Dolby

# Upsampling artifacts in neural audio synthesis

_

**Jordi Pons** (@jordiponsdotme – www.jordipons.me)

work with Santiago Pasqual, Giulio Cengarle and Joan Serrà

arxiv.org/abs/1703.09452

MUSIC SOURCE SEPARATION IN THE WAVEFORM DOMAIN

**Alexandre Défossez**
Facebook AI Research
INRIA
École Normale Supérieure
PSL Research University
defossez@fb.com

**Nicolas Usunier**
Facebook AI Research
usunier@fb.com

**Léon Bottou**
Facebook AI Research
leonb@fb.com

**Francis Bach**
INRIA
École Normale Supérie
PSL Research Univers
francis.bach@er

Source sepa

**MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis**

**Kundan Ku**
Lyrebird A

**Rithesh Kumar***
Lyrebird AI
hesh@descript.com

**Jose Sotelo**
Lyrebird AI, Mila

Program Co-director

**SEGAN: Speech Enhancement Generative Adversarial Network**

*Santiago Pascual*[1], *Antonio Bonafonte*[1], *Joan Serrà*[2]

[1]Universitat Politècnica de Catalunya, Barcelona, Spain
[2]Telefónica Research, Barcelona, Spain

santi.pascual@upc.edu, antonio.bonafonte@upc.edu joan.serra@telefonica.com

**Abstract**

Current speech enhancement techniqu
domain and/

ch enhancement [17]. However, fur-
t significant improvements of speech
when a clean phase spectrum is
deep network

Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation

Yi Luo, Nima Mesgarani

*Abstract*—Single-channel, s
ion methods have recent
e accuracy, latency, and c
main insufficient. The maj
rmulated the separation p
epresentation of the mixed signal,
the decoupling of the phase and magnitude of the
accuracy representation for speech
ograms.

**WAVE-U-NET: A MULTI-SCALE NEURAL NETWORK FOR END-TO-END AUDIO SOURCE SEPARATION**

**Daniel Stoller**
Queen Mary University of London
d.stoller@qmul

**Sebastian Ewert**

**Simon D**
Queen Mary Unive
s.e.dixon@qm

method and the mask estimation
lated using the inver

vestigate end-to-end source
which all

ance dependant on hyper-

ch has several limitations.
on many parameters, such
frequency resolution.

What do these architectures have in common?

MUSIC SOURCE SEPARATION IN THE WAVEFORM DOMAIN

**Alexandre Défossez**
Facebook AI Research
INRIA
École Normale Supérieure
PSL Research University
defossez@fb.com

**Nicolas Usunier**
Facebook AI Research
usunier@fb.com

**Léon Bottou**
Facebook AI Research
leonb@fb.com

**Francis Bach**
INRIA
École Normale Supérie
PSL Research Univers
francis.bach@er

Source sepa

**MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis**

**Kundan Ku**
Lyrebird A

**Rithesh Kumar***
Lyrebird AI
hesh@descript.com

**Jose Sotelo**
Lyrebird AI, Mila

Program Co-director

**SEGAN: Speech Enhancement Generative Adversarial Network**

*Santiago Pascual*[1], *Antonio Bonafonte*[1], *Joan Serrà*[2]

[1]Universitat Politècnica de Catalunya, Barcelona, Spain
[2]Telefónica Research, Barcelona, Spain

santi.pascual@upc.edu, antonio.bonafonte@upc.edu joan.serra@telefonica.com

**Abstract**

Current speech enhancement techniqu
domain and/

ch enhancement [17]. However, fur-
t significant improvements of speech
when a clean phase spectrum is
deep network

Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation

Yi Luo, Nima Mesgarani

*Abstract*—Single-channel, s
ion methods have recent
e accuracy, latency, and c
main insufficient. The maj
rmulated the separation p
epresentation of the mixed signal,
the decoupling of the phase and magnitude of the
representation for speech
spectrograms.

accuracy of
method and the mask estimation
lated using the inver

**WAVE-U-NET: A MULTI-SCALE NEURAL NETWORK FOR END-TO-END AUDIO SOURCE SEPARATION**

**Daniel Stoller**
Queen Mary University of London
d.stoller@qmu

**Sebastian Ewert**

**Simon D**
Queen Mary Unive
s.e.dixon@qm

ch has several limitations.
on many parameters, such
frequency resolution.

vestigate end-to-end source
which all

ance dependant on hyper-
estimation

p of audio frames, which can affe

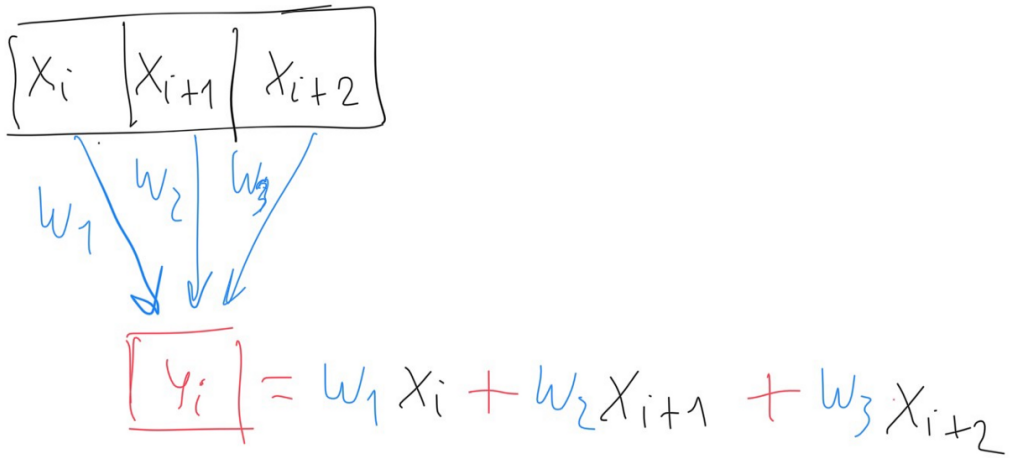
They use neural upsamplers

# Main upsamplers:

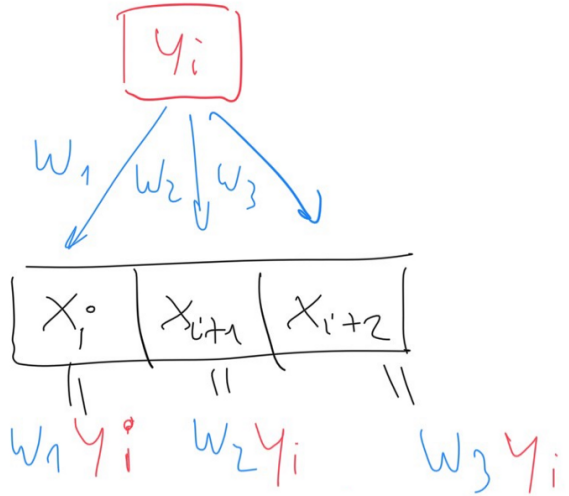
**Transposed convolutions**

- Widely used

**Convolutions** (“collapse”)




**Transposed convolutions** (“expand”)




SOURCE: MEDIUM.COM/@SANTI.PDP/HOW-PYTORCH-TRANSPOSED-CONVS1D-WORK-A7ADAC63C4A5

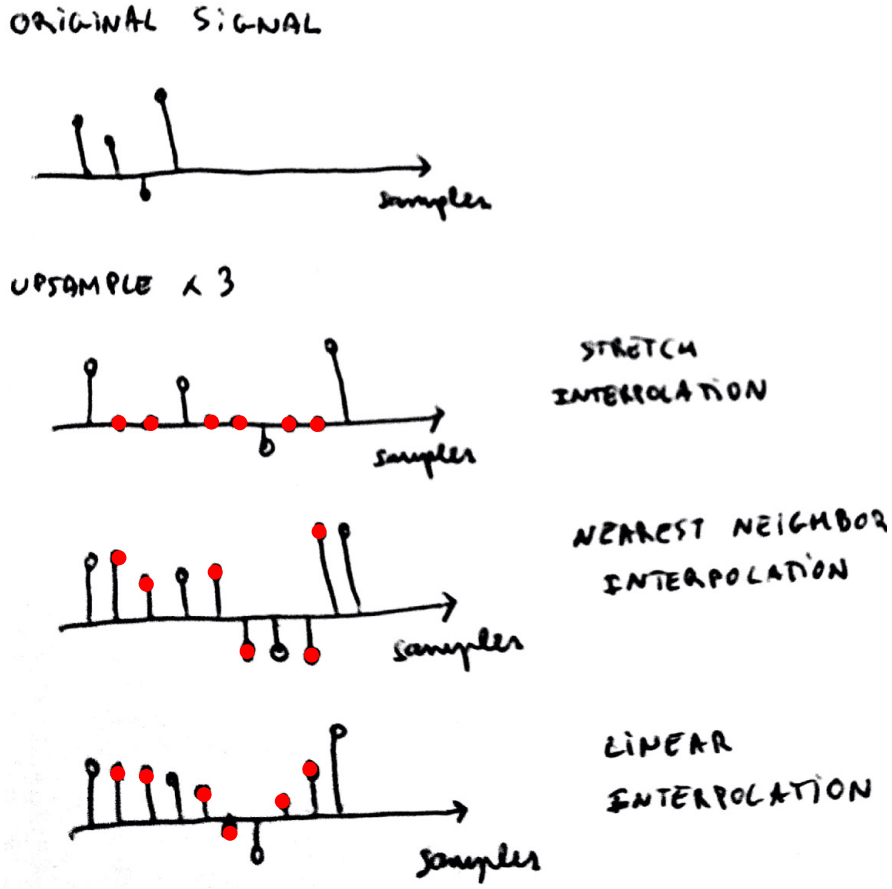
# Main upsamplers:

**Transposed convolutions**

- Widely used

**Interpolation + convolution**

- Often-times used



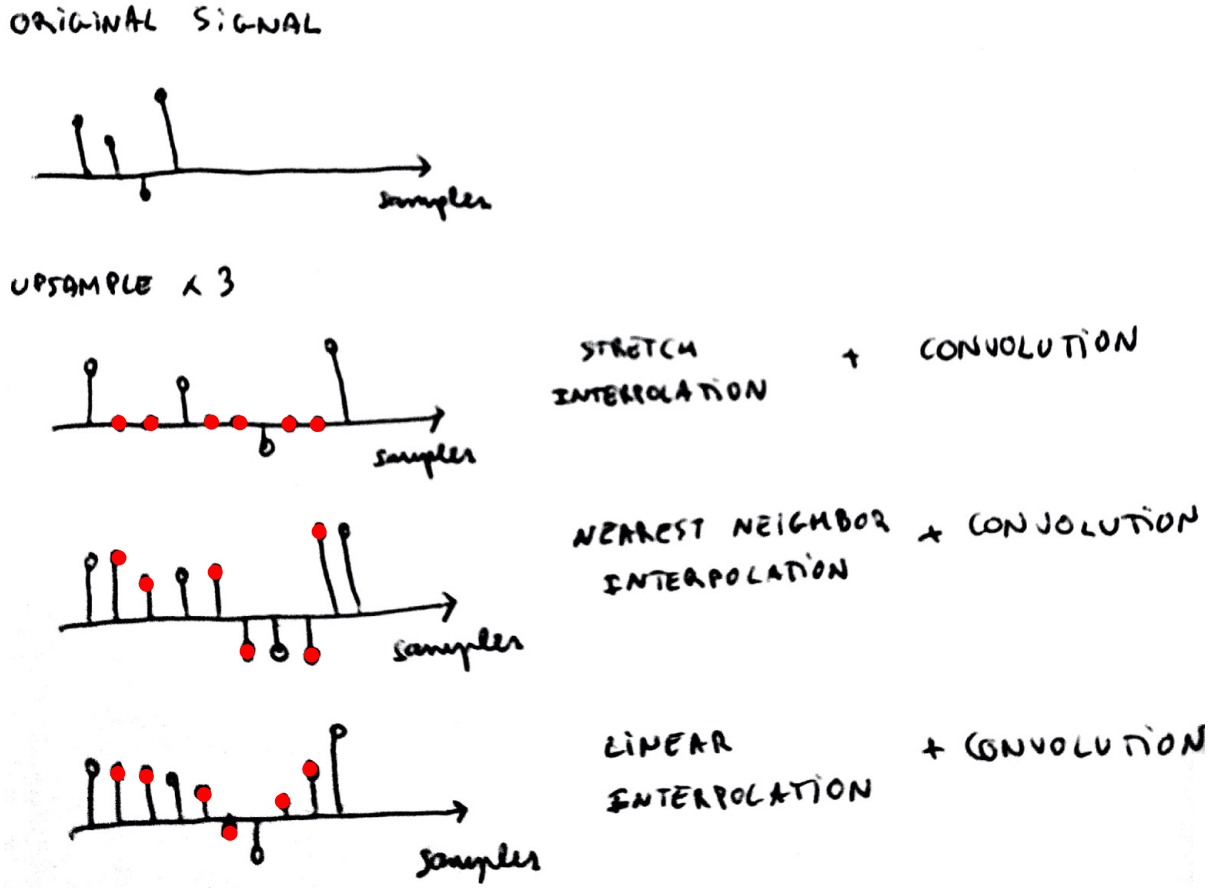
# Main upsamplers:

**Transposed convolutions**

- Widely used

**Interpolation + convolution**

- Often-times used



# Main upsamplers:

**Transposed convolutions**

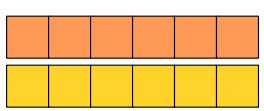
- Widely used

**Interpolation + convolution**

- Often-times used

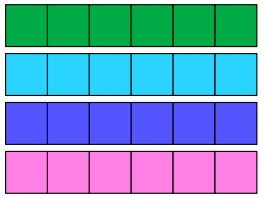
**Subpixel convolutions**

- Rarely used



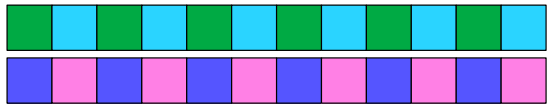
in_channels = 2
length = 6

**Convolution** (“channel upsampling”)



out_channels · up factor = 2 · 2 = 4
length = 6

**Reshape** (“periodic shuffle”)



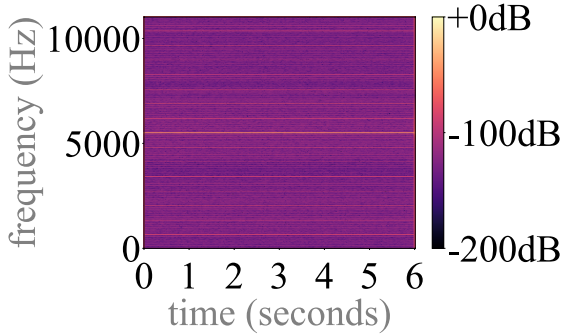
out_channels = 2
length · factor = 6 · 2 = 12

# Upsampling artifacts:

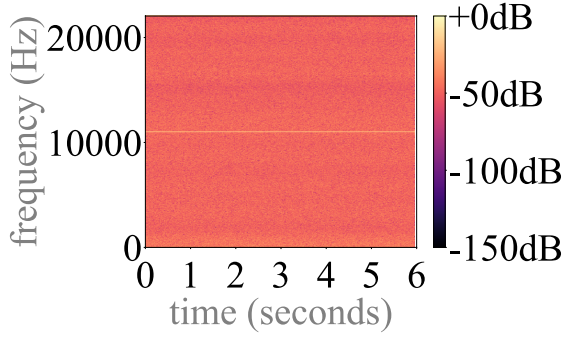
**Transposed convolutions**

Tonal artifacts

MelGAN (initialization)



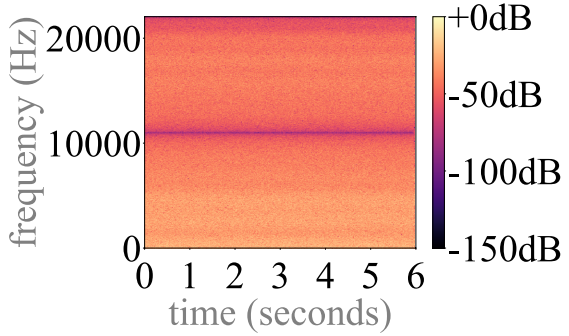

Demucs (initialization)



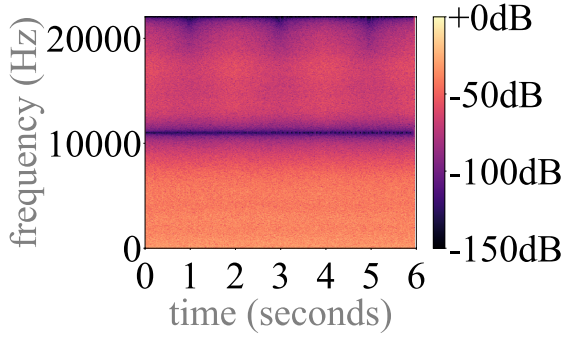

**Interpolation + convolution**

Filtering artifacts

Nearest Neighbor (initialization)

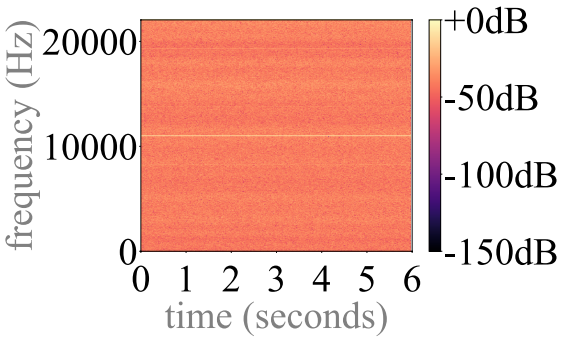



Linear (initialization)




**Subpixel convolutions**

Tonal artifacts




Subpixel Convolution (initialization)

# Agenda:

**Transposed convolutions**

- Why do they introduce tonal artifacts?

**Interpolation + convolution**

- Why do they introduce filtering artifacts?

**Subpixel convolutions**

- Why do they introduce tonal artifacts?

**Artifacts due to spectral replicas**

- Signal processing perspective

**The role of training**

- Learning from data reduces artifacts

# Transposed convolutions

# Transposed convolutions: tonal artifacts

**Main sources of tonal artifacts:**

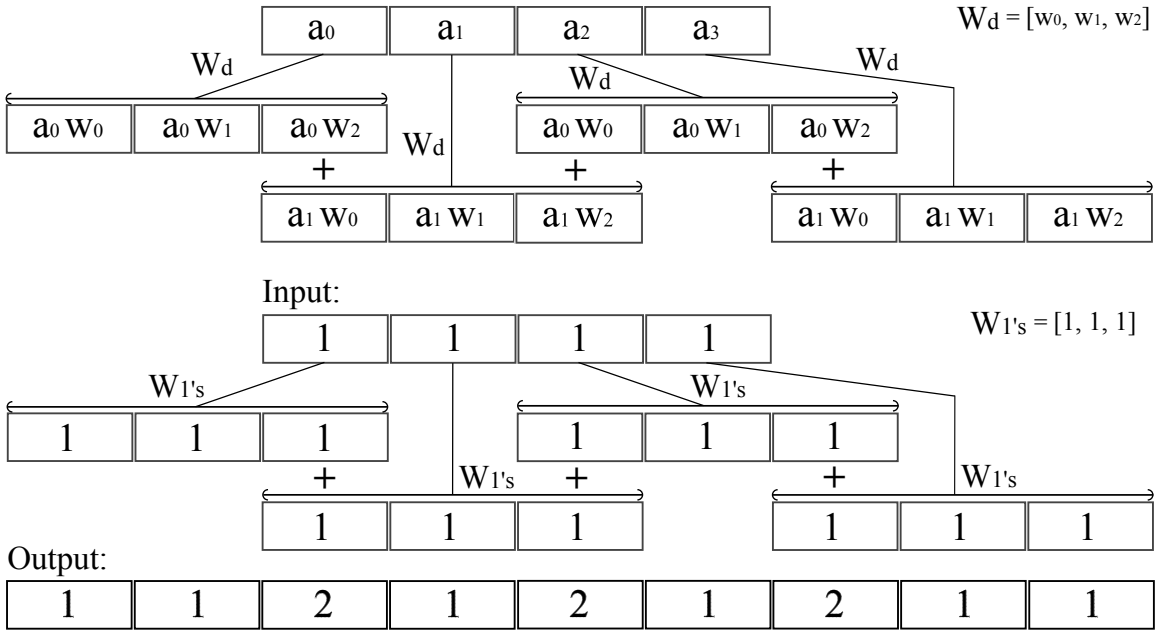
- Weights initialization
- Overlap issues

**Examples for discussion:**

- No-overlap: stride = length.
- Partial-overlap: length is *not* a multiple of stride.
  - *Example: filter length = 5, and stride = 4.*
- Full-overlap: length is a multiple of stride.
  - *Example: filter length = 8, and stride = 4.*

Odena et al., 2016: “Deconvolution and Checkerboard Artifacts” in Distill.

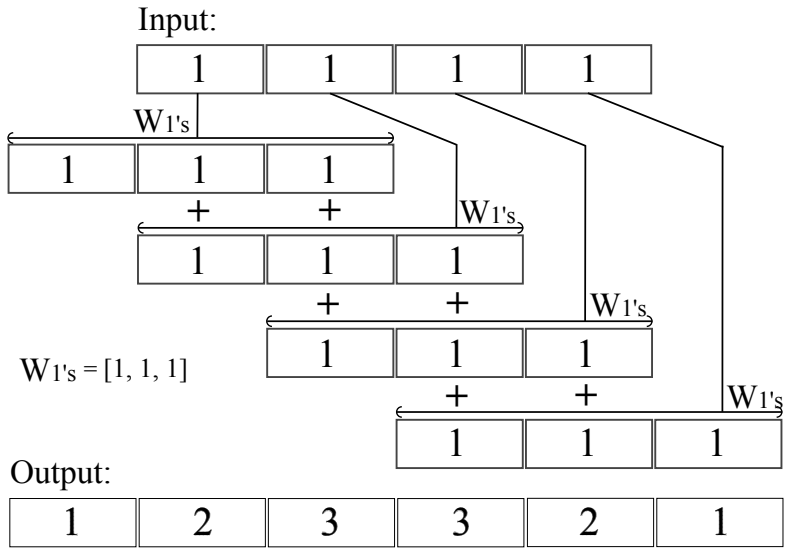
# Transposed convolutions: partial-overlap case




Example: length=3, stride=2

Note the periodicities due to **overlap issues**

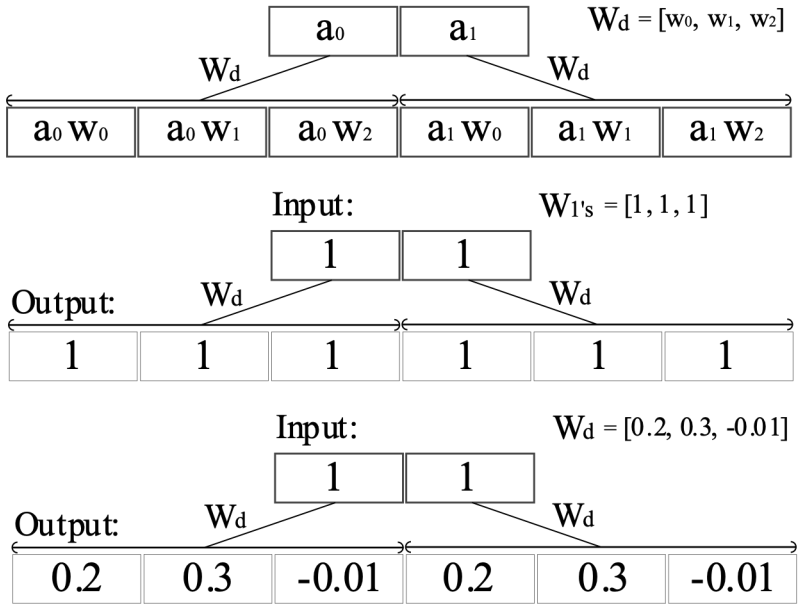
# Transposed convolutions: full-overlap case




Example: length=3, stride=1

**NO periodicities** due to constant overlap

# Transposed convolutions: no-overlap case




Example: length=3, stride=3.

**NO periodicities** due to overlap
Note the **weights initialization** issue

# Transposed convolutions

**Main sources of tonal artifacts:**

- Weights initialization
- Overlap issues

**Transposed convolution categories:**

- No-overlap: stride = length.
- Partial-overlap: length is *not* a multiple of stride.
- Full-overlap: length is a multiple of stride.

**Important remark:**

Even though you solve the overlap issue, the weights initialization issue remains due to random initialization!

# Agenda:

~~**Transposed convolutions**~~

~~- Why do they introduce tonal artifacts?~~

**Interpolation + convolution**

- Why do they introduce filtering artifacts?

**Subpixel convolutions**

- Why do they introduce tonal artifacts?

**Artifacts due to spectral replicas**

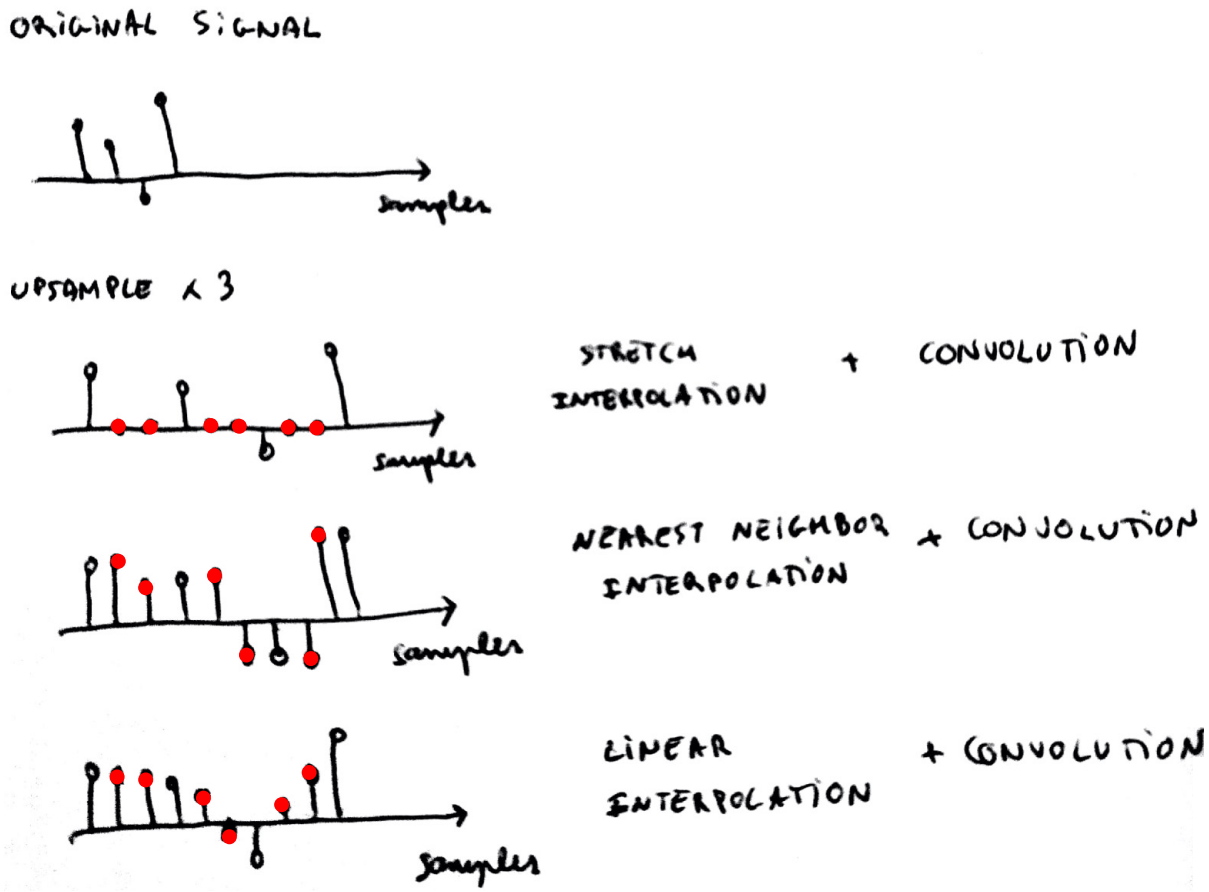
- Signal processing perspective

**The role of training**

- Learning from data reduces artifacts

# Interpolation + convolution

# Interpolation + convolution: filtering artifacts



# Interpolation: stretch + (non-learnable) convolution

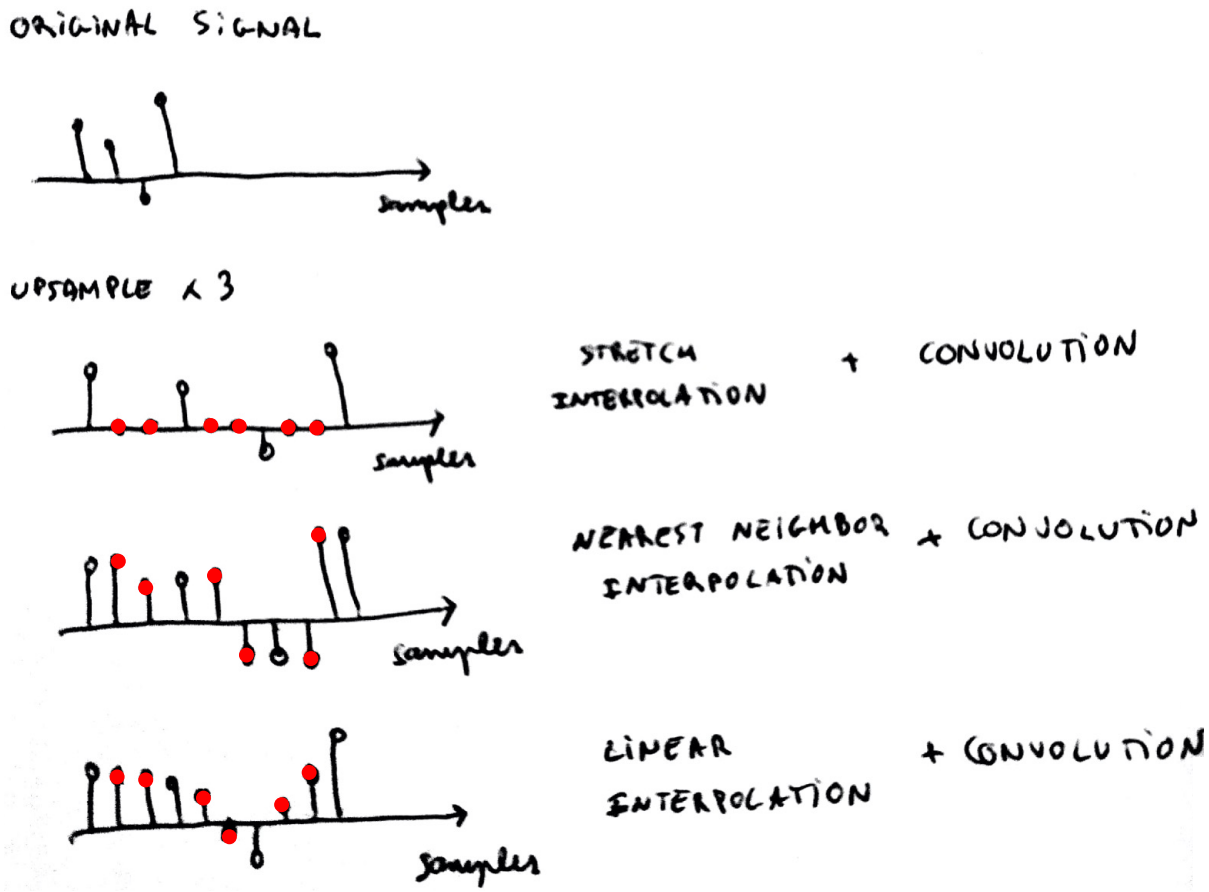
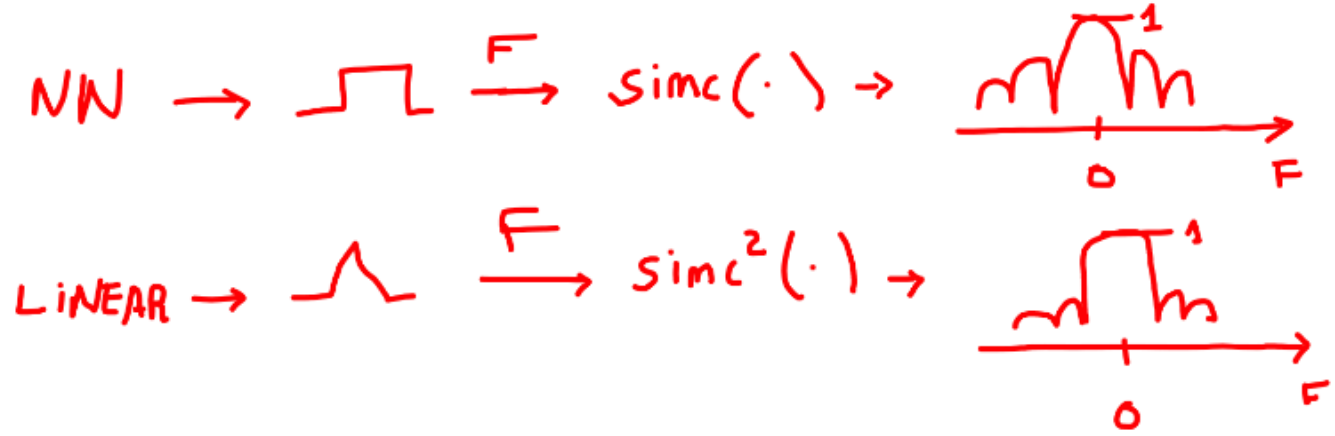
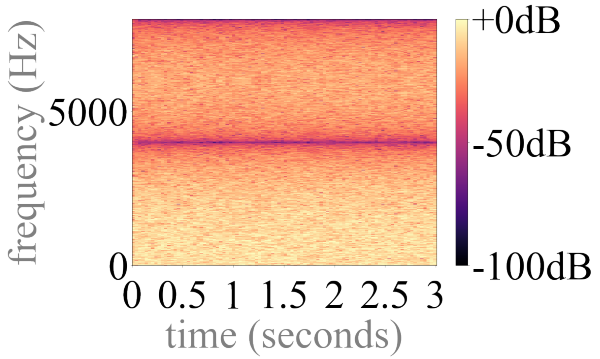


**TABLE 3.1**
Short Table of Fourier Transforms

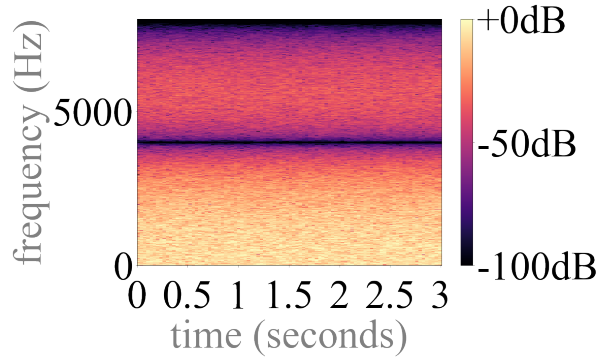
| | $g(t)$ | $G(f)$ | |
|---|---|---|---|
| | $e^{-at}u(t)$ | $\frac{1}{a+j2\pi f}$ | $a>0$ |
| 2 | $e^{at}u(-t)$ | $\frac{1}{a-j2\pi f}$ | $a>0$ |
| 3 | $e^{-a\|t\|}$ | $\frac{2a}{a^2+(2\pi f)^2}$ | |
| 4 | $te^{-at}u(t)$ | $\frac{1}{(a+j2\pi f)^2}$ | $a>0$ |
| 5 | $t^n e^{-at}u(t)$ | $\frac{n!}{(a+j2\pi f)^{n+1}}$ | |
| 6 | $\delta(t)$ | | |
| 7 | 1 | $\delta(f)$ | |
| 8 | $e^{j2\pi f_0 t}$ | $\delta(f-f_0)$ | |
| 9 | $\cos 2\pi f_0 t$ | $0.5[\delta(f+f_0)+\delta(f-f_0)]$ | |
| 10 | $\sin 2\pi f_0 t$ | $j0.5[\delta(f+f_0)-\delta(f-f_0)]$ | |
| 11 | $u(t)$ | $\frac{1}{2}\delta(f)+\frac{1}{j2\pi f}$ | |
| 12 | $\operatorname{sgn} t$ | $\frac{2}{j2\pi f}$ | |
| | $\cos 2\pi f_0 t\, u(t)$ | $\frac{1}{4}[\delta(f-f_0)+\delta(f+f_0)]+\frac{j2\pi f}{(2\pi f_0)^2-(2\pi f)^2}$ | |
| 14 | $\sin 2\pi f_0 t\, u(t)$ | $\frac{1}{4j}[\delta(f-f_0)-\delta(f+f_0)]+\frac{2\pi f_0}{(2\pi f_0)^2-(2\pi f)^2}$ | |
| 15 | $e^{-at}\sin 2\pi f_0 t\, u(t)$ | $\frac{2\pi f_0}{(a+j2\pi f)^2+4\pi^2 f_0^2}$ | |
| 16 | $e^{-at}\cos 2\pi f_0 t\, u(t)$ | $\frac{a+j2\pi f}{(a+j2\pi f)^2+4\pi^2 f_0^2}$ | |
| | $\Pi\left(\frac{t}{\tau}\right)$ | $\tau \operatorname{sinc}(\pi f \tau)$ | |
| 18 | $2B\operatorname{sinc}(2\pi B t)$ | $\Pi\left(\frac{f}{2B}\right)$ | |
| 19 | $\Delta\left(\frac{t}{\tau}\right)$ | $\frac{\tau}{2}\operatorname{sinc}^2\left(\frac{\pi f \tau}{2}\right)$ | |
| 20 | $B\operatorname{sinc}^2(\pi B t)$ | $\Delta\left(\frac{f}{2B}\right)$ | |
| 21 | $\sum_{n=-\infty}^{\infty}\delta(t-nT)$ | $f_0\sum_{n=-\infty}^{\infty}\delta(f-nf_0)$ | |
| 22 | $e^{-t^2/2\sigma^2}$ | $\sigma\sqrt{2\pi}e^{-2(\sigma\pi f)^2}$ | |




Upsample white noise at 4kHz by 4

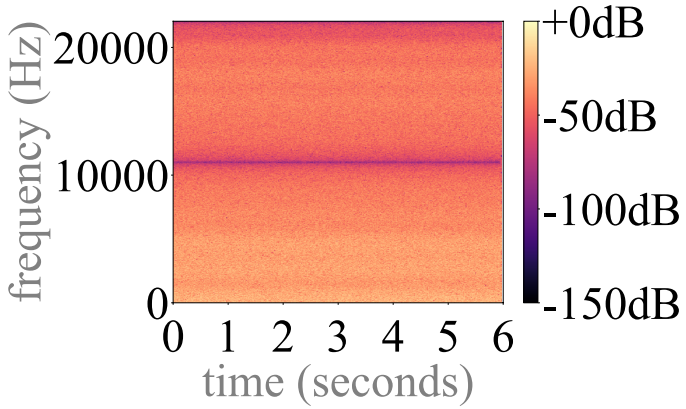



Nearest Neighbor

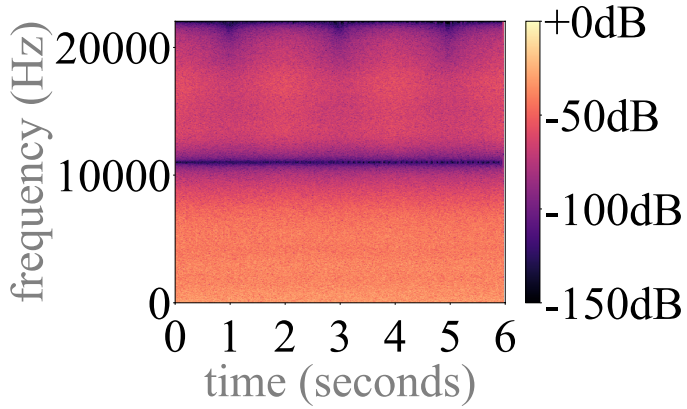



Linear interpolation

# Interpolation + convolution: filtering artifacts




Demucs-like
Nearest Neighbor
(initialization)




Demucs-like
Linear interpolation
(initialization)

**Important remark:**

Filtering artifacts emerge because the frequency response of each interpolation colors the signal.

# Agenda:

~~**Transposed convolutions**~~

~~- Why do they introduce tonal artifacts?~~

~~**Interpolation + convolution**~~

~~- Why do they introduce filtering artifacts?~~

**Subpixel convolutions**

- Why do they introduce tonal artifacts?

**Artifacts due to spectral replicas**

- Signal processing perspective

**The role of training**

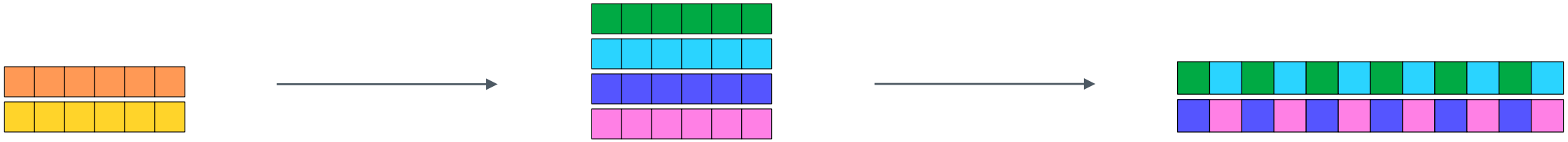
- Learning from data reduces artifacts

# Subpixel convolution

# Subpixel convolution: tonal artifacts

**Convolution**
("channel upsampling")

**Reshape**
("periodic shuffle")



in_channels = 2
length = 6

out_channels · up factor = 2 · 2 = 4
length = 6

out_channels = 2
length · factor = 6 · 2 = 12

# Agenda:

~~**Transposed convolutions**~~

~~- Why do they introduce tonal artifacts?~~

~~**Interpolation + convolution**~~

~~- Why do they introduce filtering artifacts?~~

~~**Subpixel convolutions**~~

~~- Why do they introduce tonal artifacts?~~

**Artifacts due to spectral replicas**

- Signal processing perspective

**The role of training**

- Learning from data reduces artifacts

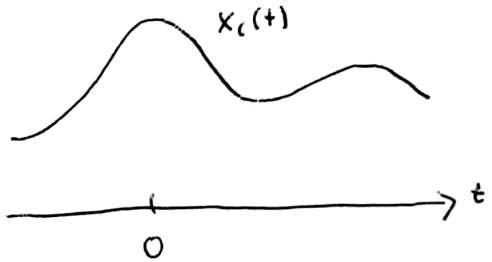
# Artifacts due to spectral replicas

# Signal processing review

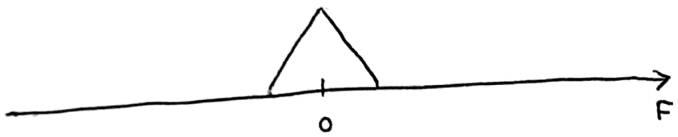
**IDEA 1:**
Spectral replicas emerge when sampling/discretizing a signal!

**IDEA 2:**
When upsampling, one performs bandwith extension – be aware of spectral replicas!

**Time domain**

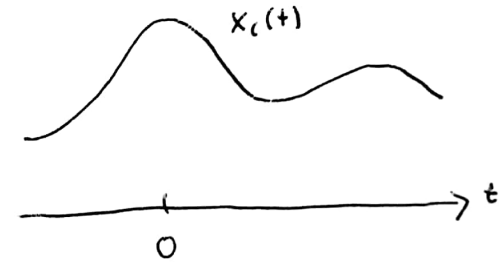


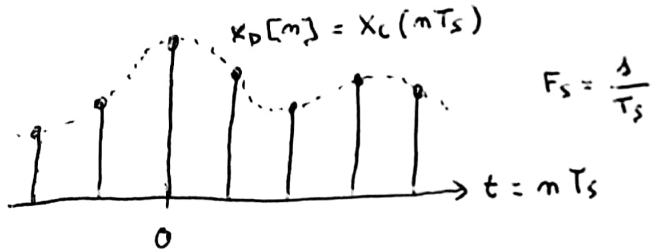

**Frequency domain**



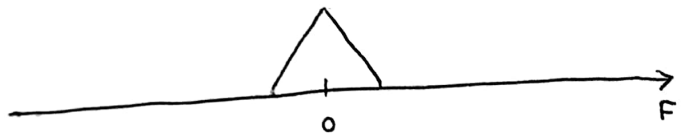

SOURCE OF INSPIRATION: CHEGG.COM/HOMEWORK-HELP/QUESTIONS-AND-ANSWERS/SUBJECT-DISCRETE-TIME-SIGNALS-SYSTEMS-TITLE-APPLIED-DIGITAL-SIGNAL-PROCESSING-AUTHOR-S-MAN-Q11372197

**Time domain**



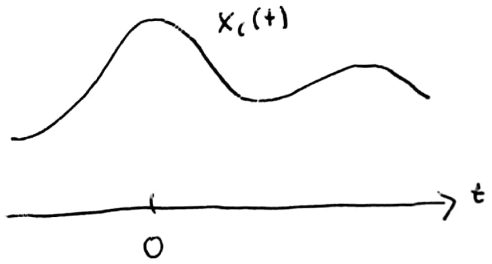



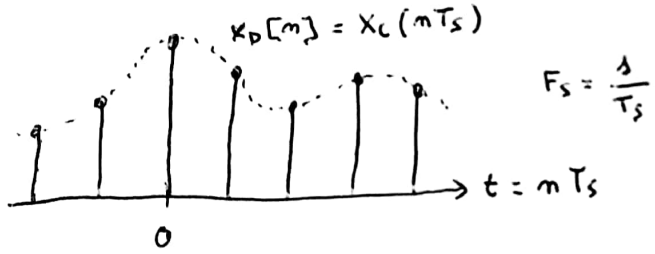

**Frequency domain**



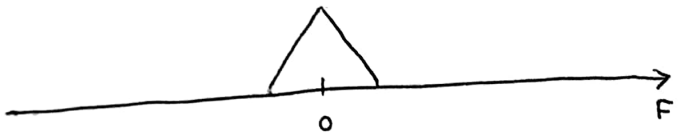

SOURCE OF INSPIRATION: CHEGG.COM/HOMEWORK-HELP/QUESTIONS-AND-ANSWERS/SUBJECT-DISCRETE-TIME-SIGNALS-SYSTEMS-TITLE-APPLIED-DIGITAL-SIGNAL-PROCESSING-AUTHOR-S-MAN-Q11372197

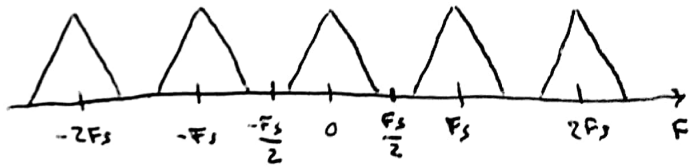
**Time domain**




**Frequency domain**



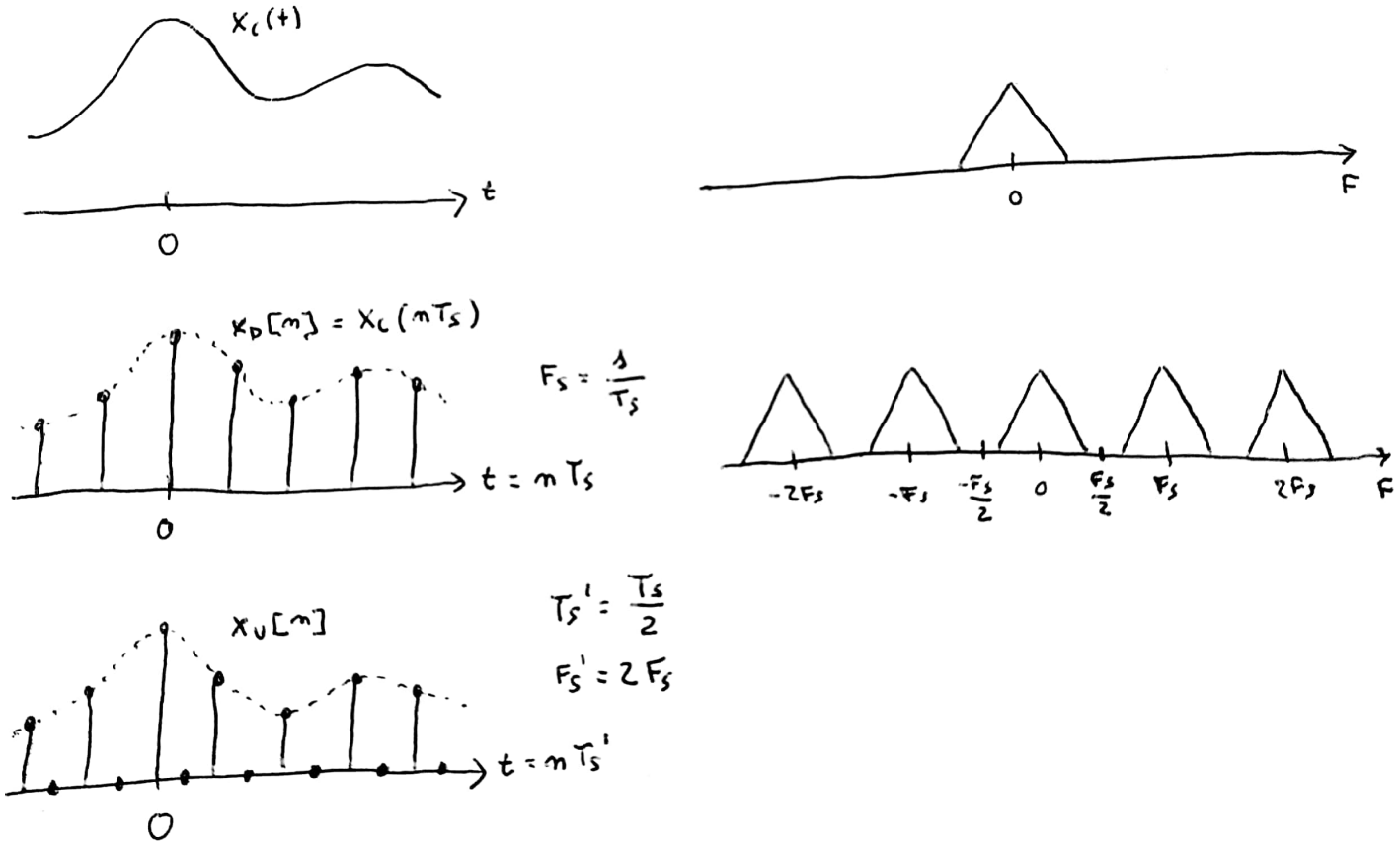




**IDEA 1:**
Spectral replicas emerge when sampling/discretizing a signal!

SOURCE OF INSPIRATION: CHEGG.COM/HOMEWORK-HELP/QUESTIONS-AND-ANSWERS/SUBJECT-DISCRETE-TIME-SIGNALS-SYSTEMS-TITLE-APPLIED-DIGITAL-SIGNAL-PROCESSING-AUTHOR-S-MAN-Q11372197

**Time domain**

**Frequency domain**




**IDEA 1:**
Spectral replicas emerge when sampling/discretizing a signal!

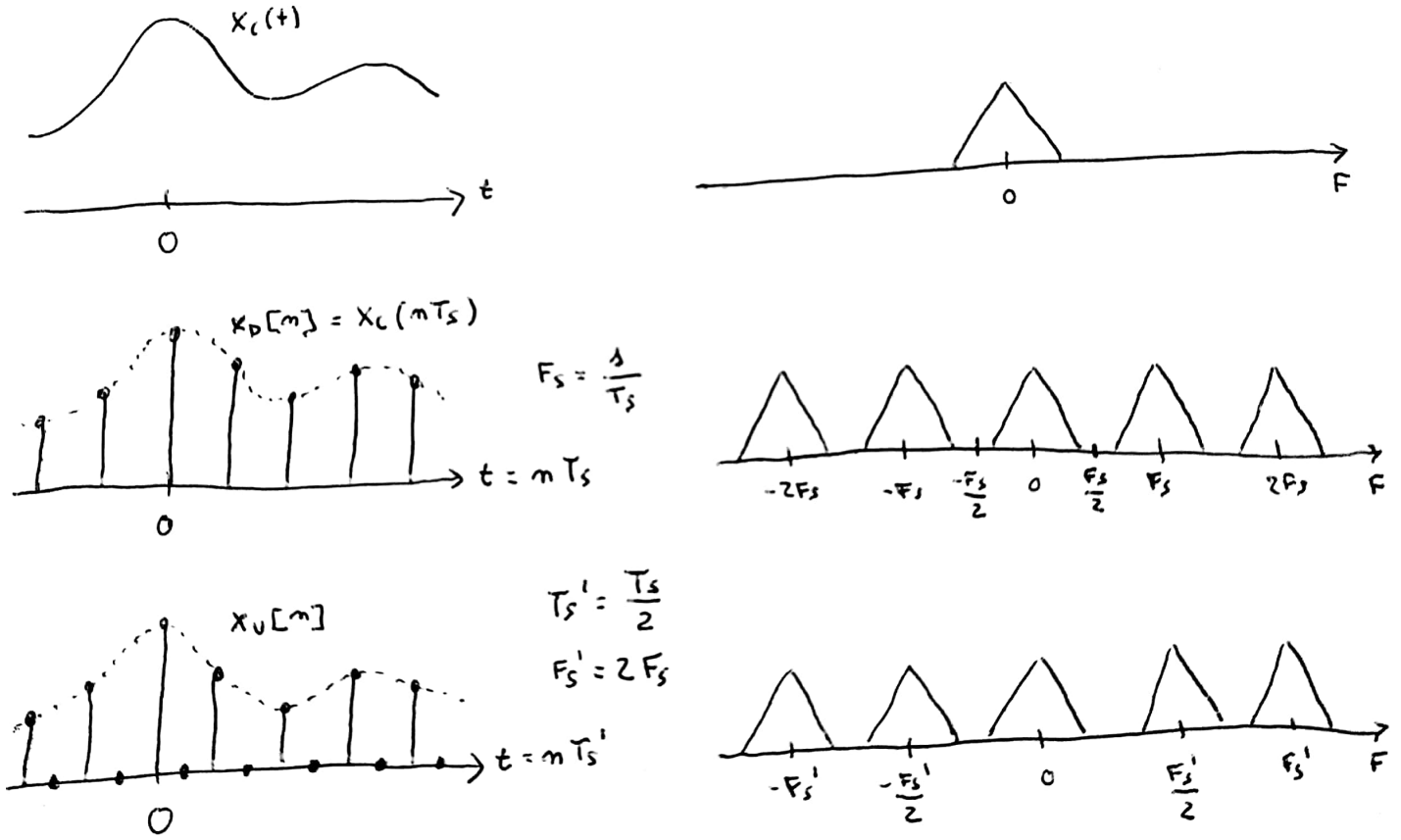
**Stretch interpolation x2**
(upsampling with zeros)

SOURCE OF INSPIRATION: CHEGG.COM/HOMEWORK-HELP/QUESTIONS-AND-ANSWERS/SUBJECT-DISCRETE-TIME-SIGNALS-SYSTEMS-TITLE-APPLIED-DIGITAL-SIGNAL-PROCESSING-AUTHOR-S-MAN-Q11372197

**Time domain**

**Frequency domain**

**IDEA 1:**
Spectral replicas emerge when sampling/discretizing a signal!

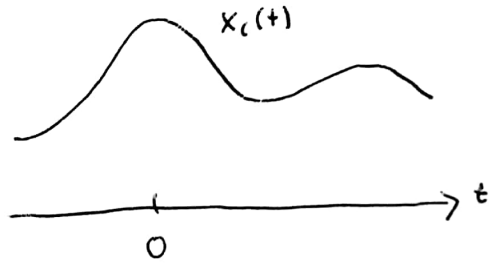


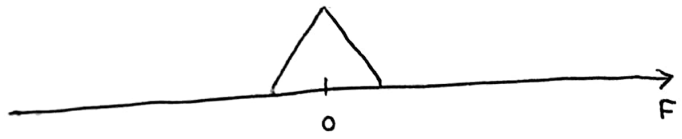

**Stretch interpolation x2**
(upsampling with zeros)

SOURCE OF INSPIRATION: CHEGG.COM/HOMEWORK-HELP/QUESTIONS-AND-ANSWERS/SUBJECT-DISCRETE-TIME-SIGNALS-SYSTEMS-TITLE-APPLIED-DIGITAL-SIGNAL-PROCESSING-AUTHOR-S-MAN-Q11372197

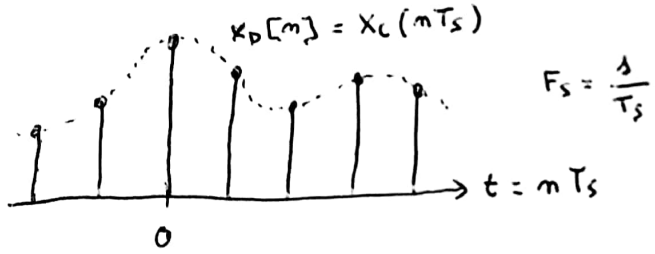
**Time domain**

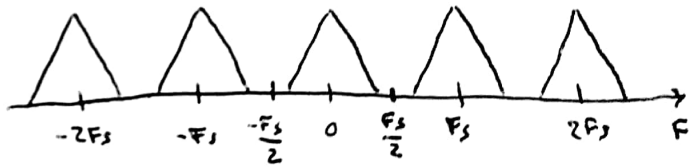
**Frequency domain**



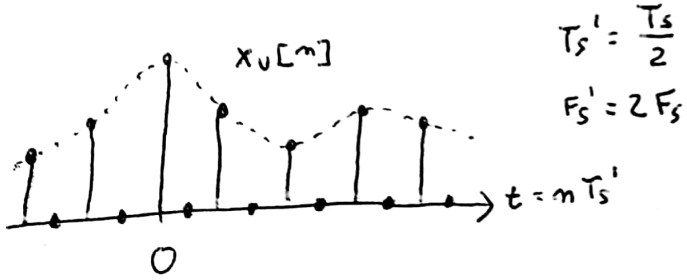



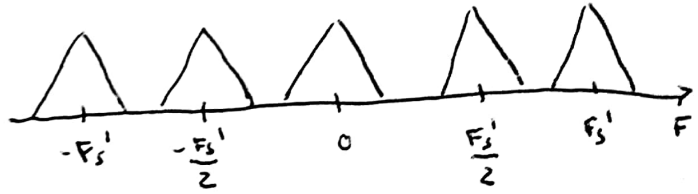

**IDEA 1:**
Spectral replicas emerge when sampling/discretizing a signal!




**IDEA 2:**
When upsampling, one performs bandwith extension – be aware of spectral replicas!



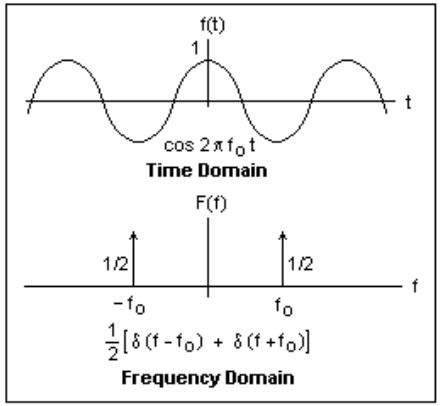

**Stretch interpolation x2**
(upsampling with zeros)

SOURCE OF INSPIRATION: CHEGG.COM/HOMEWORK-HELP/QUESTIONS-AND-ANSWERS/SUBJECT-DISCRETE-TIME-SIGNALS-SYSTEMS-TITLE-APPLIED-DIGITAL-SIGNAL-PROCESSING-AUTHOR-S-MAN-Q11372197

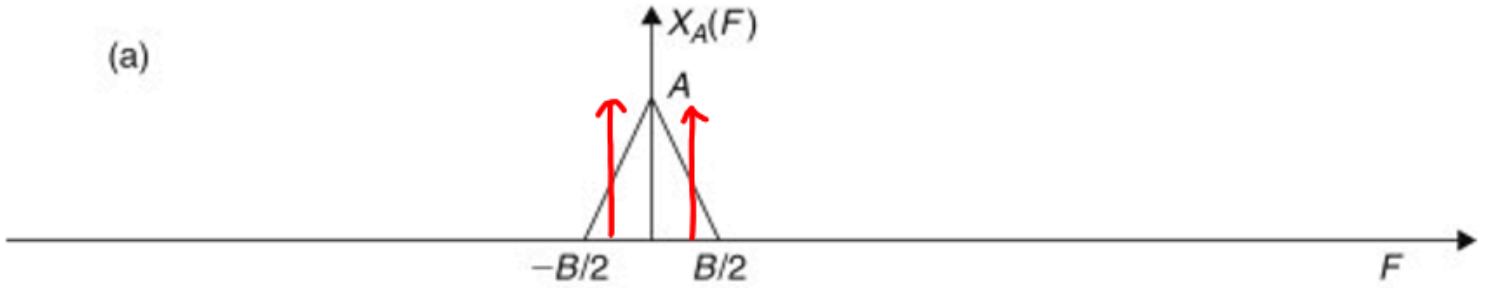
# Signal processing review

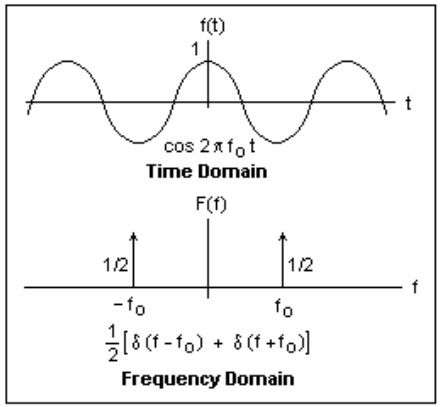
**IDEA 1:**
Spectral replicas emerge when sampling/discretizing a signal!

**IDEA 2:**
When upsampling, one performs bandwith extension – be aware of spectral replicas!

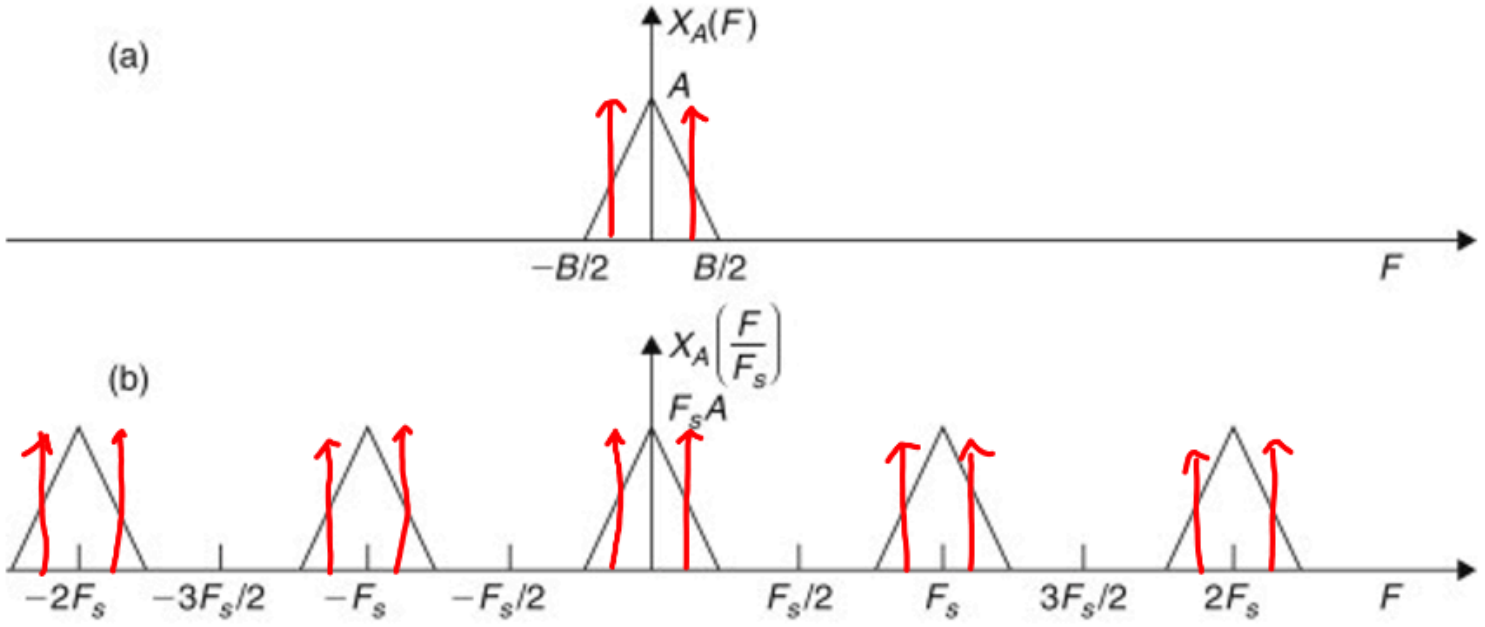


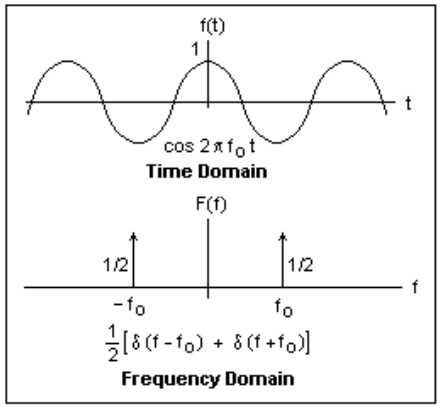
# Spectral replicas of tonal artifacts: multilayered case



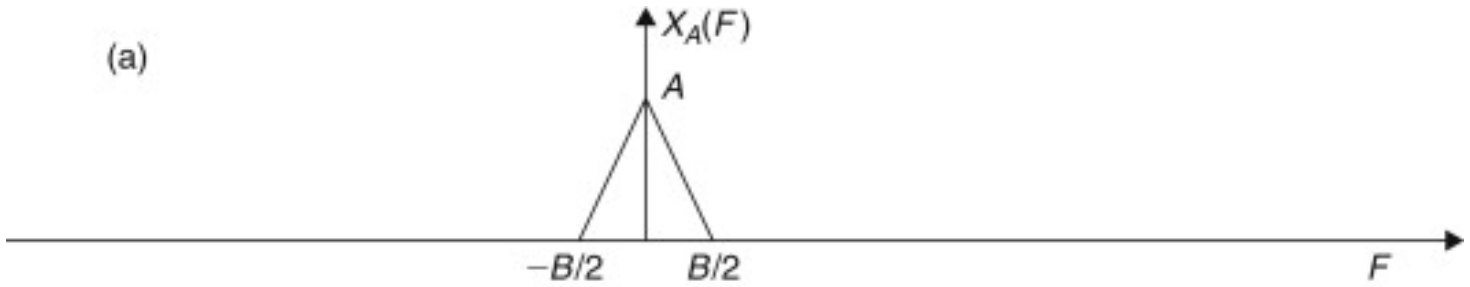



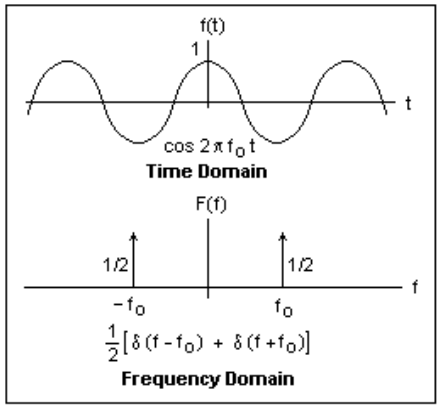
# Spectral replicas of tonal artifacts: multilayered case



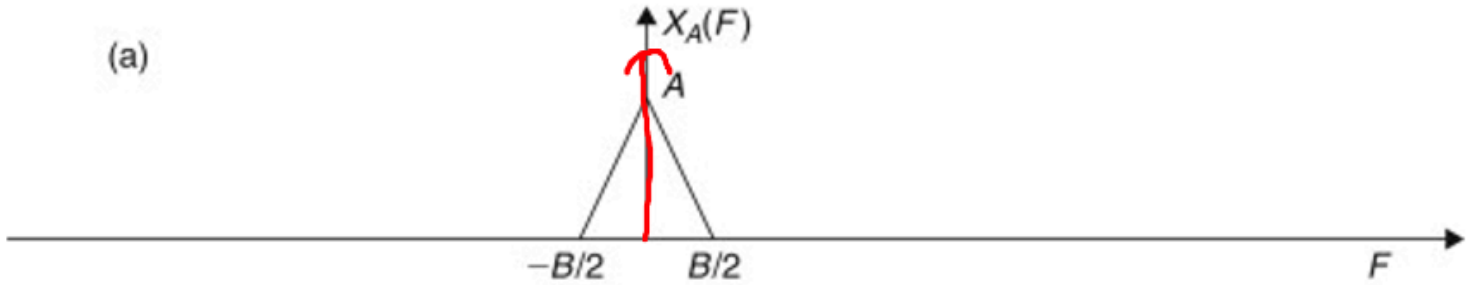



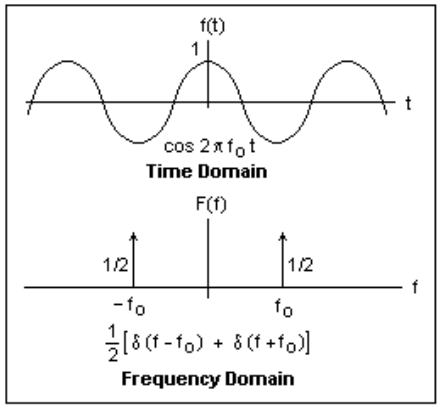
# Spectral replicas of signal offsets



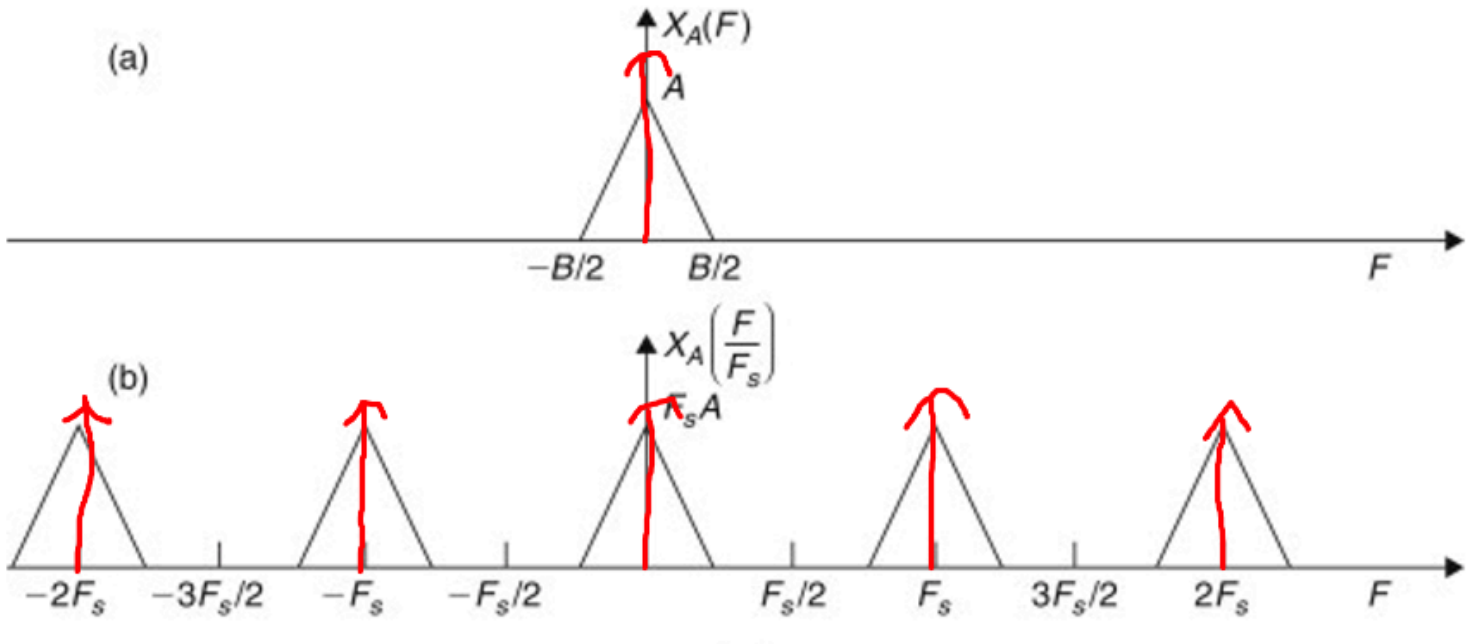



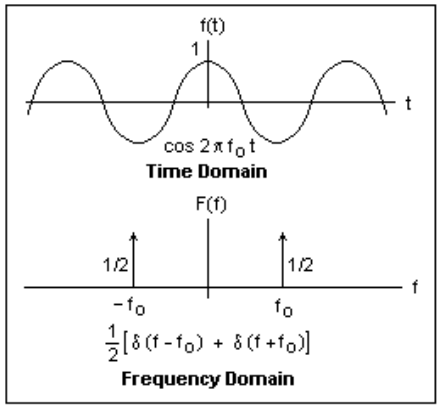
# Spectral replicas of signal offsets



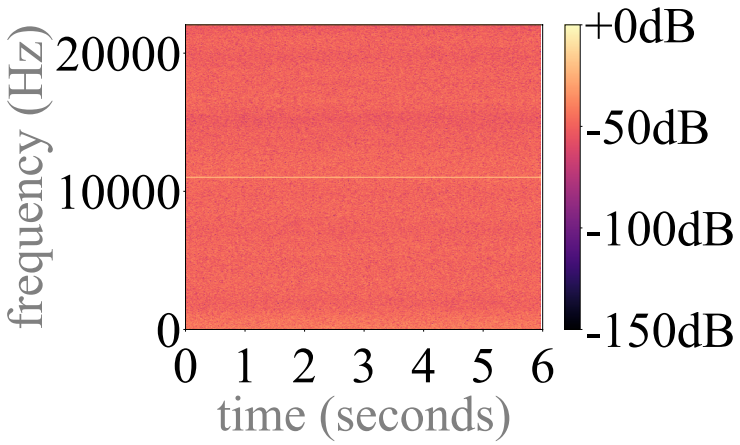



# Spectral replicas of signal offsets



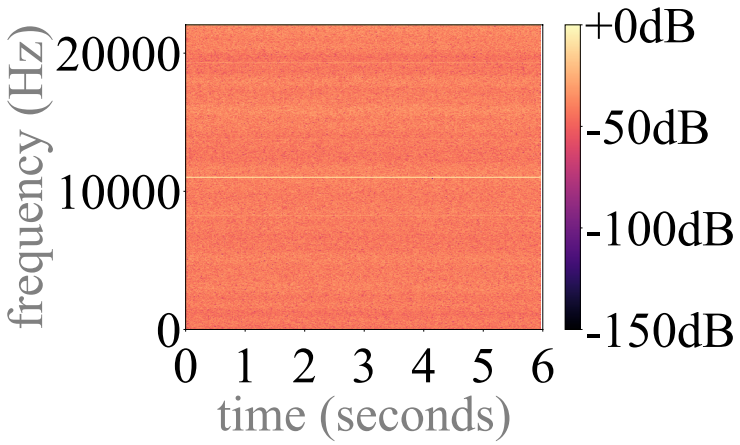



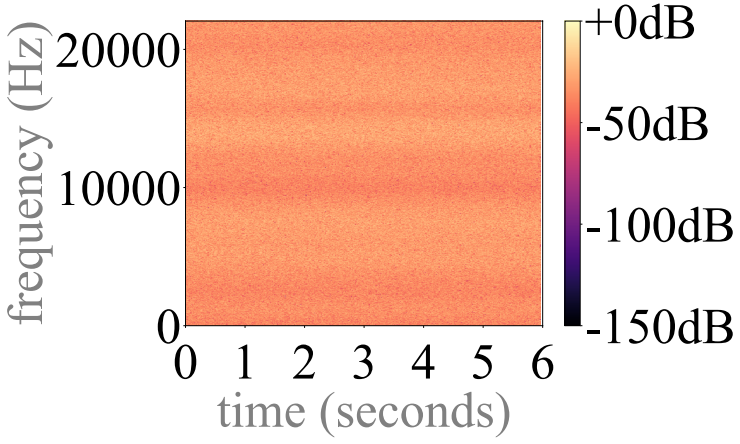
Demucs
(initialization)



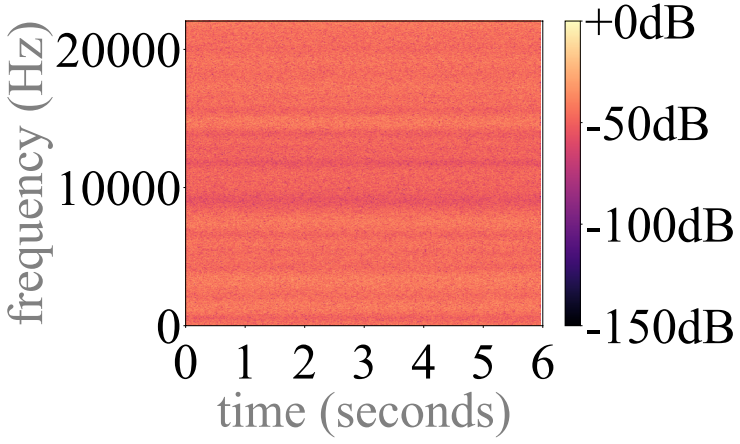

**with**
biases & ReLUs
(introduce signal offsets)

Subpixel Convolution
(initialization)




**without**
biases & ReLUs



# Artifacts due to spectral replicas

**Additional sources of upsampling artifacts:**

- Spectral replicas of tonal artifacts
- Spectral replicas of filtering artifacts
- Spectral replicas of signal offsets

# Agenda:

~~**Transposed convolutions**~~

~~- Why do they introduce tonal artifacts?~~

~~**Interpolation + convolution**~~

~~- Why do they introduce filtering artifacts?~~

~~**Subpixel convolutions**~~

~~- Why do they introduce tonal artifacts?~~

~~**Artifacts due to spectral replicas**~~

~~- Signal processing perspective~~

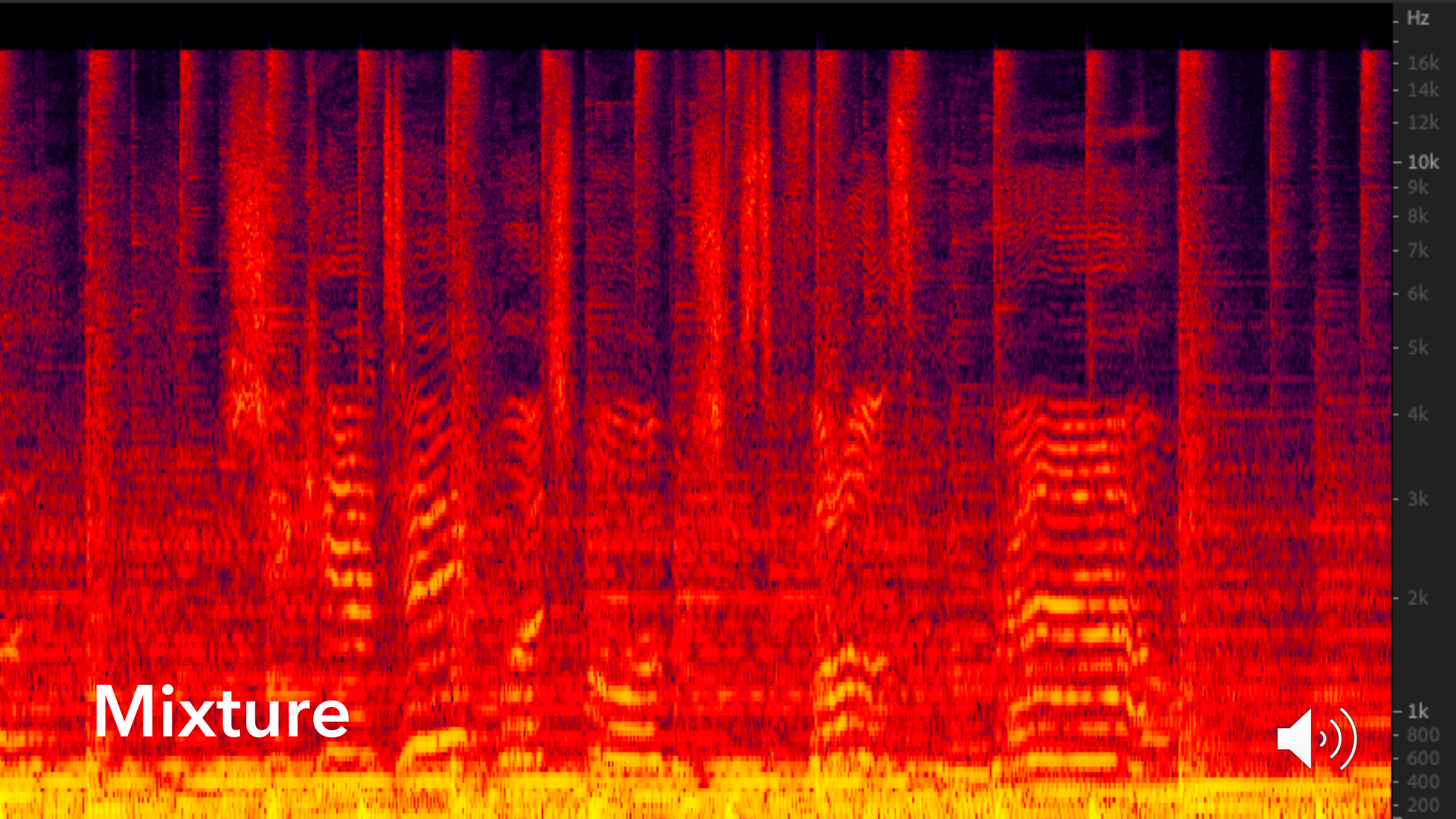
**The role of training**

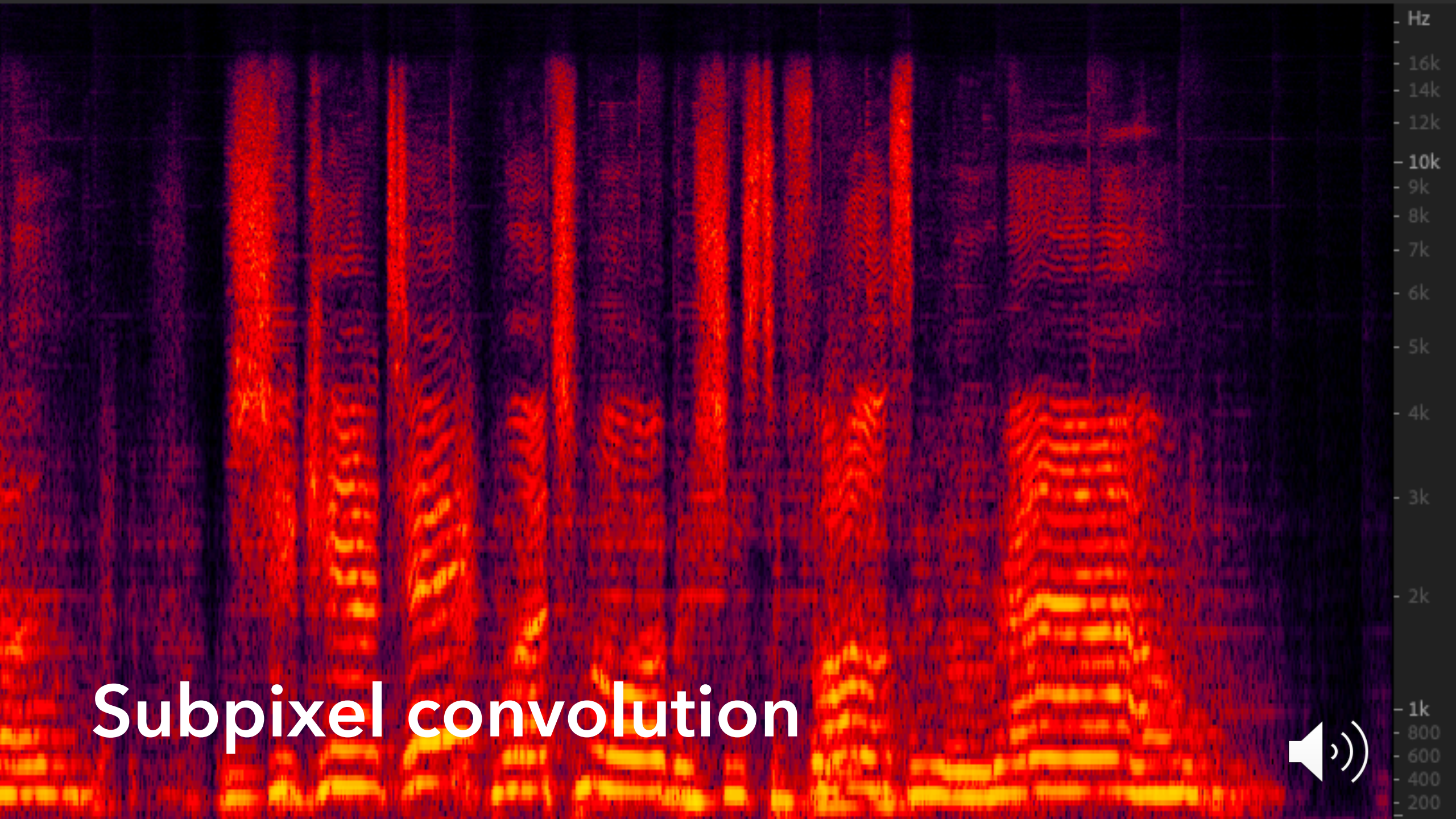
- Learning from data reduces artifacts

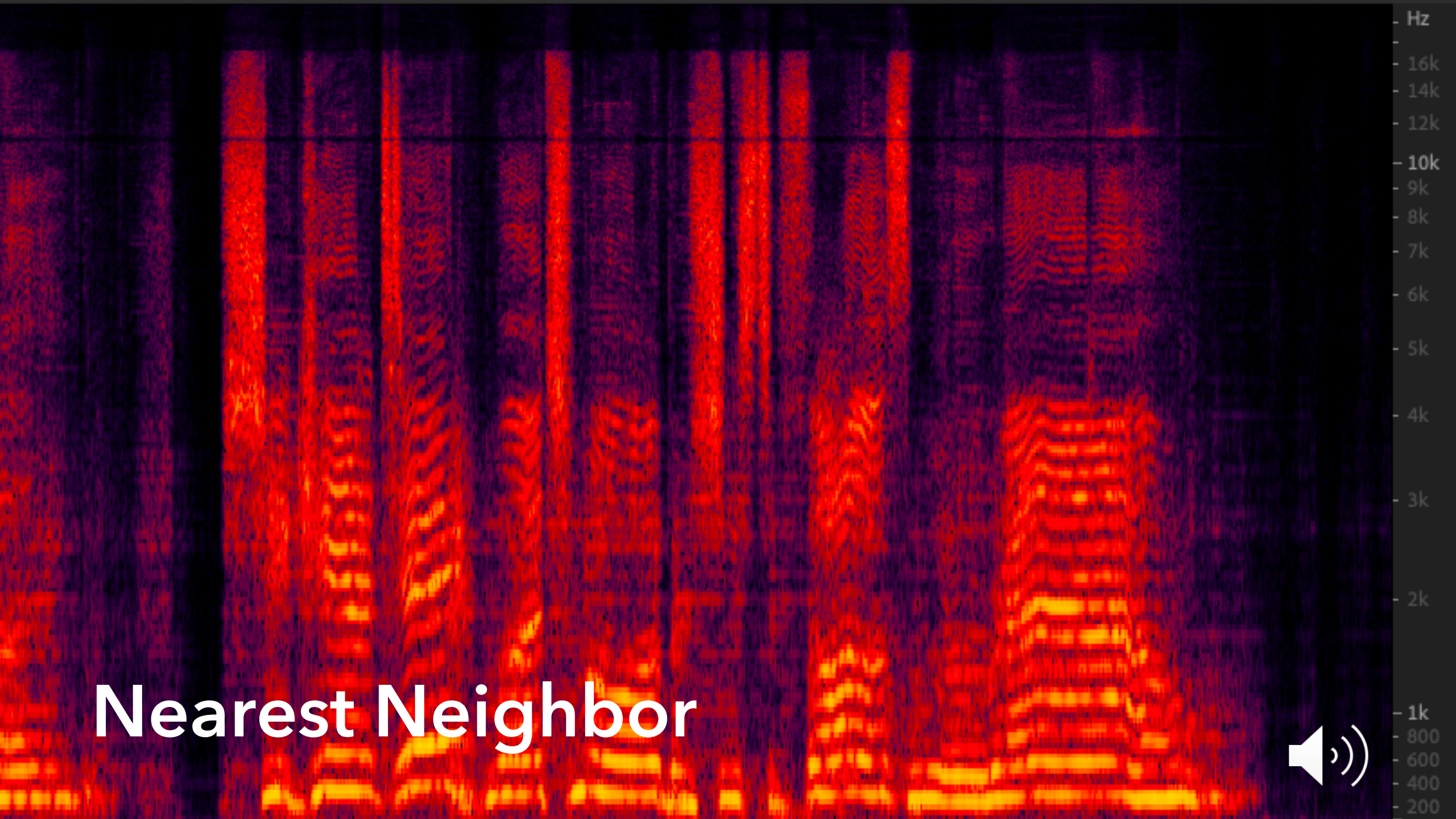
# The role of training

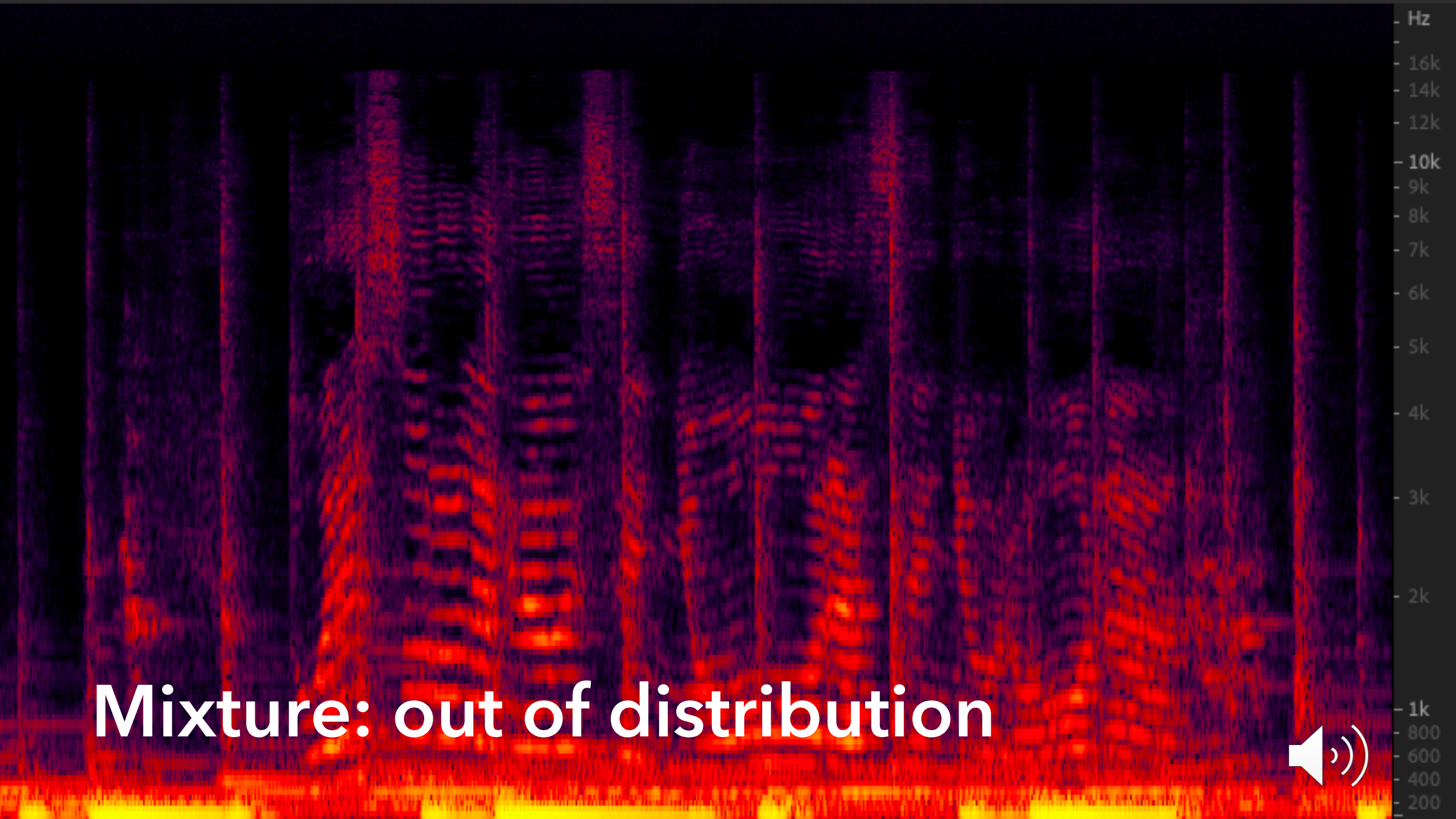
# Is training dealing with the problematic initializations?

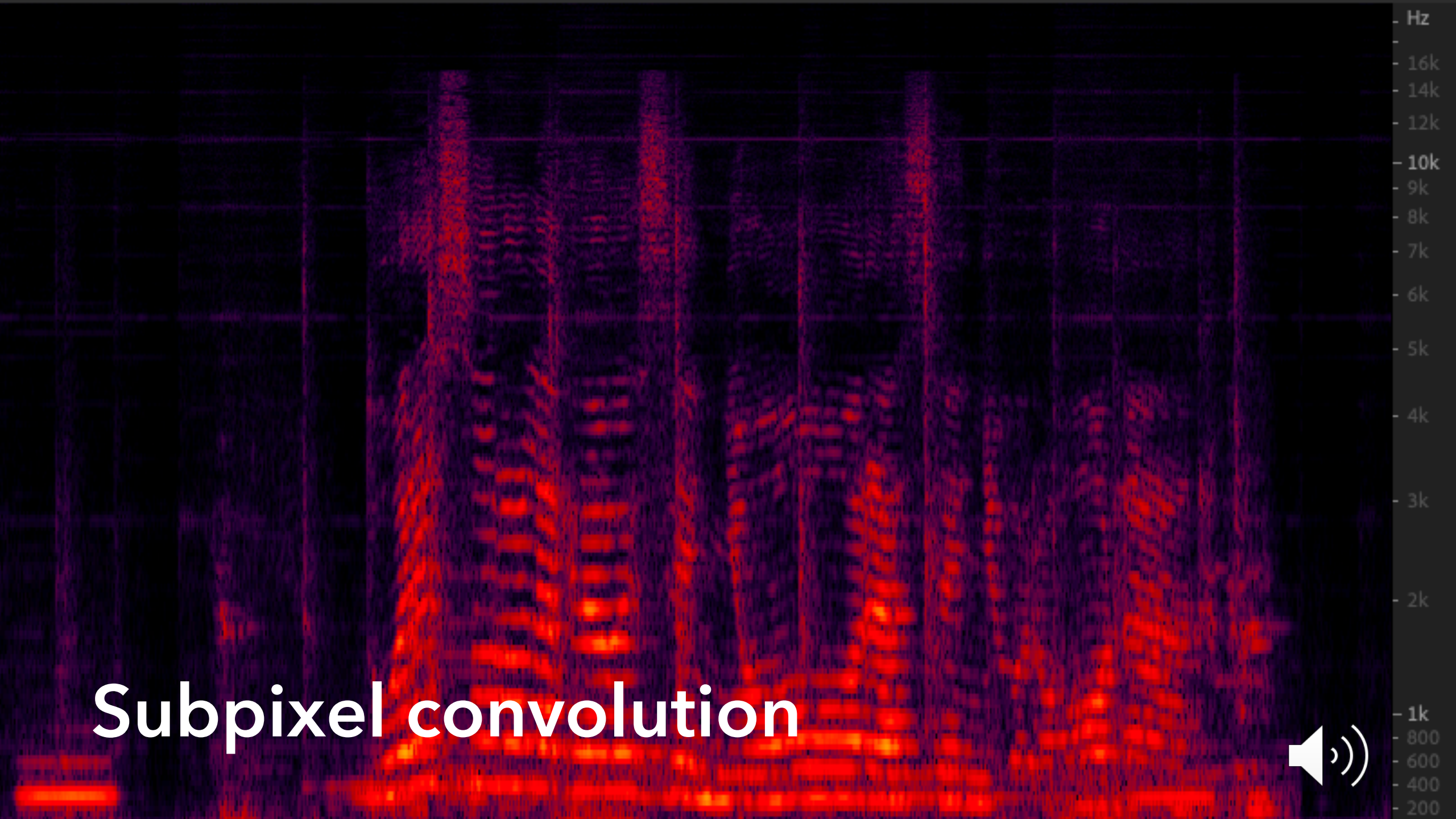
| **Music source separation (MUSDB [28] benchmark)** | **SDR ↑** | **epoch** | **#parm** |
|---|---|---|---|
| Demucs-like: transposed CNN (full-overlap) | 5.35 | 319 s | 703M |
| Demucs-like: nearest neighbor interpolation | 5.17 | 423 s | 716M |
| Demucs-like: linear interpolation | 4.62 | 430 s | 716M |
| Demucs-like: subpixel CNN | 5.38 | 311 s | 729M |

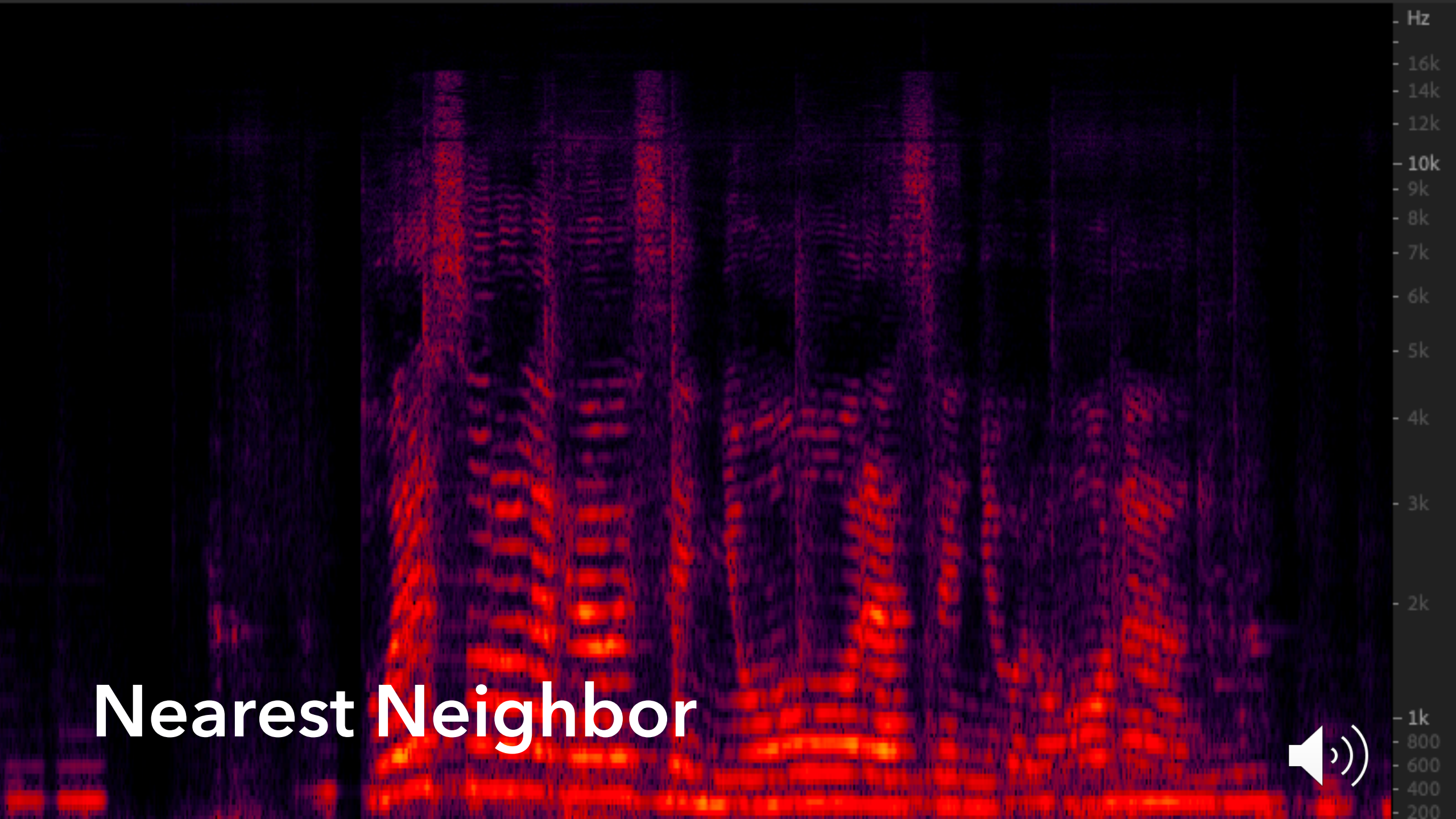

Hz
16k
14k
12k
10k
9k
8k
7k
6k
5k
4k
3k
2k
1k
800
600
400
200
Mixture


Hz
16k
14k
12k
10k
9k
8k
7k
6k
5k
4k
3k
2k
1k
800
600
400
200
Subpixel convolution


Hz
16k
14k
12k
10k
9k
8k
7k
6k
5k
4k
3k
2k
1k
800
600
400
200
Nearest Neighbor


Hz
16k
14k
12k
10k
9k
8k
7k
6k
5k
4k
3k
2k
1k
800
600
400
200
Mixture: out of distribution


Hz
16k
14k
12k
10k
9k
8k
7k
6k
5k
4k
3k
2k
1k
800
600
400
200
Subpixel convolution


Hz
16k
14k
12k
10k
9k
8k
7k
6k
5k
4k
3k
2k
1k
800
600
400
200
Nearest Neighbor

# The role of training: helps overcoming the noisy initializations to get state-of-the-art results

**Formal evaluation:**

- Transposed and subpixel CNNs achieve the best SDR scores.
  - Despite their poor initialization!
- Nearest neighbour upsampler follows closely!

**Informal listening:**

- Upsampling artifacts can emerge even after training!
  - Tonal artifacts: silent parts and with out-of-distribution data.
  - Filtering artifacts: they are not that perceptually annoying.

# Agenda:

~~**Transposed convolutions**~~

~~- Why do they introduce tonal artifacts?~~

~~**Interpolation + convolution**~~

~~- Why do they introduce filtering artifacts?~~

~~**Subpixel convolutions**~~

~~- Why do they introduce tonal artifacts?~~

~~**Artifacts due to spectral replicas**~~

~~- Signal processing perspective~~

~~**The role of training**~~

~~- Learning from data reduces artifacts~~


arxiv.org/pdf/2010.14356.pdf

@jordiponsdotme – www.jordipons.me