Targeted Audio-Visual Speech Separation Under Data-Limited Conditions: A Comparative Analysis

Nikolai Prudnikov
Faculty of World Economy and International Affairs
HSE University, Moscow
nprudnickov62@gmail.com

Ilyas Khassenov
Faculty of World Economy and International Affairs
HSE University, Moscow
hasenov.ilias@gmail.com

***Abstract*—This report investigates the task of Targeted Audio-Visual Speech Separation using the dataset provided by the instructors. We outline the problem setting, review relevant approaches, and present the full experimental methodology. The report describes the dataset, training process, the evaluation protocol, and the general structure of the processing pipeline. The results (presented in later sections) demonstrate noticeable differences in performance and training stability between the evaluated models, highlighting the importance of effective integration of audio-visual features. We conclude by discussing the strengths and limitations of the Targeted AVSS setting and outlining potential directions for further improvement.**

I. Introduction

Audio-visual speech separation (AVSS) seeks to isolate the speech of a target speaker from a mixture by using both acoustic and visual cues. In the Targeted AVSS setting, the model receives a visual reference of the target speaker whose speech is to be separated from the mixture. This formulation is especially relevant for applications such as video conferencing, assistive hearing systems, and multimodal human–computer interaction.

Recent deep-learning methods fuse time–frequency audio features with visual embeddings derived from facial or lip regions. Despite notable progress, these models often remain difficult to train, computationally costly, and sensitive to data variability, which motivates a systematic comparison of modern architectures within a unified framework.

In this work, we evaluate several contemporary models on the Targeted AVSS task using the proprietary dataset provided by the instructors, working in a data-limited setting where all models are trained entirely from scratch. Our goal is to assess their separation quality, training stability, and robustness in this data-limited setting.

The rest of the report is organized as follows: Section II reviews related work, Section III outlines the methodology, Section IV describes the experimental setup, and Section V presents the results. Section VI concludes the report, and Section VII details the individual contributions. Our implementation is publicly available online. [1].

II. Related Work

Research on speech separation began with audio-only models operating in the time or time–frequency domain. Architectures such as ConvTasNet [1] and DPRNN [2] demonstrated strong single-channel separation performance, yet remained inherently limited by the absence of visual information, particularly in heavily overlapping speech.

Audio-visual speech separation (AVSS) methods address this limitation by incorporating visual cues such as facial appearance or lip motion [3]. Typical approaches extract visual embeddings from the face or mouth region and fuse them with audio features through attention mechanisms or conditioning layers, leading to substantial improvements in target-speaker extraction.

Among recent AVSS models, Recurrent Time-Frequency Separation Network (RTFS-Net) [4] and Top-Down-Fusion Net (TDFNet) [5] offer two contrasting design philosophies. RTFS-Net employs recurrent modeling along both the time and frequency dimensions and integrates visual features through lightweight cross-modal attention. Notably, it is highly parameter-efficient—on the order of less than a million parameters—making it well suited for low memory storage needs. In contrast, TDFNet builds on a top-down fusion strategy that repeatedly injects visual information into the audio pathway, achieving strong separation performance at the cost of higher architectural complexity and computational load.

These methods represent modern baselines for Targeted AVSS and serve as useful reference points for evaluating model behavior in a data-limited, training-from-scratch setting.

III. Methodology

In the Targeted Audio-Visual Speech Separation task, the model receives a mixture waveform together with a visual reference of the target speaker. The objective is to reconstruct the speech signal corresponding to that speaker, while suppressing all interfering sources. Our overall methodology follows different pipelines while utilizing the same pretrained and frozen video encoder for visual embedding extraction. This encoder is based on the architecture described in the Lipreading using Temporal Convolutional

[1] https://github.com/eiky0u/AVSS-Project

Networks work [6], with its implementation and weights sourced from the corresponding public repository on GitHub. This backbone processes a reference mouth crop and outputs a fixed-dimensional embedding, and it deviates from both the original approaches as there were used different pretrained lipreading encoders.

A. RTFS-Net methodology

The RTFS-Net operates within the time-frequency domain to achieve efficient audio-visual speech separation. The pipeline initiates by converting the mixed audio signal into complex time-frequency bins via a Short-Time Fourier Transform (STFT) encoder, while visual lip-motion features are simultaneously extracted using a pretrained video encoder. Following initial refinement in independent processing blocks for both audio and visual features, the modalities are integrated via the Cross-dimensional Attention Fusion (CAF) block. This lightweight module fuses visual data into audio features through a dual mechanism: an attention fusion step that aggregates visual cues via multi-head attention, and a gated fusion step that expands visual information into the time-frequency domain using audio-generated gates.

The backbone of the separation process consists of stacked Recursive Time-Frequency Separation (RTFS) Blocks, which share parameters to minimize the model size. Each block employs a dual-path architecture that first compresses feature resolution and subsequently models the time and frequency dimensions independently using Simple Recurrent Units (SRUs) and time-frequency self-attention; however, we use simple bidirectional LSTM instead of it in our setup for better compatibility which is a key difference from the original setup. Crucially, to mitigate information loss during the up-sampling phase, the model incorporates Temporal-Frequency Attention Reconstruction (TF-AR) units. These units utilize attention mechanisms to prioritize the reconstruction of key features, effectively restoring time and frequency resolution without introducing the checkerboard artifacts typically associated with standard interpolation or transposed convolutions.

For the final extraction of the target speaker's speech, RTFS-Net utilizes Spectral Source Separation (S^3) block rather than traditional element-wise masking. Recognizing the complex nature of STFT features, the S^3 block applies a high-dimensional complex number multiplication strategy to the refined features. This approach explicitly reconstructs the real and imaginary components of the target audio, preserving critical phase and amplitude information that is often lost in conventional mask-based methods. The separated complex features are finally converted back into a time-domain waveform via a decoder and inverse STFT.

B. TDFNet methodology

The TDFNet operates directly on the raw waveform and adheres to the original design based on a learnable 1D convolutional encoder, a hierarchy of temporal convolutional blocks, and a symmetric decoder used to reconstruct the separated signal.

TDFNet conditions the audio pathway on a visual representation of the target speaker at multiple stages of the separation network. To address the dimensionality mismatch between this embedding from pretrained video encoder and the internal TDFNet feature size, we introduce an additional linear projection layer mapping the visual features into the latent audio feature space. The projected visual features are then employed for stage-wise conditioning through additive modulation or gating, following the original top-down fusion strategy of TDFNet.

A distinctive component of TDFNet is its Refinement Module, which performs iterative audio-visual refinement within a compact bottleneck space. Following the encoder, both audio and visual features are projected to reduced channel dimensions and passed through several refinement stages. In the early stages, the audio and visual branches are updated jointly: each refinement step applies separate temporal convolutional updates to the audio and visual streams and subsequently merges them via a lightweight cross-modal fusion block. This design enables the model to repeatedly exchange information between modalities rather than relying on a single fusion point.

In the later refinement stages, the module focuses primarily on the audio branch while keeping the visual representation fixed. This shift from fully audio-visual refinement to predominantly audio-only refinement reduces computational cost and encourages the network to utilize the visual signal mainly to disambiguate the target speaker, while the final separation details are handled within the audio pathway. As a result, the Refinement Module enables TDFNet to maintain strong audio-visual coupling where it is most beneficial, while remaining relatively efficient and stable to train in a data-limited setting.

No other architectural modifications were introduced.

IV. Experimental Setup

The proprietary dataset consists of paired audio-visual samples containing overlapping speech from two speakers. Each sample includes a single-channel mixed audio waveform and two corresponding video streams of unicolor cropped mouths for the speakers. The ground truth targets are the clean, isolated speech waveforms for Speaker 1 and Speaker 2, enabling supervised training for signal reconstruction.

A. RTFS-Net setup

Trained for 20 epochs on an NVIDIA L4 GPU, the model utilized a batch size of 4, 32 gradient accumulation steps (effective batch size of 128), and a Cosine Annealing Learning Rate scheduler. While our configuration largely follows the original baseline, specific hyperparameters were adjusted to balance computational efficiency with memory constraints:

- **Separation Strategy:** The model processes one 'mouth' at a single iteration and predicts a single waveform. Therefore, it is required to pass 2 iterations with 2 target mouth crops independently to fully separate a mixed audio to 2 speakers.
- **Optimization Strategy:** Following the RTFS-Net authors' approach, we used the AdamW optimizer with an initial learning rate of 1×10^{-3} and a weight decay of 1×10^{-1}.
- **Audio Encoder:** We retained the Short-Time Fourier Transform (STFT) parameters with $N_{fft} = 1024$, a hop length of 128, and a window length of 256. The audio-visual channel dimension was set to $C_{av} = 256$.
- **Compression Factor** (q)**:** To reduce Video RAM (VRAM) consumption, we increased the compression factor in the RTFS blocks to $q = 3$ (compared to $q = 1$ or 2 in the original study). This allows for stronger compression of time and frequency resolutions but reduces overall quality.
- **Recurrent Steps** (R)**:** The number of repetitive passes through the shared RTFS block was set to $R = 3$ to accelerate training and inference times and to lower the VRAM consumption again (in contrast, authors used $R = [4, 6, 12]$).
- **Fusion Mechanism:** In the Cross-modal Attention Fusion (CAF) block, we increased the number of attention heads to 8 to enhance the capture of dependencies between audio and visual modalities.

Additionally, for a specific comparative experiment regarding model scaling, we utilized an NVIDIA A100 GPU to perform fine-tuning with the number of recurrent steps increased to $R = 7$.

B. TDFNet setup

The TDFNet model was trained on NVIDIA L4 GPU for an initial period of 50 epochs using a batch size of 8 with gradient accumulation over 16 steps (effective batch size of 128).

Following the setup conventions, the specific training and architectural parameters were defined as follows:

- **Separation Strategy:** The model processes two 'mouths' at a single iteration and predicts two waveforms simultaneously, Therefore, it is required to pass only 1 iteration with 2 target mouth crops at the same time to fully separate a mix audio to 2 speakers.
- **Optimization Strategy:** We utilized the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and a weight decay of 0.1. The learning rate was managed via a cosine schedule: warming up over the first 5 epochs from 1×10^{-4} to a peak of 1×10^{-3}, followed by annealing down to 1×10^{-6}.
- **Architecture:** The base configuration consisted of 16 refinement modules, 3 of which functioned as fusion blocks for combining audio and visual information. This module count and fusion block ratio align with the specifications of the base TDFNet architecture as presented in the original publication.
- **Fine-tuning Configuration:** A secondary training stage was conducted on an NVIDIA P100 GPU. For this stage, the depth of the audio processing stack was increased from 16 to 22 refinement modules. The optimization parameters were adjusted to $\beta_2 = 0.999$ and weight decay 0.01. The learning rate started at 1×10^{-4}, peaked at 3×10^{-4} after 2 epochs, and annealed to 1×10^{-5} over 18 epochs.

V. Results

A. RTFS-Net Evaluation

1) Separation Performance: After training for 20 epochs, the RTFS-Net configuration ($R = 3$, $q = 3$) achieved a mean Scale-Invariant Signal-to-Noise Ratio (SI-SNR) of **8.258 dB** on the validation set.

We observed a significant variance in separation quality between the two speakers in the validation split. As detailed in Table I, while Speaker 1 achieved a higher absolute SI-SNR (11.07 dB), the model provided a greater relative improvement (SI-SNRi) for Speaker 2 (10.47 dB vs 6.05 dB). This behavior suggests dataset-intrinsic properties affecting the separation difficulty rather than model bias.

Table I
RTFS-Net Performance Metrics (Validation)

Metric	Speaker 1	Speaker 2	Average
SI-SNR (dB)	11.071	5.445	**8.258**
SI-SNRi (dB)	6.045	10.472	8.258
SDR (dB)	11.663	6.419	9.041
PESQ	2.041	1.405	1.723
STOI	0.900	0.789	0.844

2) Computational Efficiency and Resource Usage: Training was conducted on a single NVIDIA L4 GPU. Despite the batch size of 4, the model required **22.1 GB** of VRAM due to the unfolding operations in the time and frequency dimensions. The duration of a single training epoch was approximately 5 hours and 12 minutes.

Benchmark results for a batch size of 1 (for inference mode) are presented in Table III. The model utilizes only **0.76M trainable parameters** (excluding the frozen video frontend), resulting in a model size of approximately 8.89 MB, with a throughput of 1.13 examples per second.

3) Ablation Studies and Experiments: We conducted several experiments to analyze the model's behavior under different inference and training conditions:

- **Iterative Cleaning:** We attempted a multi-stage inference approach by feeding the separated audio output from the first iteration back into the model as input. Contrary to expectations, this degraded signal quality, introducing distortions into the already separated speech.
- **Inference-time Recurrence** (R) **scaling:** Since the RTFS blocks share weights, we increased the number of recurrent steps R during inference (without retraining) to test if deeper processing improved quality. This negatively impacted performance, as the model had learned feature distributions specific to $R = 3$.
- **Fine-tuning with** $R = 7$**:** We performed fine-tuning for 1 epoch with $R = 7$ on an NVIDIA A100 GPU. This configuration reduced epoch time to 4 hours and increased VRAM consumption to **37 GB**. The resulting SI-SNR improvement was negligible (8.25 → 8.27 dB), confirming that our lightweight configuration ($R = 3$) offers the optimal trade-off between resources and performance.

B. TDFNet Evaluation

1) Training Progression and Fine-tuning: The TDFNet model, processing both speakers simultaneously, demonstrated robust convergence. The base configuration with 16 refinement modules achieved an **SI-SNRi of 10.08 dB** at epoch 45. Extended training with reduced learning rates ($\beta_2 = 0.999$) yielded no further improvements.

However, fine-tuning by increasing the depth of the audio processing stack from 16 to 22 modules proved effective. After 20 epochs of fine-tuning on an NVIDIA P100 GPU, the performance improved to **10.25 dB SI-SNRi**.

2) Performance and Efficiency: The final metrics for the 22-block TDFNet are shown in Table II. TDFNet achieved a higher overall quality (10.25 dB SI-SNR) compared to RTFS-Net.

Table II
TDFNet (Depth 22) Performance Metrics

Metric	**Speaker 1**	**Speaker 2**	**Average**
SI-SNR (dB)	13.028	7.476	**10.252**
SI-SNRi (dB)	8.002	12.504	10.253
SDR (dB)	13.509	8.284	10.896
PESQ	2.230	1.516	1.873
STOI	0.923	0.830	0.877

Training was conducted in two stages. In the first stage, we trained the model with a batch size of 8, requiring **22.49 GB** of VRAM and yielding an epoch duration of approximately 1 hour and 8 minutes. In the second stage, we fine-tuned it with a batch size of 4, where the model used **15.72 GB** of VRAM and each training epoch took approximately 2 hours and 13 minutes.

As shown in Table III, both TDFNet variants outperform RTFS-Net in computational efficiency despite their larger parameter counts (8.93M vs 0.76M). TDF (16) leads with a throughput of **3.47 examples/s** and the lowest MACs, followed by TDFNet (22) at **2.66 examples/s**, whereas RTFS lags behind at 1.13 examples/s.

Table III
Computational benchmark comparison (Batch Size = 1).

Metric	**RTFS-Net** (3)	**TDFNet** (16)	**TDFNet** (22)
Trainable Params [M]	**0.76**	8.93	8.93
MACs [G]	125.4	**81.8**	94.3
Step Time [s]	0.89	**0.29**	0.38
Throughput [ex/s]	1.13	**3.47**	2.66
Max VRAM [MB]	1101	**434**	**434**

VI. Conclusion

In this report, we conducted a systematic comparative analysis of two contemporary Audio-Visual Speech Separation (AVSS) architectures, RTFS-Net and TDFNet, within the challenging context of a data-limited, training-from-scratch scenario for the Targeted AVSS task. Our primary objective was to evaluate their separation quality, training stability, and resource efficiency under these constraints.

The experimental results definitively demonstrated the superior performance of the TDFNet architecture, which achieved a final Scale-Invariant Signal-to-Noise Ratio (SI-SNR) of **10.252 dB** after a fine-tuning stage. This performance substantially exceeded the **8.258 dB** SI-SNR attained by the highly parameter-efficient RTFS-Net. Furthermore, the analysis highlighted TDFNet's architectural advantages in computational efficiency: despite a larger parameter count (8.93M vs 0.76M), TDFNet achieved higher throughput (up to 3.47 examples/s) and dramatically lower VRAM consumption during inference.

Our findings suggest that the TDFNet's architecture design is better suited for the data-limited, training-from-scratch environment. In contrast, the RTFS-Net's reliance on complex time-frequency representations and recurrent unfolding proved highly VRAM-intensive, severely limiting our ability to utilize its intended deep recurrence (R) structure for quality improvement. Attempts to increase recurrence during inference or fine-tuning yielded negligible gains, confirming that the lightweight configuration ($R = 3$) was the optimal resource-performance trade-off for the available computational budget.

A key limitation of this study was the inability to unlock the full potential of the RTFS-Net due to the memory cost of its complex feature space and inability to set lower compression factor q and higher R in contrast to the original work. Future work must critically investigate the generalization capabilities of SOTA-positioned models, particularly under data-scarce or domain-shifted conditions.

VII. Contributions

The responsibilities were divided as follows:

Nikolai Prudnikov was responsible for TDFNet architecture. He established the initial training pipeline, implemented the evaluation metrics script, and conducted the computational speed benchmarks. Additionally, he prepared the demonstration notebook for inference.

Ilyas Khassenov was responsible for the RTFS-Net architecture and the integration of the pretrained video backbone (visual encoder). He also played the lead role in writing, structuring, and editing this report.

References

[1] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[2] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP 2020 – 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 46–50.

[3] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, "Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 112:1–112:11, 2018.

[4] S. Pegg, K. Li, and X. Hu, "RTFS-Net: Recurrent time–frequency modelling for efficient audio-visual speech separation," *arXiv preprint arXiv:2309.17189*, 2023.

[5] ——, "TDFNet: An efficient audio-visual speech separation model with top-down fusion," *arXiv preprint arXiv:2401.14185*, 2024.

[6] P. Ma, B. Martinez, Y. Wang, S. Petridis, J. Shen, and M. Pantic, "Lipreading using temporal convolutional networks," in *ICASSP 2020 – 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6319–6323.