Advanced Signal Processing Architectures for Monaural Acoustic Scene Analysis in Domestic Environments

1. Introduction: The Challenge of the Single-Channel Home

The modern domestic environment represents one of the most acoustically complex frontiers for artificial intelligence. Unlike the controlled silence of a recording studio or the predictable hum of a server room, a home is a chaotic superposition of diverse sound sources. From the impulsive clatter of cutlery and the rhythmic drone of a washing machine to the transient bark of a pet and the semantic richness of human speech, the "household soundscape" is a dynamic, non-stationary mixture. For a voice assistant, the ability to parse this mixture using only a single microphone (monaural input) is not merely a feature but a fundamental prerequisite for robust interaction. This report presents an exhaustive research analysis of the signal processing methodologies required to solve this problem, specifically targeting the separation and classification of household noises in a Python-based computational context.

The problem definition posits a strict constraint: the system must operate monaurally. This immediately disqualifies the spatial filtering techniques—such as beamforming or generalized sidelobe cancellation—that define multi-microphone arrays.[1] In a multi-channel setup, spatial covariance matrices allow the system to mathematically "steer" a listening cone towards a user while nulling out a TV in the corner. In a single-channel setup, this spatial dimension is lost. The system suffers from what is formally known as the **underdetermined source separation problem**, where the number of active sources (N) exceeds the number of sensors ($M=1$).[2] Mathematically, the observed signal $x(t)$ is a collapsed sum of all source signals $s_i(t)$ convolved with the room impulse response $h_i(t)$, plus additive background noise $n(t)$:

$$x(t) = \sum_{i=1}^{N} s_i(t) * h_i(t) + n(t)$$

Solving for $s_i(t)$ is an ill-posed inverse problem. Without spatial cues, the signal processing architecture must exploit **spectro-temporal sparsity** and **statistical independence**. It must identify that the harmonic stack of a voice differs from the broadband chaotic signal of running water, or that the temporal envelope of a door knock is distinct from the steady-state drone of a fan.[4]

This report surveys the entire landscape of potential solutions, from classical statistical signal processing to the vanguard of deep generative modeling. It then selects and rigorously details five specific, high-viability methodologies for implementation: **Supervised Non-Negative Matrix Factorization (NMF)**, **Conv-TasNet for Universal Sound Separation**, **Convolutional Recurrent Neural Networks (CRNN)**, **Audio Spectrogram Transformers (AST)**, and the **Mean Teacher** semi-supervised learning framework. These methodologies are chosen not just for their theoretical elegance but for their proven efficacy in recent benchmarks, such as the DCASE (Detection and Classification of Acoustic Scenes and Events) challenges, and their compatibility with the Python ecosystem (e.g., PyTorch, Asteroid, Librosa).[5]

---

2. A Comprehensive Survey of the Signal Processing Field

Before isolating the optimal methodologies for a modern voice assistant, it is necessary to survey the broader field of audio signal processing to understand why certain approaches are discarded and why others have risen to prominence. The history of monaural sound processing can be categorized into three distinct eras: the Era of Spectral Subtraction, the Era of Statistical Decomposition, and the Era of Deep Representation Learning.

2.1 Classical Filtering and Spectral Subtraction

The earliest attempts to clean single-channel audio relied on statistical assumptions about the noise floor. Algorithms like Spectral Subtraction and Wiener Filtering operate on the premise that noise is stationary—meaning its statistical properties (mean and variance) do not change significantly over time.8 In this paradigm, the system estimates a noise profile during periods of silence (Voice Activity Detection) and subtracts this profile from the active speech segments.

While computationally inexpensive, these methods are catastrophic for the "household" use case. Domestic noise is inherently non-stationary. A door slamming, a dog barking, or a plate dropping are transient events; they do not have a stable noise floor that can be estimated and subtracted.9 Applying a Wiener filter to non-stationary transients results in "musical noise"—artificial, metallic artifacts that can degrade the performance of downstream classifiers more than the original noise itself.9 Furthermore, these methods are primarily subtractive; they cannot separate two active foreground sources (e.g., a user speaking while the TV is on), they can only suppress a background floor.

2.2 Computational Auditory Scene Analysis (CASA)

Inspired by human hearing, CASA attempts to group Time-Frequency (T-F) bins based on perceptual cues like common onset time, harmonicity, and pitch continuity.2 If a set of frequency components starts at the exact same millisecond and follows a harmonic series, CASA algorithms group them into a single auditory stream.

While theoretically sound, CASA systems historically relied on hand-crafted rules that proved brittle in complex, reverberant environments. The "grouping" logic often fails when sources overlap heavily in time and frequency, a common occurrence in a kitchen or living room.2 However, the philosophy of CASA—masking T-F bins to separate sources—survives in modern deep learning approaches.

2.3 Independent Component Analysis (ICA)

ICA is a powerful statistical technique that separates a multivariate signal into additive subcomponents by assuming the subcomponents are non-Gaussian and statistically independent.[2] In multi-microphone setups, ICA is a standard solution. However, in the monaural domain (Single-Channel ICA), the problem requires projecting the single signal into a higher-dimensional space (e.g., via delay embedding or spectral decomposition) to create "virtual" channels. This process is computationally intensive and often unstable for real-world audio where sources are not strictly independent (e.g., a person reacting to a TV sound).[3]

2.4 The Deep Learning Revolution: T-F Masking vs. Time-Domain

The contemporary field is dominated by Deep Neural Networks (DNNs). Initially, these networks mimicked CASA by predicting Time-Frequency Masks. A network would take a noisy spectrogram as input and output a "mask" matrix (values 0 to 1) that, when multiplied by the noisy spectrogram, would reveal the clean source.11

However, a critical limitation emerged: the Phase Problem. The Short-Time Fourier Transform (STFT) produces a complex-valued spectrogram (Magnitude and Phase). Neural networks typically predict only the Magnitude mask. To reconstruct the audio, the system must reuse the noisy phase of the original input. This mismatched phase places a theoretical ceiling on the quality of separation.12

This limitation led to the development of Time-Domain approaches (like Conv-TasNet), which bypass the STFT entirely, learning to process the raw waveform directly. This represents the current state-of-the-art for high-fidelity separation.13

2.5 Selection of the Top Five Methodologies

Based on this survey, we identify five methodologies that constitute a complete, robust pipeline for a household voice assistant. We select **Supervised NMF** (Methodology I) as a scientifically robust baseline for stationary noise handling. We select **Conv-TasNet** (Methodology II) to handle the complex, non-stationary separation of arbitrary household sounds (Universal Sound Separation). We select **CRNN** (Methodology III) for efficient, real-time Sound Event Detection. We select **AST/BEATs** (Methodology IV) for high-accuracy classification using attention mechanisms. Finally, we select the **Mean Teacher** framework (Methodology V) to address the critical shortage of labeled domestic audio data.

3. Methodology I: Supervised Non-Negative Matrix Factorization (NMF)

Category: Statistical Signal Processing / Dictionary Learning
Role: Source Separation & Denoising
Key Insight: Decomposing the spectrogram into additive parts using pre-learned spectral dictionaries.
Non-Negative Matrix Factorization (NMF) stands as a foundational technique in audio signal processing. While deep learning offers higher performance ceilings, NMF provides interpretability, mathematical rigor, and the ability to operate effectively with significantly less training data than massive neural networks.[2] For a final project, NMF serves as an excellent "white box" methodology to contrast against "black box" deep learning approaches.

3.1 Theoretical Formulation

NMF operates on the magnitude spectrogram of the audio signal. Let $\mathbf{V} \in \mathbb{R}_{\ge 0}^{F \times T}$ be the magnitude spectrogram of the monaural mixture, where F is the number of frequency bins and T is the number of time frames. The objective of NMF is to approximate $\mathbf{V}$ as the product of two non-negative matrices:

$$\mathbf{V} \approx \mathbf{W}\mathbf{H}$$

Here, $\mathbf{W} \in \mathbb{R}_{\ge 0}^{F \times K}$ is the **basis matrix** (or dictionary), containing K spectral basis vectors. Each column of $\mathbf{W}$ represents a fundamental spectral shape (e.g., the harmonic stack of a single piano note, or the broadband shape of a snare hit).[14] $\mathbf{H} \in \mathbb{R}_{\ge 0}^{K \times T}$ is the **activation matrix**, describing how strongly each basis vector is active at each time step.[15]

The factorization is achieved by minimizing a cost function (divergence) $D(\mathbf{V} | \mathbf{W}\mathbf{H})$ subject to non-negativity constraints $\mathbf{W}, \mathbf{H} \ge 0$. Common divergences used in audio include:

- **Frobenius Norm:** Assumes Gaussian noise; minimizes Euclidean distance.
- **Kullback-Leibler (KL) Divergence:** More appropriate for Poisson-distributed data; often yields better results for audio magnitude spectra.[15]
- **Itakura-Saito (IS) Divergence:** Scale-invariant; highly suitable for audio as it captures perceptual similarities better than Euclidean distance.[15]

3.2 Supervised Separation Strategy

In a "Blind" NMF context, the algorithm learns $\mathbf{W}$ and $\mathbf{H}$ simultaneously from the mixture, which leads to the "permutation problem" (not knowing which basis belongs to which source). For a voice assistant targeting known classes (e.g., "Speech" vs. "Vacuum" vs. "Dog"), **Supervised NMF** is the superior approach.[16]

The workflow proceeds in two distinct stages:

1. **Dictionary Learning (Training):** We collect isolated clean recordings of the target sources. We construct a dictionary $\mathbf{W}_{speech}$ by running NMF on clean speech data, and a separate dictionary $\mathbf{W}_{noise}$ (or multiple dictionaries for different noise types) from noise data. In this stage, both $\mathbf{W}$ and $\mathbf{H}$ are updated freely to learn the spectral signatures.
2. **Source Separation (Inference):** Given a new noisy mixture $\mathbf{V}_{mix}$, we fix the composite dictionary $\mathbf{W}_{total} =$. We then run the NMF update rules *only* for the activation matrix $\mathbf{H}$. The algorithm finds the optimal linear combination of the pre-learned speech and noise bases to reconstruct the mixture.

Once the global activation matrix $\mathbf{H}_{total}$ is estimated, it is split into $\mathbf{H}_{speech}$ and $\mathbf{H}_{noise}$ corresponding to the indices of their respective bases. The separated spectrograms are reconstructed as:

$$\hat{\mathbf{V}}_{speech} = \mathbf{W}_{speech} \mathbf{H}_{speech}$$

$$\hat{\mathbf{V}}_{noise} = \mathbf{W}_{noise} \mathbf{H}_{noise}$$

3.3 Advanced Variation: NMF with Basis Deformation

A critical weakness of standard Supervised NMF is the rigidity of the dictionary. A vacuum cleaner in the user's home may sound slightly different than the one in the training set due to room acoustics (reverberation) or manufacturing differences. This mismatch causes the clean bases to fail in fitting the observed data, leading to leakage (noise appearing in the speech estimate).[16]

To address this, **NMF with Basis Deformation** introduces a deformation matrix/tensor that allows the pre-trained bases to warp spectrally during the inference phase. The optimization problem becomes finding an activation $\mathbf{H}$ and a small deformation $\Delta \mathbf{W}$ such that $\mathbf{V} \approx (\mathbf{W}_{fixed} + \Delta \mathbf{W})\mathbf{H}$, with a heavy penalty on $\Delta \mathbf{W}$ to prevent it from morphing into the wrong source class. This approach significantly enhances robustness in dynamic domestic environments.[16]

3.4 Python Implementation and Feasibility

NMF is highly accessible in the Python ecosystem. The sklearn.decomposition.NMF class provides a robust implementation of the alternating least squares and coordinate descent solvers.[14] For audio-specific workflows, librosa.decompose.decompose wraps the scikit-learn implementation, offering easy integration with STFT and inverse-STFT pipelines.[18]

Implementation Artifacts:

- **Input:** STFT Magnitude Spectrogram (calculated via librosa.stft).
- **Hyperparameters:** Number of components $K$ (crucial tuning parameter; too few misses spectral detail, too many overfits noise), Divergence type (Beta-divergence).
- **Latency:** Iterative inference can be slow. For real-time applications, "Online NMF" or fixing the number of iterations (e.g., to 50) is necessary.

Pros of Supervised NMF	**Cons of Supervised NMF**
Data Efficiency: Can be trained effectively on small datasets (minutes of audio).	**Phase Decoupling:** Reconstructs magnitude only; requires using noisy phase for iSTFT, limiting quality.[12]
Explainability: Basis vectors can be visualized as "spectral atoms," making debugging intuitive.	**Computational Cost:** Inference requires iterative optimization, unlike the single forward pass of a neural network.
Adaptability: Basis deformation allows handling of unseen variations in known noise classes.	**Linearity Assumption:** Assumes strictly additive magnitudes, which is physically inexact for overlapping waves.

---

4. Methodology II: End-to-End Time-Domain Separation (Conv-TasNet)

Category: Deep Learning / Universal Sound Separation (USS)
Role: High-Fidelity Source Separation
Key Insight: replacing the STFT with a learnable encoder to separate arbitrary sounds in the time domain.
While NMF is a robust baseline, the "Phase Problem" described in Section 2.4 limits its performance for high-fidelity applications. To achieve state-of-the-art results in separating a

voice assistant's target audio from complex household background noise, the **Conv-TasNet (Convolutional Time-domain Audio Separation Network)** architecture is the current gold standard.[12] Crucially, this methodology supports **Universal Sound Separation (USS)**, extending beyond speech-only tasks to separating arbitrary distinct sound classes (e.g., a siren from a dog bark).[20]

4.1 Architecture: Bypassing the Fourier Transform

Conv-TasNet represents a paradigm shift from "Time-Frequency Masking" to "Time-Domain Masking." It operates directly on the raw waveform $x(t)$. The architecture is composed of three functional blocks:

1. The Learnable Encoder:
   Instead of decomposing the signal into sine waves (STFT), the Encoder applies a 1D convolution (acting as a filterbank) to the overlapping segments of the waveform.

   $$w_k = \text{ReLU}(x_k * \mathbf{U})$$

   Here, $\mathbf{U}$ is the learnable basis matrix. The network learns the optimal basis functions for the specific task. For domestic noise, these learned bases often resemble "wavelets" or transient filters that are far better at representing impulsive sounds (knocks, claps) than standard Fourier sinusoids.21
   - **Research Insight:** For Universal Sound Separation involving non-speech sounds, research indicates that **shorter window sizes** (approx. 2-3ms) in the encoder significantly outperform the longer windows (20-30ms) used for speech. This allows the model to resolve the fine temporal structure of transient household events.[20]
2. The Separation Network (TCN):
   The core separation engine uses a Temporal Convolutional Network (TCN). This network estimates a mask m_k for each source in the latent space. The TCN is built from stacked blocks of 1D Dilated Depth-wise Separable Convolutions.
   - **Dilation:** The dilation factor increases exponentially ($1, 2, 4, 8, \dots$) in successive layers. This expands the **receptive field** of the network to thousands of samples without exploding the parameter count.[12]
   - **Significance for Home Audio:** A large receptive field is vital. To distinguish a rhythmic alarm from a random clatter, the network needs to "see" several seconds of context. The dilation allows the network to model long-term temporal dependencies (rhythm, melody) alongside short-term features (timbre).
3. The Learnable Decoder:
   A Transposed 1D Convolution takes the masked latent representation ($d_k = w_k \odot m_k$) and reconstructs the time-domain waveform $\hat{s}(t)$. This step implicitly reconstructs the phase, solving the major bottleneck of NMF.13

4.2 Universal Sound Separation (USS) and the FUSS Dataset

Most source separation literature focuses on the "Cocktail Party" (speech vs. speech). However, a home assistant faces "Universal" separation (speech vs. vacuum vs. TV vs. dog). The FUSS (Free Universal Sound Separation) dataset was created specifically to train models like Conv-TasNet for this broader domain.22

The FUSS dataset mixes arbitrary sounds from the FSD50K corpus (domestic and environmental sounds). Training Conv-TasNet on FUSS enables the model to generalize to the "class-agnostic" separation of sources based on their structural distinctiveness rather than just pitch.11 This is critical for the "household noises" requirement of the user's project.

4.3 Training Objective: SI-SNR

Conv-TasNet is trained to maximize Scale-Invariant Signal-to-Noise Ratio (SI-SNR). Unlike Mean Squared Error (MSE), SI-SNR measures the alignment of the estimated signal with the target signal while ignoring simple scaling (volume) differences.

$$\text{SI-SNR} = 10 \log_{10} \frac{\|s_{target}\|^2}{\|e_{noise}\|^2}$$

This metric is crucial for voice assistants, as the distance of the user from the microphone (and thus the volume) varies constantly. The model learns to recover the shape of the waveform regardless of its amplitude.13

4.4 Python Implementation with Asteroid

The **Asteroid** toolkit is the industry-standard Python library for implementing Conv-TasNet.[6] It provides:

- Pre-trained models on FUSS and LibriMix.
- "Recipes" (scripts) that handle data preparation, training loops, and evaluation.
- Seamless integration with PyTorch.

Code Logic Example (Conceptual):

Python

```python
from asteroid.models import ConvTasNet
# Load a model pre-trained on Universal Sound Separation (FUSS)
model = ConvTasNet.from_pretrained("mpariente/ConvTasNet_FUSS_baseline")
# Separate a monaural household mixture
estimated_sources = model.separate(noisy_audio_waveform)
```

This simplicity masks the immense complexity of the underlying TCN, making it an accessible high-performance component for the final project.

---

5. Methodology III: Convolutional Recurrent Neural Networks (CRNN)

Category: Deep Learning / Sound Event Detection (SED)
Role: Detection and Temporal Localization
Key Insight: Combining spatial feature extraction (CNN) with temporal context modeling (RNN) for polyphonic event detection.
Once the audio stream is cleaned (or even if applied directly to the mixture), the system must understand *what* is happening. **Sound Event Detection (SED)** is the task of identifying the onset and offset timestamps of sound classes (e.g., "Vacuum Cleaner active from 00:05 to 00:15"). The **Convolutional Recurrent Neural Network (CRNN)** is the dominant baseline for this task, consistently anchoring the top submissions in DCASE Task 4.[5]

5.1 The Spatio-Temporal Duality of Sound

Domestic sound events exhibit characteristic signatures in both the frequency domain (timbre) and the time domain (evolution).

- **The Spatial Dimension (Frequency):** A bird chirp has a specific rising curve in the spectrogram; a plate smash has a broadband vertical spike. These are "visual" features in the Time-Frequency map.
- **The Temporal Dimension (Time):** A vacuum cleaner is continuous; a dog bark is repetitive; a footstep is periodic. Distinguishing these requires understanding the *sequence* of spectral frames.

The CRNN architecture hybridizes two neural networks to address this duality:

1. **CNN Front-End:** The input is a Log-Mel Spectrogram. Stacked 2D Convolutional layers act as feature extractors, identifying local spectral patterns.[25]
   - **Advanced Insight: Frequency-Dynamic Convolutions (FDY).** Standard CNNs use the same kernel across the entire image. In visual images, a cat is a cat whether it is at the top or bottom of the frame (translation invariance). In audio, a sound shifted vertically (in frequency) changes its physical meaning (e.g., a low rumble vs. a high whistle). **Frequency-Dynamic Convolutions** adapt the convolution kernels based on the frequency band, allowing the network to learn frequency-specific features, significantly improving detection accuracy for diverse domestic sounds.[5]

2. **RNN Back-End:** The feature maps from the CNN are flattened along the frequency axis and fed into **Bidirectional Gated Recurrent Units (Bi-GRUs)**. The Bi-GRU scans the sequence forward and backward, aggregating temporal context. This allows the model to smooth out predictions—knowing that a vacuum cleaner is unlikely to turn on and off every 10 milliseconds, but a door knock might.[26]

5.2 Attention Pooling for Weak Labels

A major challenge in domestic audio projects is data annotation. It is easy to label a 10-second clip as "contains dog bark" (Weak Label), but time-consuming to mark the exact start and end times (Strong Label).

To bridge this gap, modern CRNNs employ Attention Pooling. The RNN outputs a frame-level probability sequence (strong prediction). An attention layer then learns to weigh these frames to produce a single clip-level probability (weak prediction). During training, the model can be optimized using only weak labels, but the attention mechanism forces it to implicitly learn the time-localization of the event to minimize the loss.5

5.3 Implementation and DCASE Baseline

The DCASE 2023 Task 4 baseline provides a reference implementation of this architecture in Python/PyTorch.[25]

- **Input:** 64-band Mel-Spectrogram, 16kHz sample rate.
- **Augmentation:** Crucial for robustness. **Mixup** (blending two clips and their labels) and **FilterAugment** (applying random frequency filters) are standard practices to prevent the model from overfitting to specific recording conditions.[28]
- **Post-Processing:** The raw output of the CRNN is often noisy. **Median Filtering** is applied to the output probabilities to smooth the onsets and offsets, preventing fragmented detections.[5]

---

6. Methodology IV: Audio Spectrogram Transformers (AST) and BEATs

Category: State-of-the-Art Deep Learning / Semantic Classification
Role: High-Accuracy Audio Tagging
Key Insight: Applying Vision Transformer (ViT) architectures to audio spectrograms to capture global dependencies.

While CRNNs are efficient workhorses, the field of Deep Learning has largely shifted towards **Transformers**. For the specific task of **Audio Tagging** (classifying the content of a clip with high accuracy), **Audio Spectrogram Transformers (AST)** and their derivative **BEATs** represent the cutting edge, offering performance superior to CNN-based approaches on massive datasets like AudioSet.[29]

6.1 The Self-Attention Mechanism

Convolutional networks have a limited "receptive field"—they only look at local neighborhoods of pixels. Transformers, through the Self-Attention mechanism, allow every part of the input to relate to every other part simultaneously.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

In the context of a spectrogram:

1. **Patch Embedding:** The spectrogram is sliced into a grid of 16x16 patches. Each patch is projected into a linear embedding.
2. **Positional Encoding:** Since the transformer has no inherent sense of "left/right" or "up/down," learnable positional embeddings are added to specific time and frequency coordinates.
3. **Global Context:** The attention mechanism can correlate a frequency onset at the beginning of a clip with a decay tail at the end, capturing long-range dependencies that a CNN might miss. This is particularly powerful for recognizing complex "scenes" (e.g., "cooking") which are defined by a sequence of disparate events (chopping, sizzling, running water).[30]

6.2 BEATs: Bidirectional Encoder Representation from Audio Transformers

A specific, highly effective iteration of this methodology is **BEATs**. Inspired by BERT in NLP, BEATs introduces the concept of an **Acoustic Tokenizer**.[31]

- **Pre-training:** BEATs is pre-trained on massive unlabeled datasets. It learns to discretize continuous audio into a finite set of "acoustic tokens." The model is trained to predict masked (hidden) tokens, forcing it to learn a deep, semantic understanding of audio structure.
- **Transfer Learning:** For the user's "household noise" project, one does not train BEATs from scratch. Instead, one uses a pre-trained BEATs model (available via Hugging Face or Microsoft's UniLM) as a **Feature Extractor**. The audio is passed through the frozen BEATs backbone, yielding rich, high-level embeddings. A simple linear classifier or shallow MLP is then trained on top of these embeddings to classify the specific 5-10 household classes of interest.
- **Efficacy:** This approach was central to the winning systems in DCASE 2023, proving its dominance over pure CRNNs for classification tasks.[33]

6.3 Python Integration

The **Hugging Face Transformers** library and **SpeechBrain** toolkit provide Python interfaces for AST and BEATs.[32]

- **Pipeline:** Audio -> Log-Mel Spectrogram -> Patch Embeddings -> Transformer Encoder -> CLS Token -> Classifier Head.
- **Compute Trade-off:** Transformers are computationally heavier than CRNNs. For a voice assistant on edge hardware (like a Raspberry Pi), a CRNN might be preferred for latency. However, if processing on a server or a Jetson Nano, AST/BEATs offers the highest accuracy.[29]

7. Methodology V: Semi-Supervised Learning via Mean Teacher

Category: Learning Paradigm / Training Strategy
Role: Overcoming Data Scarcity
Key Insight: Leveraging unlabeled data by enforcing consistency between a "Student" and a "Teacher" model.
The final methodology is not a signal processing architecture per se, but a **training framework**. A pervasive problem in domestic audio projects is the lack of labeled data. A user can easily record 24 hours of their home's ambient noise, but annotating every event in that recording is impractical. The **Mean Teacher** framework is the industry-standard solution for this **Semi-Supervised Learning** problem in DCASE Task 4.[35]

7.1 The Student-Teacher Consistency Architecture

The framework employs two networks with identical architecture (e.g., two CRNNs): the **Student** and the **Teacher**.

1. **Student Model:** This is the standard network. Its weights $\theta$ are updated via backpropagation using standard supervised loss on the limited labeled data available.
2. Teacher Model: This network does not train via backpropagation. Its weights $\theta'$ are updated as an Exponential Moving Average (EMA) of the Student's weights:

   $$\theta'_t = \alpha \theta'_{t-1} + (1-\alpha)\theta_t$$

   where $\alpha$ is a decay parameter (typically 0.999). This makes the Teacher a temporal ensemble of the Student, producing more stable and accurate predictions.37

7.2 Leveraging Unlabeled Data

The core innovation is the **Consistency Loss**. When training on *unlabeled* data:

1. The system takes an unlabeled audio clip x.
2. It applies two different random augmentations (e.g., noise injection, time shift) to produce $x_{student}$ and $x_{teacher}$.
3. The Student predicts $P(x_{student})$ and the Teacher predicts $P(x_{teacher})$.
4. The model minimizes the distance (MSE or KL-Divergence) between these two predictions: $L_{consistency} = |

| P(x_{student}) - P(x_{teacher}) ||^2$.

7.3 Implication for Voice Assistants

This forces the Student to learn a representation that is robust to perturbations and consistent with the stable Teacher, effectively learning the "manifold" of the unlabeled data.[38] For a household voice assistant, this allows the system to improve itself using the raw, unannotated audio it captures daily. It learns the specific acoustic characteristics of the user's home (the "domain shift") without needing explicit labels, essentially "adapting" to the household environment over time.[36]

8. System Integration and Conclusion

8.1 Proposed Pipeline Architecture

To satisfy the requirement of a cohesive voice assistant system, these five methodologies should be integrated into a modular pipeline:

Stage	Methodology	Function	Python Tool
1. Input	**Monaural Audio**	Raw waveform capture.	PyAudio / sounddevice
2. Separation	**Conv-TasNet (Method II)**	Separates target foreground events from background noise (Universal Separation).	asteroid (PyTorch)
3. Feature	**Mel-Spectrogram**	Converts audio to optimal	librosa /

Extraction	**/ BEATs**	representation.	transformers
4. Detection	**CRNN (Method III)**	Detects Onset/Offset of events (SED) in real-time.	dcase_util / Custom PyTorch
5. Classification	**AST/BEATs (Method IV)**	Refines classification of detected events with high accuracy.	Hugging Face
6. Training Loop	**Mean Teacher (Method V)**	Continuous semi-supervised learning using daily unlabeled recordings.	Custom Training Script
Alternative	**Supervised NMF (Method I)**	Low-latency backup for removing stationary noise (fans/hum).	sklearn

8.2 Conclusion

The signal processing challenge of a single-channel household voice assistant is defined by underdetermination and non-stationarity. Classical methods like spectral subtraction are insufficient for the diverse, transient nature of domestic noise. This report has identified a robust, scientifically accurate suite of methodologies to address this.

Conv-TasNet solves the source separation problem by operating in the time domain, bypassing the phase limitations of the Fourier transform. **CRNNs** provide the necessary spatio-temporal modeling to detect event boundaries in real-time. **Transformers (AST/BEATs)** offer state-of-the-art classification accuracy by leveraging global attention mechanisms and large-scale pre-training. Finally, the **Mean Teacher** framework enables the system to learn from the abundant unlabeled audio of a home environment, ensuring the assistant adapts and improves over time. By implementing these five methodologies within the Python ecosystem, a developer can construct a system that not only hears but computationally "understands" the complex acoustic scene of a modern home.

Citations:
.1

Works cited

1. Audio-visual source separation with localization and individual control - PMC - NIH, accessed January 11, 2026, https://pmc.ncbi.nlm.nih.gov/articles/PMC12101657/
2. HARMONIC SINGLE CHANNEL SOURCE SEPARATION - Hajim School of Engineering & Applied Sciences, accessed January 11, 2026, https://hajim.rochester.edu/ece/sites/zduan/teaching/ece477/projects/2013/Ishaan_Rao_LitReview_Final.pdf
3. Signal separation - Wikipedia, accessed January 11, 2026, https://en.wikipedia.org/wiki/Signal_separation
4. Source Separation, Noise Suppression, and Dereverberation Technologies, accessed January 11, 2026, https://www.yamaha.com/en/tech-design/research/technologies/source-separation-supression/
5. Sound Event Detection in Domestic Environment Using Frequency-Dynamic Convolution and Local Attention - MDPI, accessed January 11, 2026, https://www.mdpi.com/2078-2489/14/10/534?
6. asteroid-team/asteroid: The PyTorch-based audio source separation toolkit for researchers - GitHub, accessed January 11, 2026, https://github.com/asteroid-team/asteroid
7. Tutorial — librosa 0.11.0 documentation, accessed January 11, 2026, https://librosa.org/doc/0.11.0/tutorial.html
8. (PDF) Environmental sound recognition: A survey - ResearchGate, accessed January 11, 2026, https://www.researchgate.net/publication/261208284_Environmental_sound_recognition_A_survey
9. One Microphone Source Separation - NIPS papers, accessed January 11, 2026, http://papers.neurips.cc/paper/1885-one-microphone-source-separation.pdf
10. "Blind Single Channel Sound Source Separation" by Mark Leddy - Arrow@TU Dublin, accessed January 11, 2026, https://arrow.tudublin.ie/engmas/40/
11. [PDF] Universal Sound Separation - Semantic Scholar, accessed January 11, 2026, https://www.semanticscholar.org/paper/Universal-Sound-Separation-Kavalerov-Wisdom/88de6a6cc24a2d1876fcec9cf91fa6c6386c16ec
12. Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation - arXiv, accessed January 11, 2026, https://arxiv.org/pdf/1809.07454

13. Conv-TasNet: End-to-End Time-Domain Audio Separation - Emergent Mind, accessed January 11, 2026, https://www.emergentmind.com/topics/conv-tasnet
14. NMF — scikit-learn 1.8.0 documentation, accessed January 11, 2026, https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html
15. Audio Source Separation Using Non-Negative Matrix Factorization (NMF) - Medium, accessed January 11, 2026, https://medium.com/@zahrahafida.benslimane/audio-source-separation-using-non-negative-matrix-factorization-nmf-a8b204490c7d
16. Audio source separation based on supervised nonnegative matrix factorization with basis deformation - Daichi Kitamura, accessed January 11, 2026, http://d-kitamura.net/demo-defNMF_en.html
17. Supervised non-negative matrix factorization for audio source separation, accessed January 11, 2026, https://collaborate.princeton.edu/en/publications/supervised-non-negative-matrix-factorization-for-audio-source-sep/
18. tutorial/Librosa tutorial.ipynb at master - GitHub, accessed January 11, 2026, https://github.com/librosa/tutorial/blob/master/Librosa%20tutorial.ipynb
19. nmf.ipynb - Nonnegative Matrix Factorization - Colab, accessed January 11, 2026, https://colab.research.google.com/github/HuwCheston/musicinformationretrieval.com/blob/gh-pages/mirdotcom/content/10_decomposition/nmf.ipynb
20. Universal Sound Separation - Mitsubishi Electric Research Laboratories, accessed January 11, 2026, https://www.merl.com/publications/docs/TR2019-123.pdf
21. Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation | Request PDF - ResearchGate, accessed January 11, 2026, https://www.researchgate.net/publication/332917627_Conv-TasNet_Surpassing_Ideal_Time-Frequency_Magnitude_Masking_for_Speech_Separation
22. Free Universal Sound Separation | Google Open Source Blog, accessed January 11, 2026, https://opensource.googleblog.com/2020/04/free-universal-sound-separation.html
23. Free Universal Sound Separation Dataset - Zenodo, accessed January 11, 2026, https://zenodo.org/records/3694384
24. Asteroid: the PyTorch-based audio source separation toolkit for researchers - ar5iv - arXiv, accessed January 11, 2026, https://ar5iv.labs.arxiv.org/html/2005.04132
25. marmoi/dcase2023_task4b_baseline: Baseline code for DCASE 2023 task 4 B - GitHub, accessed January 11, 2026, https://github.com/marmoi/dcase2023_task4b_baseline
26. arXiv:2402.02781v1 [cs.SD] 5 Feb 2024, accessed January 11, 2026, https://arxiv.org/pdf/2402.02781
27. Sound Event Detection in Domestic Environment Using Frequency-Dynamic Convolution and Local Attention - ResearchGate, accessed January 11, 2026,

https://www.researchgate.net/publication/374411937_Sound_Event_Detection_in_Domestic_Environment_Using_Frequency-Dynamic_Convolution_and_Local_Attention

28. semi-supervised sound event detection based on mean teacher with selective kernel multiscale - DCASE, accessed January 11, 2026, https://dcase.community/documents/challenge2022/technical_reports/DCASE2022_Gan_14_t4.pdf
29. Audio Spectrogram Transformers Beyond the Lab | by Maciej Adamiak - Medium, accessed January 11, 2026, https://medium.com/@maciej.adamiak/audio-spectrogram-transformers-beyond-the-lab-1be80a0b1ce4
30. Trainingless Adaptation of Pretrained Models for Environmental Sound Classification - arXiv, accessed January 11, 2026, https://arxiv.org/html/2412.17212v1
31. BEATs: Audio Pre-Training with Acoustic Tokenizers - Kaggle, accessed January 11, 2026, https://www.kaggle.com/datasets/hubfor/microsoft-beats-model
32. speechbrain.lobes.models.beats module - Read the Docs, accessed January 11, 2026, https://speechbrain.readthedocs.io/en/latest/API/speechbrain.lobes.models.beats.html
33. dcase 2023 challenge task4 technical report, accessed January 11, 2026, https://dcase.community/documents/challenge2023/technical_reports/DCASE2023_Liu_81_t4b.pdf
34. Audio Classification using Transformers - GeeksforGeeks, accessed January 11, 2026, https://www.geeksforgeeks.org/nlp/audio-classification-using-transformers/
35. MEAN TEACHER MODEL BASED ON CMRANN NETWORK FOR SOUND EVENT DETECTION Technical Report Qian Yang Jing Xia Jinjia Wang College of - DCASE, accessed January 11, 2026, https://dcase.community/documents/challenge2019/technical_reports/DCASE2019_Yang_18.pdf
36. An Effective Mutual Mean Teaching Based Domain Adaptation Method for Sound Event Detection - ISCA Archive, accessed January 11, 2026, https://www.isca-archive.org/interspeech_2021/zheng21_interspeech.pdf
37. CuriousAI/mean-teacher: A state-of-the-art semi-supervised method for image recognition - GitHub, accessed January 11, 2026, https://github.com/CuriousAI/mean-teacher
38. SELF-TRAINING WITH NOISY STUDENT MODEL AND SEMI-SUPERVISED LOSS FUNCTION FOR DCASE 2021 CHALLENGE TASK 4 Technical Report Nam Ky, accessed January 11, 2026, https://dcase.community/documents/challenge2021/technical_reports/DCASE2021_Kim_23_t4.pdf
39. unilm/beats/README.md at master - GitHub, accessed January 11, 2026,

https://github.com/microsoft/unilm/blob/master/beats/README.md