Learning, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR. http://proceedings.mlr.press/v97/thulasidasan19a.html

[20] Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. 2018. The Devil of Face Recognition is in the Noise. arXiv preprint arXiv:1807.11649 (2018).

[21] Xueying Zhan, Yuwei Wang, Yanghui Rao, and Qing Li. 2019. Learning from Multi-Annnotator Data: A Noise-Aware Classification Framework. ACM Trans. Inf. Syst. 37, 2, Article 26 (Feb. 2019), 28 pages. https://doi.org/10.1145/3309543

[22] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. ICLR (2016). http://arxiv.org/abs/1611.03530 cite arxiv:1611.03530 Comment: Published in ICLR 2017.

[23] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In International Conference on Learning Representations. https://openreview.net/forum?id=r1Ddp1-Rb

[24] Aleksandar Zlateski, Romnachai Jaronski, Prafull Sharma, and Fredo Durand. 2018. On the Importance of Label Quality for Semantic Segmentation. In CVPR. 1479–1487. https://doi.org/10.1109/CVPR.2018.00160