

NOPE-HYPE: A Structured Simulation Workflow for Robust Speech-to-Text Across Diverse Acoustic Environments

Niramay M. Patel
IISER Bhopal, India
niramay23@iiserb.ac.in

Bibek Behera
IIT Bombay, India
bibek.iitkgp@gmail.com

Raksha Sharma
IIT Roorkee, India
raksha.sharma@cs.iitr.ac.in

Abstract—Robust speech-to-text translation systems should perform reliably across diverse acoustic conditions, yet practical pipelines lack controllable tools for systematic environment exploration. Large speech models remain sensitive to unseen acoustic conditions, as training data rarely cover the full range of real environments. We present NOPE-HYPE, a structured training workflow that combines a controllable environment simulator, coverage-optimal environment reduction on Power Spectral Density (PSD) templates, and a small, interpretable hyperparameter search over simulator knobs. We show that simulator-generated noise achieves performance comparable to balanced real-noise training across Whisper and SeamlessM4T models, provide principled environment prototype sets, and identify practical default simulator configurations from a structured 27-run hyperparameter sweep.

Index Terms—speech recognition, noise augmentation, acoustic robustness, environment simulation, hyperparameter search

I. INTRODUCTION

Speech-to-text (S2TT) translation systems often operate outside curated studio audio settings. The audio of the deployment often includes transportation noise, indoor hums, transient events, and speech-like interference from nearby people. Even strong speech foundation models can degrade under such shifts because training data and augmentation choices rarely cover the full space of real acoustic variability in a controlled way. As shown in Table I, models trained only on clean speech perform worse than those trained with noise injection.

A common mitigation is *noise augmentation*. In practice, this collapses into a single axis, such as Gaussian noise, a single colored-noise type, or a narrow SNR range [1], [2]. These choices are convenient, but they do not isolate factors that matter in real environments — the spectral shape differs by scene, energy changes over time, transient activity varies, and human activity adds interference that overlaps with speech. Learned audio generators can increase realism, but their latent factors are often hard to interpret and sweep independently, limiting their usefulness for a targeted robustness search [7]–[9].

In this paper, we shift the focus from adding arbitrary noise to systematically modeling environment-specific noise to build an S2TT system that performs well under both clean and noisy conditions. We propose NOPE-HYPE, a practical workflow designed to improve systematic robustness through *interpretable control* and *structured search*. We test our framework on two language pairs: English to Hindi and English to German. The workflow has three parts:

- **SIMULATOR (SIM)**: A parameterized simulator for environmental noise and noise-in-speech mixtures. Rather than aiming for exactly realistic noise, it enables structured modeling of acoustic factors relevant to speech-to-text performance.
- **Environment Reduction**: When many environmental categories exist, uniform sampling becomes computationally expensive because each environment requires a full hyperparameter sweep of the simulator. We summarize each environment using a PSD template and select a small subset of prototypes that covers the full set under explicit distances.
- **Hyperparameter Search**: We perform a sequential and interpretable search over simulator parameters that yield a robust default configuration and clarify which acoustic factors strongly influence model performance.

The paper is organized as follows. Section II reviews robust speech recognition and augmentation. Section III summarizes the NOPE-HYPE workflow, including the controllable noise simulator (SIM) and environment reduction. Section VI describes experiments evaluating Whisper and SeamlessM4T, including ablations and a 27-run hyperparameter sweep. Sections X and XI discuss the results and summarize the main findings.

II. RELATED WORK

Robust automatic speech recognition (ASR) has often used acoustic augmentation to improve performance in noisy conditions, including additive noise [1], [2], reverberation [3], speed and pitch perturbation [1], and time-frequency masking [4]. SpecAugment [4] established

structured masking as a strong baseline, with later extensions to mixed-sample and hidden-space augmentation [5], [6]. However, these methods typically rely on coarse heuristics and do not allow systematic exploration of environment structure.

Synthetic data pipelines generate diverse acoustic conditions via simulated noise and reverberation, using either simple RIR/noise addition [1], [3] or generative models for realistic noise synthesis [10], [11]. While effective for robustness, they often trade simplicity for controllability, making structured ablation difficult [12].

Large-scale S2TT systems such as Whisper [13] and SeamlessM4T [14] achieve strong baseline robustness, but remain sensitive to training distributions and unseen noise. Evaluations typically focus on fixed benchmarks rather than systematic augmentation design or environment reduction, leaving limited understanding of how environment structure impacts robustness.

III. OVERVIEW OF NOPE-HYPE

NOPE-HYPE is a structured workflow for improving and analyzing acoustic robustness in speech-to-text systems, where robustness refers to maintaining strong performance under changing noise conditions and environment types. Instead of treating noise augmentation as a fixed pre-processing step, it decomposes robustness into three controllable stages.

We use real environmental recordings from the DE-MAND dataset [15] with the environment categories listed in supplementary material. First, each environment is summarized by a Power Spectral Density (PSD) template on a shared frequency grid (SFG). A SFG is used so that all environments are represented on the same frequency bins, allowing their PSD curves to be compared directly. Second, when many environment categories exist, we reduce the set by selecting k prototype environments using a coverage objective over PSD templates. Third, we train with SIM using fitted targets from selected environments, generating controlled environment noise mixed with clean speech at sampled SNR levels, and run structured sweeps over simulator parameters to select robustness-oriented configurations. As summarized in Algorithm 1, this workflow enables direct comparison across environments under a unified training protocol.

IV. SIM: A CONTROLLABLE ENVIRONMENT AUDIO SIMULATOR

The simulator supports robust training and structured parameter search while providing direct control over key acoustic factors. It produces deterministic outputs under a fixed seed, enabling reliable ablations and controlled comparisons. It targets realistic environmental noise by matching stable statistical properties of real audio rather than trying to sound indistinguishable from real recordings.

Algorithm 1 Robustness Workflow

- 1: Compute PSD template x_e for each environment e
 - 2: Optionally select few noise environments by coverage optimization on $\{x_e\}$
 - 3: **for** each environment e in P (or all environments) **do**
 - 4: Fit simulator targets: PSD, envelope stats, transient rate, optional tonal peaks
 - 5: **end for**
 - 6: **for** each simulator configuration θ (default or sweep) **do**
 - 7: Generate noise with simulator using θ
 - 8: Mix with clean speech at sampled SNR (optional RIR)
 - 9: Train S2TT model with fixed compute budget
 - 10: Evaluate on clean, real-noise, and sim-noise test sets
 - 11: **end for**
 - 12: Select θ by robustness objective and report ablations
-

A. Interface and Modes

SIM provides three commands: `gen` (ambience-only), `mix` (speech plus ambience at target SNR), and `eval` (noise similarity statistics). Synthesis uses $f_s = 16$ kHz by default. It supports two modes:

- *Preset mode*: A fully procedural synthesis pipeline built from layered primitives. It generates a colored-noise bed (white, pink, or brown), applies multi-band spectral shaping with per-band gain control, and introduces slow stochastic modulation to create nonstationarity. Event primitives such as band-limited transients or ramps can also be added. This mode requires no real recordings and provides full parameter control, making it suitable for bootstrap experiments and factorized sweeps.
- *Generated mode*: Anchors synthesis to real recordings through targets estimated from real ambience, including a mean PSD curve, envelope log-RMS statistics with a slow-dynamics rate hint, transient-rate estimates from onset peaks, and optional tonal peaks. This mode aims to match the stable spectral and temporal properties of real environments while retaining explicit control over each component.

B. Generated Mode Core Building Blocks

The generated mode synthesizes noise from real-environment recordings by matching PSD structure, envelope dynamics, transient activity, chatter, and RIR. Fig. 1 summarizes the pipeline.

PSD estimation and PSD-matched bed. We estimate a target PSD using Welch averaging [17] and synthesize a stationary background bed by matching this target spectrum in the Fourier domain, using target magnitudes

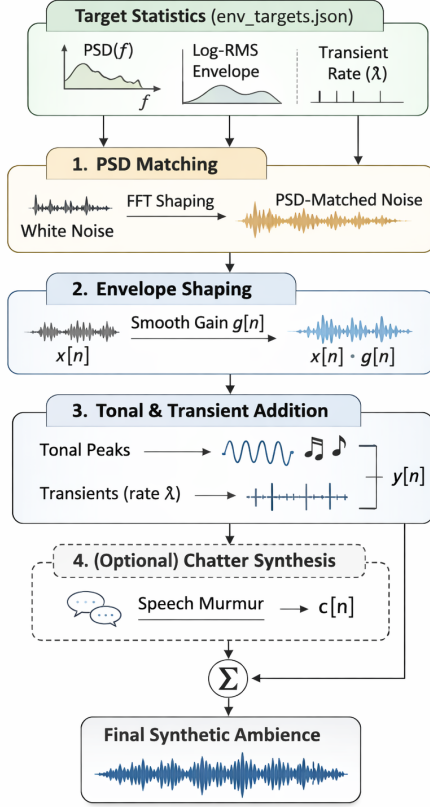


Fig. 1. SIM model pipeline.

with randomized phases and a small number of calibration passes to reduce residual PSD mismatch. This design is motivated by the fact that, for stationary or slowly varying background noise, the PSD captures the dominant spectral structure of the environment, while phase randomization provides a simple way to generate noise in the PSD. During calibration, the PSD correction ratios are clipped to bounded ranges to maintain numerical stability and reproducibility.

Envelope shaping. We apply a smooth multiplicative envelope $a(t)$ to the stationary PSD-matched bed to model nonstationarity. We generate a low-frequency control signal

$$u(t) = \text{LPF}_{r_e}(\epsilon(t)),$$

where $\epsilon(t)$ is white noise and $\text{LPF}_{r_e}(\cdot)$ denotes low-pass filtering with rate parameter r_e . The resulting signal is then exponentiated to produce a positive envelope,

$$a(t) = \exp(\sigma_e u(t)),$$

where r_e controls how quickly the envelope varies and σ_e controls the magnitude of energy fluctuations. This stage allows the simulator to improve beyond a purely

stationary background bed and capture the slow amplitude variation observed in real environments.

Transient injection. Real environments contain short impulsive events such as clicks or impacts. We model these by injecting band-limited transient bursts at a target rate λ events per second, shaped by a smooth envelope and filtered within an environment-specific frequency band.

Chatter injection. We model human background activity by mixing short speech segments sampled from a separate speech pool, band-limited and scheduled in burst-like intervals to mimic background murmur or conversation.

RIR. For indoor environments, we optionally apply a room impulse response during mixing to simulate reverberant propagation. This introduces reflections characteristic of enclosed spaces and provides explicit control over reverberation.

C. Mixing

Given clean speech $s(t)$ and simulated ambience $x_e(t)$, we form a noisy utterance at a target SNR:

$$y(t) = s(t) + g x_e(t),$$

where the scaling factor is

$$g = \frac{\text{RMS}(s)}{\text{RMS}(x_e)} 10^{-\text{SNR}/20}.$$

so that the resulting mixture matches the target SNR. The derivation of this expression is provided in supplementary material. After mixing, we apply mild soft clipping and peak normalization to keep the waveform within a stable numeric range and avoid occasional amplitude spikes.

V. REDUCING ENVIRONMENT SETS

When many environments are available, sweeping simulator hyperparameters across all environments becomes computationally expensive and often redundant due to similar spectral structure. We reduce the environment set by treating it as a coverage problem in PSD-template space and selecting representative prototypes.

PSD templates and distances. Each environment e_i is represented by a centered PSD template $x_i \in \mathbb{R}^F$ on a SFG. We compute distances between PSD templates using two complementary measures:

- *Correlation distance:* This distance captures whether two environments have similar spectral patterns.

$$d_{\text{corr}}(x, y) = 1 - \text{corr}(x, y).$$

- *RMSE:* This distance measures how close two PSD templates are in absolute magnitude across frequencies.

$$d_{\text{rmse}}(x, y) = \sqrt{\frac{1}{F} \sum_f (x_f - y_f)^2}.$$

Objectives of prototype selection. Let $E = \{e_1, \dots, e_N\}$ denote the set of environments represented by PSD templates. Our goal is to select a small subset of environments $P \subseteq E$ of size k , which will serve as prototypes. For any environment e_i , we measure its distance from the selected prototype set as

$$d(e_i, P) = \min_{p \in P} d(e_i, p),$$

which assigns each environment to its closest prototype.

Since the number of environments is small ($N = 12$), we evaluate candidate prototype sets using exhaustive search. For each value of k , we enumerate all possible subsets $P \subseteq E$ with $|P| = k$ and evaluate each subset under both distance metrics. We consider two coverage objectives.

Worst-case coverage, or minimax coverage: As illustrated in the left panel of Fig. 2, this objective is determined by the farthest environment from its nearest prototype. It encourages every environment to remain close to at least one selected prototype:

$$\min_{P \subseteq E, |P|=k} \max_{e_i \in E} d(e_i, P).$$

Average-case coverage: As illustrated in the right panel of Fig. 2, this objective depends on the mean distance from all environments to their nearest prototypes. It favors prototype sets that provide good overall coverage on average:

$$\min_{P \subseteq E, |P|=k} \frac{1}{N} \sum_{e_i \in E} d(e_i, P). \quad (1)$$

This process is conceptually related to clustering methods such as k -means, where cluster centers represent groups of data points. In our setting, the selected environments act as prototype centers, and each environment is assigned to its nearest prototype according to the chosen PSD-template distance.

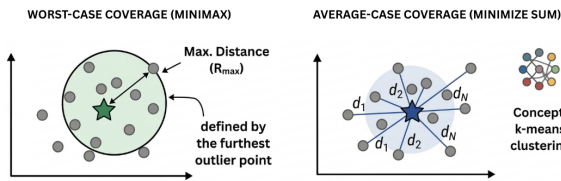


Fig. 2. Prototype-selection objectives: minimax (left) and average-case (right).

Selection of prototypes and choice of k . Using the coverage objectives defined above, we evaluate candidate prototype sets as a function of the number of selected environments k . As shown in Fig. 3, the largest reduction in representation error occurs between $k = 1$ and $k = 3$, after which improvements diminish. Based on worst-case coverage, we treat $k = 3$ and $k = 4$ as the

main operating points. The set $k = 3$ provides compact coverage suitable for efficient experimentation, while $k = 4$ slightly improves worst-case coverage. The exact prototype sets and coverage metrics obtained through exhaustive evaluation are reported in supplementary material.

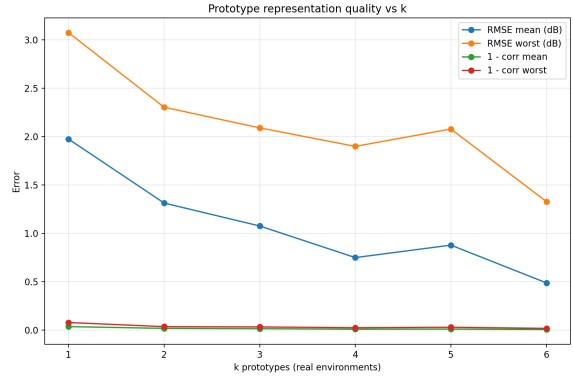


Fig. 3. Prototype representation quality vs. number of selected environments k . Both RMSE and correlation-distance exhibit diminishing returns after $k \in \{3, 4\}$.

VI. EXPERIMENTAL PROTOCOL

We used the Indic-ST corpus [16], focusing on 50 hours of English-Hindi speech for training and a 5-hour held-out evaluation set, with all audio processed at 16 kHz. We also use the same setting for English-German speech from the CoVoST dataset [18]. We evaluate robustness on Whisper [13] and SeamlessM4T [14], fine-tuning medium-sized pretrained checkpoints and evaluating Hindi and German generation on clean, real-noise, and simulator-noise conditions. We report BLEU, chrF, and WER as evaluation metrics. Full training and decoding details are provided in supplementary material.

VII. PRIMARY ROBUSTNESS COMPARISON

We benchmark the primary robustness comparison on the English-Hindi dataset under five training noise conditions using the same 50-hour training budget. The regimes include clean-only training, Gaussian noise, pink noise, balanced real environment noise (uniformly sampled from 12 environments), and SIM-generated noise.

The key outcome is the close similarity between real-noise and simulator-noise training (Table I). In SeamlessM4T, the simulator slightly improves WER relative to real-noise training while maintaining comparable BLEU and chrF. On Whisper, the two are essentially tied. These results show that SIM-generated noise achieves performance comparable to balanced real-noise training under the evaluated conditions.

TABLE I
MAIN ROBUSTNESS COMPARISON (50H TRAIN, 5H TEST).

Training condition	Seamless			Whisper		
	BLEU↑	WER↓	chrF↑	BLEU↑	WER↓	chrF↑
Clean only	47.628	0.432	66.400	34.168	0.572	53.034
Gaussian	46.523	0.447	65.609	29.994	0.627	46.340
Pink (10 dB)	47.606	0.431	66.848	34.019	0.570	53.117
Real env (12)	48.959	0.429	67.893	35.554	0.556	54.143
SIM noise	48.996	0.422	67.896	35.494	0.555	54.160

As shown in Table II, Gaussian noise is a weak proxy for real environments, with especially large degradation on Whisper. Pink noise is stronger than Gaussian, but it still remains consistently worse than real-noise and simulator-noise training on both models. This suggests that matching only coarse spectral color is insufficient, and that the consistent gains of SIM noise over Gaussian and pink noise provide a proof of concept that more structured and realistic environment simulation is beneficial for robust training. It motivates simulator components that go beyond stationary PSD matching to envelope dynamics, transient activity, and chatter.

TABLE II
PERFORMANCE GAP RELATIVE TO SIM NOISE. NEGATIVE BLEU/chrF AND POSITIVE WER INDICATE WORSE PERFORMANCE THAN THE SIMULATOR.

Model	Baseline	Δ BLEU	Δ WER	Δ chrF
SeamlessM4T	Gaussian	-2.473	0.025	-2.287
SeamlessM4T	Pink	-1.390	0.009	-1.048
Whisper	Gaussian	-5.500	0.072	-7.820
Whisper	Pink	-1.475	0.015	-1.043

TABLE III
EFFECT OF CLEAN:NOISE MIXTURE RATIO. ENTRIES ARE MEAN $\pm$ STD OVER SEEDS.

Setting	Noisy BLEU↑	Noisy WER↓	Noisy chrF↑
clean20_noise80	46.140 $\pm$ 0.430	0.465 $\pm$ 0.008	66.630 $\pm$ 0.230
clean40_noise60	46.450 $\pm$ 1.200	0.469 $\pm$ 0.019	66.430 $\pm$ 0.260
clean50_noise50	46.990 $\pm$ 1.580	0.460 $\pm$ 0.026	66.570 $\pm$ 0.310
clean70_noise30	47.120 $\pm$ 0.480	0.457 $\pm$ 0.008	66.420 $\pm$ 0.150
Setting	Clean BLEU↑	Clean WER↓	Clean chrF↑
clean20_noise80	48.760 $\pm$ 0.190	0.436 $\pm$ 0.003	67.880 $\pm$ 0.110
clean40_noise60	48.470 $\pm$ 0.630	0.443 $\pm$ 0.012	67.700 $\pm$ 0.200
clean50_noise50	48.720 $\pm$ 0.860	0.440 $\pm$ 0.014	67.800 $\pm$ 0.030
clean70_noise30	49.910 $\pm$ 0.190	0.425 $\pm$ 0.003	67.950 $\pm$ 0.130

VIII. IMPACT OF SIMULATOR COMPONENTS

We evaluate whether the simulator controls correspond to meaningful factors for robust training on SeamlessM4T. In each ablation, we keep the training pipeline fixed and vary only one control group.

Chatter ON/OFF. Enabling chatter improves both noisy and clean performance (Table IV), suggesting that

speech-like interference is an important factor not well captured by stationary background noise alone.

TABLE IV
CHATTER ABLATION. ENTRIES ARE MEAN $\pm$ STD OVER SEEDS

Setting	Noisy BLEU↑	Noisy WER↓	Noisy chrF↑
CHATTER_OFF	46.450 $\pm$ 0.100	0.464 $\pm$ 0.006	66.610 $\pm$ 0.320
CHATTER_ON	47.870 $\pm$ 0.140	0.450 $\pm$ 0.001	67.440 $\pm$ 0.860
Setting	Clean BLEU↑	Clean WER↓	Clean chrF↑
CHATTER_OFF	48.240 $\pm$ 0.240	0.445 $\pm$ 0.002	67.630 $\pm$ 0.400
CHATTER_ON	49.360 $\pm$ 0.030	0.431 $\pm$ 0.000	68.520 $\pm$ 0.170

RIR ON/OFF. Enabling RIR reduces performance on both noisy and clean evaluation (Table V), consistent with reverberation acting as a domain shift when training and test reverberation are mismatched [3]. RIR should be treated as an explicit control rather than a default.

TABLE V
ADDITIONAL ABLATIONS ACROSS SIMULATOR COMPONENTS.

Setting	Noisy BLEU↑	Noisy WER↓	Noisy chrF↑
RIR off	45.86	0.4593	65.62
RIR on	44.90	0.4798	64.36
Bursty	45.40	0.4676	65.10
Continuous	45.48	0.4658	65.49
Full sim	46.97	0.4464	66.90
PSD only	45.07	0.4645	65.78
Setting	Clean BLEU↑	Clean WER↓	Clean chrF↑
RIR off	47.73	0.4367	67.10
RIR on	47.21	0.4542	65.96
Bursty	47.04	0.4482	66.80
Continuous	47.53	0.4405	67.13
Full sim	48.94	0.4316	67.90
PSD only	47.58	0.4399	67.10

Continuous vs. bursty chatter. The difference between continuous and bursty chatter is small, as shown in Table V. This suggests that the presence of chatter and its overall level matter more than its fine-grained temporal scheduling in this setting.

PSD-only vs. full simulator. The PSD-only variant performs worse than the full simulator, shown in Table V. This indicates that stationary spectral matching alone is insufficient; temporal dynamics and structured interference contribute meaningfully to robustness.

IX. HYPERPARAMETER SEARCH OVER SIMULATOR KNOBS

After establishing that the simulator does not degrade performance and testing the importance of the knobs, we conduct a structured sweep over the simulator hyperparameters to systematically assess how different knob configurations affect performance. We fine-tune only the SeamlessM4T model, as it performs better than Whisper based on previous results.

TABLE VI
SWEEP CONFIGURATIONS (SHARED ACROSS PSD TEMPLATES).

Cfg	$\mu_{\log \text{RMS}}$	$\sigma_{\log \text{RMS}}$	r_e (Hz)	λ	σ_λ	μ_{SNR}	σ_{SNR}
cfg01	-6.600	0.300	0.015	2.200	0.300	0.000	3.000
cfg02	-6.600	0.300	0.046	3.600	0.600	10.000	4.000
cfg03	-6.600	0.300	0.107	4.800	0.250	20.000	5.000
cfg04	-5.200	0.500	0.015	3.600	0.600	20.000	5.000
cfg05	-5.200	0.500	0.046	4.800	0.250	0.000	3.000
cfg06	-5.200	0.500	0.107	2.200	0.300	10.000	4.000
cfg07	-3.300	0.700	0.015	4.800	0.250	10.000	4.000
cfg08	-3.300	0.700	0.046	2.200	0.300	20.000	5.000
cfg09	-3.300	0.700	0.107	3.600	0.600	0.000	3.000

We conduct a structured sweep consisting of 27 experiments on both the language pair datasets: 9 simulator configurations (Table VI) evaluated across 3 fitted PSD templates—cafe, washing (laundry), and car (Section V). Each configuration specifies a set of simulator controls governing background energy statistics, temporal dynamics, and speech-to-noise mixing.

The simulator configurations are defined by a set of parameters; the full list and definitions are provided in supplementary material. We select the best configurations by minimizing the robustness objective

$$\text{AvgWER}_{\text{rob}} = \frac{1}{2} (\text{WER}_{\text{real}} + \text{WER}_{\text{sim}}), \quad (2)$$

while monitoring clean WER to avoid sacrificing clean accuracy.

A. Sweep results and selected default

We summarize the sweep outcomes by aggregating performance across the three PSD templates. The goal is to identify configurations that perform consistently well under the robustness objective in Equation 2.

As shown in Table VII and Table VIII, configuration 3 (cfg03) and configuration 6 (cfg06) achieve the strongest robustness outcomes in this sweep, with consistently low variance across environments:

- **cfg03:** $\mu_{\log \text{RMS}} = -6.6$, $\sigma_{\log \text{RMS}} = 0.3$, $r_e = 0.106812 \text{ Hz}$, $\lambda = 4.8$, $\sigma_\lambda = 0.25$, $\text{SNR} \sim \mathcal{N}(20, 5) \text{ dB}$.
- **cfg06:** $\mu_{\log \text{RMS}} = -5.2$, $\sigma_{\log \text{RMS}} = 0.5$, $r_e = 0.106812 \text{ Hz}$, $\lambda = 2.2$, $\sigma_\lambda = 0.30$, $\text{SNR} \sim \mathcal{N}(10, 4) \text{ dB}$.

TABLE VII
SWEEP RESULTS AGGREGATED OVER PSD TEMPLATES (CAFE, WASHING, CAR). ENTRIES ARE MEAN $\pm$ STD OVER ENVIRONMENTS.

Cfg	Clean WER $\downarrow$	Real WER $\downarrow$	Sim WER $\downarrow$	AvgWER $\downarrow$
cfg01	0.427 $\pm$ 0.003	0.450 $\pm$ 0.005	0.442 $\pm$ 0.002	0.446 $\pm$ 0.003
cfg02	0.425 $\pm$ 0.002	0.444 $\pm$ 0.005	0.443 $\pm$ 0.011	0.443 $\pm$ 0.007
cfg03	0.422$\pm$0.001	0.437$\pm$0.005	0.439$\pm$0.010	0.438$\pm$0.007
cfg04	0.434 $\pm$ 0.012	0.452 $\pm$ 0.016	0.446 $\pm$ 0.016	0.449 $\pm$ 0.016
cfg05	0.432 $\pm$ 0.009	0.449 $\pm$ 0.009	0.447 $\pm$ 0.007	0.448 $\pm$ 0.008
cfg06	0.428$\pm$0.001	0.437$\pm$0.001	0.433$\pm$0.003	0.435$\pm$0.001
cfg07	0.438 $\pm$ 0.008	0.456 $\pm$ 0.005	0.456 $\pm$ 0.013	0.456 $\pm$ 0.009
cfg08	0.426 $\pm$ 0.013	0.450 $\pm$ 0.009	0.443 $\pm$ 0.010	0.446 $\pm$ 0.008
cfg09	0.438 $\pm$ 0.010	0.454 $\pm$ 0.009	0.449 $\pm$ 0.008	0.452 $\pm$ 0.009

TABLE IX
BEST RUN PER PSD TEMPLATE (MIN AvgWER $\downarrow$) FOR EN-HI.

Env	Best exp	Clean WER	Real WER	Sim WER	AvgWER $\downarrow$
cafe	exp02	0.424	0.438	0.433	0.436
washing	exp12	0.420	0.432	0.429	0.430
car	exp24	0.426	0.438	0.431	0.434

TABLE VIII
SWEEP RESULTS FOR ENGLISH TO GERMAN AGGREGATED OVER PSD TEMPLATES. ENTRIES ARE MEAN $\pm$ STD OVER ENVIRONMENTS.

Cfg	Clean WER $\downarrow$	Real WER $\downarrow$	Sim WER $\downarrow$	AvgWER $\downarrow$
cfg01	0.554 $\pm$ 0.006	0.580 $\pm$ 0.003	0.582 $\pm$ 0.004	0.581 $\pm$ 0.003
cfg02	0.553 $\pm$ 0.003	0.582 $\pm$ 0.003	0.583 $\pm$ 0.002	0.583 $\pm$ 0.003
cfg03	0.549$\pm$0.002	0.578$\pm$0.001	0.580$\pm$0.003	0.579$\pm$0.002
cfg04	0.553 $\pm$ 0.006	0.581 $\pm$ 0.006	0.583 $\pm$ 0.004	0.582 $\pm$ 0.005
cfg05	0.552 $\pm$ 0.002	0.581 $\pm$ 0.004	0.582 $\pm$ 0.004	0.581 $\pm$ 0.004
cfg06	0.550$\pm$0.001	0.579$\pm$0.002	0.581$\pm$0.000	0.580$\pm$0.001
cfg07	0.552 $\pm$ 0.004	0.579 $\pm$ 0.003	0.582 $\pm$ 0.005	0.581 $\pm$ 0.004
cfg08	0.552 $\pm$ 0.006	0.581 $\pm$ 0.005	0.585 $\pm$ 0.008	0.583 $\pm$ 0.007
cfg09	0.551 $\pm$ 0.002	0.580 $\pm$ 0.005	0.581 $\pm$ 0.001	0.580 $\pm$ 0.003

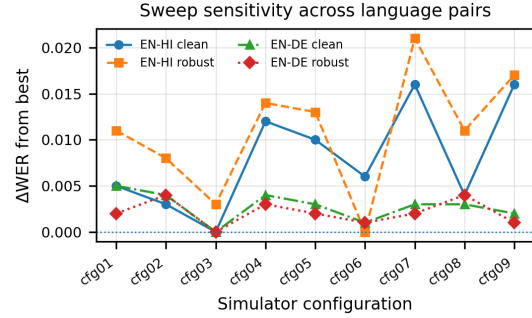


Fig. 4. Sweep sensitivity across language pairs. Values show the WER increase relative to the best configuration for each metric and language pair, using the clean WER and AvgWER $\downarrow$ columns from Tables VII and VIII. Lower is better.

Fig. 4 shows that cfg03 and cfg06 are the most stable choices across both language pairs and across clean and robust evaluation. We do not claim that cfg03 or cfg06 is globally optimal. Rather, they are strong defaults within this small structured search space and across the three PSD templates studied. The comparison suggests that faster envelope dynamics (r_e) is consistently helpful, while robustness can be obtained either through a higher-SNR level with higher event rates (cfg03) or through a moderate-SNR level with lower event rates (cfg06). These trends are empirical for our setting and may change across deployment conditions.

To examine whether the selected default varies by environment, we also report the best run for each PSD template separately.

The best configuration varies slightly by PSD template, as shown in Table IX and Table X, which is expected because different spectral shapes and temporal properties

TABLE X
BEST RUN PER PSD TEMPLATE (MIN AvgWER_{rob}) FOR EN-DE.

Env	Best exp	Clean WER	Real WER	Sim WER	AvgWER _{rob}
cafe	exp04	0.554	0.578	0.581	0.580
washing	exp12	0.548	0.576	0.580	0.578
car	exp19	0.547	0.577	0.577	0.577

imply different masking regimes. In practice, cfg03 and cfg06 provide good general defaults, while per-environment tuning can be treated as a second-stage refinement when deployment is dominated by a single environment type.

X. DISCUSSION

The central design choice in NOPE-HYPE is *controllability*, so that changes in robustness can be attributed to explicit simulator controls or selection objectives. Training with simulator-generated noise matches balanced real-noise training under the same data budget and evaluation conditions, indicating that the simulator captures the statistics that matter for robustness in the studied settings. This framework highlights that prototype selection is most useful when the environment catalog is large and training budgets are limited. Compute PSD templates, select k environments via the coverage objective, and focus fitting and tuning on those prototypes. A conservative workflow is to start with $k = 3$ prototypes for rapid iteration, fit simulator targets from real recordings for those prototypes, train with cfg03 or cfg06 as robust default configurations, and then use ablations to decide whether chatter and RIR are appropriate for the intended deployment domain. When deployment is dominated by a single environment type, the same procedure can be refined by tuning knobs per PSD template using the per-environment selection approach.

XI. CONCLUSION

We introduced NOPE-HYPE, a structured robustness workflow for speech-to-text training that combines a controllable environment simulator, coverage-optimal environment reduction on PSD templates, and a structured hyperparameter search over simulator knobs. We demonstrated performance comparable to balanced real-noise training across Whisper and SeamlessM4T. Through ablations, we showed that dynamics matter beyond stationary spectral matching and provided principled environment prototype sets. Finally, we selected practical default simulator configurations from a 27-run sweep. Together, these components form a reproducible and interpretable path toward robust S2TT training.

REFERENCES

- [1] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015, pp. 3586–3589, doi: 10.21437/Interspeech.2015-711.
- [2] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep Speech: Scaling up end-to-end speech recognition," *arXiv:1412.5567*, 2014.
- [3] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2017, pp. 5220–5224, doi: 10.1109/ICASSP.2017.7953152.
- [4] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617, doi: 10.21437/Interspeech.2019-2680.
- [5] G. Kim, D. K. Han, and H. Ko, "SpecMix: A mixed sample data augmentation method for training with time-frequency domain features," *arXiv:2108.03020*, 2021.
- [6] H. Wang, Y. Zou, and W. Wang, "SpecAugment++: A hidden space data augmentation method for acoustic scene classification," *arXiv:2103.16858*, 2021.
- [7] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv:1609.03499*, 2016.
- [8] K. Kumar, R. Kumar, T. de Boissiere, L. Geste, W. Z. Teoh, J. Sotelo, A. de Brebisson, Y. Bengio, and A. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," *arXiv:1910.06711*, 2019.
- [9] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "AudioLDM: Text-to-audio generation with latent diffusion models," *arXiv:2301.12503*, 2023.
- [10] A. Fazel, W. Yang, Y. Liu, R. Barra-Chicote, Y. Meng, R. Maas, and J. Droppo, "SynthASR: Unlocking synthetic data for speech recognition," *arXiv:2106.07803*, 2021.
- [11] E. Casanova, C. Shulby, A. Korolev, A. Candido Junior, A. da Silva Soares, S. Alufio, and M. A. Ponti, "ASR data augmentation in low-resource settings using cross-lingual multi-speaker TTS and cross-lingual voice conversion," *arXiv:2204.00618*, 2023.
- [12] D. Liu, A. Nassereldine, C. Xu, and J. Xiong, "Towards pre-training robust ASR foundation model with acoustic-aware data augmentation," *arXiv:2505.20606*, 2025.
- [13] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv:2212.04356*, 2022.
- [14] Seamless Communication et al., "SeamlessM4T: Massively multilingual & multimodal machine translation," *arXiv:2308.11596*, 2023.
- [15] J. Thiemann, N. Ito, and E. Vincent, "DEMAND: A collection of multi-channel recordings of acoustic noise in diverse environments," in *Proc. Int. Congr. Acoust. (ICA)*, 2013.
- [16] S. Shah, K. R. Saxena, K. M. Bharadwaj, S. Adavanne, and N. Adiga, "IndicST: Indian multilingual translation corpus for evaluating speech large language models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. Workshops (ICASSPW)*, 2025, pp. 1–5, doi: 10.1109/ICASSPW65056.2025.11011192.
- [17] P. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Trans. Audio Electroacoust.*, vol. 15, no. 2, pp. 70–73, 1967, doi: 10.1109/TAU.1967.1161901.
- [18] C. Wang, J. Pino, A. Wu, and J. Gu, "CoVoST: A diverse multilingual speech-to-text translation corpus," *arXiv:2002.01320*, 2020.

ACKNOWLEDGMENT

AI-Generated Content Disclosure: The authors used AI-based writing and research-assistance tools, including OpenAI ChatGPT, Anthropic Claude Sonnet, Google

Gemini 3.1 Pro, and xAI Grok 4.3, to assist with manuscript review, language polishing, grammar correction, organization suggestions, LaTeX formatting suggestions, and drafting of explanatory text. AI assistance was applied to portions of the Abstract, Introduction, Related Work, Method description, Experimental Protocol, Discussion, Conclusion, and Acknowledgment sections.

The AI systems were not used to fabricate experimental results, datasets, evaluation metrics, figures, tables, or citations. All experimental design choices, simulator implementation, training runs, evaluation results, analyses, and final scientific claims were reviewed, edited, and verified by the authors, who take full responsibility for the final content of the paper.

SUPPLEMENTARY MATERIAL

APPENDIX A METHOD DETAILS

A. Derivation of Noise Scaling Factor

Let $s(t)$ denote the clean speech signal and $x_e(t)$ the simulated ambience. The mixed signal is

$$y(t) = s(t) + g x_e(t), \quad (3)$$

where g is chosen to achieve a target signal-to-noise ratio (SNR).

We define the SNR in decibels using RMS amplitudes:

$$\text{SNR} = 20 \log_{10} \left(\frac{\text{RMS}(s)}{\text{RMS}(g x_e)} \right). \quad (4)$$

Using the scaling property of RMS,

$$\text{RMS}(g x_e) = g \text{RMS}(x_e), \quad (5)$$

we obtain

$$\text{SNR} = 20 \log_{10} \left(\frac{\text{RMS}(s)}{g \text{RMS}(x_e)} \right), \quad (6)$$

$$\frac{\text{SNR}}{20} = \log_{10} \left(\frac{\text{RMS}(s)}{g \text{RMS}(x_e)} \right), \quad (7)$$

$$10^{\text{SNR}/20} = \frac{\text{RMS}(s)}{g \text{RMS}(x_e)}. \quad (8)$$

Solving for g gives

$$g = \frac{\text{RMS}(s)}{\text{RMS}(x_e)} 10^{-\text{SNR}/20}. \quad (9)$$

This is the scaling factor used during speech-noise mixing in SIM.

APPENDIX B EXPERIMENTAL DETAILS

A. Training and Decoding Details

Table XI summarizes the principal training and decoding settings used for the evaluated models.

TABLE XI
TRAINING AND DECODING SETTINGS FOR SEAMLESSM4T AND WHISPER.

Setting	SeamlessM4T	Whisper
Checkpoint	hf-seamless-m4t-medium	whisper-medium
Trainer	HuggingFace Trainer	HuggingFace Seq2SeqTrainer
Audio sampling rate	16 kHz	16 kHz
Batch size	4	8
Gradient accumulation	4	4
Optimizer	AdamW	AdamW
Learning rate	3×10^{-5}	3×10^{-5}
Scheduler	Cosine	Cosine
Warmup ratio	0.1	0.1
Weight decay	0.03	0.03
Maximum gradient norm	0.5	0.5
Epoch budget	Up to 10	Up to 10
Early stopping	Patience 5, threshold 0.005	Patience 5, threshold 0.005
Precision	fp16	bf16 if available, otherwise fp16
Gradient checkpointing	Yes	Yes
Target decoding language	Hindi (en-hi); German (en-de)	Hindi (en-hi); German (en-de)
Beam size	4	4
Maximum new tokens	160	160

B. Coverage-Based Environment Reduction

Table XII reports the environment prototypes obtained under the PSD-based coverage analysis described in the main paper. The prototypes are actual environments rather than synthetic cluster centroids.

TABLE XII
PROTOTYPE ENVIRONMENTS SELECTED USING PSD COVERAGE.

k	Selected prototypes	max d_{corr}	mean d_{corr}	RMSE $_{\mu}$ (dB)	RMSE $_{\text{max}}$ (dB)	Corr $_{\mu}$	Corr $_{\text{min}}$
1	bus	0.077	0.034	1.986	3.073	0.966	0.923
2	cafe, office	0.040	0.021	1.316	2.302	0.979	0.966
3	cafe, washing, car	0.032	0.015	1.074	2.089	0.987	0.969
4	washing, station, car, meeting	0.028	0.012	0.748	1.897	0.992	0.977
5	cafe, living, washing, bus, meeting	0.027	0.010	0.670	2.077	0.994	0.973
6	living, washing, field, station, car, meeting	0.020	0.008	0.489	1.327	0.996	0.984

Increasing k improves spectral coverage of the environment set. The reduction is largest over the first few prototypes and becomes smaller after approximately $k = 3$ –4, consistent with the knee observed in the coverage curve in the main paper. We therefore use the $k = 3$ set, consisting of cafe, washing, and car, as the compact set for the subsequent simulator sweep.

C. Full Simulator Parameter Sweep

For each language pair, the structured sweep contains nine simulator configurations evaluated over three PSD templates: cafe, washing, and car. This gives 27 experiments per language pair.

The full English-to-Hindi run-level results are reported below. To maintain readable text, the results are divided into clean/real-noise and simulator-noise tables rather than compressed into a single extremely wide table.

TABLE XIII
FULL ENGLISH-TO-HINDI SWEEP: CLEAN AND REAL-NOISE EVALUATION.

Env	Exp	Cfg	Clean BLEU	Clean chrF	Clean WER	Real BLEU	Real chrF	Real WER
cafe	exp01	cfg01	48.668	67.761	0.425	47.421	66.771	0.444
cafe	exp02	cfg02	48.762	68.015	0.424	47.736	66.782	0.438
cafe	exp03	cfg03	48.959	67.903	0.423	47.422	66.582	0.438
cafe	exp04	cfg04	47.807	68.204	0.447	46.102	66.876	0.470
cafe	exp05	cfg05	47.869	67.975	0.443	46.771	66.912	0.457
cafe	exp06	cfg06	48.696	67.701	0.429	47.923	66.819	0.436
cafe	exp07	cfg07	48.491	67.617	0.435	47.037	66.399	0.456
cafe	exp08	cfg08	49.695	68.499	0.412	47.612	67.057	0.440
cafe	exp09	cfg09	47.261	67.436	0.449	46.143	66.484	0.463
washing	exp10	cfg01	48.315	68.126	0.430	46.949	67.160	0.452
washing	exp11	cfg02	49.264	68.164	0.424	47.855	66.812	0.446
washing	exp12	cfg03	49.125	68.263	0.420	47.745	66.681	0.432
washing	exp13	cfg04	48.520	67.527	0.433	47.059	66.057	0.443
washing	exp14	cfg05	48.445	68.079	0.428	47.361	67.011	0.440
washing	exp15	cfg06	48.642	68.168	0.429	47.983	67.127	0.436
washing	exp16	cfg07	47.692	67.853	0.446	46.554	66.637	0.461
washing	exp17	cfg08	48.764	67.964	0.428	46.976	66.352	0.454
washing	exp18	cfg09	48.325	67.860	0.429	47.144	66.718	0.445
car	exp19	cfg01	48.480	68.323	0.426	46.801	66.980	0.454
car	exp20	cfg02	48.777	67.943	0.427	47.328	66.603	0.447
car	exp21	cfg03	49.161	67.941	0.422	47.349	66.567	0.441
car	exp22	cfg04	48.911	68.028	0.424	47.445	66.541	0.442
car	exp23	cfg05	48.638	67.574	0.427	47.131	66.293	0.450
car	exp24	cfg06	48.852	68.243	0.426	47.752	66.862	0.438
car	exp25	cfg07	48.664	68.078	0.432	47.051	66.564	0.450
car	exp26	cfg08	48.169	67.857	0.438	46.718	66.479	0.456
car	exp27	cfg09	48.223	68.011	0.436	46.984	66.863	0.455

TABLE XIV
FULL ENGLISH-TO-HINDI SWEEP: SIMULATOR-NOISE EVALUATION.

Env	Exp	Cfg	Sim BLEU	Sim chrF	Sim WER
cafe	exp01	cfg01	47.707	66.971	0.441
cafe	exp02	cfg02	48.112	67.316	0.433
cafe	exp03	cfg03	47.487	66.710	0.440
cafe	exp04	cfg04	46.550	67.108	0.464
cafe	exp05	cfg05	46.913	67.113	0.455
cafe	exp06	cfg06	47.955	66.902	0.435
cafe	exp07	cfg07	46.512	66.485	0.466
cafe	exp08	cfg08	47.726	67.340	0.437
cafe	exp09	cfg09	46.450	66.725	0.458
washing	exp10	cfg01	47.294	67.329	0.445
washing	exp11	cfg02	47.914	66.940	0.440
washing	exp12	cfg03	48.077	67.086	0.429
washing	exp13	cfg04	47.414	66.524	0.438
washing	exp14	cfg05	47.256	67.191	0.442
washing	exp15	cfg06	48.097	67.359	0.434
washing	exp16	cfg07	46.540	66.982	0.461
washing	exp17	cfg08	47.751	66.959	0.437
washing	exp18	cfg09	47.475	67.104	0.443
car	exp19	cfg01	47.416	67.355	0.441
car	exp20	cfg02	46.788	66.647	0.454
car	exp21	cfg03	47.350	66.523	0.447
car	exp22	cfg04	47.774	66.792	0.435
car	exp23	cfg05	47.520	66.540	0.444
car	exp24	cfg06	48.148	67.303	0.430
car	exp25	cfg07	47.563	66.882	0.441
car	exp26	cfg08	46.970	66.703	0.454
car	exp27	cfg09	47.647	67.278	0.445

Each experiment combines one of the nine configurations with one of the three PSD templates. These run-level results are aggregated by configuration in the main paper to compute clean, real-noise, simulator-noise, and robustness-oriented WER statistics.

APPENDIX C DATA AND ENVIRONMENT DETAILS

A. DEMAND Environment Categories

The DEMAND dataset contains recordings from 18 real-world acoustic environments spanning indoor, outdoor, public, and transportation conditions. The environment categories considered in our data preparation are:

- **Indoor:** kitchen, living room, washing area, hallway, meeting room, office, cafeteria, and restaurant.
- **Outdoor and public spaces:** open field, park, river, station, street cafe, and public square.
- **Transportation and traffic:** traffic, bus, car, and metro.

The main experiments use a fixed subset of 12 environments from this catalog for the real-noise comparison and PSD-based environment reduction. The prototype-selection procedure is therefore performed with $N = 12$, as stated in the main paper. The subsequent structured hyperparameter sweep uses the three selected PSD templates: cafe, washing, and car.

B. Spectral Examples from DEMAND

Figure 5 provides qualitative examples of acoustic structure in two DEMAND environments using mean Welch PSD curves and mean STFT spectrograms. The PSD summarizes the long-term frequency-wise energy distribution, while the spectrogram shows how spectral energy evolves over time. The examples illustrate that different environments exhibit distinct spectral profiles and temporal structures, motivating the use of environment-specific PSD templates in NOPE-HYPE.

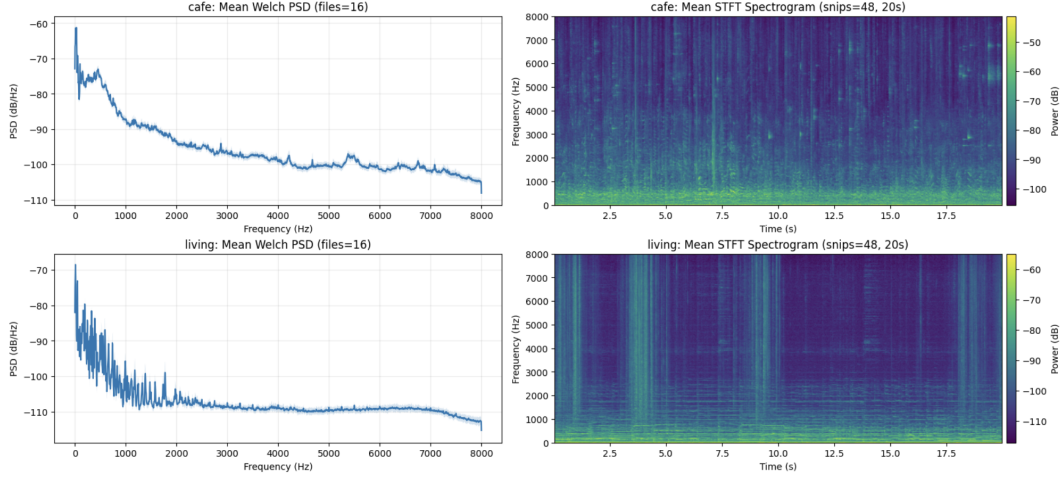


Fig. 5. Qualitative examples of acoustic structure for two DEMAND environments, shown using mean Welch PSD curves and mean STFT spectrograms.

C. Sweep Parameter Definitions

The simulator configurations are defined using the following parameters:

- $\mu_{\log \text{RMS}}$: mean log-RMS level controlling the characteristic background noise energy.
- $\sigma_{\log \text{RMS}}$: variability of the log-RMS background energy.
- r_e : rate parameter controlling slow temporal variation of the ambience envelope.
- λ : mean rate of transient acoustic events.
- σ_λ : variability of the transient-event rate.
- μ_{SNR} : mean SNR used when mixing clean speech with environmental noise.
- σ_{SNR} : variability of the sampled SNR across utterances.