HEAR 2021: Holistic Evaluation of Audio Representations

Joseph Turian  
Jordie Shier  
Humair Raj Khan

Bhiksha Raj  
Björn W. Schuller  
Christian J. Steinmetz  
Colin Malloy  
George Tzanetakis  
Gissel Velarde  
Kirk McNally  
Max Henry  
Nicolas Pinto  
Camille Noufi  
Christian Clough  
Dorien Herremans  
Eduardo Fonseca  
Jesse Engel  
Justin Salamon  
Philippe Esling  
Pranay Manocha  
Shinji Watanabe  
Zeyu Jin  
Yonatan Bisk

© J. Turian et al.

Editor: Editor’s name

Abstract

What audio embedding approach generalizes best to a wide range of downstream tasks across a variety of everyday domains without fine-tuning? The aim of the HEAR 2021 NeurIPS challenge is to develop a general-purpose audio representation that provides a strong basis for learning in a wide variety of tasks and scenarios. HEAR 2021 evaluates audio representations using a benchmark suite across a variety of domains, including speech, environmental sound, and music. In the spirit of shared exchange, each participant submitted an audio embedding model following a common API that is general-purpose, open-source, and freely available to use. Twenty-nine models by thirteen external teams were evaluated on nineteen diverse downstream tasks derived from sixteen datasets. Open evaluation code, submitted models and datasets are key contributions, enabling comprehensive and reproducible evaluation, as well as previously impossible longitudinal studies. It still remains an open question whether one single general-purpose audio representation can perform as holistically as the human ear.

Keywords: audio representations, representation learning, embeddings, transfer learning, multi-task learning, multi-modal learning, classification, tagging
1. Introduction

The codification of strong general-purpose representations in natural language and computer vision has led to a renaissance in multimodal modeling and increased cross-discipline collaboration. Audio is an equally rich source of information about the world, but outside of speech recognition it has not achieved the same degree of attention from the machine learning community. This is a key challenge for the community, as good representations support good machine learning. And robust evaluation enables general representations. Broad evaluation suites help prevent overfitting to common test sets (Recht et al., 2018) and have improved the state-of-the-art on language and vision representation learning (Wang et al., 2019b,a; Goyal et al., 2019; Zhai et al., 2019; DeYoung et al., 2020). In general practice, audio representations are not evaluated on a broad range of audio problems, and as a result, it is difficult to know which audio representation to use for a novel audio learning task.

The Holistic Evaluation of Audio Representations (HEAR) challenge was created to encourage the development of flexible audio representations, to give greater insight into how audio representations will generalize, and to enable fast development cycles both for researchers developing new models and researchers applying existing models. HEAR participants submitted audio representation models that are general-purpose, open-source, and freely available to use off-the-shelf. All HEAR compatible models follow a common API, which makes switching between models as simple as changing one line of code.

The HEAR benchmark includes nineteen tasks: five open tasks derived from three datasets for which the problem definition and evaluation data were available to participants, and 14 additional secret tasks for evaluation, to which participants were completely blind throughout the challenge. While most of the tasks (open or secret) have good or promising solutions when worked on in isolation, the novelty of this challenge is that the same representation must be used to solve all of them. These tasks encompass multiple audio domains: speech, environmental sound, and music, with tasks that involve short and long time spans. HEAR datasets are easy to use: all are preprocessed to a common format with standard splits and self-explanatory human-readable metadata, and are distributed as tarfiles online.1 This alleviates the engineering effort required to work with datasets that require YouTube scraping, have variably documented preprocessing requirements, or are gatekept through closed-access request forms. Researchers are also welcome to use HEAR datasets under entirely open licenses (many of which allow commercial use), without using our downstream evaluation code.

Evaluation consists of classification tasks, both multiclass and multilabel, requiring either prediction over the entire audio scene (clip), or temporal-based onset detection of sound event (Mesaros et al., 2016). HEAR-compatible models can generate an embedding of arbitrary size, which is fed into a simple generic predictor by our open-source evaluation algorithm. Evaluation code, submitted models, and datasets are all available at https://neuralaudio.ai/hear.html.

1. https://zenodo.org/record/5885750
2. Background on representation learning

At a high level, a learned representation (embedding) consists of a machine learning model that takes a low-level representation of the input and outputs a numerical representation, typically a fixed-size vector, that lends itself well to discriminative tasks (e.g., by training a simple MLP on these embeddings). A good representation should (1) transfer to a wide range of different tasks and (2) transfer with limited supervision (Goyal et al., 2019, 2022).

In the following paragraphs, we describe trends from the natural language processing (NLP) and vision literature on representation learning, some of which have been applied to audio. Vision work is particularly relevant (Amiriparian et al., 2017), as 2-D transformations of audio, such as the widely used log-Mel spectrogram (Davis and Mermelstein, 1980), lend themselves well to methods designed to process 2-D input data. For this reason, a common thread in the literature on audio representations is that vision models are applied to 2-D audio representations. With that said, many of the insights from text-based language modeling, such as autoregressive neural modeling (Bengio et al., 2001), predicting tokens masking as an unsupervised pretext task (Collobert et al., 2011), and bidirectional transformers (Devlin et al., 2019), have found their way into the audio literature, e.g., WaveNet (van den Oord et al., 2016), wav2vec (Schneider et al., 2019), and HuBERT (Hsu et al., 2021), respectively. Textless NLP like Generative Spoken Language Modeling (GSLM, Lakhotia et al. (2021)), applies an NLP lens to spoken audio instead of written text.

Inducing representations The shallowest representation for audio is the raw digital audio signal itself. However, its extremely high dimensionality means it is rarely useful for discriminative tasks without additional processing, whether via manually crafted DSP engineering or transformations learned by training a neural network (Trigeorgis et al., 2016). Better representations might be obtained by applying a hand-crafted transformation based upon domain-expertise, such as the log-scaled Mel spectrogram (Davis and Mermelstein, 1980), Mel Frequency Cepstral Coefficients (MFCC, Logan (2000)), constant Q-transform (Schörkhuber and Klapuri, 2010), or scattering transform (Andén and Mallat, 2014). Audio filterbanks can also be learned (Zeghidour et al., 2021). Deep ML architectures can extract even more abstract, high-level representations (Aytar et al., 2016; Hershey et al., 2017; Cramer et al., 2019). Purely randomly weighted architectures impose particular inductive biases on data and can do better than hand-crafted baselines (Saxe et al., 2011; Pons and Serra, 2019). However, it is more common to train these architectures.

Architectures The architecture of the model typically includes an encoder to transform the input, and can optionally also include temporal modelling to capture context, and/or a generative decoder. A common encoder architecture uses Convolutional Neural Networks (CNN) applied to a 2-D input (Hershey et al., 2017; Cramer et al., 2019), or directly to the 1-D audio signal (van den Oord et al., 2016; Baevski et al., 2020). Temporal context modelling is often achieved via Recurrent Neural Networks (RNN) (Merhi et al., 2017; Kalchbrenner et al., 2018), or Transformers (Baevski et al., 2020). The latter, in particular, have achieved strong results for audio classification (Gong et al., 2021a), though they are costly to train from scratch. Koutini et al. (2021) (§4) demonstrate a faster training approach for audio transformer, which requires two GPU-days to pretrain on AudioSet. In
reaction to the use of transformers, all-MLP architectures have demonstrated competitive
results on language and vision tasks (Liu et al., 2021; Tolstikhin et al., 2021).

**Training regimes** Models can be trained on a (large-scale) supervised task, such as
ImageNet (Deng et al., 2009) for vision and AudioSet (Gemmeke et al., 2017) for audio.
Multitask supervised training can further improve generalization (Pascual et al., 2019).

To avoid the need for human-labeling, self-supervised models (a form of unsupervised
learning) learn from large-scale unlabeled corpora. Many self-supervised approaches learn
to correspond the original input with a different view on that same input, such as a se-
manitically identical augmentation (Chen et al., 2020b; Tian et al., 2020). To avoid col-
lapsed solutions, self-supervised approaches historically used negative samples with a triplet
loss (Chopra et al., 2005), possibly requiring large negative batches (Chen et al., 2020b;
Saeed et al., 2021), which can be expensive to train. Alternatives include quantization ap-
proaches to define uniform clusterings of representations (Baevski et al., 2020), or carefully
implemented asymmetric training architectures like BYOL (Grill et al., 2020; Niizumi et al.,
2021) and SimSiam (Chen and He, 2021). More recent are self-supervised approaches that
avoid these aforementioned techniques, relying instead upon explicit and fundamental pri-
or (Zbontar et al., 2021; Bardes et al., 2022). Input augmentations can be used to increase
the size of training data or provide corresponding views on the input (Salamon and Bello,
2017). Fonseca et al. (2021b) and Wang and van den Oord (2021) discuss augmentations,
including audio mixing, which Gong et al. (2021b); Wang et al. (2021b) explore in greater
dept and argue is useful both for supervised and unsupervised regimes.

Multi-modal approaches learn the correspondence between different modalities of the
input. Different modalities can accelerate compact learning in a single target modality by
exploiting cross-modality structure. OpenL3 (§4, Cramer et al. (2019)) is a broad-domain
audio model trained on the correspondence between audio and video. Contrastive Language-
Image Pre-training (CLIP, Radford et al. (2021)) learns a model from 400 M image-text
pairs, and was successfully applied on zero-shot tasks. Wang and van den Oord (2021) con-
trastively induce audio representations from waveforms (1-D audio) and spectrograms, and
Wang et al. (2021b) extend that to include correspondence with video frames.

Because pretraining large-scale models requires large quantities of data and can be com-
cputationally expensive, another research direction has been on distilling information from
existing models that were trained on another modality for which more data are available.
Aytar et al. (2016) train SoundNET which distills audio representations from a pre-trained
image classification model trained on large image datasets such as ImageNet (Deng et al.,
2009). Wu et al. (2022a) (§4) distill an audio representation (Wav2CLIP) from a large
text-image model (CLIP) using video data to link the visual and audio modalities.

**Using and evaluating representations** Representation models can be used in down-
stream tasks with full fine-tuning; but the naïve approach is simply to treat interme-
diate pre-trained model outputs as frozen embeddings, and this nonetheless provides a
stark improvement over using raw features (Turian et al., 2010). Broad-scale evaluation
of learned representations has been done in other ML domains, in NLP, for example: GLUE
(Wang et al., 2019b), the harder SuperGLUE (Wang et al., 2019a), and ERASER
(DeYoung et al., 2020). Vision includes the FAIR self-supervision benchmark (Goyal et al.,
2019) and VTAB (Zhai et al., 2019).
3. HEAR: Holistic Evaluation of Audio Representations

A strong general-purpose audio representation should be as holistic as the human ear. The goal of the HEAR competition is to evaluate audio representations across a variety of everyday domains, audio phenomena, with tasks that involve short and long time spans, sometimes with few labeled instances. Formal rules are provided on the HEAR website.²

3.1. Related audio shared tasks

Historical audio shared tasks, such as those from MIREX (Downie et al., 2014), DCASE (Mesaros et al., 2017), and INTERSPEECH ComParE (Schuller et al., 2013) have improved the community’s understanding of audio modeling substantially. However, the bespoke nature of these tasks is a double-edged sword, requiring substantial custom tooling both by the challenge organizers and participants. More recent audio shared tasks focus on reusability and generic task APIs. SUPERB (Yang et al., 2021) focuses on a broad spectrum of speech tasks, and includes downstream evaluation ranging from simple classification to LSTM-based sequence modeling. Although the Speech Commands v2 task is shared with HEAR, the other downstream tasks in SUPERB mainly deal with speech processing applications, including speech recognition, speaker verification, keyword spotting, etc., and these two evaluation activities are complementary to each other. The NON-Semantic Speech Benchmark (NOSS, Shor et al. (2020)) comprises 6 paralinguistic tasks. Two tasks are shared with HEAR: CREMA-D and Speech Commands v2. Unfortunately, SAVEE and DementiaBank require filling out a request form, and VoxCeleb requires scraping YouTube. HARES (Holistic Audio Representation Evaluation Suite)—not to be confused with our HEAR challenge—is concurrently published work (Wang et al., 2021c). HARES comprises 12 well-known downstream tasks including—like HEAR—ESC-50, Speech Commands v2, and an NSynth Pitch task, benchmarked on 13 models. Where HARES differs from HEAR includes: a) HARES tasks are well-known benchmarks, whereas HEAR is a mix of well-known and novel benchmarks, b) HARES includes no few-shot tasks, all tasks have ≥ 2K samples, c) HARES results currently include no external submissions, d) evaluation code and dataset links are not provided and e) two of the tasks (AudioSet and VoxCeleb) tasks involve scraping YouTube. Datasets based upon YouTube require specialized code and lack reproducibility because videos are removed unpredictably (Cramer et al., 2019). These generic audio evaluation suites, including our HEAR challenge, intend to make it easy to evaluate existing models on novel tasks, at the expense of possible SOTA performance.

3.2. Evaluation methodology

Wrapping existing models into the HEAR API requires roughly 75 lines of code, much of which is boilerplate. New HEAR tasks can be run with no code changes. HEAR 2021 includes two types of tasks: 1) Scene-based: Multi-class or multi-label classification of an entire audio clip; 2) Timestamp-based: Sound event detection/transcription, which involves detecting when exactly sound events occur over time by providing a start time, end time, and label for each sound event. In both cases, the audio representation is frozen and used as the input feature vector to a shallow downstream MLP classifier, with no fine-tuning.

Fine-tuning improves downstream performance (Baevski et al. (2020); Shor et al. (2020)), but increases training time. Crucially, the use of frozen embeddings means that HEAR downstream evaluation code can be maintained solely in PyTorch, regardless of whether the embedding model was in TensorFlow or PyTorch. 3

A timestamp-based task can be simplified to a frame-based sequence-labeling task of the audio at regular intervals (Kelz et al., 2016), and we use a common postprocessing step to compose predictions from multiple timesteps and extract discrete labeled events with start and ends times (Mesaros et al., 2016). Framewise accuracy (the decomposed multilabel prediction, computed at regular timesteps) does not always correlate well with the perceptual quality of event-onset FMS (Hawthorne et al., 2018) because they ignore the interplay between the frame representations and more sophisticated downstream inference (Cheuk et al., 2021). See Section B for details on the downstream training regime.

3.3. Evaluation tasks

The following are the evaluation tasks for HEAR 2021. For simplicity and reproducibility, we have preprocessed each relevant datasets to all commonly used sample rates (16000, 22050, 32000, 44100), fixed the length of the audio clips, predefined training splits, and packaged each dataset in a self-explanatory common format with human-readable metadata. They all have open licenses (some of which permit commercial use), with the exception of the GTZAN corpora which are widely used but of unknown license status. We encourage the community to use our datasets, even if they do not follow all or any the HEAR 2021 rules, but encourage deviations from the HEAR 2021 rules to be described.

Open tasks were released early in the challenge, to encourage participation and to allow participants to debug and refine their submissions: Speech Commands v2 (full and 5h versions), NSynth Pitch (50h and 5h versions), and DCASE 2016 Task 2. Tasks are summarized in Table 1 described with more detail in Table 2 and Section A.

4. Models evaluated

Evaluated models are described below. Table 3 summarizes model properties. HEAR began with three strong baseline models (§4.1), each pretrained on a different audio domain. We report on 13 external teams’ submissions to HEAR 2021 (§4.2).

4.1. Baseline models

**wav2vec2**  wav2vec2 (1-D CNN and positional transformer) (Baevski et al., 2020). Self-supervised pretraining on 100K hours of speech from VoxPopuli (Wang et al., 2021a).

**CREPE**  1-D CNN. Supervised pretraining of pitch-tracking on 16 hours of synthesized music. (Kim et al., 2018b)

---

3. We initially believed that imposing a restriction that all submitted models must be TensorFlow 2.x or PyTorch and pip3-installable would facilitate easy orchestration of model testing. However, models submitted with competing TensorFlow, CUDA, CuDNN, and pypi dependencies lead us to suggest that future ML challenge organizers standardize on the latest stable microversion of all deep learning packages.
Table 1: HEAR tasks.

Speech Commands (version 2), 5h and full Spoken commands classification.
NSynth Pitch, 5h and 50h Pitch classification of synthesized sounds.
DCASE 2016 Task 2 Office sound event detection in synthesized scenes.
Beehive States Binary classification of normal vs. queen-less beehives.
Beijing Opera Percussion Classification of six Beijing Opera percussion instruments.
CREMA-D Speech emotion recognition.
ESC-50 Environmental sound classification.
FSD50K Broad-domain audio multi-labeling.
Gunshot Triangulation Identify location of microphone recording a gunshot, using classification.
GTZAN Genre Music genre classification.
GTZAN Music Speech Classification of audio into music or speech.
LibriCount Multiclass speaker count identification.
MAESTRO 5h Music transcription.
Mridingham Stroke and Mridingham Tonic Non-Western pitched percussion. Classification of stroke or tonic.
Vocal Imitations Match a vocal imitation to the type of sound imitated, using classification.
VoxLingua107 Top 10 Spoken language identification.

OpenL3 2-D CNN. Multi-modal contrastive self-supervised pretraining of audio/video correspondence on 6K hours of AudioSet broad-domain YouTube content. (Cramer et al. (2019), earlier Arandjelovic and Zisserman (2017)) HEAR implementation by Jon Nordby.

4.2. Submitted models

AMAAI Lab SUTD wav2vec2+DDSP An ensemble of wav2vec2 (Baevski et al., 2020) and two DDSP encoders (Engel et al., 2020). The wav2vec2 model is pretrained on the Librispeech (Panayotov et al., 2015) and MAESTRO (Hawthorne et al., 2019) datasets. One DDSP encoder is CREPE, the other is a non-pretrained loudness encoder.

AMAAI wav2vec2 music+speech wav2vec2 model (Baevski et al., 2020). Pretrained on Librispeech (Panayotov et al., 2015) and MAESTRO (Hawthorne et al., 2019).

Audioshake UDONS ViT Vision transformer (ViT, Kolesnikov et al. (2021)). Pretrained on 360h of Librispeech to predict the correct permutation (Noroozi and Favaro, 2016; Carr et al., 2021) of up to 5 patches of mel-spectrogram input, shuffled in time.

CP-JKU PaSST base, base2level, base2levelmel Patchout fast (2-D) spectrogram transformer (PaSST, Koutini et al. (2021)). Initialized from an ImageNet vision transformer model, and further pretrained on 10s audio from AudioSet to perform supervised tagging. base2level concatenates a longer window (160 ms and 800ms) for timestamp embeddings. base2levelmel additionally concatenates the raw melspectrogram as well.
CVSSP (University of Surrey) PANNs  2-D CNN14. Pretrained on AudioSet with supervision (Kong et al., 2020).

**Descript/MARL Wav2CLIP**  2-D ResNet18. Pretrained multimodally using contrastive learning on the 600h VGGSound corpus (Chen et al., 2020a) (without supervised labels) to distill the Contrastive Language-Image Pre-training (CLIP, Radford et al. (2021)) language and image model to a corresponding audio embedding (Wu et al., 2022a).

**IUT-CSE kwmlp and audiomlp**  Sequentially stacked gated MLP model (Liu et al., 2021), taking (2-D) MFFCs as input. kwmlp (Morshed et al., 2022) is pretrained with supervision on Speech Commands v2. audiomlp is pretrained with supervision on HEAR open task datasets: Speech Commands v2, DCASE 2016 Task 2, and NSynth Pitch.

**Kuroyanagi hearline**  2-D conformer model. Pretraining unknown.

**Logitech AI SERAB BYOL-S**  2-D CNN. Self-supervised pretraining using the BYOL self-supervised approach (Grill et al., 2020) adapted to audio (BYOL-A, Niizumi et al. (2021)), pretrained on the speech subset of AudioSet (Elbanna et al., 2022).

**NTU-GURA (fusion) avg/cat hubert/wav2vec2/crepe**  Three models (HuBERT Hsu et al. (2021), wav2vec2, CREPE) combined in a variety of ways: averaged or concatenated (Wu et al., 2022b). Fusion of multiple model layers was optionally included. fusion_cat_xwc_time is a variation of fusion_cat_xwc with a different approach to matching timestamps when concatenating different models’ embeddings.


**Soundsensing YAMNet**  2-D MobileNet (Howard et al., 2017). Pretrained to tag AudioSet.

**Stellenbosch LSL Audio DBERT**  1-D CNN encoder and modified BERT transformer. Pretrained as the discriminator with a GAN objective, using the clustering model as the generator, on 960 hours of Librispeech (Panayotov et al., 2015). Embeddings are taken from layer 16 of 24 by default.

5. Results and Discussion

In Figure 1 we present the primary score of submitted models on each HEAR 2021 task. By default, evaluation uses a deterministic seed, for reproducibility. Nonetheless, scores are stable across our evaluation, with a median 95% confidence interval of 2.5e-3 when seeding of model weights and hyperparameter grid points is selected non-deterministically. Shor et al. (2020); Wu et al. (2022a) present scores for some of the the same models and tasks. HEAR reported scores are similar but not identical, due to downstream training differences.

To display model similarity at a glance, we present t-SNE visualizations of normalized scores by task (Figure 2(a)) and by model (Figure 2(b)). We also show correlation tables for tasks (Figure 3) and models (Figure 4) to give greater insight into model and task similarity,
HEAR 2021: Holistic Evaluation of Audio Representations

in similar spirit to the confusion matrices of Wu et al. (2022a). Zhai et al. (2019) compare a variety of aggregation techniques for evaluating cross-task model performance, and find that they are all highly correlated, settling upon simple mean top-1. Gosiewska et al. (2020) proposes an ELO-like meta-score for cross-task model performance, similar to a chess rating. Although it is tempting to give a single score for every model, we believe that would strip out important nuances shown in the full score table (DeYoung et al., 2020).

For these summary figures, we normalize each model/task score. Normalized scores allow us to compare models and tasks against each other, under the assumption each task is equally weighted. The normalization procedure is as follows: 1) For each task, we standardize the scores to zero mean and unit variance. Unlike transforming tasks to ranks, we assume that the scale of intra-task scores is important. 2) The standardized scores are Winsorized (clamped) to have variance within [-1, +1]. By limiting the importance of extremely high or low scores on a single task, this approach allows for better inter-task comparison.

In the following paragraphs, we describe a few interesting patterns and trends in the submitted models. Many evaluated models use the last layer as the representation. It is known that non-final layers and/or fusing various layers might capture more information (Shor et al., 2020; Baevski, 2020; Yang et al., 2021). Intermediate layers often model audio phenomena that are not necessary for the final loss. NTU-GURA’s ablation studies support that, as evidence by the relative performance of their different models. For conciseness, we use the term “strong speech models” to refer to NTU-GURA’s fused models that include pretrained speech models.

**Pitch tasks** NSynth pitch and Maestro tasks have similar results, and models that include CREPE embeddings (Kim et al., 2018b) perform best. This makes sense as these tasks require modeling of pitch, which CREPE was specifically trained for, while many other representations focus on discriminating between semantic objects (e.g., cat vs dog or guitar vs piano) but are pitch agnostic. Interestingly, models trained for semantic discrimination (e.g., via AudioSet) and speech models do nonetheless represent pitch to some degree, as evidenced by the decent performance of OpenL3 and wav2vec2 on these tasks.

**Broad Domain Semantic-Object Tagging** FSD50K and ESC-50 semantic-object tagging results are strongly correlated, as well as—perhaps surprisingly—GTZAN genre tagging. The models that perform the best on this group are the ones pretrained on the AudioSet semantic-object tagging task. What we glean from this large-scale survey of diverse models is that results on ESC-50 and GTZAN genre tagging are strongly predictive of results on the more nuanced FSD50K task, despite being an order of magnitude smaller and not using the corrected GTZAN artist-conditional splits from (Sturm, 2013), suggesting faster inroads for research iteration. One valuable point-based contribution of HEAR 2021 is that the CP-JKU PaSST base model achieves a new state-of-the-art on FSD50K despite no fine-tuning, a mean average precision (mAP) of 0.640 on FSD50K, compared to the recent literature (Gong et al., 2021b; Wu et al., 2022a; Fonseca et al., 2021a).

**Vocals** FSD50K scores are also similar to those of Vocal Imitations and LibriCount. This is perhaps because Vocal Imitations comprises broad non-semantic vocalizations and LibriCount involves detecting multiple simultaneous audio events. The strong speech and PaSST models do the best on Vocal Imitations. On LibriCount, SERAB BYOL-S does the best as a non-semantic speech model, with decent performance from strong speech models.
Figure 1: Primary score of submitted models on each HEAR 2021 task. Normalized scores are used to show the heat-value of each cell. Missing cells indicate that the model did not successfully complete the task (exhausting GPU memory or exceeding 24 hours downstream training time).
HEAR 2021: Holistic Evaluation of Audio Representations

Speech  As we move into the speech domain, LibriCount and Vocal Imitations have the most similarity to CREMA-D emotion detection, which then is most similar to VoxLingua107 Top 10 language identification, which in turn is correlated with Speech Commands, following a trend from “environmental” to paralinguistic to semantic. The strong speech models do the best on these tasks.

What is most interesting about our diverse survey of 29 models × 19 tasks is, perhaps, the most difficult to explain results: tasks that defy neat categorization suggest the fragile, unpredictable boundaries of existing models. DCASE 2016 Task 2 seems a priori similar to FSD50K and ESC-50, but not in practice. Vocal Imitations are human-depictions of all kinds of sounds. Gunshot Triangulation is an extremely low-resource task with only 88 instances. Beijing Opera and Mridingham Stroke and Tonic are non-Western music tasks. For these tasks, our contribution is a negative result: we have no simple story or obvious pretraining data to attack them. Robust generalization of >10-billion-parameter models from NLP (Brown et al., 2020) and vision (Goyal et al., 2022) suggest one path forward.

6. Conclusion

General-purpose models that transfer to few-shot and zero-shot scenarios are highly desirable. The audio community has followed the NLP and vision communities in using increasingly sophisticated representation learning approaches. The HEAR challenge allows the audio community also to follow the trend of broad-scale reproducible evaluation.

HEAR 2021 is about openness. The datasets and the submissions are as open as possible. All HEAR 2021 datasets are preprocessed to a common format with standard splits, and distributed as tarfiles. This alleviates the risk of dataset rot common in YouTube scraping, and the difficulty of acquiring data locked behind closed-access request forms. All HEAR 2021 submissions have code that is Apache 2.0 compatible, models that are CC-Attribution compatible, and follow a common API, so switching between them requires a single line of code. Evaluation code, submitted models, and datasets are key contributions of HEAR 2021, available at https://neuralaudio.ai/hear.html.

Twenty-nine models were evaluated on 19 diverse downstream tasks, spanning speech, environmental sounds, and music, and datasets that don’t fit neatly into any rubric, as well as datasets that span the boundaries of multiple audio domains. This large standardized set of tasks and models pave the way for comprehensive and reproducible evaluation, enabling previously impossible longitudinal studies. We are eager to help onboard new tasks into the HEAR benchmark suite, particularly unusual and/or few-shot audio tasks. The largest-scale HEAR scene-embedding tasks and the CPU-gated evaluation of timestamp-embedding tasks were the most difficult tasks to run, sometimes requiring 24 hours for downstream evaluation of a single model-task pair on an A100 GPU, despite no fine-tuning.

Before an evaluation like HEAR, it would be easy for the community to suggest which audio tasks are predictably hard: large-scale, well-defined datasets with no more low-hanging fruit that are known to be difficult to hill-climb. Our contribution—the existence and easy accessibility of HEAR datasets, models, and evaluation code—allows the community to probe what we don’t know. And the central question posed by HEAR 2021 remains open: Can one single general-purpose audio representation perform as holistically as the human ear? If one does, then there is clearly more work to be done towards achieving it.
Acknowledgments

HEAR 2021 was sponsored by Google, and competition evaluation was performed on Google Cloud Platform.

References


Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen,


Kin Wai Cheuk, Yin-Jyun Luo, Emmanouil Benetos, and Dorien Herremans. Revisiting the Onsets and Frames Model with Additive Attention. In International Joint Conference on Neural Networks, (IJCNN), pages 1–8, 2021.


Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling Factorized Piano


Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition.


Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Peter Steiner, Daniel Keysers, Jakob Uszkoreit, Mario


Table 2: Summary of the 19 evaluation tasks for HEAR 2021. Includes the embedding type (timestamp (T) or scene (S)), the predictor type (multiclass (C) or multilabel (L)), the split method used during downstream evaluation (train/validation/test (TVT) or K-fold), the duration of clips in seconds, the total number of clips for each task, the primary evaluation metric, and whether or not the task is novel. Novel tasks are not comparable to the literature. For all tasks except FSD50k, clips were standardized to one length using padding or trimming, typically the 95th percentile length in the original corpus.

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Embed Type</th>
<th>Predictor Type</th>
<th>Split Method</th>
<th>Duration (seconds)</th>
<th># clips</th>
<th>Evaluation Metric</th>
<th>Novel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Open Tasks</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCASE 2016 Task 2</td>
<td>T</td>
<td>L</td>
<td>TVT</td>
<td>120.0</td>
<td>72</td>
<td>Onset FMS</td>
<td>✓</td>
</tr>
<tr>
<td>NSynth Pitch 5hr</td>
<td>S</td>
<td>C</td>
<td>TVT</td>
<td>4.0</td>
<td>5000</td>
<td>Pitch Acc.</td>
<td>✓</td>
</tr>
<tr>
<td>NSynth Pitch 50hr</td>
<td>S</td>
<td>C</td>
<td>TVT</td>
<td>4.0</td>
<td>49060</td>
<td>Pitch Acc.</td>
<td>✓</td>
</tr>
<tr>
<td>Speech Commands 5hr</td>
<td>S</td>
<td>C</td>
<td>TVT</td>
<td>1.0</td>
<td>22890</td>
<td>Accuracy</td>
<td>✓</td>
</tr>
<tr>
<td>Speech Commands Full</td>
<td>S</td>
<td>C</td>
<td>TVT</td>
<td>1.0</td>
<td>100503</td>
<td>Accuracy</td>
<td></td>
</tr>
<tr>
<td><strong>Secret Tasks</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beehive States</td>
<td>S</td>
<td>C</td>
<td>TVT</td>
<td>600.0</td>
<td>576</td>
<td>AUCROC</td>
<td>✓</td>
</tr>
<tr>
<td>Beijing Opera Percussion</td>
<td>S</td>
<td>C</td>
<td>5-fold</td>
<td>4.77</td>
<td>236</td>
<td>Accuracy</td>
<td>✓</td>
</tr>
<tr>
<td>CREMA-D</td>
<td>S</td>
<td>C</td>
<td>5-fold</td>
<td>5.0</td>
<td>7438</td>
<td>Accuracy</td>
<td></td>
</tr>
<tr>
<td>ESC-50</td>
<td>S</td>
<td>C</td>
<td>5-fold</td>
<td>5.0</td>
<td>2000</td>
<td>Accuracy</td>
<td></td>
</tr>
<tr>
<td>FSD50K</td>
<td>S</td>
<td>L</td>
<td>TVT</td>
<td>0.3 - 30.0</td>
<td>51185</td>
<td>mAP</td>
<td></td>
</tr>
<tr>
<td>Gunshot Triangulation</td>
<td>S</td>
<td>C</td>
<td>7-fold</td>
<td>1.5</td>
<td>88</td>
<td>Accuracy</td>
<td>✓</td>
</tr>
<tr>
<td>GTZAN Genre</td>
<td>S</td>
<td>C</td>
<td>10-fold</td>
<td>30.0</td>
<td>1000</td>
<td>Accuracy</td>
<td>✓</td>
</tr>
<tr>
<td>GTZAN Music Speech</td>
<td>S</td>
<td>C</td>
<td>10-fold</td>
<td>30.0</td>
<td>128</td>
<td>Accuracy</td>
<td></td>
</tr>
<tr>
<td>LibriCount</td>
<td>S</td>
<td>C</td>
<td>5-fold</td>
<td>5.0</td>
<td>5720</td>
<td>Accuracy</td>
<td></td>
</tr>
<tr>
<td>MAESTRO 5hr</td>
<td>T</td>
<td>L</td>
<td>5-fold</td>
<td>120.0</td>
<td>185</td>
<td>Onset FMS</td>
<td>✓</td>
</tr>
<tr>
<td>Mridangam Stroke</td>
<td>S</td>
<td>C</td>
<td>5-fold</td>
<td>0.81</td>
<td>6977</td>
<td>Accuracy</td>
<td>✓</td>
</tr>
<tr>
<td>Mridangam Tonic</td>
<td>S</td>
<td>C</td>
<td>5-fold</td>
<td>0.81</td>
<td>6977</td>
<td>Accuracy</td>
<td>✓</td>
</tr>
<tr>
<td>Vocal Imitations</td>
<td>S</td>
<td>C</td>
<td>3-fold</td>
<td>11.26</td>
<td>5601</td>
<td>mAP</td>
<td>✓</td>
</tr>
<tr>
<td>VoxLingua107 Top10</td>
<td>S</td>
<td>C</td>
<td>5-fold</td>
<td>18.64</td>
<td>972</td>
<td>Accuracy</td>
<td>✓</td>
</tr>
</tbody>
</table>

Appendix A. Evaluation Tasks

Our 19 tasks were derived from 16 datasets, as described in more detail below. Tasks described as “novel” are not comparable to the literature. A summary of task statistics is available in Table 2.

Speech Commands (version 2), 5h and full Classification of known spoken commands, with additional categories for silence and unknown commands (Warden, 2018). As per the literature, models are evaluated by prediction accuracy. For this challenge we also provide a 5-hour subset of the training data. We use the predefined train and test split, and note that the test data has a different distribution of labels from the training data.

NSynth Pitch, 5h and 50h NSynth Pitch is a novel multiclass classification problem. The goal of this task is to classify instrumental sounds from the NSynth Dataset (Engel et al., 2017) into one of 88 pitches. Results for this task are measured by pitch accuracy, as well as chroma accuracy. Chroma accuracy considers only the pitch “class”
i.e., pitches that are a multiple-of-octaves apart are considered equivalent. For HEAR 2021 we created two versions of this dataset: a 5 hour and 50 hour version. Unlike Carr et al. (2021), we treat this as a classification, not regression problem.

**DCASE 2016 Task 2** A novel office sound event detection in synthesized scenes, adapted from DCASE 2016 Task 2 (Mesaros et al., 2018). Novel, insofar as our evaluation uses different splits. The original imbalanced splits did not work well our generic cross-validation.

Postprocessing: Predictions were postprocessed using 250 ms median filtering. At each validation step, a minimum event duration of 125 or 250 ms was chosen to maximize onset-only event-based F-measure (with 200 ms tolerance). Scores were computed using sed_eval (Mesaros et al., 2016).

**Beehive States** This is a binary classification task using audio recordings of two beehives (Nolasco et al., 2019). The beehives are in one of two states: a normal state, and one in which the queen bee is missing (“queen-less”). At 10 minutes long, this task has the longest audio clips in HEAR.

**Beijing Opera Percussion** This is a novel audio classification task developed using the Beijing Opera Percussion Instrument Dataset (Tian et al., 2014). The Beijing Opera uses six main percussion instruments that can be classified into four main categories: Bangu, Naobo, Daluo, and Xiaoluo.

**CREMA-D** CREMA-D is a dataset for emotion recognition (Cao et al., 2014). The original dataset contains audiovisual data of actors reciting sentences with one of six different emotions (anger, disgust, fear, happy, neutral and sad). For HEAR 2021, we only use the audio recordings (which differs from much but not all of the literature).

**ESC-50** This is a multiclass classification task on environmental sounds. The ESC-50 dataset is a collection of 2000 environmental sounds organized into 50 classes (Piczak, 2015). Scores are averaged over 5 folds. (The folds are predefined in the original dataset.)

**FSD50K** FSD50K is a multilabel task (Fonseca et al., 2020). This dataset contains over 100 hours of human-labeled sound events from Freesound (https://freesound.org/). Each of the ≈51k audio clips is labeled using one or more of 200 classes from the AudioSet Ontology, encompassing environmental sounds, speech, and music. Unlike the other datasets, for FSD50K scene embeddings we did not alter the audio clip length. Each clip is between 0.3 and 30 seconds long. We use the predefined train/val/eval split. Evaluation is done using mean average precision (mAP).

**Gunshot Triangulation** Gunshot triangulation is a novel resource multiclass classification task that utilizes a unique dataset: gunshots recorded in an open field using iPod Touch devices (Cooper and Shaw, 2020). This data consist of 22 shots from 7 different firearms, for a total of 88 audio clips, the smallest dataset in HEAR. Each shot is recorded using four different iPod Touches, located at different distances from the shooter. The goal of this task is to classify audio by the iPod Touch that recorded it, i.e., to identify the location of the microphone. The dataset was split into 7 different folds, where each firearm belonged to only one fold. Results are averaged over each fold.

23
**GTZAN Genre**  The GTZAN Genre Collection (Tzanetakis and Cook, 2002) is a dataset of 1000 audio tracks (each 30 seconds in duration) that are categorized into ten genres (100 tracks per genre). The task is multiclass classification. As per the literature, scores are averaged over 10 folds. However, we don’t used the corrected artist-conditional splits from (Sturm, 2013).

**GTZAN Music Speech**  GTZAN Music Speech is a binary classification task, where the goal is to distinguish between music and speech. The dataset consists of 120 tracks (each 30 seconds in duration) and each class (music/speech) has 60 examples.

**LibriCount**  LibriCount is a multiclass speaker count identification task (Stöter et al., 2018b). The dataset contains audio of a simulated cocktail party environment with between 0 to 10 speakers. The goal of this task is to classify how many speakers are present in each of the recordings. Following Stöter et al. (2018a), we treat this as a classification, not regression, problem.

**MAESTRO 5h**  This is a novel music transcription task adapted from MAESTRO. For HEAR 2021, we created a subsampled version that includes 5 hours of training and validation audio, in 120 second clips. To evaluate submissions, a shallow transcription model was trained on timestamp-based embeddings provided by the participant models.

  We use note onset FMS and note onset with offset FMS for evaluation, as per the original MAESTRO paper (Hawthorne et al., 2019) and the preceding Onsets and Frames paper (Hawthorne et al., 2018).

  Note onset measures the ability of the model to estimate note onsets with 50 ms tolerance and ignores offsets. Note onset w/ offset includes onsets as well as requires note duration within 20% of ground truth or within 50 ms, whichever is greater.

**Mridingham Stroke and Mridingham Tonic**  We used the Mridangam Stroke Dataset (Anantapadmanabhan et al., 2013) for two novel multiclass classification tasks: Stroke classification and Tonic classification. The Mridingam is a pitched percussion instrument used in carnatic music, which is a sub-genre of Indian classical music. This dataset comprises 10 different strokes played on Mridingams with 6 different tonics.

**Vocal Imitations**  Vocal Imitations (Kim et al., 2018a) is a novel multiclass classification task, where the goal is to match a vocal imitation of a sound with the sound that is being imitated. The dataset contains 5601 vocal imitations of 302 reference sounds, organized by AudioSet ontology. Given a vocal sound, the classification task is to retrieve the original audio it is imitating.

**VoxLingua107 Top 10**  This is a novel multiclass classification task derived from the VoxLingua107 dataset (Valk and Alumäe, 2021). The goal of the task is to identify the spoken language in an audio file. For HEAR 2021 we selected the top 10 most frequent languages from the development set, which resulted in just over 5 hours of audio over 972 audio clips.
Table 3: Properties of baseline and submitted models, including: whether the model processes raw audio (1-D) or spectrograms (2D); on what kind of data the model is pretrained; the number of million parameters; the size of the output embedding for scene and timestamp tasks; and the number of minutes the model spends embedding Speech Commands V2. We caution that embedding time is not the entire picture, if participants did not do simple speed optimizations. For example, the CREPE wrapper (also used by GURA) is known not to exploit GPU batch parallelism.

<table>
<thead>
<tr>
<th>Model</th>
<th>Input 1D</th>
<th>2D</th>
<th>Pretraining data</th>
<th># M params</th>
<th>Embed dim</th>
<th>Time min</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenL3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>4.7</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>wav2vec2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>315.4</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>CREPE</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>22.2</td>
<td>2048</td>
<td>2048</td>
</tr>
<tr>
<td>AMAAI Lab wav2vec2+DDSP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>98.8</td>
<td>871</td>
<td>871</td>
</tr>
<tr>
<td>AMAAI wav2vec2 music+speech</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>300.0</td>
<td>768</td>
<td>768</td>
</tr>
<tr>
<td>Audioshake UDONS ViT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>11.1</td>
<td>768</td>
<td>768</td>
</tr>
<tr>
<td>CP-JKU PaSST 2vl</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>86.2</td>
<td>1295</td>
<td>2590</td>
</tr>
<tr>
<td>CP-JKU PaSST 2vl+mel</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>86.2</td>
<td>1295</td>
<td>3358</td>
</tr>
<tr>
<td>CP-JKU PaSST base</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>86.2</td>
<td>1295</td>
<td>1295</td>
</tr>
<tr>
<td>CVSSP PANNS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>80.8</td>
<td>2048</td>
<td>2048</td>
</tr>
<tr>
<td>Descript/MARL Wav2CLIP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>11.7</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>GURA Avg H+w+C</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1339.0</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>GURA Avg Hubert+Crepe</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1022.0</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>GURA Avg Hubert+wav2vec2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>634.0</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>GURA Cat H+w+C</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1339.0</td>
<td>3072</td>
<td>3072</td>
</tr>
<tr>
<td>GURA Cat Hubert+wav2vec2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>634.0</td>
<td>2048</td>
<td>2048</td>
</tr>
<tr>
<td>GURA Cat wav2vec2+crepe</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>339.0</td>
<td>2048</td>
<td>2048</td>
</tr>
<tr>
<td>GURA Fuse Cat H+w+C</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1339.0</td>
<td>3072</td>
<td>3072</td>
</tr>
<tr>
<td>GURA Fuse Cat H+w+C (time)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1339.0</td>
<td>15360</td>
<td>3072</td>
</tr>
<tr>
<td>GURA Fuse Hubert</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1000.0</td>
<td>1280</td>
<td>1280</td>
</tr>
<tr>
<td>GURA Fuse wav2vec2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>317.0</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>GURA Hubert</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1000.0</td>
<td>1280</td>
<td>1280</td>
</tr>
<tr>
<td>IUT-CSE MLP (audio)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.2</td>
<td>1584</td>
<td>8</td>
</tr>
<tr>
<td>IUT-CSE MLP (keyword)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.4</td>
<td>1024</td>
<td>64</td>
</tr>
<tr>
<td>Logitech AI SERAB BYOL-S</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>5.3</td>
<td>2048</td>
<td>2048</td>
</tr>
<tr>
<td>RedRice EfficientNet-B2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>7.7</td>
<td>1408</td>
<td>1408</td>
</tr>
<tr>
<td>Soundsensing YAMNet</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>3.8</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>Stellenbosch LSL DBERT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>316.8</td>
<td>2048</td>
<td>2048</td>
</tr>
</tbody>
</table>
Figure 2: t-SNE visualizations of tasks and models, based upon normalized scores. Missing normalized scores were imputed using sklearn’s multivariate IterativeImputer.
Figure 3: Task versus task correlation scores, based upon normalized scores. Only the highest and lowest correlations are displayed. Cells are sorted to minimize the traveling salesperson distance, mapping correlations [-1, +1] to distances [+2, 0].
Figure 4: Model versus model correlation scores, based upon normalized scores. Only the highest and lowest correlations are displayed. Cells are sorted to minimize the traveling salesperson distance, mapping correlations $[-1, +1]$ to distances $[+2, 0]$. 
Appendix B. Downstream training details

For each task, using a given model’s frozen embeddings as input features, we train a downstream MLP classifier. For scene-based multiclass tasks, the final layer is a softmax with cross-entropy loss. For scene-based multilabel tasks and multilabel frame reductions of timestamp tasks, the final layer is a sigmoid with cross-entropy loss.

We monitor the score (not loss) on the validation set. For timestamp tasks, computing the validation score involves a full CPU-based `sedeval` (Mesaros et al., 2016) run with median filter of 250ms and minimum event duration 125 ms and 250 ms. (Both event durations are tried at each validation step and the best hyperparameter is retained for that validation step.) We train for a maximum of 500 epochs, checking the validation score every 3 epochs, early stopping if no improvement is seen after 20 validation steps. For DCASE 2015 Task 2, we check the validation score every 10 epochs.

The validation score is used for early-stopping, as well as for model selection. The same RNG seed is used for every model-task downstream training, ensuring that grid points and weight initialization is identical. Model selection is performed over 8 deterministic random grid points out of 16 possible grid points. Hyperparameters are shown in Table 4. This grid was chosen after using a much larger hyperparameter grid with the three baseline models on the open tasks. In these preliminary hyperparameter grid pruning experiments, the grid was progressively refined by discarding hyperparameter choices that were not predictive of relatively high model performance, similarly to how Kelz et al. (2016) use tree ensemble learning to prune their hyperparameter grid.

Table 4: Hyperparameters used for training.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden layers</td>
<td>[1, 2]</td>
</tr>
<tr>
<td>Hidden dimensions</td>
<td>1024</td>
</tr>
<tr>
<td>Dropout</td>
<td>0.1</td>
</tr>
<tr>
<td>Learning rate</td>
<td>[3.2e-3, 1e-3, 3.2e-4, 1e-4]</td>
</tr>
<tr>
<td>Batch size</td>
<td>1024</td>
</tr>
<tr>
<td>Hidden norm</td>
<td>Batch Norm</td>
</tr>
<tr>
<td>Initialization</td>
<td>[Xavier Uniform, Xavier Normal]</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
</tbody>
</table>

(Glorot and Bengio, 2010)