tinyML® Summit

Miniature dreams can come true…

MARCH 28-30, 2022 | SAN FRANCISCO BAY AREA

TINY ML

www.tinyML.org

REAL-TIME DEEP SPEECH ENHANCEMENT SYSTEM FOR EMBEDDED VOICE UI

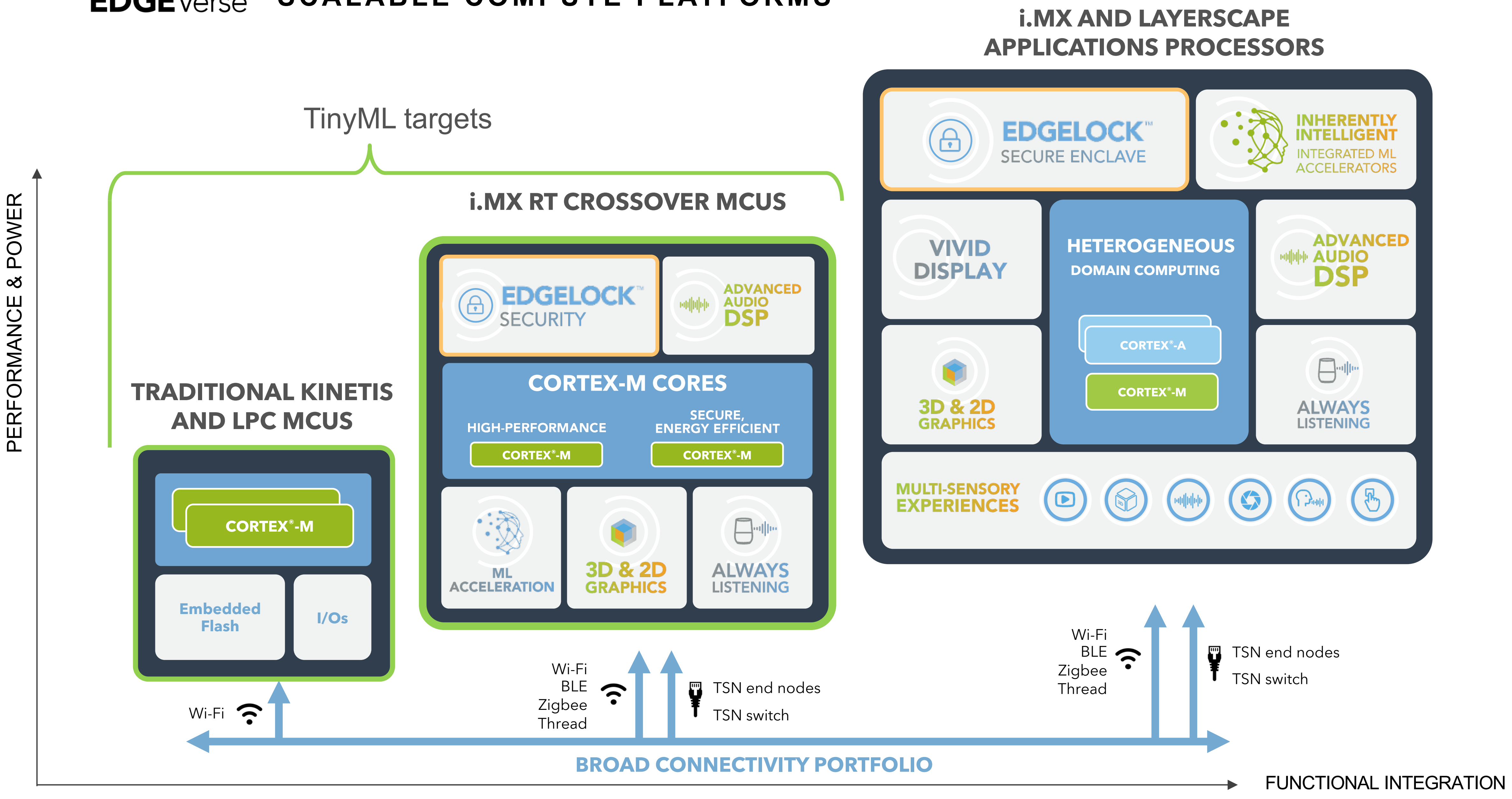
TESS BOIVIN

ML ENGINEER

Aurelie Donjon, Guido Roncarolo,
Voice and Audio team

SECURE CONNECTIONS
FOR A SMARTER WORLD

EDGEVerse SCALABLE COMPUTE PLATFORMS



TinyML targets

TRADITIONAL KINETIS AND LPC MCUS

CORTEX®-M

Embedded Flash

I/Os

i.MX RT CROSSOVER MCUS

EDGELOCK™ SECURITY

ADVANCED AUDIO DSP

CORTEX-M CORES

HIGH-PERFORMANCE CORTEX®-M

SECURE, ENERGY EFFICIENT CORTEX®-M

ML ACCELERATION

3D & 2D GRAPHICS

ALWAYS LISTENING

i.MX AND LAYERSCAPE APPLICATIONS PROCESSORS

EDGELOCK™ SECURE ENCLAVE

INHERENTLY INTELLIGENT INTEGRATED ML ACCELERATORS

VIVID DISPLAY

HETEROGENEOUS DOMAIN COMPUTING

CORTEX®-A

CORTEX®-M

ADVANCED AUDIO DSP

3D & 2D GRAPHICS

ALWAYS LISTENING

MULTI-SENSORY EXPERIENCES

Wi-Fi

Wi-Fi
BLE
Zigbee
Thread

TSN end nodes
TSN switch

Wi-Fi
BLE
Zigbee
Thread

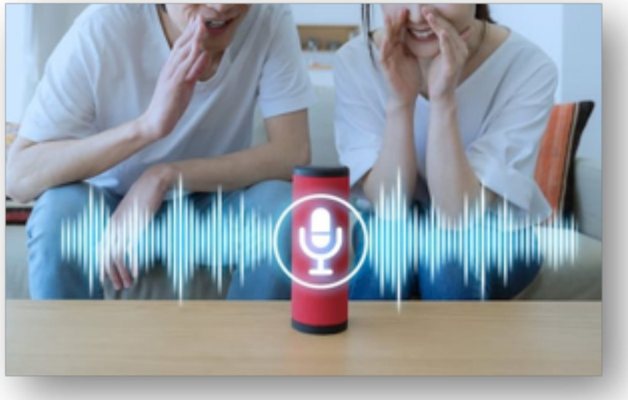
TSN end nodes
TSN switch

BROAD CONNECTIVITY PORTFOLIO

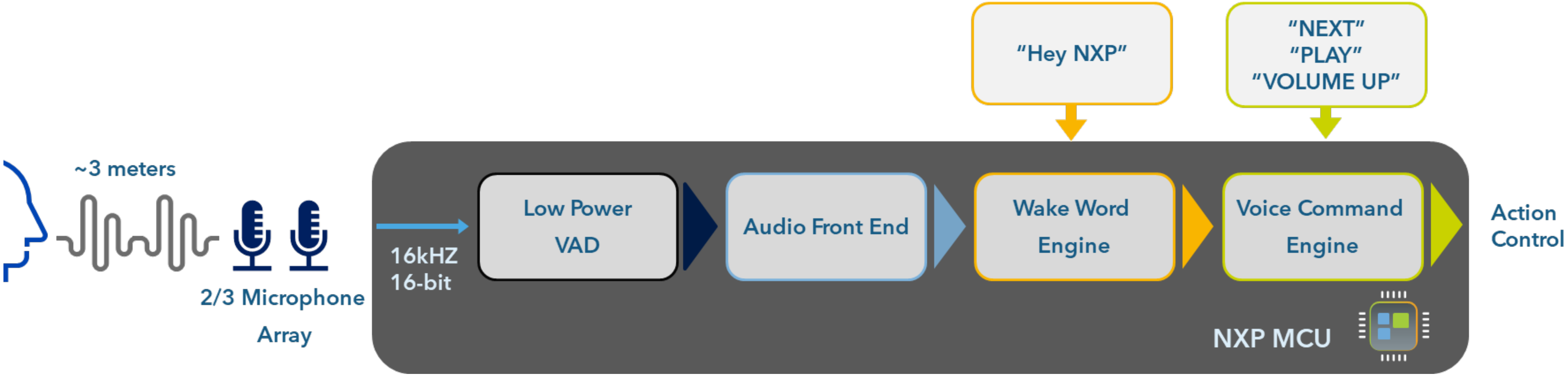
PERFORMANCE & POWER

FUNCTIONAL INTEGRATION

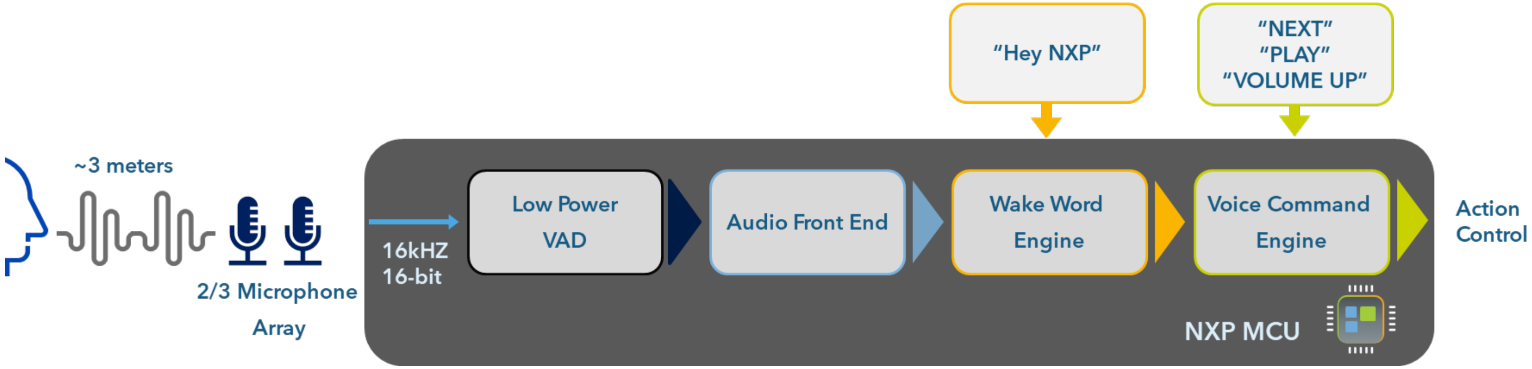
VOICE UI



Wake up your device and control action using your voice



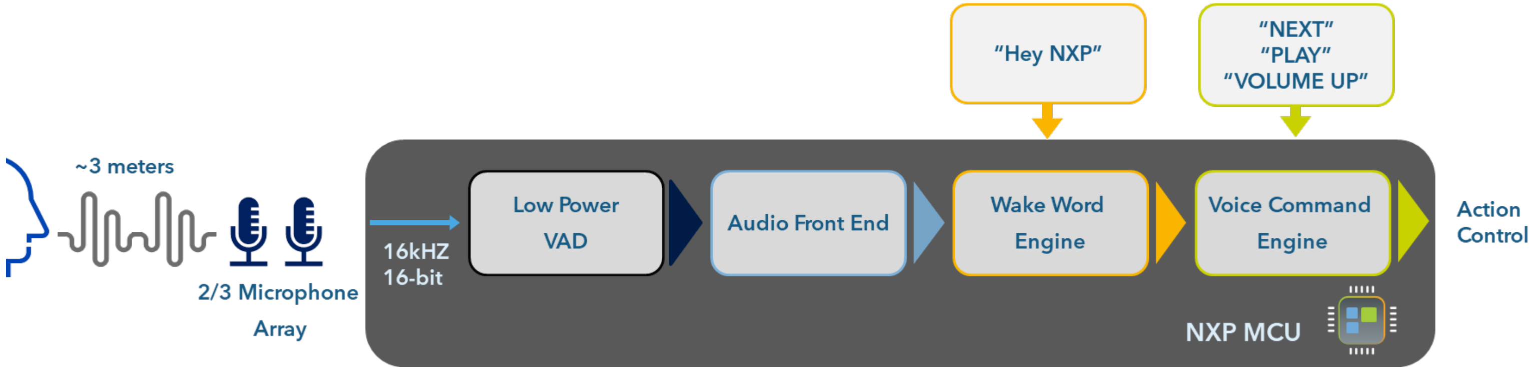
VOICE UI



- Voice UI constrained by low power

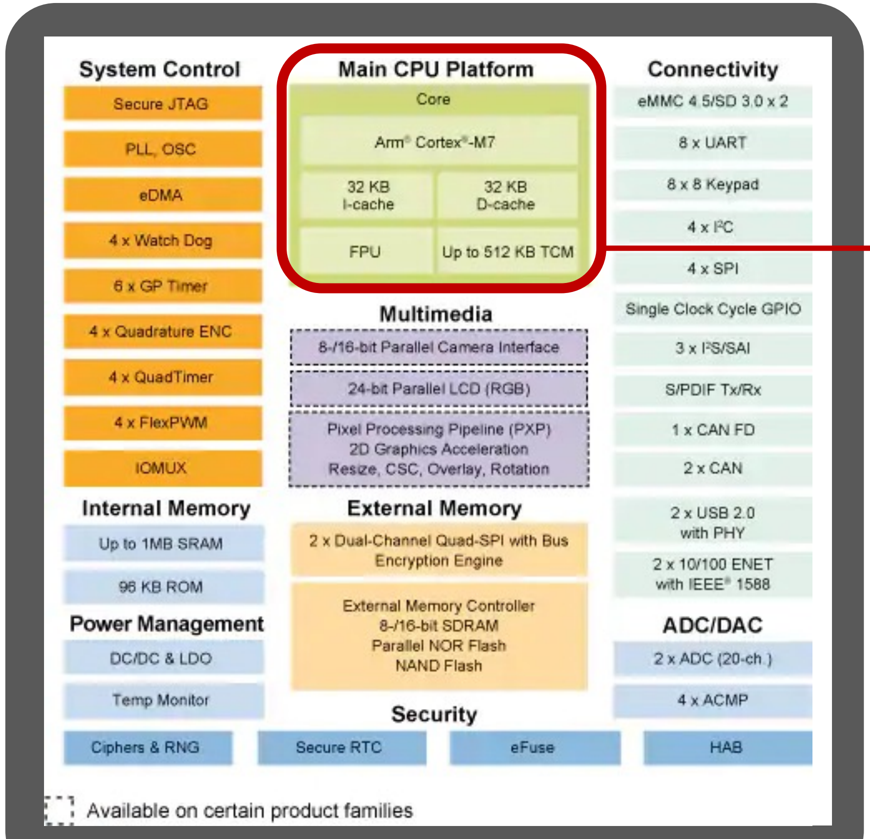
Voice UI runs on NXP i.MX RT1060

VOICE UI



- Voice UI constrained by low power

Voice UI runs on NXP i.MX RT1060



System Control
- Secure JTAG
- PLL, OSC
- eDMA
- 4 x Watch Dog
- 6 x GP Timer
- 4 x Quadrature ENC
- 4 x QuadTimer
- 4 x FlexPWM
- IOMUX

Internal Memory
- Up to 1MB SRAM
- 96 KB ROM

Power Management
- DC/DC & LDO
- Temp Monitor

Main CPU Platform
- Core
- Arm® Cortex®-M7
- 32 KB I-cache
- 32 KB D-cache
- FPU
- Up to 512 KB TCM

Multimedia
- 8-/16-bit Parallel Camera Interface
- 24-bit Parallel LCD (RGB)
- Pixel Processing Pipeline (PXP) 2D Graphics Acceleration Resize, CSC, Overlay, Rotation

External Memory
- 2 x Dual-Channel Quad-SPI with Bus Encryption Engine
- External Memory Controller 8-/16-bit SDRAM Parallel NOR Flash NAND Flash

Connectivity
- eMMC 4.5/SD 3.0 x 2
- 8 x UART
- 8 x 8 Keypad
- 4 x I²C
- 4 x SPI
- Single Clock Cycle GPIO
- 3 x I²S/SAI
- S/PDIF Tx/Rx
- 1 x CAN FD
- 2 x CAN
- 2 x USB 2.0 with PHY
- 2 x 10/100 ENET with IEEE® 1588

ADC/DAC
- 2 x ADC (20-ch.)
- 4 x ACMP

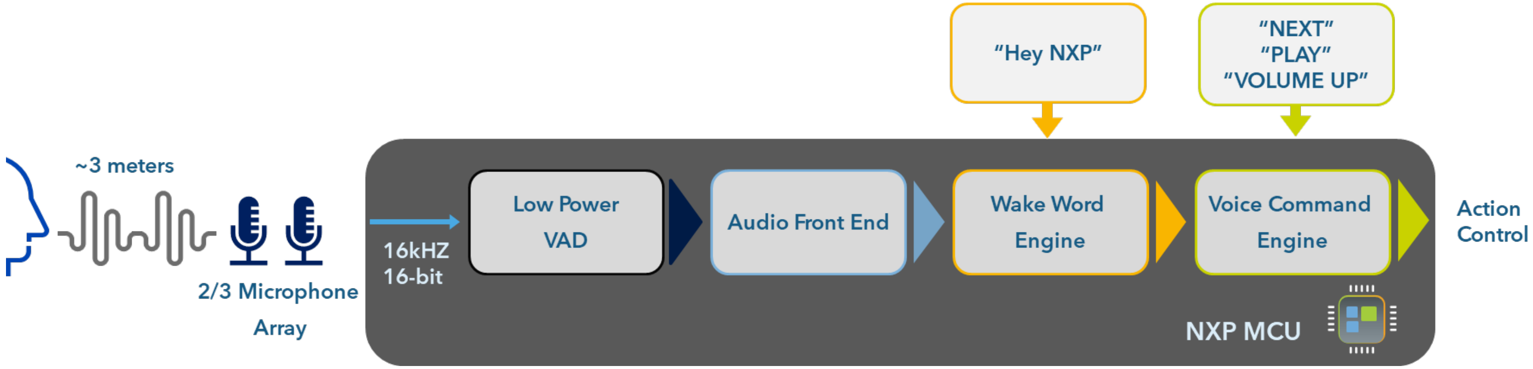
Security
- Ciphers & RNG
- Secure RTC
- eFuse
- HAB

Available on certain product families

NXP i.MX RT1060 Block diagram

Arm Cortex-M7 runs at 600MHz

VOICE UI



- Voice UI constrained by low power

  Voice UI runs on NXP i.MX RT1060

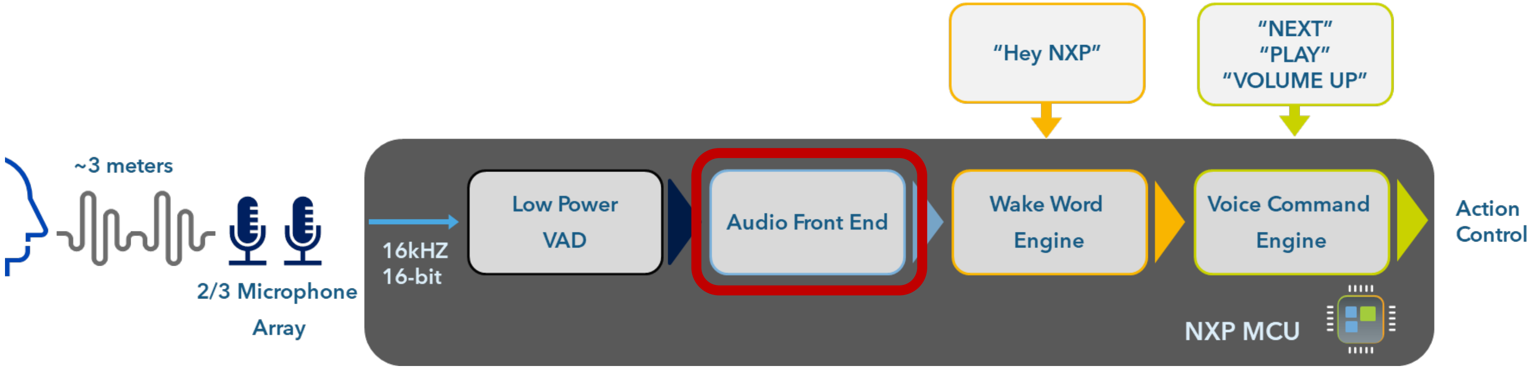
- Low latency UI

  Trigger *delay < 200ms to fit market requirements*

- High performance requirements

  False Positives (FP) on the market are ≤3 / 24h

VOICE UI



- Voice UI constrained by low power

  Voice UI runs on NXP i.MX RT1060

- Low latency UI

  Trigger *delay < 200ms to fit market requirements*

- High performance requirements

  False Positives (FP) on the market are ≤3 / 24h

$$FP = 3 * \frac{10ms}{24h * 60\min * 60s}$$

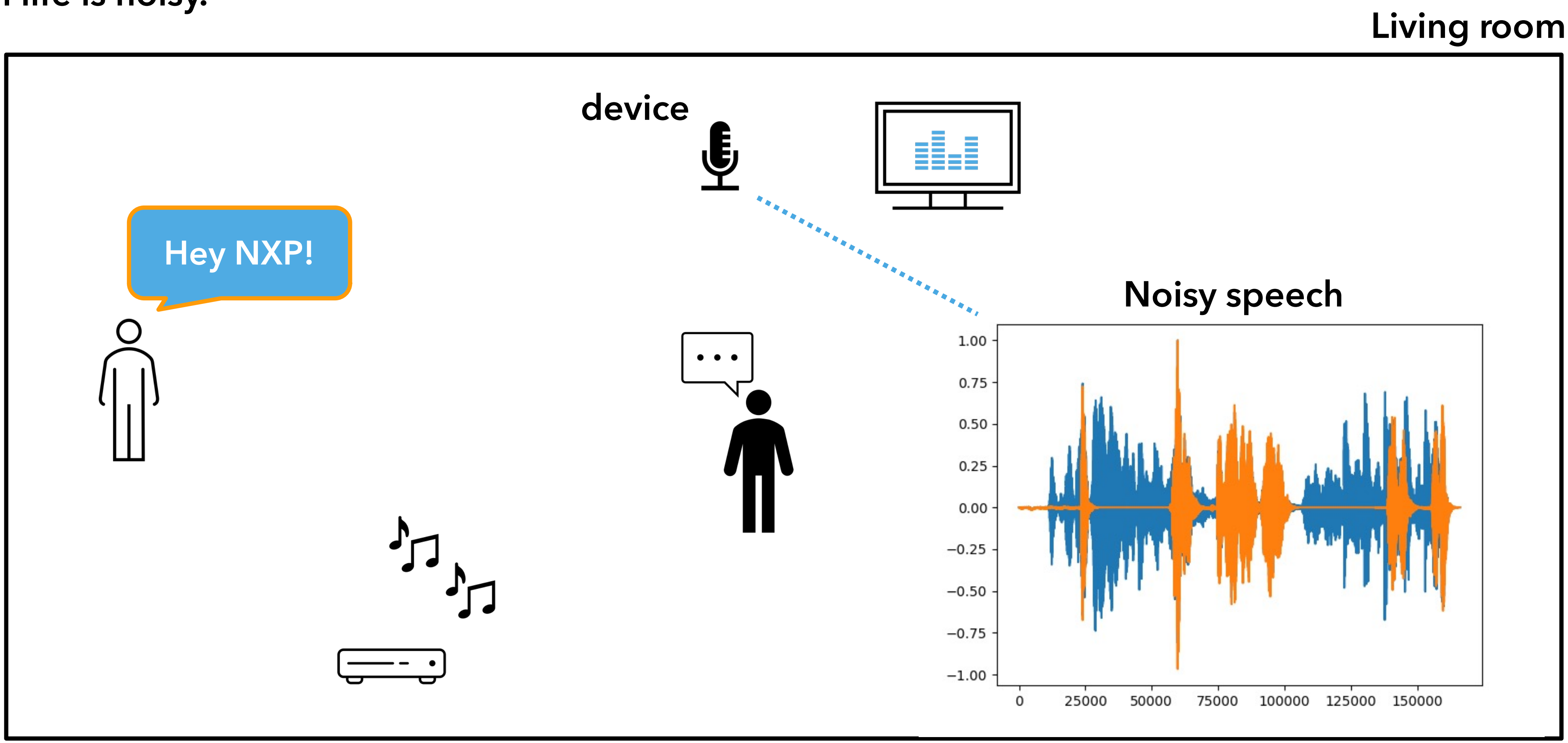
$$FPrate = 34.10^{-6}\%$$

$$TNrate = 99.99996\%$$

Very high requirements!!

WHY DO WE NEED AUDIO FRONT END?

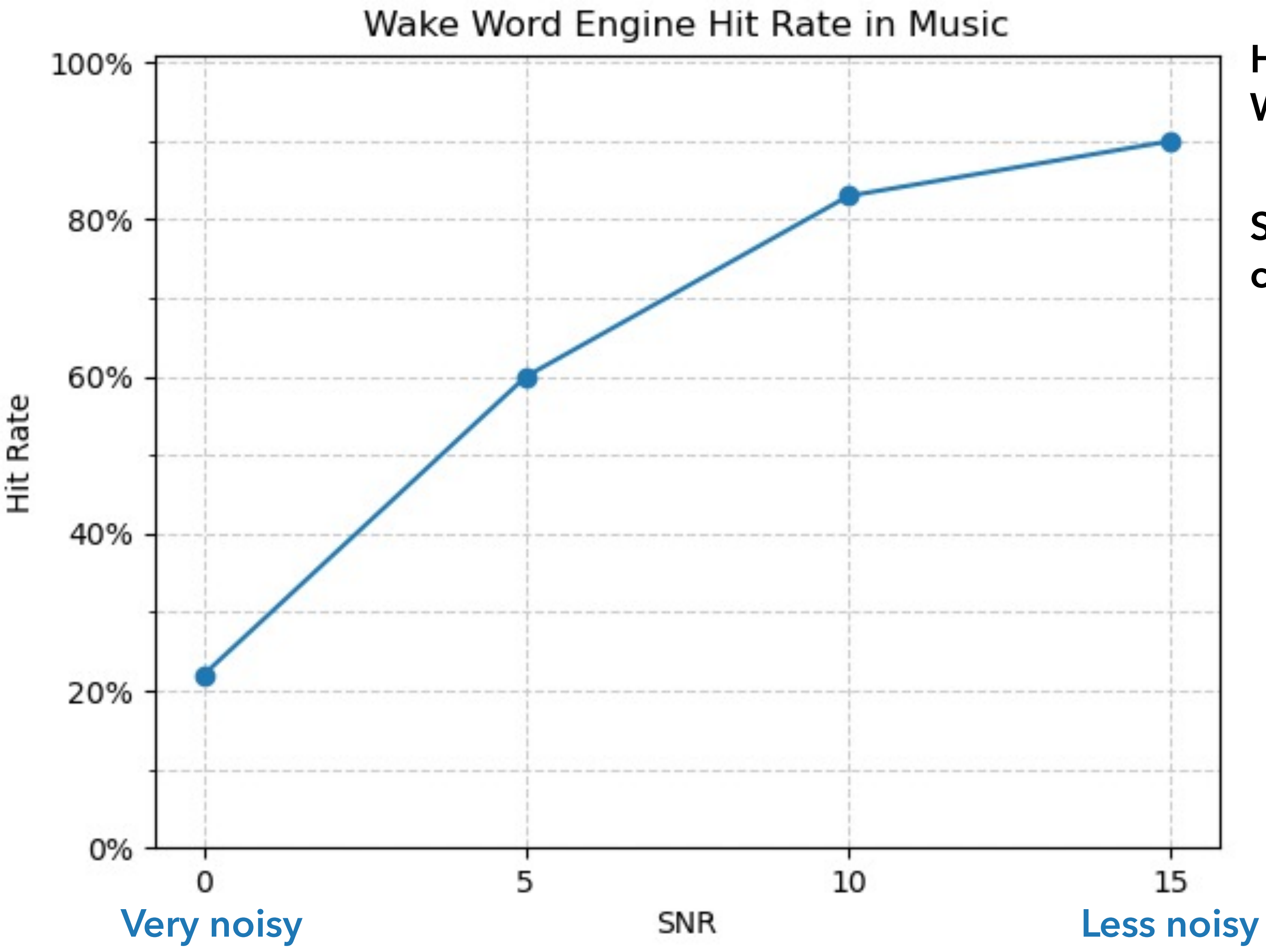
Real life is noisy.



Combination of speech and noise

Cocktail party problem (Cherry, 1953)

WHY DO WE NEED AUDIO FRONT END?



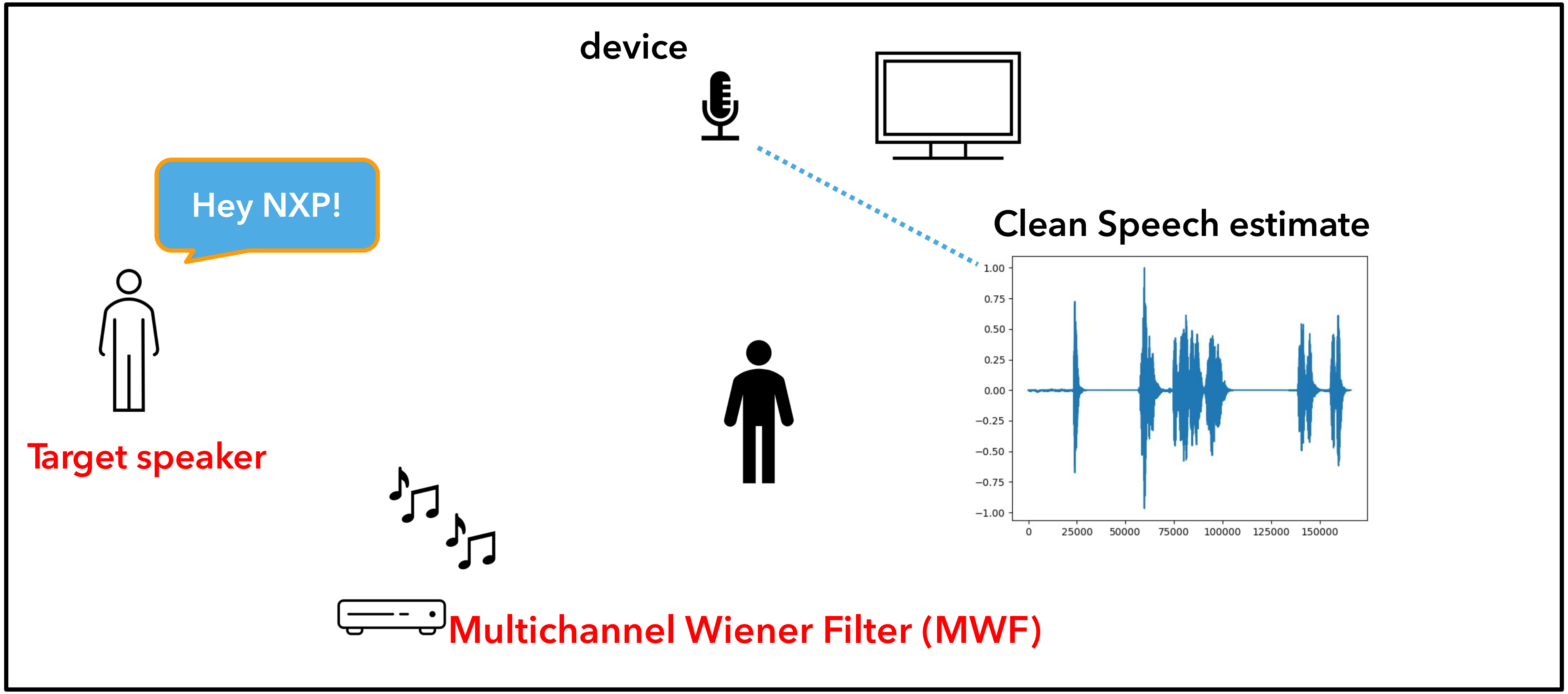
Hit Rate: Percentage of well detected Wake Word

SNR (signal-to-noise ratio): Level of speech compared to level of noise

Performance drops when the Signal-to-noise ratio (SNR) decreases.

WHY DO WE NEED AUDIO FRONT END?

Real life is noisy.



Living room

device

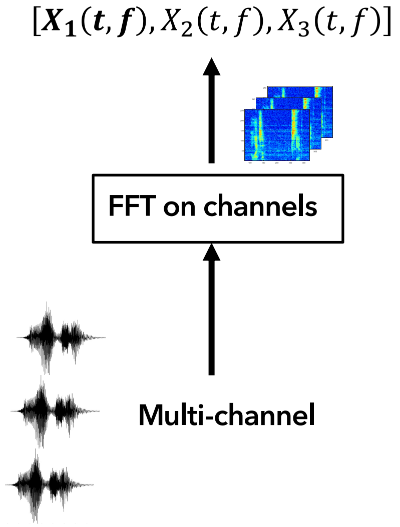
Hey NXP!

Target speaker

Clean Speech estimate

Multichannel Wiener Filter (MWF)

From classical hybrid Multichannel Wiener Filter…



Parameters

Time frame: 10ms, 16kHz

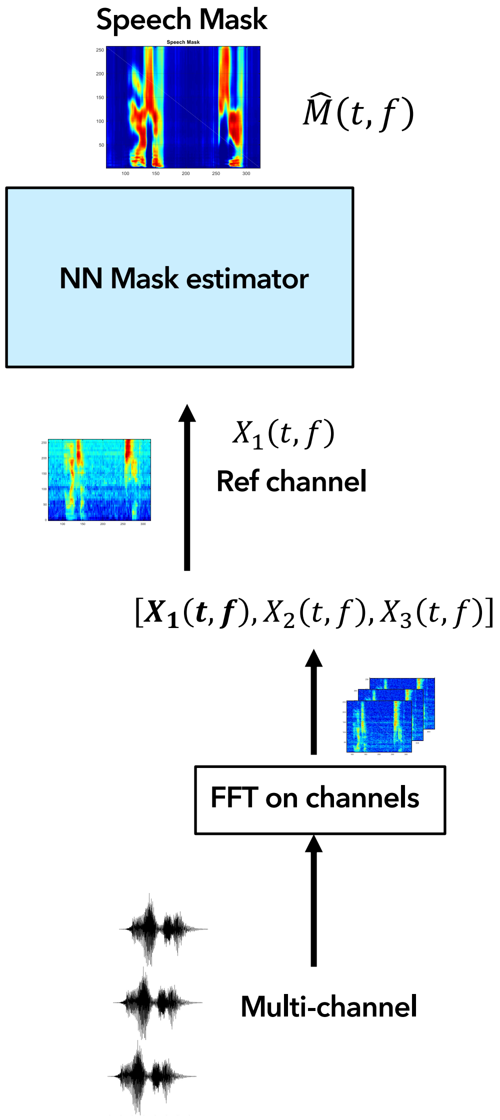
FFT size: 512 pts

$$\boldsymbol{X_1(t,f)}$$

t : frame index

f : frequency bin

From classical hybrid Multichannel Wiener Filter …



Speech Mask

$\widehat{M}(t,f)$

NN Mask estimator

$X_1(t,f)$

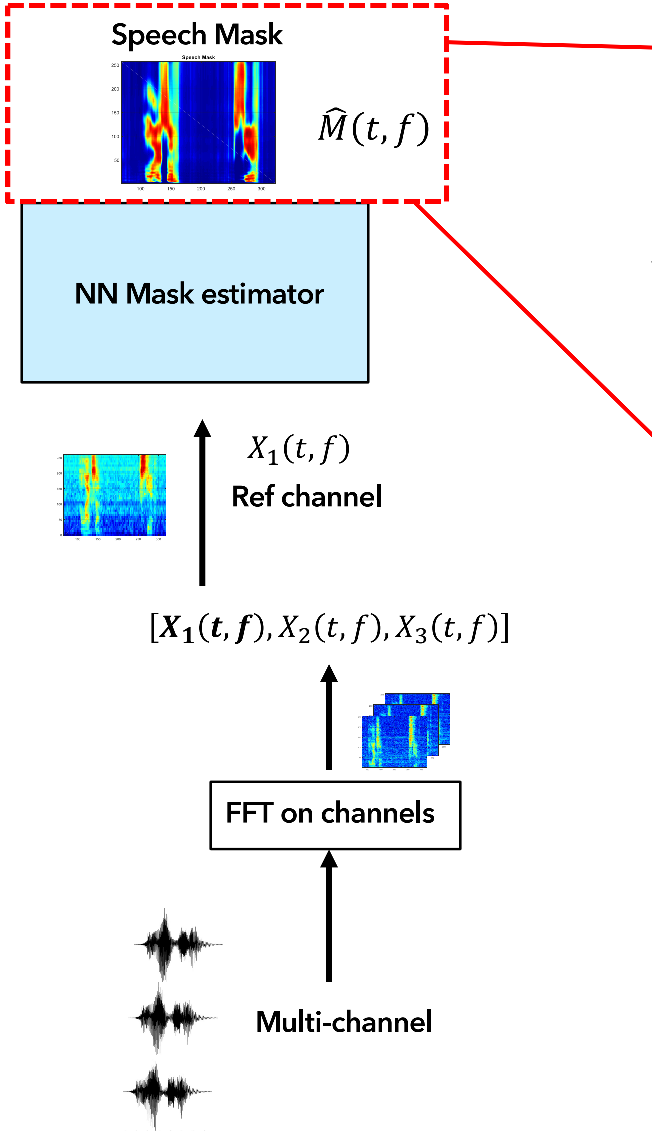
Ref channel

$[\boldsymbol{X_1(t,f)}, X_2(t,f), X_3(t,f)]$

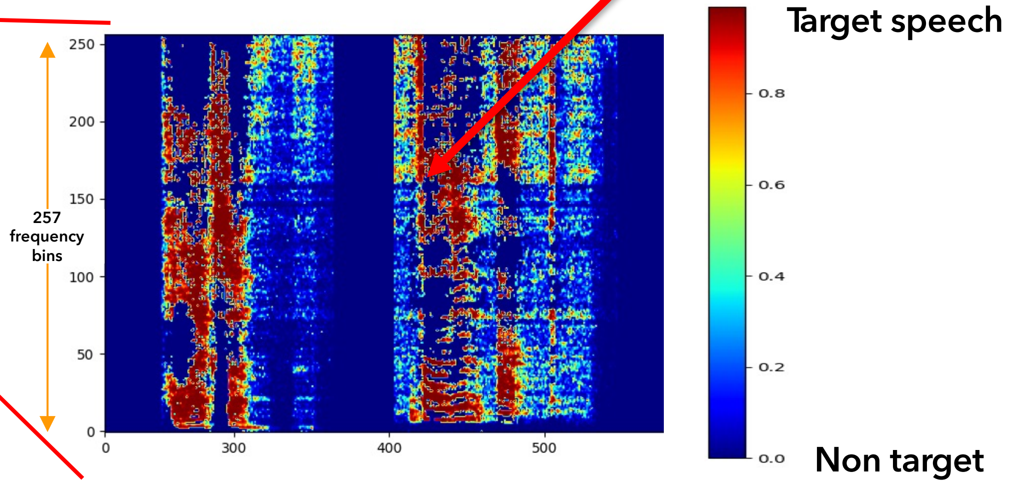
FFT on channels

Multi-channel

From classical hybrid MWF…



0.3 Probability of the time frequency element to belong to the target speech



Input signal

Speech

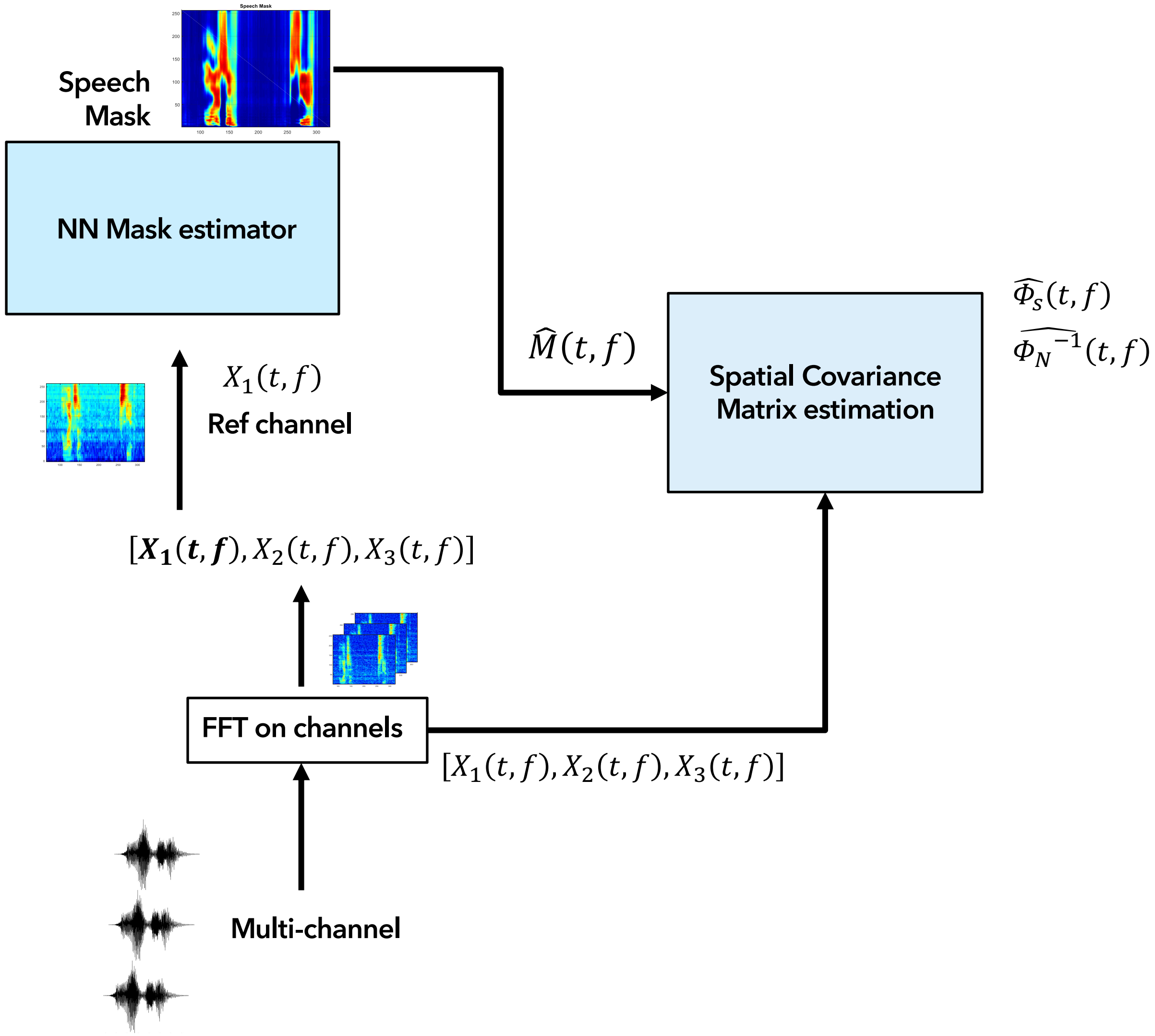
Noise

$$X(t,f) = S(t,f) + N(t,f)$$

Ideal Ratio Mask:

$$IRM(t,f) = \left(\frac{|S(t,f)|^2}{|S(t,f)|^2 + |N(t,f)|^2}\right)^{1/2}$$

From classical hybrid Multichannel Wiener Filter …



Speech Mask

NN Mask estimator

$X_1(t,f)$

Ref channel

$[\boldsymbol{X_1(t,f)}, X_2(t,f), X_3(t,f)]$

FFT on channels

Multi-channel

$[X_1(t,f), X_2(t,f), X_3(t,f)]$

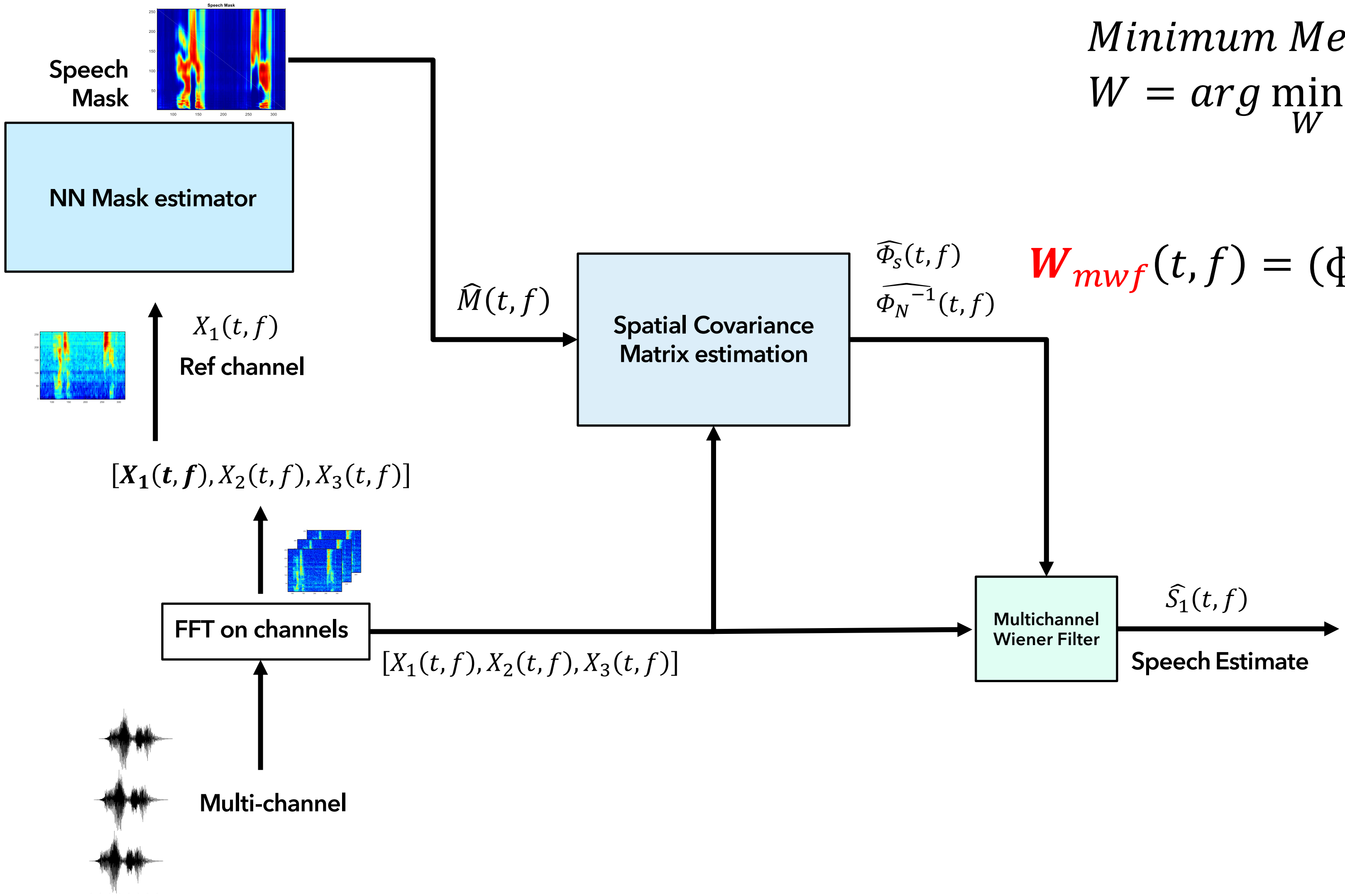
$\widehat{M}(t,f)$

Spatial Covariance Matrix estimation

$\widehat{\Phi_S}(t,f)$

$\widehat{\Phi_N{}^{-1}}(t,f)$

From classical hybrid Multichannel Wiener Filter …

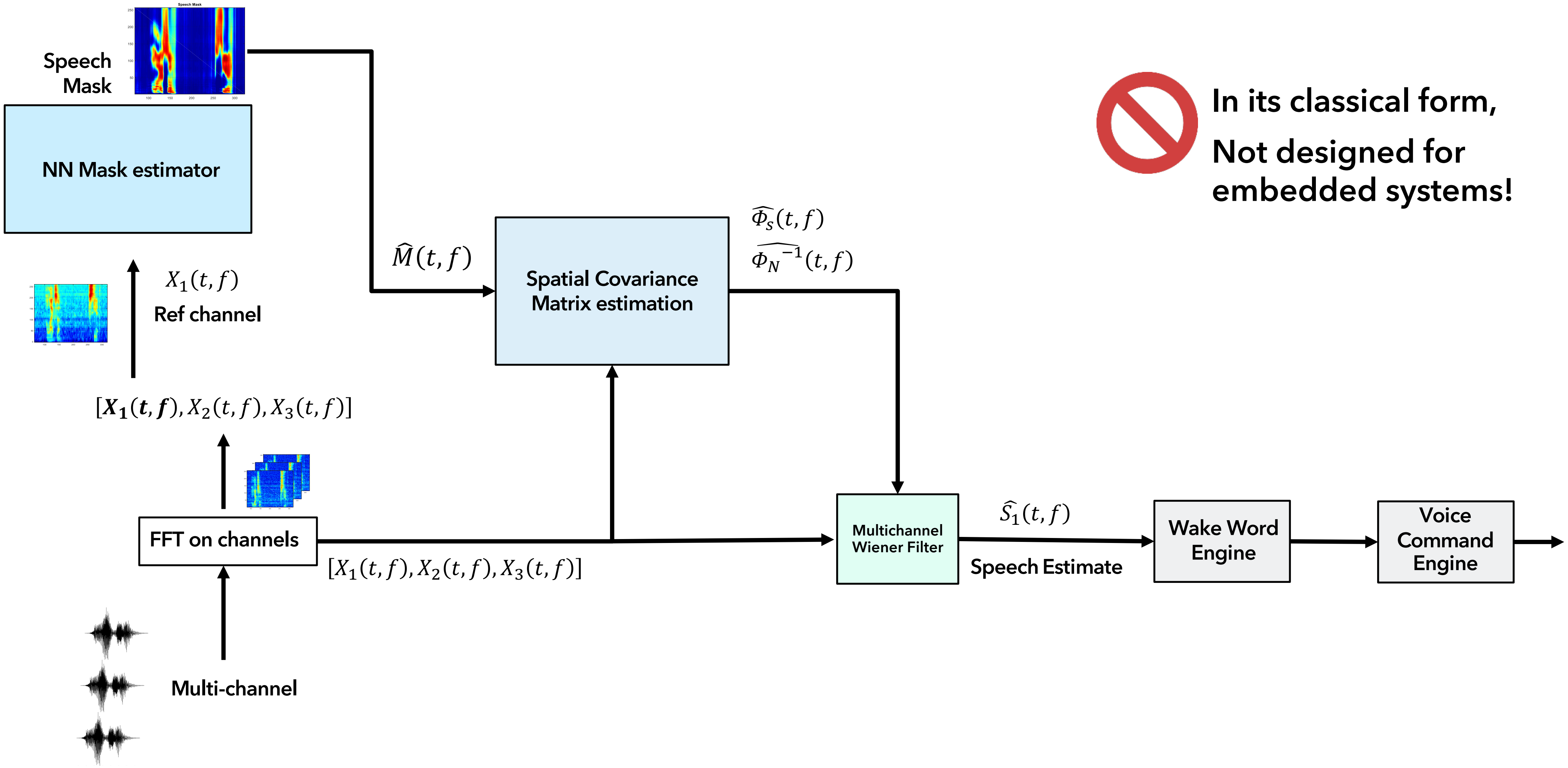

Speech Mask
NN Mask estimator
$\widehat{M}(t,f)$
$X_1(t,f)$
Ref channel
$[\boldsymbol{X_1(t,f)}, X_2(t,f), X_3(t,f)]$
Spatial Covariance Matrix estimation
$\widehat{\Phi_S}(t,f)$
$\widehat{\Phi_N^{-1}}(t,f)$
FFT on channels
$[X_1(t,f), X_2(t,f), X_3(t,f)]$
Multichannel Wiener Filter
$\widehat{S_1}(t,f)$
Speech Estimate
Multi-channel

Minimum Mean Square Error (MMSE):

$$W = arg \min_{W} \boldsymbol{E}[|S_1(t,f) - \boldsymbol{W}^H X(t,f)|^2]$$

$$\boldsymbol{W}_{mwf}(t,f) = (\phi_S(t,f) + \phi_N(t,f))^{-1} \phi_S(t,f)\, \boldsymbol{e}_1$$

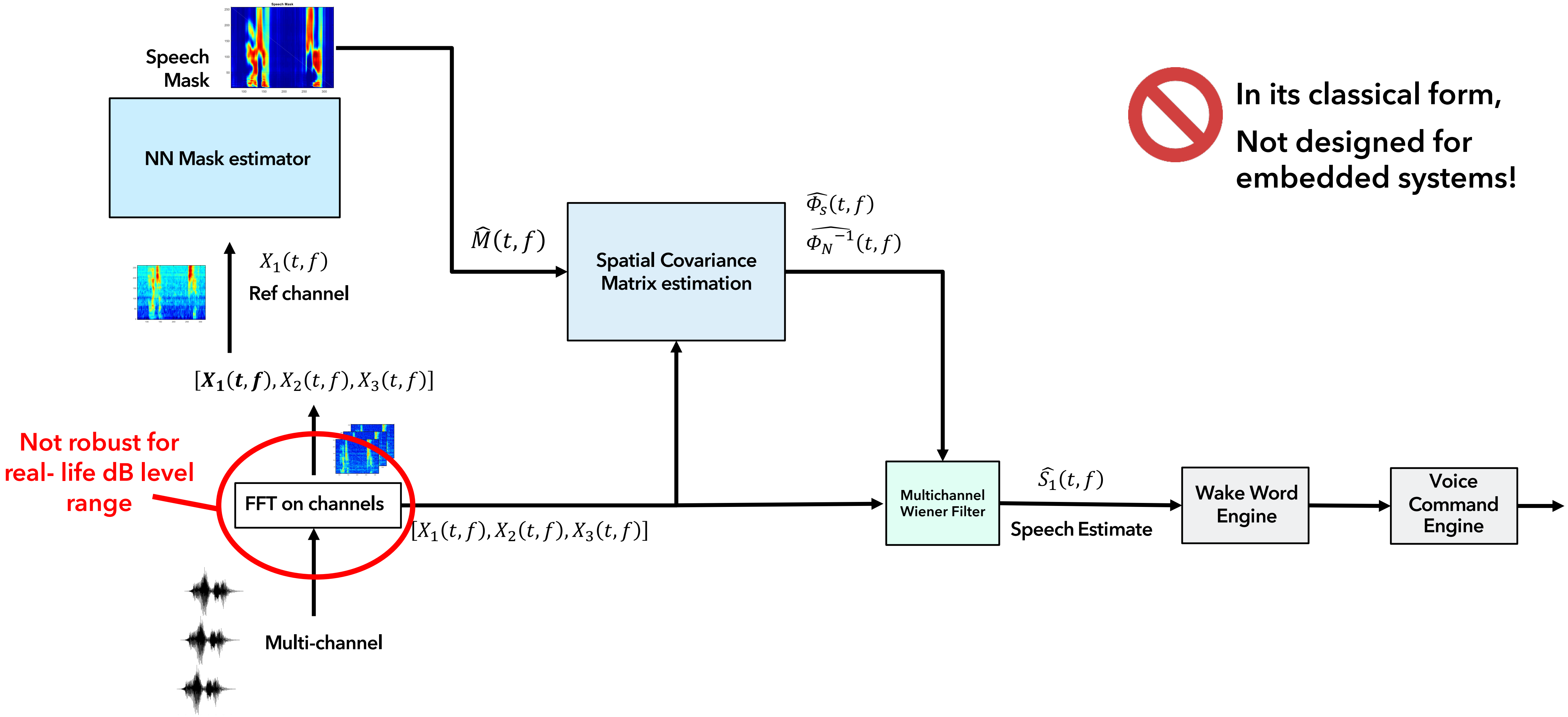
From classical hybrid Multichannel Wiener Filter …


Speech
Mask
NN Mask estimator
$X_1(t,f)$
Ref channel
$[\boldsymbol{X_1}(\boldsymbol{t},\boldsymbol{f}), X_2(t,f), X_3(t,f)]$
FFT on channels
Multi-channel
$\widehat{M}(t,f)$
Spatial Covariance Matrix estimation
$\widehat{\Phi_S}(t,f)$
$\widehat{\Phi_N}^{-1}(t,f)$
$[X_1(t,f), X_2(t,f), X_3(t,f)]$
Multichannel Wiener Filter
$\widehat{S_1}(t,f)$
Speech Estimate
Wake Word Engine
Voice Command Engine

In its classical form,

Not designed for embedded systems!

From classical hybrid Multichannel Wiener Filter …



Speech Mask

NN Mask estimator

$X_1(t,f)$

Ref channel

$[\boldsymbol{X_1(t,f)}, X_2(t,f), X_3(t,f)]$

Not robust for real- life dB level range

FFT on channels

Multi-channel

$\widehat{M}(t,f)$

Spatial Covariance Matrix estimation

$\widehat{\Phi_S}(t,f)$

$\widehat{\Phi_N{}^{-1}}(t,f)$

$[X_1(t,f), X_2(t,f), X_3(t,f)]$

Multichannel Wiener Filter

$\widehat{S_1}(t,f)$

Speech Estimate

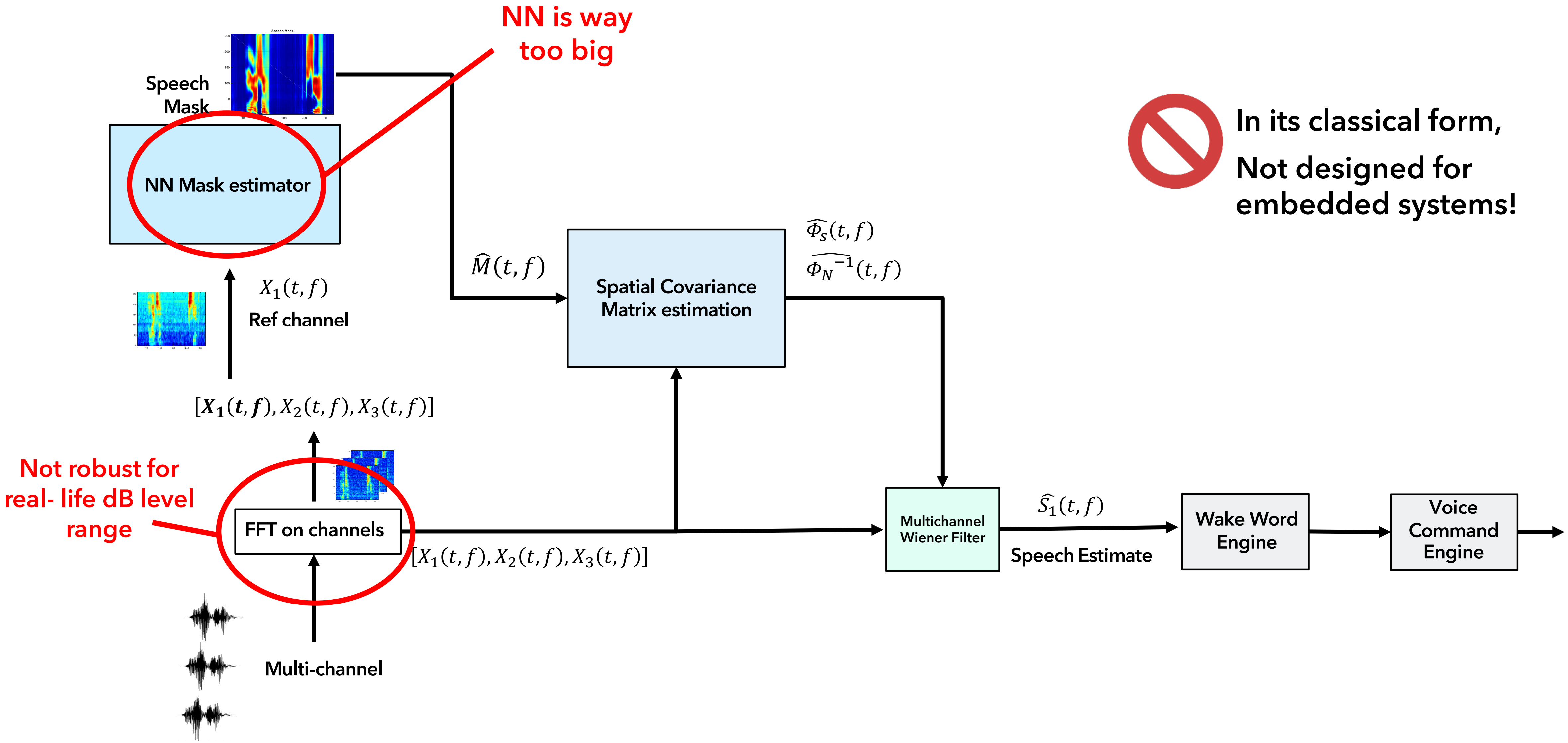
Wake Word Engine

Voice Command Engine

In its classical form,

Not designed for embedded systems!

From classical hybrid Multichannel Wiener Filter …



NN is way too big

Speech Mask

NN Mask estimator

$X_1(t,f)$

Ref channel

$\widehat{M}(t,f)$

Spatial Covariance Matrix estimation

$\widehat{\Phi_S}(t,f)$

$\widehat{\Phi_N}^{-1}(t,f)$

$[\boldsymbol{X_1(t,f)}, X_2(t,f), X_3(t,f)]$

Not robust for real- life dB level range

FFT on channels

$[X_1(t,f), X_2(t,f), X_3(t,f)]$

Multi-channel

Multichannel Wiener Filter

$\widehat{S}_1(t,f)$

Speech Estimate

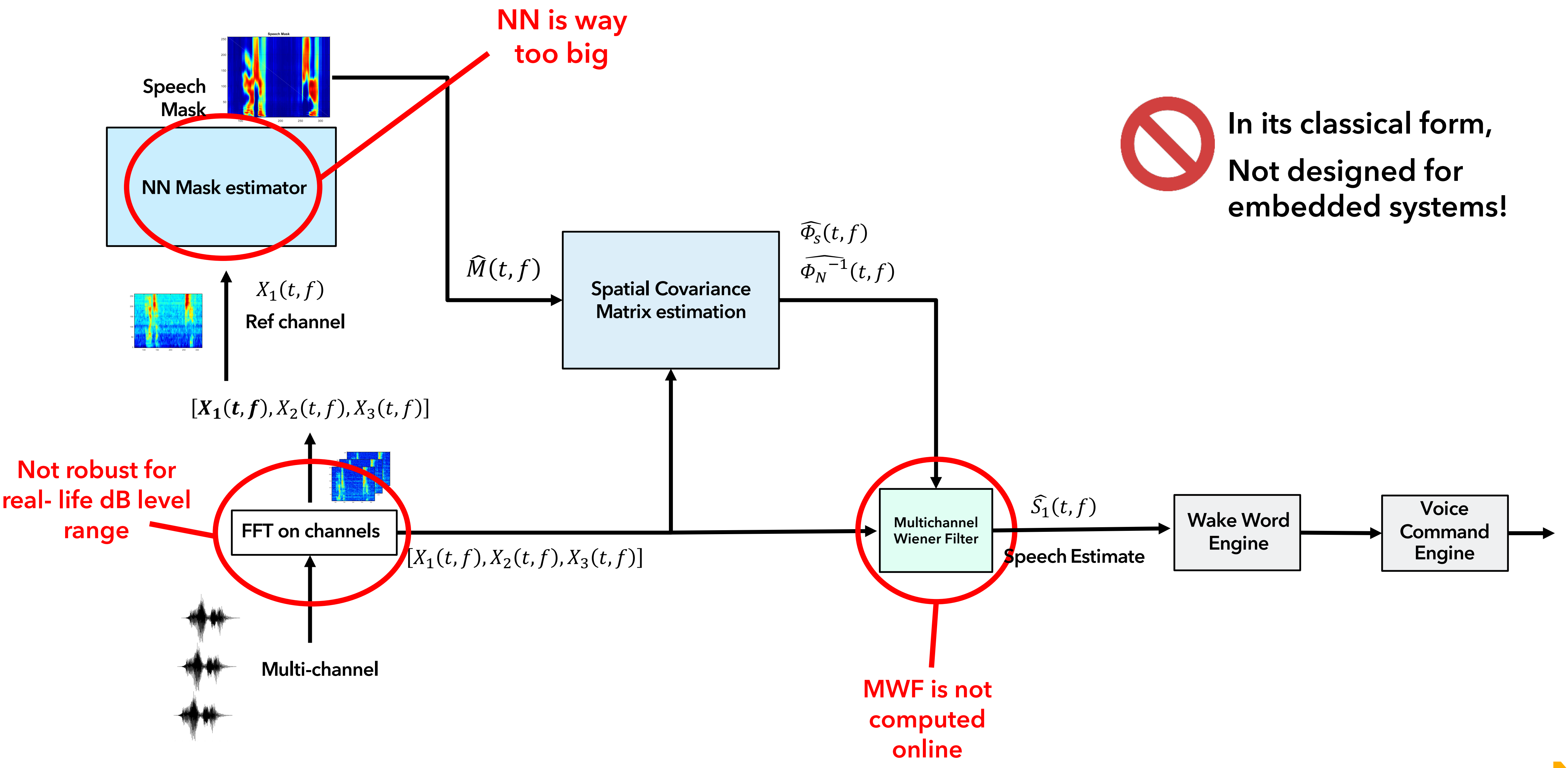
Wake Word Engine

Voice Command Engine

In its classical form,

Not designed for embedded systems!

From classical hybrid Multichannel Wiener Filter …


Speech Mask
NN Mask estimator
$X_1(t,f)$
Ref channel
$[\boldsymbol{X_1}(\boldsymbol{t},\boldsymbol{f}), X_2(t,f), X_3(t,f)]$
FFT on channels
Multi-channel
$\widehat{M}(t,f)$
Spatial Covariance Matrix estimation
$\widehat{\Phi_S}(t,f)$
$\widehat{\Phi_N^{-1}}(t,f)$
$[X_1(t,f), X_2(t,f), X_3(t,f)]$
Multichannel Wiener Filter
$\widehat{S_1}(t,f)$
Speech Estimate
Wake Word Engine
Voice Command Engine

NN is way too big

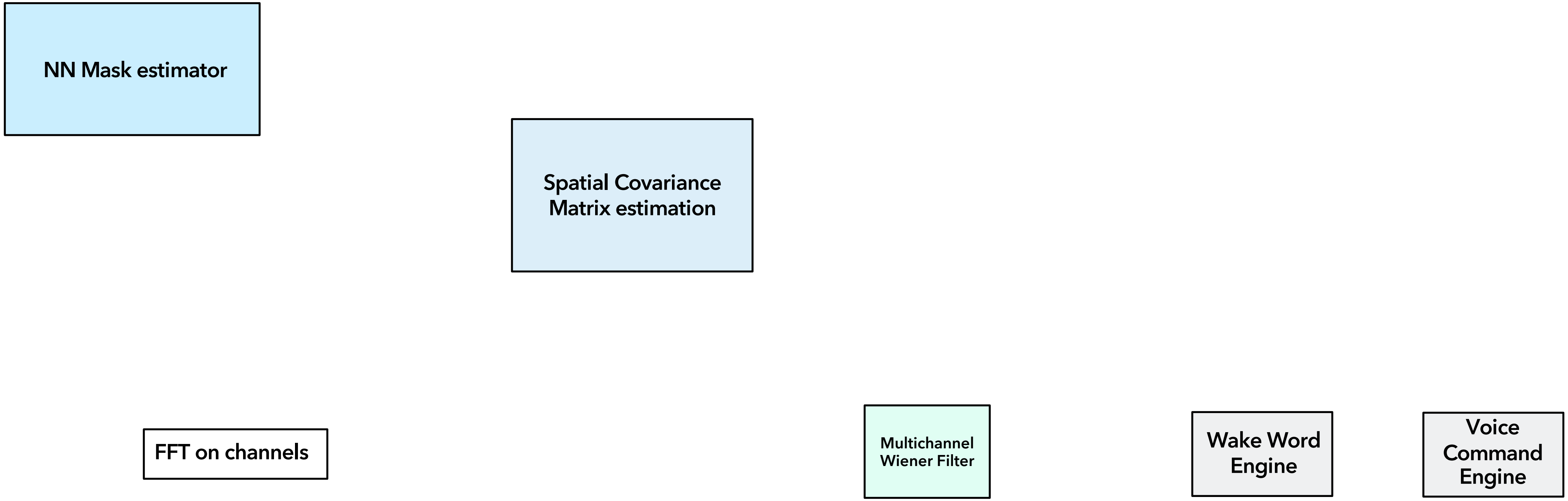
Not robust for real- life dB level range

MWF is not computed online

In its classical form,

Not designed for embedded systems!

From classical hybrid Multichannel Wiener Filter …



NN Mask estimator

Spatial Covariance Matrix estimation

FFT on channels

Multichannel Wiener Filter

Wake Word Engine

Voice Command Engine

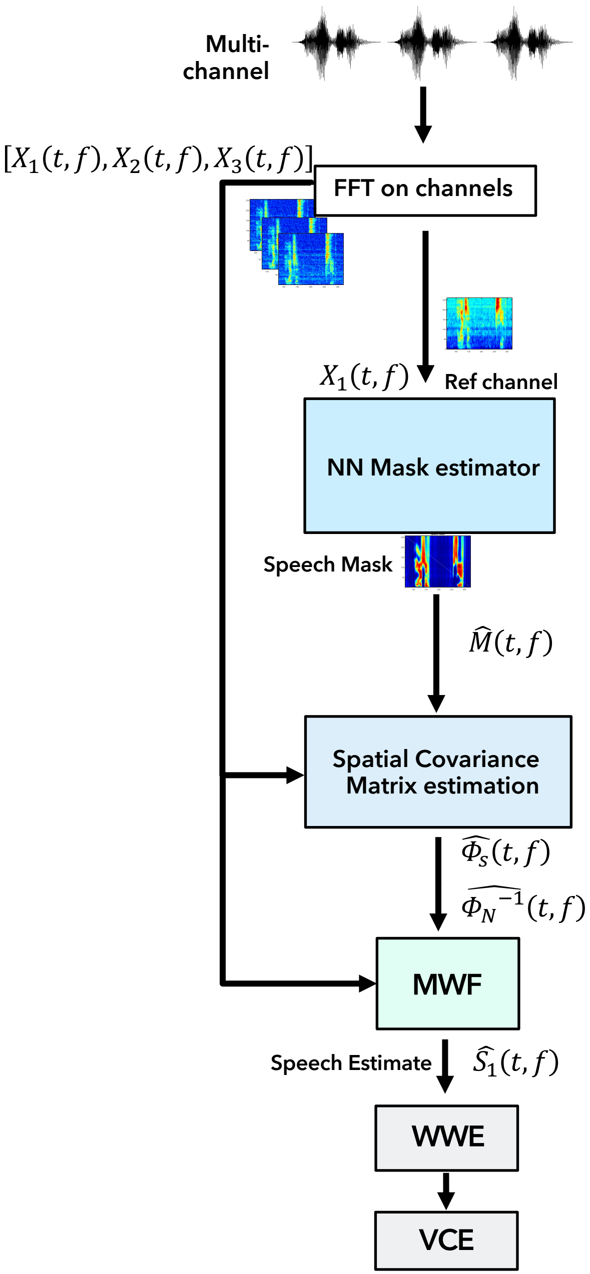
Challenges of the embedded solution

- Main algorithm are Wake Word and Voice Command Engines block

Audio Front End is added, so we have a size constraint on platform integration

- Focus on increase performances of the Wake Word Detection.

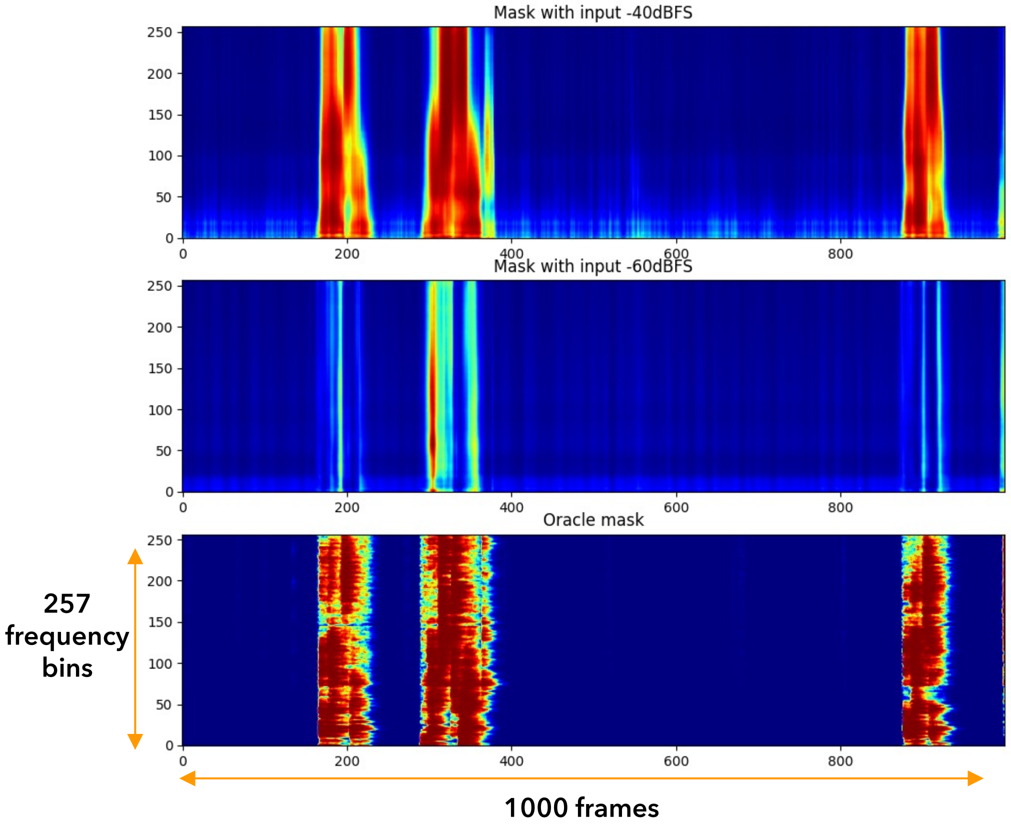
We didn't see any clear correlation with direct improvement of classic metrics like Signal-to-Noise ratio, Signal-to-Distortion ratio…



NN robustness to input dB level

Neural Network not robust to input dB level

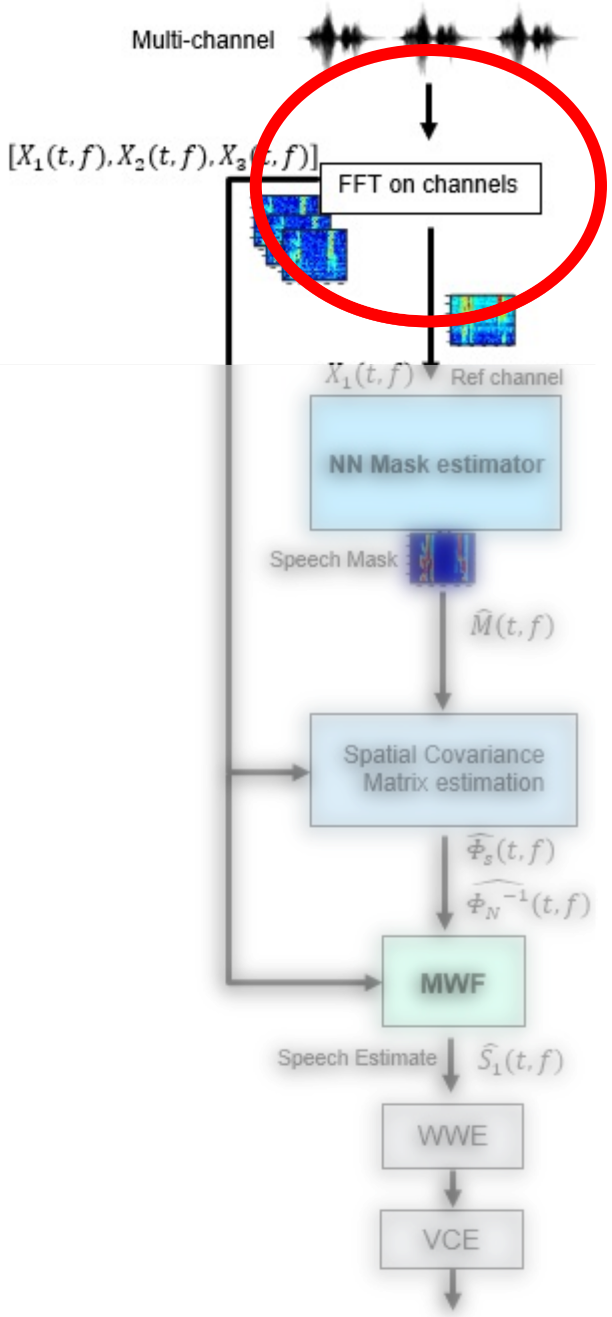
- Input dB level [-40dB full scale (dBFS),-60dBFS]
- Trained at -40dBFS, we see drop of performances at -60dBFS



Mask -40dBFS

Mask -60dBFS

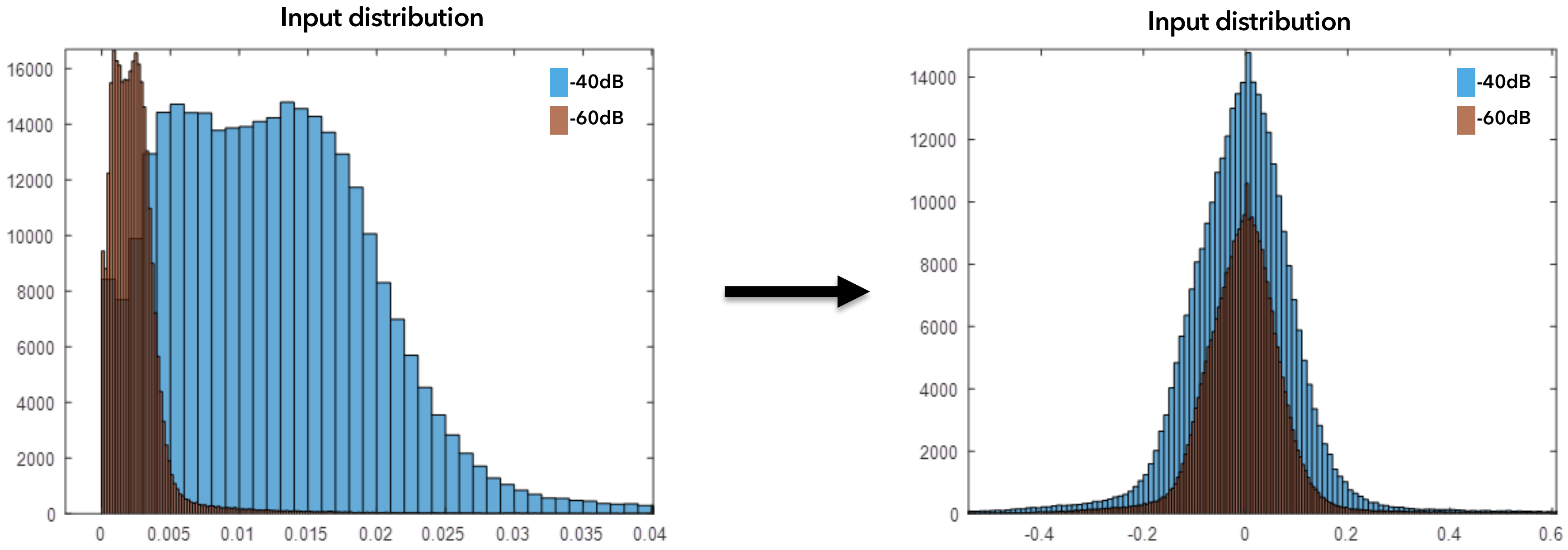
Oracle Mask

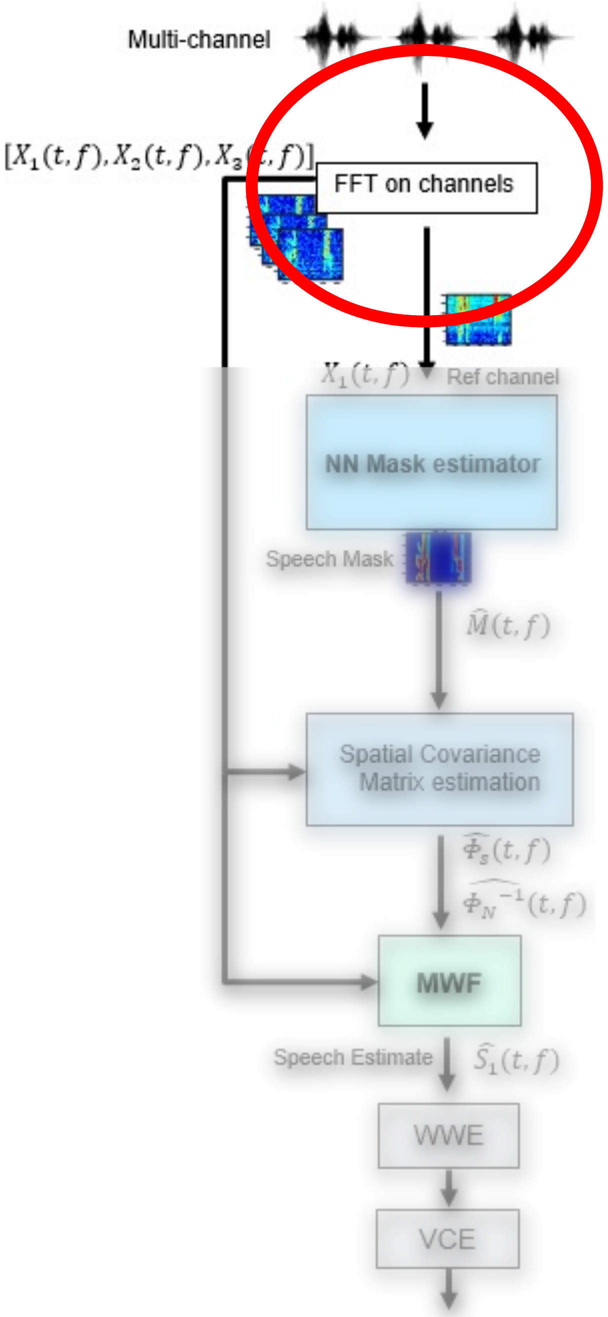


NN robustness to input dB level

- Apply transformation on input data

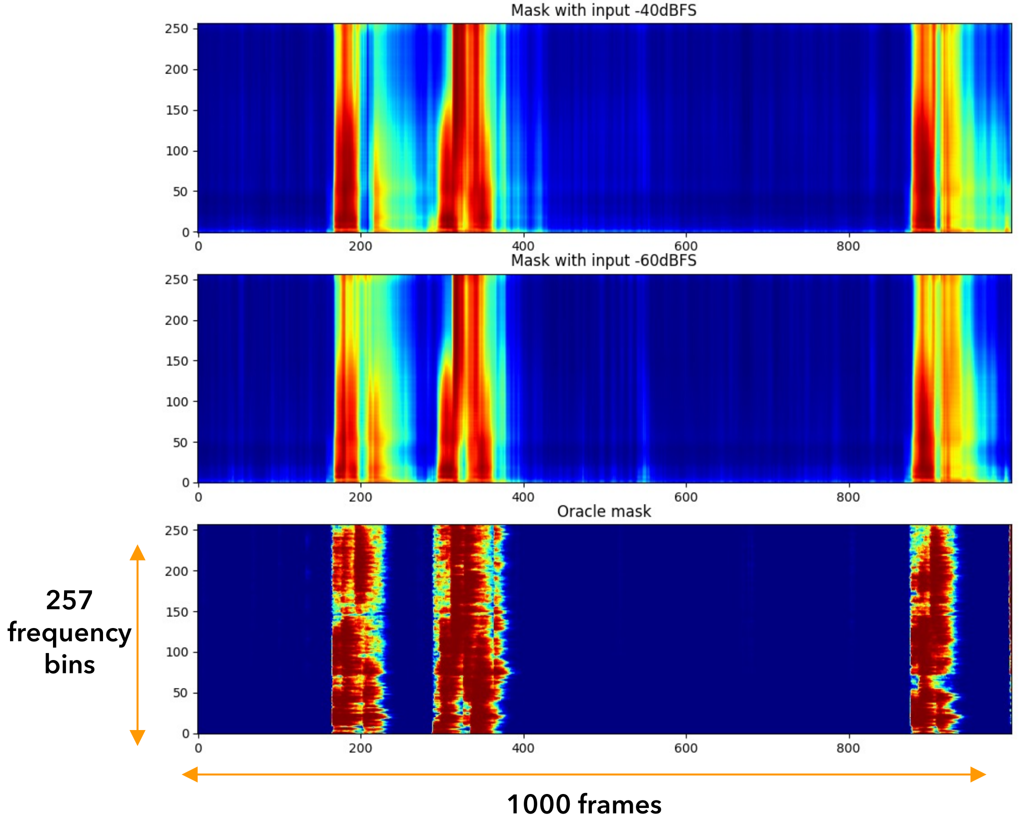
Normalize the data based on energy and root compression to arrange distribution





NN robustness to input dB level

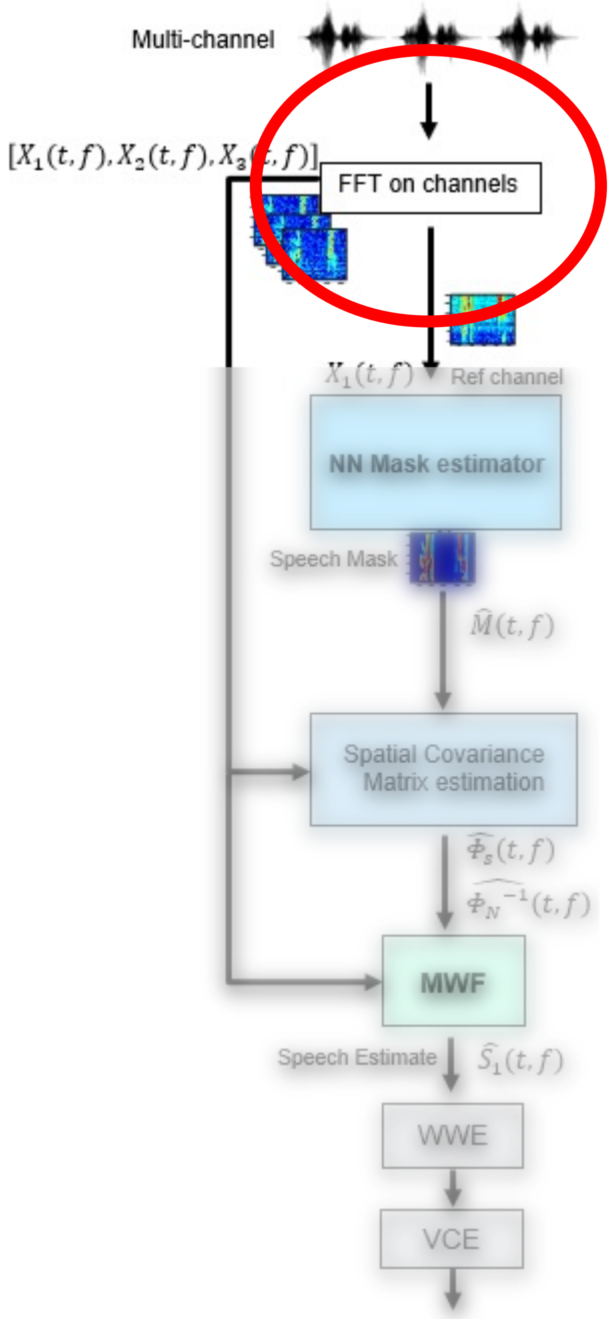
- NN is now robust to input dB level range [-40dBFS, -60dBFS]!



Mask -40dBFS

Mask -60dBFS

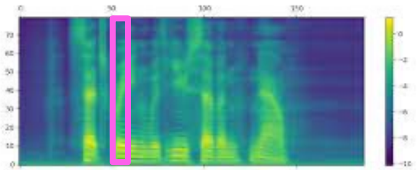
Oracle Mask



NN optimization

NN too big to fit on platform

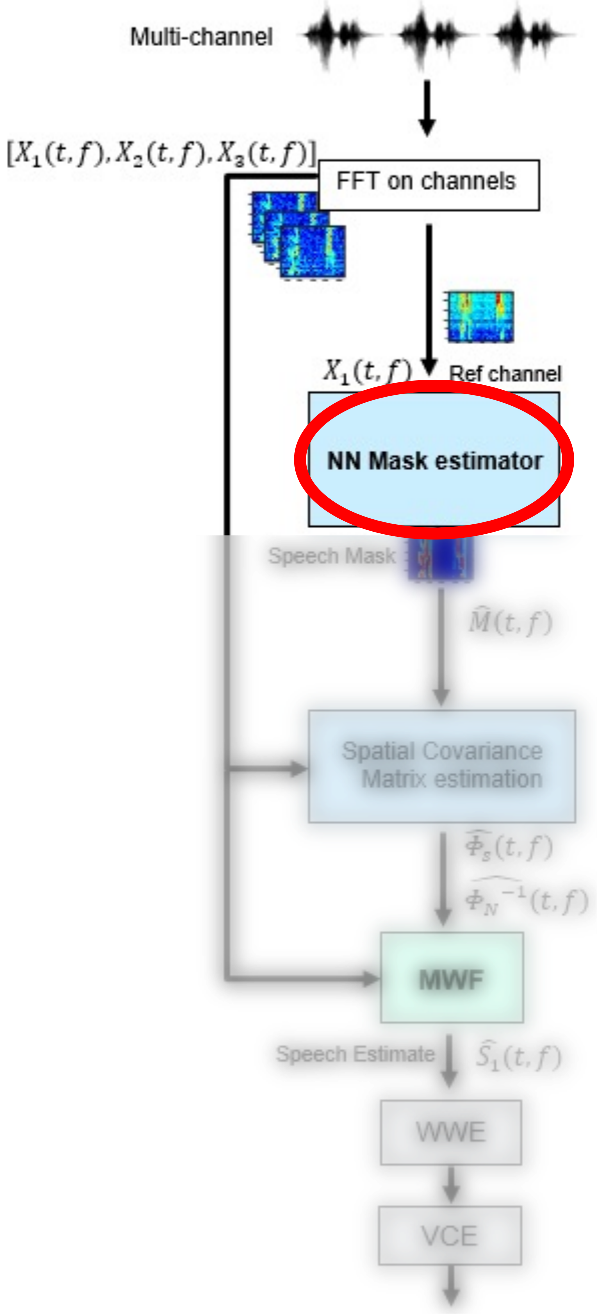
$X_1(t,f)$



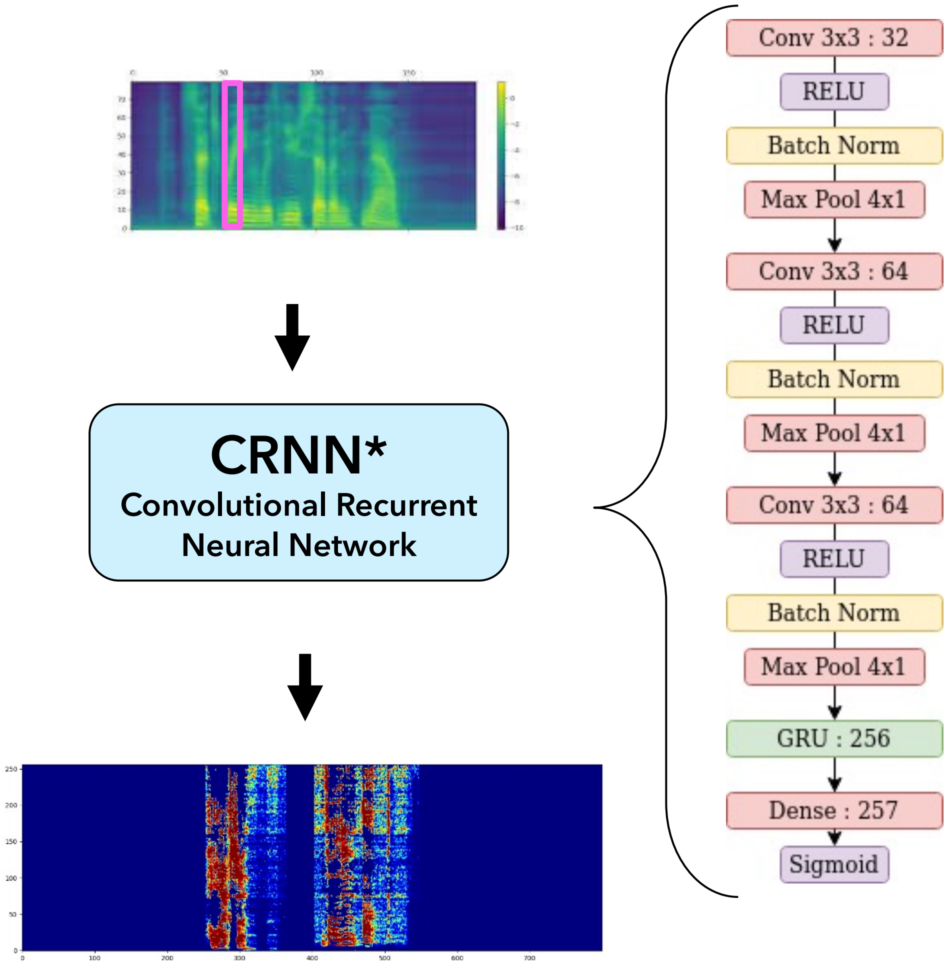
21 consecutive frames

x

257 FFT frequency bins



NN optimization



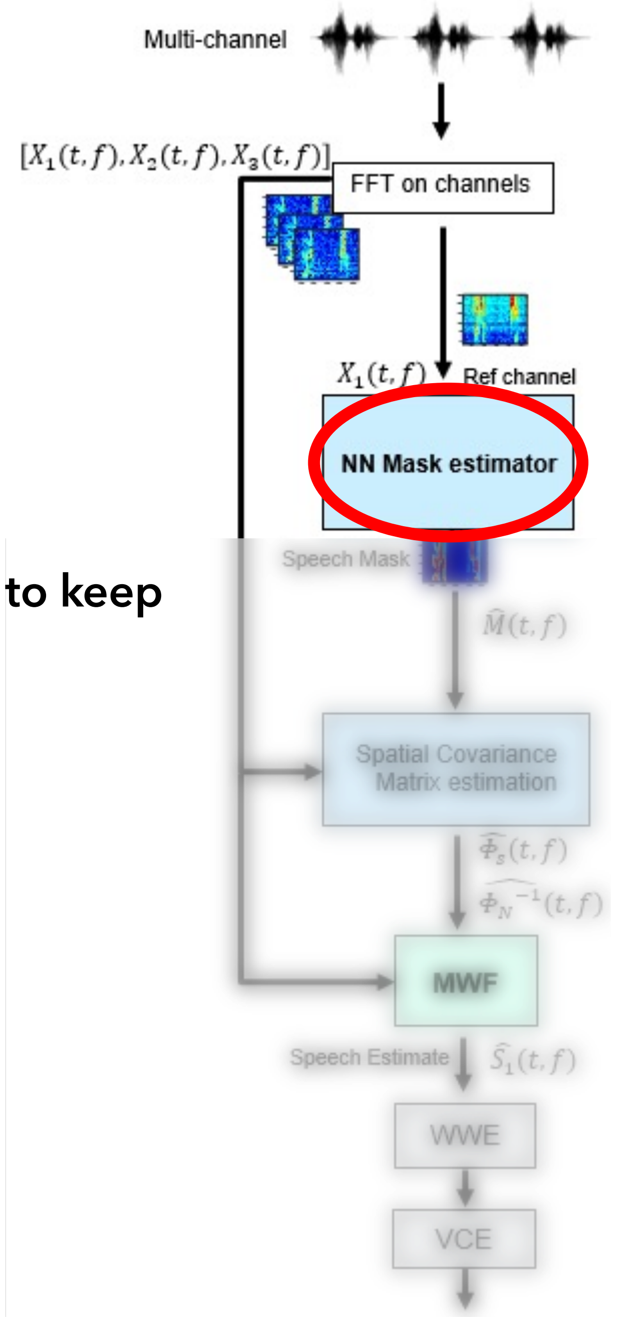
CRNN*

Convolutional Recurrent Neural Network

Parameters: 470k

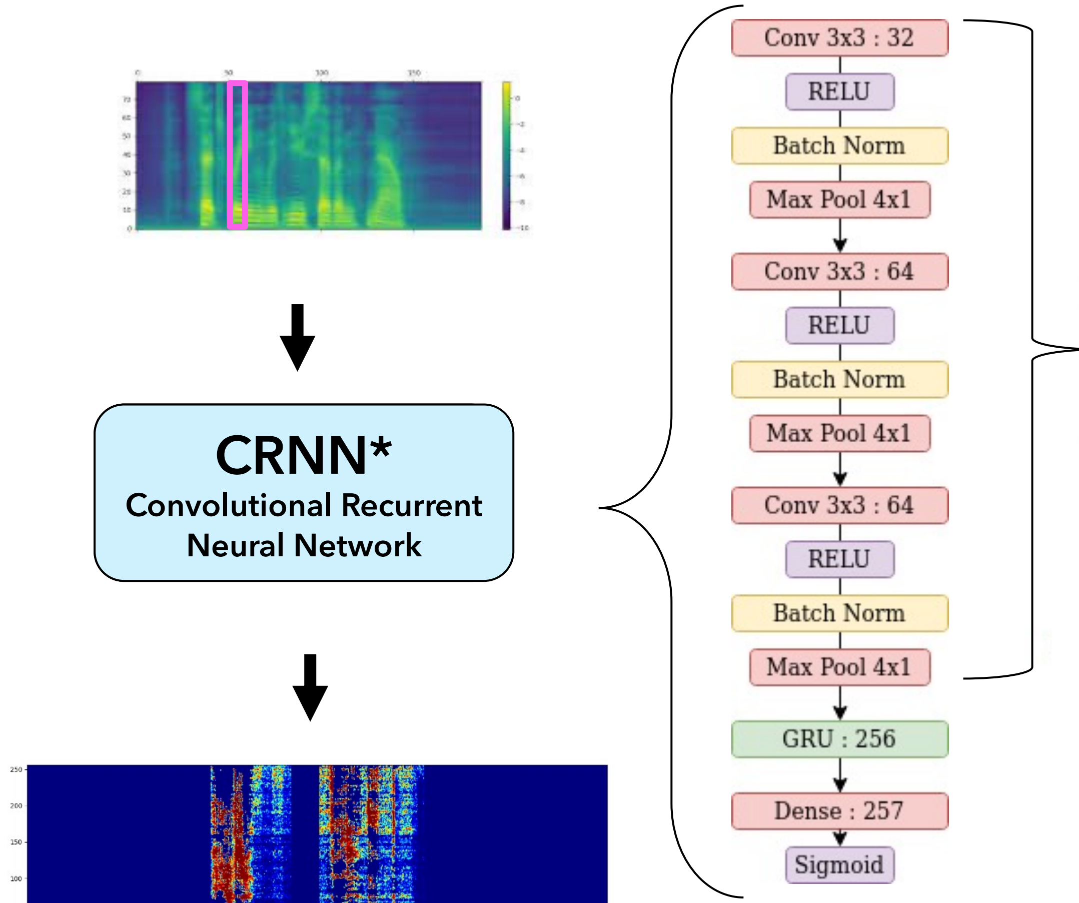
Number of MACs: 33M

- CPU should run at 63240 MHz to keep real-time predictions
- Objective <300MHz

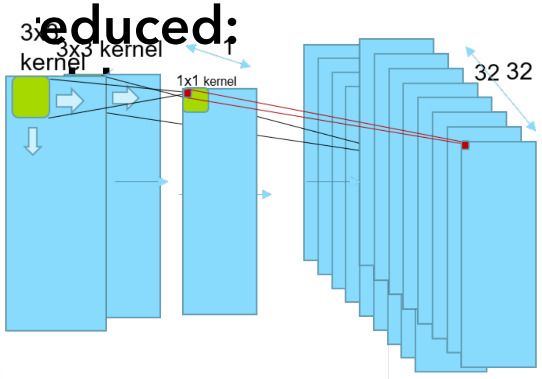


*Furnon et al., LORIA University

NN optimization

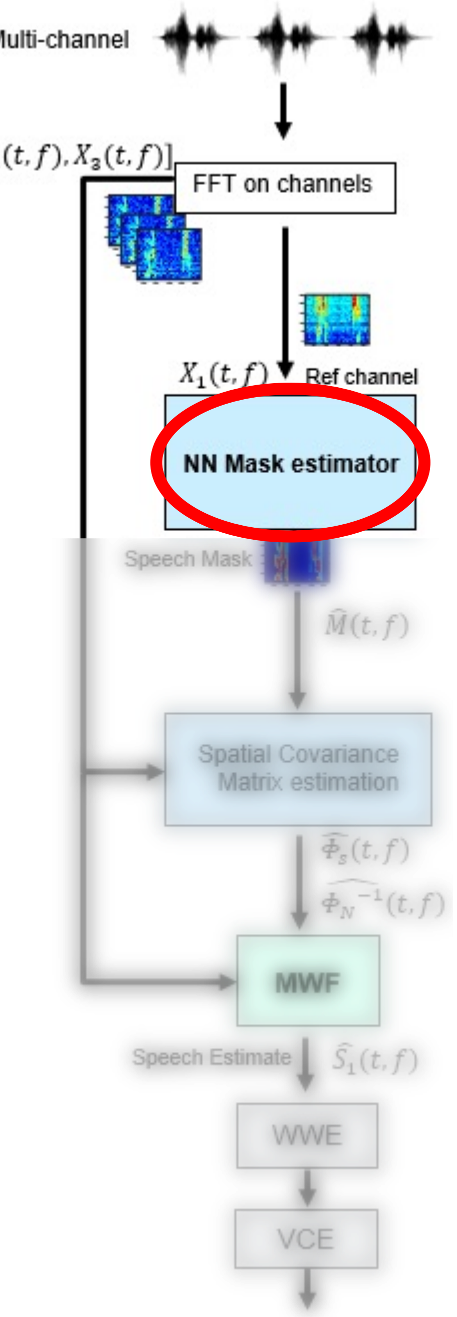


- Input is [257,21]
- Kernel is (3,3)
- Number output filters is 32.



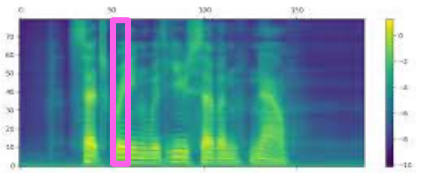
- CNN: 3x3 x 257x21 x 32 ≈ 1.5M operations

(10 times less!)



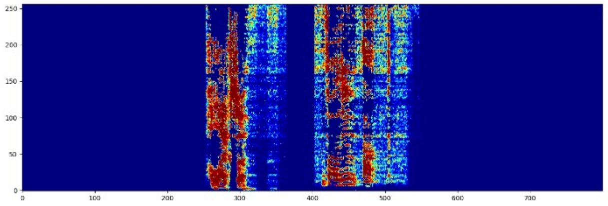
*Furnon et al., LORIA University

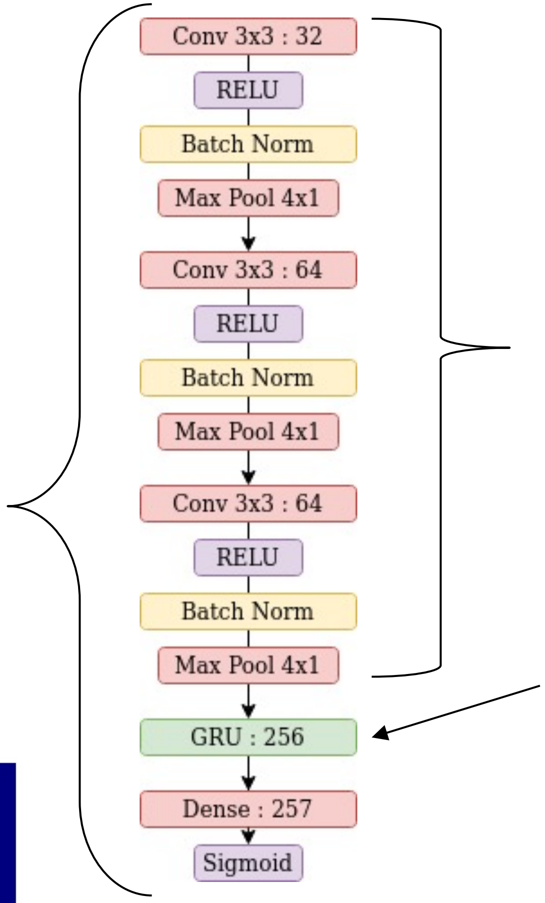
NN optimization



CRNN*

Convolutional Recurrent Neural Network





Conv 3x3 : 32

RELU

Batch Norm

Max Pool 4x1

Conv 3x3 : 64

RELU

Batch Norm

Max Pool 4x1

Conv 3x3 : 64

RELU

Batch Norm

Max Pool 4x1

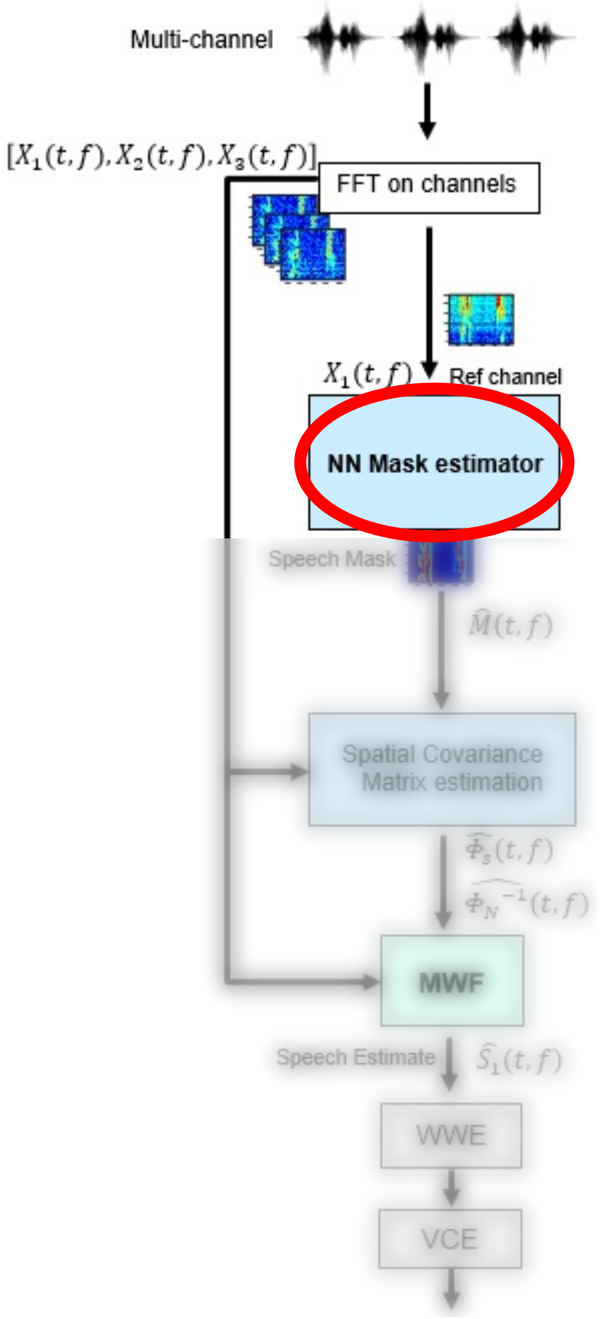
GRU : 256

Dense : 257

Sigmoid

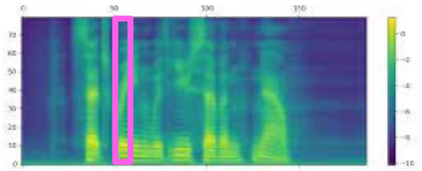
CNN part can be reduced:
-Depthwise Separable convolution

Gated Recurrent Unit (GRU) features can be reduced

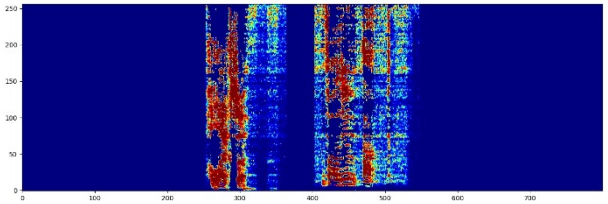


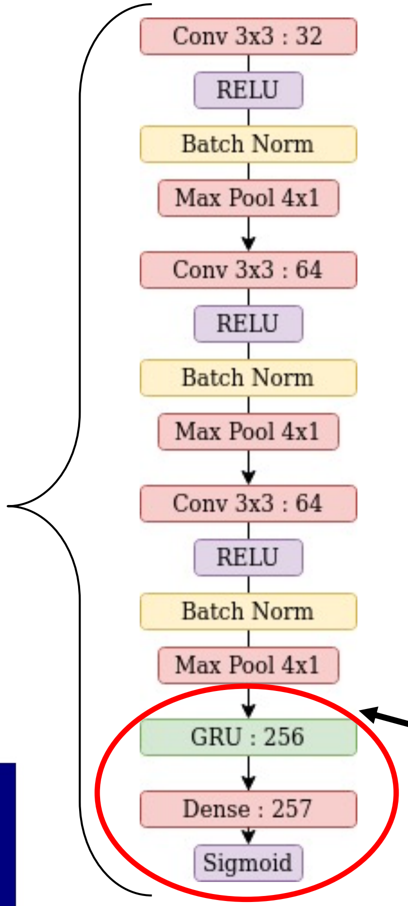
*Furnon et al., LORIA University

NN optimization

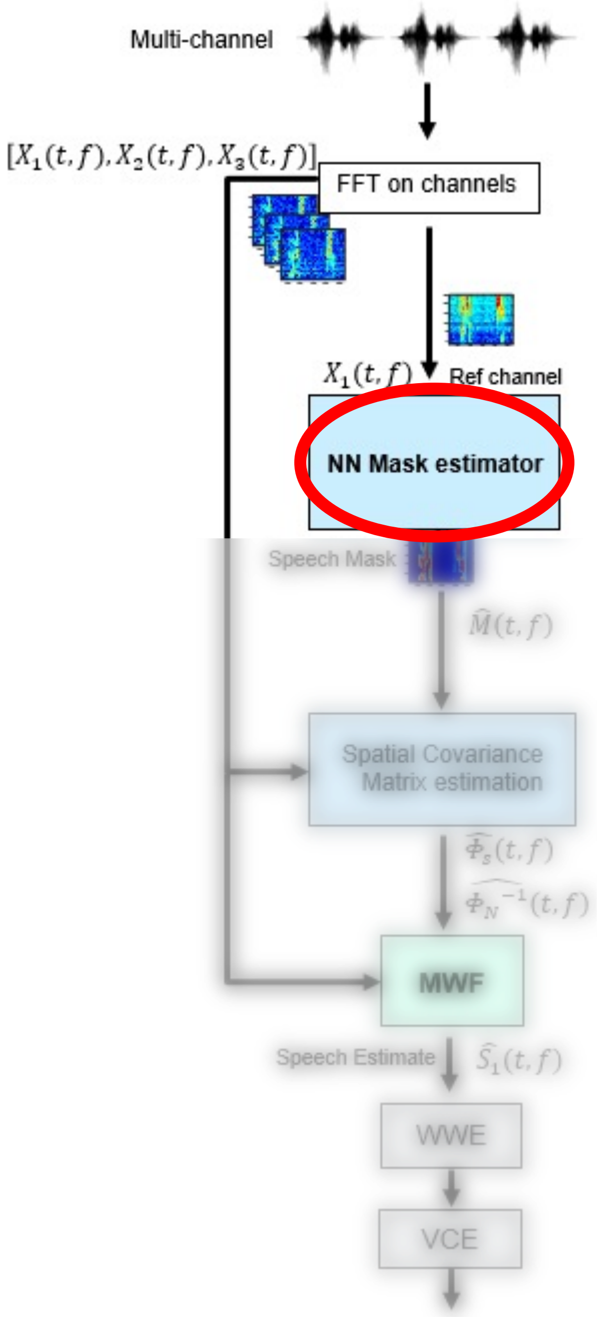


CRNN*
Convolutional Recurrent Neural Network



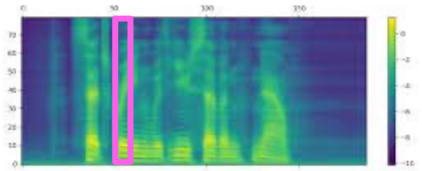


Directly give an embedded feature as input: mel-spectrogram and only keep recurrent layer

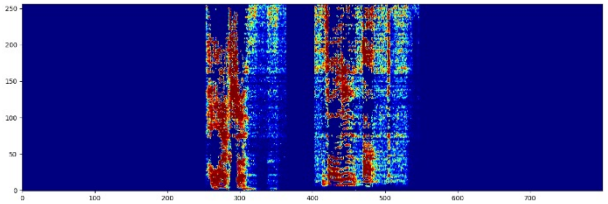


*Furnon et al., LORIA University

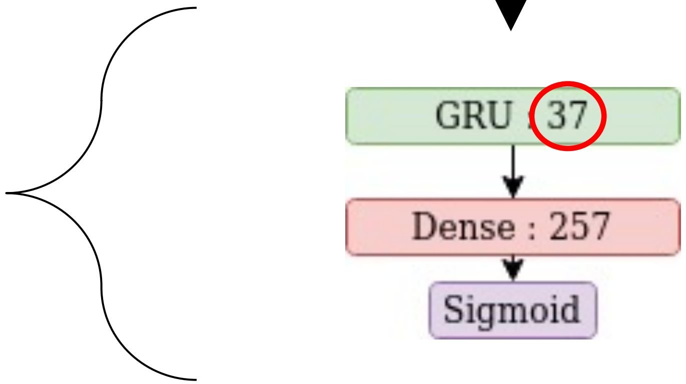
NN optimization



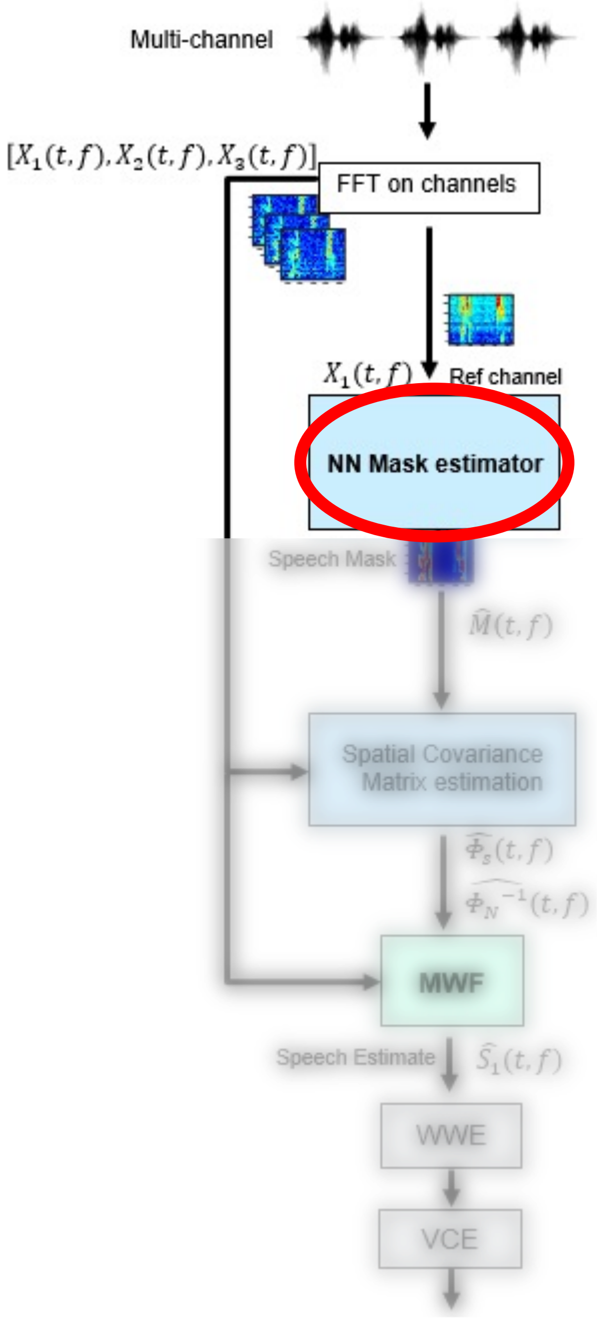
RNN_model
Recurrent Neural Network



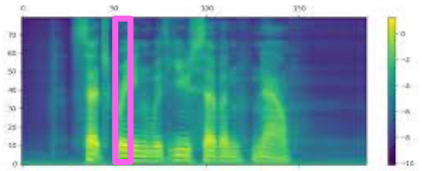
21 consecutive frames
x
40 normalized mel bins



Network architecture was optimized: hyperparameter tuning with random search (Raytune)

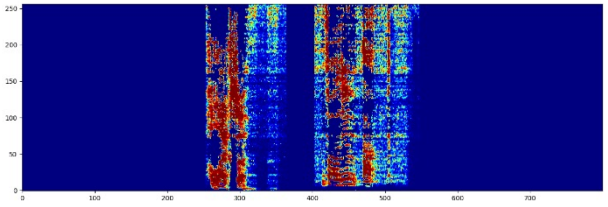


NN optimization



RNN_model

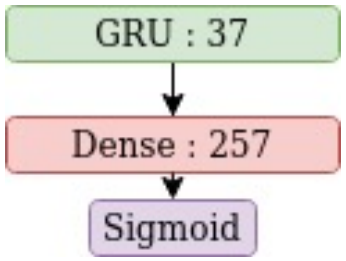
Recurrent Neural Network



21 consecutive frames

x

40 normalized mel bins

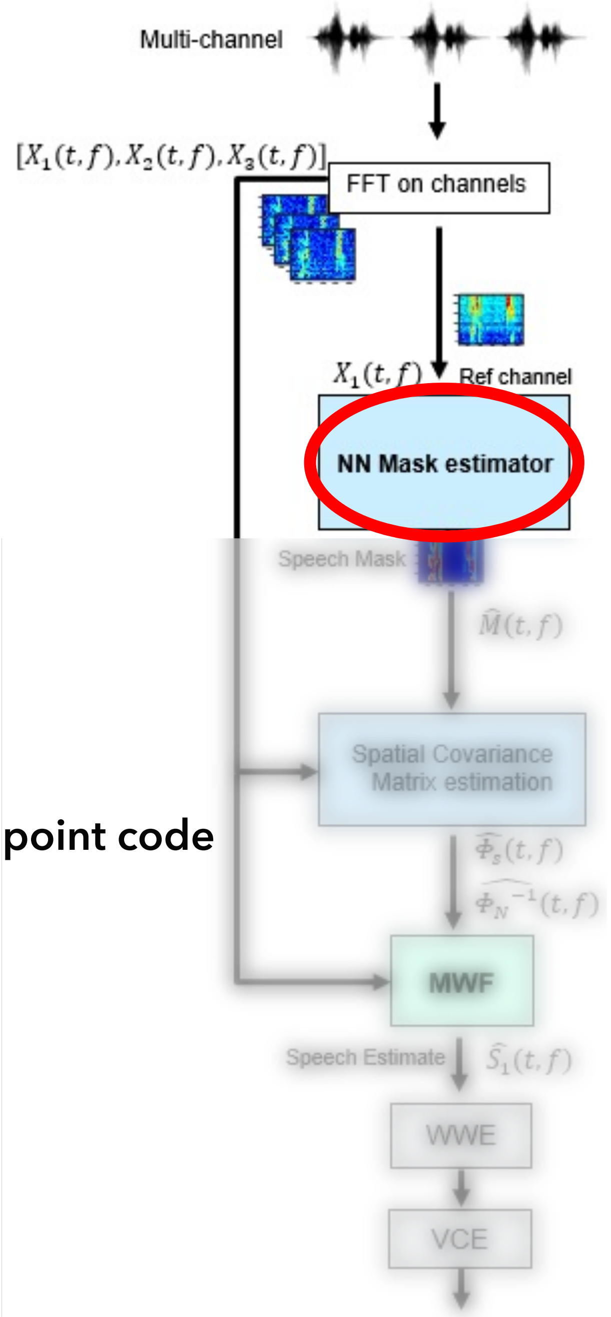


Number of Parameters: 18k

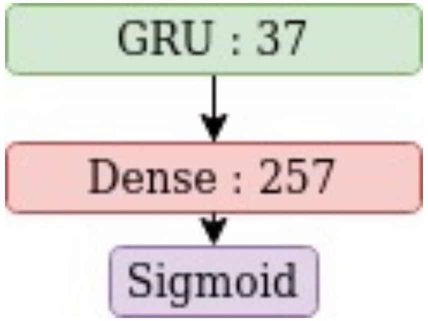
Number of MACs: 200k

300MHz C floating point code

Network architecture was optimized: hyperparameter tuning with random search (Raytune)



NN optimization



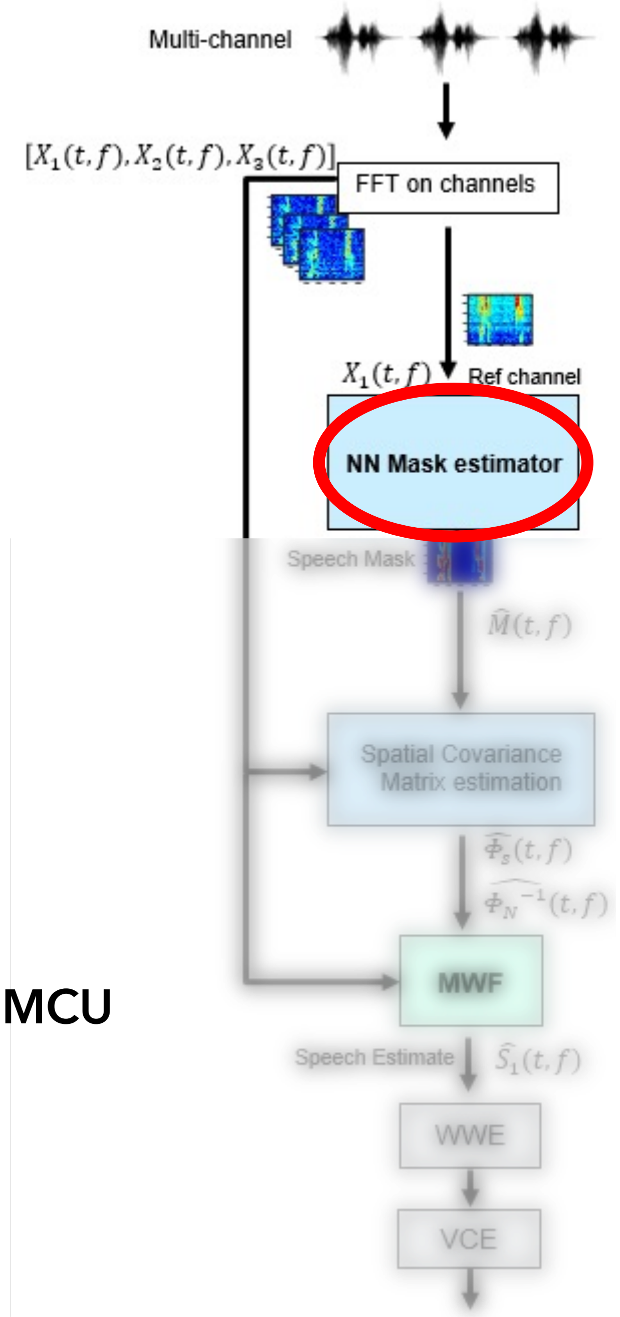
- 16-bit symmetric post-training quantization using GLOW

From Float: 300MHz
To Quantized 16 bits: 150MHz

- Using Truncated Backpropagation Through Time (TBPTT): Frame-by-frame decisions
- (We used to process 21 frames to compute 1 output)

28.2MHz float on the Arm Cortex-M7 (NXP-RT1060) MCU

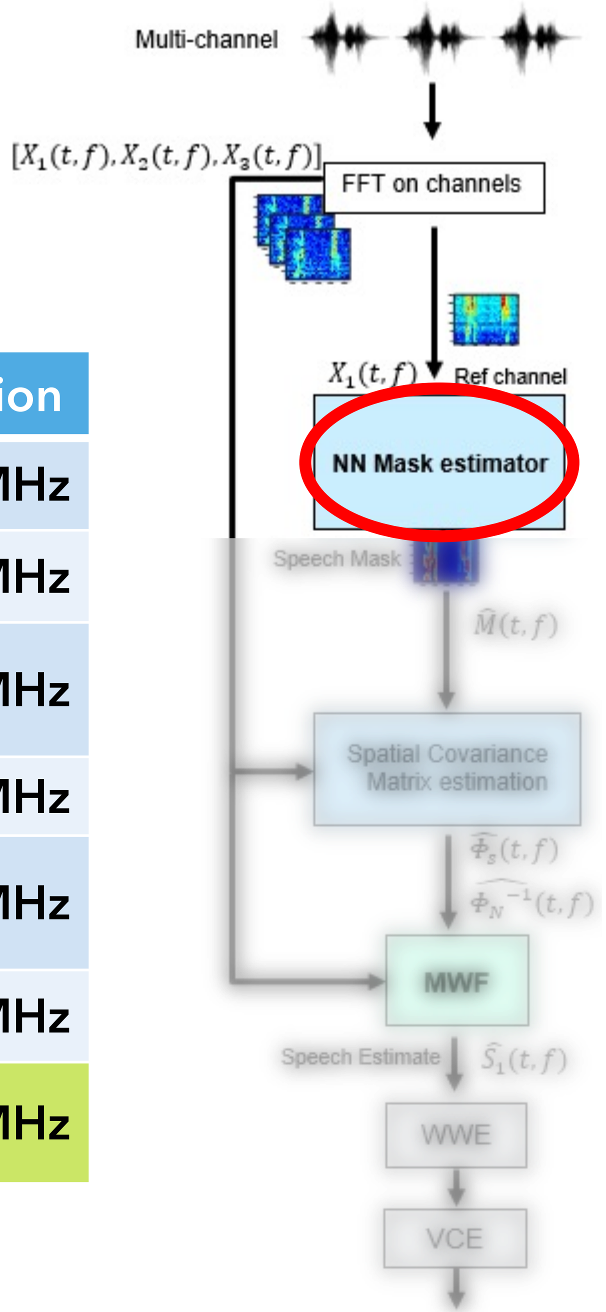
12MHz for the16-bits quantized version



NN optimization

NN summary

Model	Input	Parameters	MACs	Consumption
CRNN	FFT [21, 257]	470k	33M	63240 MHz
CRNN Light	FFT [21, 257]	43k	2.5M	1800 MHz
Depth-CRNN Light	FFT [21, 257]	36k	800k	840 MHz
RNN	Mel [21, 40]	18k	200k	300 MHz
$\text{RNN}_{\text{quant}}$	Mel [21, 40]	18k	200k	150 MHz
TBPTT-RNN	Mel [1, 40]	18k	18k	28.2 MHz
$\text{TBPTT-RNN}_{\text{quant}}$	Mel [1, 40]	18k	18k	12 MHz



Multichannel Wiener Filter optimization

$\widehat{\Phi_S}(t,f)$
$\widehat{\Phi_N{}^{-1}}(t,f)$

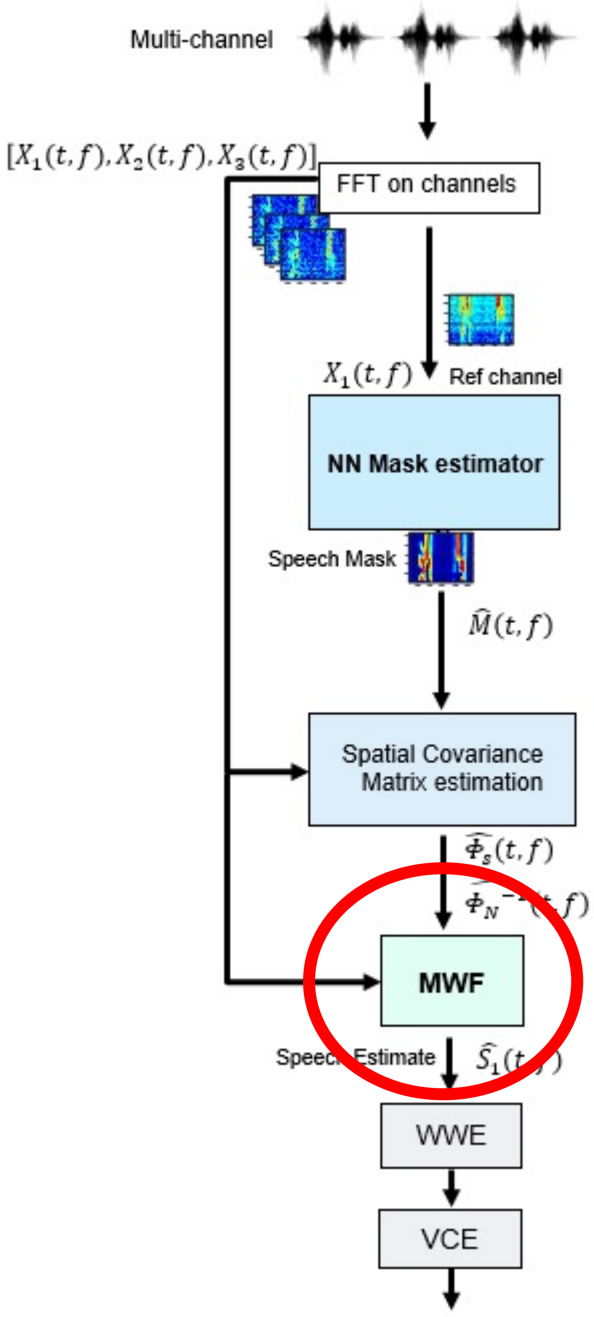
Used to be computed not in real-time

- We now recursively estimate covariance matrices of noise to solve the Multi Channel Wiener equation:

$$W_{mwf}(t,f) = (\phi_S(t,f) + \phi_N(t,f))^{-1}\phi_S(t,f)\, e_1$$

$$MMSE\!: W = arg \min_{W} \boldsymbol{E}[|S_1(t,f) - W^H X(t,f)|^2]$$

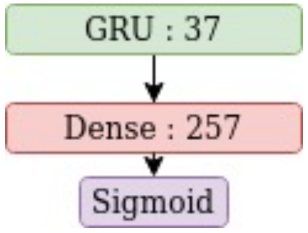
Target speaker



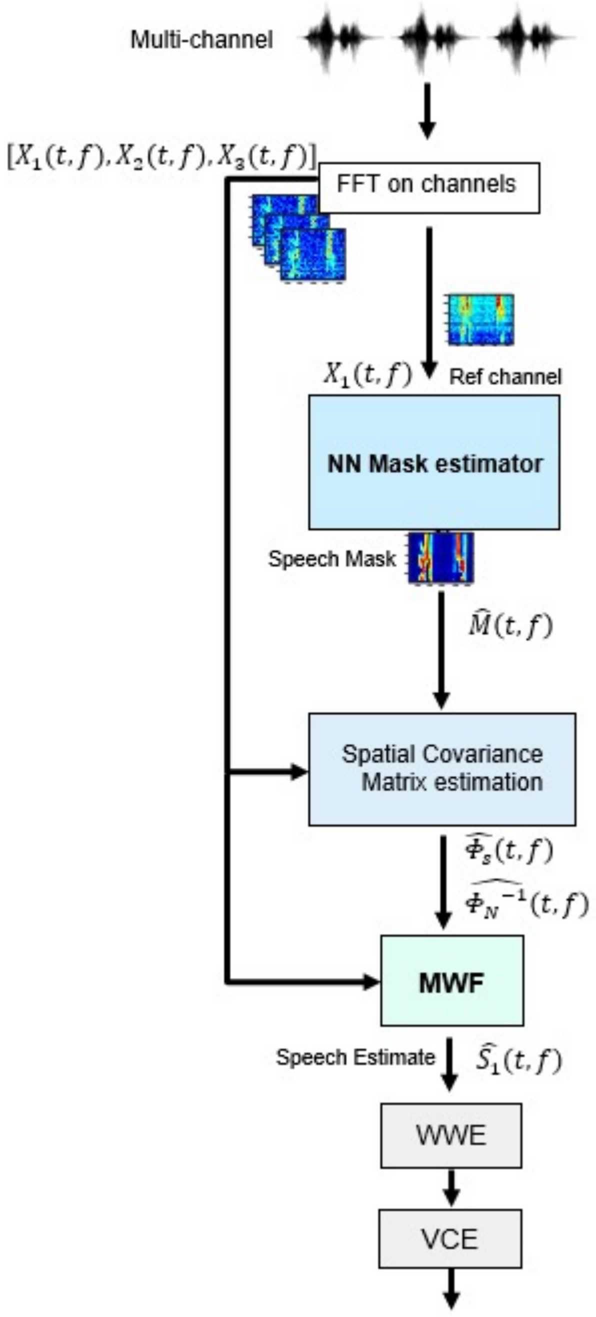
Embedded solution

CHOSEN SOLUTION

-18k parameter NN quantized in 16 bits, taking only 12MHz to predict a mask-frame



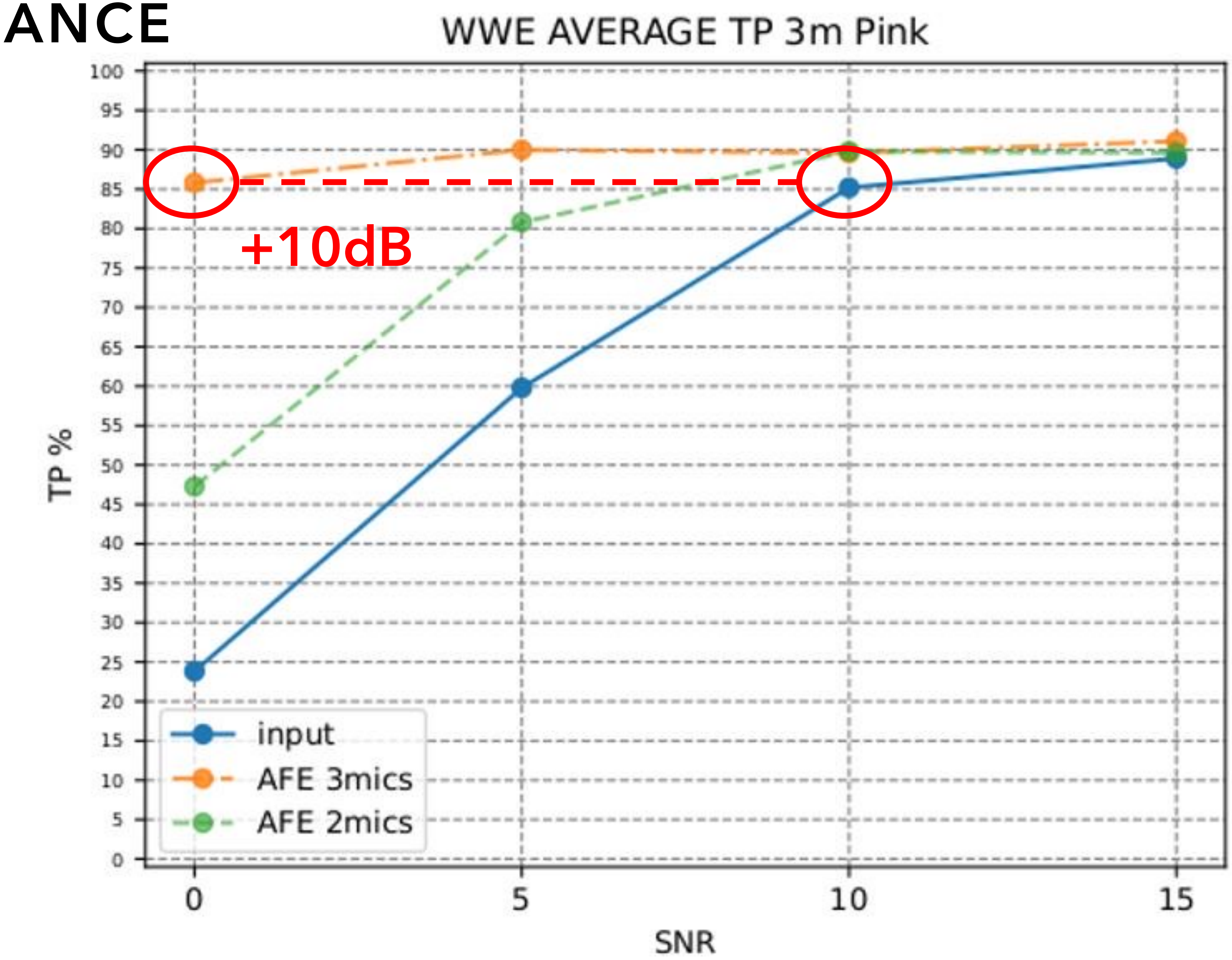
-Full Speech enhancement solution is taking 160MHz in the 3-mics configuration and about 105MHz in the 2-mics configuration



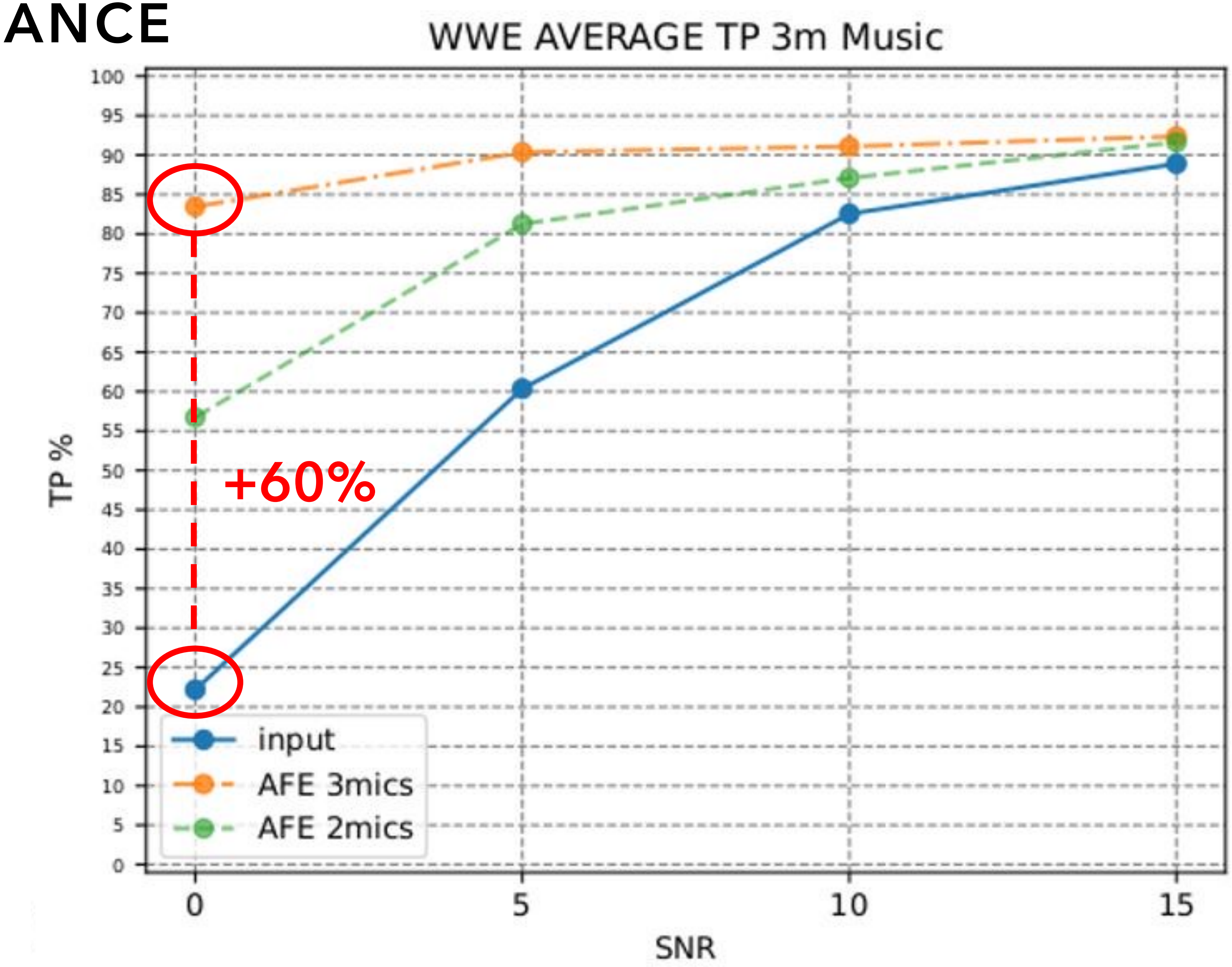
PERFORMANCE FROM AMAZON FAR-FIELD TEST

- Test file is composed of 50 pairs of Wake Word + Voice commands
- The speaker is at 3m distance from the device
- We test in different noise configurations: Silence, Pink, Music, Multi-Talker
- Signal-to-Noise ratio is taken between 0dB (same level speech and noise) and 15dB (power of speech is about 4.5x noise level)
- We measure True Positive Wake Word Hit rate: Well detected keywords at the right time

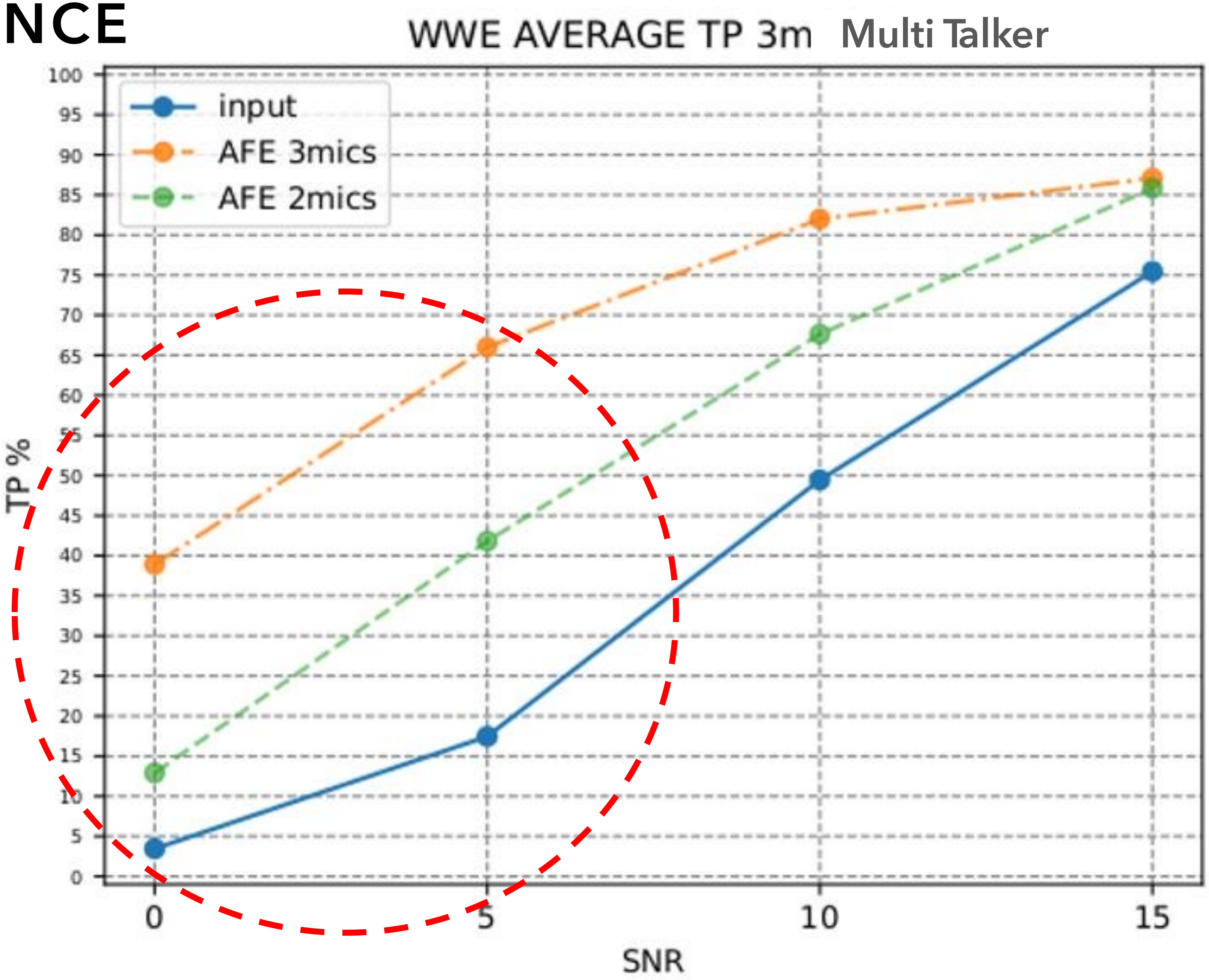
PERFORMANCE



PERFORMANCE

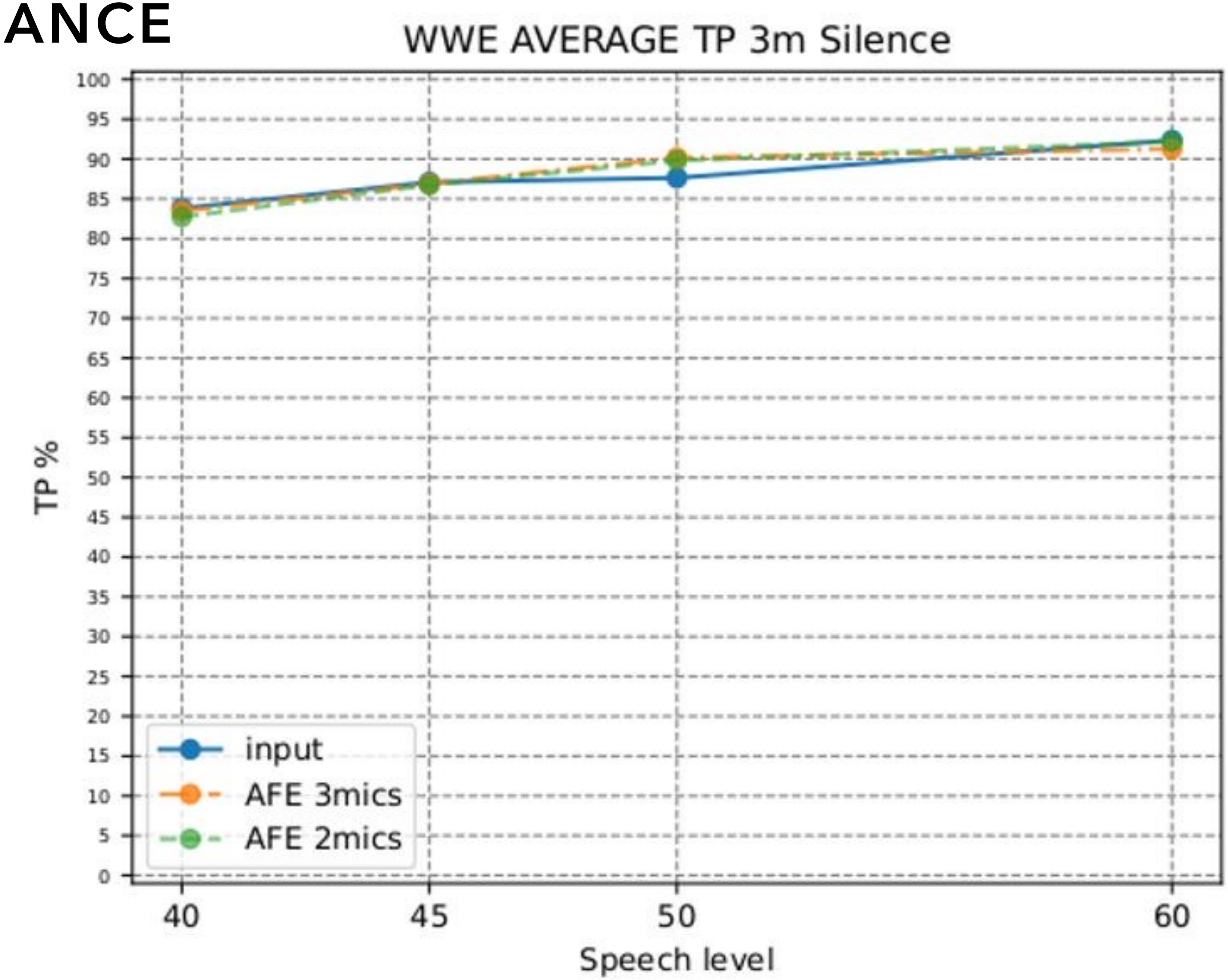


PERFORMANCE



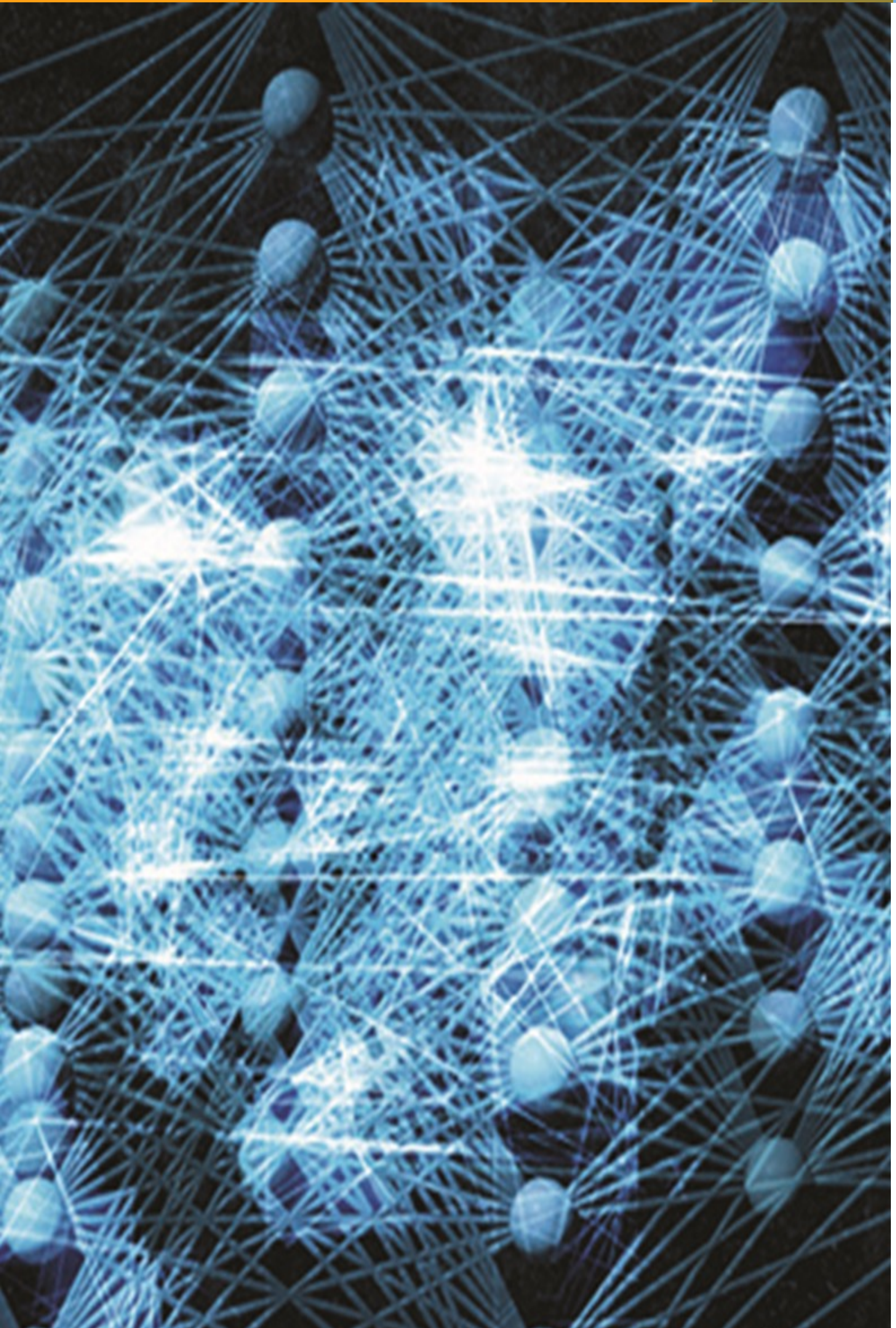
Very difficult to know who is the target speaker

PERFORMANCE



CONCLUSION

- Introduced a **speech enhancement solution for low power devices**
- The solution is real-time and embedded on a small platform
- Improved by 40% the Wake word and Voice Commands hit rate in a three microphone (3-mic) configuration



ANY QUESTIONS ?

References and helpful links

- eIQ® ML Software Development Environment (https://www.nxp.com/eiq)
- NXP's voice intelligent technology (VIT) library (https://www.nxp.com/vit)
- eIQ ML/AI Training Series (https://www.nxp.com/mltraining)
- MCUXpresso Software and Tools (https://www.nxp.com/mcuxpresso)

SECURE CONNECTIONS
FOR A SMARTER WORLD

TINY ML SUMMIT

tinyML Summit 2022 Sponsors

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

AONdevIces

arm

ASPINITY

brainchip
The Neuromorphic Computing Company

CEVA®

Deeplite

EDGE IMPULSE

emza
visual sense

FotaHub

GREENWAVES
TECHNOLOGIES

Grovety Inc.

Himax

HOTG

imagimob

infineon

itemis

KLIKA·TECH
GLOBAL IOT SOLUTIONS

LatentAI

LATTICE
SEMICONDUCTOR

Micro.ai

OmniML

NXP

PXI

Plumerai

PROPHESEE

Qeexo

Qualcomm

Rackner

RealityAI®
Engineering Solutions for the Edge

REEXEN
technology

RENESAS

SAP

seeed studio
The IoT Hardware Enabler

SensiML™

Sony
Semiconductor
Solutions
Corporation

ST
life.augmented

SA STREAM ANALYZE

synaptics®

SynSense

SYNTIANT

Tensil.ai

TensorFlow

XMOS

TINY ML SUMMIT

Copyright Notice

www.tinyml.org