MULTI-DECODER DPRNN: SOURCE SEPARATION FOR VARIABLE NUMBER OF SPEAKERS

Junzhe Zhu, Raymond A. Yeh, Mark Hasegawa-Johnson

Separation with variable number of speakers

- Source separation: Given an audio mixture $X = \sum S_1, S_2, S_3, \ldots,$ find S_1, S_2, S_3
- Dataset: a possible range of number of speakers (we set it to 2-5)
- Challenge: How do we use a single neural network to output a variable-sized tensor containing the estimated source signals?

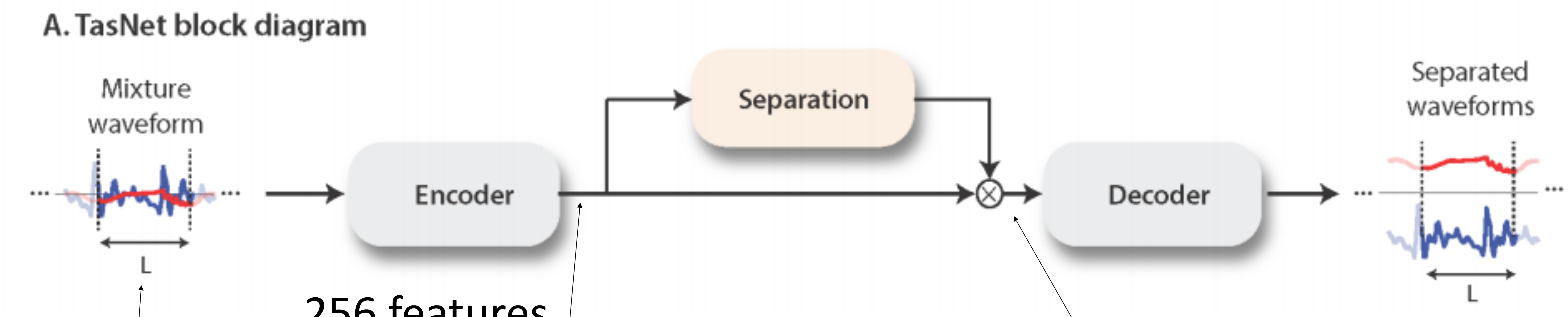
Challenges of Existing methods

- OR-PIT: Separate one source at a time, combine with permeation-invariant training(PIT)
- Nachmani et al proposed to train a model for each dataset with a fixed number of speakers.
- Luo et al. proposed to use a model with a high number of output channels, and discard extra channels
- The first two methods perform better, but runtime scale up with the number of sources. The third method has low SNR when number of sources goes up

Problem formulation

- We divide the problem into two parts - 1. Source counting(determining number of speakers) 2. Given the source count, estimate the source signals)
- We assume that if we use a sequence neural network(e.g. LSTM) to process a mixture signal, the output distribution will change depending on the number of ground-truth sources.
- Question: how do we make a single neural network adapt to a variable number of sources

Backbone Architecture: TasNet



1 x 16000

256 features,
Kernel=8,
Skip=4

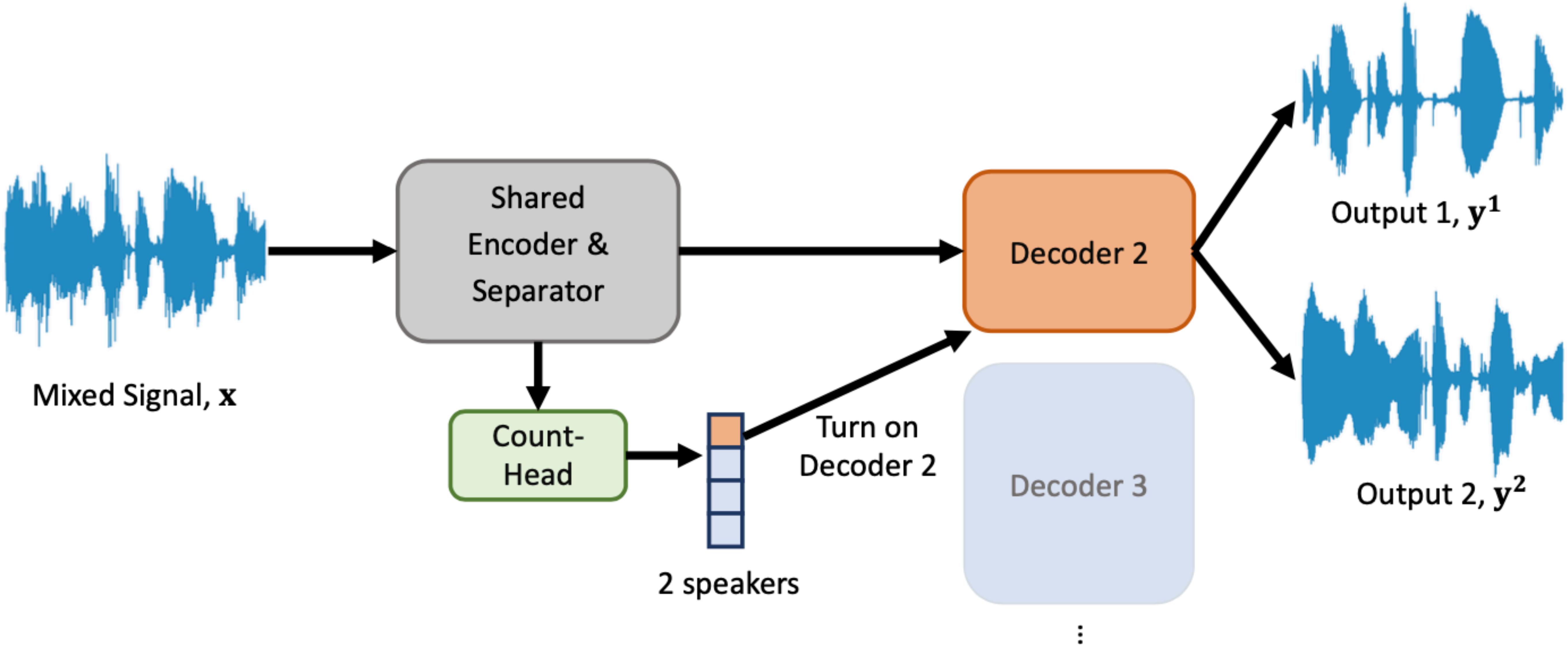
256 x 3999

Projection layer: N –> (N x num_spks)

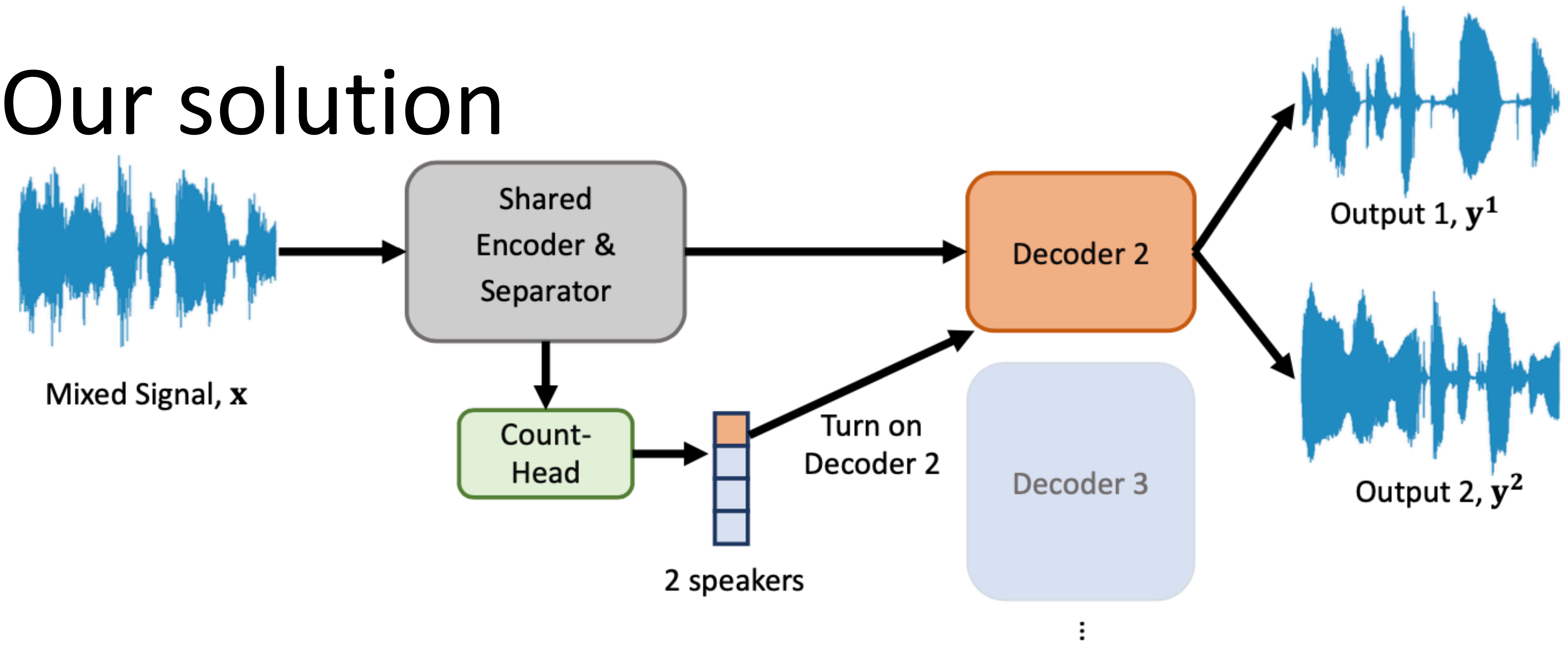
Instead of having one decoder, we used a number of decoders, each with a different num_spks

Our solution

- Use the same encoder & backbone network, but a different decoder for each number of sources!



Our solution

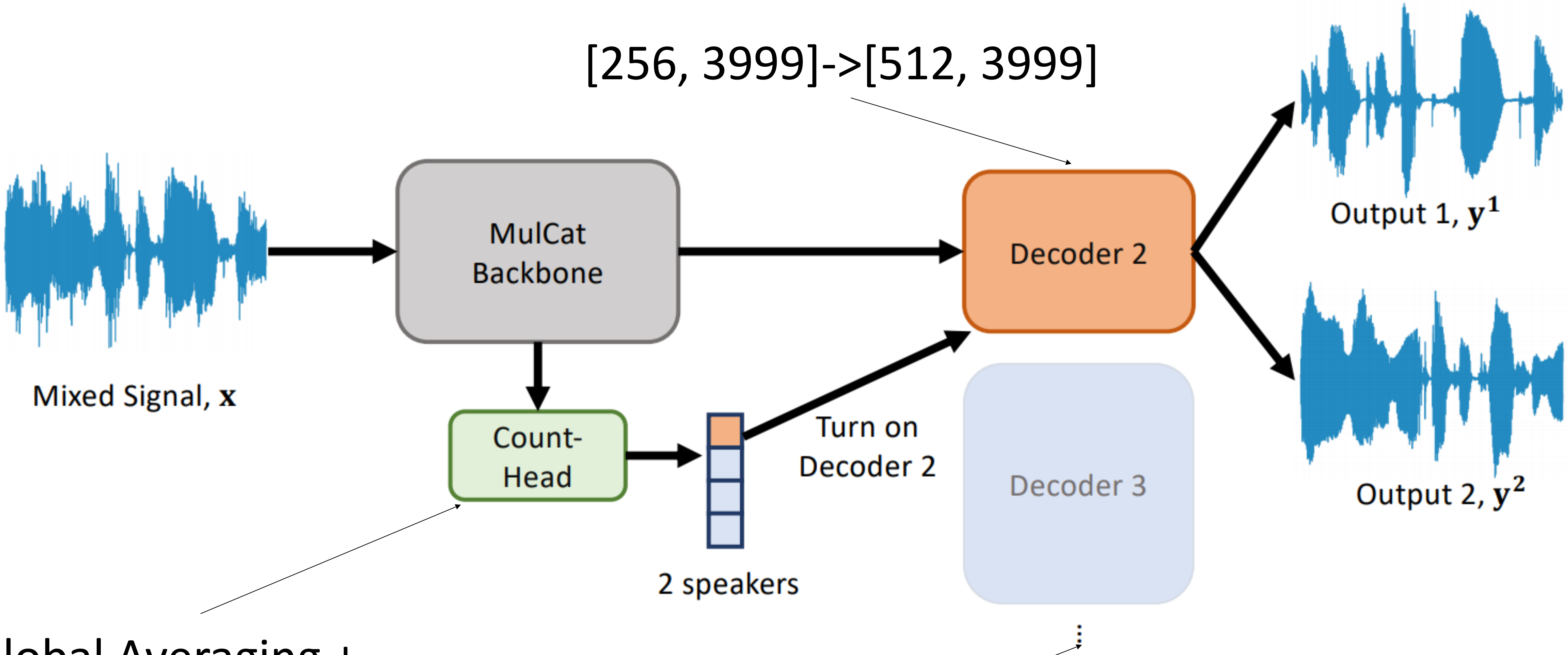


- We always use the same encoder & backbone, shared by all decoders
- Each decoder takes the same input, but has a different number of output channels (achieved by a different projection layer)
- During training, we select the decoder based on the ground-truth
- We also train a classifier that selects which decoder to use during inference

Implementation

- The number of output channels(sources) scales up with output size of projection layer
- The count-head(which selects decoder) is trained, but not used during training

[256, 3999]->[512, 3999]



Global Averaging +
Fully Connected

[256, 3999]->[768, 3999]

Training

- Two components: decoder(signal loss) and count-head(classification loss)
- Training steps:

  1. Input mixture signal M
  2. Run encoder & LSTM backbone, compute intermediate output Z
  3. Run count-head with Z as input, compute probability P
  4. Run a decoder(selected based on ground truth) with Z as input, compute estimated sources S
  5. Loss is weighted sum of cross entropy loss for P and reconstruction loss for S

$$\mathcal{L}_{\text{count}(\mathbf{x},\mathcal{Y})} = -\sum_{k}^{K} \mathbf{1}_{|\mathcal{Y}|=k} \cdot \log \hat{p}(|\mathcal{Y}| = k \big| \mathbf{x}), \qquad \mathcal{L}_{\text{decoders}}(\mathbf{x},\mathcal{Y}) = \sum_{k} \mathbf{1}_{|\mathcal{Y}|=k} \cdot \text{uPIT}(\mathcal{Y}, \hat{\mathcal{Y}}_k),$$

Final Loss: $$\min_{\theta} \sum_{(\mathbf{x},\mathcal{Y})\in\mathcal{D}} \alpha \cdot \mathcal{L}_{\text{count}}(\mathbf{x},\mathcal{Y}) + (1-\alpha) \cdot \mathcal{L}_{\text{decoders}}(\mathbf{x},\mathcal{Y}).$$

Inference

- Two components: decoder(signal loss) and count-head(classification loss)
- Inference steps:
    1. Input mixture signal M
    2. Run encoder & LSTM backbone, compute intermediate output Z
    3. Run count-head with Z as input, compute probability P
    4. Run the decoder selected based on P to get estimated signals S = {S1, S2, …}

Performance Metric

- We define penalized-SNR, which has two components:
  1. Si-SNR for the matched part between ground truth(GT) and estimated sources
  2. Penalty term = {Constant P_ref} x {number of mismatched channels}
- We choose P_ref to either be -30 or the average SNR the system would achieve when provided with oracle number of speakers

P-Si-SNR = $$\frac{1}{|\mathcal{D}|}\sum_{(\mathbf{x},\mathcal{Y})\in\mathcal{D}}\frac{1}{\max(|\mathcal{Y}|,|\hat{\mathcal{Y}}|)}(\mathcal{L}_{\text{match}}+\mathcal{L}_{\text{pad}}),$$

where

$$\mathcal{L}_{\text{match}}=\max_{\pi}\sum_{n=1}^{\min(|\mathcal{Y}|,|\hat{\mathcal{Y}}|)}\text{SI-SNR}(\mathbf{y}^{\pi(n)},\hat{\mathbf{y}}^{n})$$ Signal-to-noise ratio

$$\mathcal{L}_{\text{pad}}=\mathcal{P}_{\text{ref}}\cdot\Big||\mathcal{Y}|-|\hat{\mathcal{Y}}|\Big|.$$ (penalty term)

The count-head performs with high accuracy(98.5%)

Pred \ True	**2**	**3**	**4**	**5**
2	2998	17	1	0
3	2	2977	27	0
4	0	6	2928	80
5	0	0	44	2920

Result: Source counting accuracy & oracle SNR

Model	2	3	4	5
Model-Select(DPRNN)[9]*	81.3	64.4	46.2	85.6
Model-Select(Mulcat)[9]*	84.6	69.0	47.5	92.3
Attractor Network[12]	95.7	97.6	-	-
OR-PIT[10]	95.7		-	-
Ours	99.9	99.2	97.6	97.3

Table 1. Performance of source counting; Each column is recall for corresponding number of speakers. For OR-PIT, only overall accuracy is provided.

2.4. Inference

Model	2	3	4	5
Conv-Tasnet[6]*	15.3	12.7	-	-
DPRNN[7]*	18.8	-	-	-
DPRNN[9]*	18.21	14.71	10.37	8.65
Mulcat[9]*	20.12	16.85	12.88	10.56
Attractor Network[12]	15.3	14.5	-	-
OR-PIT[10]	14.8	12.6	10.2	-
Ours	19.1	14.1	9.3	5.9

Table 2. Oracle SNR; Each column shows results averaged from all mixtures with corresponding number of speakers. *models above double-line are models with fixed number of speakers.

- Demo can be found at https://junzhejosephzhu.github.io/Multi-Decoder-DPRNN/
- Open-sourced implementation and pre-trained lightweight model can be found in asteroid toolkit
- Full model can be found at https://github.com/JunzheJosephZhu/MultiDecoder-DPRNN

Result: P-Si-SNR

- Model-Select indicates the method where a different model is trained for each number of speakers
- As can be seen, our method performs similarly well, but requires much less training resources, and is faster during inference.

$\mathcal{P}_{\text{ref}} = -30dB$	**2**	**3**	**4**	**5**
Model-Select(DPRNN)[10]*	15.2	10.7	6.0	7.7
Model-Select(Mulcat)[10]*	17.5	13.21	8.4	10.0
Attractor Network[14]	14.7	14.2	-	-
OR-PIT[11]	13.1		-	-
Ours	19.1	14.0	9.2	5.8

$\mathcal{P}_{\text{ref}} = -\text{SI-SNR}_{\text{oracle}}$	**2**	**3**	**4**	**5**
Model-Select(DPRNN)[10]*	15.9	12.1	8.1	8.2
Model-Select(Mulcat)[10]*	18.1	14.2	10.2	10.3
Attractor Network[14]	14.9	14.3	-	-
OR-PIT[11]	13.4		-	-
Ours	19.1	14.0	9.3	5.9

Table 4. P-SI-SNR of each model; For OR-PIT, result is computed by averaging the P-SI-SNR for both 2 and 3 speakers computed with 95.7% recall. Note that models with lower max speaker count generally have higher accuracy, since fewer classes implies a higher P-SI-SNR. * denotes models trained on fixed number of speakers.

Summary

- A method for source separation w/ unknown number of sources
- 2 Problems:
    1. how many sources? Solution: Use shared backbone & classifier(count-head)
    2. How to recover signal? Solution Use multiple decoder heads
- Result: High classification accuracy, no increase in complexity