Experiments

Training data distortion simulation

Algorithm 1: Add high quality speech x with random distortions

In: Speech $x \leftarrow S(\mathcal{X})$; Noise $n \leftarrow S(\mathcal{N})$; Room impulse response $r \leftarrow S(\mathcal{R})$

$x' = x$

with p_1 probability:

  $x' = x * r$; /* Convolute with RIR filter */

with p_2 probability:

  $\theta = S(\mathcal{U}(\Theta_{low}, \Theta_{high}))$; /* Choose clipping ratio */

  $x' = max(min(x', \theta), -\theta)$; /* Hard clipping */

with p_3 probability:

  $t = randomFilterType()$;

  $c = S(\mathcal{U}(C_{low}, C_{high}))$; /* Random cutoff frequency */

  $o = S(\mathcal{U}(O_{low}, O_{high}))$; /* Random filter order */

  $x' = x' * getFilter(t, c, o)$; /* Low pass filtering */

  $x' = Resample(Resample(x', 44100, c * 2), c * 2, 44100)$; /* Resample */

  with p_4 probability:

    $n = n * getFilter(t, c, o)$; /* Low pass filtering on noise */

    $n = Resample(Resample(n, c * 2), 44100)$; /* Resample */

with p_5 probability:

  $s_1 = S(\mathcal{U}(S_{1low}, S_{1high}))$; /* Random SNR */

  $s_2 = S(\mathcal{U}(S_{2low}, S_{2high}))$; /* Random Scale */

  $n = \frac{n}{mean(abs(n))/mean(abs(x'))}$; /* Normalize the energy of noise */

  $x' = (x' + \frac{n}{10^{snr/20}})$; /* Add noise */

$x' = x' * s_2$; /* Scaling */

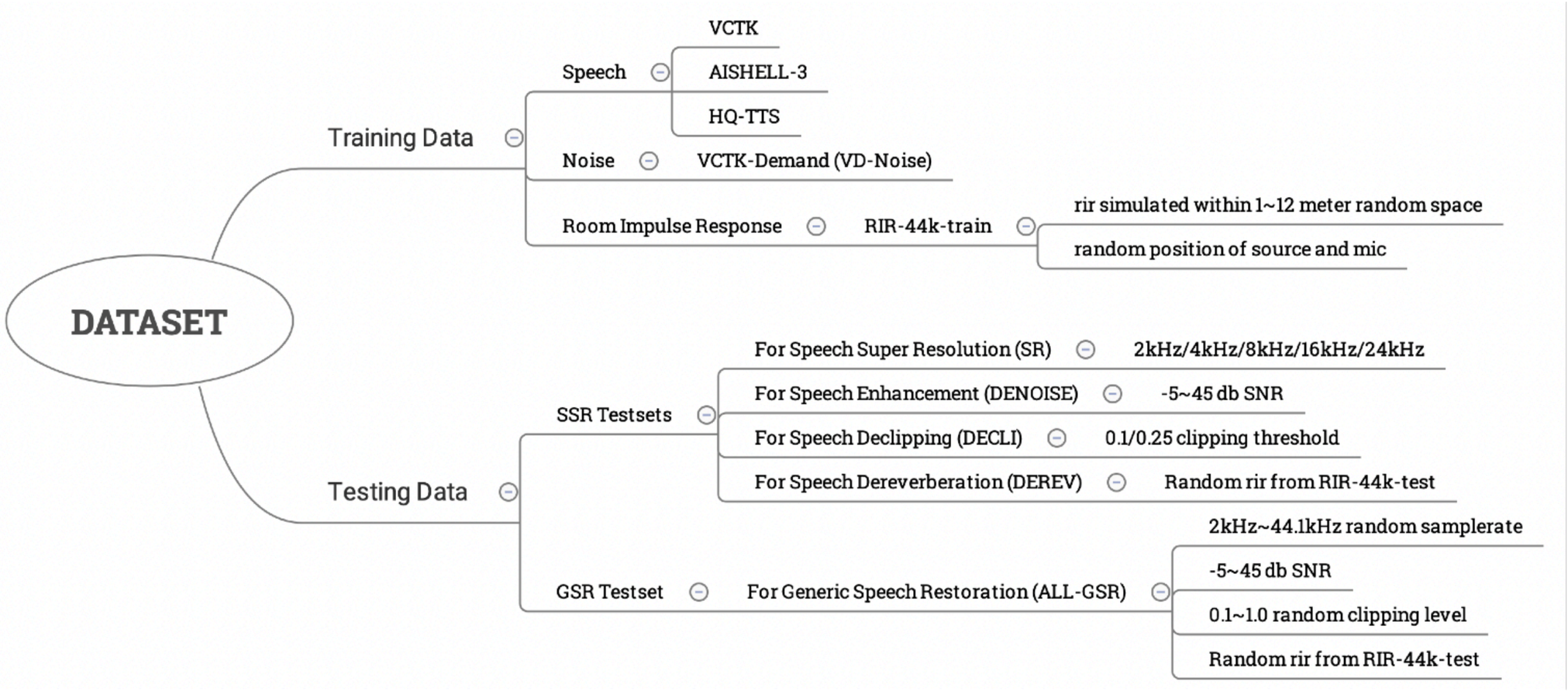
$x = x * s_2$; /* Scaling */

Out: The randomly distorted speech x' and its target x

Distortion simulation algorithm we used for general speech restoration.

Experiments

Dataset



The dataset we use in this paper.