Latent and Adversarial Data Augmentation for Sound Event Detection and Classification

David Perera, Slim Essid, Gaël Richard

▶ **To cite this version:**

David Perera, Slim Essid, Gaël Richard. Latent and Adversarial Data Augmentation for Sound Event Detection and Classification. International workshop on Detection and Classiffication of Acoustic Scenes and Events (DCASE), Nov 2022, Nancy, France. hal-03782827

HAL Id: hal-03782827
https://hal.science/hal-03782827

Submitted on 21 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

LATENT AND ADVERSARIAL DATA AUGMENTATIONS FOR SOUND EVENT DETECTION AND CLASSIFICATION

David Perera, Slim Essid, Gaël Richard

LTCI, Télécom Paris, Institut Polytechnique de Paris, France
{david.perera, slim.essid, gael.richard}@telecom-paris.fr

ABSTRACT

Invariance-based learning is a promising approach in deep learning. Among other benefits, it can mitigate the lack of diversity of available datasets and increase the interpretability of trained models. To this end, practitioners often use a consistency cost penalizing the sensitivity of a model to a set of carefully selected data augmentations. However, there is no consensus about how these augmentations should be selected. In this paper, we study the behavior of several augmentation strategies. We consider the task of sound event detection and classification for our experiments. In particular, we show that transformations operating on the internal layers of a deep neural network are beneficial for this task.

Index Terms— sound event detection, data augmentation, adversarial learning

1. INTRODUCTION

Ambient sound analysis is a rapidly growing field, which has several important industrial applications such as security (e.g., audio surveillance), health (e.g., home monitoring, diagnosis based on sound), transportation (e.g., autonomous driving), manufacturing industry (e.g., predictive maintenance) and bioacoustics (e.g., ecosystem evolution tracking). In this context, sound event detection and classification consists in identifying and temporally localizing sound events in a complex acoustic scene. Deep learning has been successfully applied to this task, significantly improving the state of the art. However, this approach has two major drawbacks [1]. On the one hand, it requires a large amount of annotated data, which is costly and time-consuming to gather. On the other hand, the models obtained by this approach lack interpretability, making it hard to assess their reliability in unseen situations.

Data augmentation is an efficient technique which can mitigate the lack of available data and exploit unlabeled data [2]. This method also makes it possible to force trained models to learn specific invariants. By selecting relevant data augmentations, one can increase the model robustness to annotation errors and make them easier to interpret [2]. Moreover, data augmentation can be easily applied to any training algorithm and can improve its performance at a minimal development cost. This explains the popularity of this approach, especially in the field of sound event detection.

Many data augmentation strategies have been designed over the past decades. They now form a large set of methods that is costly to explore exhaustively. As a consequence, default settings are often passed from one system to another with only limited investigation [3], which leads to sub-optimal design choices.

This paper proposes a comparative study of the impact of several data augmentation strategies on task 4 of the DCASE challenge.[1] We focus on domain agnostic data augmentations. In this context, we study the impact of adversarial augmentations and latent augmentations (data augmentations applied directly on the latent space of a deep neural network). We propose a common training framework to compare these augmentation strategies. By carefully selecting them, we show that it is possible to outperform the baseline model we compete against and to simplify its training objective.

This article is organized as follows. Section 2 presents a brief state of the art. Section 3 introduces task 4 of the DCASE challenge and describes its baseline. Section 4 explains the experimental framework that we use. Finally, section 5 discusses the obtained results.

2. STATE OF THE ART

There are two main strategies used to overcome the lack of training data: exploiting additional data from a related domain, and constraining the algorithm using domain knowledge [2]. Combining both ideas, invariance-based approaches penalize the variations of a model f in the neighborhood of the training data. More precisely, these approaches aim to reduce the quantity $\|f(x) - f(\tau(x))\|$, where x denotes a training point, $\|\cdot\|$ denotes a norm and τ denotes a data augmentation. This forces the model f to learn some invariants, which can be defined explicitly. We thus obtain some guarantees on the behavior of f, and consequently improve its interpretability. We briefly review here the main developments in this field.

The Ladder Network [4] adds to the objective function a cost enforcing the invariance of the model to small perturbations of its input and internal representations. A key point is that this method can leverage unlabeled data. In [5], the authors use Dropout augmentation [6] instead of random noise.

Temporal Ensembling (TE) [7] compares the prediction $f(\tau(x))$ computed for an augmented input $\tau(x)$ with an average of the predictions $f^{(n)}(\tau(x))$ computed at different epochs n. This method is inspired by the state of the art in stochastic optimization [8]. The idea is to accelerate the convergence of f by mitigating the randomness resulting from the selection of the mini-batches and the augmentation of the input.

Mean Teacher (MT) [9] is an improvement of TE: instead of averaging the predictions at each training epoch, it maintains an average of the model parameters. Moreover, it updates this average after each mini-batch instead of updating it after each epoch, which speeds up training.

In [10], the authors use Mixup augmentation [11] within the MT framework. This new training method encourages convex in-

[1] https://dcase.community/

terpolation between two input samples. It is based on the following hypothesis: the decision boundaries of a classifier should be located between samples of different classes. Consequently, instead of moving an input sample in a random direction, which is inefficient, it is better to move it in the direction of a sample from a different class.

Following a similar line of research, Virtual Adversarial Training (VAT) promotes invariance to small perturbations of the input. This method uses adversarial perturbations (which maximize the variation of f) instead of using random perturbations. According to the authors, adversarial perturbations constitute a better heuristic, and can speed up the convergence of the algorithm. This algorithm is however based on a second order approximation of the adversarial perturbations, which is slow to compute. The Adversarial Noise Layer [12] generalizes the idea of adversarial perturbation to the intermediate layers of a model. It uses a first order approximation, which is faster to compute. In [13], the authors study a variant of this method based on the Dropout augmentation.

Universal Adversarial Perturbation (UAP) [14] is a further generalization of adversarial attacks. While the approaches presented above focused on fooling one model on a single instance of the training set, UAP seek to fool several models on most instances of the training set. The authors have shown that the resulting perturbations exploit the geometry of the classifiers' decision boundaries. This makes them useful to design an invariance metric. However, their high computational cost discourages any use, aside from evaluation.

Finally, we can mention Mixmatch [15]. In addition to a consistency cost enforcing invariance to data augmentations, this method takes into account an entropy cost. The purpose of this cost is to increase the confidence level of the predictions made by the trained model.

Some of the methods presented above have already been applied to DCASE Task 4. Since 2019, the MT method has been used as the baseline for this task, and many algorithms submitted to the challenge are also based on this method. The authors of [3] propose an in-depth study of the application of MT to this task. The VAT method has also been applied to this task, with three different architectures: A Recurrent Convolutional Network (CRNN) [16], a Gated Recurrent Neural Network [17], and a Gated Recurrent Convolutional Neural Network [18]. More recently, the authors of [19] have studied a variant in which the noise is not adversarial but random. However, both the training method and the architecture change from one of these articles to another. It is therefore difficult to factor out the impact of the data augmentation strategy alone.

Closest to our work, [20] and [21] study the impact of several types of data augmentations on a single model. The authors focus on audio-specific data augmentations: pitch shifting, time shifting, reverberation, frequency masking and time masking. Unlike these studies, however, we focus our work on adversarial and latent data augmentation. There are two main reasons for this. First, these augmentations are domain agnostic, and could potentially work across a vast range of tasks. This makes them very potent and interesting to study. Second, these augmentations can be restricted to the audio domain, giving birth to adversarial audio augmentations (which could be seen as realistic worst-case scenarios), and latent audio augmentations (which would force the internal representations of a deep neural network to keep the structure of audio data, such as time-shift or pitch-shift invariance). The aim of this paper is to provide insights about the impact of adversarial and latent augmentations on sound event detection. We use task 4 of DCASE as a case study.

3. TASK DEFINITION

Task 4 of DCASE uses the dataset Domestic Environment Sound Event Detection (DESED) [20], which is composed of 10-second audio recordings made in domestic environments. DESED is divided into three distinct datasets: an unlabeled dataset $\mathcal{D}_u$, a weakly labelled dataset $\mathcal{D}_w$, and a synthetic strongly labelled dataset $\mathcal{D}_s$. There are ten possible classes for the annotation of sound recordings. Each recording in $\mathcal{D}_w$ is annotated with the set of sound events it contains. For $\mathcal{D}_s$, each event is temporally localized in addition to being identified by a label.

Two metrics are used to evaluate the models. To measure the temporal accuracy of the predictions, we use an event-based F1 macro score, which is denoted by $macro$ in the following. In order to measure the labeling accuracy, we use the Polyphonic Sound Detection Score [22], which is denoted by $psds$ in the following. For the computation of these two metrics, we adopt the parameters proposed for the 2020 edition of the DCASE challenge. We use the $macro$ score, computed on the public evaluation dataset[2] proposed in this same edition, to evaluate and compare the models.

In order to increase the usability of our results, we build our study on [20], which already investigates several design choices concerning task 4 of DCASE. In particular, we use the same baseline model[3] as a basis for our experiments. This baseline exploits a CRNN architecture, which is commonly used in audio. It achieves high performance on the task. Moreover, the impact of its various components has been exhaustively studied. The baseline takes as input Mel spectrograms with 128 Mel bands, built with an analysis window of size 2048 and a hop length of 255. The input signals are sampled at 16kHz. The convolutional block of the CRNN is composed of 7 layers with filter sizes (16, 32, 64, 128, 128, 128, 128) and a kernel of size 3x3. Each convolution is followed by a Batch Normalization layer, a Gated Linear Unit, a Dropout layer with probability 0.5, and a Maxpooling layer. The recurrent block is composed of two Gated Recurrent Units with 128 layers each. It is followed by the attention pooling layer described in [23]. Median filtering is applied during post-processing. This architecture features 11 million trainable parameters. The model is trained on 200 epochs with the Adam optimizer.

4. AUGMENTATION STRATEGIES

4.1. Training framework

We compare several training objectives. In each case, the training objective can be decomposed into three cost functions: a classification metric penalizing prediction errors $\mathcal{L}_{class}$, a consistency metric promoting invariance to data augmentations $\mathcal{L}_{const}$ and a regularization metric $\mathcal{L}_{reg}$.

We segment the training mini-batches into three parts, each corresponding to one of the datasets $\mathcal{D}_u$, $\mathcal{D}_w$ and $\mathcal{D}_s$. We use the respective proportions ($1/2$, $1/4$, $1/4$).

We note f the baseline and f_{ema} its exponential moving average across iterations. If we note $f^{(n)}$ the model obtained after

[2] https://zenodo.org/record/3588172

[3] https://github.com/turpaultn/dcase20_task4/

iteration n (similar notation for f_{ema}) and $\alpha^{(n)}$ a coefficient varying during the training, then f_{ema} is defined by

$$f_{ema}^{(n+1)} = \alpha^{(n)} f^{(n)} + (1 - \alpha^{(n)}) f_{ema}^{(n)}. \quad (1)$$

Noting x a recording in the DESED dataset, y the corresponding label when it exists, d a Gaussian noise vector, $\mathcal{L}_{BCE}$ the binary cross-entropy and $\mathcal{L}_{MSE}$ the mean square error, we can define the training objective of the baseline as follows:

$$\mathcal{L}_{class} = \begin{cases} \mathcal{L}_{BCE}[f(x),\ y] & \text{if } (x,y) \in \mathcal{D}_w \cup \mathcal{D}_s \\ 0 & \text{else} \end{cases}, \quad (2)$$

$$\mathcal{L}_{const} = \mathcal{L}_{MSE}[f(x),\ f_{ema}(x+d)], \quad (3)$$

$$\mathcal{L}_{reg} = 0. \quad (4)$$

Through our experiments, we keep the classification cost used by the baseline. However, we will experiment with other consistency and regularization costs.

4.2. Consistency costs

The simplest consistency cost we considered is the $\mathcal{L}_2$ distance between a prediction $f(x)$ and a perturbed prediction $f(x+d)$ (with d a Gaussian noise vector):

$$\mathcal{L}_{const} = \mathcal{L}_{MSE}[f(x),\ f(x+d)]. \quad (5)$$

We have also considered an adversarial consistency cost,

$$d = \nabla_d\ \mathcal{L}_{MSE}[f(x),\ f(x+d)]\,|_{d=0}, \quad (6)$$

$$\mathcal{L}_{const} = \mathcal{L}_{MSE}[f(x),\ f(x+d)], \quad (7)$$

as well as the following variant of VAT:

$$d = argmax_{\|d\| \leq \epsilon}\ \mathcal{L}_{MSE}[f(x),\ f(x+d)], \quad (8)$$

$$\mathcal{L}_{const} = \mathcal{L}_{MSE}[f(x),\ f(x+d)]. \quad (9)$$

The original VAT algorithm uses the KL-divergence as a metric to compare the two outputs of the classifier $f(x)$ and $f(x+d)$. However, the $\mathcal{L}_2$ distance proved to be empirically better for this task. This choice also has the added benefit to homogenize the definition of the consistency costs $\mathcal{L}_{const}$ that we study. Indeed, these costs differ now only by the definition of the perturbation d.

Finally, we use a Mixup consistency cost, computed from a second sample x'

$$\mathcal{L}_{const} = \mathcal{L}_{MSE}[Mixup(f(x), f(x')),\ f(Mixup(x, x'))]. \quad (10)$$

All these augmentations can be applied to the internal representations of the model. We apply data augmentation to the output of the CNN block and to the output of the RNN block of the baseline (see Figure 1).

4.3. Regularization costs

The baseline does not use an explicit regularization cost. This is because MT combines a consistency criterion and regularization criterion into a single training objective

$$\mathcal{L}_{const} = \mathcal{L}_{MSE}[f(x),\ f_{ema}(x+d)]. \quad (11)$$

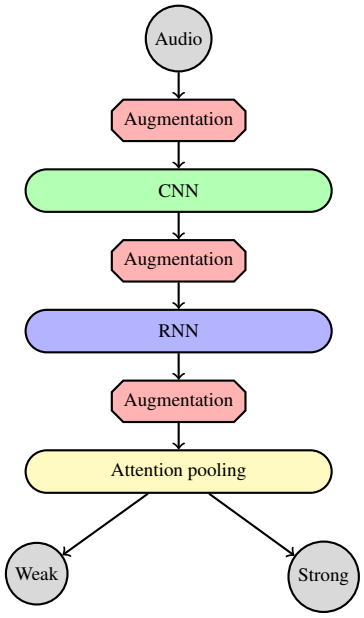

Figure 1: Architecture of the baseline system and location of the data augmentation modules.

However, this method prevents us from fine tuning the trade-off between regularization and consistency in the final training objective. Yet, we have observed that this fine tuning is crucial when the model is dealing with difficult augmentations, such as adversarial attacks (see subsection 5.1).

For this reason, we have also experimented with a more traditional $\mathcal{L}_2$ regularization cost

$$\mathcal{L}_{const} = \mathcal{L}_{MSE}[f(x),\ f(x+d)], \quad (12)$$

$$\mathcal{L}_{reg} = \mathcal{L}_2[f], \quad (13)$$

where $\mathcal{L}_2[f]$ is computed on the model parameters.

5. EXPERIMENTAL RESULTS

The different training methods that we have used and the scores that we have obtained are summarized in Table 1. Hyperparameters have been selected using the validation score computed on a split of $\mathcal{D}_s$.

Table 1: Comparison of training methods. Highest values are shown in bold, and scores above the baseline are underlined. The baseline uses random noise as an augmentation strategy.

Augmentation	Location		Regularization			Scores	
	Input	Latent	None	MT	$\mathcal{L}_2$	macro	psds
None			x			0.291	0.500
Baseline	x			x		0.381	0.552
Random	x				x	**0.397**	0.534
Random		x			x	0.388	**0.565**
Adversarial		x			x	0.374	0.559
VAT	x			x		0.311	0.461
VAT	x				x	0.358	0.539
VAT		x			x	0.362	0.542
Mixup		x			x	0.351	0.507

5.1. Separate regularization and consistency costs

The experiments that we conducted on random and adversarial augmentations suggest that it is advantageous to keep a separate regularization and consistency cost.

If we compare lines 2 and 3 of Table 1, which both use input noise as an augmentation strategy and differ only by the regularization method, we notice an improvement in $macro$ score (increasing from 0.381 to 0.397) balanced by a drop in $psds$ score (decreasing from 0.552 to 0.534). This leads to the following observation. Using $\mathcal{L}_2$ regularization instead of MT regularization gives us an additional degree of freedom, which can be used to optimize either the $macro$ score or the $psds$ score. This property is useful. Indeed, $macro$ and $psds$ scores are partially conflicting: depending on the use case, it might be advantageous to optimize either of them [24].

If we compare lines 6 and 7 of Table 1, which both use VAT as an augmentation strategy and differ only by the regularization method, we notice this time an improvement both in $macro$ score (increasing from 0.311 to 0.358) and $psds$ score (increasing from 0.461 to 0.539). We hypothesize that, when the data augmentations become harder to handle for the model, it becomes increasingly advantageous to keep a separate regularization and consistency cost.

5.2. Simplified training method

As we have seen by focusing on the $macro$ metric and comparing lines 2 and 3 of Table 1, the MT objective can be advantageously replaced by a $\mathcal{L}_2$ regularization cost and a consistency cost penalizing the sensitivity of the model f to input noise. Further experiments have shown that this advantage is maintained even when the amount of data used for training is decreased. Moreover, the classification score is improved across classes. This algorithm thus seems to escape the curse of class dependency that has been recently discussed in the literature on regularization and data augmentations [25].

This new method is a simplification of the MT approach. As we have already mentioned, it makes it easier to analyze the individual impact of the training objectives (classification, regularization, and consistency), and to fine-tune their relative contribution during training. Moreover, since the f_{ema} model is no longer necessary, we can divide by two the number of parameters kept in memory during training. Although we did not observe it during our experiments, this method may also lead to faster training. Indeed, the update of f_{ema} is not necessary anymore, and the added regularization cost is already computed implicitly by some implementation of the Adam optimizer (for instance, this is the case in the widely used framework *Pytorch*). However, these advantages come at the cost of an additional hyperparameter to adjust.

5.3. Advantage of depth

The experiments that we conducted on random and adversarial augmentations suggest that it is advantageous to use latent augmentations.

If we compare lines 3 and 4 of Table 1, we notice that using latent noise instead of input noise leads to an increase in $psds$ score. This training method offers a good compromise between $macro$ score (decreasing slightly from 0.397 to 0.388) and $psds$ score (increasing from 0.534 to 0.565). This result could be explained by the following. The authors of [2] suggest that adding noise to the input of a classifier moves its decision boundaries away from the training data, and thus improves its generalization power. Following this hypothesis, adding noise to the internal layers of f improves its performance as a classifier. This leads to an improvement of the $psds$ score, which is more sensitive to the labeling accuracy. On the other hand, the $macro$ score, which is more sensitive to the temporal accuracy, does not vary as much.

5.4. Adversarial and Mixup augmentations

Despite encouraging first results, the experiments we performed with adversarial and Mixup augmentations did not yield any improvement over our experiments with random noise. We may explain the failure of adversarial perturbations by observing that the generated samples are not realistic. Consequently, the invariants developed with these methods are not useful for the detection and classification of sound events. This motivates the study of data augmentations that are specific to the audio domain, and are more relevant for this task.

6. CONCLUSION

This article analyzes the impact of several domain agnostic data augmentation strategies on the baseline of task 4 of the DCASE challenge. We propose a simple variant of the MT method, which improves its performance. The results that we obtained with adversarial augmentations suggest that it may be advantageous to restrict the search space to realistic augmentations. In a future study, we will test this hypothesis and examine the impact of audio related augmentations on sound event detection. As the dimension of the augmentation space increases, we anticipate that the search strategy will become a central issue.

7. ACKNOWLEDGMENT

The material contained in this document is based upon work funded by the Agence National de la Recherche en Intelligence Artificielle (PhD program in AI) and Hi! PARIS through its PhD funding program.

8. REFERENCES

[1] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.

[2] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[3] N. Turpault and R. Serizel, "Training sound event detection on a heterogeneous dataset," *arXiv preprint arXiv:2007.03931*, 2020.

[4] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *Advances in Neural Information Processing Systems*, vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/378a063b8fdb1db941e34f4bde584c7d-Abstract.html

[5] P. Bachman, O. Alsharif, and D. Precup, "Learning with pseudo-ensembles," in *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/hash/66be31e4c40d676991f2405aaecc6934-Abstract.html

[6] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[7] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[8] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *Journal on control and optimization*, vol. 30, no. 4, pp. 838–855, 1992.

[9] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in neural information processing systems*, vol. 30, 2017.

[10] V. Verma, K. Kawaguchi, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," *arXiv preprint arXiv:1903.03825*, 2019.

[11] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[12] Z. You, J. Ye, K. Li, Z. Xu, and P. Wang, "Adversarial noise layer: Regularize neural network by adding noise," in *International Conference on Image Processing (ICIP)*. IEEE, 2019.

[13] S. Park, J. Park, S.-J. Shin, and I.-C. Moon, "Adversarial dropout for supervised and semi-supervised learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[14] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1765–1773.

[15] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[16] A. Agnone and U. Altaf, "Virtual adversarial training system for DCASE 2019 task 4," *Detection and Classification of Acoustics Scenes and Events (DCASE) Challenge*, 2019.

[17] M. Zöhrer and F. Pernkopf, "Virtual adversarial training and data augmentation for acoustic event detection with gated recurrent neural networks." in *Interspeech*, 2017.

[18] R. Harb and F. Pernkopf, "Sound event detection using weakly-labeled semi-supervised data with GCRNNS, vat and self-adaptive label refinement," *arXiv preprint arXiv:1810.06897*, 2018.

[19] H. Dinkel, X. Cai, Z. Yan, Y. Wang, J. Zhang, and Y. Wang, "A lightweight approach for semi-supervised sound event detection with unsupervised data augmentation," in *Proceedings of the 6th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE), Online*, 2021.

[20] N. Turpault, R. Serizel, A. Parag Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, New York City, United States, October 2019. [Online]. Available: https://hal.inria.fr/hal-02160855

[21] M. Bertola, "Data augmentation methods exploration for sound event detection," DCASE2022 Challenge, Tech. Rep., June 2022.

[22] C. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic, "A framework for the robust evaluation of sound event detection," *arXiv preprint arXiv:1910.08440*, 2019. [Online]. Available: https://arxiv.org/abs/1910.08440

[23] N. Turpault, R. Serizel, and E. Vincent, "Analysis of weak labels for sound event tagging," Apr. 2021, working paper or preprint. [Online]. Available: https://hal.inria.fr/hal-03203692

[24] G. Ferroni, N. Turpault, J. Azcarreta, F. Tuveri, R. Serizel, Ç. Bilen, and S. Krstulović, "Improving sound event detection metrics: insights from DCASE 2020," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.

[25] R. Balestriero, L. Bottou, and Y. LeCun, "The effects of regularization and data augmentation are class dependent," *arXiv preprint arXiv:2204.03632*, 2022.