LaSAFT: Latent Source Attentive Frequency Transformation for Conditioned Source Separation

Woosung Choi, Minseok Kim, Jaehwa Chung, and Soonyoung Jung

Our code and models are available online.

Demonstrations: Conditioned Source Separation

```
print('bass')
separated = model.separate_track(data.T, 'bass')
bass, sr=librosa.load('temp.wav', mono=False)
display(Audio('temp.wav'))

print('other')
separated = model.separate_track(data.T, 'other')
other, sr=librosa.load('temp.wav', mono=False)
display(Audio('temp.wav'))

print('v+d+b+o')
librosa.output.write_wav('temp.wav', vocals+drums+bass+other, sr)
display(Audio('temp.wav'))
```

Colab Demonstration - Stella Jang's, Feel the breeze, Other Examples

Youtube Versions: Stella Jang's, Feel this breeze, Other Examples

Abstract

- Recent deep-learning approaches have shown that Frequency Transformation (FT) blocks can significantly improve spectrogram-based single-source separation models by capturing frequency patterns.
- The goal of this paper is to extend the FT block to fit the multi-source task.
- We propose
  - Latent Source Attentive Frequency Transformation (LaSAFT) block to capture source-dependent frequency patterns.
  - Gated Point-wise Convolutional Modulation (GPoCM), an extension of Feature-wise Linear Modulation (FiLM), to modulate internal features.
- By employing these two novel methods, we extend the Conditioned-U-Net (CUNet) for multi-source separation, and the experimental results indicate that our LaSAFT and GPoCM can improve the CUNet's performance, achieving state-of-the-art SDR performance on several MUSDB18 source separation tasks.

Preliminaries 1: Categories of Source separation models

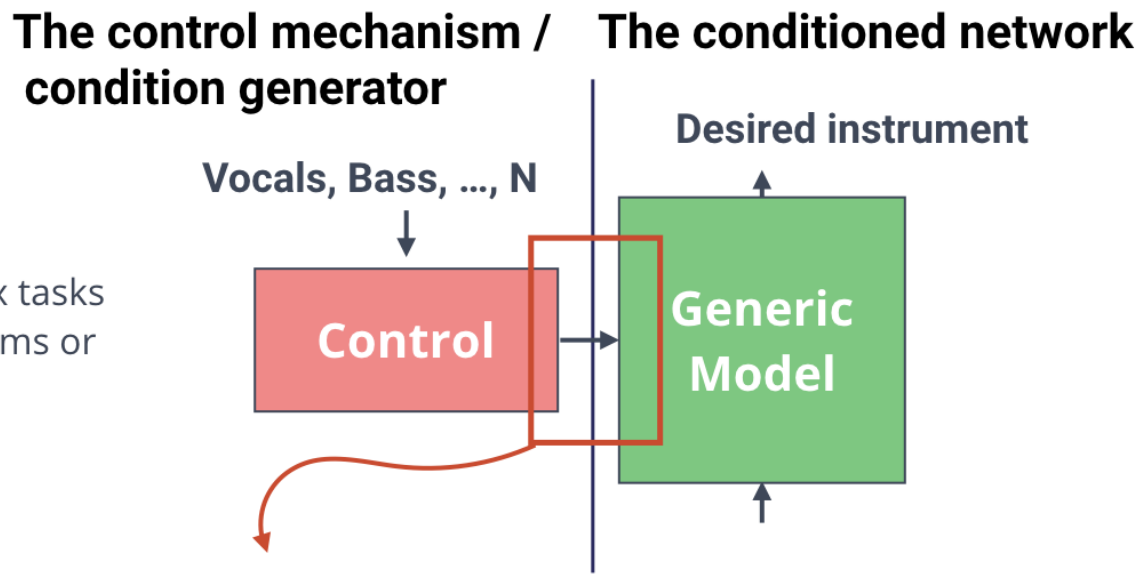
- *Dedicated models*
    - Most of the deep learning-based models for Music Source Separation (MSS) are dedicated to a single instrument.
    - cons1: forces us to train an individual model for each instrument.
    - cons2: models cannot use the commonalities between different instruments.
- *Multi-head models*
    - Let us generate several outputs at once with a multi-head.
    - Although it shows promising results, this approach still has a scaling issue: the number of heads increases as the number of instrument increases, leading
        a. performance degradation caused by the shared bottleneck
        b. inefficient memory usage.

Preliminaries 1: An alternative approach

- Conditioning/Meta-Learning
  - can separate different instruments with the aid of the **control mechanism**.
  - no shared bottleneck, no multi-head output layer

Core idea: An input x is processed differently depending on external context

Even complex tasks like bass+drums or voice+drums



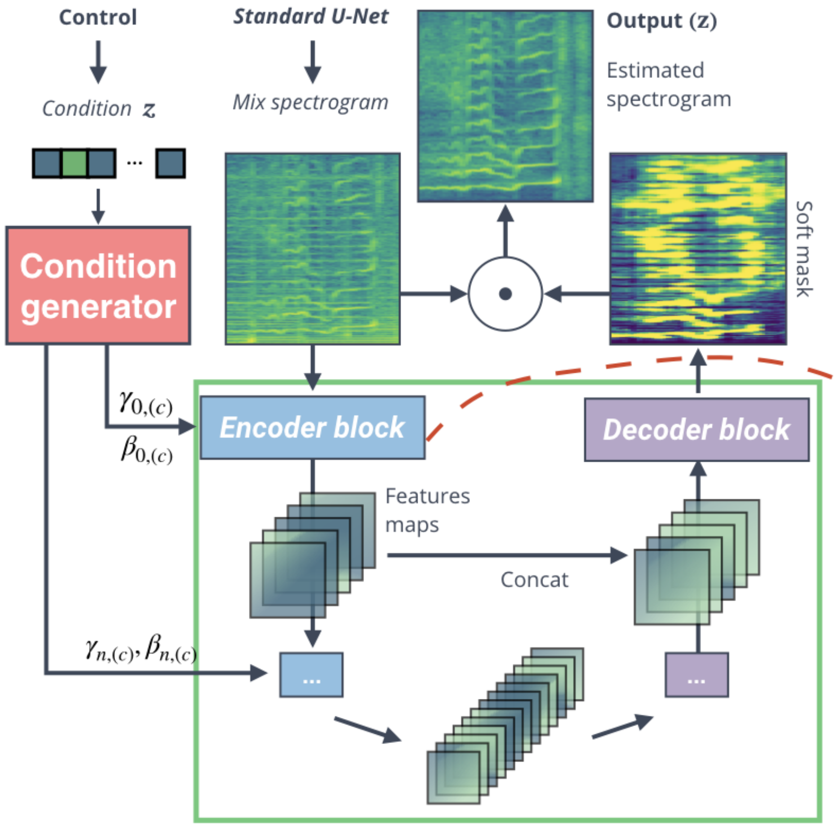
The control is done via FiLM (Feature-wise transformations) layers

Preliminaries 1: Conditioned Source Separation

- Task Definition
  - Input: an input audio track A and a one-hot encoding vector C that specifies which instrument we want to separate
  - Output: separated track of the target instrument

Preliminaries 1: Example - Conditioned U-Net

- Conditioned-U-Net extends the U-Net by exploiting Feature-wise Linear Modulation (FiLM)

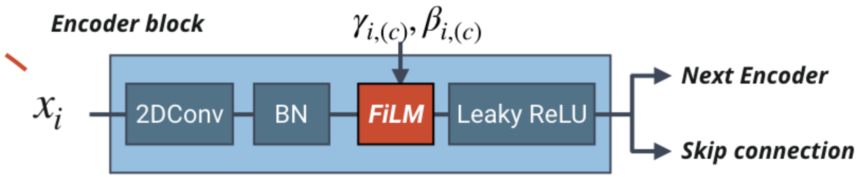


The conditioned network

U-Net architecture for source separation: (A. Jansson et al., 2017).

Where to condition?

Our hypothesis: different way of encoding each instrument i.e., different deep representations, a common 'universal' decoder to interpret all of them.



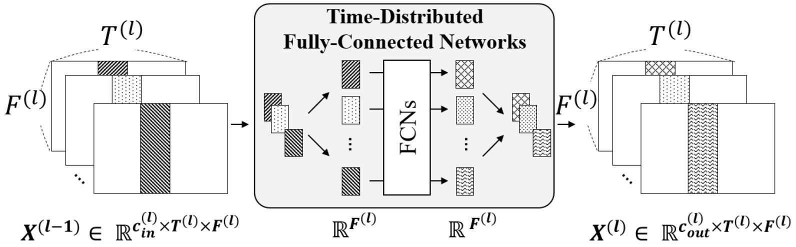
The control mechanism /condition generator

It has to transform a one hot that encode the desired instrument to isolate into the needed γ's and β's:

1. **Fully-Connected embedding (F)**
2. **CNN embedding (C)**

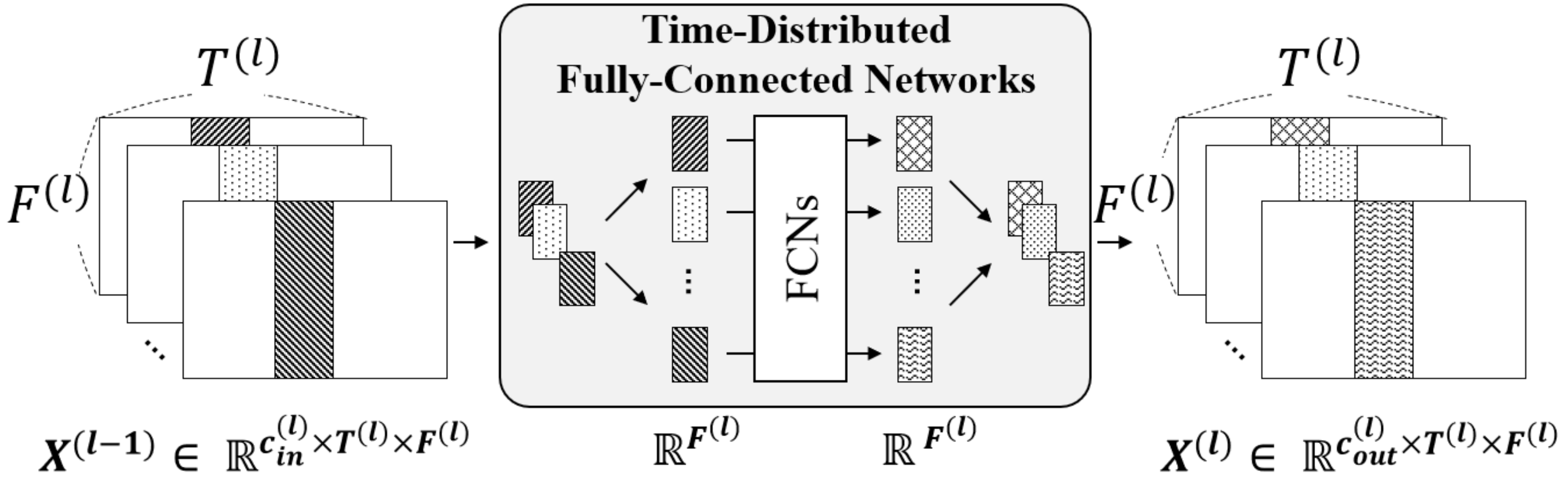
Preliminaries 2: Frequency Transformation Block

- Frequency patterns
  - Recent spectrogram-based methods for Singing Voice Separation (SVS) or Speech Enhancement (SE) employed Frequency Transformation (FT) blocks to capture ***frequency patterns***.
  - Although stacking 2-D convolutions has shown remarkable results, it is hard to capture long-range dependencies along the frequency axis for fully convolutional networks with small sizes of kernels.
  - FT blocks, which have ***fully-connected layers*** applied in a time-distributed manner, are useful to this end.

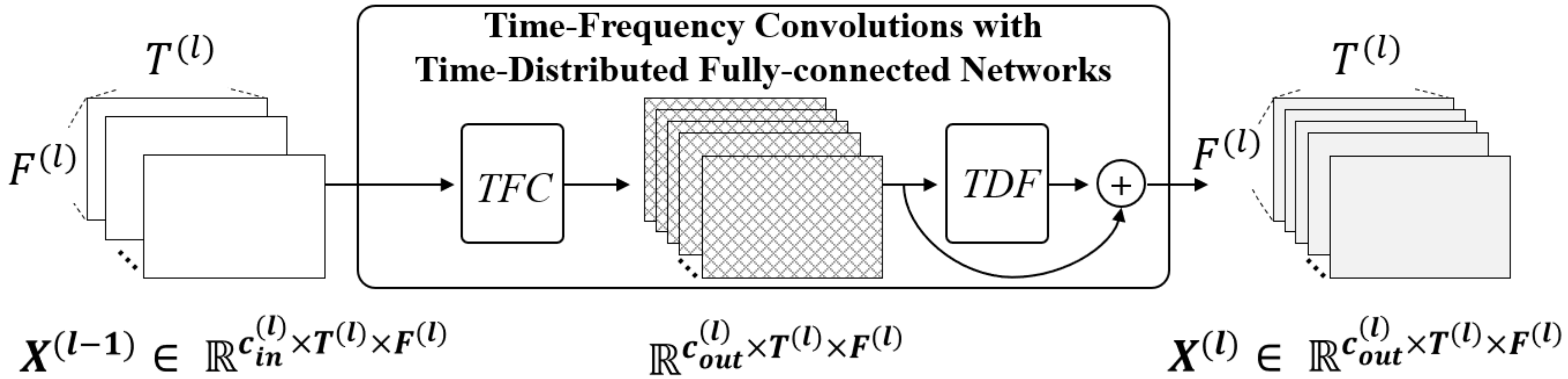


Preliminaries 2: Injecting FT blocks into U-Nets

- An FT block called Time-Distributed Fully-connected Layer (TDF):

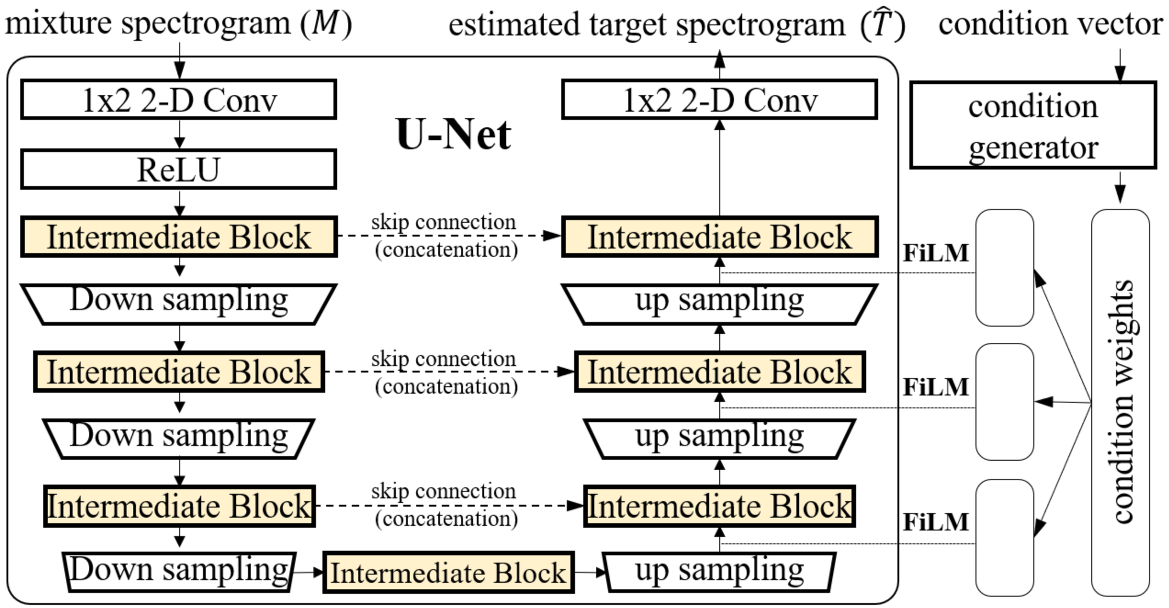


- TFC-TDF: SDR 6.75dB $\longrightarrow$ 7.12dB in Singing Voice Separation



Naive Extention: Injecting FT blocks into C-U-Net?

- Baseline U-Net



- TFC vs TFC-TDF

model	vocals	drums	bass	other	AVG
dedicated [8]	7.07	5.38	5.62	4.61	5.66
FiLM CUNet	5.14	5.25	4.20	3.40	4.49
+ TDF	5.88	5.70	4.55	3.67	4.95

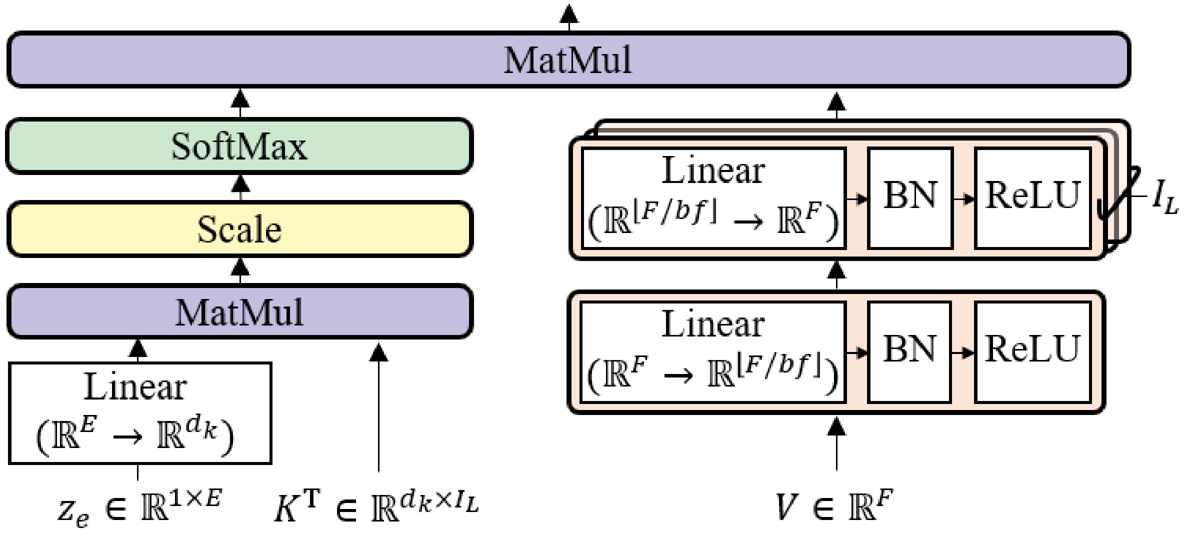
Naive Extention: Above our expectation

- TFC vs TFC-TDF

model	vocals	drums	bass	other	AVG
dedicated [8]	7.07	5.38	5.62	4.61	5.66
FiLM CUNet	5.14	5.25	4.20	3.40	4.49
+ TDF	5.88	5.70	4.55	3.67	4.95

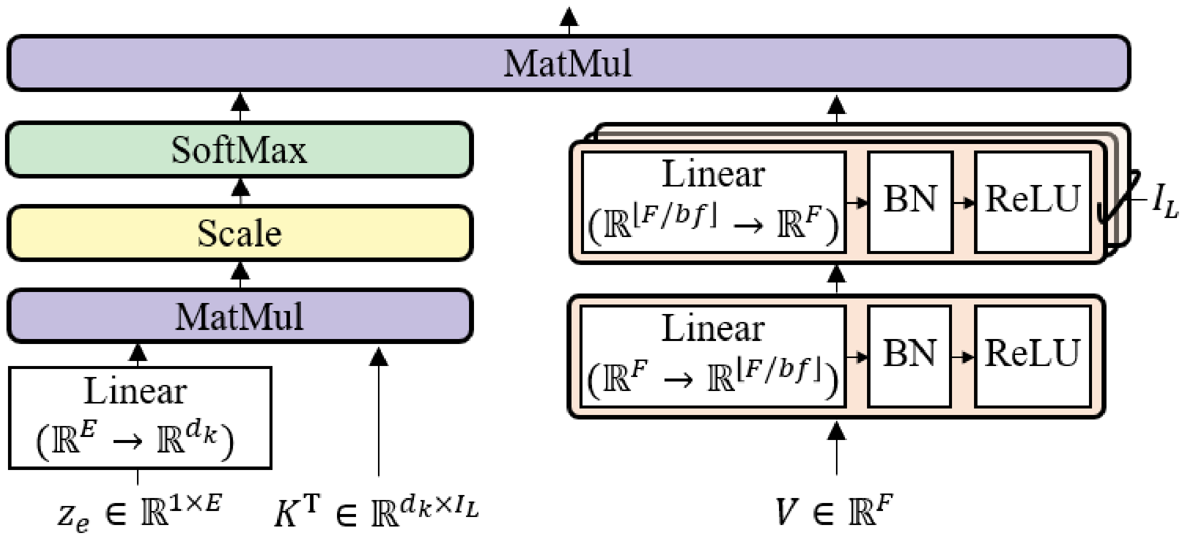
- Although it does improve SDR performance by capturing common frequency patterns observed across all instruments
- Merely injecting an FT block to a CUNet **does not inherit the spirit of FT block**
- In this paper,
  - We propose the Latent Source-Attentive Frequency Transformation (LaSAFT), a novel frequency transformation block that can capture instrument-dependent frequency patterns by exploiting the scaled dot-product attention

LaSAFT: Extending TDF to the Multi-Source Task (1)



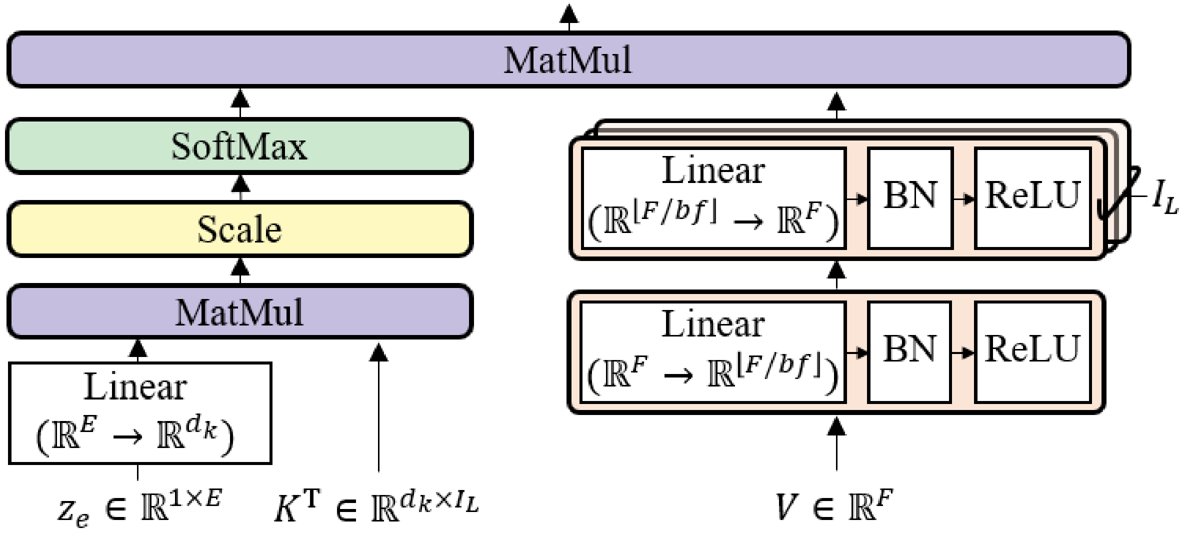
- duplicate $\mathcal{I}_L$ copies of the second layer of the TDF, where $\mathcal{I}_L$ refers to the number of ***latent instruments***.
  - $\mathcal{I}_L$ is not necessarily the same as $\mathcal{I}$ for the sake of flexibility
- For the given frame $V \in \mathbb{R}^F$, we obtain the $\mathcal{I}_L$ latent instrument-dependent frequency-to-frequency correlations, denoted by $V' \in \mathbb{R}^{F \times \mathcal{I}_L}$.

LaSAFT: Extending TDF to the Multi-Source Task (2)



- The left side determines how much each ***latent source*** should be attended
- The LaSAFT takes as input the instrument embedding $z_e \in \mathbb{R}^{1\times E}$.
- It has a learnable weight matrix $K \in \mathbb{R}^{\mathcal{I}_L \times d_k}$, where we denote the dimension of each instrument's hidden representation by d_k.
- By applying a linear layer of size d_k to z_e, we obtain $Q \in \mathbb{R}^{d_k}$.

LaSAFT: Extending TDF to the Multi-Source Task (3)



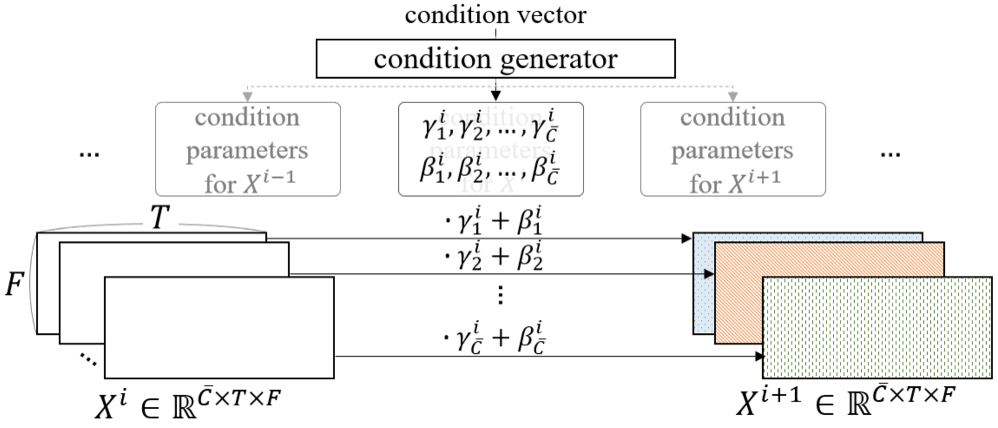
- We now can compute the output of the LaSAFT as follows:
  - $Attention(Q, K, V') = softmax(\frac{QK^T}{\sqrt{d_k}})V'$
- We apply a LaSAFT after each TFC in the encoder and after each Film/GPoCM layer in the decoder. We employ a skip connection for LaSAFT and TDF, as in TFC-TDF.

Effects of employing LaSAFTs instead of TFC-TDFs

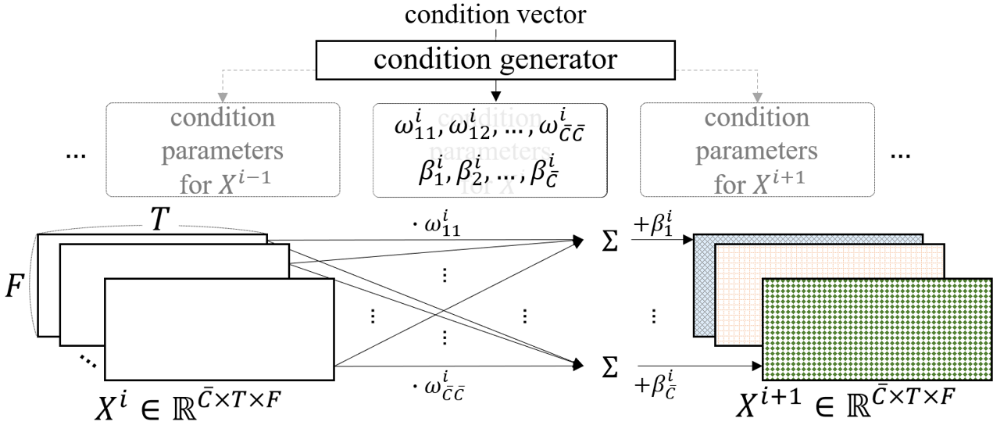
model	vocals	drums	bass	other	AVG
dedicated [8]	7.07	5.38	5.62	4.61	5.66
FiLM CUNet	5.14	5.25	4.20	3.40	4.49
+ TDF	5.88	5.70	4.55	3.67	4.95
+ LaSAFT	6.74	5.64	5.13	4.32	5.46

GPoCM: FiLM is also not enough

- Feature-wise Linear Modulation (FiLM)

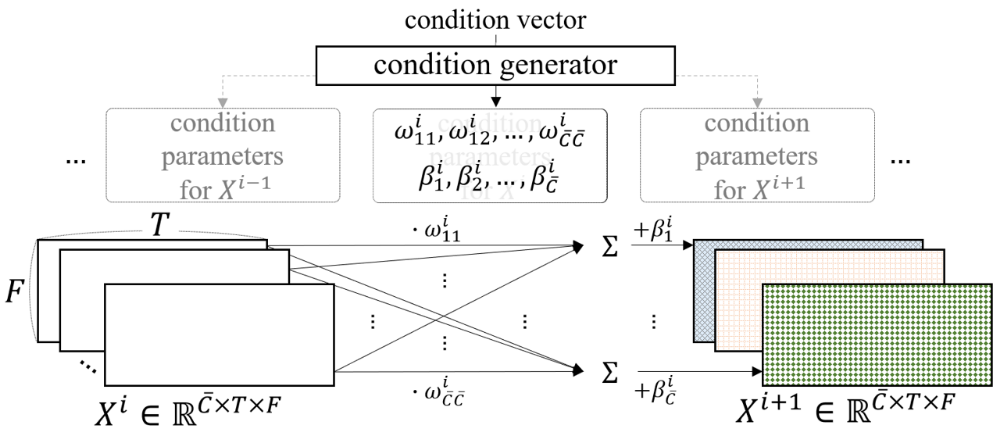


- Point-wise Conolutional Modulation (PoCM)



GPoCM: PoCM

- PoCM is an extension of FiLM. While FiLM does not have inter-channel operations
  - $FiLM(X_c^i|\gamma_c^i, \beta_c^i) = \gamma_c^i \cdot X_c^i + \beta_c^i$
  - $PoCM(X_c^i|\omega_c^i, \beta_c^i) = \beta_c^i + \sum_j \omega_{cj}^i \cdot X_j^i$
    - where γ_c^i and β_c^i are parameters generated by the condition generator, and X^i is the output of the i^{th} decoder's intermediate block, whose subscript refers to the c^{th} channel of X



GPoCM: Gated PoCM

- Since this channel-wise linear combination can also be viewed as a point-wise convolution, we name it PoCM. With inter-channel operations, PoCM can modulate features more flexibly and expressively than FiLM.
- Instaed of PoCM, we use Gated PoCM (GPoCM), since GPoCN is robust for source separation task. It is natural to use ***gated*** apporach the source separation tasks becuase a sparse latent vector (that contains many near-zero elements) obtained by applying GPoCMs, naturally generates separated result (i.e. more silent than the original).
- $GPoCM(X_c^i|\omega_c^i, \beta_c^i) = \sigma(PoCM(X_c^i|\omega_c^i, \beta_c^i)) \odot X_c^i$
  - where σ is a sigmoid and $\odot$ means the Hadamard product.

Experimental Results

model	vocals	drums	bass	other	AVG
dedicated [8]	7.07	5.38	5.62	4.61	5.66
FiLM CUNet	5.14	5.25	4.20	3.40	4.49
+ TDF	5.88	5.70	4.55	3.67	4.95
+ LaSAFT	6.74	5.64	5.13	4.32	5.46
GPoCM CUNet	5.43	5.30	4.43	3.51	4.67
+ TDF	5.94	5.46	4.47	3.81	4.92
+ LaSAFT	**6.96**	**5.84**	**5.24**	**4.54**	**5.64**

Table 1. An ablation study: *dedicated* means U-Nets for the single source separation, trained separately. FiLM CUNet refers the baseline in §2. The last row is our proposed model.

model	vocals	drums	bass	other	AVG
DGRU-DConv[1]	6.85	5.86	4.86	4.65	5.56
Meta-TasNet[4]*	6.40	5.91	5.58	4.19	5.52
Nachmani[19]*	6.92	6.15	**5.88**	4.32	5.82
D3Net [2]	7.24	**7.01**	5.25	4.53	**6.01**
proposed	**7.33**	5.68	5.63	**4.87**	5.88

Table 2. A comparison SDR performance of our models with other systems. '∗' denotes model operating in time domain.

our model's excellent SDR performance on vocals.

LaSAFT + GPoCM

- achieved state-of-the-art SDR performance on vocals and other tasks in Musdb18.

RANK	MODEL	SDR (VOCALS)	SDR (DRUMS)	SDR (BASS)	SDR (OTHER)	EXTRA TRAINING DATA	PAPER	CODE	RESULT	YEAR
1	LaSAFT+GPoCM	7.33	5.68	5.63	4.87	×	LaSAFT: Latent Source Attentive Frequency Transformation for Conditioned Source Separation			2020
2	DEMUCS (extra)	7.05	7.08	6.70	4.47	✓	Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed			2019
3	Spleeter (MWF)	6.86	6.71	5.51	4.02	✓	Spleeter: A Fast And State-of-the Art Music Source Separation Tool With Pre-trained Models			2019
4	Conv-TasNet	6.81	6.08	5.66	4.37	×	Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation			2019
5	Conv-TasNet (extra)	6.74	7.11	7.00	4.44	✓	Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation			2019

Discussion

- The authors of cunet tried to manipulate latent space in the encoder,
    - assuming the decoder can perform as a general spectrogram generator, which is `shared' by different sources.
- However, we found that this approach is not practical since it makes the latent space (i.e., the decoder's input feature space) more discontinuous.
- Via preliminary experiments, we observed that applying FiLMs in the decoder was consistently better than applying FilMs in the encoder.