# Learning Visual Localization of a Quadrotor Using Its Noise as Self-Supervision

Mirko Nava, Antonio Paolillo, *Member, IEEE*, Jérôme Guzzi, Luca Maria Gambardella,
and Alessandro Giusti

***Abstract*—We introduce an approach to train neural network models for visual object localization using a small training set, labeled with ground truth object positions and a large unlabeled one. We assume that the object to be localized emits sound, which is perceived by a microphone rigidly affixed to the camera. This information is used as the target of a cross-modal pretext task: predicting sound features from camera frames. By solving the pretext task, the model draws self-supervision from visual and audio data. The approach is well suited to robot learning: we instantiate it to localize a small quadrotor from 128 × 80 pixel images acquired by a ground robot. Experiments on a separate testing set show that introducing the auxiliary pretext task yields large performance improvements: the Mean Absolute Error (MAE) of the estimated image coordinates of the target is reduced from 7 to 4 pixels; the MAE of the estimated distance is reduced from 28 cm to 14 cm. A model that has access to labels for the entire training set yields an MAE of 2 pixels and 11 cm, respectively.**

***Index Terms*—Deep learning for visual perception, deep learning methods.**

## I. Introduction

Robot perception tasks often involve estimating spatial information, such as the pose of an object of interest (OOI), from high-dimensional data, e.g. images acquired by onboard cameras; deep learning models such as convolutional neural networks (CNNs) are a standard tool to solve this kind of problems. If no pre-trained model is available for a task of interest, one standard approach is to acquire large labeled training datasets, as representative as possible of the environment in which the robot will be deployed. Each image in the dataset is labeled with the corresponding relative pose of the OOI, obtained for example through an external tracking system. Then, one trains a deep learning model in a supervised way. However, the acquisition of such labeled datasets is expensive and not always feasible. Therefore, recent research leverages semi-supervised and self-supervised learning approaches, which combine a (typically small) labeled dataset with a large unlabeled dataset; in robotics applications, the latter can be acquired efficiently, by the robot itself, even during deployment.

A common strategy consists in using the unlabeled dataset for training an autoencoder; the encoder part is then used as a feature extractor for learning the perception task of interest in a supervised way using the labeled dataset [1]. Recent advances in the field of self-supervised learning bring this idea further. To solve a task of interest (*end task*), for which a limited labeled dataset is available, it is advantageous to simultaneously learn auxiliary *pretext tasks*, that are defined on large unlabeled datasets. In this context, pretext tasks should: i) require similar perception skills as the end task; ii) not require the explicit acquisition of ground truth labels, but instead use information in the data itself for supervision (hence, “self-supervised”). The intuition is that auxiliary tasks force the model to recognize patterns in the input that are similar to those that must be recognized to solve the task of interest, and thus aid in learning meaningful intermediate representations [2].

In this work, we apply this approach to a ground robot for learning a visual estimator of the position of a flying quadrotor, given the following training data: a small number of labeled samples for which the relative position of the drone is known; and a large number of unlabeled samples. The ground robot is also equipped with an uncalibrated stereo microphone, mounted at an arbitrary but fixed pose with respect to the camera, which picks up the noise of the quadrotor (see Fig. 1).

In this context, one could learn to estimate the drone position using the audio signal as an input to the model – a topic covered by the literature concerning sound source localization (see Section II). Our work does *not* aim at this goal. Instead, we use auditory perceptions to define a pretext task that aids in learning a purely-visual estimator; our model has one input (the camera frame) and predicts two outputs: the relative position of the drone (end task) and the intensity at different frequency bands of the corresponding sound (pretext task); the model is trained by minimizing the sum of the two respective regression losses: one (end loss) defined only on labeled instances; the other (pretext loss) defined on all instances. This is a novel application to robotics of self-supervised learning techniques, and represents our **main contribution**, presented in Section III. After training is completed, one can ignore the pretext task: the model only relies on visual data and can operate in arbitrarily noisy environments, or on robots without a microphone. The approach is validated

Manuscript received September 9, 2021; accepted January 7, 2022. Date of publication January 19, 2022; date of current version January 25, 2022. This letter was recommended for publication by Associate Editor S. Wang and Editor J. Yi upon evaluation of the reviewers’ comments. This work was supported by the Swiss National Science Foundation (SNSF) through the NCCR Robotics. *(Corresponding author: Mirko Nava.)*

The authors are with the Dalle Molle Institute for Artificial Intelligence (IDSIA), USI-SUPSI, 6962 Lugano, Switzerland (e-mail: mirko@idsia.ch; antonio.paolillo@supsi.ch; jerome@idsia.ch; luca@idsia.ch; alessandrog@idsia.ch).

This letter has supplementary downloadable material (Videos, code, and datasets are available at https://github.com/idsia-robotics/Sound-as-Pretext, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3143565

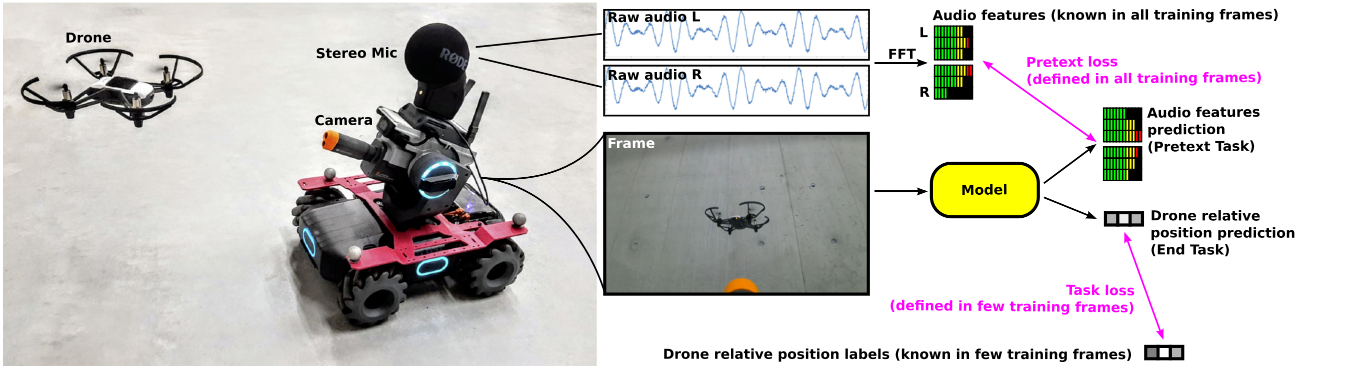
Drone
Stereo Mic
Camera
Raw audio L
Raw audio R
Frame
FFT
Audio features (known in all training frames)
L
R
Pretext loss
(defined in all training frames)
Audio features prediction (Pretext Task)
Model
Drone relative position prediction (End Task)
Task loss
(defined in few training frames)
Drone relative position labels (known in few training frames)

Fig. 1. Given an image from the ground robot's camera, our model estimates the relative position of the quadrotor; this is the *end task*, learned by minimizing a regression *end loss* on few training frames for which the true relative position is known. We show that simultaneously learning to predict audio features (*pretext task*), which are known in all training frames, yields significant performance improvements on the end task.

with experiments on extensive real-world datasets described in Section IV, which we release alongside the paper as a **secondary contribution**. The results, reported in Section V, show that learning with the sound-prediction pretext task yields models that perform significantly better than two baselines: one using no pretext task, and one using a standard image-reconstruction pretext task with an undercomplete autoencoder. Section VI concludes the paper.

## II. Related Work

### A. Self-Supervised Robot Perception

Self-supervised robot learning approaches extract supervisory information directly from a subset of the available data, or by refining it into usable target variables [2]; this allows the robot to collect data autonomously without external supervision. Self-supervised approaches show promising results in areas where perception plays an important role, including grasping [3]–[5], terrain classification [6], [7], obstacle avoidance [8], [9] and object localization [5], [10]–[12]. In grasping applications, supervision can be extracted from measuring the weight before and after a grasp attempt [3]; or from a model pre-trained in simulation and used to interact with the object, thus producing more training data [5]. In terrain classification, labels are generated from the Fourier transform of sound captured near the wheels of a mobile robot, then clustered together into terrain types [7]; or from torque sensors mounted on a quadruped robot [6]. In obstacle avoidance, labels are generated by assigning a negative label to frames close to a collision, and remaining frames a positive one [8]; or by predicting the distance to obstacles, as measured by laser sensors, from images [9]. For localization, the object's pose can be extracted through an ICP algorithm [10], [11], fitting the 3D model of the object in image space: Zeng *et al.* [10] use multiple RGB-D images to segment a single object with a background removal algorithm before applying ICP, while Xiang *et al.* [11] utilize monocular RGB images on which they directly apply ICP. In our previous work [12], we extract supervision from the robot's ego-motion and sparse ground truth generated by the robot's sensing. In contrast to the approaches mentioned above, this work does not use self-supervision to derive labels for the end task, but rather for a cross-modal pretext task that helps a model in solving the end task, using only a small amount of labels.

### B. Sound Source Localization

A large amount of literature investigates the problem of sound source localization [13]. Classic approaches compute hand-crafted audio features and algorithms based on sound propagation models to solve the problem; several recent approaches learn features directly from data with neural networks (NNs) [14]–[18]. Takeda *et al.* [14] propose to use a NN on a directional activator, while Yalta *et al.* [15] utilize a more modern ResNet architecture on the power spectrogram extracted from the short-time Fourier transform (STFT). When localizing from audio information, approaches suffer from noise present in the environment or produced by the robot itself, and from reverberations coming from different surfaces [13]. Takeda *et al.* [16] tackle these challenges by minimizing the cross-entropy loss computed on data collected in the deployment environment. Ferguson *et al.* [17] propose to jointly minimize two loss functions: a polar loss based on the direction of the source, and a mean squared error loss based on the distance of the source. He *et al.* [18] focus on the localization of multiple sources, by estimating a spatial spectrum that can be decoded into the locations of the sound sources. To aid the learning process, the network output is constrained to predictions that are coherent with the number of known sources present in a given sample.

In contrast to this body of work, our goal is not to localize the sound source from audio; instead, we take advantage of the (unknown) correlation between audio features and the position of the OOI – which is expected to emit sound at least while collecting training data – to learn a better visual feature representation.

### C. Audio in Self-Supervised Learning

Audio is a rich source of supervision utilized in many recent works to learn useful visual feature representations from data. Learned features are then used to solve classification problems [19], [20] or to identify the region of interest responsible for the sound [21]–[25]. Video classification is done by training a CNN to predict audio features from images [19], or by using one network per modality [20]: features extracted from images are clustered into classes and used as labels for the audio network, while audio features are used as labels for the image network. Then, both approaches classify the video by clustering together video frames based on the predicted audio features. Similarly to Owens *et al.* [19], our approach solves the task of predicting audio features from images; however, we do so in a robotics context to improve the performance on the end task of visually localizing an object from images. Coarse-grained object localization can be achieved by identifying the region of interest responsible for the sound, obtained by applying class activation mapping to a trained network: Owens *et al.* [21] train the network by extracting audio samples and frames from a video and predicting the probability that the two streams are temporally aligned. Instead, Arandjelovic *et al.* [22] use a triplet loss, in which positive examples are image-sound pairs coming from the same video, while negative pairs are taken from different videos; whereas Korbar *et al.* [23] train on hard-negatives, generated by sampling sound from the same video but at a different moment. Patrick *et al.* [24] generalize the approaches above, generating different tasks by choosing which transformation to apply to the data. Training is done by sampling random transformations pairs, and then predicting the probability for the pair to correspond to the same instant of a video. Arandjelovic *et al.* [25] train a model on image-sound correspondence: the feature map, computed as the scalar product of image and sound features, is used for a coarse-grained localization of the sound source.

## III. Sound Prediction as a Pretext Task for Visual Localization

We consider the problem of estimating from a monocular image the position relative to the camera reference frame of an OOI that, at least during training-data acquisition, produces sound. In addition to the camera video stream, we record audio from a microphone placed at a fixed pose with respect to the camera reference frame. In this context, we learn a NN model from a set of instances

$$\{\langle \boldsymbol{I}_i, \boldsymbol{f}_i, \boldsymbol{p}_i \rangle\}_{i=1}^{N} \tag{1}$$

where $\boldsymbol{I}$ denotes a camera frame, $\boldsymbol{f}$ audio features computed from the corresponding audio signals, and $\boldsymbol{p}$ the position of the OOI with respect to the camera reference frame. More specifically, the (possibly very small) subset of instances for which $\boldsymbol{p}$ is available, is denoted as the labeled training set $\mathcal{T}_\ell$; remaining instances compose the unlabeled training set $\mathcal{T}_u$, which is assumed to be much larger, as it can be acquired by a robot without external supervision.

### A. Model

We learn a Sound as Pretext (SaP) model that, given the image $\boldsymbol{I}$, estimates:

- $\hat{\boldsymbol{p}}$, the relative position of the OOI (*end task*); this task is trained using data in $\mathcal{T}_\ell$, and is the task whose performance we are interested in optimizing.
- $\hat{\boldsymbol{f}}$, the corresponding audio features (*pretext task*); this task is trained using data in $\mathcal{T}_\ell \cup \mathcal{T}_u$, and is of no direct use for model deployment.

The model implements a function parametrized by $\boldsymbol{\theta}$

$$\left(\hat{\boldsymbol{p}}, \hat{\boldsymbol{f}}\right) = \boldsymbol{m}_{\text{SaP}}\left(\boldsymbol{I}|\boldsymbol{\theta}\right). \tag{2}$$

Training consists in obtaining optimal values of $\boldsymbol{\theta}$ by minimizing through gradient descent the following loss function

$$\frac{\lambda}{|\mathcal{T}_\ell|}\sum_{i=1}^{|\mathcal{T}_\ell|}\mathcal{L}_{\text{end}}(\boldsymbol{p}_i, \hat{\boldsymbol{p}}_i) + \frac{1}{|\mathcal{T}_u \cup \mathcal{T}_\ell|}\sum_{i=1}^{|\mathcal{T}_u \cup \mathcal{T}_\ell|}\mathcal{L}_{\text{pretext}}(\boldsymbol{f}_i, \hat{\boldsymbol{f}}_i) \tag{3}$$

where $\mathcal{L}_{\text{end}}$ and $\mathcal{L}_{\text{pretext}}$ both compute the absolute error between their two arguments, $|\cdot|$ denotes the cardinality of a set, and $\lambda$ acts as a tradeoff on the optimization of the two losses. Note that training instances in $\mathcal{T}_u$ have no label $\boldsymbol{p}_i$.

### B. Application to Quadrotor Position Estimation

In the rest of the paper, we instantiate the general approach presented above to a concrete problem: learn from scratch a visual estimator of the position of a flying quadrotor from images captured by a ground robot, using audio data acquired by an onboard stereo microphone to aid learning. We exploit a small amount of training data $\mathcal{T}_\ell$, in which the position of the drone relative to the camera is known from a motion tracking system; and a large unlabeled training set $\mathcal{T}_u$, acquired by the robot without supervision.

The scenario is attractive because it matches many similar robot perception tasks of wide interest, in which deep learning models must be trained from scratch and large pre-existing labeled datasets are not available. Acquiring those labeled datasets ad-hoc is potentially very expensive and time-consuming; exploiting self-supervision from cross-modal cues is in this context an attractive alternative.

## IV. Experimental Setup

### A. Robot Platforms and Sensors

We use a modified DJI RoboMaster S1 ground robot (see Figs. 1 and 2); the robot features a controllable pan-tilt turret and four Swedish wheels, which provide omnidirectional motion capabilities. On the turret, the robot is equipped with: an integrated RGB camera acquiring images at a resolution of 1280 × 720 pixels; a stereo microphone (RØDE Stereo VideoMic Pro), mounted in such a way that the frontal direction is aligned with the camera optical axis; an NVIDIA Jetson Nano single-board computer mounted on the back; and an Intel RealSense camera that is not used in our experiments.

The microphone uses a matched pair of high sensitivity 0.5 in cardioid condenser capsules mounted in a coincident XY stereo configuration [26] (Fig. 2); therefore, the two capsules point $45^\circ$ to the left (L) and right (R) of the camera optical axis. Each capsule has a cardioid pickup pattern that attenuates sounds depending on the angle of the sound source with respect to the axis of the capsule, as depicted in Fig. 2 (right): sounds coming from a direction aligned with the capsule axis are not attenuated

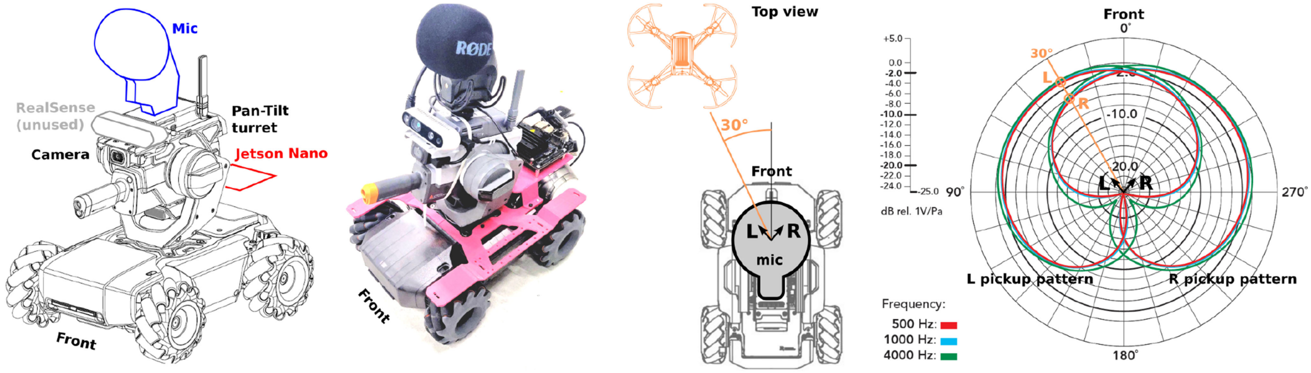

Fig. 2. From left to right. Technical drawing of the RoboMaster S1 platform (black) and some of our addons (gray, blue, red). Photograph of the real platform. Top view of the robot with placement and orientation of the two condenser capsules (L, R) in the stereo mic. Polar pickup pattern for the two cardioid capsules for three different frequencies. Orange elements refer to the example of direct sound coming from a drone at a 30° angle, see text.

(0 dB); sounds coming at an angle of 90° are attenuated by approximately 6 dB.

As an example, Fig. 2 shows the case of a drone (orange) hovering at the same height of the microphone, in a position that lies at an angle of 30° to the left of the camera optical axis; we further assume that the microphone is pointing horizontally (turret tilt equal to 0). Interpreting the polar plot, we expect that the *direct* noise from the drone will be attenuated by approximately $-0.5$ dB in the left channel (15° with respect to the capsule axis), and by approximately -4 dB in the right channel (75° with respect to the capsule axis); note that in most indoor environments including ours, a significant part of the drone noise will not originate directly from the drone direction, but instead be reflected by the surrounding environment as reverberations. Note also that low frequencies (red) are attenuated more than high frequencies (green). Noise and rumble induced on the microphone from robot movements and vibrations can be attenuated with an optional anti-shock acoustic suspension (RØDE Rycote Lyre). It is important to remark that the configuration of the microphone setup is not assumed known in our system; our only assumption is that there exists some unknown, potentially weak, and potentially nonlinear correlation between the drone relative position (target variables for the end task) and the corresponding audio features (target variables for the pretext task). If this is the case, solving the pretext task indirectly favors learning visual features that encode the drone position.

The drone we use in our experiments is a Ryze Tello quadrotor; the RoboMaster turret and the Tello body are outfitted with infrared reflective markers and tracked during data collection by a motion tracking system (12 Optitrack Prime-13 cameras), which provides the precise Tello position w.r.t. the RobotMaster's camera frame[1] used as labels in the training procedures.

### B. Data Collection

We collect data in 22 different recording sessions taking place in a laboratory environment, comprising a total of approximately 50 minutes of data. During each session, the Tello is teleoperated by a user with a joypad, flying 3D trajectories that attempt to stay within a maximum distance of 3m from the RoboMaster, such that the drone remains visible in the scaled-down camera feed used for learning (128 × 80 pixels); at the same time, a different user teleoperates the RoboMaster to move in the environment and pan/tilt the turret to keep the Tello visible in the camera feed – this ensures that the image background is very variable in the entire dataset. For each session, we record camera frames at 30 Hz, raw stereo audio from the microphone, and absolute pose information of the RoboMaster turret and Tello body. A set of instances is then generated associating to each camera frame $\boldsymbol{I}$ the corresponding audio features $\boldsymbol{f}$ and drone position $\boldsymbol{p}$ with respect to the camera. Following [11], the position is represented as the normalized coordinates $x$ and $y$ of the drone as it appears on the image plane, and its Cartesian distance $d$ from the camera. Instances in which the drone is not visible (defined as having the drone position projected in the image plane out of the bounds of the camera frame) are discarded.

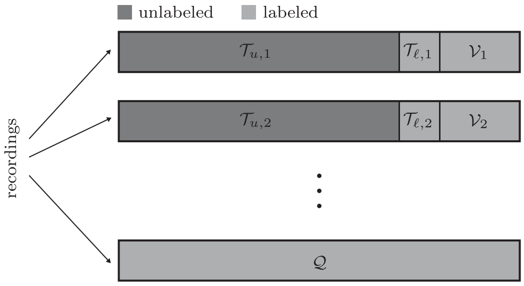

Fig. 3. Data is collected in 22 recording sessions, and then split in unlabeled ($\mathcal{T}_u$) and labeled ($\mathcal{T}_\ell$) training sets; validation set ($\mathcal{V}$); and testing set ($\mathcal{Q}$).

All instances from one recording are used in the following as a testing set $\mathcal{Q}$, on which we compute performance metrics. Instances from the remaining 21 recordings are partitioned into three sets: the first 70% of the frames as the unlabeled training set $\mathcal{T}_u$, the following 10% for the labeled training set $\mathcal{T}_\ell$, and the remaining 20% for the labeled validation set $\mathcal{V}$, which is only used to monitor models during training (see Fig. 3). In total, we collected 89 k instances, of which 60 k are in $\mathcal{T}_u$, 8 k in $\mathcal{T}_\ell$, 17 k in $\mathcal{V}$, and 4 k in $\mathcal{Q}$.

[1] The offset between the OptiTrack markers placed on RobotMaster turret and its camera frame is taken into account.

### C. Audio Features

Audio features corresponding to a given frame acquired at time $t$ are extracted from raw audio data acquired in a time window of 0.15s centered on $t$ (6615 audio samples at 44.1 kHz); within this short time window, we assume that the quadrotor noise profile is approximately stationary. For each of the two channels, we use the discrete Fourier transform to compute the logarithm of the average magnitude of the frequency spectrum within each of three frequency bands: 1 to 2 kHz; 2 to 5 kHz; and 5 to 15 kHz. This yields 3 features for each channel, which can be interpreted as the values of a log-spectrogram of the audio signal, sampled at a single timestamp, and average-pooled along the frequency axis. We further compute the difference between the features in the two channels, for a total of 9 features, because we expect intensity differences between the two channels to be correlated to the drone position relative to the robot camera and mic. We consider multiple frequency bands since, depending on the angle of the sound source, each band is attenuated by a different amount, and that environmental reflections are more prevalent in some frequencies than in others. In contrast, because our microphone uses virtually-coincident positions for the two capsules, our approach can not rely on inter-channel time difference cues [27].

### D. Alternative Strategies

To validate the rationale of our approach, we analyze and compare its performance with several alternatives. First, we train the Baseline (B) model only on the end task (i.e., $\mathcal{L}_{\text{pretext}}$ is not considered in (3)), using data in $\mathcal{T}_\ell$. To estimate the maximum achievable performance, we train the Upper Bound (UB) model, using only the end task as in the previous case, but considering the whole training set as if it was all labeled. The model is trained with labels for $\mathcal{T}_u$ that we collected only for this purpose and ignored for all other models. A common approach in the robotics literature to exploit unlabeled image data is to train on such data an undercomplete autoencoder [1], and use the resulting compressed representation as features for subsequently learning the end task; the expectation is that these features encode high-level information in the original image. In our context, this corresponds to using autoencoding as a pretext task. Therefore, we train the Autoencoding as Pretext (AaP) model, by simultaneously considering the end task (on $\mathcal{T}_\ell$), as well as the autoencoding pretext task (on $\mathcal{T}_u \cup \mathcal{T}_\ell$). Taking inspiration from sound source localization approaches, we consider the Audio Only (AO) model which is trained solely on regressing the position from audio features $\boldsymbol{f}$, using $\mathcal{T}_\ell$. Similar to other audio-only approaches [14], it is negatively affected by noise generated by the robot itself, environment reverberations, and other external sound sources, which makes the approach less desirable for deployment in unstructured scenarios.

When provided with data from multiple sensors, a common approach in the literature is to fuse readings coming from different sensors together [28], [29]. To explore this option, we train the Sensor Fusion (SF) on the task of regressing the drone position from both image and audio features, using only instances in $\mathcal{T}_\ell$.

We also compare SaP, trained using our 9-dimensional audio features, with two models trained using different audio features. The first, SaP-Mono, uses only 3 features ($\boldsymbol{f}_{\text{Mono}}$) obtained as the average of the features for left and right channels, actually emulating a mono-channel microphone. The latter, named SaP-Mel, is trained using a richer audio representation ($\boldsymbol{f}_{\text{Mel}}$) based on Mel-spectrograms [30]. More specifically, we compute for each channel a 64-band Mel-spectrogram in the range 1 to 15 kHz, using 0.15 s windows; for each frame, we associate the $64 \times 2$ values sampled from the two spectrograms at the corresponding time, plus their difference, for a total of $64 \times 3$ audio features.

### E. Neural Network Architectures and Training

Most of the strategies considered in this work employ a CNN architecture based on MobileNet-V2 [31], with a total of 1 million parameters, and a variable number of output neurons dependant on the chosen strategy (3 for strictly supervised approaches, 12 for the SaP model). The SF model utilizes the same convolutional architecture for the image branch, while a series of 4 feed-forward layers with ReLU non-linearities processes the audio information and 3 more feed-forward layers fuse the two streams, similarly to [29]. The AaP model implements an encoder-decoder CNN architecture, with a bottleneck of size 128 [1]. A separate head takes the latent representation and through a series of 3 feed-forward layers with ReLU non-linearities produces a prediction of the position. The AO model is composed of a series of 5 feed-forward layers with ReLU non-linearities, for a total of 60 k parameters. For all models, training uses the Adam [32] optimizer with an initial learning rate of $10^{-3}$, which is reduced by a factor of 10 halfway through the training process, which lasts a total of 60 epochs. In designing our loss, we choose a tradeoff factor $\lambda = 1$. Batches of size 64 (or possibly less, for the last one) are drawn from either of the two training datasets, and the corresponding loss is used for the optimization.

## V. Results

This section explores the effectiveness of our approach against alternatives (Section V-A), different choices for the sound features (Section V-B), the impact on the amount of labeled data (Section V-C), and a comparison with approaches that use audio features as inputs (Section V-D).

For our evaluation, we consider the following metrics computed on the entire testing set $\mathcal{Q}$: the mean absolute error (MAE) of the prediction w.r.t. the ground truth, computed separately for $xy$ (MAE$_{xy}$, expressed in pixels), and distance (MAE$_d$, expressed in cm); and the coefficient of determination $R^2$ for each of the three components of the relative position, denoted with $R^2_x$ ($x$ image coordinate), $R^2_y$ ($y$ image coordinate) and $R^2_d$ (distance from the camera).

The coefficient of determination is a standard adimensional metric for regression performance. It represents the fraction of the variance of the target variable that is correctly explained by the model (higher values are better). A trivial model that predicts the average of the target variable in the whole testing dataset yields $R^2 = 0$; an ideal model yields $R^2 = 1.0$; a model

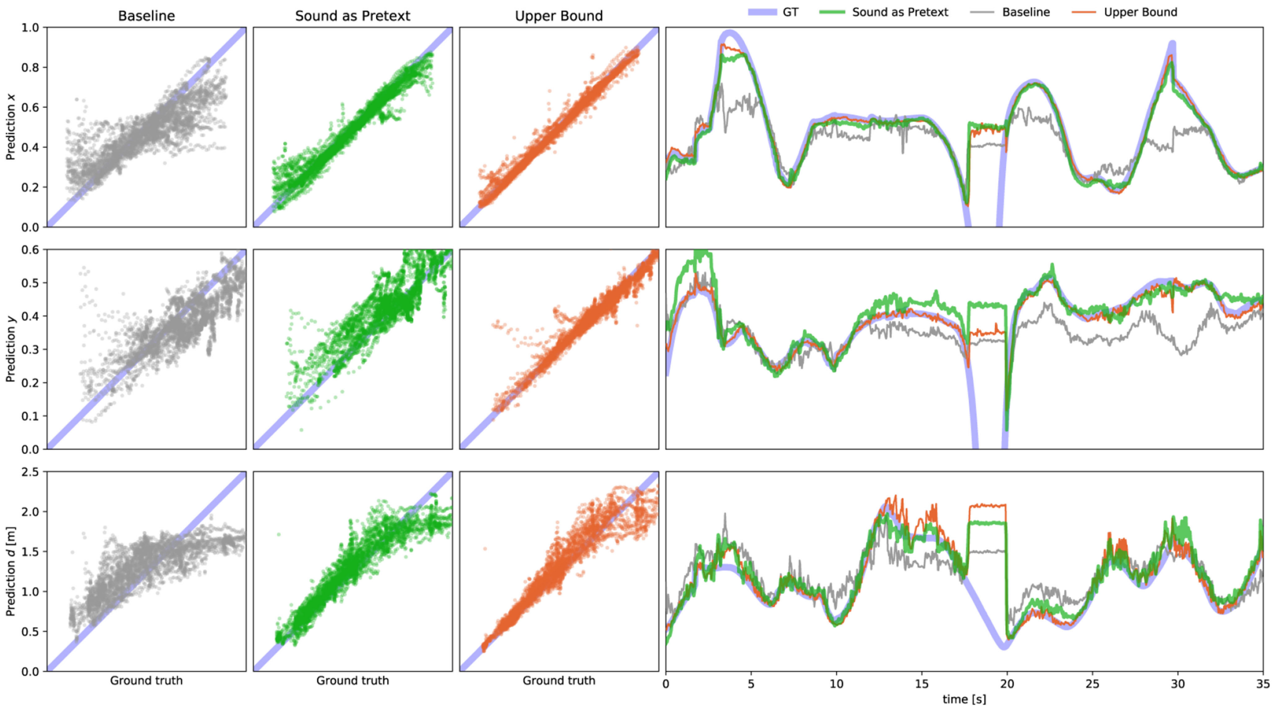

Fig. 4. End Task Regression Performance on the testing set $\mathcal{Q}$. On the left side we compare ground truth ($x$-axis) and predictions ($y$-axis) for different models (columns) and variables (rows). On the right, predictions on 35s of the testing set. Between seconds 17 and 20 the drone exits of the camera FOV, causing all models to temporarily fail.

might have a negative $R^2$ in case its MAE exceeds the variance of the data, which frequently occurs with weak models operating on high-dimensional inputs. The $R^2$ metric enables meaningful comparison of regression performance on different variables, since, unlike the MAE, it does not depend on the variance of the target variable.

### A. *Sound as Pretext Improves Performance*

In the top section of Fig. 6, we compare the performance of our SaP approach with baselines, alternatives, and a Dummy model that simply returns the average of each target variable in the training set. We observe that compared to the baseline B, our approach significantly improves performance on $x$ and $d$, and to a lesser extent on $y$. Considering the relative improvement in terms of $R^2$ of each variable, where B represents the 0% reference and UB is considered as the maximum achievable performance (100%), our approach reaches 87% improvement on $x$, 32% on $y$ and 88% on $d$.

The $x$ and $d$ variables are most directly related to audio features: for example, $x$ affects the intensity difference between the two audio channels, and $d$ the intensity on both channels; on these two variables, SaP yields large improvements over the baseline. This is because the pretext task induces the model to learn visual features that capture well the horizontal position of the quadrotor and its distance. In contrast, the $y$ variable is only weakly and non-monotonically related to audio features (and in fact, $y$ can not be estimated well by the AO model). Still, our pretext task significantly improves performance on the $y$ variable; one possible explanation is that the same visual features that capture $x$ and $d$ are also useful to estimate $y$.

Fig. 4 shows scatter plots comparing ground truth to predictions of the baseline, our approach, and the upper bound, as well as a qualitative comparison of their predictions on 35s of data taken from the testing set. Considering the scatter plots on the left, our approach improves over the baseline, having a tighter distribution that is closer to the diagonal line representing the ideal case; this confirms the quantitative evaluation of Fig. 6. It can be noticed how all models correctly predict the distance $d$ when the drone is close to the camera, while longer distances are harder to estimate. Regarding the time plot on the right in Fig. 4, comparing different model predictions on a portion of $\mathcal{Q}$ shows that our approach follows closely the upper bound, while the baseline struggles when the drone is not in the central area of the image. Between seconds 4 and 5 all models struggle in predicting the horizontal location of the drone: this is explained by the drone moving close to the edge of the field of view; similarly, between seconds 17 and 20, the drone briefly exits the camera's field of view, causing all models to predictably fail. In Fig. 5 we present a qualitative evaluation of SaP on 10 camera frames taken from the testing set. Failure cases, in which the model's prediction (green) does not overlap with the ground truth (blue), occur when the drone blends with a cluttered background, or when it reaches a distance greater than 3 m, rendering the quadrotor recognition from very few pixels difficult.

### B. *Impact of Sound Features*

To further explore this fascinating finding, we also trained SaP-Mono, the version of our approach that uses as the target of the pretext task only 3 features. Without access to stereo information, audio features do not allow discriminating whether

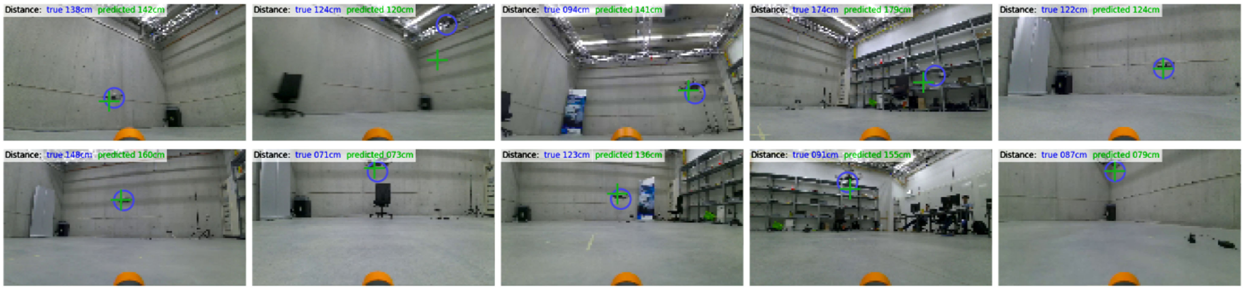
Distance: true 138cm predicted 142cm
Distance: true 124cm predicted 120cm
Distance: true 094cm predicted 141cm
Distance: true 174cm predicted 179cm
Distance: true 122cm predicted 124cm
Distance: true 148cm predicted 160cm
Distance: true 071cm predicted 073cm
Distance: true 123cm predicted 136cm
Distance: true 091cm predicted 155cm
Distance: true 087cm predicted 079cm

Fig. 5. Predictions of the Sound as Pretext model (green cross) compared to ground truth (blue circle) on ten frames taken from the testing set. .

| Model | Input | Output | Training set for<br>End Task | <br>Pretext Task | MAE$_{xy}$<br>[px] | MAE$_d$<br>[cm] | $R^2_x$<br>[%] | $R^2_y$<br>[%] | $R^2_d$<br>[%] |
|---|---|---|---|---|---|---|---|---|---|
| Dummy | – | – | – | – | 14 | 50 | 0 | $-4$ | $-23$ |
| B | $\boldsymbol{I}$ | $\boldsymbol{p}$ | $\mathcal{T}_\ell$ | – | 7 | 28 | 66 | 56 | 58 |
| AaP | $\boldsymbol{I}$ | $\boldsymbol{I}, \boldsymbol{p}$ | $\mathcal{T}_\ell$ | $\mathcal{T}_\ell \cup \mathcal{T}_u$ | 14 | 47 | $-2$ | $-9$ | $-12$ |
| SaP-Mono | $\boldsymbol{I}$ | $\boldsymbol{f}_{\text{Mono}}, \boldsymbol{p}$ | $\mathcal{T}_\ell$ | $\mathcal{T}_\ell \cup \mathcal{T}_u$ | 6 | 15 | 83 | 52 | 86 |
| SaP-Mel | $\boldsymbol{I}$ | $\boldsymbol{f}_{\text{Mel}}, \boldsymbol{p}$ | $\mathcal{T}_\ell$ | $\mathcal{T}_\ell \cup \mathcal{T}_u$ | 6 | 22 | 82 | 66 | 71 |
| SaP *(ours)* | $\boldsymbol{I}$ | $\boldsymbol{f}, \boldsymbol{p}$ | $\mathcal{T}_\ell$ | $\mathcal{T}_\ell \cup \mathcal{T}_u$ | **4** | **14** | **94** | **68** | **87** |
| SaP-60 | $\boldsymbol{I}$ | $\boldsymbol{f}, \boldsymbol{p}$ | 60% $\mathcal{T}_\ell$ | 60% $\mathcal{T}_\ell \cup \mathcal{T}_u$ | 7 | 24 | 72 | 53 | 63 |
| B-60 | $\boldsymbol{I}$ | $\boldsymbol{p}$ | 60% $\mathcal{T}_\ell$ | – | 13 | 40 | 10 | $-28$ | 14 |
| SaP-30 | $\boldsymbol{I}$ | $\boldsymbol{f}, \boldsymbol{p}$ | 30% $\mathcal{T}_\ell$ | 30% $\mathcal{T}_\ell \cup \mathcal{T}_u$ | 8 | 31 | 73 | 13 | 45 |
| B-30 | $\boldsymbol{I}$ | $\boldsymbol{p}$ | 30% $\mathcal{T}_\ell$ | – | 14 | 43 | $-3$ | $-12$ | $-4$ |
| SaP-10 | $\boldsymbol{I}$ | $\boldsymbol{f}, \boldsymbol{p}$ | 10% $\mathcal{T}_\ell$ | 10% $\mathcal{T}_\ell \cup \mathcal{T}_u$ | 9 | 32 | 68 | $-55$ | 47 |
| B-10 | $\boldsymbol{I}$ | $\boldsymbol{p}$ | 10% $\mathcal{T}_\ell$ | – | 15 | 49 | $-2$ | $-88$ | $-21$ |
| AO | $\boldsymbol{f}$ | $\boldsymbol{p}$ | $\mathcal{T}_\ell$ | – | 7 | 18 | 87 | $-23$ | 81 |
| SF | $\boldsymbol{I}, \boldsymbol{f}$ | $\boldsymbol{p}$ | $\mathcal{T}_\ell$ | – | 6 | 27 | 83 | 68 | 60 |
| UB | $\boldsymbol{I}$ | $\boldsymbol{p}$ | $\mathcal{T}_\ell \cup \mathcal{T}_u^\star$ | – | 2 | 11 | 98 | 93 | 91 |

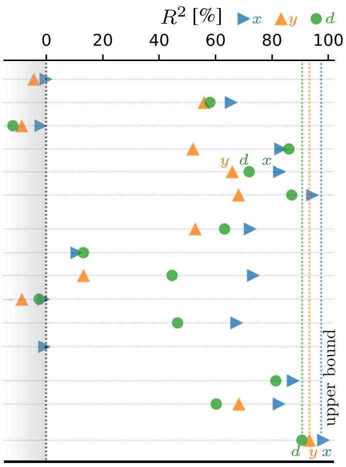

Fig. 6. Summary of results; each model reports the position MAE (lower is better) and the $R^2$ metric for each output variable (higher is better).

the drone is at the left or right side of the image, but still allow good resolution concerning the drone distance; therefore, the beneficial effects of the cross-modal pretext task are reduced on $x$ and $y$, when compared with its stereo counterpart, while on $d$ SaP-Mono and SaP perform equally well. Predicting richer audio features as the pretext (SaP-Mel) still exhibits improvements over the baseline, but not as much as with our features; this is probably due to the larger dimensionality of Mel features, which exposes the model to overfitting issues while containing limited additional useful information when compared to our features. Whereas solving an autoencoding pretext task (AaP) yields very poor results, with a performance well below the baseline B. The reason is that autoencoding is a poor pretext task in this scenario: the drone is most often small in the input image and covers a small fraction of pixels; the autoencoding loss minimizes the image reconstruction error, and will not promote representing meaningful information concerning the drone position. Instead, we expect that the learned features will be dominated by modeling the different possible backgrounds, which cover most of the image and exhibit very high contrast. Unfortunately, the image background is exactly what we want our features to be insensitive to.

## C. Impact of the Labeled Set Size

To explore the impact of the amount of available labels for the end task, we trained the SaP model on decreasing amounts of labeled data while keeping the unlabeled data fixed. The second panel of Fig. 6 reports the respective results; thanks to the effectiveness of the sound pretext task in learning meaningful visual features, the $x$ variable can be estimated well even with 10% of the labeled data, corresponding to just 800 frames, while $d$ requires a larger amount of labels. In contrast, the performance on $y$ rapidly drops as fewer labeled instances are considered.

## D. Strategies Using Sound as Input

The previous analysis compared our approach against others using just images as input; we now extend our focus on strategies that utilize different modalities, whose results are reported in the third panel of Fig. 6. The AO model shows promising results on the variables $x$ and $d$ while having no predictive power on the $y$. This is easily explained by the microphone's geometry, for which the difference between sound intensity in the two channels is highly informative on the horizontal axis but not on the vertical axis; humans and animals overcome this issue by accounting for different spectral filtering of the ear geometry on sound coming from different elevations, or by actively tilting the head to better localize sound sources. Model SF can leverage images to estimate $y$, resulting in a higher $R^2$ score on that variable; while on $x$ SF has a similar performance as AO, it is penalized on the distance $d$. In fact, images yield little additional information to audio when predicting distance, especially when the drone is far from the camera. Compared to all alternatives, SaP performs better on all three variables.

## VI. Conclusion

We presented Sound as Pretext, an approach for tackling visual object localization problems by employing a cross-modal pretext task, well suited to many applications of self-supervised robot learning. By collecting images of the quadrotor as well as its noise, we alleviate the need for a large labeled training dataset, and provide supervision to the model by adopting the auxiliary pretext task of predicting audio features from images. The approach requires only that the object to be localized produces sound during training data collection – a condition that, for silent objects, could be satisfied with the help of a wireless speaker – while during deployment no audio information is necessary. An extensive evaluation shows that our approach outperforms a supervised baseline, a standard image-reconstruction pretext task, and approaches that directly use audio, or a combination of vision and audio, to solve the same task.

## References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[2] L. Jing and Y. Tian, “Self-supervised visual feature learning with deep neural networks: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2021.

[3] L. Pinto and A. Gupta, “Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 3406–3413.

[4] E. Jang, C. Devin, V. Vanhoucke, and S. Levine, “Grasp2vec: Learning object representations from self-supervised grasping,” in *Proc. Conf. Robot Learn.*, 2018, pp. 99–112.

[5] X. Deng, Y. Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox, “Self-supervised 6D object pose estimation for robot manipulation,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 3665–3671.

[6] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, “Where should i walk? Predicting terrain properties from images via self-supervised learning,” *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1509–1516, Apr. 2019.

[7] J. Zürn, W. Burgard, and A. Valada, “Self-supervised visual terrain classification from unsupervised acoustic feature learning,” *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 466–481, Apr. 2021.

[8] D. Gandhi, L. Pinto, and A. Gupta, “Learning to fly by crashing,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 3948–3955.

[9] A. Kouris and C.-S. Bouganis, “Learning to fly by myself: A self-supervised CNN-based approach for autonomous navigation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–9.

[10] A. Zeng *et al.*, “Multi-view self-supervised deep learning for 6D pose estimation in the amazon picking challenge,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 1386–1383.

[11] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, “PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6693–6700, 2018.

[12] M. Nava, A. Paolillo, J. Guzzi, L. M. Gambardella, and A. Giusti, “Uncertainty-aware self-supervised learning of spatial perception tasks,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6693–6700, Oct. 2021.

[13] C. Rascon and I. Meza, “Localization of sound sources in robotics: A review,” *Robot. Auton. Syst.*, vol. 96, pp. 184–210, 2017.

[14] R. Takeda and K. Komatani, “Sound source localization based on deep neural networks with directional activate function exploiting phase information,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, pp. 405–409.

[15] N. Yalta, K. Nakadai, and T. Ogata, “Sound source localization using deep learning models,” *J. Robot. Mechatronics*, vol. 29, no. 1, pp. 37–48, 2017.

[16] R. Takeda and K. Komatani, “Unsupervised adaptation of deep neural networks for sound source localization using entropy minimization,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 2217–2221.

[17] E. L. Ferguson, S. B. Williams, and C. T. Jin, “Sound source localization in a multipath environment using convolutional neural networks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 2386–2390.

[18] W. He, P. Motlicek, and J.-M. Odobez, “Adaptation of multiple sound source localization neural networks with weak supervision and domain-adversarial training,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 770–774.

[19] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba, “Ambient sound provides supervision for visual learning,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 801–816.

[20] H. Alwassel, D. Mahajan, B. Korbar, L. Torresani, B. Ghanem, and D. Tran, “Self-supervised learning by cross-modal audio-video clustering,” *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 9758–9770, 2020.

[21] A. Owens and A. A. Efros, “Audio-visual scene analysis with self-supervised multisensory features,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 631–648.

[22] R. Arandjelovic and A. Zisserman, “Look, listen and learn,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 609–617.

[23] B. Korbar, D. Tran, and L. Torresani, “Cooperative learning of audio and video models from self-supervised synchronization,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7774–7785.

[24] M. Patrick *et al.*, “Multi-modal self-supervision from generalized data transformations,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021.

[25] R. Arandjelovic and A. Zisserman, “Objects that sound,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 435–451.

[26] R. Streicher and W. Dooley, “Basic stereo microphone perspectives-a review,” *J. Audio Eng. Soc.*, vol. 33, no. 7/8, pp. 548–556, 1985.

[27] T. Rodemann, G. Ince, F. Joublin, and C. Goerick, “Using binaural and spectral cues for azimuth and elevation localization,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 2185–2190.

[28] D. Xu, D. Anguelov, and A. Jain, “Pointfusion: Deep sensor fusion for 3D bounding box estimation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 244–253.

[29] M. Valente, C. Joly, and A. de La Fortelle, “Deep sensor fusion for real-time odometry estimation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 6679–6685.

[30] L. Rabiner and R. Schafer, *Theory and Applications of Digital Speech Processing*. Hoboken, NJ, USA: Prentice Hall Press, 2010.

[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[32] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Representations*, 2015.