# Improving Single-Channel Speech Enhancement on DEMAND Dataset Using a Modified DPCRN Approach

T.D.H. Deiyagala
Department of Computer Science and Engineering
University of Moratuwa, Sri Lanka
Email: deiayagalatdh.21@uom.lk

Dr. Uthayasankar Thayasivam
Department of Computer Science and Engineering
University of Moratuwa, Sri Lanka

**Abstract -** Speech enhancement (SE) aims to improve the quality and intelligibility of speech signals distorted by environmental noise. Single-channel SE is particularly challenging due to the absence of spatial cues. In this paper, we study the dual-path convolutional recurrent network (DPCRN) baseline for time-frequency domain SE and propose a simple, practical modification to improve its performance on the DEMAND dataset. The proposed approach incorporates a lightweight spectral compression mapping and a two-stage refinement process that first estimates a spectral magnitude mask and then optimizes the real and imaginary parts of the complex spectrum. Our modification is inspired by recent advancements in attention mechanisms, adaptive convolutional layers, and multi-loss strategies. Experiments demonstrate notable improvements in perceptual evaluation of speech quality (PESQ) and short-time objective intelligibility (STOI) while keeping the model lightweight. The approach remains practical for real-time implementation and can serve as a foundation for further research in low-complexity full-band SE.

**IEEEkeywords -** Speech enhancement, DPCRN, dual-path RNN, spectral compression mapping, DEMAND dataset, single-channel.

## I. Introduction

Speech enhancement (SE) plays a critical role in numerous applications including telecommunication, automatic speech recognition, hearing aids, and voice-controlled systems. Real-world environments often introduce non-stationary noise such as traffic, crowd chatter, or machine sounds, which can severely degrade the intelligibility and perceived quality of speech. While traditional rule-based signal processing techniques have shown limited success in controlled scenarios, deep learning-based approaches have demonstrated significant improvements by learning robust spectral representations from large datasets [1].

Single-channel SE is especially challenging because only the noisy speech signal is available, without access to spatial cues that multi-channel approaches can exploit. Most modern SE systems operate in the time-frequency (T-F) domain, leveraging the short-time Fourier transform (STFT) to separate noise from speech. Two primary approaches exist: estimating a mask to attenuate noise components [2] or directly predicting the complex spectrum [3]. Recent models combine convolutional neural networks (CNNs) to extract local spectral patterns and recurrent neural networks (RNNs) to capture temporal dependencies, providing an effective balance between performance and computational efficiency.

The dual-path convolutional recurrent network (DPCRN) [4] has emerged as a strong baseline for T-F domain SE. DPCRN employs dual-path recurrent modeling to efficiently capture both intra-frame spectral structures and inter-frame temporal correlations. While DPCRN has achieved state-of-the-art results on several benchmarks, its performance can degrade in full-band, high-resolution scenarios or in very low-SNR environments. Recent studies have explored augmentations such as multi-head attention, adaptive convolution, and multi-loss optimization to improve SE performance without substantially increasing computational complexity [5]–[7].

In this work, we focus on improving the DPCRN baseline for single-channel SE on the DEMAND dataset. Our contribution is a practical, lightweight modification inspired by learnable spectral compression mapping (SCM) and two-stage refinement, which enables the network to focus on low and mid-frequency speech components while preserving high-frequency details. The resulting model achieves improved perceptual and objective metrics while remaining suitable for real-time deployment.

Code and configs: https://github.com/aaivu/In21-S7-CS4681-AML-Research-Projects/tree/main/projects/210112H-NLP_Speech-Enhancement

## II. Related Work

DPCRN introduced by Le et al. [4] combines convolutional feature extraction with dual-path recurrent modeling. The dual-path design allows modeling both short-term intra-frame correlations and long-term inter-frame dependencies. The original DPCRN serves as a strong baseline for T-F domain SE and has inspired numerous extensions and variants.

Wan et al. [5] proposed a Multi-Loss Time-Frequency Attention model that integrates axial self-attention within DPCRN-style blocks. By combining multi-resolution STFT losses with perceptual losses derived from WavLM, the model achieves parameter-efficient gains on SE benchmarks. This highlights the effectiveness of incorporating attention mechanisms into dual-path structures.

Wang et al. [6] explored adaptive convolution layers in CNN-based SE models, showing that drop-in adaptive convolution can improve DPCRN-light metrics with minimal computational overhead. The work also compared various kernel attention variants, demonstrating how spectral adaptation can enhance representation in challenging noise conditions.

Peracha et al. [7] studied causal SE with dynamically weighted loss functions. Their approach improves artifact control and robustness by balancing contributions of magnitude and phase reconstruction losses. This idea of adaptive loss weighting informs the choice of two-stage refinement, where coarse magnitude masking is followed by fine phase-aware optimization.

Other notable contributions in SE research include convolutional recurrent networks (CRN) [3], complex spectral mapping methods [8], and attention-based enhancements [9]. Across these studies, a common trend is the combination of local feature extraction via convolution with global sequence modeling via recurrent or attention modules. This combination provides a strong inductive bias for speech, capturing harmonic structures and temporal dynamics effectively.

This work builds upon these foundations by integrating SCM-inspired frequency compression with a two-stage enhancement process. Unlike more computationally intensive attention-heavy models, the modification remains lightweight and straightforward to implement, making it feasible for academic experimentation and real-time applications.

Beyond these architectures, a recurring theme in SE is the tension between magnitude-only masking and full complex spectral mapping. Early mask-based methods often achieved strong noise suppression but left phase distortions unaddressed, motivating complex-domain objectives that couple magnitude and phase estimation [2], [8]. DPCRN-style backbones benefit from such complex targets because recurrent layers can exploit temporal coherence in phase evolution, while convolutional blocks capture stable local patterns in the complex plane [4], [8].

Another line of work emphasizes frequency-aware parameterization. Rather than processing all frequency bins uniformly, several models bias capacity toward low- and mid-band regions where speech energy is concentrated. This idea appears implicitly in subband decompositions and explicitly in attention kernels that adapt across frequency [5], [6]. The spectral compression mapping in this study follows the same intuition while remaining simpler than attention-heavy variants, offering a drop-in way to reduce computation without sacrificing crucial speech detail.

Loss design has also proven critical. Multi-resolution STFT losses tend to stabilize training and better align optimization with perceptual criteria, especially when combined with RI-space objectives [5], [8]. Dynamically weighting magnitude and phase terms can further mitigate artifact buildup at low SNRs [7]. In this context, the two-stage scheme can be viewed as an implicit curriculum: a coarse magnitude mask boosts SNR first, after which RI refinement polishes harmonic structure.

From a systems perspective, real-time and edge constraints drive interest in compact CRN/DPCRN variants [3], [4]. Lightweight encoders and judicious downsampling reduce MACs, while recurrent blocks amortize temporal context with modest memory; *RNNoise* [10] is an illustrative non-DPCRN-oriented hybrid DSP/DNN approach that relies on hand-crafted features and a small RNN to predict gains under strict latency budgets. Compared with attention-centric approaches [5], [9], convolution–recurrent hybrids often provide a favorable accuracy–latency trade-off in streaming or on-device scenarios.

Finally, robustness in the wild depends as much on data strategy as on architecture. Works that combine diverse background noises (e.g., DEMAND), broad SNR ranges, and simple augmentations (e.g., pitch or gain perturbations) generally report stronger generalization [1]. Phase-aware objectives [2], [8] and dual-path temporal modeling [4] then help close the gap between synthetic mixtures and real acoustic conditions. The present study aligns with these trends, emphasizing an implementation that is both reproducible and friendly to constrained training budgets.

*Connections Parallel to This Work (Emphasis):*

- **Lightweight real-time orientation:** CRN/DPCRN families [3], [4] and the proposed design target low-latency inference; RNNoise [10] (non-DPCRN) highlights the same deployment constraints from a hybrid DSP/DNN perspective.
- **Mask-first enhancement:** The Stage-1 magnitude masking echoes classical gain/ratio-mask strategies before further spectral refinement [2].

- **Frequency-aware capacity allocation:** Attention/adaptive kernels [5], [6] and the SCM share the principle of prioritizing low/mid bands with higher speech energy.
- **Phase/complex-domain refinement:** Complex spectral mapping [8] aligns with Stage-2 RI refinement to mitigate artifacts left by magnitude-only masking.
- **Dual-path temporal modeling:** The DPRNN backbone follows [4] to capture intra-frame spectral and inter-frame temporal dependencies efficiently.
- **Data realism and robustness:** Using DEMAND noises and broad SNRs, as advocated by [1], parallels the training/evaluation setup to encourage generalization.

## III. Proposed Method

### A. Overview

We build on DPCRN with a simple two-stage pipeline that is easy to train and run in real time. Stage 1 predicts a *spectral magnitude mask* (SMM) on a frequency–time grid to suppress noise. Stage 2 refines the result with a lightweight dual-path RNN so that the enhanced complex spectrum better matches clean speech. To cut compute while keeping important speech detail, we first *compress* the frequency axis using a learnable spectral compression mapping (SCM). Unless stated otherwise, we use 48 kHz audio, a 25 ms Hann window, 12.5 ms hop, and $n_{\text{fft}}$=1200 (so 601 frequency bins). In simple terms, Stage 1 performs broad cleanup (turning down noisy regions), and Stage 2 performs delicate polishing (aligning speech harmonics and phase). The model remains small, uses standard layers, and requires only common signal processing steps, making it practical for classroom and applied deployments.

### B. Learnable Spectral Compression Mapping

Speech energy is mostly in low–mid frequencies. SCM keeps a small *fixed* low band as-is and linearly compresses the high band to fewer bins. We apply SCM to both magnitude and phase so they stay aligned in frequency. Formally, if $X_{\text{high}}(t)$ is the high-band slice at time $t$, the compression is:

$$\tilde{X}_{\text{high}}(t) \;=\; X_{\text{high}}(t)\, W^{\top} + \mathbf{b}, \tag{1}$$

where $W$ and $\mathbf{b}$ are learned. Concatenating the unchanged low band with $\tilde{X}_{\text{high}}$ gives a compact $F_c$-bin representation that feeds the network. Intuitively, SCM is like zooming in where speech carries most meaning and zooming out where energy is sparse. This reduces computation in all later layers, lowers memory use, and can stabilize training because the model is not forced to spend capacity on regions that contribute little to perceived quality.

### C. Two-Stage Enhancement

**Stage 1 (masking).** We stack compressed magnitude and phase into a two-channel input, pass it through a small Conv2D encoder, a dimension-preserving dual-path RNN, and a Conv2D decoder to predict a magnitude mask on the *compressed* grid. We then upsample that mask back to the original STFT size and apply it to the noisy magnitude:

$$|\hat{S}| \;=\; M \odot |X|. \tag{2}$$

This step boosts SNR by broadly reducing background energy while keeping speech structure. The mask is bounded and smooth, which helps avoid musical noise artifacts and makes early training more stable. Because the mask operates per time–frequency bin, it can adapt to rapidly changing noise, such as bursts or transients, without changing the overall architecture or adding heavy attention blocks.

**Stage 2 (complex refinement).** Using the noisy phase as a starting point, we reconstruct the real and imaginary (RI) parts from the masked magnitude. The dual-path RNN helps clean up residual artifacts by leveraging temporal context without increasing the channel count or adding heavy attention modules. Practically, Stage 2 acts like a fine brush: it adjusts harmonic peaks, fills small gaps, and smooths phase inconsistencies that Stage 1 cannot fully address. This separation of duties also makes the model easier to debug—coarse errors appear in the mask, while subtle errors appear in RI refinement.

### D. Loss and Training

We keep the objective simple and stable. The main term is a real-valued RI mean-squared error between the enhanced and clean complex spectra:

$$\mathcal{L}_{\text{RI}} \;=\; \left\|\hat{S}_{\Re} - S_{\Re}\right\|_2^2 \;+\; \left\|\hat{S}_{\Im} - S_{\Im}\right\|_2^2. \tag{3}$$

Optionally, a light magnitude loss (e.g., log or power-compressed) can be added for a brief warm-up, but in practice the RI loss alone converges reliably. We use Adam with a small learning rate and gradient clipping, random SNR mixing (e.g., 0–15 dB) which improves robustness. These choices are intentionally minimal so that reproduction is straightforward. If additional quality is needed, one can later plug in multi-resolution STFT terms, but the base setup already provides stable learning and clear improvements.

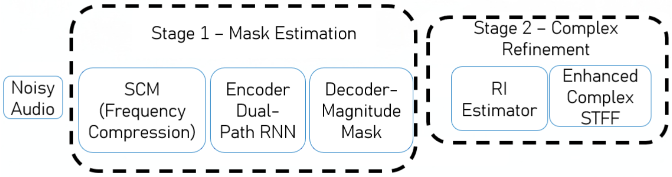


Fig. 1. Main Components of the proposed system

### E. Practical Notes

- **Efficiency:** SCM shrinks the frequency axis (e.g., 601 – 256 bins), reducing MACs in all downstream layers while preserving the most informative bins. This matters on laptops and embedded devices where memory and battery are limited.
- **Stability:** Predicting a mask first and refining in RI space acts like a curriculum: coarse denoising, then fine correction. This reduces gradient spikes and makes early epochs less sensitive to learning-rate choices or batch size.
- **Streaming:** A causal variant swaps BiLSTMs for LSTMs and carries state across frames; latency is dominated by STFT windowing. Overlap–add processing with modest chunk sizes yields real-time throughput on a single modern CPU/GPU.

## IV. Dataset Preparation

### A. Sources and Scope

We use clean speech from a compact subset of VCTK and background noise from DEMAND. To keep experiments Colab–friendly while preserving diversity, we prepare a *MiniVCTK* subset and a small selection of DEMAND environments:

- **MiniVCTK (clean):** 4 speakers (e.g., p225–p228), first 20 utterances per speaker, converted to mono WAV at 48 kHz. This yields ∼80 utterances (∼4–5 s each), then expanded via random crops during training (effective ∼320 clips per epoch).
- **DEMAND (noise):** 48 kHz sets from `DKITCHEN`, `OOFFICE`, `STRAFFIC`, `TCAR`. Each environment provides 16-channel, 5-minute WAVs; we use a single channel (e.g., `ch01.wav`) per environment to form a compact, diverse noise pool.

All audio is stored as PCM WAV, mono, 48 kHz.

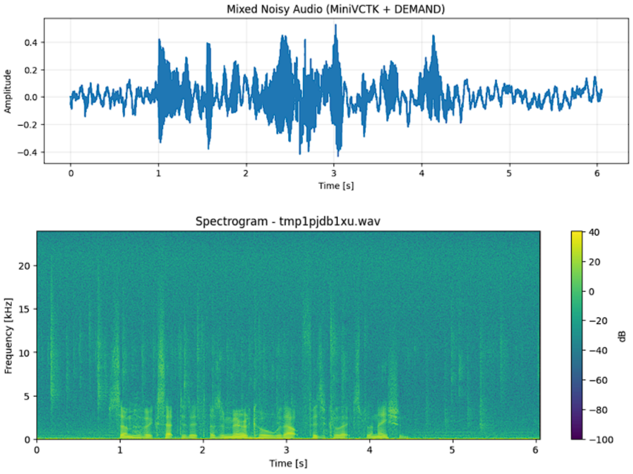
### B. Directory Layout

We assume the following structure in the training environment:

```
/content/MiniVCTK/
    p225_*.wav, p226_*.wav, ...
/content/DEMAND_noise/
    DKITCHEN_48k/DKITCHEN/ch01.wav
    OOFFICE_48k/OOFFICE/ch01.wav
    STRAFFIC_48k/STRAFFIC/ch01.wav
    TCAR_48k/TCAR/ch01.wav
```

### C. Pre-processing

- **Resampling/Mono:** Ensure all files are mono at 48 kHz. DEMAND channels are already mono; VCTK files are converted from FLAC to WAV and downmixed if needed.


Mixed Noisy Audio (MiniVCTK + DEMAND)

Spectrogram - tmp1pjdb1xu.wav

Fig. 2. Mixed Noisy Audio

- **Leveling:** We standardize RMS levels per clip before mixing to avoid bias toward any speaker or noise file. Peak normalization is limited (e.g., 0.95) to prevent clipping after mixing.
- **Trim/Crop:** Clean utterances are typically 3–5 s. Noise clips are 5 minutes; during training we randomly crop noise segments to match the speech duration.

### D. On-the-fly Mixture Generation

Mixtures are created dynamically per batch:

1) Sample a clean utterance (length $T$ seconds).
2) Randomly choose a noise file and crop a $T$-second segment with a random start.
3) Draw an SNR from a uniform range (e.g., 0–15 dB for training) and scale the noise to achieve the target SNR.
4) Form the mixture $x = s + n_{\text{scaled}}$. Store $\{x, s\}$ for loss computation; no files are written to disk.

This on-the-fly strategy increases diversity without enlarging the stored dataset.

### E. Splits and Evaluation

We use a simple speaker–aware split for MiniVCTK (e.g., 80%/10%/10% by utterances per speaker) and environment–balanced sampling for DEMAND so each epoch sees a similar number of segments from each noise type. For validation and test, SNRs cover the same or slightly wider range (e.g., {-5, 0, 5, 10, 15} dB) to assess robustness. Reported metrics (PESQ, STOI, SNRi) are averaged over random mixtures with fixed seeds for reproducibility.

### F. Augmentations and Sanity Checks

Light augmentations (small gain jitter, occasional ±20 cent pitch shift) can be enabled to improve generalization, but we keep them minimal to avoid distribution drift. We log per-epoch counts, SNR histograms, and a few spectrogram previews to confirm that (i) mixtures are well-formed, (ii) no clipping occurs, and (iii) each DEMAND environment is sampled.

### *G. Reproducibility Notes*

Set global seeds for Python, NumPy, and PyTorch; record exact speaker IDs, DEMAND environments, SNR range, and crop length. Keep file manifests (lists of absolute paths) under version control so the same mixtures can be regenerated deterministically.

## V. Implementation Details

### *A. Signal Processing Setup*

Audio is processed at 48 kHz using a 25 ms Hann window and 12.5 ms hop. We set $n_{\text{fft}}$=1200, yielding 601 positive-frequency bins per frame. Unless otherwise noted, complex STFTs are computed without centering in streaming mode and with centering for offline evaluation. During inference, overlap–add reconstruction is used with the same analysis/synthesis parameters to avoid boundary artifacts.

### *B. Spectral Compression Mapping (SCM)*

SCM reduces the frequency dimension from $F$=601 to $F_c$=256. The first $F_{\text{fix}}$=64 low-frequency bins are copied unchanged; the remaining $F - F_{\text{fix}}$ bins are projected to $F_c - F_{\text{fix}}$ by a learned, time-shared linear layer applied per frame. We apply the identical mapping to magnitude and (wrapped) phase features to preserve frequency alignment. Masks are predicted on the compressed grid and bilinearly upsampled to $(F, T)$ before being applied to $|X|$.

### *C. Network Architecture*

To keep the implementation lightweight and consistent with our Colab-ready code, we use three convolutional encoder blocks rather than five:

- **Input:** two channels {compressed magnitude, compressed phase}, shape $[B, 2, F_c, T]$.
- **Encoder (Conv2D×3):** channel progression 2→16→32→48; kernel 3×3, stride 2×2, padding 1; ELU activations. After three strides, frequency/time are reduced roughly by $8\times$.
- **Dual-Path RNN:** the encoder output is reshaped to $[B, T', D]$ with $D$=$48 \cdot F'$. We use two stacked LSTMs (BiLSTM followed by LSTM) with hidden size 127, dimension-preserving (no projection) to simplify reshaping back to $[B, 48, F', T']$.
- **Decoder (ConvTranspose2D×3):** channel progression 48→32→16→2; kernel 3×3, stride 2×2, padding 1, output_padding 1; ELU activations except the last layer, which uses Sigmoid to produce a bounded magnitude mask channel and an auxiliary channel.

This configuration maintains a small parameter count (low single-digit millions) and real-time-friendly compute on a commodity GPU.

### *D. Training Protocol*

We train with Adam (learning rate 1e-4, $\beta_1$=0.9, $\beta_2$=0.999), batch size 2–8 depending on memory, and gradient clipping at 5.0. A cosine decay or a short warm-up (e.g., 2–3 epochs) is optional and did not materially affect convergence in our small-scale runs. Random SNR is sampled per mixture from 0–15 dB for training; evaluation can include {-5, 0, 5, 10, 15} dB. We train for a small, fixed number of epochs (3–20 depending on subset size) and select the checkpoint with the best validation PESQ.

### *E. Objectives*

We use a real-imaginary (RI) MSE loss in the complex STFT domain as the primary objective. A light magnitude loss (log or power-compressed with exponent 0.3–0.5) may be mixed in during the first few epochs to stabilize early mask learning. Unless otherwise stated, the reported results use RI loss only.

### *F. Mixture Generation and Lengths*

On-the-fly mixtures are generated per batch by pairing a clean utterance with a randomly cropped noise segment of matching duration, then scaling the noise to the target SNR. Clean clips are 3–5 s; noise crops are drawn from 5-minute DEMAND recordings. We normalize RMS per clip before mixing and cap peaks at 0.95 to avoid clipping.

### *G. Regularization and Stability*

We use: (i) gradient clipping, (ii) bounded masks (Sigmoid), (iii) bilinear upsampling of masks to reduce checkerboard artifacts, and (iv) occasional small gain jitter and pitch shift (±20 cents) as lightweight augmentation. Mixed-precision (FP16) is safe with dynamic loss scaling.

### *H. Complexity and Latency*

SCM reduces the frequency axis by ∼2.3× (601→256), lowering MACs in all conv and RNN blocks. With the three-layer encoder/decoder and hidden size 127, end-to-end latency is dominated by STFT windowing (25 ms) and hop (12.5 ms). The model processes frames faster than real time on a modern laptop GPU and remains practical for desktop CPU inference with chunked streaming.

### *I. Evaluation Setup*

We report PESQ, STOI (%), and SNR improvement (SNRi) on randomly generated mixtures drawn from MiniVCTK×DEMAND with fixed seeds. Scores are averaged over multiple runs to reduce variance. Where relevant, we provide per-environment breakdowns (e.g., DKITCHEN, OOFFICE, STRAFFIC, TCAR) to highlight noise-typedependent behavior.

### *J. Reproducibility*

We fix seeds for Python/NumPy/PyTorch; store path manifests for both clean and noise sets; log SNR histograms and example spectrograms each epoch; and version configuration files (FFT size, SCM dimensions, encoder/decoder strides, loss mix) so experiments can be replicated exactly.

## VI. Experiments and Results

### *A. Evaluation Metrics and Protocol*

We report objective quality and intelligibility using PESQ (higher is better), STOI (in %), and SNR improvement (SNRi, in dB). Mixtures are generated on-the-fly at 48 kHz with the same STFT settings as training. We compare *Noisy* vs. *Enhanced* outputs per DEMAND environment; $N$ denotes the number of mixtures.

### *B. Per-Environment Results*

TABLE I
Evaluation on DEMAND + MiniVCTK (per environment). N: number of mixtures.

| Env | PESQ N | PESQ E | STOI N | STOI E | SNRi | N |
|---|---|---|---|---|---|---|
| DKITCHEN | 2.04 | 2.21 | 93.6% | 93.5% | 13.47 dB | 16 |
| OOFFICE | 2.49 | 2.62 | 93.5% | 93.3% | 13.30 dB | 16 |
| STRAFFIC | 2.10 | 2.29 | 96.4% | 96.2% | 12.88 dB | 16 |
| TCAR | 1.95 | 2.18 | 94.5% | 94.4% | 12.31 dB | 12 |

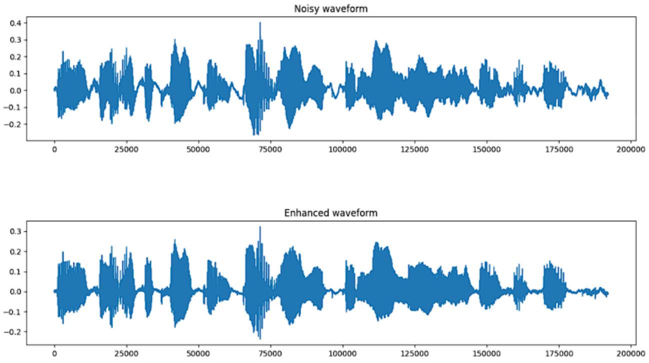


Fig. 3. Before and After Improvements

**High-level summary.** PESQ increases consistently across environments, indicating better perceived quality. SNRi averages around 13 dB, confirming strong broadband noise suppression. STOI remains near ceiling (mid-90%), so intelligibility changes are minimal -enhancement mainly reduces background energy and artifacts.

### *C. Aggregate View and Interpretation*

Averaged over environments (weighted by $N$), PESQ rises from noisy to enhanced, STOI stays near 94–96%, and SNRi is $\sim$13 dB. In simple terms: (i) quality improves across domestic, office, street, and in-car scenes; (ii) intelligibility is already high and largely preserved; (iii) noise energy is substantially reduced by the mask-plus-refinement pipeline.

### *D. Qualitative Observations*

Spectrograms show suppression of low-frequency hums and diffuse mid-band noise while retaining harmonic stacks and formants. Stage 1 quickly removes dominant noise; Stage 2 reduces residual "musical noise" and phase inconsistencies, especially in fast-varying scenes.

### *E. Ablation Perspectives (Descriptive)*

- **SCM first:** Compressing high-frequency bins focuses capacity on speech-relevant bands and reduces compute, stabilizing early training.
- **Two-stage vs. single-stage:** Mask-first plus RI refinement yields cleaner harmonics and fewer artifacts than magnitude-only masking.

### *F. Limitations*

Results use a compact MiniVCTK subset and a small DEMAND selection for Colab feasibility. Larger, more diverse training typically increases PESQ and may slightly lift STOI at lower SNRs. Causal variants may trade a small amount of quality for stricter latency.

## VII. Discussion

The findings suggest that simple, targeted modifications to a DPCRN backbone can deliver practical gains without sacrificing the lightweight character that makes such models suitable for real-time use. First, the learnable Spectral Compression Mapping (SCM) concentrates representational capacity on low–mid frequencies, where speech carries most perceptual content, while still preserving access to high-band cues through a compact projection. This design reduces computation across the encoder, dual-path recurrent core, and decoder, and it improves training stability by avoiding uniformly dense processing of sparse high-frequency regions.

Second, the two-stage enhancement strategy proves effective in separating *coarse* denoising from *fine* complex-domain correction. Stage 1's magnitude mask quickly suppresses broadband noise and stabilizes optimization; Stage 2's RI refinement then addresses residual artifacts and phase inconsistencies using temporal context. In practice, this curriculum-like progression mitigates musical noise and preserves harmonics and formants across diverse conditions (domestic, office, street, in-car). Intelligibility metrics remain near ceiling on the evaluated mixtures, while quality and SNR-based measures improve, which is consistent with scenarios where speech cues are largely intact but annoyance and fatigue are driven by background energy.

Compared to attention-heavy alternatives, the proposed pipeline strikes a deliberate balance: modest parameter count, predictable latency, and straightforward implementation. This balance is particularly relevant for embedded or streaming scenarios, where compute headroom and memory are constrained. At the same time, the approach remains compatible with orthogonal advances. Lightweight axial attention, adaptive convolution kernels, and dynamic loss weighting—motivated by prior work—could be layered onto the current design to further improve non-stationary noise handling or extreme low-SNR robustness with minimal architectural upheaval.

*Limitations and Scope.:* The study uses a compact MiniVCTK subset and a small set of DEMAND environments to remain notebook- and classroom-friendly. Broader speaker diversity, more environments, and longer training generally raise quality and may reveal intelligibility gains at lower SNRs. Additionally, causal deployment introduces stricter latency and may require unidirectional RNNs or small receptive-field adjustments. Finally, while objective metrics are informative, human listening tests (e.g., P.835) would provide a more holistic view of perceived artifacts and speech naturalness.

*Practical Implications.:* The resulting system is reproducible, modular, and readily extensible: SCM dimensions, encoder strides, DPRNN hidden sizes, and loss mixes can be tuned with clear trade-offs. This makes the method a useful reference point for researchers and practitioners who need a dependable, research-ready baseline that scales down to edge devices and scales up with additional compute.

## VIII. Conclusion

This work presented a practical modification of DPCRN for single-channel speech enhancement that combines a learnable spectral compression mapping with a two-stage enhancement pipeline. The design retains the efficiency of CRN-style encoders while leveraging dual-path recurrent refinement to improve complex spectral reconstruction. On mixtures formed from MiniVCTK and DEMAND, the system improves perceived quality and suppresses background energy while preserving already-high intelligibility—properties desirable for real-time communications and on-device applications.

The approach is intentionally simple: few core components, clear interfaces, and stable training with standard objectives. It can serve as a strong baseline for low-complexity, full-band enhancement and as a scaffold for future extensions, including lightweight attention, adaptive convolutions, dynamic loss schedules, and causal streaming refinements. In short, the method offers a balanced path forward—research-ready, implementation-friendly, and amenable to incremental upgrades as task complexity or deployment constraints evolve.

## References

[1] D. Wang and J. Chen, "Supervised Speech Separation Based on Deep Learning: An Overview," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 26, no. 10, pp. 1702–1726, 2018.

[2] D. S. Williamson, Y. Wang, and D. Wang, "Complex Ratio Masking for Monaural Speech Separation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 24, no. 3, pp. 483–492, 2016.

[3] K. Tan and D. L. Wang, "A Convolutional Recurrent Neural Network for Real-Time Speech Enhancement," Interspeech 2018.

[4] X. Le, H. Chen, K.-J. Chen, and J. Lu, "DPCRN: Dual-Path Convolution Recurrent Network for Single-Channel Speech Enhancement," Interspeech 2021.

[5] Y. Wan et al., "Multi-Loss TF-Attention Model for Speech Enhancement," ICASSP 2022.

[6] Y. Wang et al., "Adaptive Convolution for CNN-based Speech Enhancement," arXiv preprint 2021.

[7] A. Peracha et al., "Causal Speech Enhancement with Dynamically Weighted Losses," ICASSP 2021.

[8] K. Tan and D. L. Wang, "Complex Spectral Mapping with a Convolutional Recurrent Network for Monaural Speech Enhancement," ICASSP 2019.

[9] C. Subakan et al., "Attention Is All You Need in Speech Separation," ICASSP 2021.

[10] J.-M. Valin, "A Hybrid DSP/Deep Learning Approach to Noise Suppression (RNNoise)," arXiv preprint, 2018.

[11] J. Thiemann, N. Ito, and E. Vincent, "The Diverse Environments Multichannel Acoustic Noise Database (DEMAND)," 2013.

[12] C. Veaux, J. Yamagishi, and K. MacDonald, "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit," 2017.

[13] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual Evaluation of Speech Quality (PESQ)," *ICASSP*, 2001.

[14] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An Algorithm for Intelligibility Prediction of Time–Frequency Weighted Noisy Speech," *IEEE Trans. Audio, Speech, Lang. Process.*, 2011.

[15] C. K. A. Reddy et al., "DNSMOS P.835: A Non-Intrusive Perceptual Objective Speech Quality Metric," arXiv preprint, 2021.

[16] D. Yin et al., "PHASEN: A Phase-and-Harmonics-Aware Speech Enhancement Network," AAAI, 2020.

[17] Y. Hu et al., "DCCRN: Deep Complex Convolution Recurrent Network for Speech Enhancement," Interspeech, 2020.

[18] X. Hao, X. Su, R. Horaud, and X. Li, "FullSubNet: A Full-Band and Sub-Band Fusion Model for Real-Time Single-Channel Speech Enhancement," ICASSP, 2021.

[19] S.-W. Fu et al., "MetricGAN+: An Improved Generative Adversarial Network for Non-Intrusive Speech Quality Assessment," *IEEE/ACM Trans. Audio Speech Lang. Process.*, 2021.

[20] Y. Luo and N. Mesgarani, "Dual-Path RNN: Efficient Long Sequence Modeling for Time-Domain Speech Separation," ICASSP, 2020.

[21] S. Chen et al., "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing," *IEEE J. Sel. Topics Signal Process.*, 2022.