What Are Effective Labels for Augmented Data? Improving Calibration and Robustness with AutoLabel

Yao Qin [1] **Xuezhi Wang** [1] **Balaji Lakshminarayanan** [1] **Ed Chi** [1] **Alex Beutel** [1]

Abstract

A wide breadth of research has devised data augmentation approaches that can improve both accuracy and generalization performance for neural networks. However, augmented data can end up being far from the clean training data and what is the appropriate label is less clear. Despite this, most existing work simply uses one-hot labels for augmented data. In this paper, we show re-using one-hot labels for highly distorted data might run the risk of adding noise and degrading accuracy and calibration. To mitigate this, we propose a *generic* method `AutoLabel` to automatically learn the confidence in the labels for augmented data, based on the transformation distance between the clean distribution and augmented distribution. `AutoLabel` is built on label smoothing and is guided by the calibration-performance over a hold-out validation set. We successfully apply `AutoLabel` to the state-of-the-art RandAug and AugMix. Experiments on CIFAR-100 and ImageNet show that `AutoLabel` significantly improves existing data augmentation techniques over models' calibration and accuracy, especially under distributional shift.

1. Introduction

Deep neural networks are increasingly being used in high-stakes applications such as healthcare and autonomous driving. For safe deployment, we not only want models to be accurate on independent and identically distributed (i.i.d.) test cases, but we also want models to be robust to distribution shift (Amodei et al., 2016). Recent work has shown that the accuracy of state-of-the-art models drops significantly when tested on corrupted data (Hendrycks & Dietterich, 2019). Furthermore, these models do not just perform worse on these unexpected examples, but are also over-confident – Ovadia et al. (2019) showed that calibration of models degrades under shift. Calibration measures the gap between a model's own estimate of correctness (i.e., confidence) versus the empirical accuracy, which measures the actual probability of correctness. When a model is not well calibrated, particularly on unexpected examples, it undermines our ability to trust its predictions. Building models that are accurate *and* robust, i.e. can be *trusted* under unexpected inputs from distributional shifts, is a challenging but important research problem.

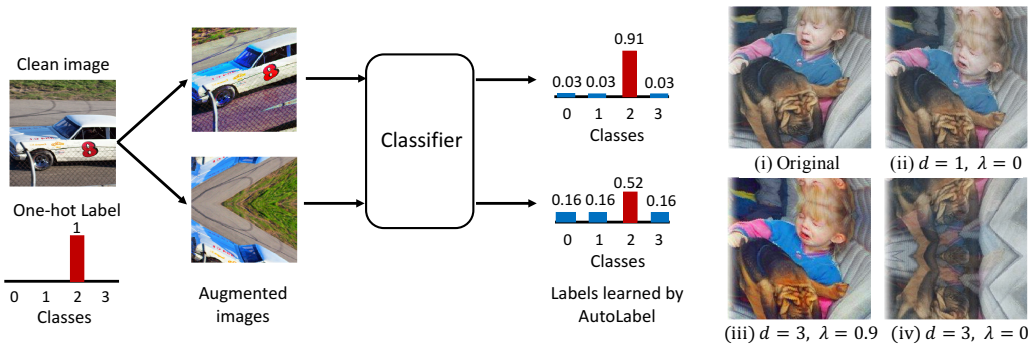
While numerous approaches have been explored for improving calibration under distribution shift, one of the fundamental building blocks is *data augmentation*: generating synthetic examples, typically by modifying existing training examples, that provide additional training data outside the empirical training distribution. A wide breadth of literature has explored what are effective ways to modify training examples, such as making use of domain knowledge through label-preserving transformations (Hendrycks et al., 2020). Approaches like these have been shown to improve the robustness and calibration of overparametrized neural networks as they alleviate the issue of neural networks overfitting to spurious features that do not generalize beyond the i.i.d. test set.

In the broad amount of research on data augmentation, most of it attempts to apply transformations that do not change the true label such that the label of the original example can also be assumed to be the label of the transformed example, without expensive manual review. While there has been a significant amount of work in how to construct such pseudo-examples in *input* space, there has been relatively little attention on whether this assumption of label-preservation holds in practice and what *label* should be assigned to such augmented inputs. For instance, many popular methods assign one-hot targets to both training data as well as augmented inputs that can be quite far away from the training data where even human raters may not be 100% sure of the label. This runs the risk of adding noise to the training process and degrading accuracy and calibration, as the model may learn to assign high confidence predictions to inputs far away from training data.

With this observation, in this paper we investigate the con-

[*]Equal contribution [1]Googol Research. Correspondence to: Yao Qin <yaoqin@google.com>.

Presented at the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning.,



(a) Pipeline

(b) Augmented images of varying distances

Figure 1. (a): An example showing `AutoLabel` assigning different labels to augmented images (e.g., by AugMix (Hendrycks et al., 2020)) based on their transformation distances to the clean image. The label for the true class is automatically learned based on the calibration performance on validation set. (b): Examples of images augmented by AugMix with different distances to the original image.

fidence assigned to target labels for augmented inputs and propose `AutoLabel`, a method that automatically adapts the confidence assigned to augmented labels, assigning high confidence to inputs close to the training data and lowering the confidence as we move farther away from the training data. Figure 1 (left) gives a high-level overview of our proposed `AutoLabel` along with examples of augmented images of varying distances generating by AugMix (Hendrycks et al., 2020) on the right. Our key contributions are:

- We propose `AutoLabel`, a *generic* approach that can automatically learn the confidence in labels for augmented data based on the transformation distance between clean and augmented distribution.
- We show that `AutoLabel` is complementary to methods which focus on generating augmented inputs by combining it with RandAug (Cubuk et al., 2020) (which includes 10 different augmentation types), and the state-of-the-art method AugMix (Hendrycks et al., 2020).
- We perform experiments on CIFAR-100 and ImageNet demonstrating that `AutoLabel` significantly improves the calibration of models on clean and corrupted data.

2. Related Work

Data Augmentation. Recent work has shown that introducing additional training examples can further improve a model's accuracy and generalization (Devries & Taylor, 2017; Cubuk et al., 2019; Yun et al., 2019; Takahashi et al., 2019; Lopes et al., 2019; Zhong et al., 2020). For example, AugMix (Hendrycks et al., 2020) utilizes stochasticity and diverse augmentations, together with a consistency loss over the augmentations, to achieve state-of-the-art corruption robustness. Mixup (Zhang et al., 2018), on the other hand, trains a neural network over convex combinations of pairs of examples and shows improved generalization of neural networks. In this paper, we investigate the choice of the target labels for augmented inputs and show how to apply `AutoLabel` to these existing data augmentation techniques to further improve model's robustness.

Calibration and Uncertainty Estimates. A variety of methods have been developed for improving a model's calibration, e.g., post-hoc calibration by temperature scaling (Guo et al., 2017) and multiclasss Dirichlet calibration (Kull et al., 2019). Model's predictive uncertainty can also be quantified using Bayesian neural networks and approximate Bayesian approaches, e.g., variational inference (Graves, 2011; Blundell et al., 2015), MCMC sampling based on stochastic gradients (Welling & Teh, 2011), and dropout-based variational inference (Kingma et al., 2015; Gal & Ghahramani, 2016). In addition to calibration over in-distribution data, more recently, Ovadia et al. (2019) show that model calibration can further degrade under unseen data shifts, where ensemble of deep neural networks (Lakshminarayanan et al., 2017) is shown to be most robust to dataset shift. On the other hand, several data augmentation methods have also been shown to improve model's calibration under data shifts. For example, AugMix is shown to improve uncertainty measures on corrupted image classification benchmarks (Hendrycks et al., 2020).

3. AutoLabel

Notations Given a clean dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1,\cdots,m}$, where $y \in \{1, ..., K\}$, the one-hot encoding of the label is denoted as $\hat{y} \in \{0, 1\}^K$, where the label for the true class $\hat{y}_{k=y} = 1$ and $\hat{y}_{k\neq y} = 0$ for others. In addition to the training data $\mathcal{D}$, we also have a clean validation set $\mathcal{D}_V$ drawn i.i.d. from the same distribution.

Our key insight is: the ***confidence*** in the labels of the augmented data depends on how distorted the transformation is. To make use of this insight, we first need to measure the transformation distance. Many data augmentation approaches have hyperparameters that reflect how large the

transformation should be. As examples, which we will discuss in-depth below, this can take the form of the number of transformations in AugMix (Hendrycks et al., 2020).

With aware of the transformation distance, we then build `AutoLabel` upon the hypothesis that effective labels for ***augmented training data*** can lead to well-calibrated predictions on the ***samely augmented validation set***, where the predicted confidence is aligned with the predicted accuracy. Thus, the training labels for augmented data can be automatically updated according to the calibration performance on the samely augmented validation data. Specifically, if a model is over-confident on the augmented validation set, then the confidence in the training labels should be decreased accordingly; otherwise the confidence should be increased. Note that computing calibration on the augmented validation set does not use the confidence in the labels of validation data.

Taking these together, our proposed `AutoLabel` is mainly composed of two components: (1) a measure of the transformation distance for the augmented data, and (2) a subroutine for updating labels of the augmented data during training.

Specifically, given a data augmentation technique `Aug` that takes in an image x and outputs an augmented image $\texttt{Aug}(x, s)$ that transformed by a distance $s \in \mathrm{R}$, `AutoLabel` updates its label based on the calibration performance on the samely augmented validation data. Since calibration can not be computed over a single data point, we must obtain an augmented validation set that is transformed by the same distance s. To this end, we discretize the transformation distance s into N buckets $\{S_1, \cdots, S_N\}$ where each S_n is a range, and we can generate augmented data for bucket S_n by sampling a distance uniformly in that range $s \sim \mathcal{U}(S_n)$ to generate $\texttt{Aug}(x, s)$. In this way, we can generate the augmented validation set $\mathcal{Q}(S_n) = \{(\texttt{Aug}(x_i, s), y_i) | (x_i, y_i) \in \mathcal{D}_V, s \sim \mathcal{U}(S_n)\}$, which is used to learn the labels for any training data transformed by a distance $s \in S_n$.

With the augmented validation set $\mathcal{Q}(S_n)$, `AutoLabel` updates the confidence of the true class $\tilde{y}_{k=y}(S_n)$ after each training epoch t according to:

$$\tilde{y}^{t+1}_{k=y}(S_n) = \tilde{y}^{t}_{k=y}(S_n) - \alpha \cdot \mathrm{ECE}^t(\mathcal{Q}(S_n)) \cdot \mathrm{sign}(\mathrm{Conf}^t(\mathcal{Q}(S_n)) - \mathrm{Acc}^t(\mathcal{Q}(S_n))) \quad (1)$$

where $\mathrm{ECE}(\mathcal{Q}(S_n))$, $\mathrm{Acc}(Q(S_n))$ and $\mathrm{Conf}(\mathcal{Q}(S_n))$ are respectively the expected calibration error, accuracy and confidence on the augmented validation set. The sign of ($\mathrm{Conf}(\mathcal{Q})$ - $\mathrm{Acc}(\mathcal{Q})$) indicates if the model is overall over-confident (> 0) or under-confident (< 0). Intuitively, if the model is over-confident on the validation set, we should reduce the confidence given to the true class $\tilde{y}_{k=y}$, otherwise we should increase $\tilde{y}_{k=y}$. The expected calibration error on the augmented validation set $\mathrm{ECE}(\mathcal{Q}) \geq 0$ suggests to what extent we should adjust the labels as the optimal result is $\mathrm{ECE}(\mathcal{Q}) = 0$ when the training converges. The hyperparameter α controls the step size of updating the labels. Since $\tilde{y}^{t+1}_{k=y}$ stands for the probability of the true class, we clip the value to be within $[\mathrm{Acc}^t(\mathcal{Q}), 1]$ after each update. $\mathrm{Acc}^t(\mathcal{Q})$ is used as the minimum clipping value to prevent $\tilde{y}^{t+1}_{k=y}$ from being too small as $\mathrm{Acc}^t(\mathcal{Q}) \to \frac{1}{K}$ when the classifier is a random guesser.

Given the updated label for the true class $\tilde{y}^{t+1}_{k=y}(S_n)$, `AutoLabel` takes a label smoothing approach to uniformly distribute the remaining probability to other classes:

$$\tilde{y}^{t+1}_{k\neq y}(S_n) = (1 - \tilde{y}^{t+1}_{k=y}(S_n)) \cdot \frac{1}{K-1}, \quad (2)$$

where K is the number of classes in the dataset and $\sum_{k=1}^{K} \tilde{y}_k = 1$. Finally, `AutoLabel` trains the model using $\tilde{y}(S_n)$ as the target for the cross-entropy loss across the augmented data. A complete pseudocode for `AutoLabel` is presented in Algorithm 1 in Appendix.

To demonstrate `AutoLabel` can easily slot into existing data augmentation methods, we show how to apply `AutoLabel` to automatically adjust the confidence in labels over different data augmentation methods: RandAug (Cubuk et al., 2020) and AugMix (Hendrycks et al., 2020). These data augmentation methods originally use one-hot labels for augmented data and we discuss in details in Section A in Appendix how to apply `AutoLabel` for each of them.

4. Experiments

4.1. Datasets and Evaluation Metrics

Datasets We report the performance on CIFAR-100 (Krizhevsky, 2009) and ImageNet (Russakovsky et al., 2015) as well as the corrupted datasets: CIFAR-100-C and ImageNet-C (Hendrycks & Dietterich, 2019), which include different corruptions types (17 types for CIFAR-100 and 15 types for ImageNet) that are frequently encountered in natural images. Note that the corruption type in the corrupted dataset ***do not overlap*** with transformations used for data augmentations.

We use a Wide ResNet-28-10 (Zagoruyko & Komodakis, 2016) for CIFAR-100, and a ResNet-50 (He et al., 2016) for ImageNet as our basic model architectures and mainly compare `AutoLabel` with standard label smoothing (LS) (Szegedy et al., 2016).

Evaluation Metrics We report the classification accuracy and expected calibration error on the clean datasets as **Acc** (higher is better) and **ECE** (lower is better) respectively. The accuracy and expected calibration error on the corrupted datasets are represented as **cAcc** and **cECE**, which are computed as an average over all the corruption types across 5 corruption severities.

Table 1. Effects of `AutoLabel` for RandAug and AugMix on CIFAR100 and ImageNet and the corresponding corrupted datasets. All numbers reported in the table are in %, an average of 4 independent runs on CIFAR-100 and 2 independent runs on ImageNet. The arrow indicates better direction. Best results are highlighted in **Bold**.

Method	Acc/cAcc (↑)		ECE/cECE (↓)	
	CIFAR100	ImageNet	CIFAR100	ImageNet
RandAug	82.0/63.0	76.9/43.4	4.1/13.0	2.0/6.4
+ LS	82.2/63.6	76.9/43.5	2.4/8.2	1.2/5.5
+ `AutoLabel`	**82.7/64.8**	**76.9/43.9**	**1.9/5.6**	**1.0/5.1**
AugMix	81.1/64.3	75.9/46.1	4.5/10.9	1.5/4.9
+ LS	81.3/64.6	76.0/46.3	2.6/6.5	1.5/4.6
+ `AutoLabel`	**81.9/65.5**	**76.4/46.5**	**1.8/4.2**	**1.4/4.2**

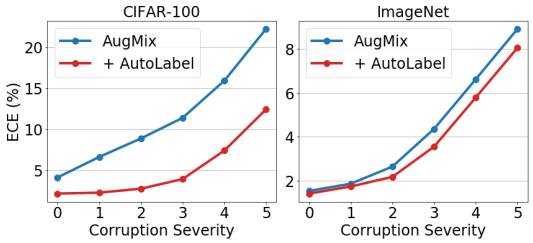


Figure 2. Expected calibration error (ECE) of AugMix trained with one-hot labels and `AutoLabel` across corruption severities on CIFAR100 and ImageNet. Severity 0 denotes clean data.

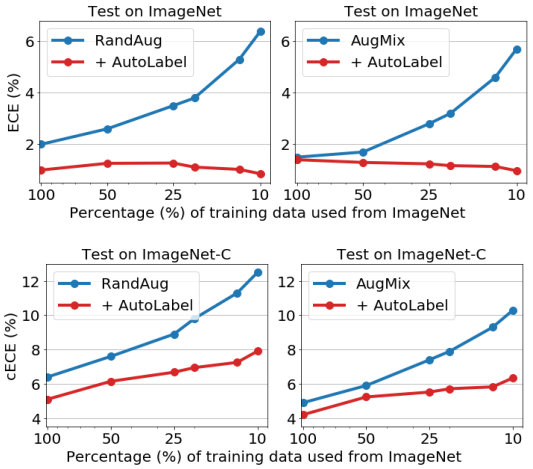


Figure 3. Effects of `AutoLabel` for RandAug and AugMix on ImageNet with a reduced size of training data. When training a model with limited training data, the improvement of `AutoLabel` over calibration is significantly increased. $\alpha = 0.02$ is used for `AutoLabel` for all experiments here without further tuning.

4.2. AutoLabel improves calibration

In this section, we apply `AutoLabel` to RandAug and AugMix to investigate if `AutoLabel` can make data augmentation approaches more effective in improving calibration, especially under distributional shifts. We see in Table 1 a clear picture: `AutoLabel` consistently helps RandAug and AugMix improve both accuracy and calibration across CIFAR100 and ImageNet, and greatly outperforms label smoothing. In addition, we can see that `AutoLabel` has a much more significant improvement on models' calibration on the corrupted datasets. As shown in Figure 2 we analyze how calibration performance changes with the severity of the corruption being tested against, comparing AugMix trained with one-hot labels and with `AutoLabel`. We see that the baseline AugMix is increasingly worse calibrated as the corruption increases, but `AutoLabel` dampens that trend effectively. Similar patterns are observed when `AutoLabel` is applied to RandAug.

Looking more closely, we find that the improvement of `AutoLabel` over calibration on ImageNet is relatively smaller compared to CIFAR-100. We conjecture that this is mainly due to the much larger training data on ImageNet, leading to a greater generalization performance and the headroom for improvement is relatively limited. To validate this, we train the same networks with a reduced size of training images, e.g., we randomly sample 50% training images from the whole training dataset (∼1.2M training images). Note that we use the same hyperparameter $\alpha = 0.02$ in Eqn (1) without further tuning for each training data size. From Figure 3 we can see that as the percentage of training data is reduced, the calibration performance of both RandAug and AugMix trained with one-hot labels becomes significantly worse. In contrast, `AutoLabel` enables models to keep a low calibration error on the clean test set and a much smaller calibration error on the corrupted dataset.

5. Conclusion

In this paper, we propose `AutoLabel` to automatically learn the confidence in labels for augmented data based on the transformation distance between the augmented data and the clean data. We demonstrate the effectiveness of `AutoLabel` by applying it to RandAug and AugMix. We see that `AutoLabel` greatly improves the models' calibration, especially on corrupted data. More generally, we believe that more nuanced approaches to setting labels for augmented data, beyond assuming label-preserving transformations, will lead to more effective data augmentation techniques.

References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. In *ICML*, volume 37, pp. 1613–1622, Lille, France, 07–09 Jul 2015. PMLR.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. RandAugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Devries, T. and Taylor, G. W. Improved regularization of convolutional neural networks with CutOut. *arXiv preprint arXiv:1708.04552*, 2017.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pp. 1050–1059, 2016.

Graves, A. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems 24*, pp. 2348–2356, 2011.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *Proceedings of the International Conference on Learning Representations*, 2019.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. In *International Conference on Learning Representations*, 2020.

Kingma, D. P., Salimans, T., and Welling, M. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pp. 2575–2583, 2015.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Kull, M., Perello Nieto, M., Kängsepp, M., Silva Filho, T., Song, H., and Flach, P. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Advances in Neural Information Processing Systems 32*, pp. 12316–12326, 2019.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6405–6416, 2017.

Lopes, R. G., Yin, D., Poole, B., Gilmer, J., and Cubuk, E. D. Improving robustness without sacrificing accuracy with patch Gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pp. 13991–14002, 2019.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., and Li, F.-F. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Takahashi, R., Matsubara, T., and Uehara, K. Data augmentation using random image cropping and patching for deep cnns. *IEEE Transactions on Circuits and Systems for Video Technology*, PP:1–1, 08 2019. doi: 10.1109/TCSVT.2019.2935128.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. *ICCV*, 2019.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *ArXiv*, abs/1605.07146, 2016.

Zhang, H., Cissé, M., Dauphin, Y., and Lopez-Paz, D. Mixup: Beyond empirical risk minimization. In *International Conference on Learning Representation*, 2018.

Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y. Random erasing data augmentation. In *AAAI*, 2020.

A. Algorithms

A.1. AutoLabel for RandAug

RandAug includes 10 different types of transformations from AutoAugment (Cubuk et al., 2019) and RandAugment (Cubuk et al., 2020): color, rotation, autocontrast, equalize, posterize, solarize, shear X, shear Y, translate X and translate Y. During training, RandAug randomly applies one transformation to generate the augmented data. Unlike RandAugment (Cubuk et al., 2020) that optimizes a single distortion magnitude for all the transformations, RandAug randomly samples a distortion magnitude $m \in \{1, \cdots, m_{max}\}$, where m_{max} is the maximum distortion magnitude. Finally, the model is trained on the augmented data with one-hot labels.

Instead of using one-hot labels, we propose `AutoLabel` to automatically learn the confidence in labels for the augmented data that are transformed by different transformation distances. In RandAug, the transformation distance is determined by two factors: (1) the type of sampled transformation, in total we have 10 different transformations, and (2) the distortion magnitude m.

To learn the labels for the augmented data transformed by a specific operation with a distortion magnitude at m, `AutoLabel` applies the same transformation distorted by the same magnitude m to construct the augmented validation set. Then `AutoLabel` updates the confidence of labels according to Eqn (1) & (2) and trains the model with these updated labels.

A.2. AutoLabel for AugMix

AugMix (Hendrycks et al., 2020) is a data augmentation technique that achieves state-of-the-art robustness and uncertainty estimates under data shift. Specifically, AugMix augments the input data via feeding the input x into an augmentation chain[1] which consists of $d \in \{1, 2, 3\}$ transformations randomly sampled from 10 different operations used in RandAug with a fixed distortion magnitude. Then a convex combination is performed to mix the augmented image x_{aug} with the original image x: $\texttt{Aug}_{augmix}(x) = \lambda \cdot x + (1 - \lambda) \cdot x_{\text{aug}}$, where the mixing parameter $\lambda \in [0, 1]$ is randomly sampled from a uniform distribution.

In AugMix, the transformation distance is mainly controlled by two parameters[2]: (1) the depth of the augmentation chain d, which decides how many augmentation operations are applied to the original image (shown in Figure 1(b) (ii) and (iv)); (2) the mixing parameter λ, which controls the ratio of the augmented image x_{aug} and the original image x (shown in Figure 1(b) (iii) and (iv)). As a result, we can define the distance bucket $S_{d,n}$ for the augmented data as: $S_{d,n} = S_{d,\lceil \lambda N \rceil}$,[3] where N is the total number of buckets at a given depth d.

Next, to learn the labels for augmented training data within a distance bucket $S_{d,n}$, `AutoLabel` constructs an augmented validation set $\mathcal{Q}(S_{d,n})$ by feeding the validation images into an augmentation chain with the depth d and then randomly sample a mixing parameter λ' from a uniformly distribution: $\lambda' \sim \mathcal{U}(\frac{n}{N}, \frac{n+1}{N})$ to mix the original image and the augmented image. Finally, `AutoLabel` updates the labels $\tilde{y}(S_{d,n})$ according to Eqn (1) & (2) and trains the model using these updated labels.

B. The proposed `AutoLabel` Algorithm

In Algorithm 1 we describe how `AutoLabel` works.

C. Implementation Details

We train the vanilla models on CIFAR100 and ImageNet using the open-sourced code for uncertainty baselines at https://github.com/google/uncertainty-baselines/tree/master/baselines.

C.1. RandAug

When applying label smoothing to RandAug, we sweep the hyperparameter ρ which decides the smoothing degree in a range [0, 0.1] with a step size 0.01 and find the best $\rho = 0.02$ for CIFAR-100 and $\rho = 0.01$ for ImageNet.

When applying `AutoLabel` to RandAug, the hyperparameter α in Eqn (1) in the main text is swept in a set and we choose the best $\alpha = 0.01$ for CIFAR-100 and $\alpha = 0.02$ for ImageNet.

C.2. AugMix

For AugMix (Hendrycks et al., 2020), the max depth of the augmentation chain is $d_{max} = 3$ for CIFAR-100 and

[1] The original AugMix (Hendrycks et al., 2020) uses 3 augmentation chains. However, we consistently observe an accuracy increase when we use one augmentation chain.

[2] Transformation types could provide us more precise transformation distance but is not our main focus in AugMix.

[3] In the special case where $\lambda = 0$, we merge it into bucket S_1 to avoid creating an additional bucket, similarly for ϵ in adversarial training.

Algorithm 1 Pseudocode of `AutoLabel`

Input: A training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1,\cdots,m}$, a validation dataset $\mathcal{D}_{\mathcal{V}}$ drawn i.i.d. from the same distribution, an augmentation method `Aug`. Number of classes K, number of training epochs T, number of distance buckets N and the hyperparameter α.
We perform `Aug` to obtain the augmented training data $\texttt{Aug}(x, s)$, where the transformation distance s is determined by the hyperparameters in the `Aug`. We discretize the transformation distance s into N buckets $\{S_1, \cdots, S_N\}$, where each S_n is a range.
For each distance bucket S_n, we initialize $\tilde{y}^0(S_n)$ as the one-hot label.
for $t = 0$ **to** $T - 1$ **do**
  Minimize cross-entropy loss over the augmented training data with smoothed labels $\tilde{y}^t(S_n)$.
  for $n = 1$ **to** N **do**
    Generate an augmented validation set: $\mathcal{Q}(S_n) = \{(\texttt{Aug}(x_i, s), y_i) | (x_i, y_i) \in \mathcal{D}_V, s \sim \mathcal{U}(S_n)\}$.
    Update the label for the true class $\tilde{y}^{t+1}_{k=y}(S_n)$:
    $\tilde{y}^{t+1}_{k=y}(S_n) = \tilde{y}^t_{k=y}(S_n) - \alpha \cdot \text{ECE}^t(\mathcal{Q}(S_n)) \cdot \text{sign}(\text{Conf}^t(\mathcal{Q}(S_n)) - \text{Acc}^t(\mathcal{Q}(S_n)))$ ▷ *according to Eqn (1)*
    Clip $\tilde{y}^{t+1}_{k=y}(S_n)$ to be within $[\text{Acc}^t(\mathcal{Q}(S_n)), 1]$
    Update the label for other classes $\tilde{y}^{t+1}_{k\neq y}(S_n)$: $\tilde{y}^{t+1}_{k\neq y}(S_n) = (1 - \tilde{y}^{t+1}_{k=y}(S_n)) \cdot \frac{1}{K-1}$ ▷ *according to Eqn (2)*
  end for
end for

ImageNet following the original work.

When applying label smoothing to AugMix, we sweep the hyperparameter ρ which decides the smoothing degree in a range [0, 0.1] with a step size 0.01 and find the best $\rho = 0.02$ for CIFAR-100 and $\rho = 0.01$ for ImageNet.

When applying `AutoLabel` to AugMix, we set the number of distance buckets to be $d_{max} \cdot N = 3 \cdot 5 = 15$ for both datasets. The hyperparameter α in Eqn (1) in the main text is swept in a set and we choose the best $\alpha = 0.02$ for CIFAR-100 and ImageNet.