# Multi-task Learning with Variational Autoencoders for Semi-supervised Sound Event Detection

Petros Giannakopoulos, Aggelos Pikrakis

***Abstract*—In this work we present a multi-task learning model, based on recurrent variational autoencoders (VAEs), for semi-supervised sound event detection. The proposed method employs recurrent VAEs with shared parameters to simultaneously learn the tasks of strong labeling, weak labeling and feature sequence reconstruction. During the training stage, the model receives as input strongly labeled, weakly labeled, and unlabeled data. It simultaneously optimizes frame-based and clip-based cross-entropy losses for strongly labeled and weakly labeled data, respectively, as well as the reconstruction loss for the unlabeled data. Using a shared posterior among all task branches, the model projects the input data for each task into a common latent space. The decoding of latents sampled from this common latent space, in combination with the shared parameters among task branches act jointly as a regularizer that prevents the model from overfitting to the individual tasks. When evaluated on the DCASE-Task4 2022 dataset, our proposed semi-supervised learning method achieves an event-based macro F1 score of 31.8% on the public evaluation set, versus 12.4% achieved by pure supervised learning. It also achieves a segment-based macro F1 score of 60.6% versus 38% achieved by pure supervised learning.**

***Index Terms*—sound event detection, multi-task learning, variational autoencoder, semi-supervised learning**

## I. Introduction

SOUND Event Detection (SED) is the process of identifying sounds in the environment, such as a human speaking, a dog barking, a vaccum cleaner etc. [1]. Figure 1 provides an overview of an SED system used to automate this process. Besides the understanding of the environment that an SED system provides, it can also be used as feedback to other systems that are capable of taking actions, as it is the case with the triggering of an alarm.

### A. Neural-based Sound Event Detection Systems

In recent years, neural networks have contributed to notable improvements in the performance of SED systems. Convolutional Neural Networks (CNNs) [2], [3], Recurrent Neural Networks (RNNs) [4], [5], Convolutional RNNs (CRNNs) [6], [7] and Transformers [8], [9], [10] have been used with success as the backbone of SED systems. The main drawback of neural-network based approaches is that a large amount of labeled data is required during a supervised training stage. There are two main SED variations, i.e., *strong* and *weak* audio event tagging. In the case of *strong* tagging, an SED system must detect both the audio event type *and* the respective endpoints. In the case of *weak* tagging, the SED system must only detect the presence of the audio event. The *strong* tagging task requires audio data to be annotated with timestamps that provide the beginning and end of each audio event occurrence, as shown in Figure 1. This type of data, known as *strongly labeled data*, are difficult, time-consuming and costly to collect in amounts that are sufficient to effectively train neural-network based approaches via supervised learning. Emphasis has therefore been placed on developing training methods which reduce the requirements for strongly annotated data, while remaining effective. These range from simple data augmentation techniques to weakly-supervised and semi-supervised learning methods. Data augmentation has proved to be an effective technique to improve the generalization capabilities of SED models by performing random or targeted processing on existing data to artificially generate new data samples [2], [3]. Furthermore, several SED model architectures and training schemes have been proposed which can take advantage of *weakly labeled* and/or *unlabeled* data to improve generalization while reducing the requirements for *strongly labeled* data [8], [11], [12].

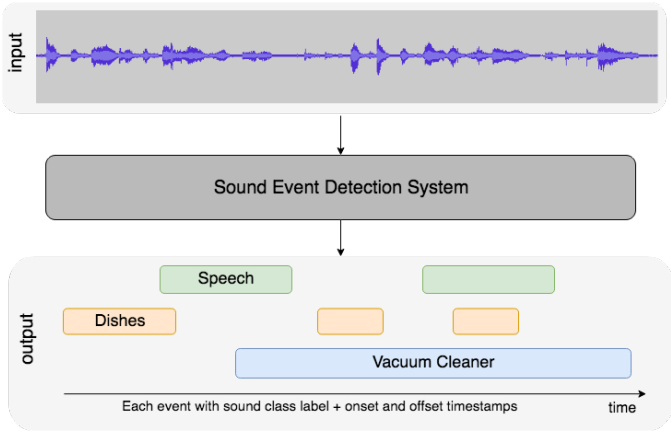


Fig. 1. Overview of a Sound Event Detection (SED) system. An SED system receives audio as input and outputs annotations of the audio events present in the audio. Each annotation usually contains the audio event label along with the beginning (onset) and end (offset) timestamps of the event.

### B. Multi-Task Learning

Multi-Task Learning (MTL) [13] is a method where a model can learn to solve multiple tasks simultaneously, while exploiting possible common characteristics and differences

Petros Giannakopoulos is a PhD candidate at the Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece (e-mail: petrosgk@di.uoa.gr).

Aggelos Pikrakis is an Assistant Professor at the Department of Informatics, University of Pireaus, Greece (e-mail: pikrakis@unipi.gr).

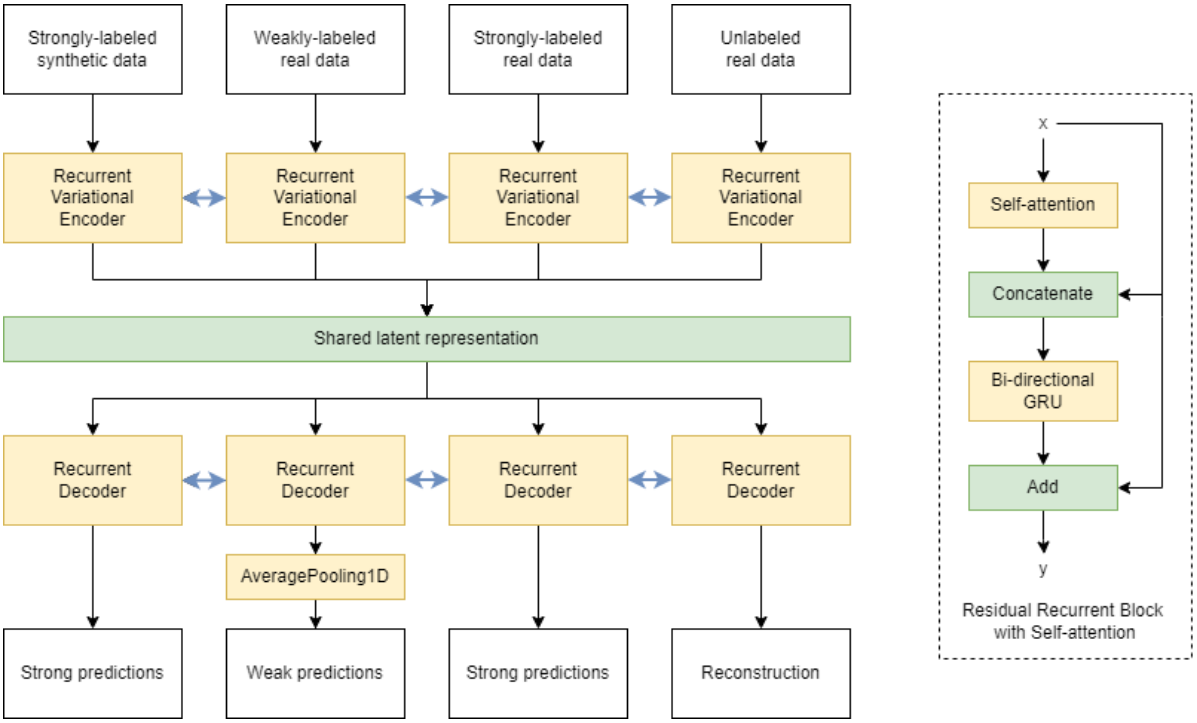


Fig. 2. Overview of the architecture of our proposed MTL-VAE architecture. Input feature sequences for each task are encoded into a shared latent representation by variational encoders with shared weights. Decoders with shared weights perform the tasks of audio event classification, at a frame-level and clip-level, as well as frame-level audio feature reconstruction. Each encoder and each decoder is constructed from stacks of residual recurrent blocks with self-attention, where $x$ and $y$ are the input and output sequences respectively.

across them. Such a model can achieve improved performance on each individual task compared to a model that learns to solve each problem in isolation. MTL has been applied to the domain of weakly-supervised and semi-supervised SED [14], [15], [16] with promising results. Previous works have combined MTL with Variational Auto-Encoders (VAEs) [17], [18], [19], [20] and showed that, in the domains of image and sentiment classification, projecting input features for each task into latent representations sampled from the posterior of a variational encoder can improve regularization of shared features for downstream tasks and is more robust to noise and outliers in the input features.

### C. Contribution

In this work we propose an SED model, based on the MTL-VAE principle and RNNs, trained in a semi-supervised manner. We achieve this by simultaneously training the model on three audio event tagging tasks, each having its own dataset as provided by DCASE-Task4 2022 [21], [22]: strong tagging on synthetic audio data, weak tagging on real audio data and strong tagging on real audio data. The model is also simultaneously trained on a fourth task: reconstruction of unlabeled audio features. We demonstrate that the model is able to leverage cross-task information to achieve superior performance on the task of strong audio event tagging on real data, which is the task of interest, compared to the case when it is trained on this task without MTL. We also demonstrate that using a VAE architecture improves generalization performance. Our MLT-VAE SED model achieved a $32.5\%$ event-based macro F1 score and a $60.6\%$ segment-based macro F1-score on the DCASE-Task4 2022 validation set. On the public evaluation set it achieved a $31.8\%$ event-based macro F1-score and a $60.6\%$ segment-based macro F1-score. We did not use any data augmentation during training.

## II. Proposed Method

In this section we: A) define the architecture of our MTL-VAE model, B) describe how we train the MTL-VAE model, for the purpose of SED, in a semi-supervised manner, and C) outline the evaluation method of the trained MTL-VAE SED system.

### A. Network architecture

The proposed MTL-VAE architecture for SED is demonstrated in Figure 2. It consists of a variational encoder for each task input and all encoders share weights. The resulting outputs of the variational encoders are shared stochastic latent representations of the input feature sequences of all downstream tasks. The latent representations are inputs to decoders with shared weights, with each decoder being responsible for a respective task. Each decoder is followed by a classification head which outputs either frame-level predictions for the *strong* audio event tagging or clip-level predictions for the *weak* tagging tasks. The classification head consists of a feed-forward layer, with number of units equal to the number of audio event classes, followed by a sigmoid activation function. Therefore, the predictions are the probability for each audio event class being present in the frame or clip. The outputs of

TABLE I
COMPOSITION OF DCASE-TASK4 2022 TRAINING SET

| Data | Audio Clips |
|---|---|
| Synthetic with strong labels | 10,000 |
| Real with weak labels | 1,578 |
| Real without labels | 14,412 |
| Real with strong labels (Audioset) | 3,195 |

the decoder responsible for the *weak* tagging task are averaged over the sequence length, via an *AveragePooling1D* operation, to obtain clip-level predictions. The decoder responsible for the frame-level reconstruction task is followed by a feature sequence reconstruction head. The reconstruction head consists of one feed-forward layer per frame in the feature sequence, with number of units equal to the number of features, followed by a linear (identity) activation function.

All encoders and decoders are made up of residual recurrent blocks with self-attention. The input sequence $x$ to each block is passed through a self-attention layer [23]. The resulting attention weights are concatenated with $x$ and passed through bi-directional Gated Recurrent Unit (GRU) layers [24]. The recurrent layers outputs are added together with input $x$, resulting in the final output sequence $y = f(x)$ of the block.

### B. Training procedure

*1) Data pre-processing:* The training set of the DCASE-Task4 2022 dataset contains: a) 10,000 synthetic audio clips with strong labels, b) 1,578 real audio clips with weak labels, c) 14,412 real audio clips without labels, and d) 3,195 real audio clips with strong labels. The training set composition is summarized in Table I. The validation set of the DCASE-Task4 2022 dataset contains 1,152 real audio clips with strong labels, used for evaluating the accuracy of the model on the task of *strong* audio event tagging. Since no explicit validation sets are provided for the other types of data, we withheld $5\%$ of the audio clips of each data type as validation data for the task associated with that data type. For the data pre-processing stage, we first converted all audio clips into a 1-channel, 16-bit format, at a 16 kHz sampling rate. For each audio clip, we then removed the direct current (DC) component and normalized the loudness of the audio to -3 dBFS [25]. We then extracted a spectrogram from each audio clip using an FFT window length of 2048 samples and a hop length of 384 samples, resulting in a feature sequence of 417 frames for a 10-second audio clip. We performed zero-padding of shorter audio clips, where needed. The inputs to our proposed model are 128-dimensional log-mel filterbanks that we extracted from each spectrogram. We then performed zero-mean and unit variance normalization over the feature sequences in the training data.

*2) Training objective:* For the concurrent training on all four tasks, the final objective $L$ that the model must optimize for is the sum of four objectives, one for each task. Specifically: a) frame-level cross-entropy for the *strong synthetic audio event tagging* task ($BCE_s$), b) clip-level cross-entropy for the *weak real audio event tagging* task ($BCE_w$), c) frame-level cross-entropy for the *strong real audio event tagging* task ($BCE_s$), and d) L2 reconstruction error for the *unlabeled data reconstruction* task ($L2_u$). To that sum we must add e) the KL-divergence objective ($KLD$) between the posterior of the VAE and a Gaussian prior $N(0, 1)$. The $KLD$ term is added to the final objective with a weight $\beta$. This is done in order to be able to control the strength of the regularization applied to the posterior, similar to [26]. Based on the above, the final objective is:

$$L = 2 * BCE_s + BCE_w + L2_u + \beta * KLD \tag{1}$$

with:

$$BCE_s = \frac{1}{T}[y_s \log x_s + (1 - y_s) \log x_s] \tag{2}$$

where $x_s$, $y_s$ are sequences of label vectors with shape $T \times classes$,

$$BCE_w = y_w \log x_w + (1 - y_w) \log x_w \tag{3}$$

where $x_w$, $y_w$ are label vectors of length $classes$,

$$L2_u = (x_u - y_u)^2 \tag{4}$$

where $x_u$, $y_u$ are sequences of feature vectors with shape $T \times dims$,

$$KLD = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)} \tag{5}$$

where $X = x_s \cup x_w \cup x_u$, $P(x)$ a distribution over strong labeled, weak labeled, and unlabeled data, and $Q(x)$ a Gaussian distribution prior.

*3) Training parameters:* We use the Adam optimizer [27] with a learning rate of $5 * 10^{-4}$ and a batch size of 32. Each batch contains randomly sampled synthetic audio data with strong labels, real audio data with weak labels, real unlabeled audio data, and real audio data with strong labels from the Audioset dataset [28] in a 1:1:1:1 ratio. Since there is an unequal number of audio clips available for each type of audio data, we perform oversampling of under-represented audio data types to achieve the 1:1:1:1 ratio for every batch during a training epoch. We also chose $\beta = 1e^{-4}$ after a coarse grid search. We set a hidden state size of 256 units for every GRU layer. We use the segment-based and event-based macro F1-scores as the metrics for validating the model's performance during training on each of the 3 classification tasks, along with L2 for validating the performance on the 4th reconstruction task. The metrics are calculated every training epoch.

### C. Evaluation

The public evaluation set of DCASE-Task4 2022 contains 692 clips of real audio with strong labels. After training for 100 epochs we keep the best performing model, according to the segment-based and event-based macro F1-scores achieved on the validation set for the *strong* event tagging task. We only keep the branch (encoder-decoder pair) responsible for the *strong* event tagging task and prune away the other task branches, as they are not needed beyond the training stage. Following that, we get the resulting model's *strong* predictions on the entire public evaluation set and calculate the segment-based and event-based macro F1-scores.

TABLE II
EVENT-BASED AND SEGMENT-BASED MACRO F1-SCORES WITH VARIOUS TYPES OF DATA USED FOR TRAINING

| Data used | Validation | | Public evaluation | |
|---|---|---|---|---|
| | EB-F1 [%] | SB-F1 [%] | EB-F1 [%] | SB-F1 [%] |
| Synthetic only | 12.6 | 31.4 | 12.4 | 38.0 |
| Synthetic + weak | 11.0 | 48.4 | 11.2 | 52.5 |
| Synthetic + weak + unlabeled | 12.5 | 50.9 | 11.6 | 54.4 |
| Synthetic + weak + unlabeled + Audioset | **32.5** | **60.6** | **31.8** | **60.6** |
| Audioset only | 19.8 | 41.8 | 18.5 | 37.6 |

TABLE III
PERFORMANCE COMPARISON BETWEEN USING DETERMINISTIC AND VARIATIONAL ENCODERS

| Encoder type | Validation | | Public evaluation | |
|---|---|---|---|---|
| | EB-F1 [%] | SB-F1 [%] | EB-F1 [%] | SB-F1 [%] |
| Deterministic | 28.6 | 59.1 | 27.2 | 59.7 |
| Variational | **32.5** | **60.6** | **31.8** | **60.6** |

TABLE IV
CLASS-WISE EVENT-BASED AND SEGMENT-BASED F1-SCORES

| Audio event class | Validation | | Public evaluation | |
|---|---|---|---|---|
| | EB-F1 [%] | SB-F1 [%] | EB-F1 [%] | SB-F1 [%] |
| Alarm bell ringing | 36.9 | 72.4 | 31.3 | 67.0 |
| Blender | 43.9 | 65.1 | 36.0 | 58.9 |
| Cat | 29.7 | 51.7 | 43.0 | 62.9 |
| Dishes | 16.5 | 36.2 | 20.1 | 37.5 |
| Dog | 12.0 | 51.7 | 14.7 | 66.9 |
| Electric shaver/toothbrush | 41.3 | 63.1 | 25.4 | 53.4 |
| Frying | 27.1 | 57.8 | 35.1 | 62.8 |
| Running water | 31.1 | 64.7 | 20.7 | 44.3 |
| Speech | 50.0 | 81.5 | 50.1 | 80.7 |
| Vacuum cleaner | 36.1 | 61.3 | 41.2 | 71.9 |
| Average | 32.5 | 60.6 | 31.8 | 60.6 |

## III. EXPERIMENTS

In this section we: A) define the performance metrics used to measure the classification performance of an SED system, B) define the post-processing applied to the predictions of an SED system, C) present the results of our experiments when using MTL to improve the classification performance of an SED system by leveraging additional types of data, D) experimentally assess the advantage of using variational instead of deterministic encoders in the MTL model to further improve generalization, E) analyze the class-wise performance of our MTL-VAE SED system on the DCASE Task4 dataset.

### A. Performance Metrics

As per [21], the audio event classification performance of our proposed SED system is evaluated using event-based and segment based macro F1-scores. The event-based F1-score is calculated with a 200 ms collar on the onsets and a collar on the offsets that is the greater of 200 ms and 20% of the sound event's length. The overall F1-score is the unweighted average of the class-wise F1-scores (macro-average). The segment-based F1-score is calculated on audio segments of 1 s duration. The metrics are computed using the *sed_eval* library [29].

### B. Post-processing

Before calculating the F1-scores, we apply a classification threshold of $0.5$ to the system audio event class probability outputs. We also apply median filtering to the frame-level class probability outputs, with a window size of 19 frames (or $456$ ms), in order to remove impulse false positive/negative detections. The optimal values for the threshold and median filtering window size were selected via grid search. More sophisticated methods for optimizing the post-processing applied to the predictions of sound event detectors, such as proposed in [30], could further improve audio event classification performance.

### C. Effects of Multi-Task Learning

The results of our experiments are summarized in Table II. We conduct an ablation study to assess the impact of each additional learned task to the performance of the multi-task model.

We observe that when the model is only trained for *strong event tagging* on synthetic audio data with strong event labels, it has the worst scores on the classification metrics with an event-based macro F1-score of $12.6\%$ and a segment-based macro F1-score of $31.4\%$ on the validation set, as well as $12.4\%$ and $38.0\%$ respectively on the public evaluation set.

Adding the task of *weak event tagging* on real audio data with weak event labels improves the segment-based F1-score to $48.4\%$ and $52.5\%$ on the validation and public evaluation sets respectively, but the event-based F1 score does not improve.

Further adding the task of *reconstruction* of unlabeled audio data improves the segment-based F1-score to $50.9\%$ and $54.4\%$ on the validation and public evaluation sets, respectively.

Finally, the addition of the task of *strong event tagging* on real audio event data with strong event labels (from the Audioset dataset) significantly improves the event-based F1-score to $32.5\%$ on the validation set and $31.8\%$ on the public evaluation set. The segment-based F1-score further improves to $60.6\%$ on both sets. This most likely occurs because the real audio dataset with strong event labels and, consequently, the task of *strong event tagging* on real audio data have the closest domain proximity to the validation and public evaluation datasets, which are also real audio data with strong event labels. Therefore, it is not a surprise that this task has the largest contribution to the information extracted by the MTL model.

However, when training only on the Audioset data and learning only the task of *strong event tagging* on real audio data, the final performance is significantly lower than when training on all tasks using all types of data. The event-based F1-score drops to $19.8\%$ and $18.5\%$ on the validation and public evaluation sets respectively, while the segment-based F1-score becomes $41.8\%$ and $37.6\%$, respectively. This underlines the effectiveness of MTL and that all 4 data types and their respective tasks contribute to the ability to learn more robust representations that generalize better.

### D. Contribution of VAEs

In Table III we also conduct an ablation study comparing the event-based and segment-based macro F1 scores achieved on the validation and public evaluation sets, by our MTL model, when each encoder is deterministic and when it is variational. Using a variational encoder leads to an improvement in the event-based F1 score of approximately $4\%$ and an improvement of $1\%$ in the segment-based F1 score. We conclude that this is due to the better generalization ability of the variational autoencoder architecture. Introducing stochasticity into the latent representations of each encoded task data features and constraining the shared latent space to be close to a Gaussian prior leads to improved regularization of learned task data representations.

### E. Class-wise performance

Table IV shows the event-based and segment-based F1-scores per audio event class achieved by the final MTL-SED model, trained on all 4 tasks. The results for both the validation and public evaluation sets of DCASE-Task4 2022 are presented. There is a total of 10 classes of audio events present in the dataset.

## IV. Conclusion

In this work we designed a Multi-Task Learning (MTL) model for Sound Event Detection (SED), based on a residual recurrent autoencoder architecture with variational information bottleneck. We applied this MTL-SED model to the challenge of learning to detect and classify audio events when only a limited amount of annotated training data is available, as outlined in DCASE-Task4. For each of the four types of data provided by the DCASE-Task4 dataset, we assigned a task to be learned: *strong audio event tagging* from the synthetic audio data with strong event labels, *weak audio event tagging* from the real audio data with weak event labels, *reconstruction* of real unlabeled data from the provided real audio data without annotations, and *strong audio event tagging* from the real audio data with strong event labels. The model is trained simultaneously on all tasks and has the ability to exploit cross-task information through parameter (weight) sharing between the autoencoders appointed to each task and through projecting the encoded features for each task data into a shared latent space. We then demonstrate that this MTL scheme significantly improves the model's classification accuracy, as measured by the event-based and segment-based macro F1 scores, in the validation and public evaluation datasets of DCASE-Task4, with each additional learned task contributing to improving the model's final performance. We also found that introducing stochasticity into the shared latent representations, by using variational instead of deterministic encoders further improves classification performance through better cross-task generalization, since the stochasticity introduced into latent representations acts as a regularizer.

As future work, we would replace the residual recurrent block with a transformer-based architecture. Transformer models have recently demonstrated strong performance in SED [31] and we believe it would be interesting to also assess the performance of an MTL model, based on transformer building blocks, on the task of semi-supervised SED.

## V. Acknowledgements

Research in this paper was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "1st Call for H.F.R.I. Research Projects to support Faculty Members & Researchers and the Procurement of High-Cost Research Equipment Grant" (Project Number: 3449).

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

## References

[1] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[2] N. Takahashi, M. Gygli, B. Pfister, and L. Van Gool, "Deep convolutional neural networks and data augmentation for acoustic event detection," *arXiv preprint arXiv:1604.07160*, 2016.

[3] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal processing letters*, vol. 24, no. 3, pp. 279–283, 2017.

[4] T. H. Vu and J.-C. Wang, "Acoustic scene and event recognition using recurrent neural networks," *Detection and Classification of Acoustic Scenes and Events*, vol. 2016, pp. 1–3, 2016.

[5] T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, and K. Takeda, "Duration-controlled lstm for polyphonic sound event detection," *IEEE/ACM Transactions on ASLP*, vol. 25, no. 11, pp. 2059–2070, 2017.

[6] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on ASLP*, vol. 25, no. 6, pp. 1291–1303, 2017.

[7] S. Adavanne, P. Pertilä, and T. Virtanen, "Sound event detection using spatial features and convolutional recurrent neural network," in *2017 IEEE ICASSP*. IEEE, 2017, pp. 771–775.

[8] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Weakly-supervised sound event detection with self-attention," in *2020 IEEE ICASSP*. IEEE, 2020, pp. 66–70.

[9] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs rnn in speech applications," in *2019 IEEE ASRU*. IEEE, 2019, pp. 449–456.

[10] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification." in *Interspeech*, vol. 2018, 2018, pp. 3573–3577.

[11] T.-W. Su, J.-Y. Liu, and Y.-H. Yang, "Weakly-supervised audio event detection using event-specific gaussian filters and fully convolutional networks," in *2017 ICASSP*. IEEE, 2017, pp. 791–795.

[12] L. Delphin-Poulat and C. Plapous, "Mean teacher with data augmentation for dcase 2019 task 4," *Orange Labs Lannion, France, Tech. Rep*, 2019.

[13] M. Crawshaw, "Multi-task learning with deep neural networks: A survey," *arXiv preprint arXiv:2009.09796*, 2020.

[14] S. Deshmukh, B. Raj, and R. Singh, "Multi-task learning for interpretable weakly labelled sound event detection," *arXiv preprint arXiv:2008.07085*, 2020.

[15] H. Liang, W. Ji, R. Wang, Y. Ma, J. Chen, and M. Chen, "A scene-dependent sound event detection approach using multi-task learning," *IEEE Sensors Journal*, 2021.

[16] K. Imoto, N. Tonami, Y. Koizumi, M. Yasuda, R. Yamanishi, and Y. Yamashita, "Sound event detection by multitask learning of sound events and scenes with soft scene labels," in *2020 IEEE ICASSP*. IEEE, 2020, pp. 621–625.

[17] W. Qian, B. Chen, Y. Zhang, G. Wen, and F. Gechter, "Multi-task variational information bottleneck," *arXiv preprint arXiv:2007.00339*, 2020.

[18] G. Lu, X. Zhao, J. Yin, W. Yang, and B. Li, "Multi-task learning using variational auto-encoder for sentiment classification," *Pattern Recognition Letters*, vol. 132, pp. 115–122, 2020.

[19] T.-H. Vo, G.-S. Lee, H.-J. Yang, S.-R. Kang, I.-J. Oh, and S.-H. Kim, "Multi-task with variational autoencoder for lung cancer prognosis on clinical data," in *The 9th International Conference on Smart Media and Applications*, 2020, pp. 234–237.

[20] J. Shen, X. Zhen, M. Worring, and L. Shao, "Variational multi-task learning with gumbel-softmax priors," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 031–21 042, 2021.

[21] 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

[22] F. Ronchini, R. Serizel, N. Turpault, and S. Cornell, "The impact of non-target events in synthetic soundscapes for sound event detection," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 115–119.

[23] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[24] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[25] C. J. Steinmetz and J. D. Reiss, "pyloudnorm: A simple yet flexible loudness meter in python," in *150th AES Convention*, 2021.

[26] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," 2016.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] J. F. Gemmeke, D. P. W. 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 *2017 IEEE ICASSP*, New Orleans, LA, 2017.

[29] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[30] P. Giannakopoulos, A. Pikrakis, and Y. Cotronis, "Improving post-processing of audio event detectors using reinforcement learning," *IEEE Access*, vol. 10, pp. 84 398–84 404, 2022.

[31] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Conformer-based sound event detection with semi-supervised learning and data augmentation," *dim*, vol. 1, p. 4, 2020.