Applying Transformers and Attention for Speech Denoising

Published August 3, 2025 40 min read

Transformers for Speech Denoising: Leveraging Attention to Remove Noise

Introduction

Cleaning human voice recordings from background noise—known as *speech enhancement* or *denoising*—is a long-standing challenge in audio signal processing. Traditional methods (e.g. spectral subtraction, Wiener filtering) rely on statistical models of noise and speech, but they often fail to eliminate highly non-stationary or unpredictable noises. In recent years, deep learning approaches have revolutionized this field by learning to map noisy audio to clean speech. Recurrent neural networks (RNNs) like LSTMs and gated units were early favorites for modeling speech sequences, and convolutional neural networks (CNNs), including temporal convolutional networks (TCNs), have shown impressive performance by capturing local patterns with large receptive fields. However, the transformer architecture—originally developed for NLP—has emerged as a

powerful alternative due to its *attention mechanism* that can model long-range dependencies in parallel. This report explores how transformers and attention can be applied to speech denoising, covering the transformer basics, audio feature representations, current research on transformer-based speech enhancement, comparisons with other methods, technical considerations (e.g. computational cost), and future directions for this promising approach.

Transformer Architecture and the Attention Mechanism

Transformers are sequence-to-sequence models that dispense with recurrence and instead use self-attention to process input sequences. A standard transformer consists of an encoder (and optionally a decoder, if doing sequence generation), each built from stacked layers of multi-head self-attention and feed-forward networks with residual connections and layer normalization. **Figure 1** illustrates the transformer's encoder-decoder architecture, where each encoder layer (left, gold) has a multi-head self-attention sublayer and a feed-forward sublayer, and each decoder layer (right, green) adds a multi-head *cross-attention* to incorporate encoder outputs in tasks like translation. In speech enhancement, often only an encoder (or a "masking" network) is used to transform a noisy input sequence into a clean output sequence, so the decoder part may be simplified or absent (especially in purely *feed-forward* enhancement models).

! https://commons.wikimedia.org/wiki/File:Transformer,_full_architecture.png

Figure 1: Transformer architecture overview (encoder on left, decoder on right). In speech enhancement tasks, the transformer is typically used in an encoder/masking network to map noisy audio features to cleaned output, leveraging multi-head self-attention to model relationships across the sequence.

The core *attention mechanism* allows the model to weigh different parts of the input sequence in relation to each other. In **self-attention**, each time-step (or "token") in the input computes a weighted sum of all time-steps, using weights (attention scores) that represent the similarity or relevance between elements. Multi-head self-attention does this in parallel across multiple "heads," so that different heads can focus on different patterns or frequency bands. This capability to consider *all pairwise interactions* in a sequence is crucial: it enables the model to capture global context and long-term structure in speech, which is difficult for RNNs or CNNs to achieve without very large context windows. For example, a transformer can learn that a segment of audio containing voiced speech (with harmonic structure) should attend to other segments with similar harmonic patterns, effectively learning to *amplify the signal (speech) and diminish less important tokens (noise)*. In other words, the attention mechanism can help the model focus on the informative parts of the audio (the human voice) while giving less weight to uncorrelated noise. This property is intuitively useful for separating speech from background noise – the speech tends to have coherent temporal-frequency patterns (like repeating harmonics or formant trajectories) that the model can lock onto, whereas noise may appear as unrelated or short-lived events that the attention weights can learn to suppress.

Transformers also bring practical advantages: since they have no recurrent states, their computations over a sequence can be fully parallelized. This leads to faster training compared to RNNs, which must process one time-step after another. In the context of audio, parallelization means we can feed an entire audio sequence (or a window of it) into the model and process all frames concurrently, which is beneficial for offline processing and training on GPUs. However, a well-known drawback is that self-attention has *quadratic* complexity in the sequence length. Audio signals are long sequences (a few seconds of audio may consist of thousands of time frames or samples), so naive application of transformers can be computationally expensive. We will discuss later how researchers address this issue (e.g. by clever chunking of sequences, sparse attention, or other efficiency tricks). Additionally, because transformers ignore sequence order unless explicitly encoded, one must add **positional encodings** (fixed sinusoids or learnable vectors) to audio sequences to inform the model about the time ordering. Designing effective positional encodings for audio is non-trivial; early works found that naive approaches limited performance, and recent studies showed that *relative* positional embeddings work better for generalizing to longer audio than absolute encodings.

In summary, the transformer architecture (through multi-head attention) offers a powerful mechanism to model both local and long-range relationships in audio. It can, in principle, learn to *separate signal from noise by attending to the relevant speech components across time-frequency and ignoring distracting noise*. This has motivated its adoption in speech denoising and related tasks, as we explore in the following sections.

Audio Preprocessing and Embeddings for Transformers

Raw audio is a one-dimensional waveform signal, but transformers expect a sequence of feature vectors (tokens). Thus, a crucial step is to convert audio into a suitable *embedding sequence*. There are several strategies for this, ranging from traditional signal processing features to learned neural representations:

- **Spectrograms and Time-Frequency Features:** A common approach is to compute a time-frequency representation via the Short-Time Fourier Transform (STFT). The STFT converts the waveform into a sequence of complex spectra (magnitude and phase) at regular time frames. Often, only the magnitude (or power) spectrogram is used as the input feature for a denoising model, sometimes on a perceptual scale (e.g. Mel spectrogram). The spectrogram can be viewed as an "image" (time vs frequency) and provides a compact representation that separates different frequency components of speech and noise. Many deep learning models operate on the *magnitude spectrum* and then apply the inverse STFT with the noisy phase or a reconstructed phase to get back to waveform. Using spectrogram magnitude as input has the advantage of reducing sequence length (since frames might be, say, 20 ms long) and focusing on spectral content. However, one must decide on an appropriate frame length: shorter frames give higher temporal resolution but longer sequences, whereas longer frames shorten the sequence but blur temporal details. Research shows a trade-off here. For example, one study replaced a learned 2 ms encoder representation with a *32 ms STFT magnitude* and found it *drastically reduced the sequence length (and thus attention cost) by ~8× without hurting enhancement performance*. Longer frames make the magnitude spectrogram more informative (phase becomes relatively less critical at 32 ms window sizes),

allowing the transformer to work with fewer time steps. Spectral features like log-power spectrum (LPS) or Mel-frequency spectra can also be used as inputs. Some models output a mask to apply on the noisy spectrogram (e.g. an ideal ratio mask estimation) to filter out noise.

- **MFCCs and Traditional Coefficients:** Mel-frequency cepstral coefficients (MFCCs) are a classic low-dimensional representation of audio, widely used in older speech recognition systems. MFCCs apply a cosine transform to the log-Mel spectrum to decorrelate features and approximate how humans perceive sound. They yield ~13 coefficients per frame, which is much smaller than, say, a 256-bin STFT magnitude. In theory, one could feed MFCCs into a transformer as the sequence features. However, MFCCs discard a lot of detail (phase information and even fine spectral variation), which might be important for high-quality speech reconstruction. Modern deep learning approaches thus tend to prefer raw spectrogram or learned features over MFCCs for enhancement, since denoising requires capturing subtle differences between clean speech and noise. MFCCs could still be useful as an *auxiliary feature* or for lightweight models, but by themselves they might constrain the achievable quality.

- **Learned Audio Embeddings (e.g. wav2vec, HuBERT):** Another powerful approach is to leverage learned representations from self-supervised models. *Wav2Vec 2.0* is a notable example where a CNN encoder maps raw waveform into a sequence of latent vectors, which are then contextualized by a transformer trained on a self-supervised objective (masking and predicting quantized units). The result is a rich embedding for speech that has been pre-trained on massive unlabeled data, capturing phonetic and speaker information. These embeddings (or the transformer encoder from wav2vec) can be adapted for speech enhancement. For instance, one could feed the wav2vec encoder's output sequence into a smaller transformer or decoder that focuses on denoising. The benefit is that the front-end features are already robust to many variations (since the model learned to handle diverse audio). Indeed, researchers have explored using pretrained models (wav2vec, HuBERT, etc.) as front-ends for speech enhancement with some fine-tuning, finding that they can improve performance especially in low-data regimes. Another related line is *vector-quantized embeddings*: projects like SoundStream/EnCodec produce discrete tokens for audio; a transformer could conceivably learn to correct noisy tokens to clean tokens in that space. However, most current enhancement models stick to continuous features rather than discrete tokens.

- **Raw Waveform with Learned Filters:** It is also possible to feed raw waveforms directly to a transformer by treating each sample as a sequence element, but this is impractical for typical audio rates (16 kHz audio means 16,000 tokens per second!). Instead, some end-to-end models learn a *waveform encoder*—for example, a 1-D convolution that splits the waveform into frames or chunks of a few samples. A famous example is Conv-TasNet's encoder which uses a learnt filterbank (e.g. 512 filters at 2 ms) to convert waveform to a higher-level representation before applying a separation/denoising network. One could replace Conv-TasNet's RNN/TCN masker with a transformer: indeed, the original SepFormer did exactly this, operating in a learned 2 ms waveform basis. The drawback, as mentioned, is the extremely long sequence for a long audio signal. In practice, state-of-the-art time-domain models use techniques like segmenting the sequence and dual-path processing (described below) to make this feasible.

In summary, preparing audio for a transformer typically involves extracting a sequence of feature vectors (whether spectrogram bins, MFCCs, or learned latent features). Spectrogram-based embeddings are most common in speech enhancement and allow the transformer to exploit the structured frequency content of speech. Using appropriate preprocessing can greatly affect the efficiency and success of the model. For example, using longer-frame STFT features *reduced the transformer's computation* without degrading quality, and focusing on magnitude (with noisy phase reused) simplifies the learning task. As we will see next, many transformer-based architectures adopt an *encoder-mask-decoder* paradigm: an encoder (could be a CNN or STFT) produces the embedding sequence, a transformer (or variation) processes it to estimate a mask or cleaned embedding, and a decoder (inverse transform or neural decoder) reconstructs the waveform.

Transformer-Based Speech Enhancement: Research and Applications

Early Transformer Models for Denoising: When first applied to speech enhancement, vanilla transformers did not immediately outperform RNNs. Speech has different structure than text, and naive use of a transformer can struggle with local continuity or require careful position encoding. One of the pioneering works was **T-GSA** by Kim *et al.* (ICASSP 2020), which introduced a *Gaussian-weighted self-attention* for speech enhancement. In T-GSA, the attention weights are multiplied by a Gaussian function of the time difference between frames, so that distant frames' influence is attenuated. This biases the transformer to focus more on near-context (which is important for speech coherence) while still allowing some long-range attention. The authors reported significantly improved enhancement performance with T-GSA compared to a standard transformer and to LSTM baselines. Essentially, by tailoring the attention mechanism to favor local context (since speech content a few frames apart is usually more correlated than speech one second apart), they made the transformer more effective at separating speech from noise. This idea of *contextual biasing* illustrates how domain knowledge can be infused into attention for audio tasks.

Another seminal architecture was the **Dual-Path Network** concept, which was originally developed with RNNs and later with transformers. **Dual-Path RNN (DPRNN)** by Luo *et al.* (2020) segmented the input sequence into chunks, processed intra-chunk and inter-chunk information with stacked RNN blocks, and thereby managed to model very long sequences efficiently (Source: isca-archive.org). Building on that, **Dual-Path Transformer Network (DPTNet)** was proposed by Chen *et al.* (Interspeech 2020), combining self-attention with RNNs: essentially they used a transformer for local (intra-chunk) modeling and an RNN for global (inter-chunk) modeling (Source: isca-archive.org). Shortly after, **SepFormer** ("Separator Transformer") by Subakan *et al.* (ICASSP 2021) took this to the next level: it is an RNN-free, fully transformer-based separation model. SepFormer uses dual-path processing but replaces both RNN stages with transformers – one transformer operates within each chunk of frames to capture fine local speech structure, and another operates across chunks (on the outputs of the first stage) to capture global dependencies between distant parts of the signal. This multi-scale approach gave SepFormer remarkable capability to learn both short-term and long-term patterns in speech. On the WSJ0-2mix benchmark (a standard two-speaker mixture separation task),

SepFormer achieved **state-of-the-art results** with an SI-SNR improvement of ~22.3 dB, outperforming prior CNN and RNN models by a significant margin. Notably, SepFormer's transformer-based mask network was *parallelizable* and could even be downsampled (processing every 8th time-step) to speed up inference with only minor performance loss. This proved that attention mechanisms can indeed rival or exceed traditional architectures in speech source separation and by extension in denoising (which can be seen as a special case with one source being noise).

Architectural Innovations: Many researchers have proposed enhancements to the basic transformer to better suit speech tasks. One direction is addressing the high computation and memory cost. **Axial attention** or **time-frequency attention splitting** was suggested to reduce complexity when using 2D spectrogram inputs. Li *et al.* (2021) introduced a *U-shaped Transformer* for speech enhancement with **Frequency-Band Aware Attention**. They noticed that low-frequency bands carry more speech energy and are more critical to intelligibility than high frequencies in many noise conditions. So they devised separate multi-head attention mechanisms for the low-frequency band vs. high-frequency band, effectively splitting the input features and processing them in parallel (LFA and HFA for low/high frequency attention). Their model also had a U-Net style encoder–decoder with skip connections (like the popular U-Net in image segmentation, but here with transformer layers in between). This U-Transformer achieved better speech estimation accuracy on several datasets, confirming that integrating spectral domain knowledge and multi-scale architecture can boost performance.

Another line of work augments transformers with CNNs. The **Conformer** architecture (originally from Google for ASR) combines convolution layers with multi-head self-attention, aiming to capture local feature patterns along with global relations. In 2021, Kim and Seo proposed **SE-Conformer**, a time-domain speech enhancement model that uses a convolutional encoder/decoder and Conformer blocks in between (Source: isca-archive.org)(Source: isca-archive.org). The convolutional layers handle frame-level transformations and local context, while the Conformer's attention heads model long-term dependencies. On the VoiceBank-DEMAND noisy speech benchmark, SE-Conformer outperformed competitive baselines in objective quality (e.g. PESQ, STOI) (Source: isca-archive.org). This success underlines a theme: *hybrid models* can leverage the strengths of both CNNs (local smoothing, translation invariance, efficiency) and transformers (global context modeling). In fact, a study of a Tiny-SepFormer model showed that within each chunk, the attention map tended to concentrate around the diagonal (i.e. mostly local frame-to-frame attention), suggesting that much of the local structure could be handled by a convolution, leaving the transformer to focus on chunk-level global interactions. This insight was used to design a **Tiny-SepFormer** that replaces one of the dual-path transformer legs with a lightweight convolutional network, significantly reducing parameters while maintaining performance. Such results demonstrate a practical way to shrink transformer models: replace or assist portions of the attention computation with convolutions for locality, and possibly share parameters across layers.

Sparse and Efficient Attention: Given the quadratic cost of standard attention, researchers have explored sparsifying the attention patterns for speech. *Ripple Attention* (Zhang *et al.*, ICASSP 2023) is one example of a sparse self-attention tailored to speech enhancement. By limiting each time frame to attend strongly only to a

subset of other frames (perhaps those within a certain window or following a pattern of "ripple" expanding through time), they achieved a good balance of accuracy and efficiency. There are also studies on *dynamic attention span* for real-time speech enhancement, where the model adaptively limits how far back it needs to attend depending on the input (shortening context during simple sections, expanding during complex ones). Additionally, **relative positional encodings** and other techniques borrowed from long-sequence NLP (like Transformers with linear complexity approximations) have been tested to allow transformers to extrapolate to longer utterances than seen in training. A 2024 study found that using relative position embeddings (instead of fixed or learnable absolute positions) substantially improved a transformer's ability to generalize from short training clips to longer speech during testing. This is crucial for real-world use, since we often train on short clips for efficiency but need the model to handle long audio streams.

Real-World Applications and Systems: The ultimate test for these models is deployment in real applications like hearing aids, teleconferencing (Zoom/Teams noise suppression), or voice assistants in noisy environments. As of 2025, many real-time noise suppression systems still use more compact models (often recurrent or convolutional) due to strict latency and compute requirements. However, transformer-based models are beginning to make inroads. For instance, the Microsoft Deep Noise Suppression (DNS) Challenge—an open competition for denoising models—has seen entries exploring transformers. Researchers have demonstrated *causal transformers* that operate frame-by-frame, using masked self-attention to prevent peeking into future frames so as to meet real-time processing needs. In such configurations, at inference the transformer can be unrolled in a streaming fashion, processing one frame at a time with attention only over past (and possibly a limited lookahead) context. One study explicitly notes: *"to address real-time noise suppression, transformers are implemented in a causal configuration, where one frame cannot attend to future frames."*. This shows that with careful design, transformers can be used in low-latency systems, though often at the expense of some performance compared to non-causal, full-context models.

There are also emerging *large-scale* and *open-source* projects applying transformers to speech enhancement. **SpeechBrain**, an open-source speech toolkit, implemented the SepFormer and even extended it to noise enhancement tasks (e.g., training on the DNS Challenge dataset). Users can directly apply a pretrained SepFormer model to their noisy audio to separate speech and noise. The reported results on DNS challenge data are promising, with a transformer-based model achieving high scores on DNSMOS (a MOS predictor for denoised speech) – for example, an overall quality (OVRL) score around 2.44 on a scale up to 5 for a model trained on DNS 2022 data. While not perfect, this is comparable to many real-time methods. Meanwhile, companies and researchers are experimenting with *generative transformer models* for speech restoration. One such example is **VoiceRestore (2024)**, which uses a 301-million-parameter transformer in a *flow-matching* generative framework to handle not just noise, but also reverberation, distortions, and dropouts. VoiceRestore takes a different approach: rather than predicting a mask or cleaned spectrogram directly, it iteratively "restores" a degraded audio through a generative process guided by a transformer (somewhat analogous to diffusion models in vision). This model is too heavy for real-time use, but it points toward a future where large transformers trained on vast data can perform *universal speech restoration*, effectively bringing studio-level quality to severely degraded recordings.

In summary, transformer-based speech enhancement has quickly evolved from initial feasibility studies to state-of-the-art research models. They have been successfully applied in monaural speech denoising, speech separation (multiple sources), and even speech *restoration* tasks that handle noise alongside other distortions. Open-source implementations (e.g. SepFormer in SpeechBrain) and continued improvements in efficiency are bringing these models closer to practical use. Next, we compare how the transformer approach differs from and improves upon previous architectures.

Comparison to RNN, CNN, and Traditional Approaches

Different neural architectures have their own strengths and weaknesses for speech denoising:

- **Recurrent Neural Networks (RNNs):** RNN-based models (including LSTMs and GRUs) were a natural fit for sequence problems and were widely used in early deep learning approaches to noise suppression. Their strength is the ability to maintain a memory of past frames, theoretically capturing arbitrarily long contexts. In practice, LSTMs can model long-range temporal dependencies better than simple frame-wise methods, and they produce good results on speech tasks. However, RNNs operate sequentially, which means they cannot take full advantage of parallel processing on modern hardware. This makes training and inference slower for long sequences. Moreover, RNNs tend to have difficulty remembering very long-term information due to finite memory and issues like vanishing gradients (LSTMs mitigate this but not entirely). As a result, an RNN might struggle if the noise profile or speech content requires very long context to distinguish (for example, a noise that sounds like speech could confuse it without broader context). Transformers, by contrast, have no persistent hidden state that must be carried through time; they look at the entire sequence via attention (or entire chunk) at once, which gives *direct access* to long-range information and avoids the step-by-step bottleneck (Source: en.wikipedia.org#:~:text=Transformers%20have%20the%20advantage%20of,LLMs%29%20on%20large). Researchers have noted that transformers, by replacing recurrent loops with parallel attention, significantly speed up training and can achieve equal or better performance once properly configured. That said, RNNs remain more lightweight for a given sequence length – they scale linearly with sequence length vs. quadratic for attention – so for extremely long inputs or strict real-time constraints, a well-optimized RNN can still be advantageous in practice.

- **Convolutional Neural Networks (CNNs) and TCNs:** CNN-based architectures bring translation invariance and efficient parallel computation. 1-D CNNs (or 2-D CNNs on spectrograms) learn local filters that slide over time or time-frequency. Stacked with dilation (spacing out the filter applications), CNNs can capture fairly long contexts. For example, *temporal convolutional networks (TCNs)* use a hierarchy of dilated convolutions to achieve a large receptive field; they have demonstrated impressive performance in speech enhancement. A well-known model, *Conv-TasNet*, uses a convolutional encoder and decoder with a TCN mask estimator, and it surpassed many RNN methods in both quality and efficiency for speech separation. CNNs are highly parallelizable (convolutions over frames can be done in parallel for each layer) and typically have a fixed receptive field, which avoids some of the unpredictability of attention. However,

purely convolutional models may require a deep stack to cover long sequences and might have trouble modeling *global interactions* that are not local in time. For instance, a CNN might learn to suppress noise bursts based on local spectral patterns, but if the noise has a long-term structure (like a repeating siren) it might not "connect the dots" as well as an attention model that can link two distant occurrences. Transformers can naturally complement CNNs: as seen in Conformer and related models, the convolution handles fine local structure (e.g. capturing the exact waveform shape or formant transitions within ~50 ms), while the transformer attends over broader swaths of audio to understand high-level context (e.g. the presence of speech vs silence over a second, or recurring noise motifs). In terms of complexity, a single convolution layer is O(n) for n time steps (ignoring filter length), whereas self-attention is O(n^2). So for long audio, CNNs have an edge in efficiency and memory. In practice, many state-of-the-art enhancement models are hybrids: e.g., a convolutional front-end with a transformer core, or a dual-path network with both convolution and attention blocks.

- **Traditional Signal Processing Methods:** Before the deep learning era, noise reduction was done with methods like spectral subtraction, Wiener filtering, or subspace methods. These often assume a statistical model of the noise (e.g. noise is stationary and has a certain spectral shape that can be estimated during non-speech segments). While such methods are extremely fast (computationally trivial by today's standards) and require no training, they struggle with complex noises. Rapidly changing or highly non-stationary noises violate their assumptions, leading to speech distortion or residual noise. For example, spectral subtraction tends to create musical noise artifacts when the noise estimate is inaccurate. Deep learning methods (RNN, CNN, or transformer) *learn* from data and can, in theory, handle a much wider variety of noise conditions by recognizing patterns that traditional algorithms can't. The attention mechanism in transformers is particularly effective in scenarios where noise may overlap with speech in time-frequency: the model can still pick out the speech by its consistent structure across time, something a traditional linear filter cannot do without introducing distortion. That said, traditional methods have the advantage of *simplicity and low latency*. In scenarios like digital hearing aids, where computational resources and delay must be minimal, some form of classical noise suppression (or a very tiny neural net like a simple DNN or shallow RNN) might be used. But as edge hardware improves, even these domains are starting to adopt neural approaches for superior quality.

- **Performance and Metrics:** In terms of output quality, deep learning models (whether CNN, RNN, or transformer) have dramatically improved objective metrics like PESQ (Perceptual Evaluation of Speech Quality) and STOI (Short-Time Objective Intelligibility) compared to traditional methods. Transformers, in particular, have pushed the state of the art on many benchmarks. For instance, one transformer model achieved **PESQ ~3.0 and STOI 95%** on the VoiceBank+DEMAND noisy corpus, significantly higher than earlier CNN or RNN models on the same task (which were often in the 2.6–2.8 PESQ range) – this is cited by works such as SE-Conformer and others (Source: isca-archive.org). In source separation, the transformer-based SepFormer's SI-SNRi of 22+ dB on WSJ0-2mix was about 1–2 dB better than the best prior CNN or RNN separators, which is a notable jump in that field. These metrics translate to clearer speech and less noise by human perception. The gap between transformers and other architectures isn't infinite – for example, an LSTM or a well-tuned CNN can still reach high performance – but transformers

provide a more flexible framework to continue improving by scaling up model size or data. One should note that larger transformer models may diminish returns if not enough training data is available or if overfitting occurs; whereas CNNs with strong inductive biases might achieve similar performance on smaller datasets. A comparison by Wang *et al.* (2020) found that on some tasks, a TCN outperformed an LSTM, which outperformed a vanilla transformer when data was limited – indicating that each architecture's effectiveness can depend on the regime. Nonetheless, given enough data and proper design, transformers have a clear advantage in expressiveness and global context handling.

In summary, **transformers vs RNNs vs CNNs** can be seen as *global attention vs sequential memory vs local filtering*. Transformers excel in capturing global structure (at higher computational cost), RNNs handle moderate-length dependencies with efficient usage of parameters but sequential operation, and CNN/TCNs capture local and intermediate-range patterns with excellent efficiency but need help for truly global context. Modern speech enhancement systems often combine elements of all three (for example, a model might use a convolutional encoder, a transformer middle, and maybe a tiny RNN for post-processing). Compared to traditional DSP methods, all these neural approaches (especially transformers) represent a qualitative leap in noise suppression capability, at the cost of requiring more computation and training data.

Technical Challenges and Performance Considerations

While transformer-based models have shown superb performance, they come with a set of technical challenges that researchers are actively working to address:

- **Computational Cost and Sequence Length:** The most immediate challenge is the quadratic scaling of self-attention with sequence length. Speech signals can be several seconds or minutes long. A 5-second 16 kHz waveform, if converted to 10 ms frames, yields 500 frames – self-attention on 500 queries × 500 keys is 250k operations for one layer (and this grows with sequence squared). For longer inputs or higher frame rates, this quickly becomes impractical, especially for real-time use or on memory-constrained hardware. Solutions to this include:
    - *Segmenting the input:* The dual-path approach (used in SepFormer, DPRNN, etc.) limits attention to within chunks of, say, 50 or 100 frames, then has another module to handle interactions between chunks. This effectively makes the complexity O(n * m) where m is chunk size, instead of O(n^2) (if n >> m). The trade-off is a small drop in theoretical optimality, but models like SepFormer show you can recover global context via the second stage and still get excellent results.
    - *Efficient attention mechanisms:* There is growing research on variants like Linformer, Performer, Reformer, etc., which approximate or restrict attention to achieve linear or *sub-quadratic* complexity. In speech enhancement, *sparse attention* patterns such as local windows or combinations of local+global (e.g., a long-range attention every k frames) have been used. The *Ripple* sparse attention

(2023) is one such method where each frame attends to only a subset of others following a certain pattern. Another approach is multi-head attention with fixed patterns or learnable patterns that are constrained (like block sparse matrices).

- *Downsampling the sequence:* As discussed, using longer STFT frames or strides in the encoder can reduce the number of time steps. Subsampling layers (as used in some ASR transformers) could also be inserted to halve or quarter the time resolution in early layers (assuming the model can still handle the coarser resolution).

- *Model scaling:* Interestingly, larger transformers (more layers or heads) do not necessarily mean drastically worse efficiency if the sequence length is the bottleneck. Some researchers have trained *tiny* transformers for speech to allow longer lengths. Luo *et al.* (Interspeech 2022) created **Tiny-SepFormer**, reducing parameters down to ~5 million (from 26M) with clever layer sharing and convolution replacements, and they managed to train a 32-layer SepFormer within 16GB GPU memory by limiting sequence lengths and reusing parameters. They even note that a 32-layer model could be trained in 16GB whereas a naive attempt would have needed 32GB.

- **Real-Time Processing (Causality and Latency):** Many applications (voice chat, live streaming, hearing aids) require *causal* processing with minimal latency. A standard transformer layer is not causal because each output attends to future inputs as well. To deploy in real time, one must use *masked self-attention* (as used in transformer language models) to prevent looking ahead. This inherently makes the task harder (no glimpse of the future), so there is often a performance gap between causal and non-causal models. Studies have applied causal masking in speech enhancers and combined it with block processing: e.g., a short lookahead of a few frames might be allowed to balance latency and performance. Another trick is to use *encoder-decoder* style with the decoder receiving one frame at a time and attending over the full past encoded states (similar to how streaming ASR transformers work). The attention itself in a causal setting can be computed incrementally (reusing previous key/value projections to avoid recomputing the full matrix each frame). All this complexity means that real-time transformer denoisers are just emerging in research. For now, many real-time noise suppression systems use simpler models like GRU-based recurrent networks (e.g., Microsoft's 2020 Teams noise suppression used a DNN with LSTM). But as hardware accelerators improve and efficient transformer algorithms develop, we expect to see more transformer models in production. One example from literature combined transformers with a *dual-rate* approach: a fast update for immediate past frames and a slower update for global context, to simulate streaming with periodic global re-attention. These kinds of innovations aim to retain transformer performance while meeting latency budgets.

- **Positional Encoding and Generalization:** Unlike audio CNNs (which implicitly respect order with local filters) or RNNs (which step through time), transformers need explicit positional cues. A poor choice of positional encoding can limit a model's ability to handle sequences longer (or shorter) than those seen in training. Zhang *et al.* (Interspeech 2024) conducted an in-depth exploration of **length generalization** for transformer SE models. They found that *relative positional embeddings (RPE)* yield much better generalization to longer utterances than absolute positional encodings. Intuitively, relative encoding lets the model learn relations like "frame i is 5 steps ahead of frame j" without anchoring to an absolute

timeline, so it can apply the same behavior if those frames end up at positions further in a longer clip. This is an important consideration if one trains on, say, 2-second clips but needs to enhance a 10-second recording. Without the right positional scheme, the transformer might degrade on the longer input. Additionally, training data often comprises fixed-length segments, so generalizing to a continuous stream (which might effectively be a length mismatch) is crucial for deployment.

- **Handling Different Noise Conditions:** A challenge for all learning-based enhancers is to maintain performance across a wide range of noise types and signal-to-noise ratios (SNRs), including ones not seen during training. Transformers, by virtue of their capacity, can overfit to the training distribution if not regularized or given enough diversity. Data augmentation (adding various noises, reverbs, etc.) is commonly used. Some works incorporate *noise embeddings* or conditioning to help the model adjust to different noise profiles. There is also a trend of *domain adaptation* or *unsupervised fine-tuning*, where a pretrained model can be adapted to a new noise environment with a small amount of data. Because transformers have a high number of parameters, one concern is whether they might overfit more than smaller models – but there are techniques like dropout, layer normalization, and early stopping to mitigate this. Empirically, transformer SE models have shown strong robustness on unseen noises – for example, Yu *et al.* report their SE-Transformer had consistently better PESQ/STOI than LSTMs on *unseen* noise conditions. The attention mechanism likely helps in such generalization by *flexibly re-weighting* parts of the input; even if a noise type is new, the model can choose to attend less to the strange patterns and focus on the speech-like patterns.

- **Memory and Hardware Constraints:** Transformers often require more memory due to storing the $O(n^2)$ attention matrices. During training this is a big issue. Techniques like mixed precision (FP16/BF16 training) and gradient checkpointing (not storing intermediate activations for all layers) are used to fit models in GPU memory. As seen in Tiny-SepFormer experiments, *parameter sharing* across layers was one way to reduce memory and still go deep. Another idea is using lower-rank approximations of the attention matrix or limiting the number of heads if possible. On the hardware side, new AI accelerators (like GPUs with Transformer Engine, or TPUs, etc.) are making attention more efficient. For inference, one can also prune models or use knowledge distillation to create a smaller model that approximates a large one. This has been less explored in SE than in NLP, but it's a potential path: train a large transformer on a big dataset for best quality, then distill its behavior into a smaller (maybe hybrid) model for deployment.

- **Evaluation and Metrics:** Measuring speech enhancement quality is not trivial – metrics like PESQ and STOI correlate with human judgments but not perfectly. Often a model might improve one metric at slight expense of another (e.g. remove more noise but also remove some speech components). Transformers allow complex loss functions and multi-task learning. For instance, one could incorporate an ASR model's feedback (so-called *speech recognition metric* as part of training) to ensure the enhanced speech is not only clean but also ASR-friendly. Some recent works propose metrics based on neural network models (e.g. DNSMOS uses a deep model to predict mean opinion scores). In the DNS challenge results,

transformer models achieved high DNSMOS scores, but interestingly not always the top – this indicates that simply throwing a transformer at the problem doesn't automatically solve everything; careful training and model design still matter.

- **Performance Benchmarks:** To give a concrete sense, here are a few benchmark comparisons reported in literature:
    - *VoiceBank+DEMAND dataset (noise + reverberation)*: A classic baseline (SEGAN, a CNN-GAN from 2017) got PESQ ~2.16. By 2020, a CNN+LSTM model (like DCCRN) reached ~2.68 PESQ. Transformer-based models (e.g. SepFormer adaptation or MetricGAN+ with transformers) have pushed PESQ to ~3.0 or higher, and STOI from ~92% to ~95%. Kim and Seo's SE-Conformer exceeded previous baselines by a margin on PESQ (e.g. 0.1–0.2 absolute, which is noticeable) (Source: isca-archive.org).
    - *WSJ0-2mix (clean speech separation)*: TasNet (CNN) gave ~15 dB SI-SNRi, DPRNN (LSTM) about ~18 dB, SepFormer (Transformer) ~22 dB. That was a huge jump in that domain.
    - *DNS Challenge test set*: DNSMOS overall quality (OVRL) scores around 3.0–3.5 have been achieved by top contest models (often complex hybrid networks). A SepFormer trained on DNS data got OVRL ~2.4 on dev (as cited above), whereas the challenge winner (in 2022) might have been ~3.2. There is still room to improve for transformers on these real-world mixed distortion tasks, but they are closing the gap quickly.

Of course, raw numbers don't tell the full story, but the trend is clear: transformer-based models either match or set the new state-of-the-art in speech denoising benchmarks, albeit with a cost in computation. Each year since 2020, conferences have seen new variants (Gaussian attention, sparse attention, conformers, multi-stage networks) that incrementally improve performance or efficiency. We are now at a point where the question is not *if* transformers can be used for audio denoising (they certainly can, and very effectively), but rather *how to best use them* and *in what scenarios they provide the most benefit*.

Future Directions and Unresolved Challenges

The application of transformers to audio denoising is a burgeoning area, and several directions are likely to shape future research and development:

- **Improving Efficiency and Model Size:** As powerful as transformers are, real-world deployment demands efficient models. Future research will keep refining *lightweight transformer architectures* for speech. This could involve methods like **model compression** (pruning, quantization of transformer weights to int8 or lower precision), **knowledge distillation** (training a small model to mimic a large transformer's outputs), or architectural changes like the aforementioned hybrid designs. The **Tiny-SepFormer** study already showed one path: using convolution to handle local structure so that a transformer can be smaller yet retain performance. Another path is the use of **linear attention mechanisms** that reduce complexity (e.g.,

transformer's dot-product attention can potentially be approximated by low-rank factorizations or kernel methods). We might see *blockwise* or *chunked attention* become standard in audio, where a streaming audio is processed in blocks and a limited number of summary tokens carry information between blocks (similar to how the human auditory system might use working memory for long-term context). Research on *memory-augmented transformers* (where a fixed-size memory stores long-term info) could be adapted to continually process audio streams without blowing up computation.

- **Lower Latency and Causal Modeling:** Achieving *real-time* performance with transformers will be a key focus. One likely direction is designing transformers explicitly for streaming—perhaps using *transformer transducers* (drawing from ASR, where a prediction network and transformer encoder work together), or using *causal convolutions plus limited self-attention* to allow operation with minimal lookahead. We might also see *event-driven attention*, where the model decides to trigger a high-power computation when a potential speech event is detected and otherwise runs at low compute for silence or steady noise. Another unresolved challenge is the algorithmic latency introduced by attention spanning many frames; dynamically adapting the context length based on content (short for fast speech, longer for slow speech or high noise) could be explored.

- **Multi-Modal and Contextual Denoising:** Transformers are inherently flexible in terms of input modalities (we already have Vision Transformers, etc.). A future direction is combining audio transformers with other inputs like video (e.g., lip reading) or context from a conversation. For instance, in video conferencing, an audio-visual transformer could use the speaker's lip movements (via a visual front-end) to guide the audio enhancement. The attention mechanism could naturally align audio and video streams. Some works have started exploring this for tasks like speaker separation with video. Similarly, leveraging *contextual information* (such as knowing the speaker's voice characteristics or an estimate of the noise type) can be done via conditioning tokens or embeddings that a transformer can attend to. This is an advantage over traditional models: a transformer can be given an extra sequence (like a noise sample or a speaker embedding) and incorporate that through cross-attention, performing *personalized* or *noise-specific* enhancement. Recent research on personalized speech enhancement used the target speaker's embedding to extract only that speaker's voice from a mixture, something well-suited to the transformer's architecture (which can treat the speaker embedding as a guide query in attention).

- **Unified Speech Restoration Models:** Projects like VoiceFixer and VoiceRestore hint at a future where one model handles *multiple distortions simultaneously*: noise, reverberation, bandwidth limitation, etc.. Transformers, with their large capacity and ability to condition on various inputs, are prime candidates for such unified models. An unresolved challenge is how to make these models *robust* across all tasks and how to train them effectively (since the training data needs to cover many types of degradation). It may require multi-task learning where the model has objectives for denoising, dereverberation, etc., perhaps with a modular design (e.g., separate output heads or sequential stages). The attention mechanism could allow the model to *decouple* distortions—for example, attend to the direct sound vs. reverberation tail separately, or identify frequency bands affected by clipping versus those masked by noise. Solving these would greatly simplify audio cleanup workflows (one model could replace a chain of specialized algorithms).

- **Incorporating Human Perception in Training:** Despite advances in objective metrics, the ultimate judge of enhancement quality is human listeners. Future work may integrate perceptual loss functions or trained "critic" networks (as in GANs) more deeply with transformer models. Transformers have been used in **MetricGAN** frameworks where a generator network tries to maximize a differentiable metric predicted by a deep model. With transformers as the generator (enhancer), one might also use transformers as the discriminator or metric predictor to better evaluate the nuanced quality of speech. Additionally, focusing on *intelligibility* in low SNR conditions could mean using an ASR model's output as part of the loss (to ensure the enhanced speech is more transcribable). The attention in transformers might help emphasize phonetic cues for ASR, potentially improving downstream speech recognition accuracy in noise. This co-design of enhancement and recognition (or other tasks) is a fertile area (sometimes called joint front-end/back-end optimization).

- **Adapting to Device Constraints:** Running a transformer on a mobile or embedded device (like a hearing aid DSP or a smartphone) is challenging. Research may produce *hardware-friendly transformer variants* for audio – e.g., using block sparsity that maps well to memory hierarchies, or quantization schemes that maintain performance. Also, there might be exploration into neuromorphic hardware or event-based processing for audio with transformers. Although speculative, one can imagine leveraging the fact that speech has pauses and variable activity to dynamically adjust the model's operations (something akin to conditional computation or mixture-of-experts where only part of the model "activates" for a given frame).

- **Dealing with Phase and Spatial Aspects:** Many current spectral domain models estimate only the magnitude of the speech and reuse the noisy phase for reconstruction, which can limit maximum achievable quality. An unresolved technical challenge is *phase reconstruction*, which is needed especially for low-frequency noise or when removing reverb (phase carries spatial and temporal fine structure information). Some works have proposed complex-valued transformers or networks outputting real and imaginary parts or sinusoidal parameters. Extending attention to complex values is non-trivial but feasible (treating real and imaginary as two channels, for instance). In multi-microphone (spatial) enhancement, transformers could also be used to attend across channels – essentially performing beamforming-like operations in a learned way. For example, a cross-channel attention could weight and sum information from a microphone array, focusing on the direction of the target speaker. This is a promising direction since traditional beamforming can be suboptimal in dynamic noise scenarios. Early experiments (e.g., Multi-channel SepFormer) indicate transformers can jointly do spatial filtering and denoising.

- **Continual Learning and Adaptation:** In real environments, noise characteristics can change over time (imagine a denoiser in a laptop that sees different noise in a cafe vs. at home). Future systems might allow the transformer to *adapt on the fly*—possibly through a small amount of unsupervised learning (using the idea that during speech pauses, whatever is present is noise, and adjusting accordingly). The challenge here is avoiding catastrophic forgetting and maintaining stability. Meta-learning or online learning algorithms could be integrated so that the model has some "quick adaptation" parameters (maybe gating some attention heads based on recent noise statistics).

In conclusion, transformer architectures have proven not only viable but extremely effective for audio denoising. They harness the attention mechanism to differentiate human voice from noise by globally contextualizing the audio signal, something prior architectures struggled with. The journey is ongoing: researchers are taming the transformer's complexity and tailoring its behavior to audio signals' unique properties. As solutions to efficiency and streaming roll out, we anticipate transformers (and their hybrid descendants) becoming standard in both offline audio processing tools and real-time communication systems. The ability of transformers to *learn what to listen for* – focusing on speech and ignoring interference – aligns perfectly with the goal of speech enhancement. With continued innovation, the gaps in deploying these models at scale will close, and clearer, noise-free speech will be more accessible than ever, whether you're on a noisy call or restoring a historical recording. The attention mechanism, it turns out, is a powerful ally in the quest for noise-free voice communication.

Sources: The analysis above cites key research papers and resources that support the discussed concepts, including works on transformer networks for speech enhancement, specialized attention mechanisms, comparisons with other architectures, and recent benchmarks and advances in the field (Source: isca-archive.org). These sources provide detailed evidence of performance improvements and the design considerations addressed in this report.

Tags: transformers, speech denoising, attention mechanism, deep learning, audio signal processing, self-attention, speech enhancement

---

About ClearlyIP

ClearlyIP Inc. — Company Profile (June 2025)

---

1. Who they are

ClearlyIP is a privately-held unified-communications (UC) vendor headquartered in Appleton, Wisconsin, with additional offices in Canada and a globally distributed workforce. Founded in 2019 by veteran FreePBX/Asterisk contributors, the firm follows a "build-and-buy" growth strategy, combining in-house R&D with targeted acquisitions (e.g., the 2023 purchase of Voneto's EPlatform UCaaS). Its mission is to "design and develop the world's most respected VoIP brand" by delivering secure, modern, cloud-first communications that reduce cost and boost collaboration, while its vision focuses on unlocking the full potential of open-source VoIP for organisations of every size. The leadership team collectively brings more than 300 years of telecom experience.

---

2. Product portfolio

- **Cloud Solutions** – Including *Clearly Cloud* (flagship UCaaS), **SIP Trunking**, **SendFax.to** cloud fax, **ClusterPBX OEM**, **Business Connect** managed cloud PBX, and **EPlatform** multitenant UCaaS. These provide fully hosted voice, video, chat and collaboration with 100+ features, per-seat licensing, geo-redundant PoPs, built-in call-recording and mobile/desktop apps.
- **On-Site Phone Systems** – Including CIP PBX appliances (FreePBX pre-installed), ClusterPBX Enterprise, and Business Connect (on-prem variant). These offer local survivability for compliance-sensitive sites; appliances start at 25 extensions and scale into HA clusters.
- **IP Phones & Softphones** – Including CIP SIP Desk-phone Series (CIP-25x/27x/28x), fully white-label branding kit, and *Clearly Anywhere* softphone (iOS, Android, desktop). Features zero-touch provisioning via Cloud Device Manager or FreePBX "Clearly Devices" module; Opus, HD-voice, BLF-rich colour LCDs.
- **VoIP Gateways** – Including Analog FXS/FXO models, VoIP Fail-Over Gateway, POTS Replacement (for copper sun-set), and 2-port T1/E1 digital gateway. These bridge legacy endpoints or PSTN circuits to SIP; fail-over models keep 911 active during WAN outages.
- **Emergency Alert Systems** – Including **CodeX** room-status dashboard, **Panic Button**, and **Silent Intercom**. This K-12-focused mass-notification suite integrates with CIP PBX or third-party FreePBX for Alyssa's-Law compliance.
- **Hospitality** – Including **ComXchange** PBX plus PMS integrations, hardware & software assurance plans. Replaces aging Mitel/NEC hotel PBXs; supports guest-room phones, 911 localisation, check-in/out APIs.
- **Device & System Management** – Including **Cloud Device Manager** and **Update Control (Mirror)**. Provides multi-vendor auto-provisioning, firmware management, and secure FreePBX mirror updates.
- **XCast Suite** – Including Hosted PBX, SIP trunking, carrier/call-centre solutions, SOHO plans, and XCL mobile app. Delivers value-oriented, high-volume VoIP from ClearlyIP's carrier network.

---

3. Services

- **Telecom Consulting & Custom Development** – FreePBX/Asterisk architecture reviews, mergers & acquisitions diligence, bespoke application builds and Tier-3 support.
- **Regulatory Compliance** – E911 planning plus **Kari's Law**, **Ray Baum's Act** and **Alyssa's Law** solutions; automated dispatchable location tagging.
- **STIR/SHAKEN Certificate Management** – Signing services for Originating Service Providers, helping customers combat robocalling and maintain full attestation.
- **Attestation Lookup Tool** – Free web utility to identify a telephone number's service-provider code and SHAKEN attestation rating.
- **FreePBX® Training** – Three-day administrator boot camps (remote or on-site) covering installation, security hardening and troubleshooting.
- **Partner & OEM Programs** – Wholesale SIP trunk bundles, white-label device programs, and ClusterPBX OEM licensing.

---

4. Executive management (June 2025)

- **CEO & Co-Founder: Tony Lewis** – Former CEO of Schmooze Com (FreePBX sponsor); drives vision, acquisitions and channel network.
- **CFO & Co-Founder: Luke Duquaine** – Ex-Sangoma software engineer; oversees finance, international operations and supply-chain.
- **CTO & Co-Founder: Bryan Walters** – Long-time Asterisk contributor; leads product security and cloud architecture.
- **Chief Revenue Officer: Preston McNair** – 25+ years in channel development at Sangoma & Hargray; owns sales, marketing and partner success.
- **Chief Hospitality Strategist: Doug Schwartz** – Former 360 Networks CEO; guides hotel vertical strategy and PMS integrations.
- **Chief Business Development Officer: Bob Webb** – 30+ years telco experience (Nsight/Cellcom); cultivates ILEC/CLEC alliances for Clearly Cloud.
- **Chief Product Officer: Corey McFadden** – Founder of Voneto; architect of EPlatform UCaaS, now shapes ClearlyIP product roadmap.
- **VP Support Services: Lorne Gaetz** (appointed Jul 2024) – Former Sangoma FreePBX lead; builds 24×7 global support organisation.
- **VP Channel Sales: Tracy Liu** (appointed Jun 2024) – Channel-program veteran; expands MSP/VAR ecosystem worldwide.

---

5. Differentiators

- **Open-Source DNA:** Deep roots in the FreePBX/Asterisk community allow rapid feature releases and robust interoperability.
- **White-Label Flexibility:** Brandable phones and ClusterPBX OEM let carriers and MSPs present a fully bespoke UCaaS stack.
- **End-to-End Stack:** From hardware endpoints to cloud, gateways and compliance services, ClearlyIP owns every layer, simplifying procurement and support.
- **Education & Safety Focus:** Panic Button, CodeX and e911 tool-sets position the firm strongly in K-12 and public-sector markets.

In summary

ClearlyIP delivers a comprehensive, modular UC ecosystem—cloud, on-prem and hybrid—backed by a management team with decades of open-source telephony pedigree. Its blend of carrier-grade infrastructure, white-label flexibility and vertical-specific solutions (hospitality, education, emergency-compliance) makes it a compelling option for ITSPs, MSPs and multi-site enterprises seeking modern, secure and cost-effective communications.
