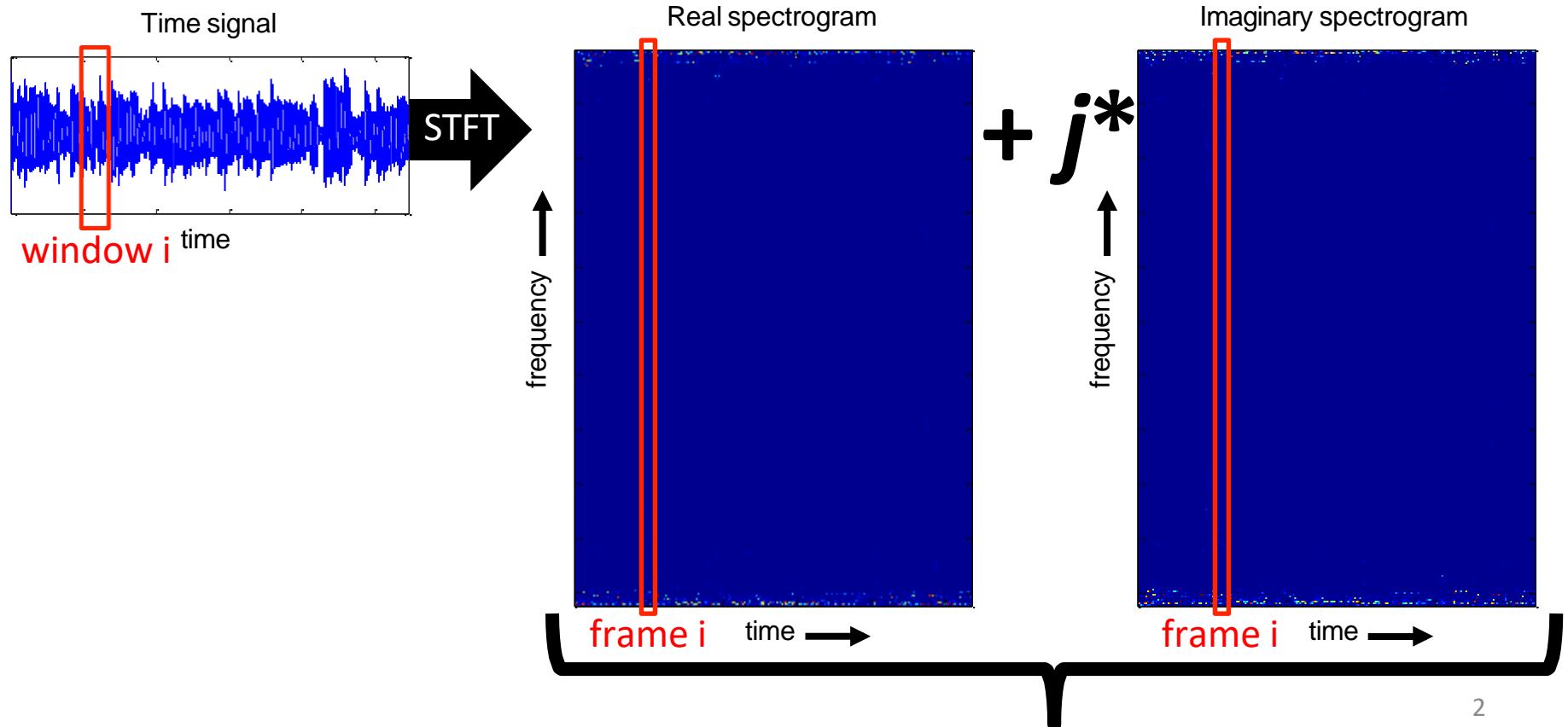


Time-frequency Masking

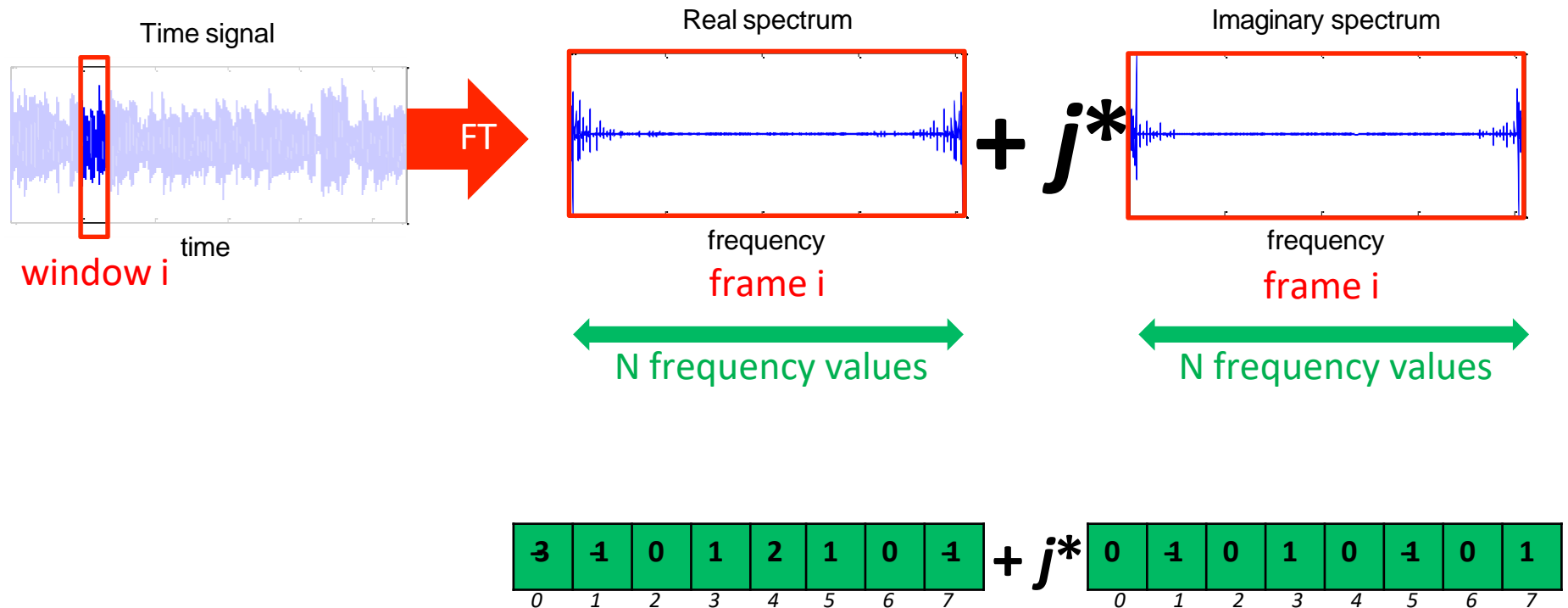
STFT

- The **Short-Time Fourier Transform** (STFT) is a succession of local Fourier Transforms (FT)



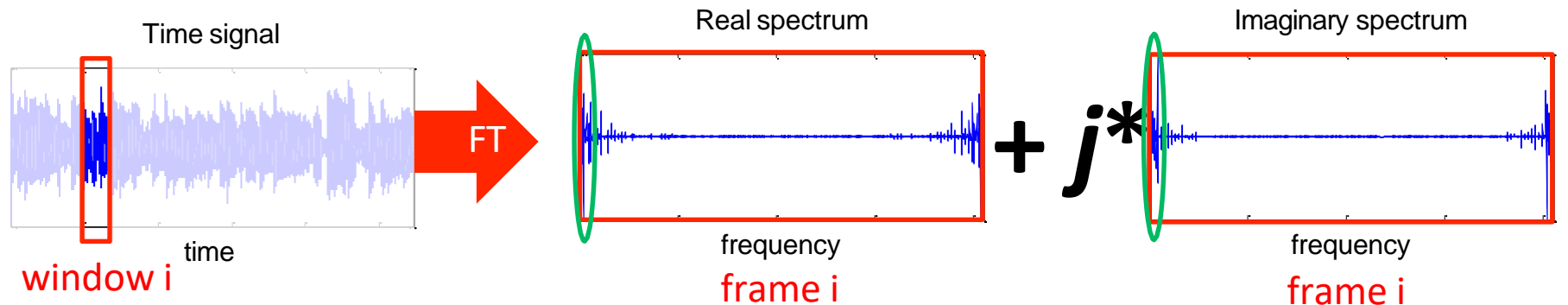
STFT

- If we used a window of N samples, the FT has N values, from 0 to $N - 1$; e.g., if $N = 8$...



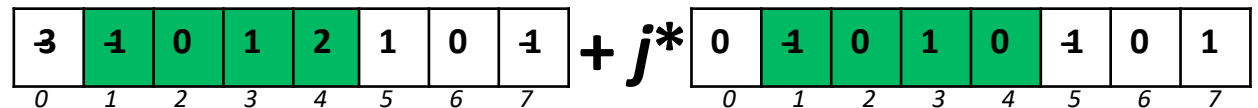
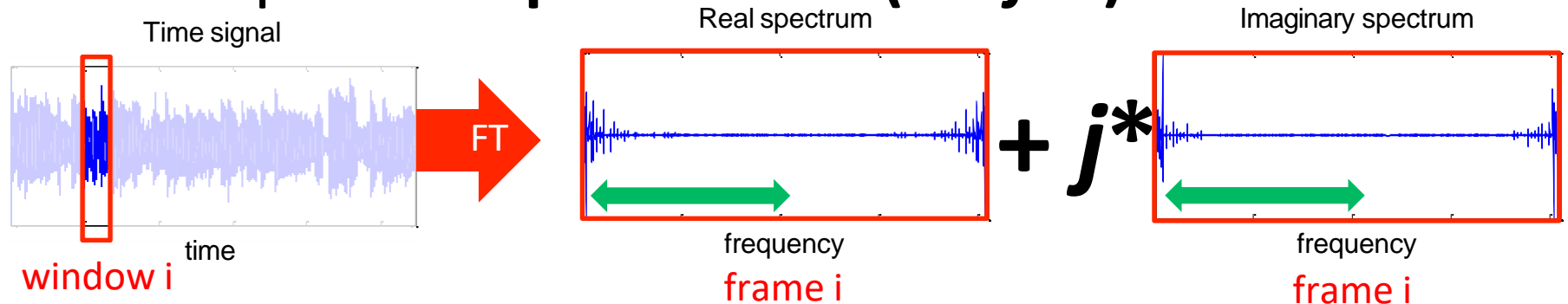
STFT

- Frequency index 0 is the **DC component**; it is always real (it is the sum of the time values!)



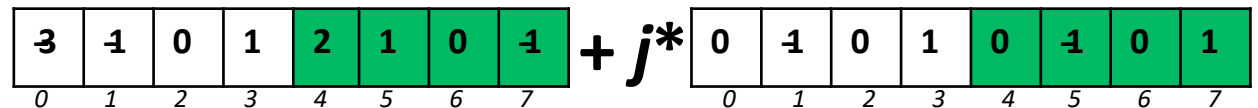
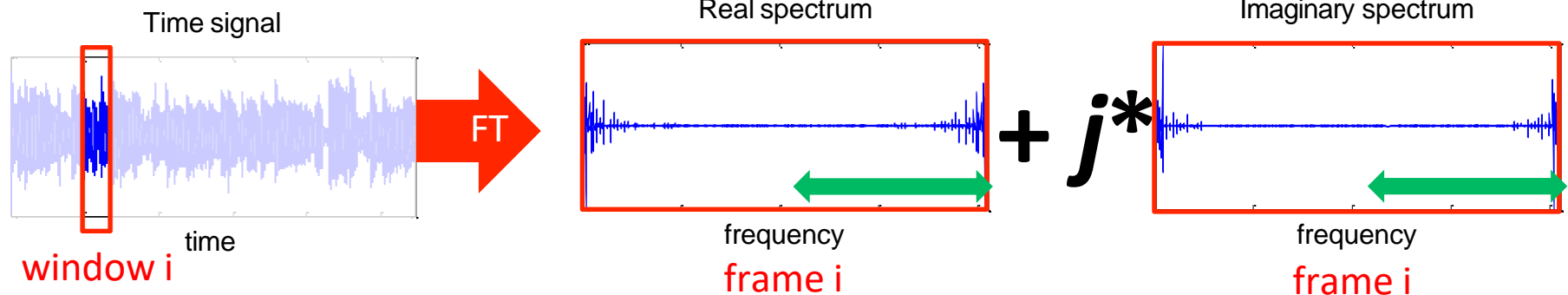
STFT

- Frequency indices from 1 to floor(N/2) are the “unique” **complex values $(a + j*b)$**



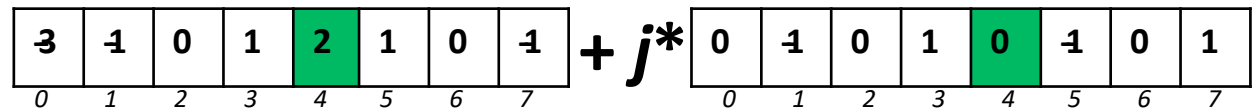
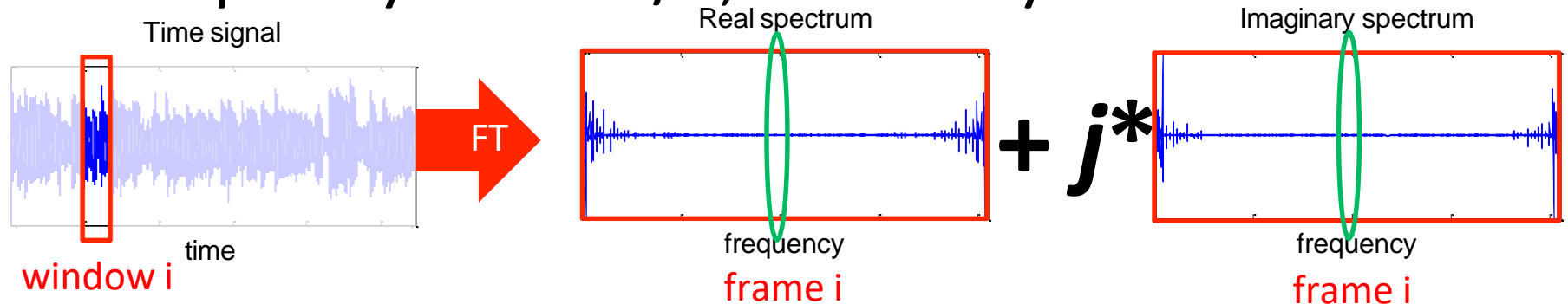
STFT

- Frequency indices from $\text{floor}(N/2)$ to $N-1$ are the “mirrored” **complex conjugates** ($a - j^*b$)



STFT

- If N is even, there is a **pivot component** at frequency index $N/2$; it is always real!



STFT

- Summary of the frequency indices and values in the STFT (in colors!)

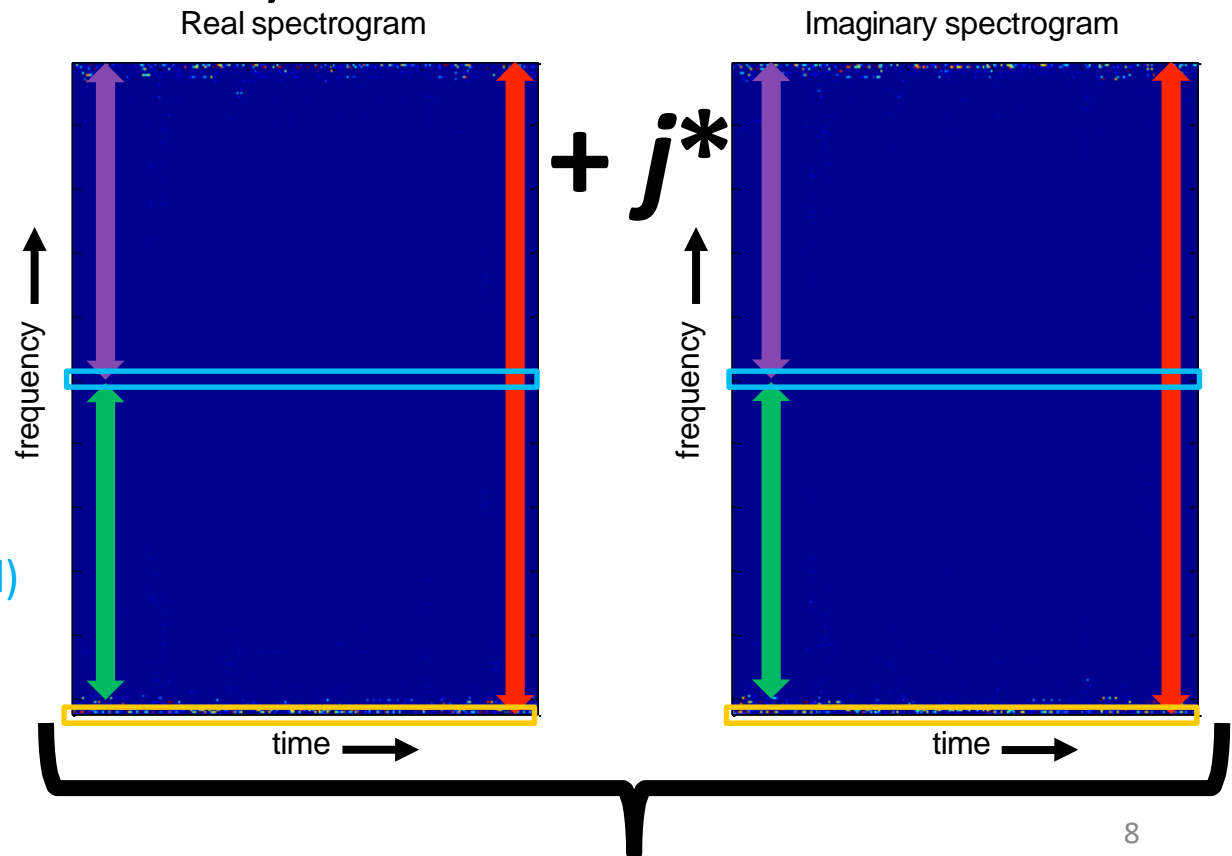
N frequency values =
frequency 0 to N-1

Frequency 0 =
DC component (always real)

Frequency 1 to floor(N/2) =
“unique” complex values

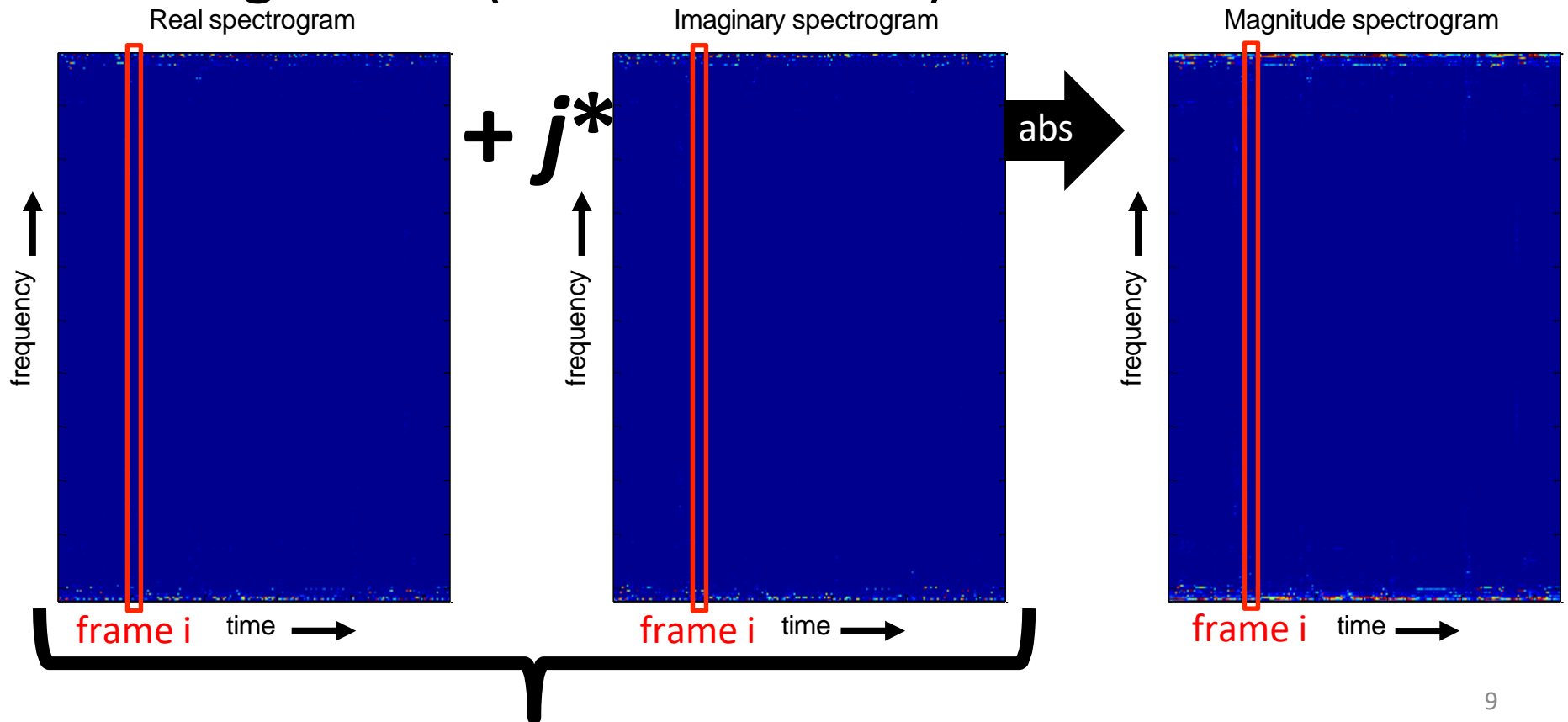
Frequency N/2 =
“pivot” component (always real)

Frequency floor(N/2) to N-1 =
“mirrored” complex conjugates



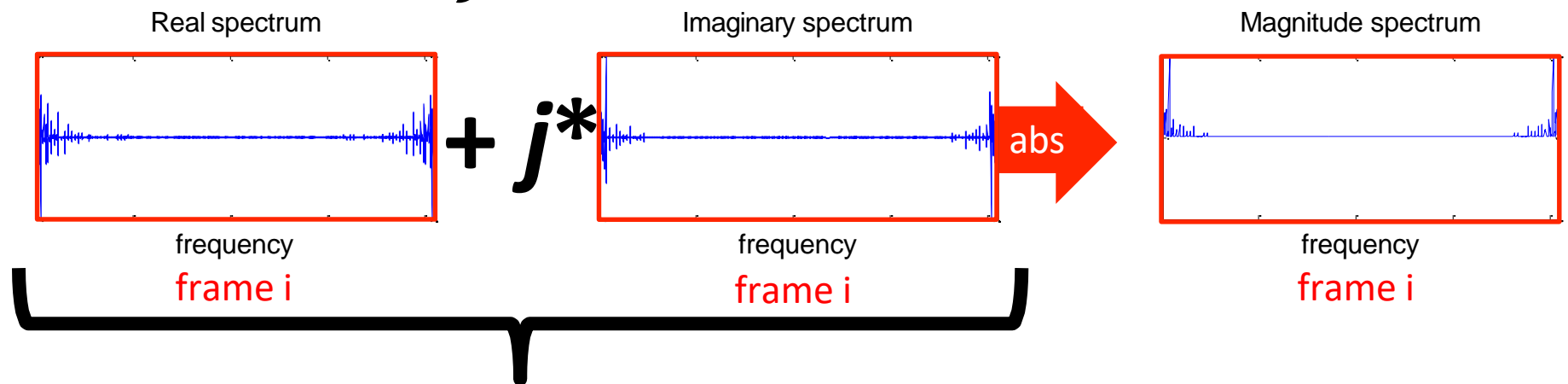
Spectrogram

- The (magnitude) **spectrogram** is the magnitude (absolute value) of the STFT



Spectrogram

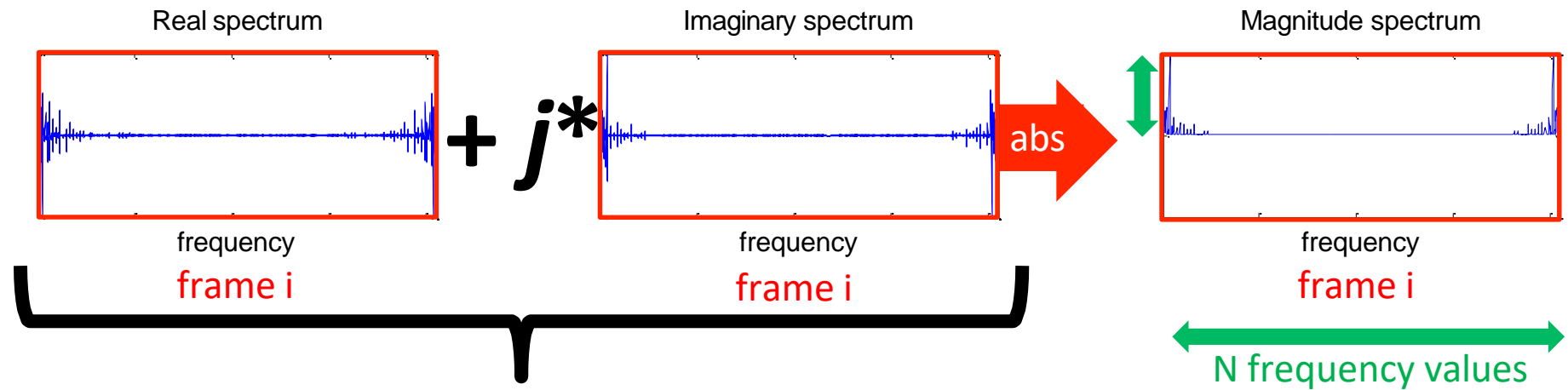
- For a complex number $a+j*b$, the absolute value is $|a+j*b|=\sqrt{a^2+b^2}$



$$\left| \begin{array}{c|c|c|c|c|c|c|c} 3 & -1 & 0 & 1 & 2 & 1 & 0 & -1 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \right| + j* \left| \begin{array}{c|c|c|c|c|c|c|c} 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \right| = \begin{array}{c|c|c|c|c|c|c|c} 3 & 1.4 & 0 & 1.4 & 2 & 1.4 & 0 & 1.4 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

Spectrogram

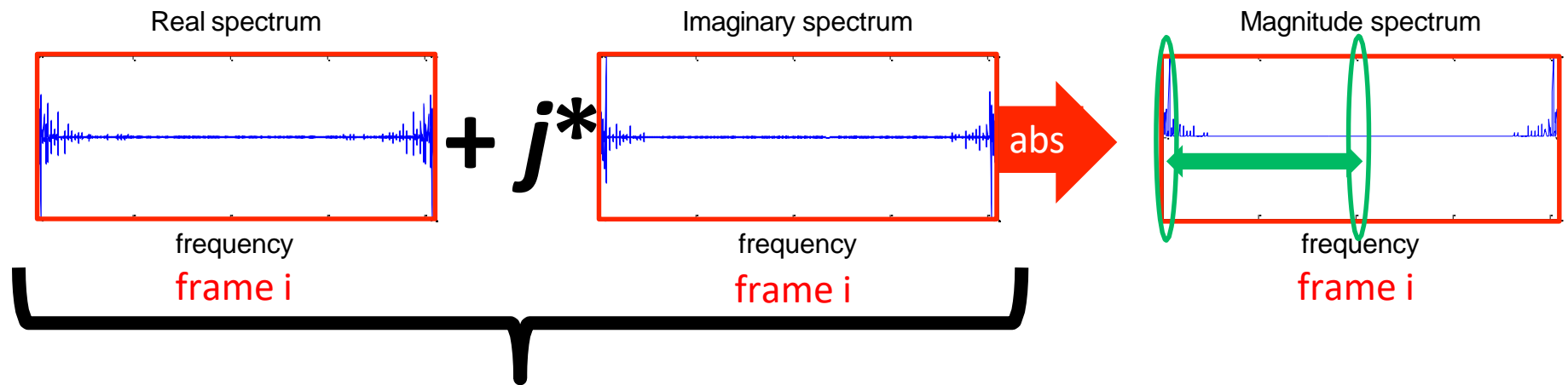
- All the N frequency values (frequency indices from 0 to $N-1$) are **real and positive** (abs!)



$$\left| \begin{array}{c|c|c|c|c|c|c|c|} \mathbf{3} & \mathbf{-1} & \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{1} & \mathbf{0} & \mathbf{-1} \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \right| + j * \left| \begin{array}{c|c|c|c|c|c|c|c|} \mathbf{0} & \mathbf{-1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{-1} & \mathbf{0} & \mathbf{1} \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \right| = \begin{array}{c|c|c|c|c|c|c|c|} \mathbf{3} & \mathbf{1.4} & \mathbf{0} & \mathbf{1.4} & \mathbf{2} & \mathbf{1.4} & \mathbf{0} & \mathbf{1.4} \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

Spectrogram

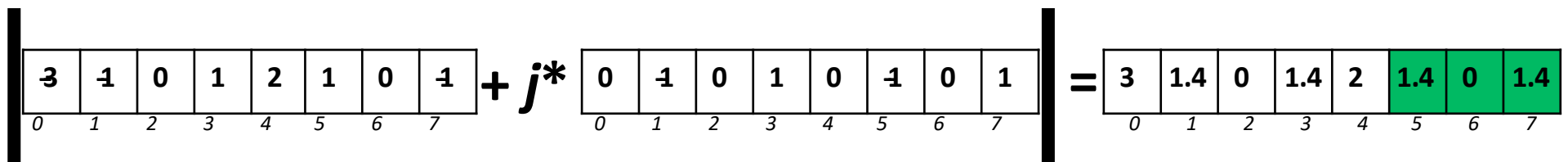
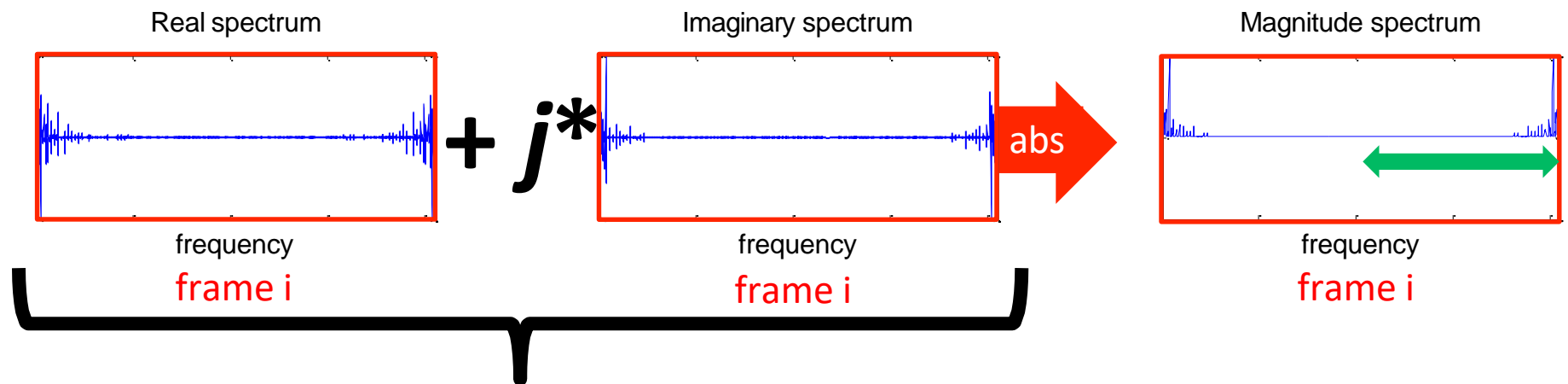
- Frequency indices from 0 to floor(N/2) are the **unique frequency values** (with DC and pivot)



$$\left| \begin{array}{c|c|c|c|c|c|c|c} 3 & -1 & 0 & 1 & 2 & 1 & 0 & -1 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \right| + j^* \left| \begin{array}{c|c|c|c|c|c|c|c} 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \right| = \begin{array}{c|c|c|c|c|c|c|c} 3 & 1.4 & 0 & 1.4 & 2 & 1.4 & 0 & 1.4 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

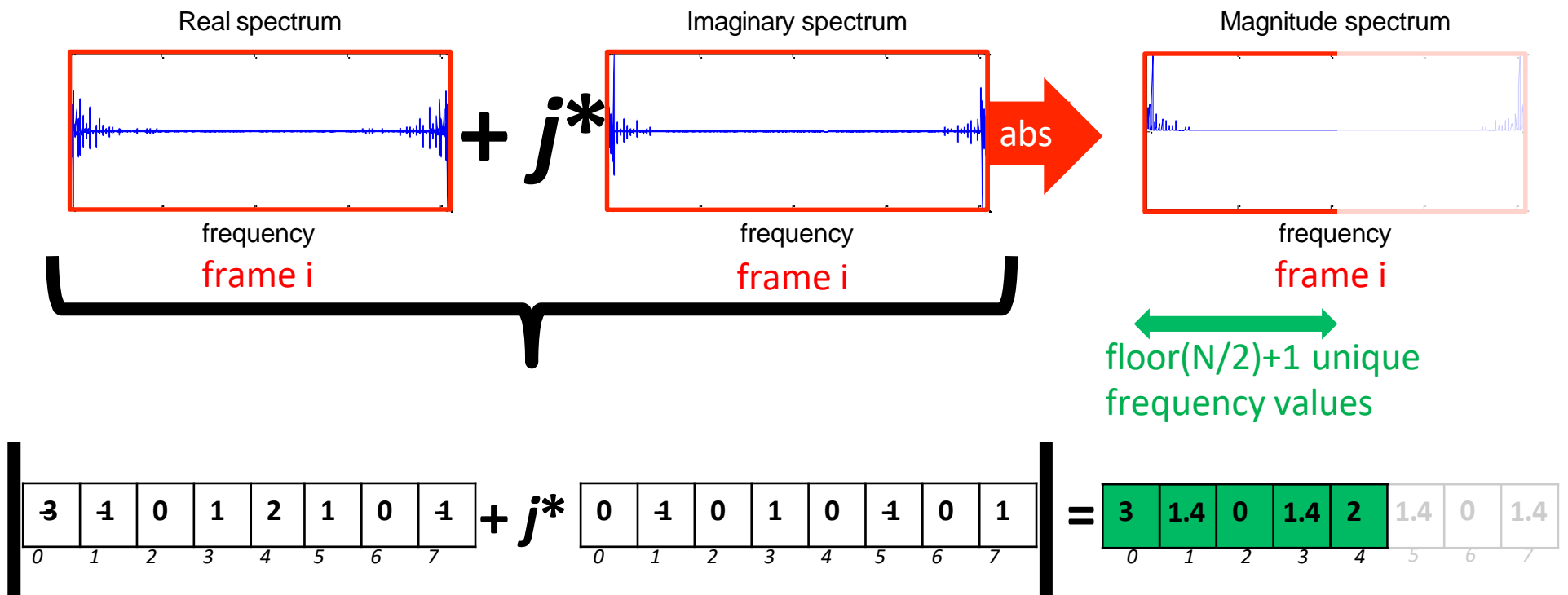
Spectrogram

- Frequency indices from $\text{floor}(N/2)+1$ to $N-1$ are the **mirrored frequency values**



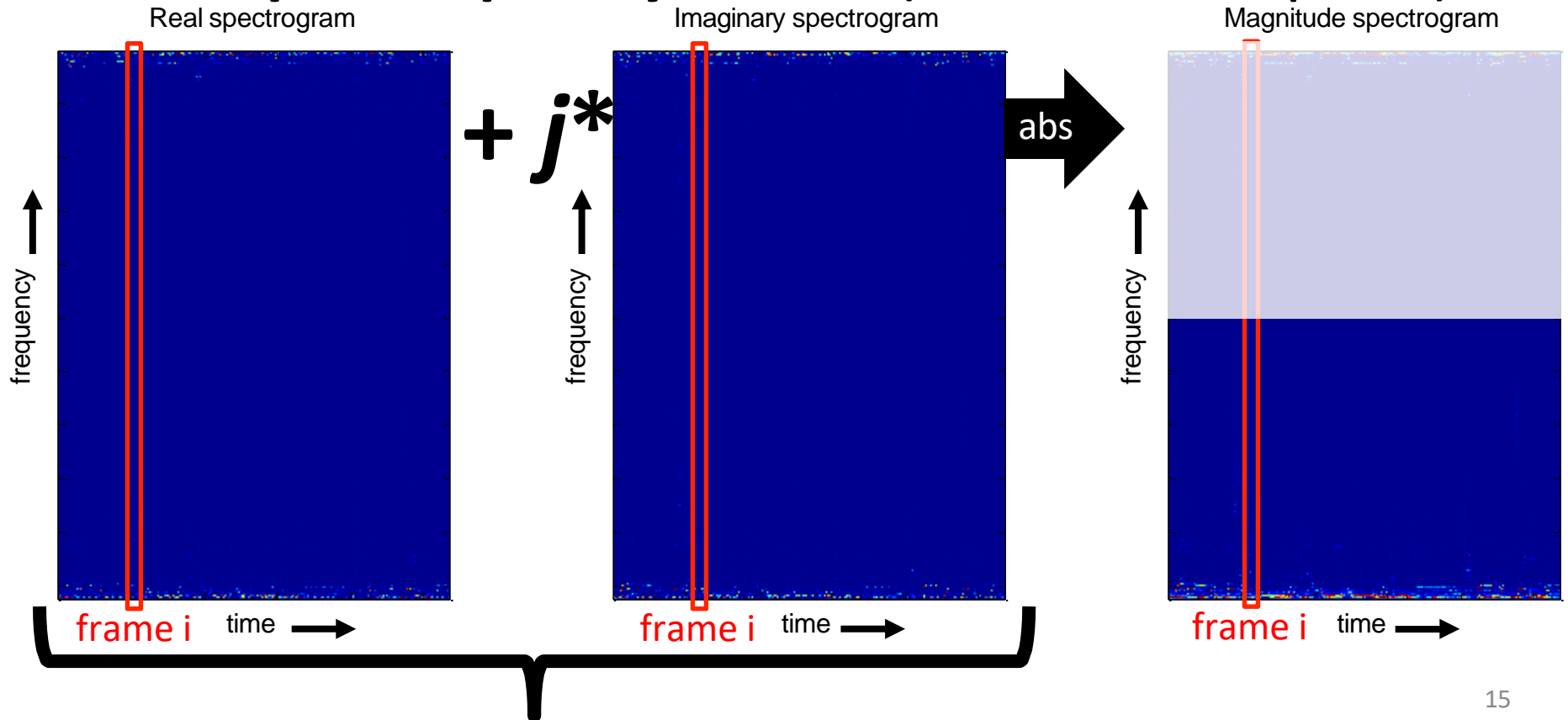
Spectrogram

- Since they are redundant, we can discard the frequency values from $\text{floor}(N/2)+1$ to $N-1$



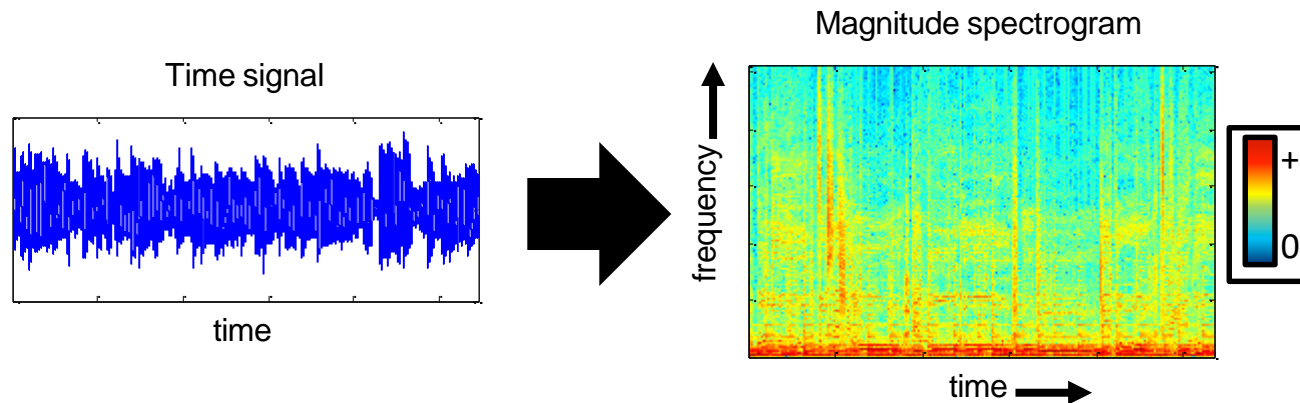
Spectrogram

- The spectrogram has therefore **$\text{floor}(N/2)+1$ unique frequency values** (with DC and pivot)



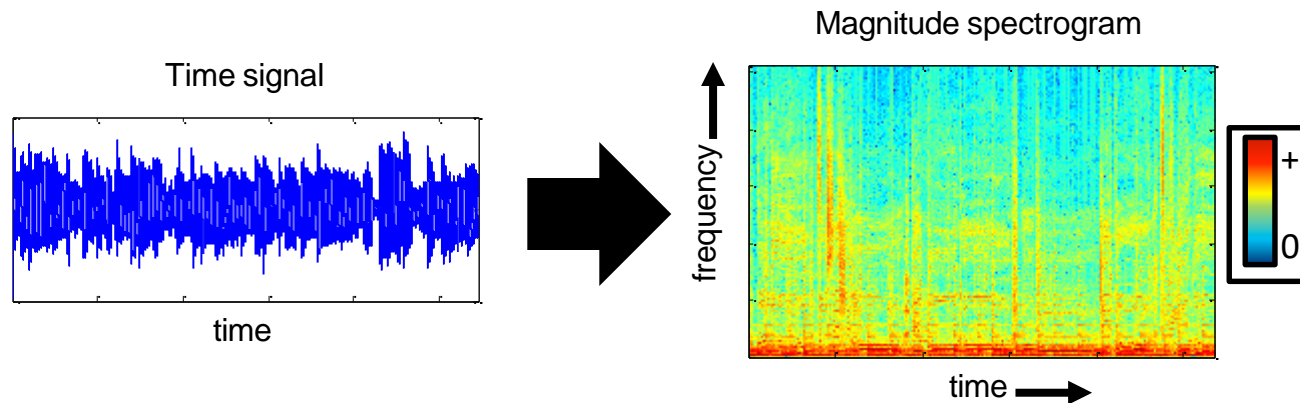
Spectrogram

- Why the magnitude spectrogram?
 - Easy to visualize (compare with the STFT)
 - Magnitude information more important
 - Human ear less sensitive to phase



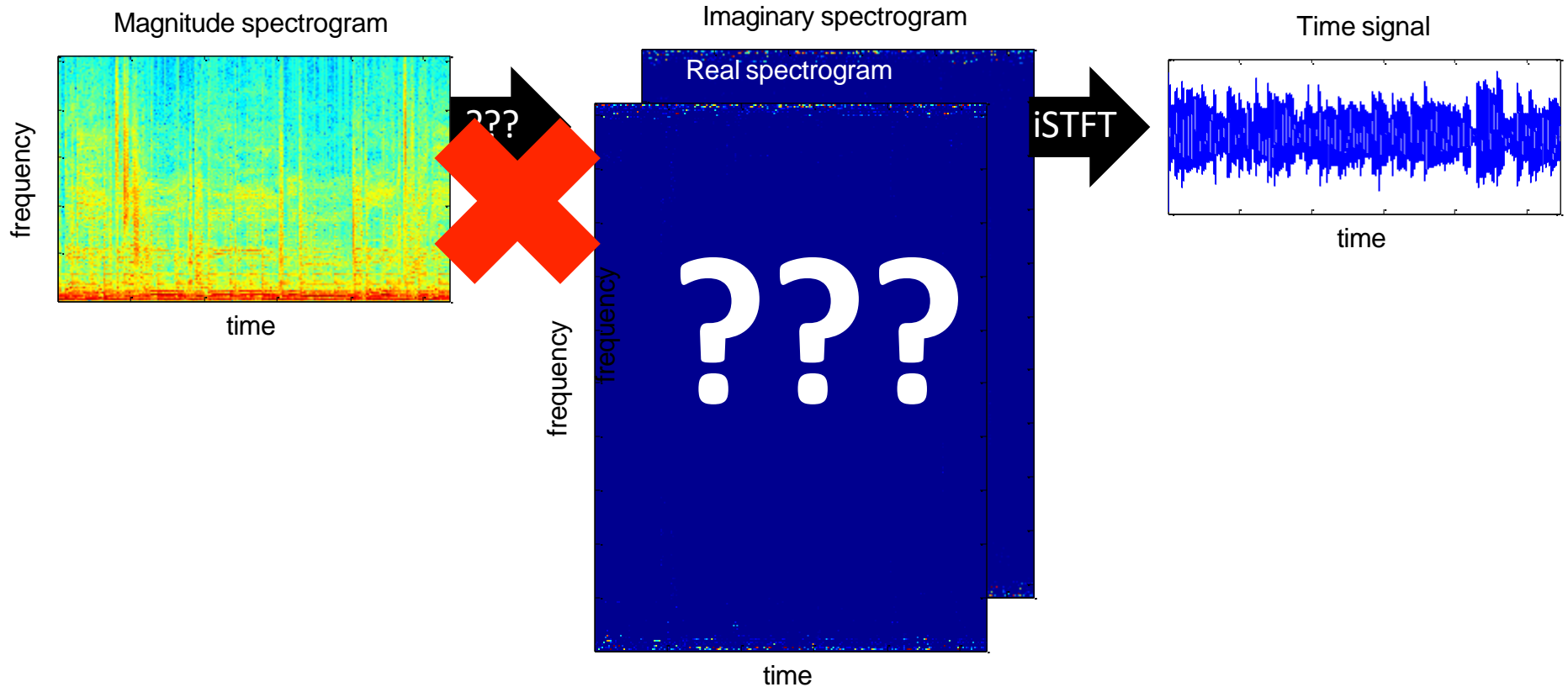
Spectrogram

- When you display a spectrogram in Matlab...
 - *imagesc*: data is scaled to use the full colormap
 - $10*\log_{10}(V)$: magnitude spectrogram in dB
 - *set(gca,'YDir','normal')*: y-axis from bottom to top



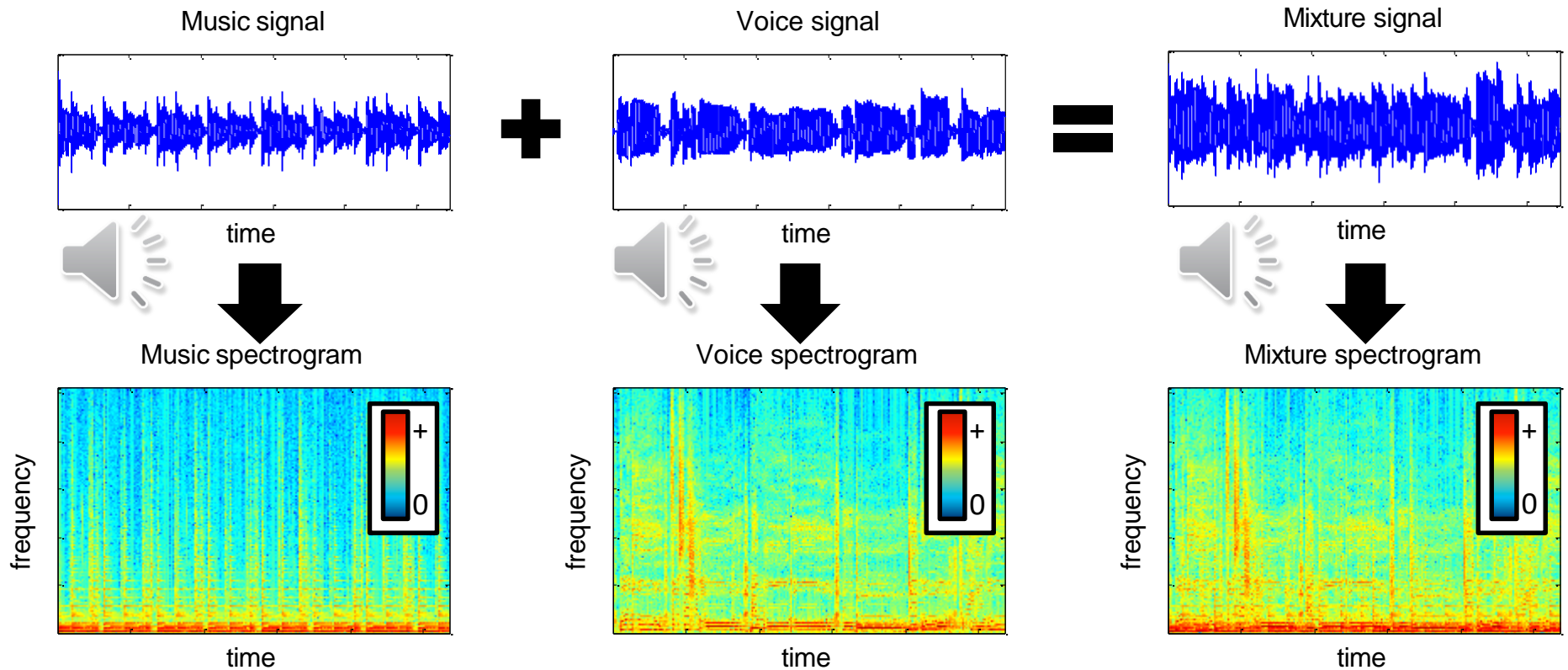
Spectrogram

- The signal **cannot be reconstructed** from the spectrogram (phase information is missing!)



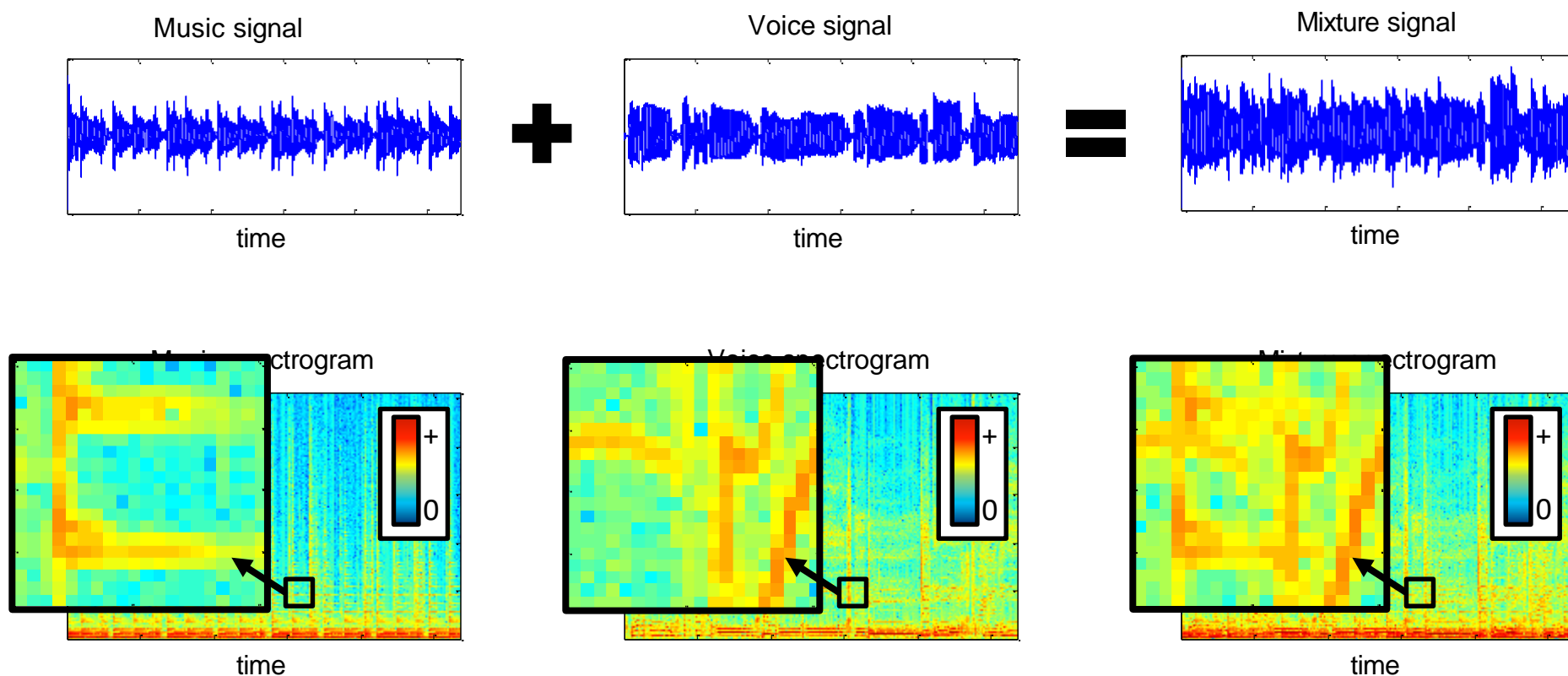
Time--frequency Masking

- Suppose we have a mixture of two sources:
a music signal and a voice signal



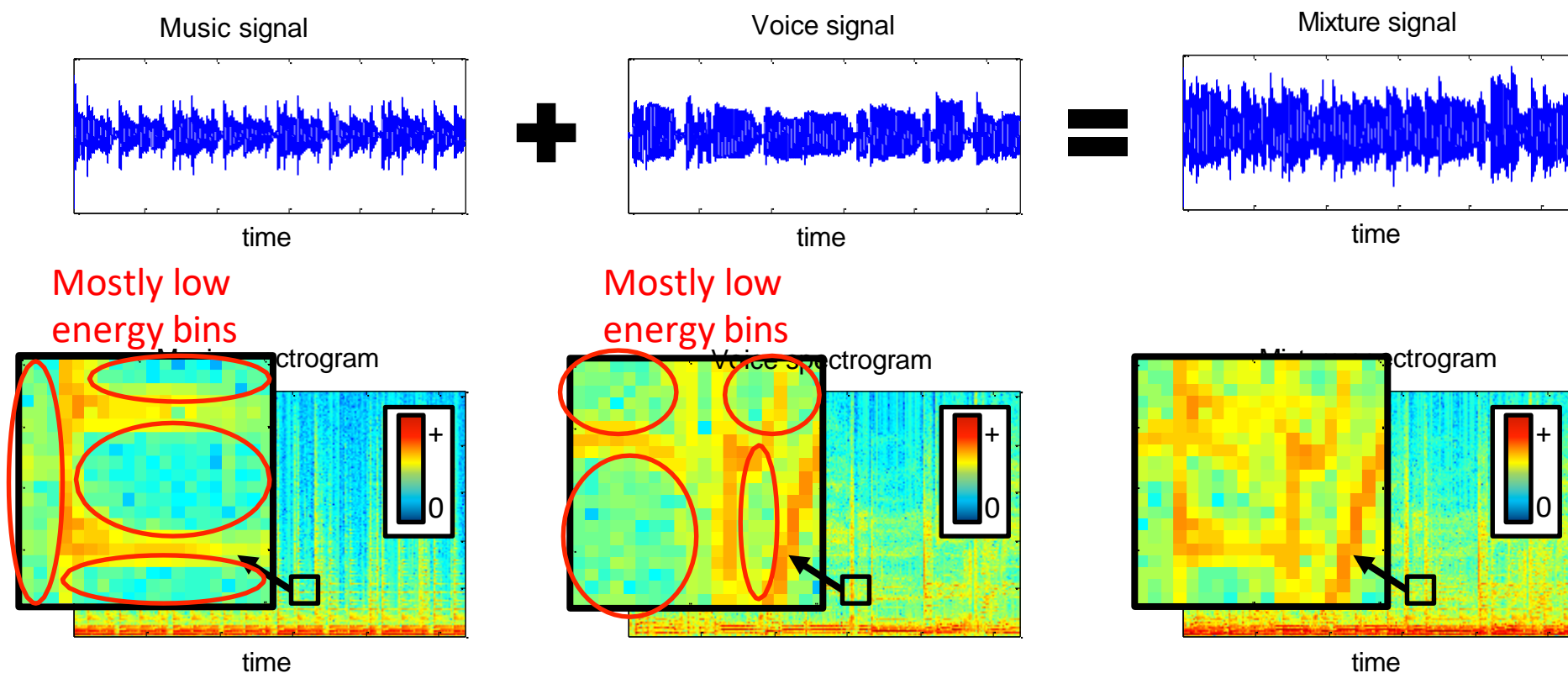
Time--frequency Masking

- We assume that the sources are **sparse** = most of the time--frequency bins have null energy



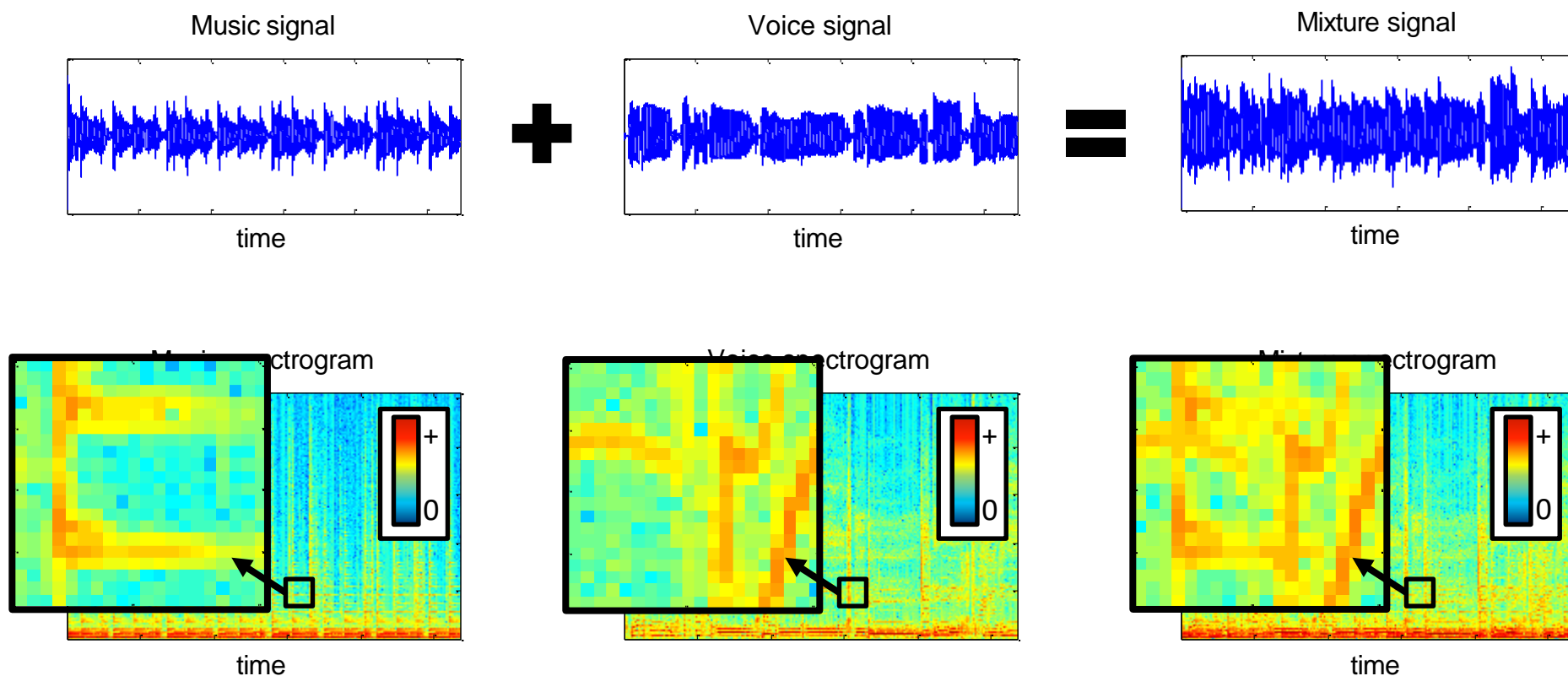
Time--frequency Masking

- We assume that the sources are **sparse** = most of the time--frequency bins have null energy



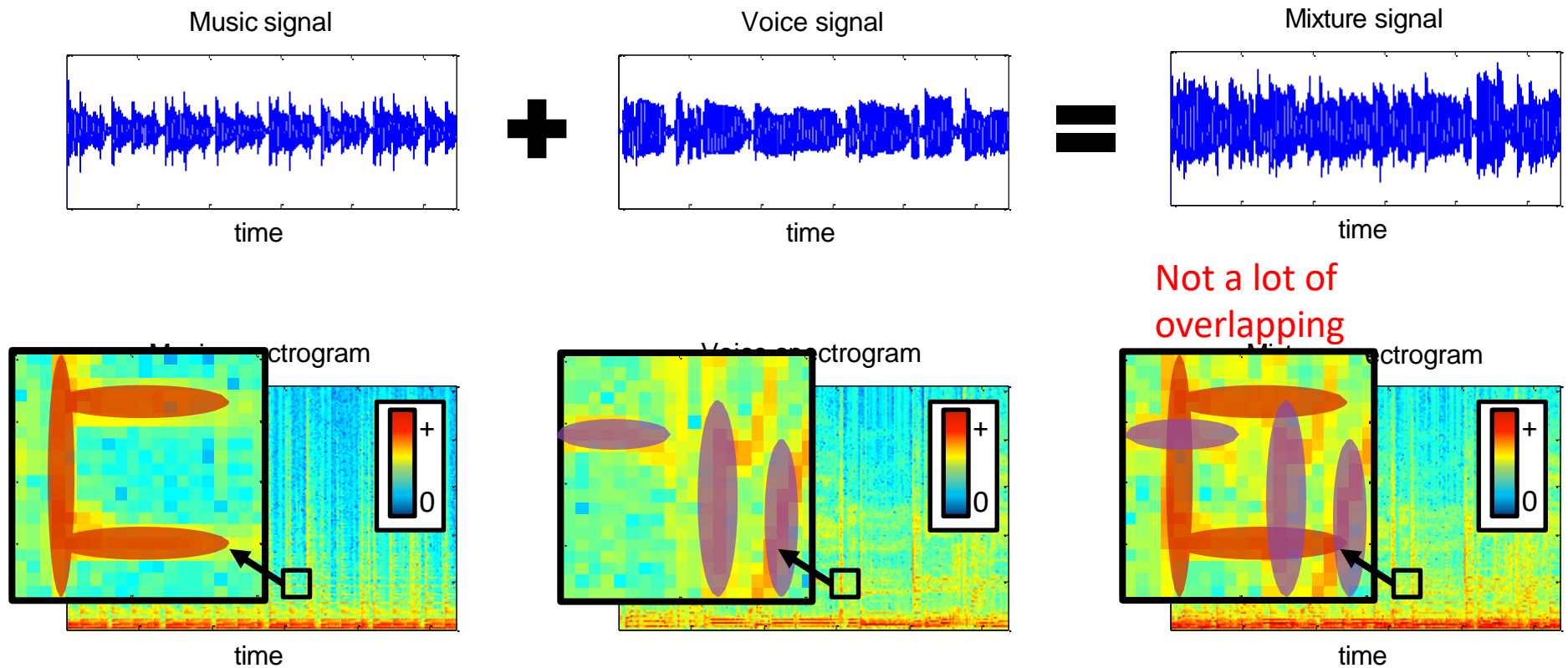
Time--frequency Masking

- We assume that the sources are **disjoint** = their time--frequency bins do not overlap



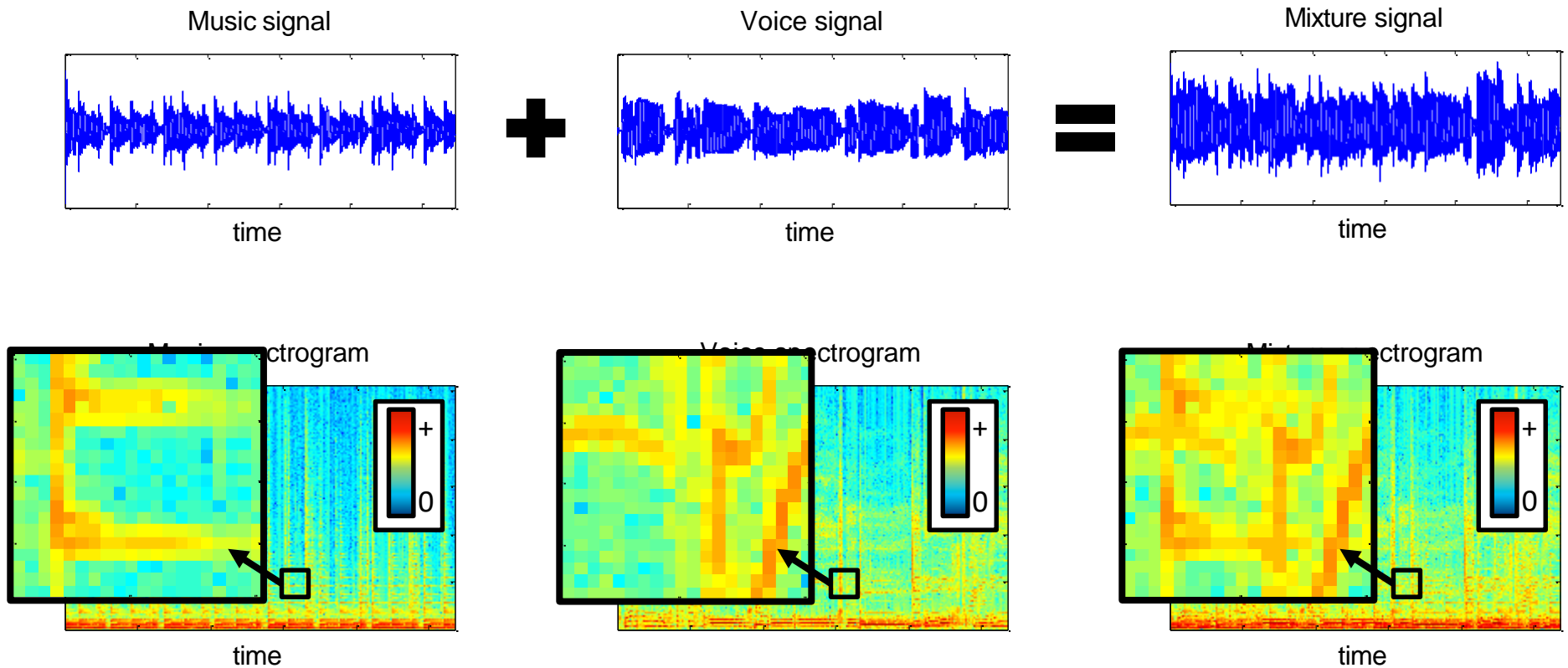
Time--frequency Masking

- We assume that the sources are **disjoint** = their time--frequency bins do not overlap



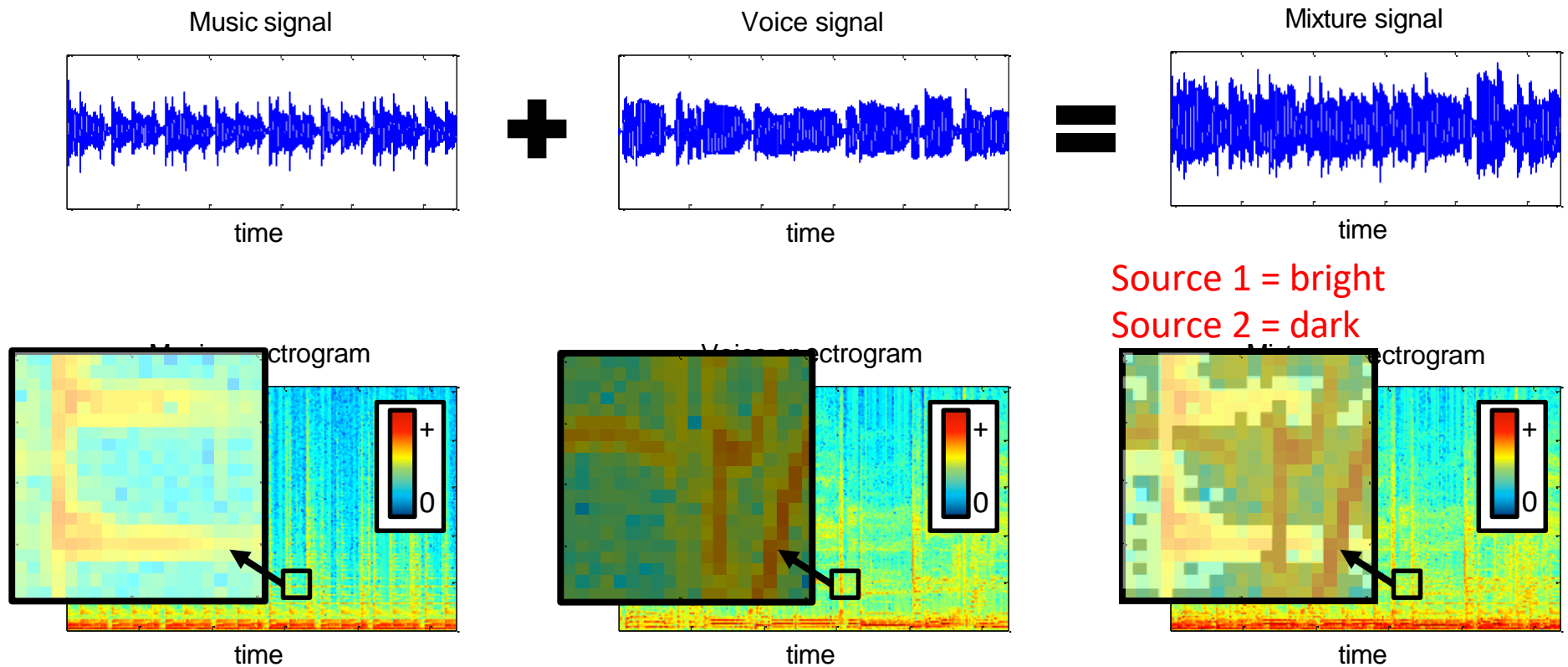
Time--frequency Masking

- Assuming sparseness and disjointness, we can **discriminate** the bins between mixed sources



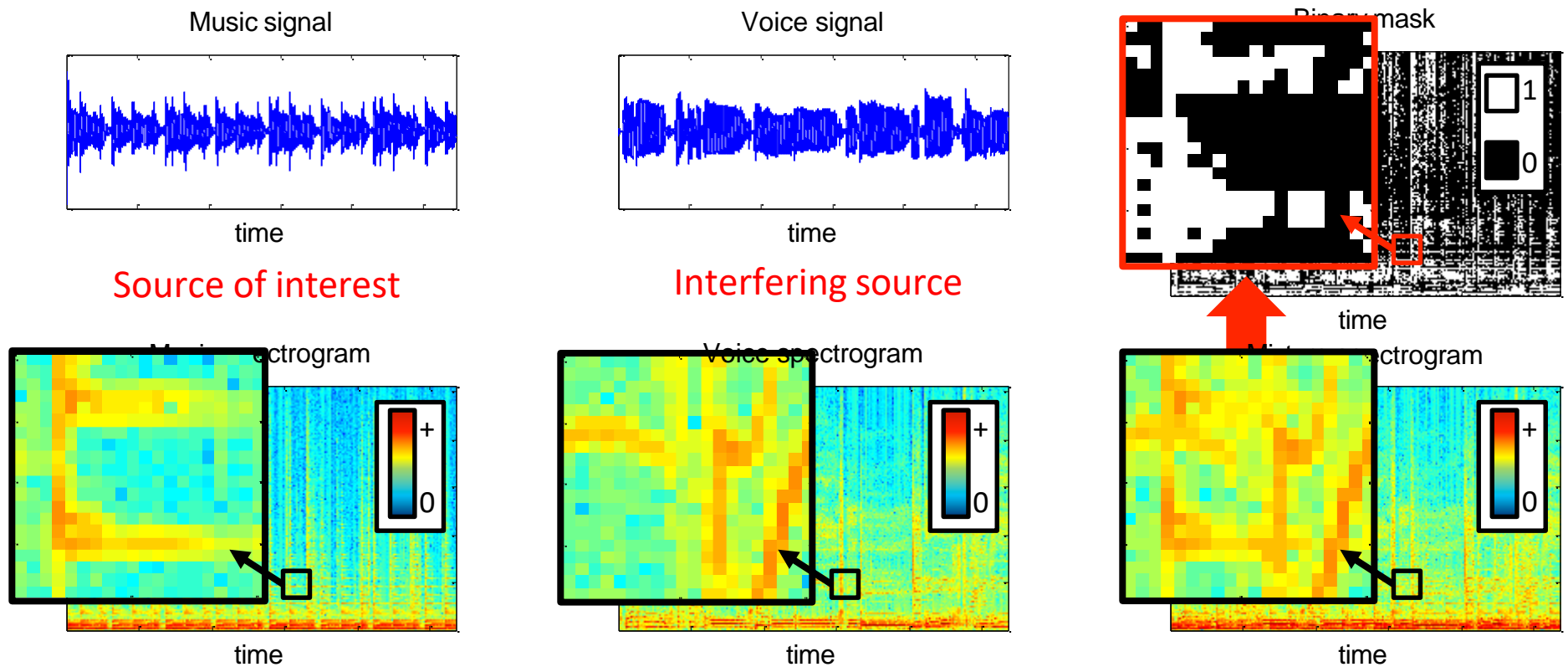
Time--frequency Masking

- Assuming sparseness and disjointness, we can **discriminate** the bins between mixed sources



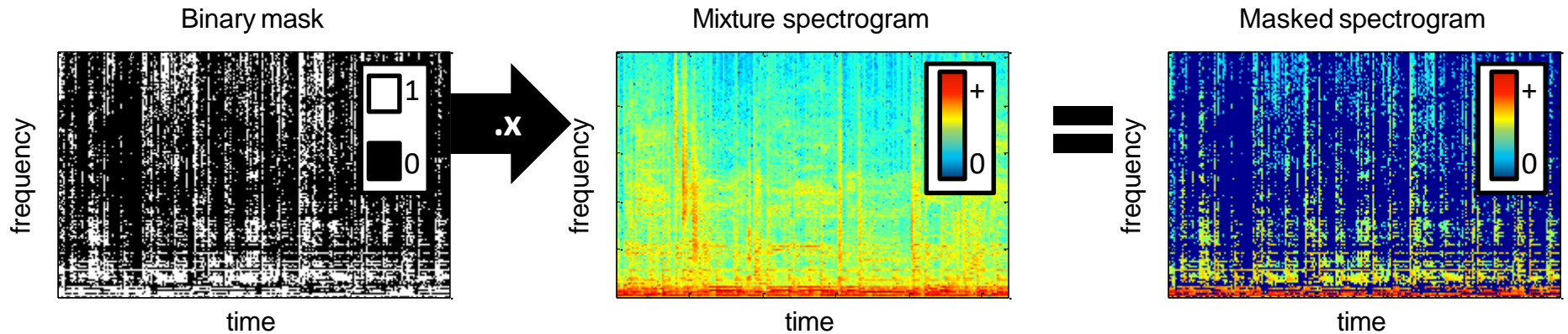
Time--frequency Masking

- Bins that are likely to belong to one source are assigned to 1, the rest to 0 = **binary masking**!



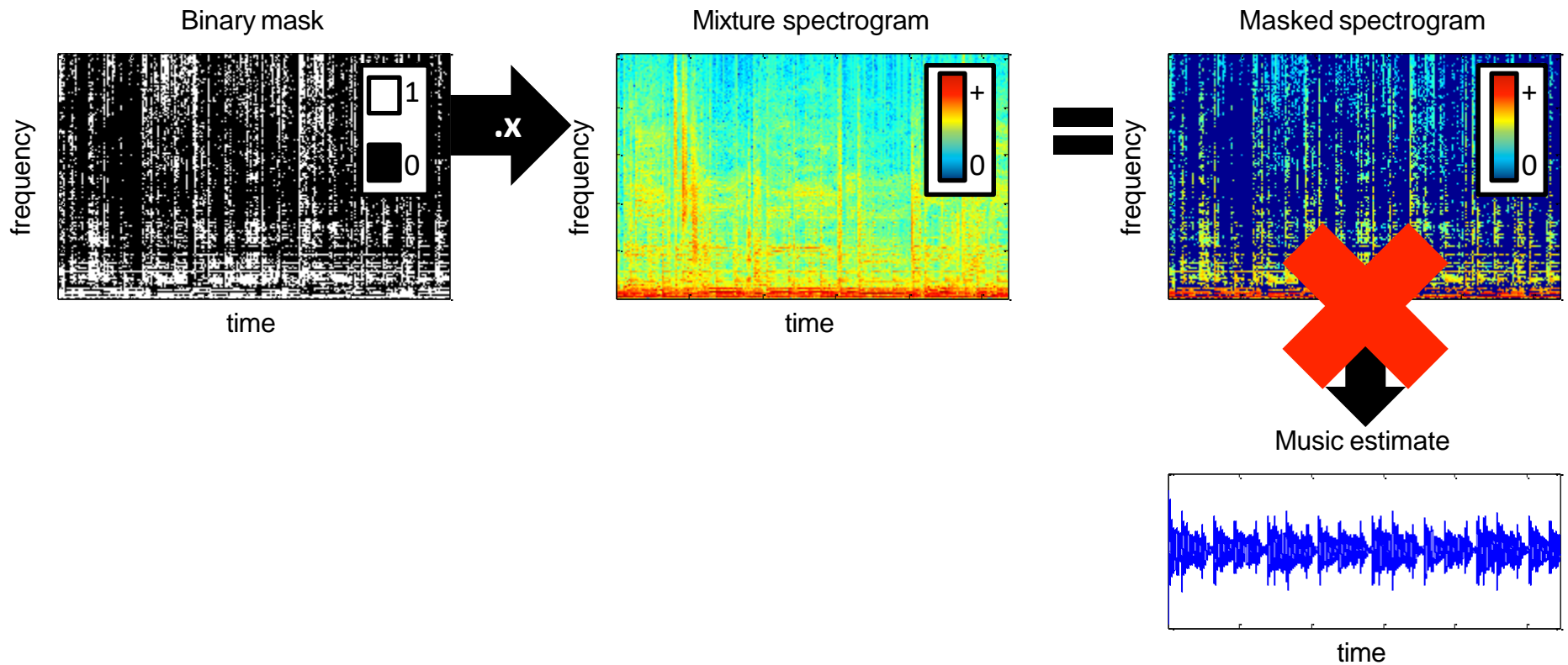
Time--frequency Masking

- By multiplying the binary mask to the mixture spectrogram, we can “preview” the estimate



Time--frequency Masking

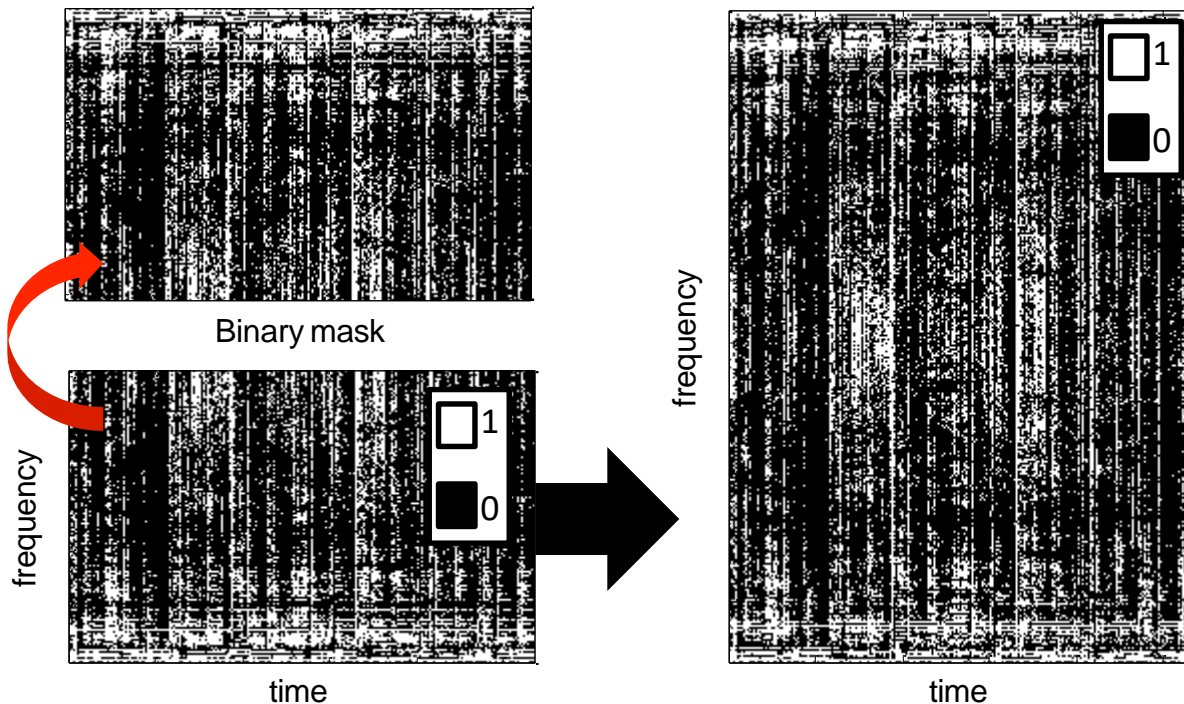
- However, we cannot derive the estimate itself because we cannot invert a spectrogram!



Time--frequency Masking

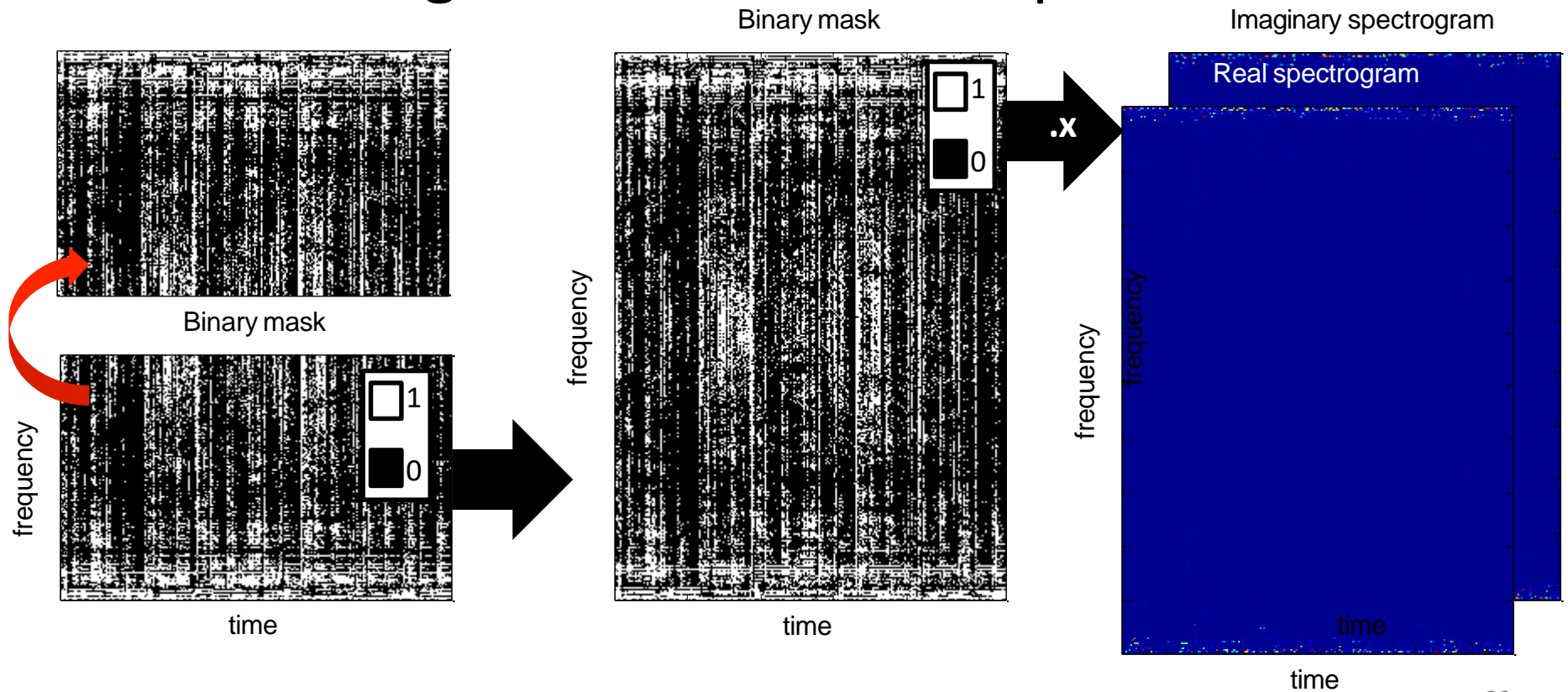
- We mirror the redundant frequencies from the unique frequencies (without DC and pivot)

Binary mask



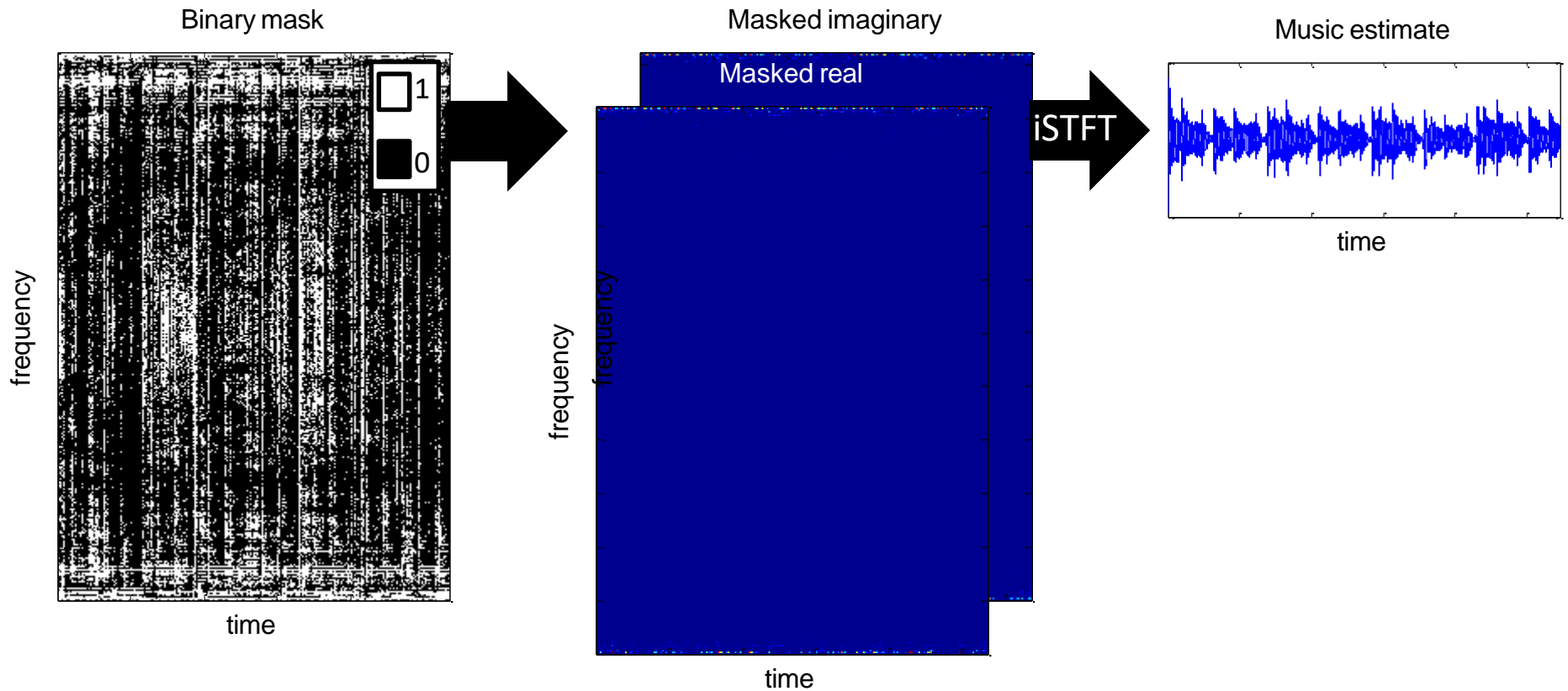
Time--frequency Masking

- We then apply this full binary mask to the STFT using a element-wise multiplication



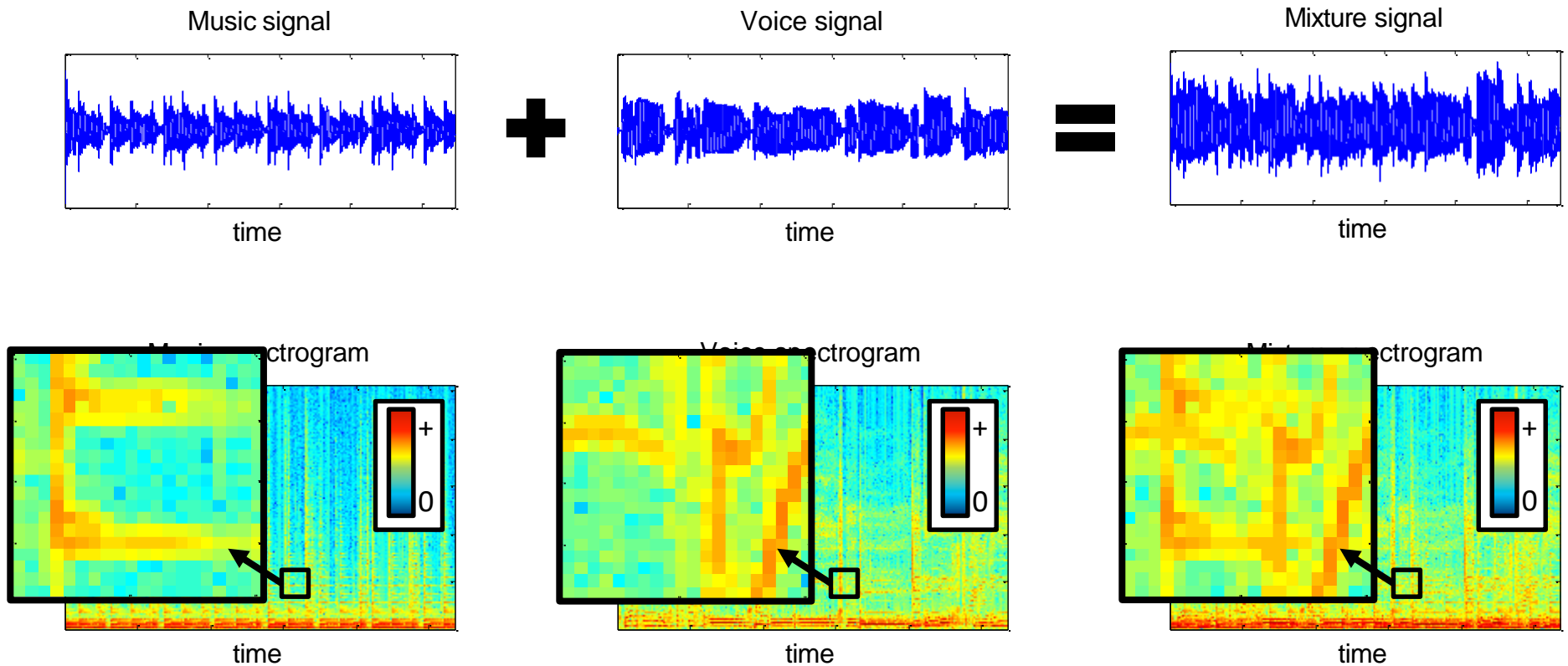
Time--frequency Masking

- The estimate signal can now be reconstructed via inverse STFT



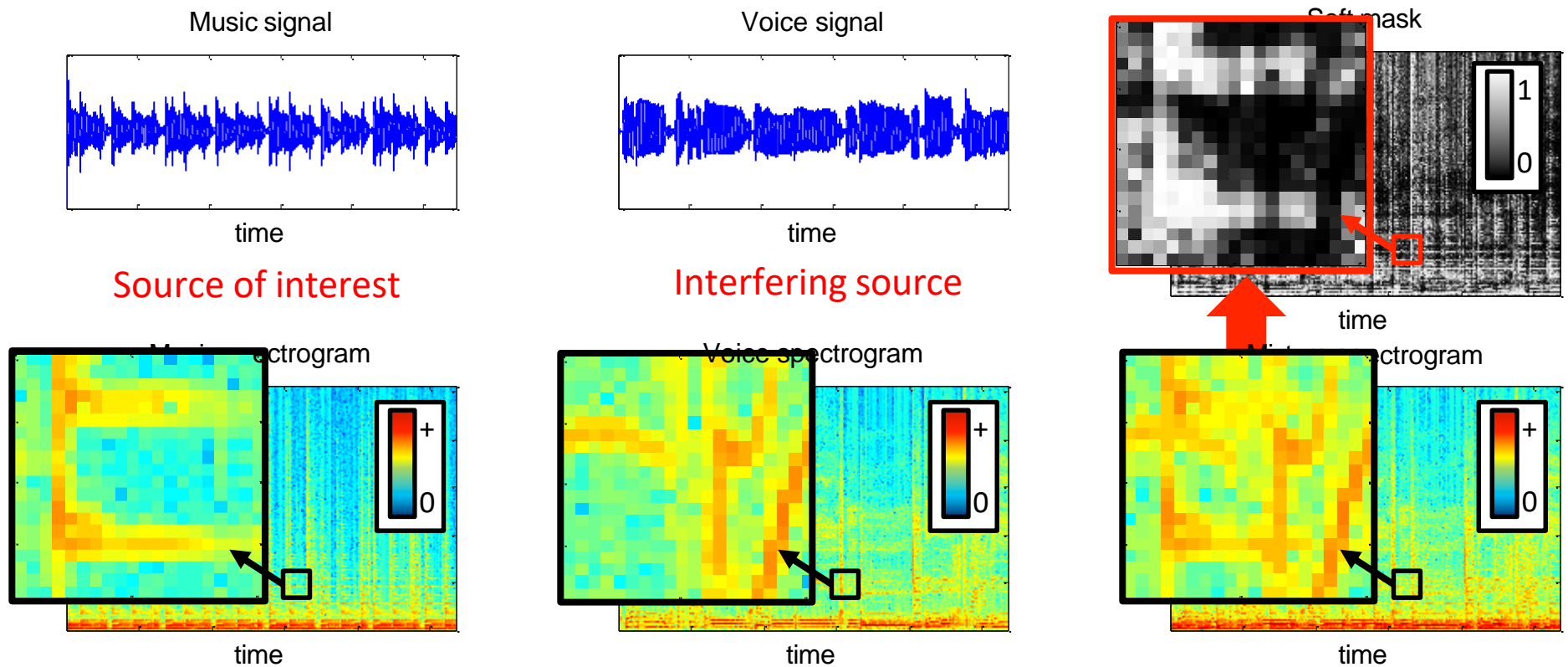
Time--frequency Masking

- Sources are not really sparse or disjoint in time--frequency in the mixture



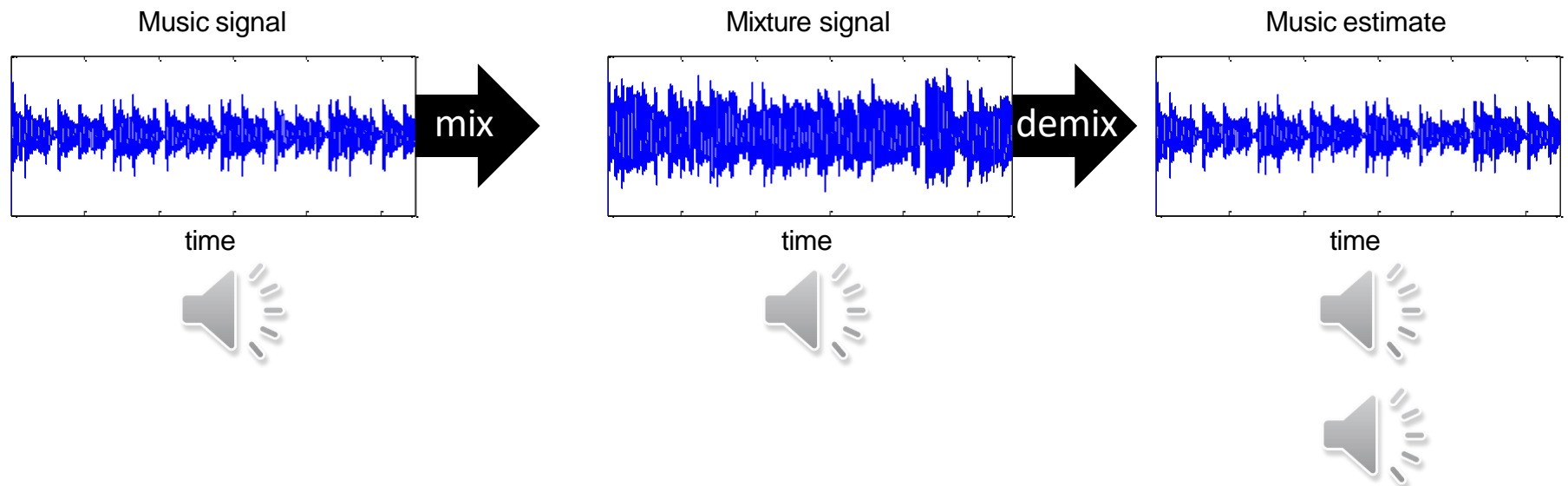
Time-frequency Masking

- Bins that are likely to belong to one source are close to 1, the rest close to 0 = **soft masking**!



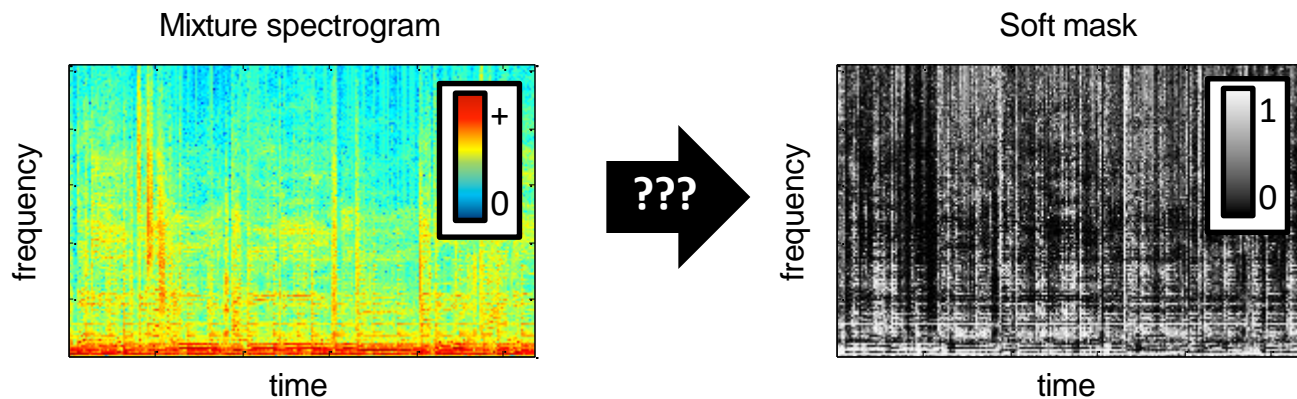
Time--frequency Masking

- Let's listen to the results!







Question

- How can we efficiently model a binary/soft time--frequency mask for source separation?...
- To be continued...



Deep learning for monaural speech separation

Outline

-  Introduction.....
-  Relation to previous work.....
-  Proposed methods.....
-  Experiments & Conclusion.....

1. Introduction

Monaural source separation is useful for many real-world applications though it is a challenging problem.

In this paper

- Study deep learning for monaural speech separation.

- Propose the joint optimization of the deep learning model (deep neural networks and recurrent neural networks) with an extra masking layer

- Explore a discriminative training criterion for the neural networks to further enhance the separation performance.

- Propose the joint optimization of the network with a soft masking function.

1. Introduction

Monaural source separation is useful for many real-world applications though it is a challenging problem.

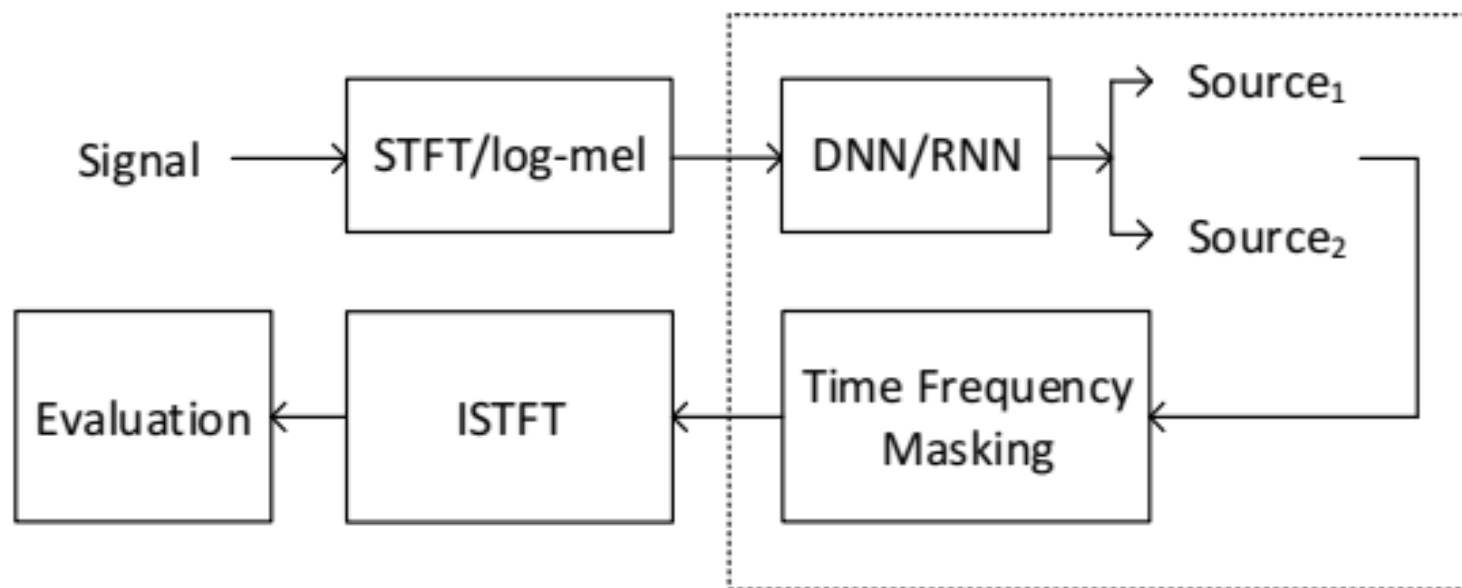


Fig. 1: Proposed framework

2. Relation to previous work

A 2-stage framework for predicting an ideal binary mask using deep neural networks was proposed.

Propose a general framework that can jointly train all feature dimensions at the same time using one neural network

Propose a method to jointly train the masking function with the network directly.

Proposed using an RNN for speech noise reduction in robust automatic speech recognition. Given the noisy signal \mathbf{x} , apply an RNN to learn clean speech \mathbf{y}

3.1 Architecture

Using a deep neural network and a recurrent neural network for learning the optimal hidden representations to reconstruct the target spectra.

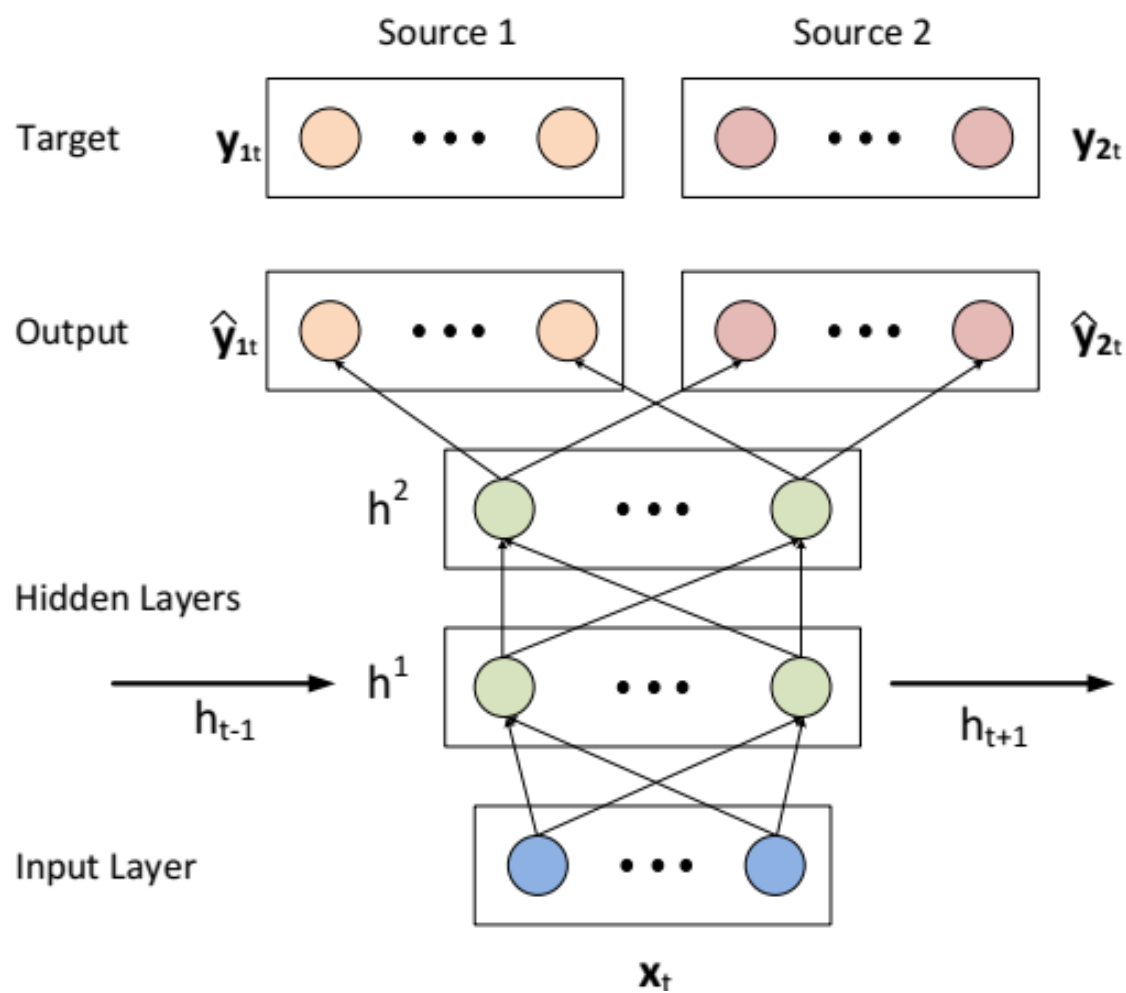
The hidden activation from the previous time step:

$$h^l(\mathbf{x}_t) = f(\mathbf{W}^l h^{l-1}(\mathbf{x}_t) + \mathbf{b}^l + \mathbf{U}^l h^l(\mathbf{x}_{t-1}))$$

The output layer is a linear layer and is computed as

$$\hat{\mathbf{y}}_t = \mathbf{W}^l h^{l-1}(\mathbf{x}_t) + \mathbf{c}$$

3.1. Architecture



3.2 Time-frequency Masking

Two commonly used masking functions are explored in this paper: binary (hard) time-frequency masking and soft time-frequency masking methods.

The bin

$$\mathbf{M}_b(f) = \begin{cases} 1 & |\hat{\mathbf{y}}_{1_t}(f)| > |\hat{\mathbf{y}}_{2_t}(f)| \\ 0 & \text{otherwise} \end{cases} \text{ /S:}$$

The soft time-frequency mask \mathbf{M}_s as follows:

$$\mathbf{M}_s(f) = \frac{|\hat{\mathbf{y}}_{1_t}(f)|}{|\hat{\mathbf{y}}_{1_t}(f)| + |\hat{\mathbf{y}}_{2_t}(f)|}$$

3.2 Time-frequency Masking

Time-frequency mask \mathbf{M} (\mathbf{M}_b or \mathbf{M}_s) is computed, it is applied to the spectra \mathbf{X}_t of the mixture \mathbf{x}_t to obtain the estimated separation spectra $\hat{\mathbf{s}}_{1t}$ and $\hat{\mathbf{s}}_{2t}$, which correspond to sources 1 and 2, as follows:

$$\begin{aligned}\hat{\mathbf{s}}_{1t}(f) &= \mathbf{M}(f) \mathbf{X}_t(f) \\ \hat{\mathbf{s}}_{2t}(f) &= (1 - \mathbf{M}(f)) \mathbf{X}_t(f)\end{aligned}$$

The binary mask \mathbf{f}

Propose the integration of the soft time-frequency masking function directly.

Add an extra layer to the original output of the neural network as follows:

3.2 Time-frequency Masking

The binary mask function is not smooth

Propose the integration of the soft time-frequency masking function directly.

Add an extra layer to the original output of the neural network as follows:

$$\begin{aligned}\tilde{\mathbf{y}}_{1_t} &= \frac{|\hat{\mathbf{y}}_{1_t}|}{|\hat{\mathbf{y}}_{1_t}| + |\hat{\mathbf{y}}_{2_t}|} \odot \mathbf{X}_t \\ \tilde{\mathbf{y}}_{2_t} &= \frac{|\hat{\mathbf{y}}_{2_t}|}{|\hat{\mathbf{y}}_{1_t}| + |\hat{\mathbf{y}}_{2_t}|} \odot \mathbf{X}_t\end{aligned}$$

3.3 Discriminative Training

The output predictions $\hat{\mathbf{y}}_{1t}$ and $\hat{\mathbf{y}}_{2t}$ (or $\tilde{\mathbf{y}}_{1t}$ and $\tilde{\mathbf{y}}_{2t}$) of the original sources \mathbf{y}_{1t} and \mathbf{y}_{2t} .

Optimize the neural network parameters by minimizing the squared error.

$$||\hat{\mathbf{y}}_{1t} - \mathbf{y}_{1t}||_2^2 + ||\hat{\mathbf{y}}_{2t} - \mathbf{y}_{2t}||_2^2$$

Minimizing is equivalent to increasing the similarity between the prediction and the target

3.3 Discriminative Training

One of the goals is to have a high signal to interference ratio (SIR). Propose a discriminative objective function that takes into account the similarity between the prediction and other sources, and between the prediction and the current target.

$$\|\hat{\mathbf{y}}_{1_t} - \mathbf{y}_{1_t}\|_2^2 - \gamma \|\hat{\mathbf{y}}_{1_t} - \mathbf{y}_{2_t}\|_2^2 + \|\hat{\mathbf{y}}_{2_t} - \mathbf{y}_{2_t}\|_2^2 - \gamma \|\hat{\mathbf{y}}_{2_t} - \mathbf{y}_{1_t}\|_2^2$$

4. Experiments

Features

Evaluate the performance of the proposed approaches for monaural speech separation using the TIMIT corpus.

Input signal: sentences from a male and a female speaker.

The log-mel filterbank is found to provide better results compared to mel-frequency cepstral coefficients (MFCC) and log FFT bins.

A 32 ms window with a 16 ms frame shift performs the best.

The input frame rate corresponds to the output spectra which are extracted using a 512-point STFT.

4. Experiments

The source separation evaluation is measured by using three quantitative values:

- Source to Interference Ratio (SIR)

- Source to Artifacts Ratio (SAR)

- Source to Distortion Ratio (SDR)

Use the standard NMF with the generalized KL divergence metric using 512-point and 1024-point STFT as baselines.

First train a set of basis vectors, \mathbf{W}_m , \mathbf{W}_f from male and female training data.

After solving coefficients, \mathbf{H}_m and \mathbf{H}_f , the binary and soft time-frequency masking functions are applied to the predicted magnitude spectrogram

4. Experiments

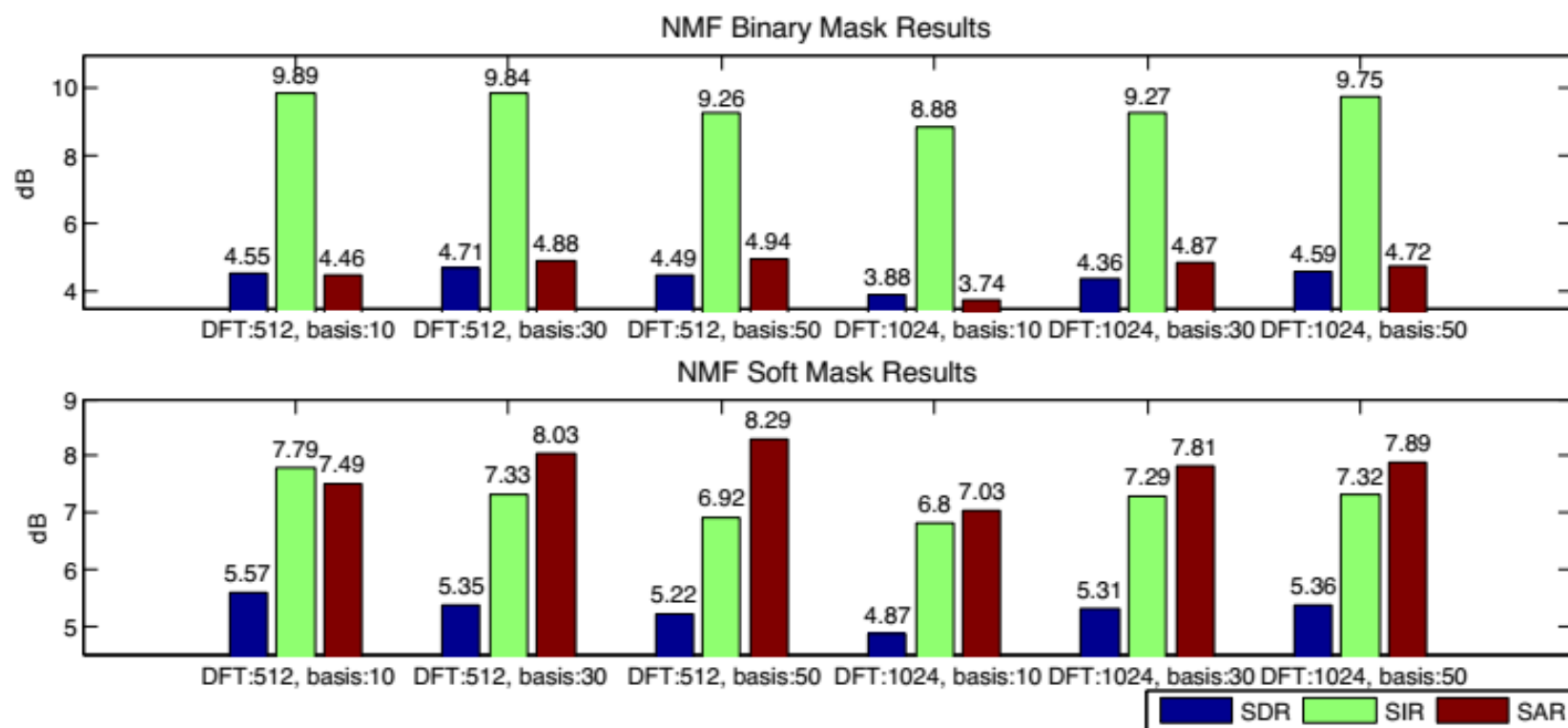


Fig. 3: NMF results with the 512-point and 1024-point STFT and basis vector sizes (10, 30, 50) using binary and soft time-frequency masking

4. Experiments

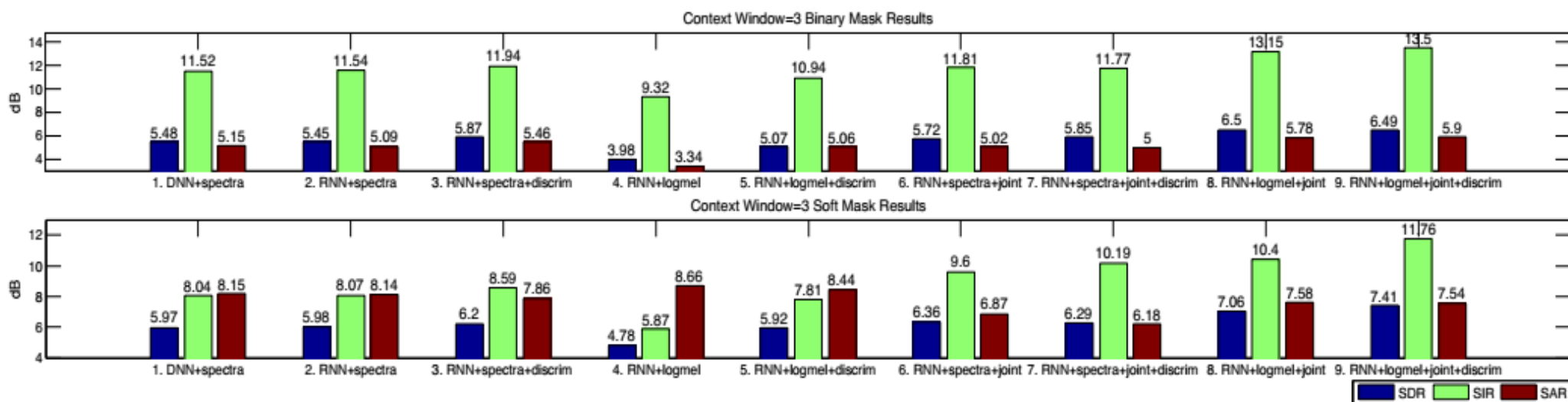


Fig. 4: Neural network results with concatenating neighboring 1 frame as input, where “joint” indicates the joint training between the network and the soft masking function, and “discrim” indicates the training with discriminative objectives

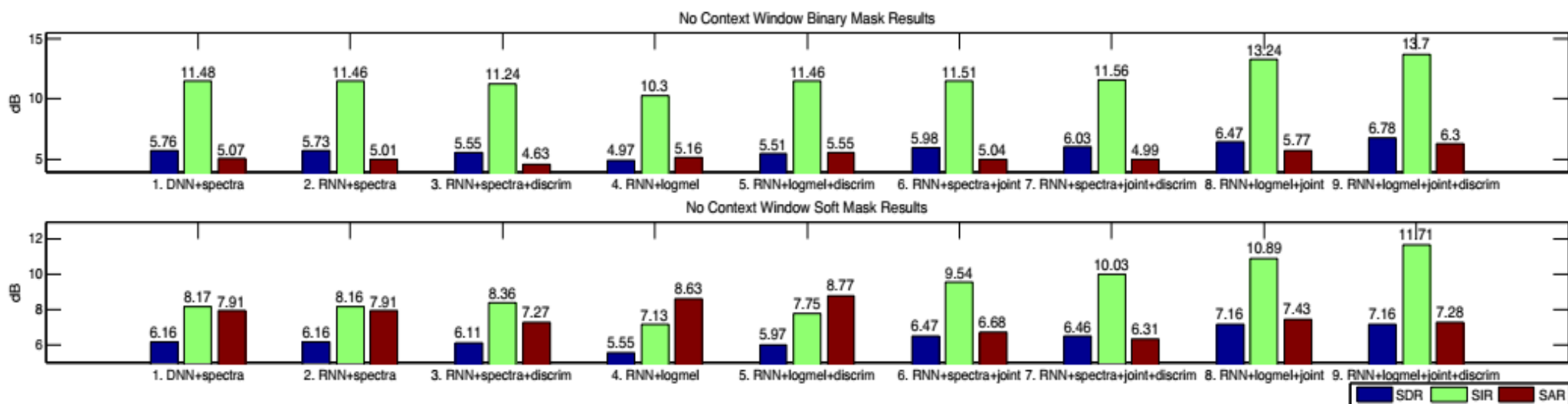


Fig. 5: Neural network results without concatenating neighboring frames as input, where “joint” indicates the joint training between the network and the soft masking function, and “discrim” indicates the training with discriminative objectives

4. Experiments

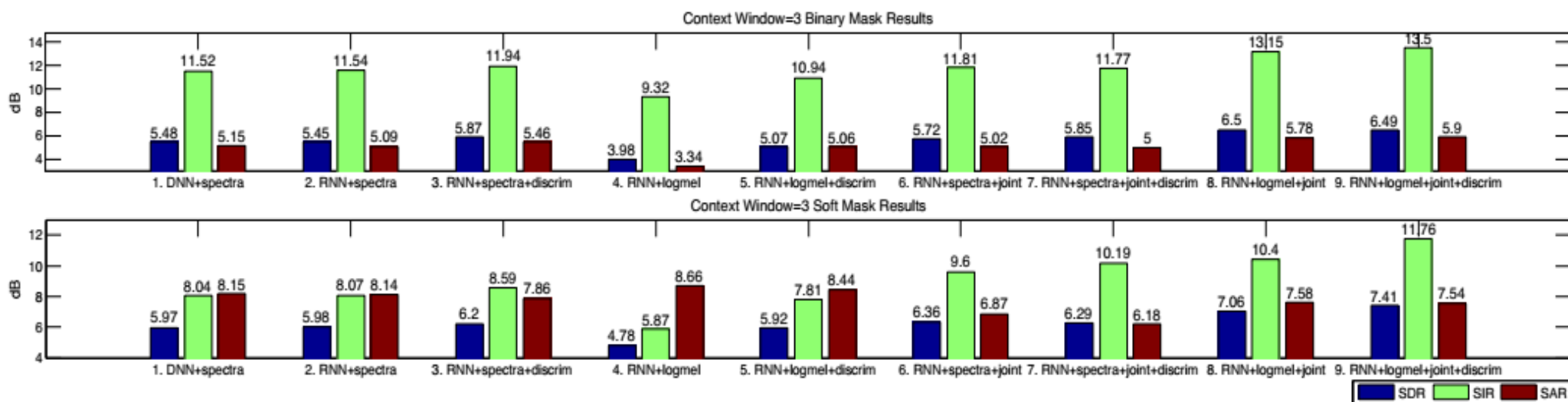


Fig. 4: Neural network results with concatenating neighboring 1 frame as input, where “joint” indicates the joint training between the network and the soft masking function, and “discrim” indicates the training with discriminative objectives

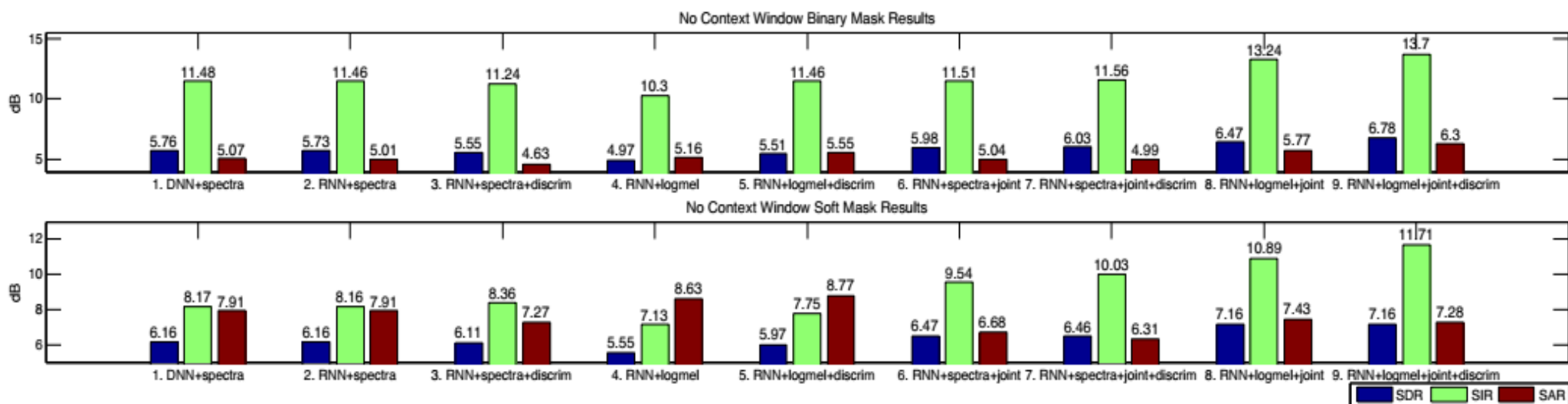


Fig. 5: Neural network results without concatenating neighboring frames as input, where “joint” indicates the joint training between the network and the soft masking function, and “discrim” indicates the training with discriminative objectives

4. Experiments

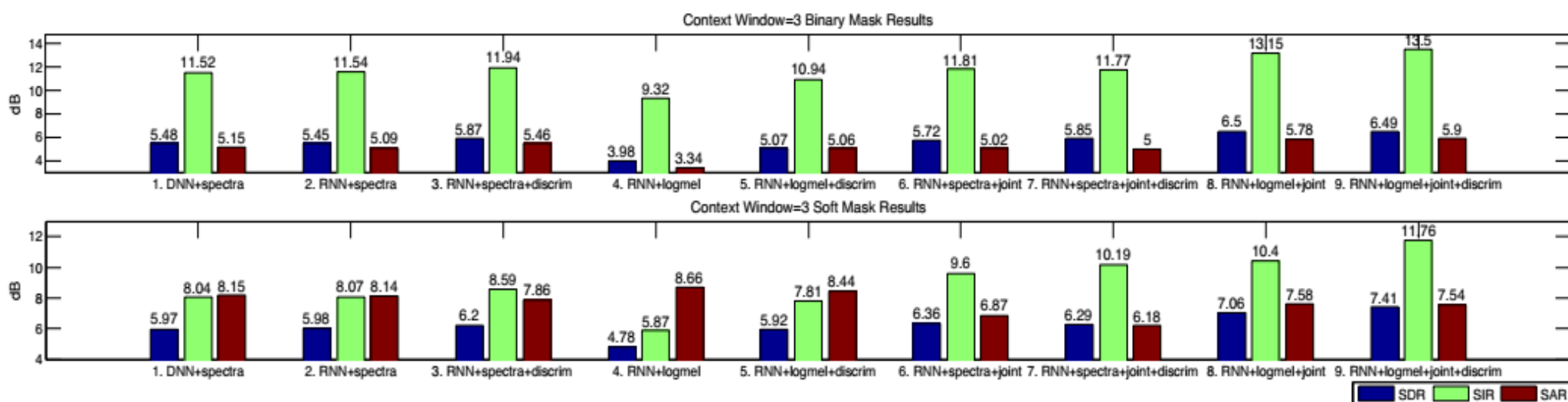


Fig. 4: Neural network results with concatenating neighboring 1 frame as input, where “joint” indicates the joint training between the network and the soft masking function, and “discrim” indicates the training with discriminative objectives

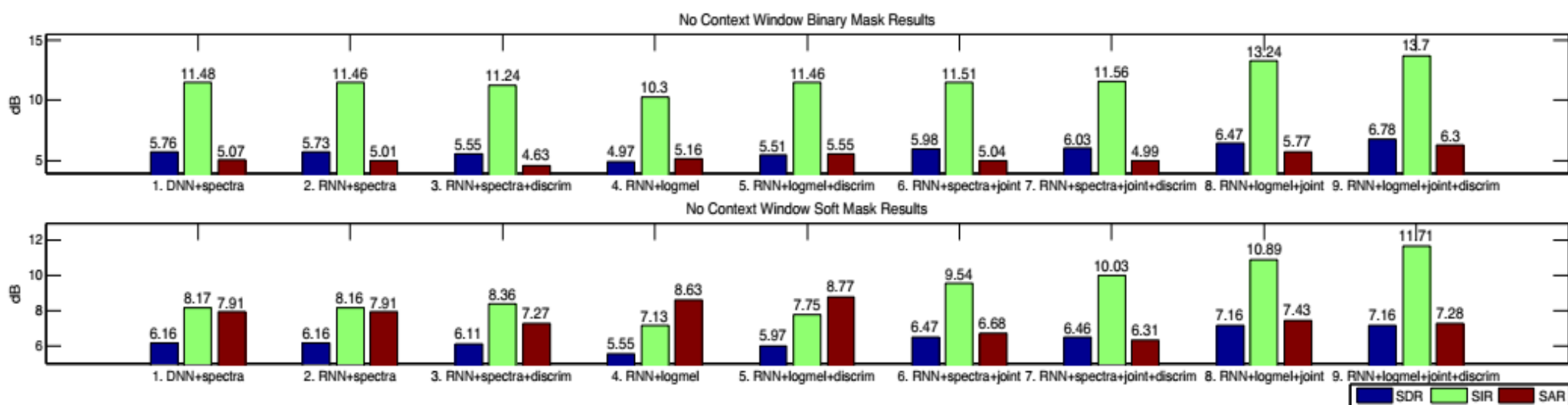


Fig. 5: Neural network results without concatenating neighboring frames as input, where “joint” indicates the joint training between the network and the soft masking function, and “discrim” indicates the training with discriminative objectives

5. Conclusion

In this work, proposed using deep learning models for monaural speech separation

Specifically, proposed the joint optimization of a soft masking function and deep learning models (DNNs and RNNs).

- Improve the SIR

- Achieve 3.8 ~4.9 dB SIR (Signal to Interference Ratio) gain compared to the NMF baseline

- Better SDRs (source to distortion ratio) and SARs (Source to artifact ratio)