A LARGE MULTI-MODAL ENSEMBLE FOR SOUND EVENT DETECTION

Technical Report

Heinrich Dinkel‡, Zhiyong Yan‡, Yongqing Wang‡, Meixu Song, Junbo Zhang, Yujun Wang

Xiaomi Corporation, Beijing, China
{dinkelheinrich,yanzhiyong, wangyongqing3,songmeixu,zhangjunbo1,wangyujun}@xiaomi.com

ABSTRACT

This paper is a system description of the XiaoRice team submission to the DCASE 2022 Task 4 challenge. Our method focuses on merging commonly used convolutional neural networks (CNNs) with transformer-based methods and recurrent-neural networks (RNNs). We deliberately divide our efforts into optimizing the two evaluation metrics for the challenge: the onset and offset sensitive PSDS-1 score and the clip-level PSDS-2 score. This work shows that a large ensemble of differently trained architectures and frameworks can lead to significant gains. Our PSDS-1 optimized system consists of an 11-way convolutional recurrent neural network (CRNN), Vision transformer (ViT) fusion, and achieves a PSDS-1 score of 48.19. Further, our PSDS-2 system comprised of a 6-way CNN and ViT fusion achieved a PSDS-2 score of 87.70 on the development dataset.

***Index Terms*—** Semi-supervised learning, Convolutional recurrent neural networks, Transformers, Weakly supervised learning.

1. INTRODUCTION

This paper proposes a system for the DCASE 2022 Task 4 challenge, which is concerned with modeling audio signals for sound event detection (SED). The main objective within SED is to categorize (i.e., tag) an event, with its respective on- and offsets.

Currently, SED can be used for a variety of applications, such as an aid for the hearing impaired, smart cities and homes [1], audio to text retrieval [2], voice activity detection [3, 4] and audio captioning [5, 6, 7, 8]. Most current approaches within SED utilize neural networks, in particular convolutional neural networks [9, 10] (CNN), convolutional recurrent neural networks [11, 12] (CRNN) and other models such as transformers and conformers [13, 14].

In previous versions of the DCASE Task 4 challenge, the usage of external data was forbidden. The DCASE 2022 Task 4 challenge aims to investigate the impact of external data on the sound event detection task in a domestic setting.

The paper is structured as follows. Section 2 describes our core system idea. Further, Section 3 introduces the experimental setup and Section 4 displays our achieved results. Finally, Section 5 concludes the work.

2. SYSTEM

Four systems are allowed to be submitted to the challenge. One of these systems cannot be trained with the help of external data. We, therefore, devised four individual networks, each excelling at a respective task.

1. A network trained without external data denoted as SCRATCH.
2. A small model aimed to reduce the carbon-code denoted as SMALL.
3. An ensemble of models optimized towards excelling at the on- and offset estimation, denoted as PRECISE.
4. An ensemble of models optimized towards coarse-scale audio tagging, denoted as TAG.

The main focus of this work is to use an ensemble of CRNN and Transformer based models. As recent research has suggested [15], CNNs and Transformers have vastly different properties when it comes to modeling data. While CNNs have shown to be high-pass filters, Transformers act as low-pass filters. This leads us to believe that transformers can complement CNNs and vice-versa and thus should be used in tandem to achieve optimal performance.

2.1. Model Pool

In the following, we describe the models we used for the ensembles of our submissions.

- CRNN: This model is identical to the publicly provided baseline, using attention-pooling.
- RCRNN: This model is an enhanced CRNN model, similar to [16].
- CDur: A CRNN model using linear-softmax as its pooling method [11].
- CDur-MR: A variant of CDur, which uses multi-resolution (MR) inputs to two independent gated linear unit (GRU) networks.
- Vit-GRU: A ViT-Tiny model pretrained on Audioset. When finetuning on DESED, we add an additional bidirectional GRU (BGRU) layer to the output of the ViT model. Similar to the baseline CRNN we use attention pooling.
- Efficienet-B0 (EffB0) and MobileNetV2 (MBv2): Standard EfficienetNet/Mobilenet [17, 18] networks. Both models follow the same training as the "EfficientLatent" approach from [19] and downsample the input sequence with a factor of 32.
- EffB0-PSL: A PSL [20] version of the above-mentioned EffB0. The DESED trained EffB0 is used as a teacher to re-estimate labels on a scale of 3s. Then a student (EFFB0-PSL) is trained on these coarse labels.

‡ equal contribution.

- Passt-Tiny: An Imagenet ViT-Tiny [21] model using the patch dropout technique from [22]. The model uses 16×16 patches with a stride of 10×10.
- RCRNN-Passt: A combination of the above RCRNN and Passt-Tiny models. We extract 192 dimensional features from the Passt-Tiny model and stack these over the frequency axis, creating a 960 dimensional embedding, which is appended to each feature frame.

2.2. Training framework

Many techniques exist to utilize unlabeled data to improve model performance. Mean Teacher (MT) [23] is currently the most popular technique used within the DCASE community. However, as it has been shown in previous works [12], unsupervised data augmentation (UDA) [24] can also be used to improve performance. Further previous works [16] have also suggested that noisy-student (NS) is capable of improving performance using unlabeled data. We believe that MT, UDA, and NS can mutually benefit from each other and thus focus on merging models with these three training frameworks for evaluation.

3. EXPERIMENTAL SETUP

3.1. Dataset

The dataset used in this work is the DCASE2022 Task 4 dataset, which focuses on sound event detection in domestic environments.

The DCASE 2022 Task 4 dataset is split into a development (used for training) and an evaluation section. The development set is further split into training and validation sections. The training section contains three datasets $\mathcal{D}_{weak}, \mathcal{D}_{syn}, \mathcal{D}_{un}$:

$$\begin{aligned}\mathcal{D}_{weak} &= \{(x_1, y_2), (x_2, y_2), \ldots, (x_N, y_N)\},\\ \mathcal{D}_{syn} &= \{(x_1, y_2), (x_2, y_2), \ldots, (x_M, y_M)\},\\ \mathcal{D}_{un} &= \{x_1, \ldots, x_P\}.\end{aligned}$$

The $\mathcal{D}_{weak}$ and $\mathcal{D}_{syn}$ datasets are labeled and $\mathcal{D}_{un}$ only consists of audio data in a matching domain with $\mathcal{D}_{weak}$.

3.2. Training hyperparameters

Log Mel-spectrogram (LMS) features are chosen as the default front-end feature for the task. Since the feature extraction parameter differs across models, we provide the hop-size (hop) and the number of Mel filterbanks (# Mels) for each model. Further, we denote use $\mathcal{R}$ to denote each model's output-label frame resolution. During training, if segments are shorter than 10 seconds, we zero-pad the input to the longest sample within a batch. During inference, we use a batch size of 1, such that padding has no effect.

All experiments start with a learning rate of 0.001 and are run for at most 500 epochs, with a linear warmup duration of 20 batches using the Adam optimizer. Batch sizes are set to be 12 for weak and synthetic data and 32 for unlabeled data. The available weak training data is split into a 90% training and a 10% cross-validation portion. Cross-validation is done on the 10% held-out weak subset with the additional synthetic validation data. The training objective is the sum of the weak F1 and the intersection-F1 score, whereas training is stopped if the model did not improve for 15 epochs. Pytorch [25] was used as the neural network back-bone.

For training, we use the standard binary cross-entropy (BCE) criterion. The following losses are employed during training:

$$\mathcal{L}_{\text{sup}} = \text{BCE}(\hat{y}, y), \{y, \hat{y}\} \in \mathcal{D}_{weak}, \tag{1}$$

$$\mathcal{L}_{\text{syn}} = \text{BCE}(\hat{y}_t, y_t), \{y_t, \hat{y}_t\} \in \mathcal{D}_{syn}, \tag{2}$$

$$\mathcal{L}_{\text{UDA}} = \mathcal{L}_{\text{Cstcy}}(\hat{y}^{\dagger}, \hat{y}) + \mathcal{L}_{\text{Cstcy}}(\hat{y}_t^{\dagger}, \hat{y}_t), x \in \mathcal{D}_{un}. \tag{3}$$

$$\mathcal{L}_{\text{MT}} = \mathcal{L}_{\text{Cstcy}}(\hat{y}^{\mu}, \hat{y}) + \mathcal{L}_{\text{Cstcy}}(\hat{y}_t^{\mu}, \hat{y}_t), x \in \mathcal{D}_{un}. \tag{4}$$

$$\mathcal{L}_{\text{unsup}}(x) = \begin{cases}\mathcal{L}_{\text{UDA}}(x) & \text{if UDA}\\ \mathcal{L}_{\text{MT}}(x) & \text{if MT}\end{cases}, \tag{5}$$

where $\hat{y}^{\dagger}$ is the model prediction of an augmented sample $x^{\dagger} = \text{Aug}(x)$ and $\hat{y}^{\mu}$ is the mean-teacher predicted label for a sample. For mean-teachers we follow the public DCASE2022 Task 4 baseline approach, while UDA is applied according to [12]. If not further stated we use BCE as the consistency loss $\mathcal{L}_{\text{Cstcy}}$. Note that NS training is identical to MT training, but updates the teacher model after each experiment. Each network is optimized using the sums of all introduced losses seen in Equation (6).

$$\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{syn}} + \mathcal{L}_{\text{unsup}} \tag{6}$$

3.3. Data Augmentation

If not further stated, we use two main data augmentation methods, namely SpecAug [26] and Mixup [27] to enhance performance. If not otherwise stated, all our approaches use a mixup with the weight $\lambda = \beta(0.5, 0.5)$ drawn from the β distribution.

3.4. Post-processing

If not further stated, we use the default median-filtering approach with a length of 448 ms. We also incorporate an adaptive median filtering scheme, denoted as "+Adapt", where we ran a grid search to find the most optimal median filter size for each respective event class. Note that this adaptive median filter differs between our PRECISE and TAG systems.

3.5. SCRATCH

The SCRATCH models can be seen in Table 1.

ID	Model	Unsup	# Mels	R (ms)	Remark
S1	CRNN	MT	128	64	
S2	RCRNN	MT	128	64	ReLU
S3	RCRNN	MT	128	64	
S4	CRNN	NS	128	64	ReLU
S5	CRNN	NS	128	64	
S6	RCRNN	NS	128	64	
S7	CDur	UDA	64	40	No-Aug
S8	CDur	UDA	64	40	
S9	CDur-MR	UDA	64	40	

Table 1: SCRATCH model architectures. "Unsup" represents the choice of $\mathcal{L}_{\text{unsup}}$. Models with the "ReLU" remark use ReLU instead of GLU as their activation.

Model	Unsup	# Mels	R (ms)
CRNN-Small	MT	128	64

Table 2: SMALL model architectures. "Unsup" represents the choice of $\mathcal{L}_{\text{unsup}}$.

3.6. SMALL

Our SMALL model can be seen in Table 2. It consists of a stripped down version of the baseline CRNN model. More specifically, it only uses 4 convolutional layers and a 32-dimensional hidden state for the RNN model.

3.7. PRECISE

The models used for PRECISE are introduced in Table 3.

ID	Model	Unsup	#Mels	$\mathcal{R}$ (s)	Remark
P1	RCRNN-Passt‡	MT	128	64	
P2	RCRNN	MT	128	64	Scratch
P3	Vit-GRU‡	MT	64	40	Epoch 7
P4	Vit-GRU‡	MT	64	40	Epoch 8
P5	Vit-GRU‡	MT	64	40	Epoch 9
P6	RCRNN-Passt‡	MT	128	64	
P7	CDur	UDA	64	40	
P8	CDur-MR	UDA	64	40	
P9	RCRNN	MT	128	64	ReLU
P10	RCRNN	MT	128	64	Teacher
P11	CRNN	NS	64	40	

Table 3: Parameter settings for the PRECISE models. "Unsup" represents the choice of $\mathcal{L}_{\text{unsup}}$. Models denoted with ‡ were (partially) pretrained on Audioset. Epoch N represents different checkpoints of the model during training.

3.8. TAG

For our PSDS-2 optimized submission, we focus on coarse-scale predictions and thus increase the receptive field size of our baseline model by i.e., increasing the subsampling factor. In general, we use Audioset pretrained models for these systems, since they greatly enhance the performance on the held-out dataset. Our TAG models are described in Table 4. Note that "RCRNN-Passt-S" represents the student and "RCRNN-Passt-T" a teacher model, respectively.

ID	Model	Unsupervised	#Mels	$\mathcal{R}$ (s)
T1	EffB0	UDA	64	10
T2	EffB0-PSL	UDA	64	10
T3	MBv2	UDA	64	10
T4	Passt-Tiny	UDA	64	10
T5	RCRNN-Passt-S	MT	128	0.32
T6	RCRNN-Passt-T	MT	128	0.32

Table 4: Introduction to the models used for TAG. Unsupervised represents the choice of $\mathcal{L}_{\text{unsup}}$.

All TAG models have been (in part or completely) pretrained on Audioset. The Audioset performance can be observed in Table 5.

Model	mAP
EffB0	44.08
MBv2	42.15
Passt-Tiny	43.20

Table 5: Results of our pretrained TAG models on Audioset.

3.9. Ensemble

One of our main goals is to ensemble a multitude of different networks, each possibly producing outputs at a different resolution i.e., each 64 ms or 40 ms. In order to average these predictions, we linearly upsample all model predictions to the highest resolution within an ensemble. Postprocessing is applied after score averaging.

4. RESULTS

We report our results in terms of the two main challenge metrics denoted as PSDS-1 and PSDS-2 [28]. Note that all results represent the performance on the held-out official development dataset.

4.1. System-1 (SCRATCH)

The results regarding our system-1 submission can be seen in Table 6.

ID	**PSDS-1**	**PSDS-2**
Baseline	33.60	53.60
S1	39.19	61.93
S2	**42.06**	65.42
S3	41.48	63.15
S4	38.95	62.32
S5	40.66	64.14
S6	41.29	64.99
S7	35.60	62.56
S8	38.92	**66.20**
S9	39.09	64.74
SCRATCH	**45.65**	**71.36**

Table 6: Results for our SCRATCH system, where no external data is used. Best results in bold.

4.2. System-2 (SMALL)

The results of our system-2 submission can be seen in Table 7.

Model	**PSDS-1**	**PSDS-2**	Energy (kWh)
Baseline	33.60	53.60	0.030
SMALL	39.50	63.08	0.025

Table 7: Results for our SMALL model, emphasizing a low energy consumption.

4.3. System-3 (PRECISE)

Results regarding our proposed system-3 optimized, towards PSDS1 scores can be seen in Table 8.

ID	**PSDS-1**	**PSDS-2**
Baseline	33.60	53.60
P1	41.16	61.80
P2	41.08	60.68
P3	39.07	66.57
P4	39.05	**66.58**
P5	38.89	64.64
P6	38.20	63.46
P7	37.98	64.48
P8	38.04	63.48
P9	**42.52**	64.44
P10	41.96	62.40
P11	41.14	63.26
PRECISE	48.01	75.01
PRECISE + Adapt	**48.19**	**75.73**

Table 8: Results for our PRECISE model, emphasizing accurate on- and offsets. Best results are displayed in bold.

4.4. System-4 (TAG)

Our results for the TAG model can be seen in Table 9. If during testing clips longer than 10s are provided we split these samples into 10s chunks and individually estimate scores for each chunk.

ID	**PSDS-1**	**PSDS-2**
Baseline	31.30	72.20
T1	9.07	86.25
T2	9.01	86.87
T3	10.13	84.62
T4	8.25	85.62
T5	**15.63**	82.79
T6	12.21	82.63
TAG	12.56	87.64
TAG + Adapt	12.67	**87.70**

Table 9: Results for our TAG model (submission 4), focusing on coarse performance. Best models for each respective metric are in bold.

5. CONCLUSION

This paper proposes our submission to the DCASE2022 Task4 challenge. Our approach is based on a large scale ensemble between different types of network architectures and training regimes. Each of our four submitted systems is optimized towards a different goal. The SCRATCH system is trained without additional data and obtains a PSDS-1 score of 45.56 and PSDS-2 score of 71.36. Our carbon-footprint optimized SMALL system achieves a PSDS-1/2 score of 39.50 and 63.08, respectively. The PRECISE ensemble method obtains a PSDS-1 score of 48.19, outperforming the baseline (33.6) by a significant margin. Finally our coarse TAG ensemble network achieves a PSDS-2 score of 87.70, exceeding the baseline performance of 72.2.

6. REFERENCES

[1] J. P. Bello, C. Mydlarz, and J. Salamon, *Sound Analysis in Smart Cities*. Cham: Springer International Publishing, 2018, pp. 373–397. [Online]. Available: https://doi.org/10.1007/978-3-319-63450-0{_}13

[2] S. Lou, X. Xu, M. Wu, and K. Yu, "Audio-text retrieval in context," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 4793–4797.

[3] Y. Chen, H. Dinkel, M. Wu, and K. Yu, "Voice activity detection in the wild via weakly supervised sound event detection," *Proc. Interspeech 2020*, pp. 3665–3669, 2020.

[4] H. Dinkel, S. Wang, X. Xu, M. Wu, and K. Yu, "Voice Activity Detection in the Wild: A Data-Driven Approach Using Teacher-Student Training," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1542–1555, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9405474/

[5] X. Xu, H. Dinkel, M. Wu, and K. Yu, "Audio caption in a car setting with a sentence-level loss," in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2021, pp. 1–5.

[6] X. Xu, H. Dinkel, M. Wu, Z. Xie, and K. Yu, "Investigating local and global information for automated audio captioning with transfer learning," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 905–909.

[7] M. Wu, H. Dinkel, and K. Yu, "Audio caption: Listen and tell," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 830–834.

[8] X. Xu, M. Wu, and K. Yu, "Diversity-controllable and accurate audio captioning based on neural condition," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 971–975.

[9] L. Lin, X. Wang, H. Liu, and Y. Qian, "Specialized Decision Surface and Disentangled Feature for Weakly-Supervised Polyphonic Sound Event Detection," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 28, pp. 1466–1478, may 2020. [Online]. Available: http://arxiv.org/abs/1905.10091

[10] J. Yan, Y. Song, L.-R. Dai, and I. McLoughlin, "Task-Aware Mean Teacher Method for Large Scale Weakly Labeled Semi-Supervised Sound Event Detection," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings 2020*. Institute of Electrical and Electronics Engineers (IEEE), apr 2020, pp. 326–330.

[11] H. Dinkel, M. Wu, and K. Yu, "Towards duration robust weakly supervised sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 887–900, 2021.

[12] 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 2021), Online, 2021, pp. 15–19.

[13] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Sound Event Detection of Weakly Labelled Data with CNN-Transformer and Automatic Threshold Optimization," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 28, pp. 2450–2460, 2020.

[14] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Convolution-augmented transformer for semi-supervised sound event detection," DCASE2020 Challenge, Tech. Rep., June 2020.

[15] N. Park and S. Kim, "How do vision transformers work?" *arXiv preprint arXiv:2202.06709*, 2022.

[16] N. K. Kim and H. K. Kim, "Self-training with noisy student model and semi-supervised loss function for dcase 2021 challenge task 4," DCASE2021 Challenge, Tech. Rep., June 2021.

[17] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[19] J. Turian, J. Shier, H. R. Khan, B. Raj, B. W. Schuller, C. J. Steinmetz, C. Malloy, G. Tzanetakis, G. Velarde, K. McNally, *et al.*, "Hear 2021: Holistic evaluation of audio representations," *arXiv preprint arXiv:2203.03022*, 2022.

[20] H. Dinkel, Z. Yan, Y. Wang, J. Zhang, and Y. Wang, "Pseudo strong labels for large scale weakly supervised audio tagging," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 336–340.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[22] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," *arXiv preprint arXiv:2110.05069*, 2021.

[23] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 1195–1204.

[24] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised Data Augmentation for Consistency Training," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2019, pp. 6256–6268. [Online]. Available: http://arxiv.org/abs/1904.12848

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8026–8037.

[26] D. S. Park, W. Chan, Y. Zhang, C. C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2019-September. International Speech Communication Association, 2019, pp. 2613–2617.

[27] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb

[28] C. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic, "A Framework for the Robust Evaluation of Sound Event Detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, may 2019, pp. 61–65. [Online]. Available: http://arxiv.org/abs/1910.08440