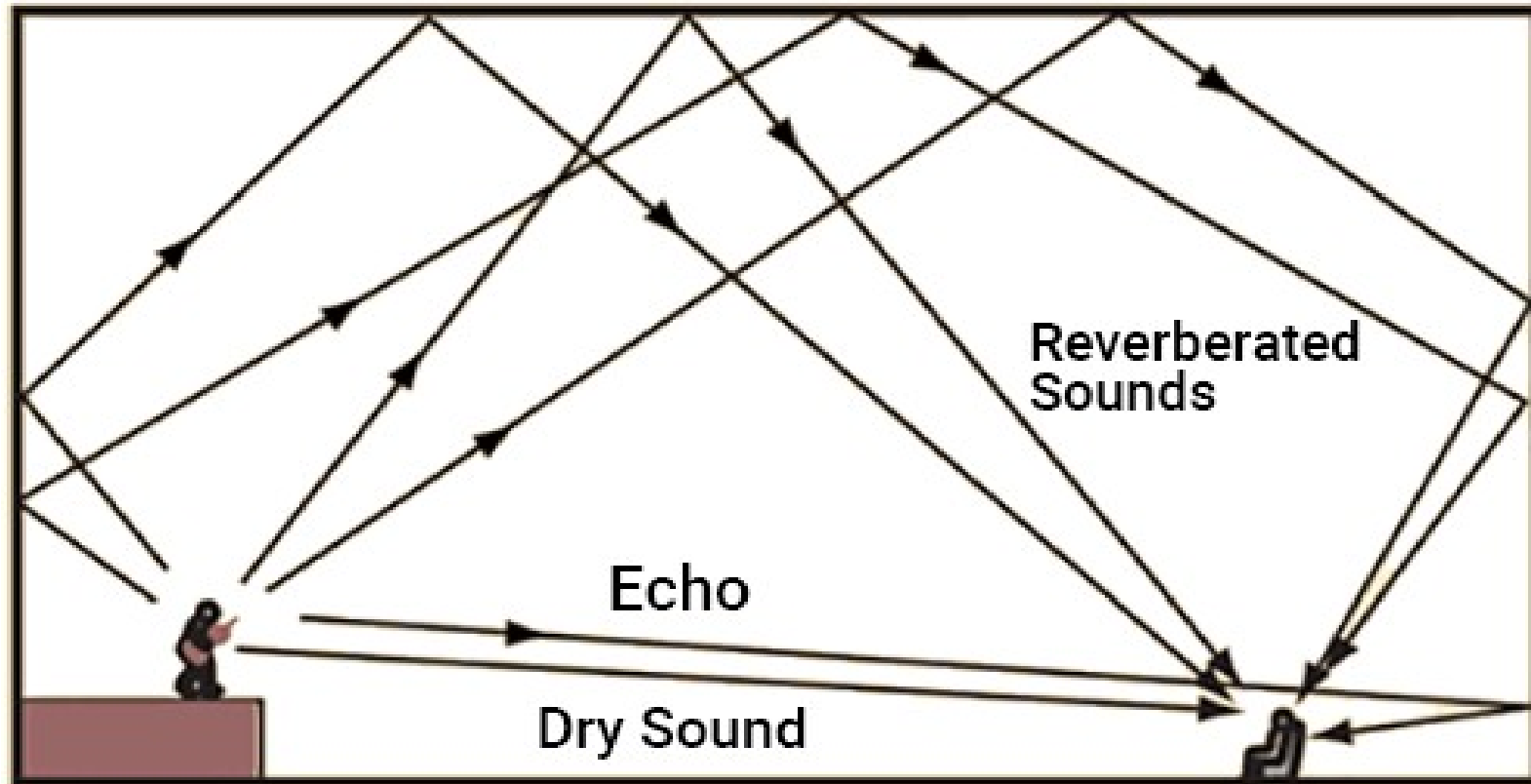


Remove reverb from sound



Reverb = echo from walls



- source: <https://www.softdb.com/what-is-reverberation-in-acoustical-analysis/>

Use cases

1) professional music production

- real time
- high quality of sound required

2) home recordings

- post production of recordings for e.g. youtubers
- not real time

3) speech transfer

- zoom, phone calls
- real time
- lower quality is sufficient

Generate sound with reverb

- anechoic sound + convolution with IR = sound with reverb

$$\int_0^t f(t-y)g(y)dy = h(t)$$

- IR: Impulse Response characterizes the room acoustics
- can be computed with FFT
- IR depends on:
 - room size and shape
 - position of source and receiver
 - shape and size of objects in the room

Strategies for dereverberation

- **IR is known:**
 - deconvolve signal with IR
- **IR is unknown:**
 - known as blind dereverberation/deconvolution
 - 1) classify IR with neural network and deconvolve with a known IR
 - 2) predict dereverberated signal directly with autoencoder

Deconvolution

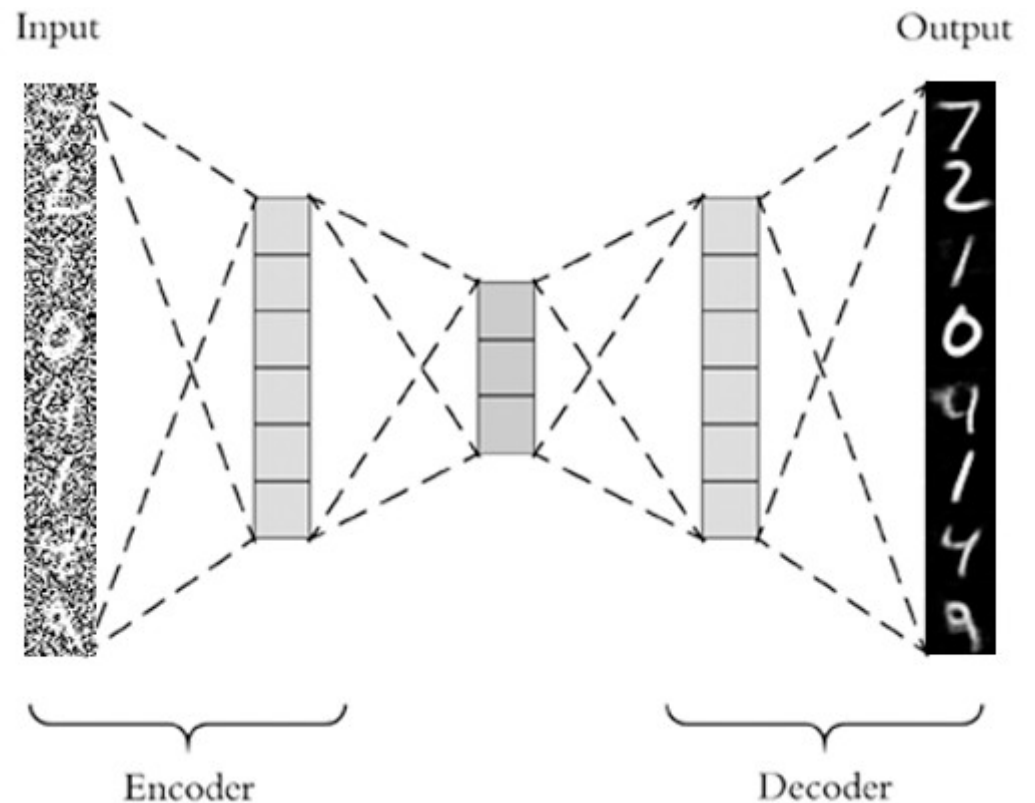
- deconvolution requires a **known** IR
- can be done by FFT
- numerically unstable: strong amplification of even small noise
- better, but not perfect: Wiener deconvolution
- many problems, e.g.
 - deconvolved sound with length of 10s
 - same sound, same deconvolution, but length of 11s

Blind dereverberation by classification

- train
 - 1)sufficiently sample space of relevant IRs
 - 2)generate sound with reverb
 - 3)train a neural network for IR classification
- deploy
 - 1)classify IR with a neural network
 - 2)deconvolve with closest IR or a linear combination of IRs
- classification of 20 different IRs works well with a network similar as for speech MNIST

Autoencoder (AE)

- WaveNet
- Spectrograms
- Autoregressive
- Combinations

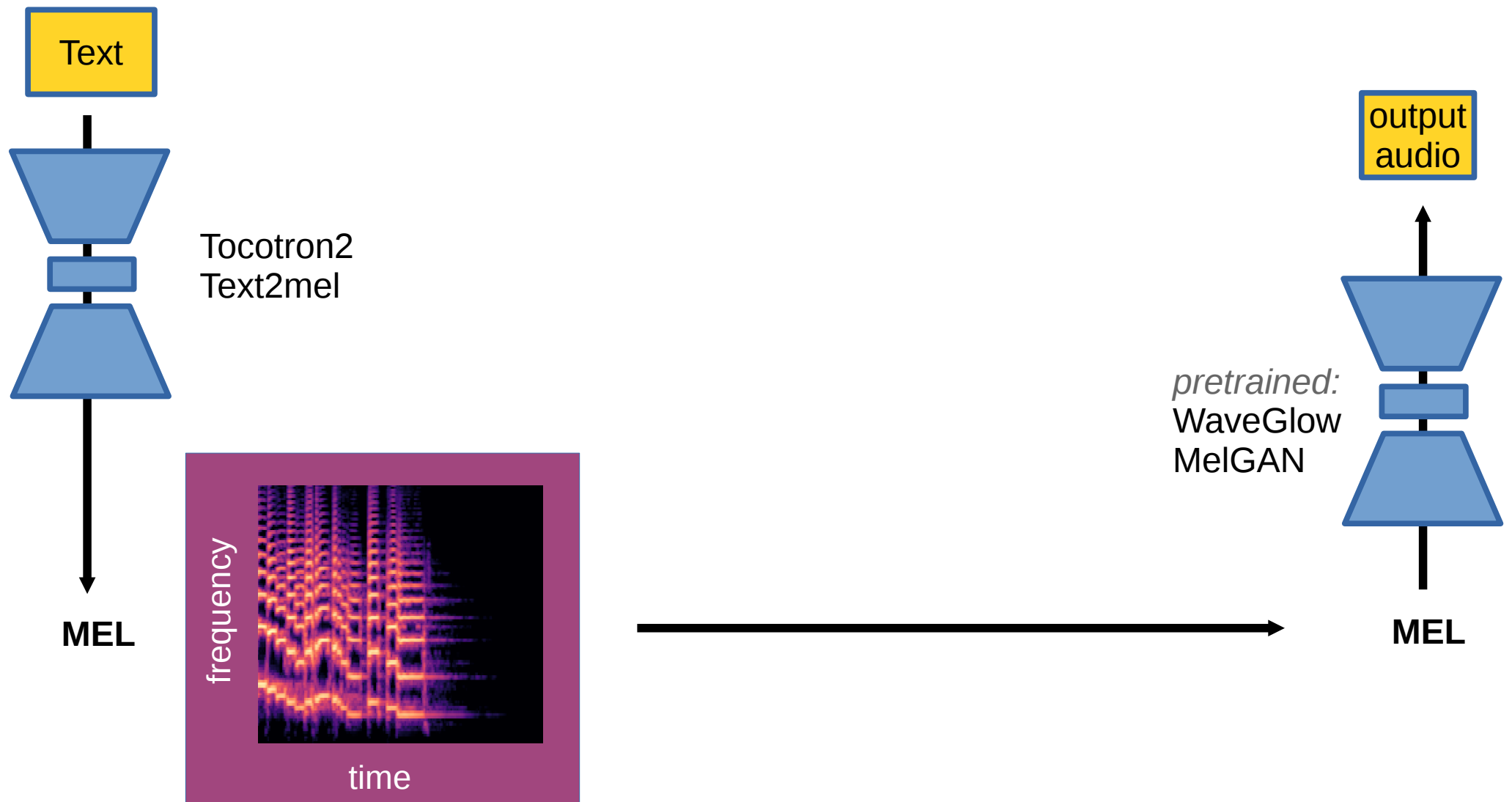


source:
pyimagesearch.com/wp-content/uploads/2020/02/keras_denoising_autoencoder_header.png

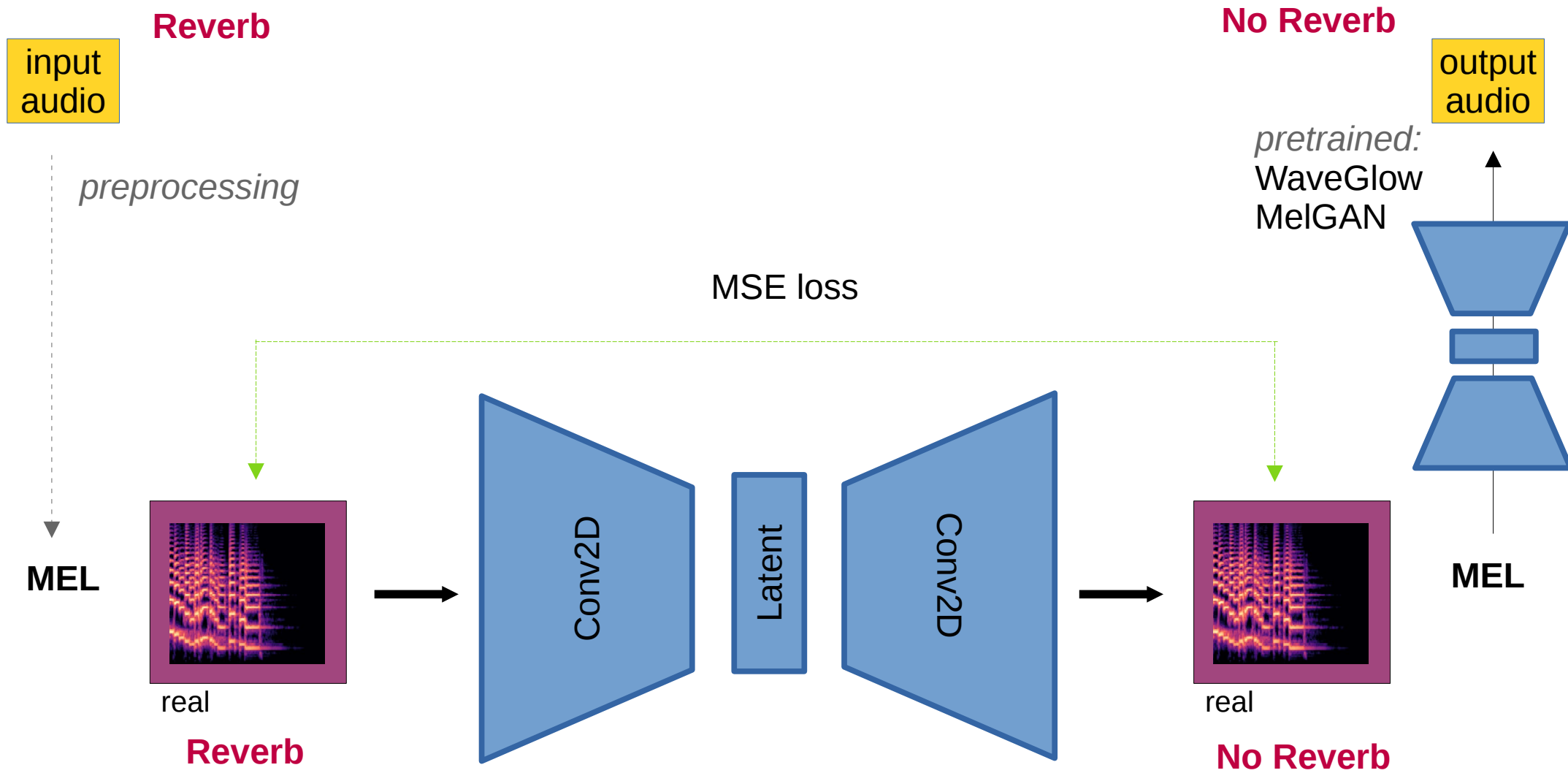
Possible strategies

- Work with MEL spectrograms
 - AE and WaveGlow (too slow for real time execution)
 - AE and WaveGAN (rather slow but possible)
- Work with audio signal
 - LSTM (bad quality)
 - Use WaveNet AE (slow convergence, no results, bad for reverb)
 - Train WaveGAN to omit reverb in translation (transfer learning)
- Work with STFT and complex numbers
 - Unknown territory!

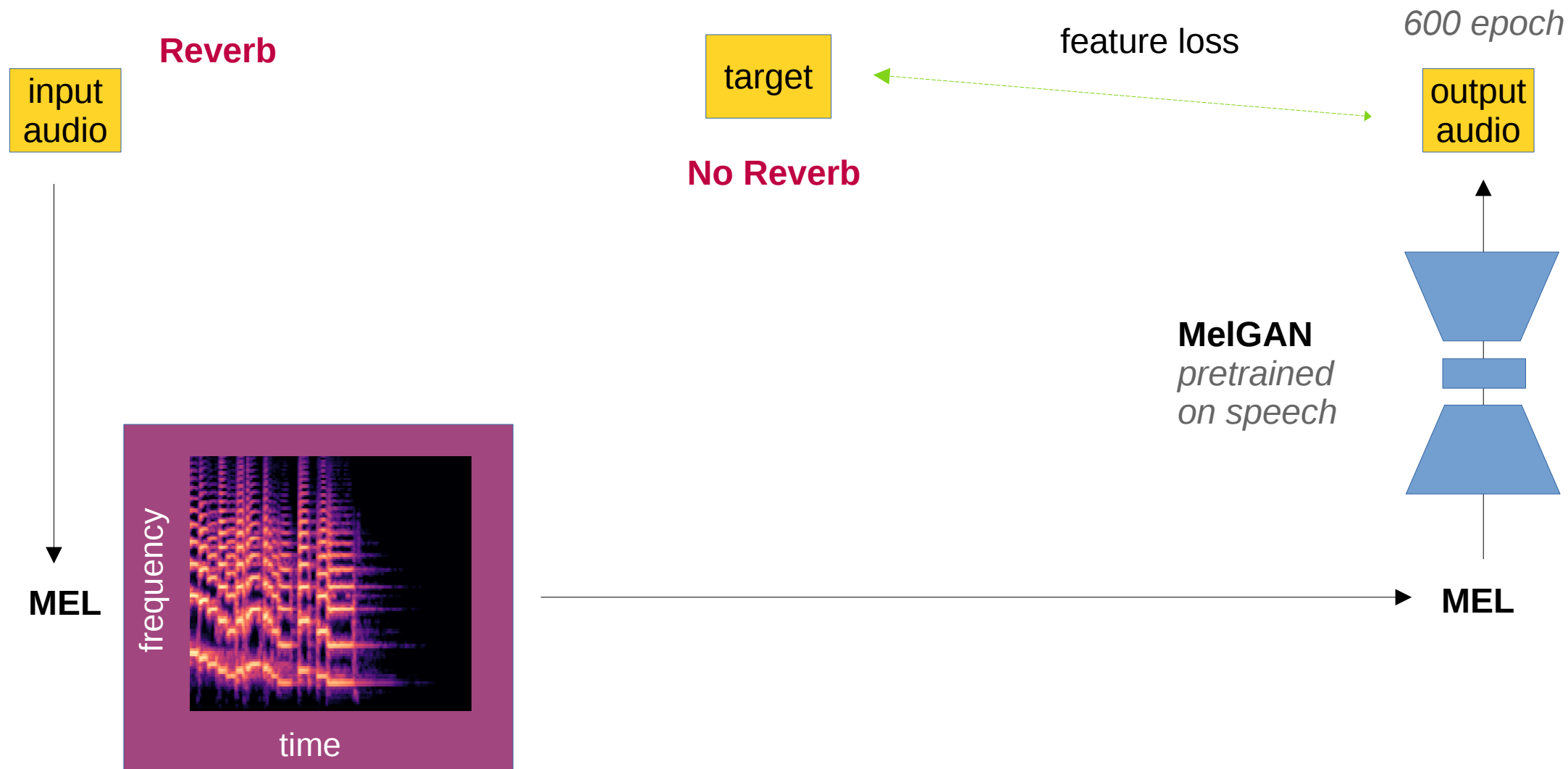
SOTA Methods (Text-to-Speech)



SOTA Methods *Slow!*



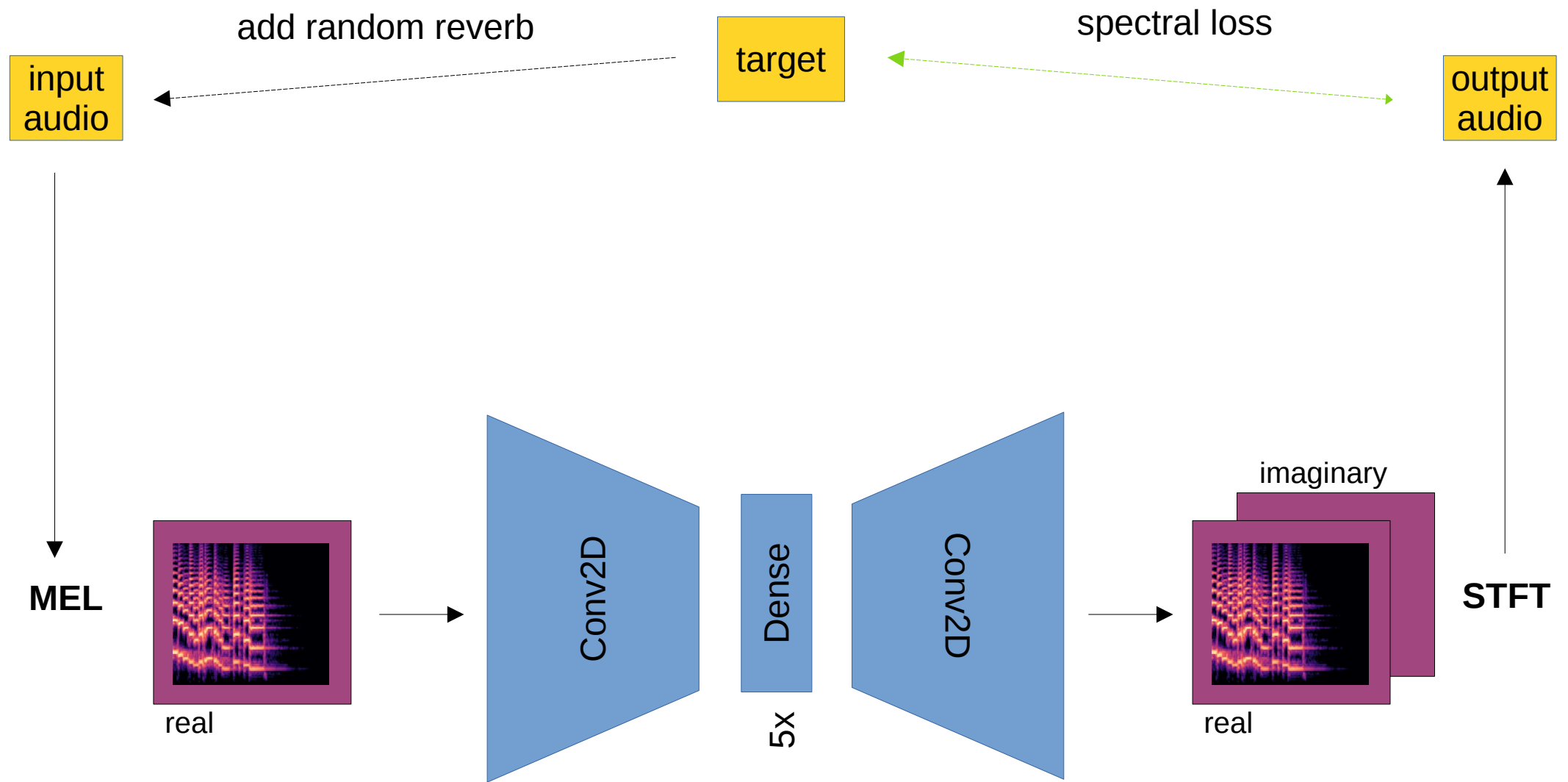
MelGAN - transfer learning



Unseen speaker

MelGAN trained on speech

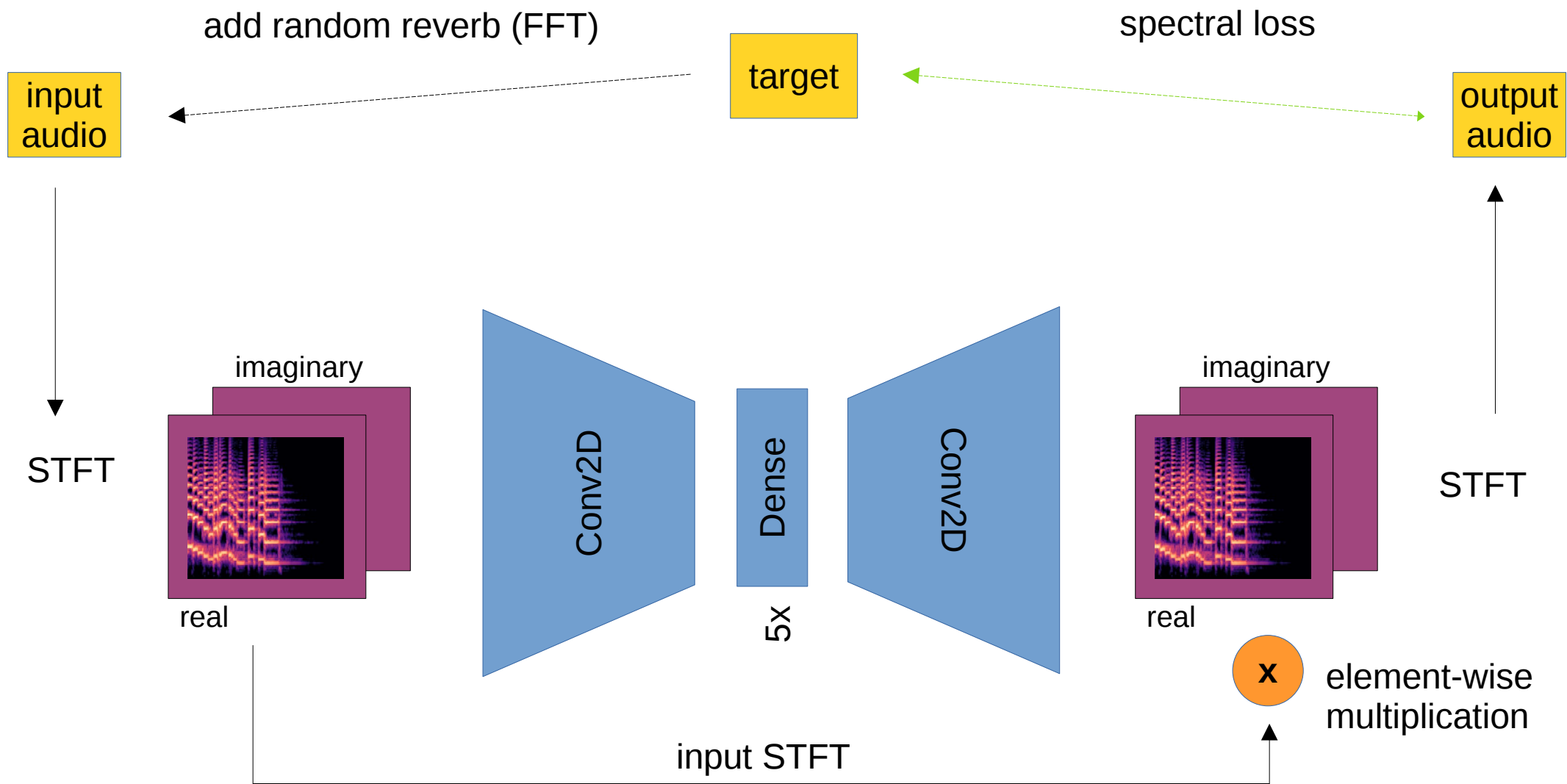
MEL-to-STFT (AE)



Unseen speaker

Mel-to-STFT test

STFT (AE)



real data

test

add room
reverb

or a big
reverb

Conclusions

- High sound quality requires sufficient sampling of IRs
 - requires a lot of resources
- NNs can work with complex numbers (STFT) when the right architecture is used
- combination of architectures and postprocessing with e.g. noise filters might yield better results