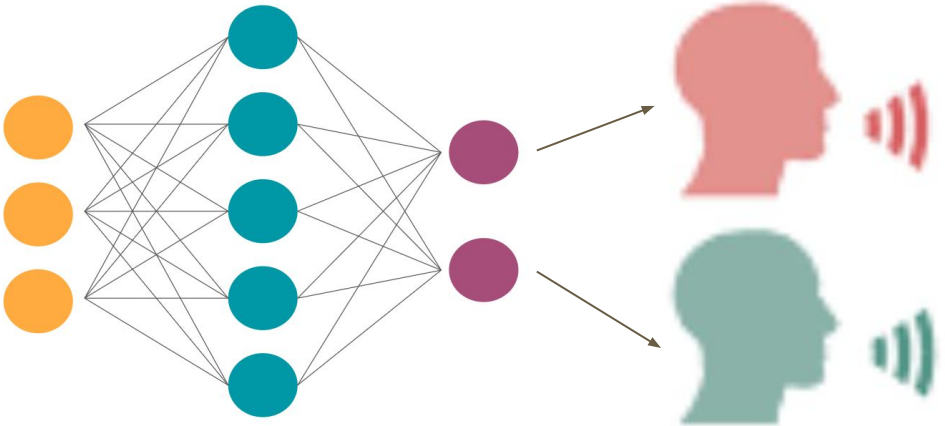
DLHLP - HW3
Source Separation

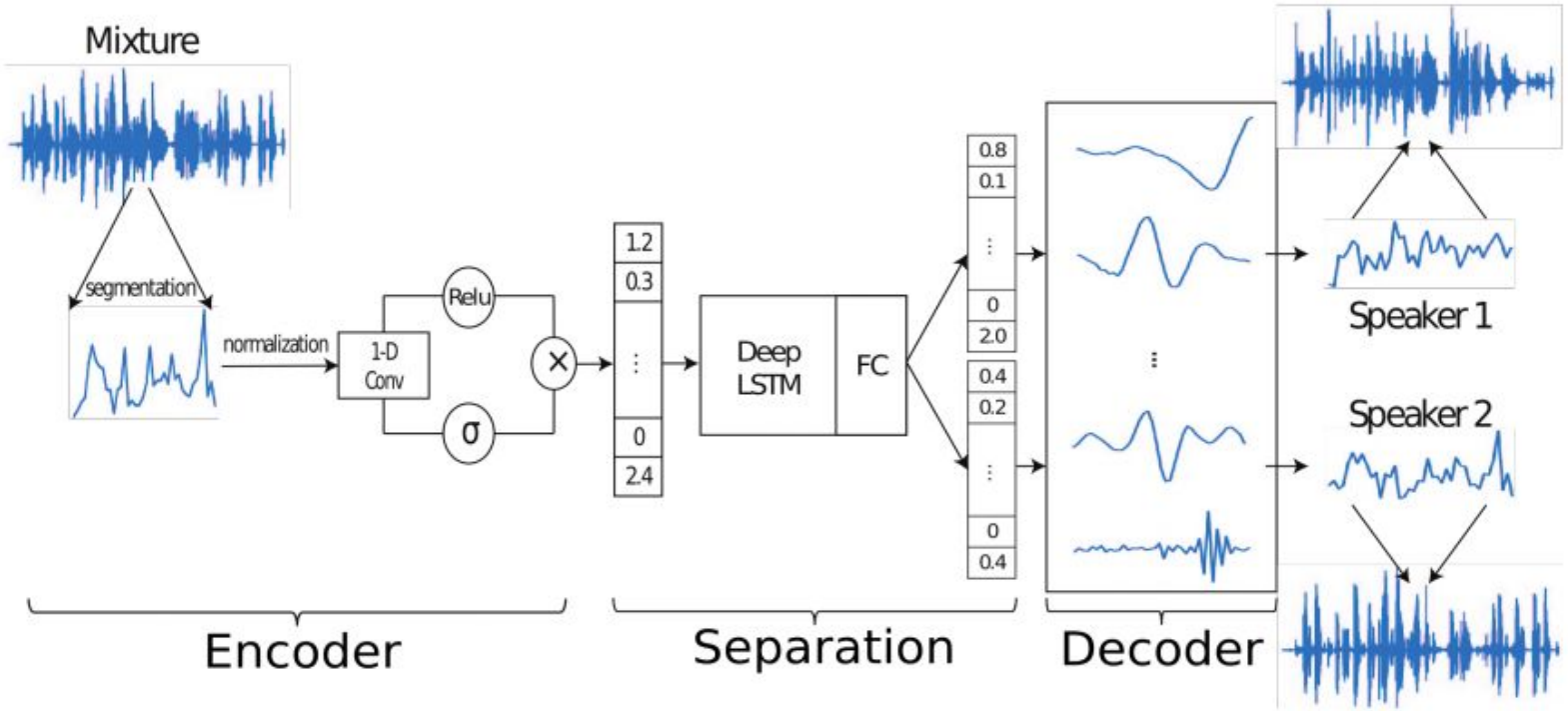
TA: 黃冠博、陳泓廷、楊采綸
dlhlp.ta@gmail.com

Source separation

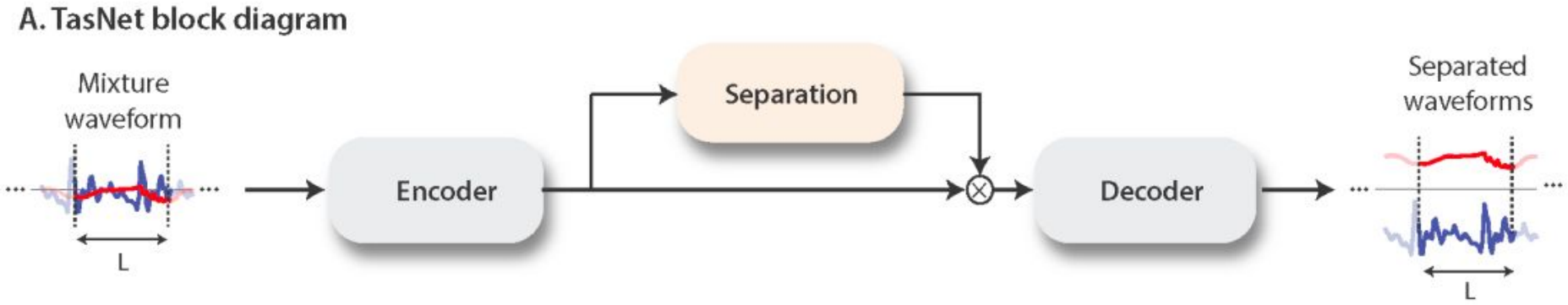
mixture of two people speaking



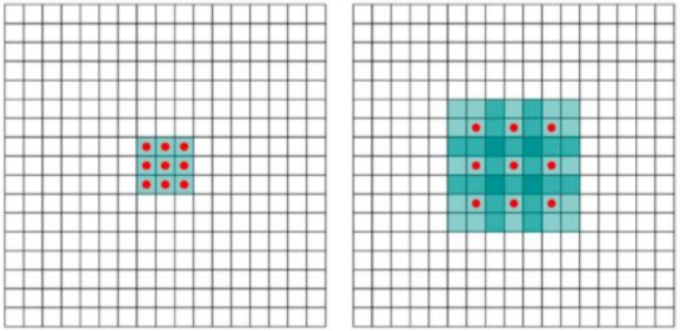
Time-domain audio separation network (TasNet)

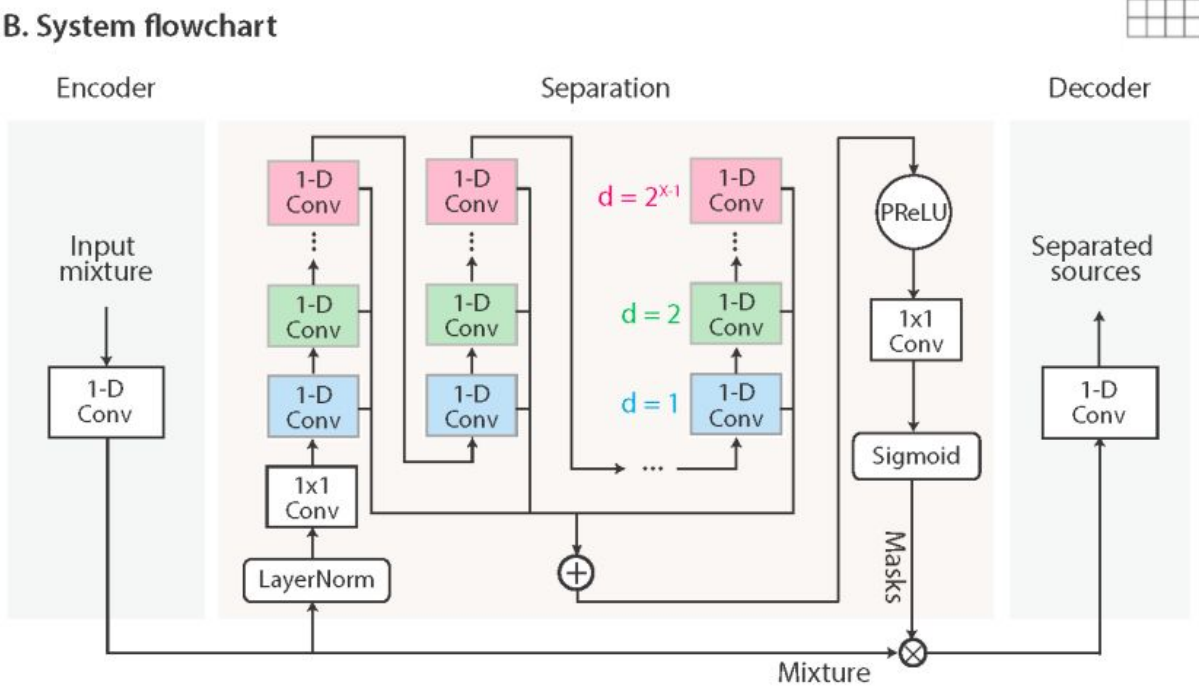


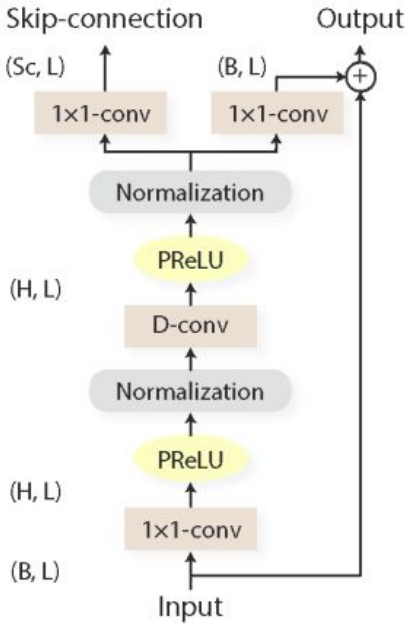
Conv-TasNet


A. TasNet block diagram
Mixture waveform
L
Encoder
Separation
Decoder
Separated waveforms
L

Conv-TasNet

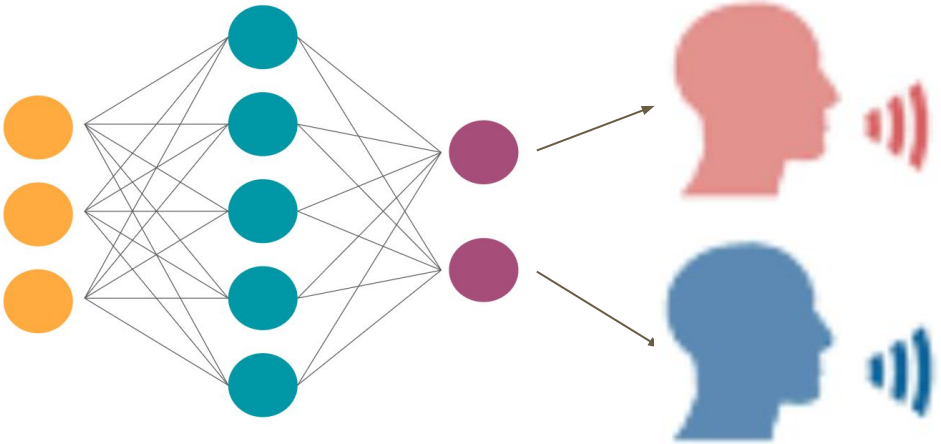



B. System flowchart
Encoder
Separation
Decoder
Input mixture
1-D Conv
1x1 Conv
LayerNorm
d = 2^X-1
d = 2
d = 1
PReLU
Sigmoid
Masks
Separated sources
Mixture


Skip-connection
Output
(Sc, L)
(B, L)
1×1-conv
Normalization
PReLU
(H, L)
D-conv
Input

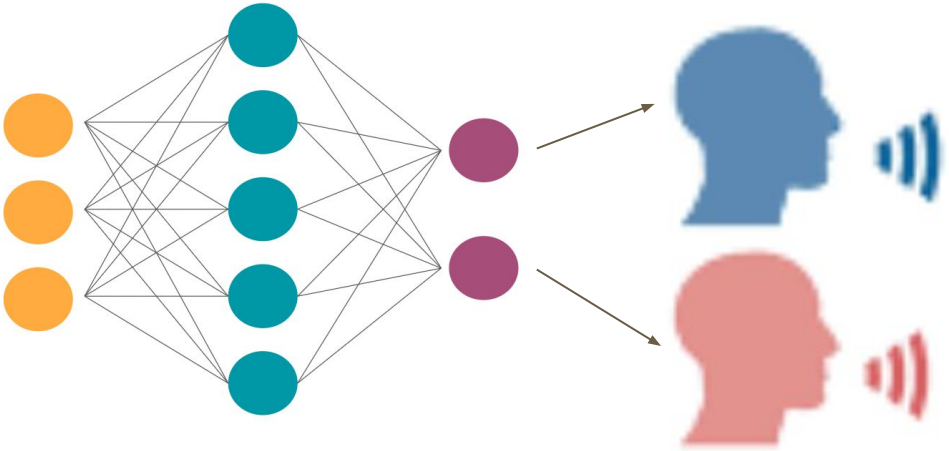
Problem

mixture of two people speaking

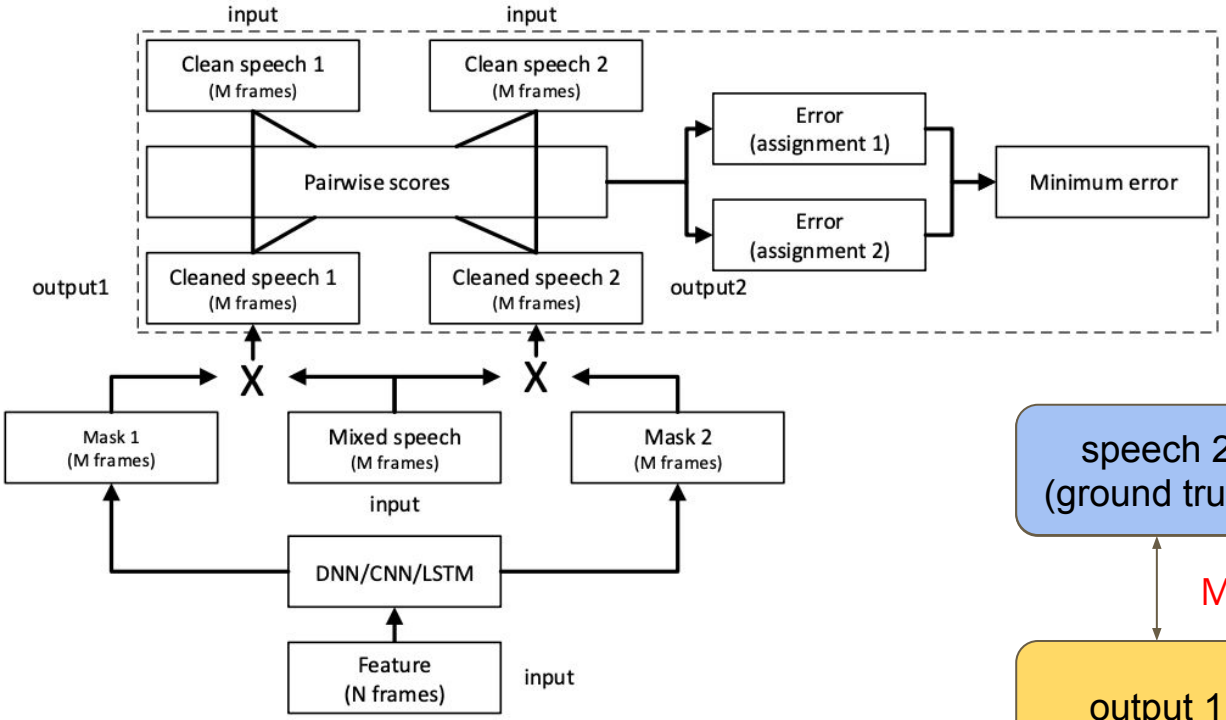


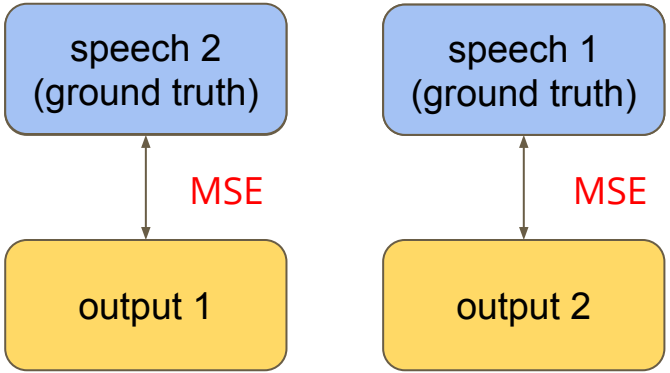
Problem

mixture of two people speaking



Permutation Invariant Training (PIT)





https://arxiv.org/pdf/1607.00325.pdf

Requirements

Run the two following tasks:

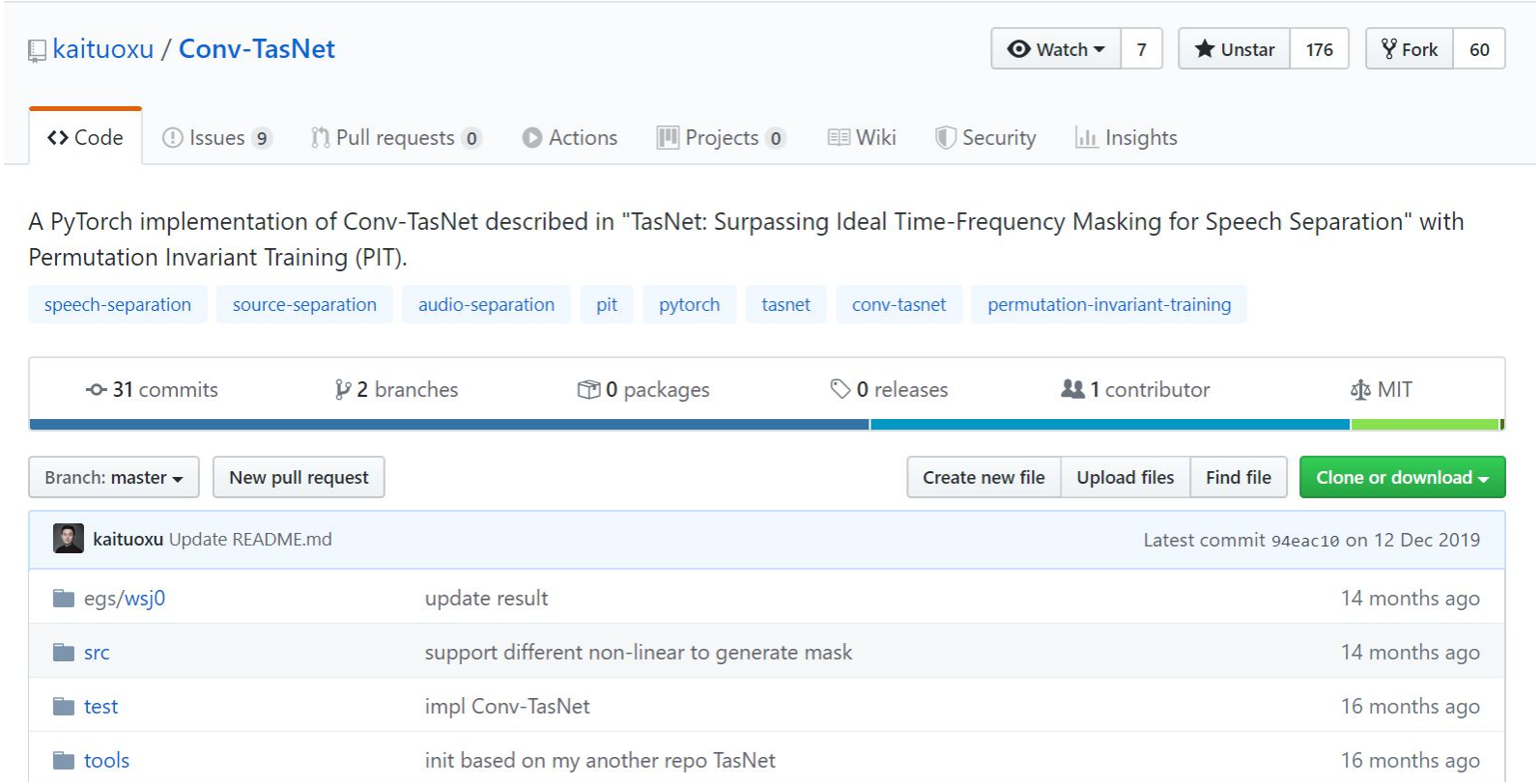
- 3-1 speaker dependent
  - only two speakers
  - training data and testing data are the same two speakers
- 3-2 speaker independent
  - many different speakers
  - the speaker set in training data and testing data do not overlap

Dataset

- 3-1 (2 GB) only two speakers
  - https://drive.google.com/open?id=1thV_9B1Noyf2Q91FTVJwGrS-Xh0BjqFT
  - gdown https://drive.google.com/uc?id=1thV_9B1Noyf2Q91FTVJwGrS-Xh0BjqFT
- 3-2 (5.7 GB) many speakers
  - https://drive.google.com/file/d/1g3ObZnCNtdYMLYe-YNwbjkMlUrFszjxY/view?usp=sharing
  - gdown https://drive.google.com/uc?id=1g3ObZnCNtdYMLYe-YNwbjkMlUrFszjxY

1.Download from google drive or
2.Use **gdown** command in the command line
(pip install gdown if not installed)

Implementation - Conv Tasnet [Link]



Run

- egs/wsj0/run.sh

```
1   #!/bin/bash
2
3   # Created on 2018/12
4   # Author: Kaituo XU
5
6   # -- START IMPORTANT
7   # * If you have mixture wsj0 audio, modify `data` to your path that including tr, cv and tt.
8   # * If you jsut have origin sphere format wsj0 , modify `wsj0_origin` to your path and
9   # modify `wsj0_wav` to path that put output wav format wsj0, then read and run stage 1 part.
10  # After that, modify `data` and run from stage 2.
11  wsj0_origin=/home/ktxu/workspace/data/CSR-I-WSJ0-LDC93S6A
12  wsj0_wav=/home/ktxu/workspace/data/wsj0-wav/wsj0
13  data=/home/ktxu/workspace/data/wsj-mix/2speakers/wav8k/min/
14  stage=1  # Modify this to control to start from which stage
```

don’t care about these two paths

make sure you modify this path(data) to the path where min/ is located

Network Configuration

- egs/wsj0/run.sh

```
# Network config
N=256
L=20
B=256
H=512
P=3
X=8
R=4
norm_type=gLN
causal=0
mask_nonlinear='relu'
C=2
```

Feel free to change the configuration by yourself.
Larger model size results in longer training time.

THE EFFECT OF DIFFERENT CONFIGURATIONS IN CONV-TASNET.

N	L	B	H	Sc	P	X	R	Normali-zation	Causal	Receptive field (s)	Model size	SI-SNRi (dB)	SDRi (dB)
128	40	128	256	128	3	7	2	gLN	×	1.28	1.5M	13.0	13.3
256	40	128	256	128	3	7	2	gLN	×	1.28	1.5M	13.1	13.4
512	40	128	256	128	3	7	2	gLN	×	1.28	1.7M	13.3	13.6
512	40	128	256	256	3	7	2	gLN	×	1.28	2.4M	13.0	13.3
512	40	128	512	128	3	7	2	gLN	×	1.28	3.1M	13.3	13.6
512	40	128	512	512	3	7	2	gLN	×	1.28	6.2M	13.5	13.8
512	40	256	256	256	3	7	2	gLN	×	1.28	3.2M	13.0	13.3
512	40	256	512	256	3	7	2	gLN	×	1.28	6.0M	13.4	13.7
512	40	256	512	512	3	7	2	gLN	×	1.28	8.1M	13.2	13.5
512	40	128	512	128	3	6	4	gLN	×	1.27	5.1M	14.1	14.4
512	40	128	512	128	3	4	6	gLN	×	0.46	5.1M	13.9	14.2
512	40	128	512	128	3	8	3	gLN	×	3.83	5.1M	14.5	14.8
512	32	128	512	128	3	8	3	gLN	×	3.06	5.1M	14.7	15.0
512	16	128	512	128	3	8	3	gLN	×	1.53	5.1M	**15.3**	**15.6**
512	16	128	512	128	3	8	3	cLN	✓	1.53	5.1M	10.6	11.0

PIT?

- egs/wsj0/run.sh

```
# Training config
use_cuda=1
id=0
epochs=100
half_lr=1
early_stop=1
max_norm=5
pit=1
```

change pit to 0
to disable PIT

Submit testing result

- for example: in the following directory “Conv-TasNet/egs/wsj0/exp/train_r8000_N256_L20_B256_H512_P3_X8_R4_C2_gLN_causal0_relu_epoch100_half1_norm5_bs3_worker4_adam_lr1e-3_mmt0_l20_tr/”
  - zip **separate/** and push it to your github repository (should be less than 100MB) **do not push separate/ to github without zipping !**
    - hw3/results/3-1/separate.zip
    - hw3/results/3-2/separate.zip
    - hw3/results/bonus/separate.zip (optional)
  - in **evaluate.log**, report average SISNR
  - **report.pdf**

```
Average SISNR improvement: 17.38
# Accounting: time=90 threads=1
```

Report(1/2) [Template]

1. (5%)請記錄 evaluate.log 裡面的SiSNR 數值，和當時所用的hyperparameter(這一題請3-1不用PIT, 3-2用PIT)
2. (5%)嘗試調整不同的hyperparameter，比較其差異，並試著分析結果(至少針對2種不同的hyperparameter進行實驗)
3. (3%)3-1, 3-2請分別試看看有無PIT的差異並記錄結果 (loss learning curve, Si-SNR)
4. (2%)思考一下為何有無PIT會影響3-1, 3-2的結果並寫下你的看法

Your best model should pass the baseline! Si-SNR = 10

NOTE: 第一題請回報最後上傳 Github的separation.zip的Si-SNR數值

Report(2/2)

[Template]

有用tasnet分離得到一個sisnr分數，improve後再分離得到更好的sisnr，並於report說明方法 ->2分
有用tasnet分離得到一個sisnr分數，只有提出方法 -> 0.5~1分
有用tasnet分離得到一個sisnr分數 沒有講關於improve的東西 -> 0分

bonus(2%) :

請自己找兩段音訊合起來(請不要使用作業給的data)測看看是否能成功分離，

上傳音訊(含原音檔、合成後音檔及經過model分離的音檔)，紀錄Si-SNR於report中，並給出至少一種improve Si-SNR的方法(調參數除外)。

將bonus結果放在hw3/results/bonus/separate.zip，裡面除了分離後的音檔，請另外創建資料夾"origin"，放進你的原音檔（s1, s2, mix）

- separate/
  - s1.wav, s2.wav
  - origin/
    - s1.wav, s2.wav, mix.wav

Reference

- Paper - TasNet
- Paper - Conv-TasNet
- Source code - Conv-TasNet implemented by kaituoxu (forked)

Deadline

- 2020/05/06 9:00