Speech Signal Disentanglement Using Kolmogorov-Arnold Networks

Solving the Cocktail Party Problem

Abhishek Chaudhary [*]
Columbia University
ac5003@columbia.edu
Sungjoon Park[*]
Columbia University
sp4050@columbia.edu

May 12, 2025

1 Abstract

We investigate the use of Kolmogorov-Arnold Networks (KANs) for speech separation under extreme resource constraints. KANs, which replace traditional activations with learnable univariate splines, offer promising benefits in interpretability and parameter efficiency. We propose a family of hybrid models that integrate KAN-based modules into the ConvTasNet architecture and evaluate them on the DiPCo dataset. All models contain fewer than 6,000 parameters and require under 0.03 MB of memory. Despite this, KAN-based variants approach the performance of baseline ConvTasNet, particularly in highly overlapped speech segments. Our results demonstrate that compact, interpretable architectures can remain competitive for time-domain source separation.

2 Introduction

The Cocktail Party Problem in Machine Learning refers to the task of separating individual source signals from a mixture of signals. The central challenge of this task involves focusing on and isolating a single voice (output) from the mixed signal of all the voices (input), when the mixing process – how each voice contributes to the overall mixture – is unknown. Mathematically, we can express this problem as:

$$\mathbf{x}(\mathbf{t}) = A\mathbf{s}(\mathbf{t})$$

[*] Both authors contributed equally to this work.

Where:

- $\mathbf{x(t)} \in R^n$: (observed) mixture signal at time t
- $\mathbf{s(t)} \in R^m$: (unknown) vectorized source signals at time t
- $A \in R^{n \times m}$: (unknown) mixing matrix

The cocktail party problem is relevant to tasks like speech recognition, transcription, and music source separation. As a result, there has been continued progress toward developing more effective models to address this challenge.

While convolutional models like ConvTasNet have demonstrated strong performance in time-domain speech separation, their internal representations—based on fixed kernel convolutions—can be difficult to interpret and may require careful tuning to balance expressiveness with efficiency. In contrast, Kolmogorov-Arnold Networks (KANs) offer a promising alternative: they are interpretable, parameter-efficient, and theoretically grounded in function decomposition. These properties make KANs an attractive candidate for replacing convolutional components in time-domain architectures like ConvTasNet, particularly for resource-constrained and interpretable speech separation.

In this work, we investigate the application of KAN to the speech separation problem, comparing their effectiveness to established convolutional approaches under strict computational constraints. Our contributions are:

- A systematic evaluation of pure KAN and hybrid convolutional-KAN architectures for speech separation.
- An empirical comparison with compact ConvTasNet baselines on the DiPCo corpus.
- Analysis of the limitations and potential of KANs for interpretable, efficient source separation.

3 Related Works

3.1 From Spectrograms to Waveforms: The ConvTasNet Approach

Historically, supervised speech separation methods have operated in the time-frequency (T-F) domain, using spectrograms derived from short-time Fourier transforms (STFT) as input features. This approach facilitates masking-based or mapping-based learning, where neural networks predict clean spectrograms or estimate ideal masks to isolate speech from mixtures. Wang and Chen's 2018 review highlights how early deep learning methods relied heavily on such spectrogram-based representations, which remain dominant due to their structured frequency content and compatibility with human auditory perception models. [1]

However, STFT-based systems introduce key limitations—including challenges in phase reconstruction, suboptimal representations for speech-specific features, and latency induced by long analysis windows. These drawbacks have motivated time-domain models that learn directly from raw waveforms. Luo and Mesgarani's ConvTasNet exemplifies this shift: a fully convolutional network that avoids spectrogram computation altogether by replacing the STFT with a learnable encoder-decoder architecture and using temporal convolutions for mask estimation. [2]

ConvTasNet consists of three core components: an encoder, a masking network, and a decoder. The encoder uses a 1-D convolutional layer to transform raw waveforms into a non-negative latent representation, effectively learning a data-driven filterbank. The masking network applies a series of temporal convolutional network (TCN) blocks with dilated, depthwise separable convolutions and residual connections to estimate speaker-specific masks in the latent space. These masks are then element-wise multiplied with the encoder output to isolate individual sources. Finally, the decoder reconstructs the time-domain waveforms using a transposed convolution that maps the masked features back to audio. This fully time-domain pipeline enables low-latency, end-to-end learning without requiring explicit phase reconstruction, outperforming traditional T-F approaches in many benchmarks. ConvTasNet's modular structure and interpretability of its latent features have made it a popular baseline for exploring alternative components in speech separation architectures, including the replacement of convolutions with more flexible representations like KANs.

3.2 The Kolmogorov-Arnold Network (KAN)

The Kolmogorov-Arnold representation theorem states that **any multivariate continuous function** can be expressed as a **finite sum of compositions of univariate continuous functions**, using addition as the only multivariate operation [3].

Building on this principle, Liu et al. (2024) introduced KAN as an alternative to multilayer perceptrons (MLPs), replacing linear weights with learnable univariate splines and using additive compositions in place of traditional nonlinear activations like ReLU. This yields models that are both more interpretable and parameter-efficient than MLPs. An illustration of this architecture is shown below in Figure 1.

KANs are especially well-suited to tasks like speech separation, where mixed audio signals are formed through additive interactions. Beyond performance, KANs offer the potential for symbolic inspection of their learned functions—making it possible to extract interpretable components that may correspond to individual audio sources.

Building on the principles of ConvTasNet, we adopt waveform-level training (see top panel of Figure 2) and explore whether KANs can serve as a compute-efficient and interpretable replacement for convolutional modules in time-domain speech separation.

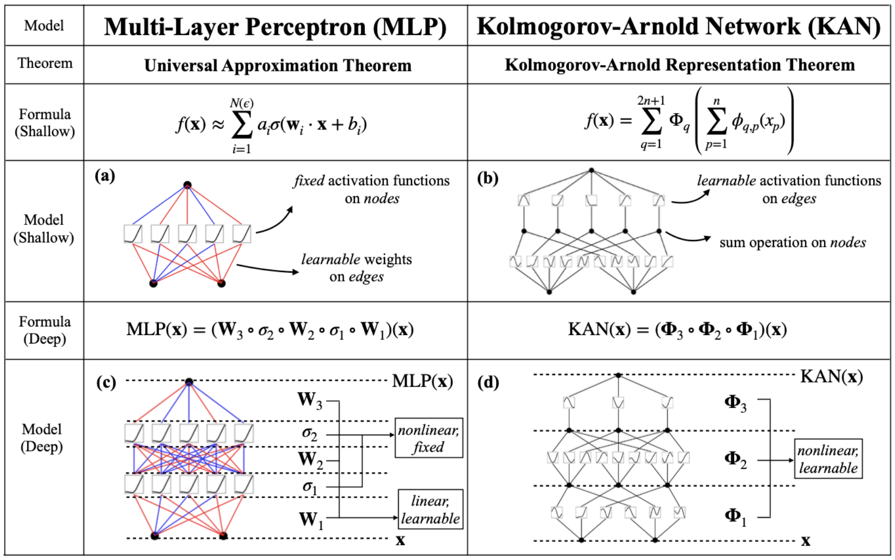

Model
Multi-Layer Perceptron (MLP)
Kolmogorov-Arnold Network (KAN)
Theorem
Universal Approximation Theorem
Kolmogorov-Arnold Representation Theorem
Formula (Shallow)
$f(\mathbf{x}) \approx \sum_{i=1}^{N(\epsilon)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$
$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$
Model (Shallow)
(a)
fixed activation functions on *nodes*
learnable weights on *edges*
(b)
learnable activation functions on *edges*
sum operation on *nodes*
Formula (Deep)
$\mathrm{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$
$\mathrm{KAN}(\mathbf{x}) = (\mathbf{\Phi}_3 \circ \mathbf{\Phi}_2 \circ \mathbf{\Phi}_1)(\mathbf{x})$
Model (Deep)
(c)
MLP(**x**)
$\mathbf{W}_3$
σ_2
$\mathbf{W}_2$
σ_1
$\mathbf{W}_1$
x
nonlinear, fixed
linear, learnable
(d)
KAN(**x**)
$\mathbf{\Phi}_3$
$\mathbf{\Phi}_2$
$\mathbf{\Phi}_1$
x
nonlinear, learnable

Figure 1: Multi-Layer Perceptrons (MLPs) vs. Kolmogorov-Arnold Networks (KANs) [4]

4 Methodology

Our code is publicly available at [1].

4.1 Dataset

We evaluated our system using two-speaker speech separation tasks constructed from the Dinner Party Corpus (DiPCo) dataset. Designed for benchmarking noise-robust and distant speech separation systems, each session of DiPCo was recorded using a single-channel close-talk microphone and five far-field 7-microphone arrays placed at various room locations. The dataset includes 10 sessions in total, each ranging from 15 to 45 minutes, with human-annotated transcripts available. [5]

For training, we used sessions S02 and S04, generating two-speaker mixtures by randomly pairing single-speaker segments and mixing them at a sample rate (SR) of 8 kHz. Each mixture was four seconds long, and up to 1 million pairs were created for a total of more than 1,111 hours of training data possible (refer to Table S1). For testing and validation, we used sessions S01 and S03, creating 100,000 mixtures totaling over 111 hours of data (Table S2). All waveform loading and preprocessing operations were performed using the `torchaudio` library [6].

[1] `https://github.com/A-Chaudhary/kan_cocktail_party`

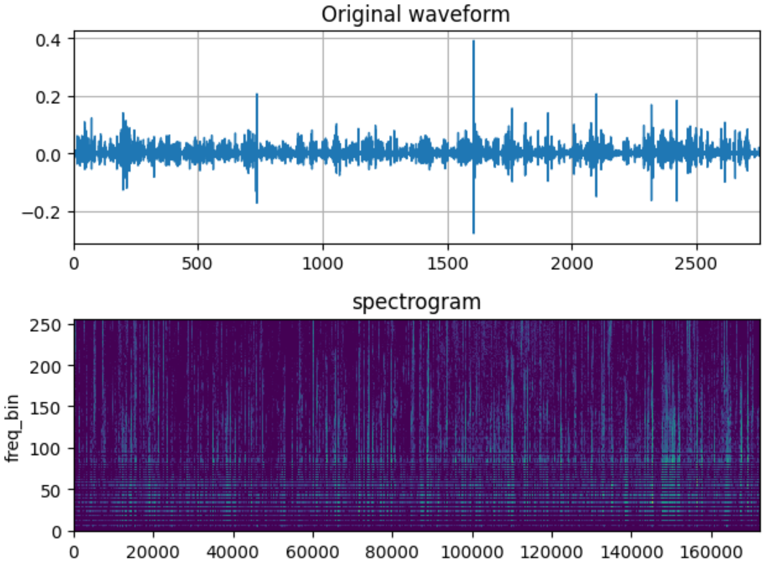


Figure 2: Audio Waveform and corresponding Mel-Spectrogram for Session 04 and Speaker 15 (S05, P1)

Each pair was constructed by selecting four-second segments where both speakers were actively speaking, based on session transcripts. The segments, labeled as clean1 and clean2, were mixed and clamped within the waveform range of $[-1, 1]$ to simulate overlapping speech. If an utterance was shorter than four seconds, padding was applied at the end—or before, if it occurred at the end of the audio track—to meet the time constraint.

4.2 Modeling Considerations

- **Compute Efficiency:** Given the constraint of a single physical GPU and a limited compute window of several days, we restricted model complexity and batch size to ensure tractable training times.
- **Model Simplicity:** To allow for rapid experimentation and debugging within the short project timeline, we favored architectures that are simple to implement and modify, ideally using established modules from existing libraries. Our implementation of KANConvTasNet builds on the `torchconvkan` framework [7], which provides efficient spline evaluation for univariate function composition.
- **Training Stability:** With limited time for hyperparameter tuning, we prioritized models and optimization techniques known for stable convergence with default settings, reducing the risk of stalled training or divergence.

Model	Encoder Kernel Size	Encoder # Features	Mask Kernel Size	Mask # Features	Mask # Layers	Mask # Stacks	Encoder Grid Size	Mask Grid Size
ConvTasNet	16	16	3	8	16	3	-	-
KANConvTasNet	7	7	3	3	5	3	4	4
FastKANConvTasNet	8	7	3	3	5	3	6	6
BottleneckKANConvTasNet	16	11	3	6	4	3	-	-

Table 1: Summary of key hyperparameters for each model variant.

Model hyperparameters, such as the number of layers, kernel sizes, and latent dimensions, were selected based on a balance between computational feasibility and representational capacity. Due to limited compute, we prioritized shallow networks with minimal parameter counts trying to match the number of parameter to the encoder, masking, and decoder of ConvTasNet for all ConvKan-based models.

4.3 Comparison Between Model Variants

To explore resource-efficient architectures, we implemented and evaluated two lightweight Conv-TasNet variants: a convolutional baseline (ConvTasNet) and three model variants incorporating Kolmogorov-Arnold Network convolutional layers (KANConvTasNet, FastKANConvTasNet, and BottleneckKANConvTasNet) on audio waveforms.

All models used under 6,000 parameters and < 0.03 MB memory (Table S3). Despite their compactness, they demonstrated strong speech separation capabilities, with the KAN-based model producing smoother and more temporally coherent output waveforms. Training for both models was completed within a constrained 3-week period on a single GPU, highlighting the feasibility of compact architectures for speech separation in low-resource environments.

4.4 Training Configurations

Training was conducted on a single GPU using randomly mixed four-second utterance segments from the DiPCo dataset. We used a batch size of 2 across all models—a constraint driven by the high memory overhead of spline-based layers in KANConvTasNet. KANs require more memory than convolutions due to spline backpropagation, especially in deep masks. This limitation made larger batch sizes infeasible without encountering out-of-memory errors. To ensure fairness, we adopted the same batch size across all model variants, including ConvTasNet.

Rather than a fixed number of epochs, we used early stopping based on validation loss: training terminated when the validation loss did not improve for three consecutive checkpoints. This resulted in approximately 24,000 training steps for KANConvTasNet, 85,000 training steps for FastKANConvTasNet, 95,000 training steps for BottleneckKANConvTasNet, and 17,000 steps for the baseline ConvTasNet. We used the Adam optimizer with an initial learning rate of 1×10^{-3}, halved if the validation loss plateaued for three evaluations. Gradient clipping with an L_2-norm threshold of 5 was applied throughout training.

4.5 Training Objective

We trained all models using a permutation-invariant training (PIT) objective based on the scale-invariant signal-to-noise ratio (SI-SNR). For each training step, SI-SNR was computed for both permutations of the estimated sources and ground truth references, and the permutation yielding the higher SI-SNR was used to compute the final loss. This objective effectively addresses the label ambiguity problem inherent in speech separation tasks involving multiple speakers.

4.6 Evaluation Metrics

Model performance was assessed using two standard metrics: scale-invariant signal-to-noise ratio improvement (SI-SNRi) and signal-to-distortion ratio improvement (SDRi). These metrics quantify the improvement of the separated sources over the original mixtures and are widely used to benchmark source separation systems. We computed SI-SNRi as the difference in SI-SNR between the separated outputs and the input mixture, following conventions from Luo & Mesgarani 2019. [2]

5 Results

5.1 Training and Validation Loss Landscape

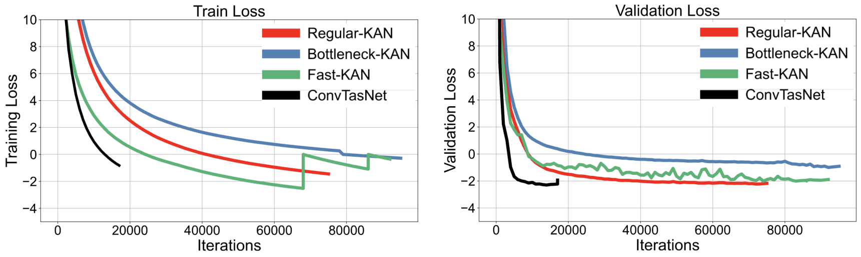


Figure 3: Training and validation loss as a function of training iteration for all models (ConvTasNet, KANConvTasNet, FastKANConvTasNet, Bottleneck-KANConvTasNet).

Figure 3 compares training and validation loss trajectories across four model variants. ConvTasNet, the baseline model (black), converges rapidly and consistently outperforms other models across both training and validation, aligning with its top SI-SNRi score of 2.259 dB in Table 2. The KANConvTasNet (red) closely follows, achieving near-parity in loss curves and a comparable SI-SNRi of 2.223 dB. This suggests that KANs can serve as viable substitutes for convolutional blocks even under severe parameter and compute constraints.

FastKANConvTasNet (green) initially descends quickly but later exhibits instability in the training loss and greater noise in validation. Its SI-SNRi of 2.004 dB corroborates this: the model performs reasonably well but shows signs of under-regularization or insufficient depth. BottleneckKANConvTasNet (blue) exhibits the slowest convergence and highest loss, suggesting that its compact encoder and deeper stack configuration limit its representational capacity.

Importantly, all KAN variants eventually achieve a below-zero validation loss, indicating successful signal reconstruction and overfitting avoidance. These findings show that while ConvTasNet remains strong in low-resource settings, KAN-based models offer competitive alternatives with potential gains in interpretability and symbolic inspection.

5.2 Performance on Overlapping Speech Segments

Model	|SI-SNRi| > 0.015 (dB)	|SI-SNRi| ≤ 0.015 (dB)	SI-SNRi (dB)
ConvTasNet	15.7520	1.9277	2.2590
KANConvTasNet	17.5300	1.7366	2.2233
FastKANConvTasNet	14.8953	1.8551	2.0044
BottleneckKANConvTasNet	18.5887	0.8090	0.9103

Table 2: Performance breakdown on DiPCo validation set by overlap difficulty. Higher values in the left column indicate better performance in heavily overlapped segments.

Table 2 reveals that KAN-based architectures, particularly KANConvTasNet and BottleneckKANConvTasNet, outperform ConvTasNet in heavily overlapped speech cases (SI-SNRi > 0.015 dB). This suggests that the additive functional representation used by KANs is well-suited for disentangling complex acoustic mixtures.

While ConvTasNet achieves the highest overall SI-SNRi, KANConvTasNet approaches it closely. BottleneckKANConvTasNet, though weaker overall, demonstrates the highest skill on overlapped segments—hinting at a possible tradeoff between generality and peak separation strength.

5.3 Qualitative Waveform Comparison

Figure S1 provides a qualitative comparison of model outputs against ground truth waveforms for a representative sample. ConvTasNet (a) and KANConvTasNet (b) closely match the true waveform structure, preserving both onset timing and amplitude envelope. KANConvTasNet in particular exhibits slightly more temporal detail and waveform sharpness—despite using spline-based function modules instead of convolution.

FastKANConvTasNet (d) and BottleneckKANConvTasNet (e) show degraded fidelity. FastKANConvTasNet captures the broad temporal structure but introduces spikier energy artifacts, while BottleneckKANConvTasNet underestimates low-energy regions and over-smooths transitions. These distortions align with their lower SI-SNRi scores.

Together, these waveform comparisons support the conclusion that spline-based architectures like KANConvTasNet can recover audio sources with clarity comparable to state-of-the-art convolutional models, while offering interpretability and compactness advantages.

6 Discussion

6.1 Overview of Model Perfromance

Our results highlight distinct performance patterns among the four model variants, each reflecting underlying architectural tradeoffs. ConvTasNet remains the strongest overall in terms of convergence speed and final SI-SNRi, benefiting from well-established convolutional design and optimized masking modules.

KANConvTasNet, despite its novel spline-based formulation and compact parameter count, performs comparably—indicating that KAN layers can serve as viable replacements for convolutional blocks, even in resource-limited settings. The model captures sharp waveform transitions and performs especially well in overlapped speech scenarios, as shown in Table 2 and Figure S1.

FastKANConvTasNet demonstrates promising early convergence but suffers from instability, likely due to insufficient network depth or regularization to balance its accelerated architecture.

BottleneckKANConvTasNet exhibits the weakest overall performance, with reduced representational capacity stemming from its narrowed encoder and deeper masking stack. Interestingly, it achieves the best score on heavily overlapped segments, suggesting a tradeoff between general performance and specialization under difficult conditions. Similar architectural ideas have been explored in recent convolutional-KAN hybrids for audio processing [8], which emphasize the tradeoff between compression and depth.

6.2 Computational Constraints and Model Size Limitations

Our current implementation of KANConvTasNet was significantly constrained by available computational resources, leading to architectural decisions that prioritized efficiency over capacity. As noted in our experimental configurations, all models were trained on a single GPU with a minimal batch size of 2, which particularly limited the KANConvTasNet variant. All models were extremely compact—under 6,000 parameters and 0.03 MB total memory.

The original Conv-TasNet architecture typically employs substantially larger parameter counts, with the baseline implementation using 512 filters in the autoencoder (N), 128 channels in the bottleneck (B), 512 channels in convolutional blocks (H), and multiple stacked dilated convolutional blocks. Our implementation necessarily reduced these dimensions to accommodate our computing constraints, potentially limiting the model's capacity to capture complex audio separation patterns.

While our compact models demonstrated reasonable separation capabilities given their size constraints, they likely represent suboptimal configurations for the KANConvTasNet architecture. The introduction of learnable spline-based activation functions through the Kolmogorov-Arnold framework theoretically enhances representational capacity per parameter, but these benefits may not be fully realized without sufficient network depth and width. Recent work by Drokin et al. [9] explores convolutional design principles for KANs and highlights how deeper architectures can significantly improve performance—suggesting that, if computationally feasible, further gains could be achieved through expanded KANConvTasNet variants.

6.3 Maximizing Value from the DiPCo Corpus

We trained on voice mixtures from sessions S02 and S04; S01 and S03 were reserved for validation. A scaled model could incorporate data from all ten available sessions, significantly increasing the diversity of speakers and contexts seen during training. Scaling to the full corpus would offer a richer training distribution and likely improve generalization.

Additionally, our synthetic mixtures contained fully overlapped speech segments—an aggressive setting that does not reflect natural turn-taking patterns. Future work could introduce mixtures with variable overlap ratios and optimize for weighted SI-SNR to better capture real conversational dynamics.

DiPCo's multi-microphone, four-speaker setup also opens the door to more ambitious tasks, such as multi-speaker (3+) disentanglement and far-field separation. Larger KANConvTasNet variants may be capable of tackling these scenarios with greater flexibility than traditional convolutional models.

References

[1] DeLiang Wang and Jitong Chen. "Supervised Speech Separation Based on Deep Learning: An Overview". In: *arXiv preprint arXiv:1708.07524* (2018). DOI: 10.48550/arXiv.1708.07524. URL: https://arxiv.org/abs/1708.07524.

[2] Yi Luo and Nima Mesgarani. "Conv-TasNet: Surpassing Ideal Time–Frequency Magnitude Masking for Speech Separation". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.8 (Aug. 2019), pp. 1256–1266. ISSN: 2329-9304. DOI: 10.1109/taslp.2019.2915167. URL: http://dx.doi.org/10.1109/TASLP.2019.2915167.

[3] A. Kolmogorov. "On the Representation of Continuous Functions of Several Variables by Superposition of Continuous Functions of One Variable and Addition". In: *Doklady Akademii Nauk SSSR* 114 (1957). Also known as Kolmogorov's Superposition Theorem, pp. 369–373.

[4] Ziming Liu et al. *KAN: Kolmogorov-Arnold Networks.* 2024. arXiv: 2404.19756 [cs.LG].

[5] Maarten Van Segbroeck et al. *DiPCo – Dinner Party Corpus.* https://catalog.ldc.upenn.edu/LDC2021S09. Speech corpus simulating natural conversations in home environments with close-talk and far-field microphone recordings. 2021.

[6] PyTorch Contributors. *Conv-TasNet Implementation in TorchAudio.* 2023. URL: https://github.com/pytorch/audio.

[7] Ivan Drokin. *Torch-Conv-KAN: Convolutional Kolmogorov-Arnold Networks.* 2024. URL: https://github.com/IvanDrokin/torch-conv-kan.

[8] Alexander Dylan Bodner et al. *Convolutional Kolmogorov-Arnold Networks.* 2025. arXiv: 2406.13155 [cs.CV]. URL: https://arxiv.org/abs/2406.13155.

[9] Ivan Drokin. *Kolmogorov-Arnold Convolutions: Design Principles and Empirical Studies.* 2024. arXiv: 2407.01092 [cs.CV]. URL: https://arxiv.org/abs/2407.01092.

Supplemental Material

Session	**Person**	**Audio Samples (SR = 8 KHz)**	**# 4 Second Clips**
S02	P05	14404750	200
S02	P06	14404750	175
S02	P07	14404750	170
S02	P08	14404750	137
S04	P13	22042084	375
S04	P14	22042084	391
S04	P15	22042084	491
S04	P16	22042084	457

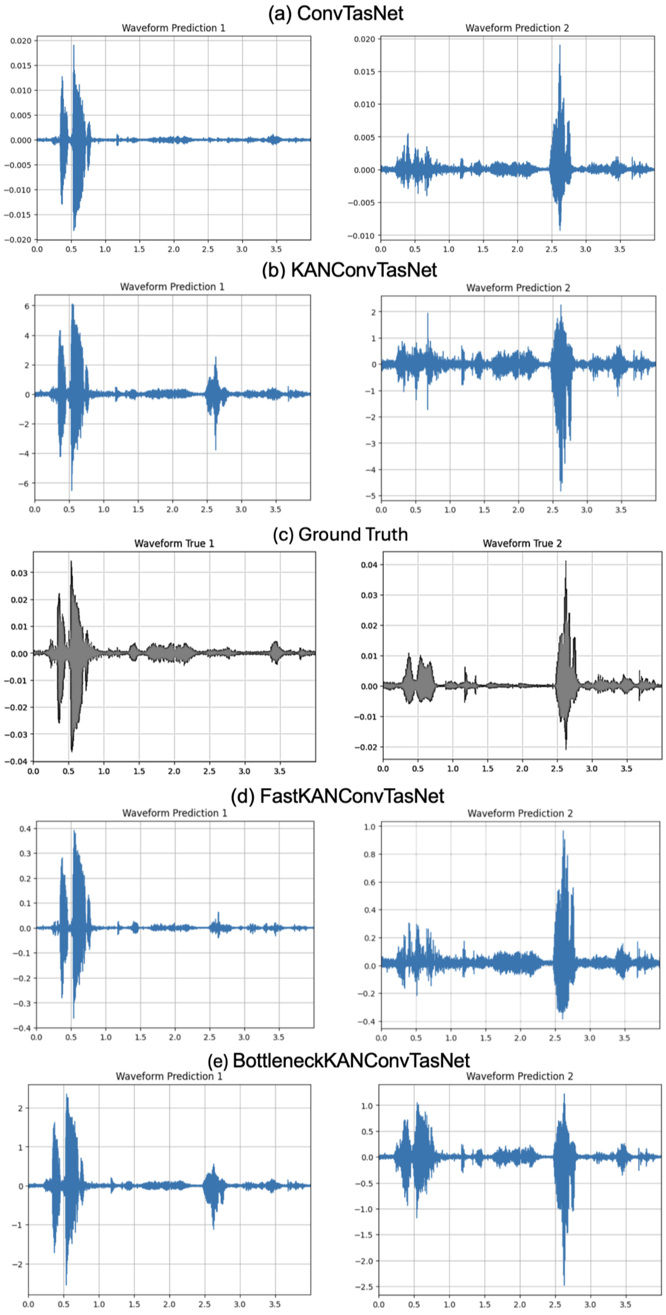
Table S1: DipCo Training Data: Session-Person number of extracted four second audio clips

Session	**Person**	**Audio Samples (SR = 8 KHz)**	**# 4 Second Clips**
S01	P01	22759000	357
S01	P02	22759000	407
S01	P03	22759000	124
S01	P04	22759000	441
S03	P09	22369000	456
S03	P10	22369000	520
S03	P11	22369000	342
S03	P12	22369000	315

Table S2: DipCo Validation Data: Session-Person number of extracted four second audio clips

Model	Encoder Size (MB)	Mask Size (MB)	Decoder Size (MB)	Full Model Size (MB)
ConvTasNet	0.00098	0.01997	0.00098	0.02192
KANConvTasNet	0.00150	0.02017	0.00019	0.02185
FastKANConvTasNet	0.00152	0.01999	0.00021	0.02172
BottleneckKANConvTasNet	0.00098	0.01918	0.00067	0.02083

Table S3: Memory sizes (in MB) for Encoder, Mask, Decoder, and Full Model for each model variant.


(a) ConvTasNet

(b) KANConvTasNet

(c) Ground Truth

(d) FastKANConvTasNet

(e) BottleneckKANConvTasNet

Figure S1: Ground truth waveforms (c) versus predictions from models (a: ConvTasNet, b: KANConvTasNet, d: FastKANConvTasNet-Fast, e: BottleneckKANConvTasNet). Visual similarity between rows indicates model skill in recovering the original signals from the mixed signal.

Appendix: Mel-Spectrogram Experiments

This appendix presents preliminary experiments using mel-spectrogram input representations instead of raw audio waveforms. We explored both pure KAN models and hybrid convolutional-KAN architectures to assess their ability to model frequency-domain structure.

A.1 Pure KAN on Mel-Spectrograms

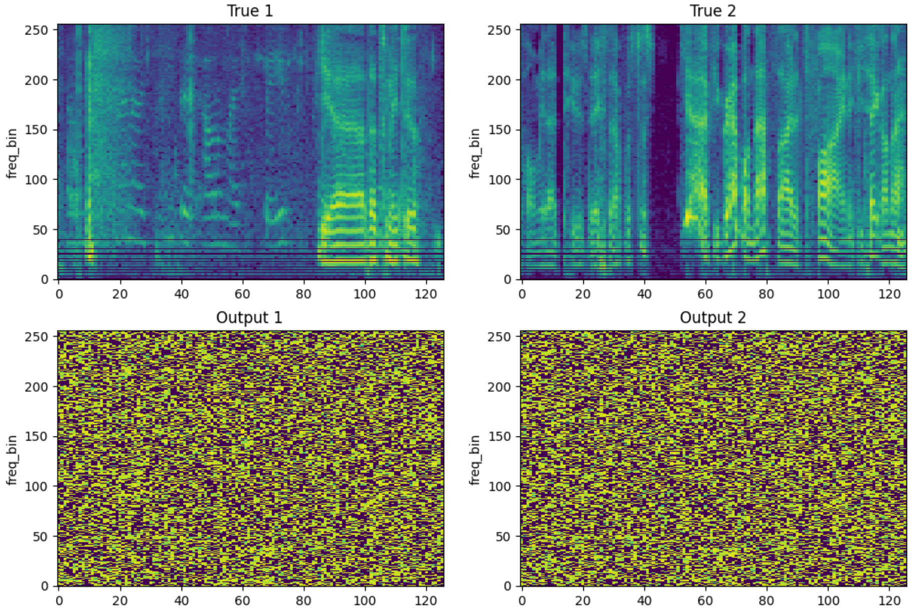


Figure A1: Predictions from a pure KAN encoder-decoder trained on mel-spectrograms. Top: ground truth spectrograms (True 1, True 2); Bottom: model predictions (Output 1, Output 2).

The pure KAN architecture fails to capture meaningful temporal or frequency structure, producing outputs that resemble colored noise. Despite KAN's universal function approximation property, direct optimization over spectrogram inputs proves unstable, likely due to the high dimensionality and fine-grained variation of mel-spectrograms.

A.2 Hybrid Convolutional-KAN Latent Space

Adding convolutional components to the encoder and decoder improves learning stability and introduces structure into the predictions. While temporal dynamics are still weak, horizontal banding indicates the model captures coarse spectral properties. This supports the idea that convolutional inductive bias in frequency modeling complements KAN's latent flexibility.

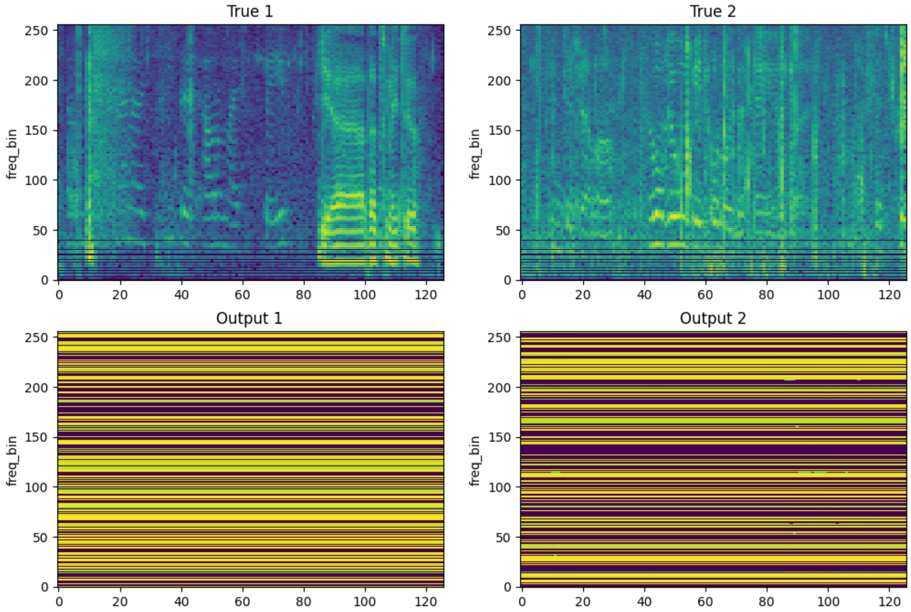


Figure A2: Predictions from a hybrid model using convolutional encoders and decoders with a KAN latent space. Top: ground truth spectrograms; Bottom: predicted outputs.

A.3 Summary of Mel-Spectrogram Findings

Our early experiments show that KANs struggle to model mel-spectrograms directly, especially in encoder-decoder form. Introducing convolutional layers at the input/output interfaces helps, but performance still lags behind waveform-based models. These results motivated our decision to switch to raw waveform inputs and adopt ConvTasNet-style architectures for better convergence and reconstruction fidelity.