ATST SELF-SUPERVISED PLUS RCT SEMI-SUPERVISED SOUND EVENT DETECTION: SUBMISSION TO DCASE 2022 CHALLENGE TASK 4

Technical Report

Nian Shao, Xian Li, Xiaofei Li

Westlake University & Westlake Institute for Advanced Study, Hangzhou, China
{shaonian, lixian, lixiaofei}@westlake.edu.cn

ABSTRACT

In this report, we present our methods proposed for participating the *Detection and Classification of Acoustic Scenes and Events (DCASE) 2022 Challenge Task 4: Sound Event Detection in Domestic Environments.* The proposed methods integrate a semi-supervised sound event detection model (called random consistency training, RCT) trained with the relatively small official dataset of the challenge, and a self-supervised model (called audio teacher-student transformer, ATST) trained with the very large AudioSet. RCT uses the baseline convolutional recurrent neural network (CRNN) of the challenge, and adopts a newly proposed semi-supervised learning scheme based on random data augmentation and a self-consistency loss. To integrate ATST into RCT, the feature extracted by ATST is concatenated with the feature extracted by the convolutional layers of RCT, and then fed to the RNN layers of RCT. It is found that these two types of feature are complementary and the performance can be largely improved by combining them. In development, RCT individually achieves 39.80% and 61.12% of $PSDS_1$ and $PSDS_2$, respectively, which are improved to 45.99% and 70.65% by integrating the ATST feature, and further to 47.71% and 73.44% by ensembling five models with different training configurations.

Index Terms— Sound event detection, self-supervised learning, audio pretraining, semi-supervised learning, consistency training, data augmentation

1. INTRODUCTION

Polyphonic sound event detection (SED) suffers from the data deficiency [1] problem for a long time. One possible solution to mitigate such problem is to leverage abundant external sources. In previous challenges, weakly-labeled real, strongly-labeled synthesized and unlabeled sound clips are utilized for SED model training. A baseline convolutional recurrent neural networks (CRNNs) model with the MeanTeacher-based semi-supervised learning (SemiSL) scheme [2] is proposed to leverage these three sorts of data in model training. Although the unsupervised audio clips used in the baseline system do not have labels, they still include some prior knowledge that only the sound events of interest are present in each audio clip. And thus, the unsupervised data in the DESED dataset [3] are all in-domain unlabeled data.

In this year, out-domain unlabeled sound clips (e.g. AudioSet [4]) are also allowed to be utilized. Unlike the in-domain ones, these audio clips contains various acoustic scenes and events that may beyond the scope of the DESED dataset. Such variety poses obstacles on directly using these out-domain data for SED system training. Instead, we could utilize these out-domain data individually with the help of self-supervised learning (selfSL) methods. Although most SelfSL models are designed for only clip-level audio tasks [5–7], we find that they could also perform well on the frame-level SED task and fits well with SemiSL methods.

As for SemiSL, we apply a random consistency training (RCT) scheme [8] in addition to the MeanTeacher scheme of the baseline system. RCT trains the CRNN model with a random data augmentation scheme and an extra self-consistency loss. As for random data augmentation, we apply two types of data augmentation to each training sample, which are *hard mixup* [8], and a randomly selected one from a total of four types of audio warping methods, including *time mask*, *frequency mask* [9], *frame shift* [10] and *filter augmentation* [11]. The extra self-consistency loss is used to constrain the predictions for an audio clip and the augmented version of the audio clip to be identical. Such loss function is compatible with the MeanTeacher loss, and is able to stabilize the training process.

As for SelfSL, we adopt the audio teacher-student transformer (ATST) model [7], trained with the AudioSet [4]. ATST utilizes a transformer encoder [12] to extract an embedding for each audio clip. The training of ATST follows the concept of contrastive learning [13, 14], where the model is trained to classify two seriously warped creations (two positive samples) from the same audio clip as the same class. ATST is designed for clip-level audio processing by introducing a special clip-level classification token, which accumulates information from the frame-level embeddings. To adapt this model for the frame-level SED task, we only use the frame-level embeddings in this work.

To integrate the ATST embeddings into the baseline CRNN model, we concatenate the ATST embeddings with the output of CNN layers as a fused feature, and feed it to the following RNN layers. From our experiments, we observe that, when cascaded with RNN layers, although solely using the ATST embeddings or CNN outputs could achieve reasonable results, their combination can further improve the SED performance.

2. PROPOSED METHOD

2.1. Baseline CRNN model for SED

To better illustrate our method, we would like to briefly introduce the baseline CRNN model. The official DESED dataset contains three sorts of data: weakly labeled real data, strongly labeled synthetic data and unlabeled data. The provided baseline CRNN model [2] sets up different mechanisms to take advantages of these datasets in the training process. The baseline model uses a 7-layer CNNs model with context-gate activation to extract the frame-level fea-

tures from the input audio clips, which are then fed to a 2-layer bidirectional GRU network to model long-term dependencies. The RNNs model outputs the frame-level classification results (corresponding to strong labels). And the clip-level classification results (corresponding to weak labels) is obtained by applying an attention module [2] to the frame-level results. The baseline system utilizes MeanTeacher [15] to tackle the unlabeled data, in which the student model is trained to give consistent predictions with the pseudo labels given by the teacher model.

2.2. RCT for semi-supervised learning

In our system, we apply RCT [8] together with MeanTeacher. Two core techniques are included in the RCT: the random data augmentation scheme and the self-consistency loss.

RCT augments each audio clip in two ways. The first one is hard mixup, where each audio clip is added with another one or two audio clips. And their corresponding labels are combined together by taking the logical OR of them. This way, all the sound events present in original audio clips will be considered as concurrent sound events in the mixed audio clip. The other way of data augmentation is to warp the audio clip. Four types of warping are used, including time shift [10], time mask, frequency mask [9] and filter augmentation [11]. One of the four types of data augmentation is randomly selected and applied to each audio clip at training. Therefore, by applying both hard mixup and random audio warping, the training batch size would be tripled after RCT data augmentation.

RCT also proposes a self-consistency loss, which constrains the model prediction for the original and augmented audio clips to be identical, by minimizing the mean square error (MSE) between the predictions. Different from the MeanTeacher loss, it does not freeze the gradient of either side. As a result, the model would learn to give consistent representations for different data augmentations. Such self-consistency constraint between the original and augmented samples always holds regardless of the correctness of the predictions, and thus is able to stabilize the training process.

2.3. ATST for self-supervised learning

We apply self-supervised learning method, ATST [7], to train a feature extraction network using the very large AudioSet [4]. We expect that this could further strengthen the representation ability of the audio feature.

ATST utilizes a transformer encoder [12], and is designed for clip-level audio tasks. In addition to the frame-level tokens, an extra classification token is created to represent the entire clip. The clip-level token accumulates information from frame-level representations, and only the clip-level token is used for self-supervised learning. ATST [7] follows the concept of teacher-student contrastive learning [13]. Same as MeanTeacher [15], during training, ATST holds an exponential moving average of the transformer encoder (the student network) as a teacher model. The training target for the student model is to represent an audio clip identically as the teacher model does. To increase the difficulties of this training target, two different views of one audio clip are created for the student and teacher model, respectively. In experiment, each 10-second audio clip is first mixed up with two different clips separately, and then randomly resized and cropped into two 6-second segments. A proportion of temporal overlap for the two views is guaranteed in cropping, which holds the rationality of identifying these two views as a positive pair. In training, an extra feedforward layer is added at the end of the student model, to avoid the model collapse [13]. The MSE between the output of the clip-level classification token from the student and teacher models is taken as the self-supervised training loss. We encourage the readers to refer to the original paper of ATST [7] for a better understanding.

The performances of ATST on several clip-level audio processing tasks are promising, which indicates its strong representation ability for audio signals.

2.4. Combining RCT and ATST

As a clip classification model, ATST outputs a clip-level classification token. Nevertheless, we find that this clip-level representation would drastically hurt the SED model performances, possibly because of the lack of temporal information. Therefore, we take the frame-level embeddings of ATST as the features, which is obtained by averaging the hidden units of the last two ATST transformer blocks. Note that ATST training only considers the clip-level classification loss, and the frame-level embeddings are not directly used in the ATST training.

We first consider to replace the CNNs outputs/features of the baseline CRNN model with ATST features, and cascade the ATST network with the RNN layers, referred to as ATST-RNN. The ATST parameters can be either frozen or finetuned during the training of the downstream SED task. However, we could only finetune the last three transformer blocks of the ATST model, since finetuning all layers would lead to serious overfitting at the very early stage of training.

Furthermore, we deem that the feature extracted by the CNNs layers could be compatible with the ATST feature. Consequently, we concatenate the ATST feature and the CNN feature, and then feed the combination to the following RNN layers. This scheme is referred to as ATST-CRNN, and is used in our final submission systems. There would be a problem if the CNN layers are trained from scratch, since the model would be trained to rely more on the pretrained ATST features. As a result, the CNN layers will be insufficiently trained, which leads to some performance decaying. To mitigate this problem, the training strategy of ATST-CRNN is setup with the following steps:

- Train ATST using AudioSet;
- Train CRNN using DESED, with RCT;
- Concatenate ATST feature with CNN feature, and feed to new RNN layers (with randomly initialized parameters);
- Retrain the whole model, using DESED, with RCT; The last three transformer blocks of ATST and CNN layers are finetuned.

3. EXPERIMENT

3.1. Feature extraction and training settings

All the audio clips are first re-sampled to 16kHz. As for the baseline CRNN model (submitted system 1), each audio clip is transformed into the short-time Fourier transform (STFT) domain with 2048 window length and 256 hop length. Then, 128-dimensional log-mel feature is extracted as the network input. Each 10-second sound clip is represented as a 626×128 mel-scale spectrogram. The output dimension of CNN layers is 128, and the hidden size of RNN layers is 256. As for the CNN branch and ATST branch in

Table 1: Development set performance of models trained with or without RCT. All the ATST models are the *small* version. In this experiment, the last three transformer blocks are finetuned for ATST-RNN, while they are frozen for ATST-CRNN.

Model	**PSDS**$_1$ (%)	**PSDS**$_2$ (%)
baseline CRNN	36.65	56.57
baseline CRNN + RCT	39.80	61.12
ATST-RNN	40.65	63.15
ATST-RNN + RCT	45.11	68.28
ATST-CRNN	41.42	63.12
ATST-CRNN + RCT	44.28	66.82

Table 2: Development set performance of ATST-RNN model with freezing or finetuning the last three transformer blocks of ATST. We use the *small* version of ATST. RCT is not used.

Pretraining strategy	**PSDS**$_1$ (%)	**PSDS**$_2$ (%)
ATST-RNN + freezing	39.52	63.52
ATST-RNN + finetuning	40.65	63.15

ATST-CRNN (submitted systems 2-4), the window length of STFT are set to 2048 and 1024, and the dimension of log-mel feature are set to 128 and 64, respectively. The hop length for both branches is set to 160. Each 10-second sound clip is represented in the dimension of 1000×128 for the CNN branch and 1000×64 for the ATST branch. There are two versions of ATST pre-trained model, i.e. so-called *small* and *base* ATST models. The only difference between them is that the *small* and *base* versions have a hidden dimension of 384 and 768, respectively. Correspondingly, the fused feature in ATST-CRNN has a dimension of 512 for the *small* version (128 from CNN + 384 from ATST), and 896 for the *base* version (128 from CNN + 768 from ATST).

The training configurations of ATST follow the settings presented in the ATST paper [7]. The training configurations of CRNN or ATST-CRNN follow the settings presented in the RCT paper [8]. The hidden sizes of RNN layers in CRNN and ATST-CRNN are set to 256 and 512, respectively. The temperature technique [16] is adopted at the inference stage. And the temperature for all models is set to 3. The prediction result is post-processed using the median filters presented in [8]. We take both $PSDS_1$ and $PSDS_2$ [17] as the SED performance metrics. The performances in the following experiments are all applied temperature post-processing except for the baseline CRNN models.

3.2. Performance of RCT for semi-supervised SED

We apply the RCT scheme to the baseline CRNN, ATST-RNN and ATST-CRNN models, and the results are shown in Table 1. We can observe that RCT is compatible with all the three models, as it could lead to an at least 3% improvement in both PSDS metrics. This verifies that RCT can have a better usage of the unlabeled data, compared with the baseline MeanTeacher method.

3.3. Performance for combining ATST model

As for the usage of ATST features, we first test the ATST-RNN model, as shown in Table 2. By replacing the CNN layers with the ATST model, ATST-RNN achieve better performance measures compared with the baseline CRNN system, even when the ATST model is frozen. This testifies that the ATST model pre-trained with an out-domain dataset is indeed able to extract an meaningful audio representation.

Table 3: Development set performances of ATST-CRNN models trained with different strategies. The last three transformer blocks of ATST are always finetuned. RCT is used.

Model	**PSDS**$_1$ (%)	**PSDS**$_2$ (%)
small ATST-RNN	45.11	68.28
small ATST-CRNN + scratch CNN	43.10	64.98
small ATST-CRNN + freezing CNN	45.77	68.24
small ATST-CRNN + finetuning CNN	46.04	69.75
base ATST-CRNN + finetuning CNN	45.99	70.65

Table 4: Final submitted system configurations as well as their performance on the development set. TS, TM, FA and FM stand for time shift, time mask, filter augmentation and frequency mask, respectively.

Submissions	**Model**	**RCT warping methods**
System 1	baseline CRNN	TS, TM, FA, FM
System 2	*small* ATST-CRNN	TS, TM, FA
System 3	*base* ATST-CRNN	TS, TM, FA
System 4	*base* ATST-CRNN	multiple strategies

Submissions	**PSDS**$_1$ (%)	**PSDS**$_2$ (%)
System 1	39.80	61.12
System 2	46.04	69.75
System 3	45.99	70.65
System 4	47.71	73.44

We then evaluate the concatenation of ATST feature and CNN feature. In experiments, we investigate the usage of CNN feature by either train-from-scratch or pre-trained then frozen or finetuned. The results are shown in Table 3. We could find that, the models with pre-trained CNN feature performs better than that with train-from-scratch feature. With pre-trained and finetuned CNN feature, ATST-CRNN noticeably outperforms ATST-RNN, which indicates that the ATST feature and CNN feature are compatible to some extent. The performance can be slightly improved when replacing the *small* ATST model with the *base* one.

3.4. Challenge submissions

From above experiments, we find the following facts: RCT is an effective semi-supervised learning strategy, combining ATST feature and CNN feature is helpful, and proper model finetuning is necessary for both the pre-trained ATST model and CNN layers. By adopting RCT, ATST/CNN feature combination and model finetuning, we propose our four submitted systems as: (i) baseline CRNN; (ii) *small* ATST-CRNN; (iii) *base* ATST-CRNN; (iv) An ensemble of five *base* ATST-CRNN systems. The detailed configurations are shown in Table 4. System 4 ensembles 5 different systems varying in training scheme and RCT audio warping strategies. To maximize the differences among these models, in addition to system 3, we

train four extra models with different freezing or finetuning strategies.

We also notice that, the ATST model is ineffective for processing very long audio clips, due to the computational complexity of the computation of self attention. In evaluation, we split those clips that are longer than 10 seconds into 10-second chunks with 2-second overlaps. The model predictions for these 10-second clips are then re-unified with the logical OR operation. And then median filters are applied for post-processing.

4. CONCLUSION

To conclude, in this work, we integrate the semi-supervised sound event detection model (RCT), and the self-supervised model (ATST). These two techniques allow us to use both in-domain and out-domain data sources, which brings significant performance improvements on the baseline CRNN model.

5. REFERENCES

[1] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational Analysis of Sound Scenes and Events*. Springer, 2018.

[2] L. JiaKai, "Mean teacher convolution system for dcase 2018 task 4," DCASE2018 Challenge, Tech. Rep., June 2018.

[3] N. Turpault, R. Serizel, A. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *DCASE2019*, 2019, p. 253.

[4] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*. IEEE, 2017, pp. 776–780.

[5] E. Fonseca, D. Ortego, K. McGuinness, N. E. O'Connor, and X. Serra, "Unsupervised contrastive learning of sound event representations," in *ICASSP*. IEEE, 2021, pp. 371–375.

[6] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Byol for audio: Self-supervised learning for general-purpose audio representation," in *IJCNN*. IEEE, 2021, pp. 1–8.

[7] X. Li and X. Li, "ATST: Audio representation learning with teacher-student transformer," in *Interspeech*, 2022.

[8] N. Shao, E. Loweimi, and X. Li, "RCT: Random consistency training for semi-supervised sound event detection," in *Interspeech*, 2021.

[9] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech*, 2019.

[10] C.-Y. Koh, Y.-S. Chen, Y.-W. Liu, and M. R. Bai, "Sound event detection by consistency training and pseudo-labeling with feature-pyramid convolutional recurrent neural networks," in *ICASSP*. IEEE, 2021, pp. 376–380.

[11] H. Nam, S.-H. Kim, and Y.-H. Park, "Filteraugment: An acoustic environmental data augmentation method," in *ICASSP*. IEEE, 2022, pp. 4308–4312.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, p. 6000–6010.

[13] X. Chen and K. He, "Exploring simple siamese representation learning," in *CVPR*, June 2021, pp. 15 750–15 758.

[14] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *ICASSP*. IEEE, 2021, pp. 3875–3879.

[15] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *NIPS*, vol. 30, p. 1195–1204, 2017.

[16] X. Zheng, H. Chen, and Y. Song, "Zheng ustc team's submission for dcase2021 task4 – semi-supervised sound event detection," DCASE2021 Challenge, Tech. Rep., June 2021.

[17] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulović, "A framework for the robust evaluation of sound event detection," in *ICASSP*. IEEE, 2020, pp. 61–65.