Improving Robustness of PESTO Pitch Estimation

Yifei WANG – DataAI M2

Motivation & Background

Importance

Pitch estimation is key in music/audio processing

PESTO

lightweight, frame-by-frame self-supervised pitch estimator

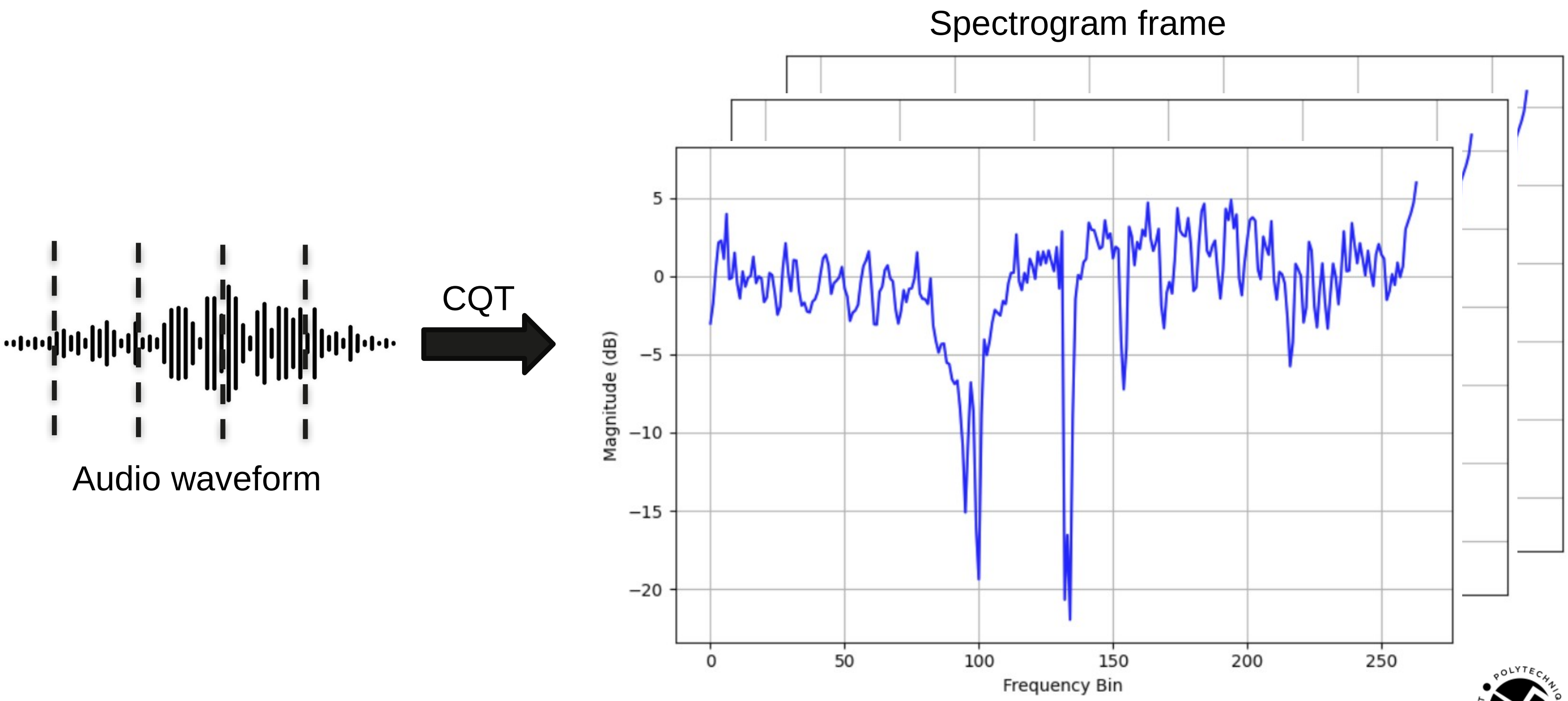
Challenge

sensitive to both low and high-frequency noise

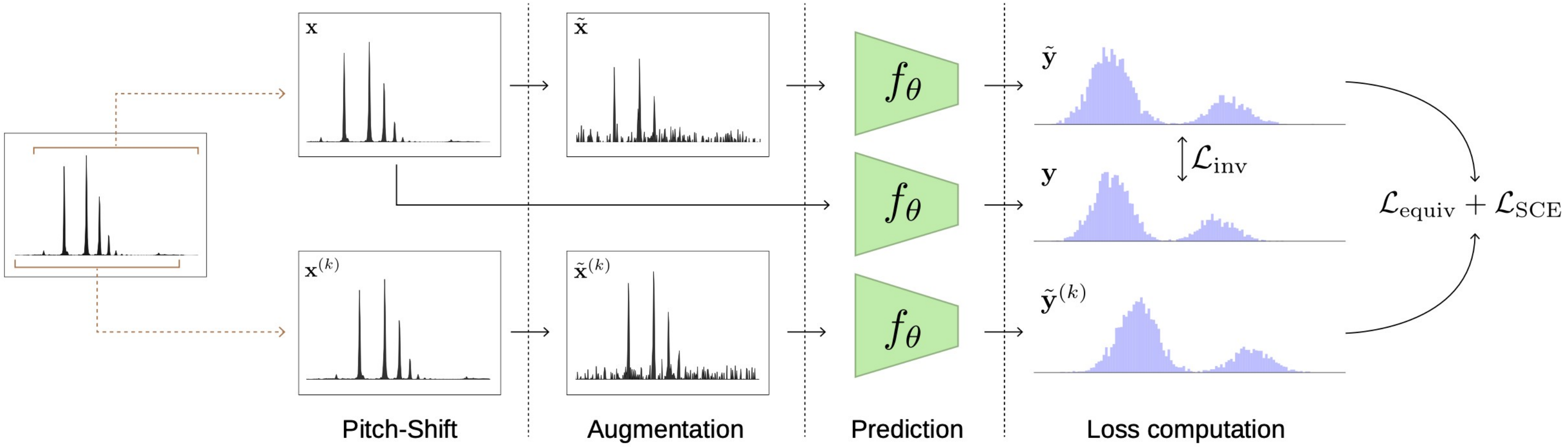
Goal

maintain clean performance while not increasing model size and not sacrificing real-time property

PESTO Framework Overview



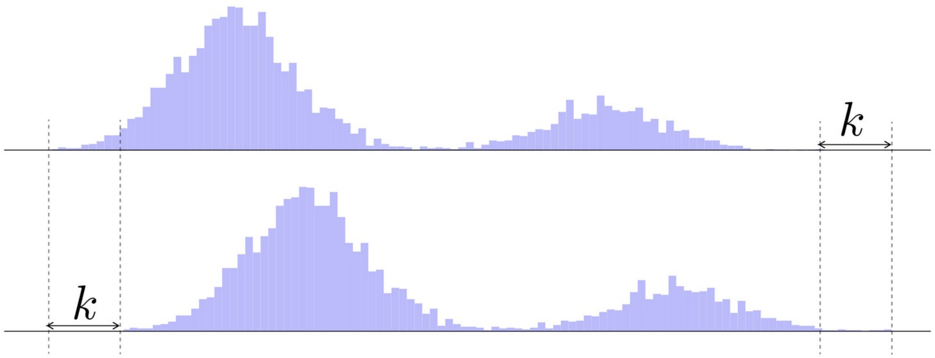
PESTO Framework Overview



$\mathcal{L}_{inv}$: Invariance (stable under augmentation)\n$\mathcal{L}_{equiv}$: Equivariance (consistent with pitch-shift)\n$\mathcal{L}_{SCE}$: Classification (guided by pseudo-labels)

$$
\begin{aligned}
\mathcal{L}(\mathbf{y}, \tilde{\mathbf{y}}, \tilde{\mathbf{y}}^{(k)}, k) = \lambda_{\text{inv}} \; & \mathcal{L}_{\text{inv}}(\mathbf{y}, \tilde{\mathbf{y}}) \\
+ \lambda_{\text{equiv}} \; & \mathcal{L}_{\text{equiv}}(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}^{(k)}, k) \\
+ \lambda_{\text{SCE}} \; & \mathcal{L}_{\text{SCE}}(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}^{(k)}, k)
\end{aligned}
$$

PESTO Framework Overview

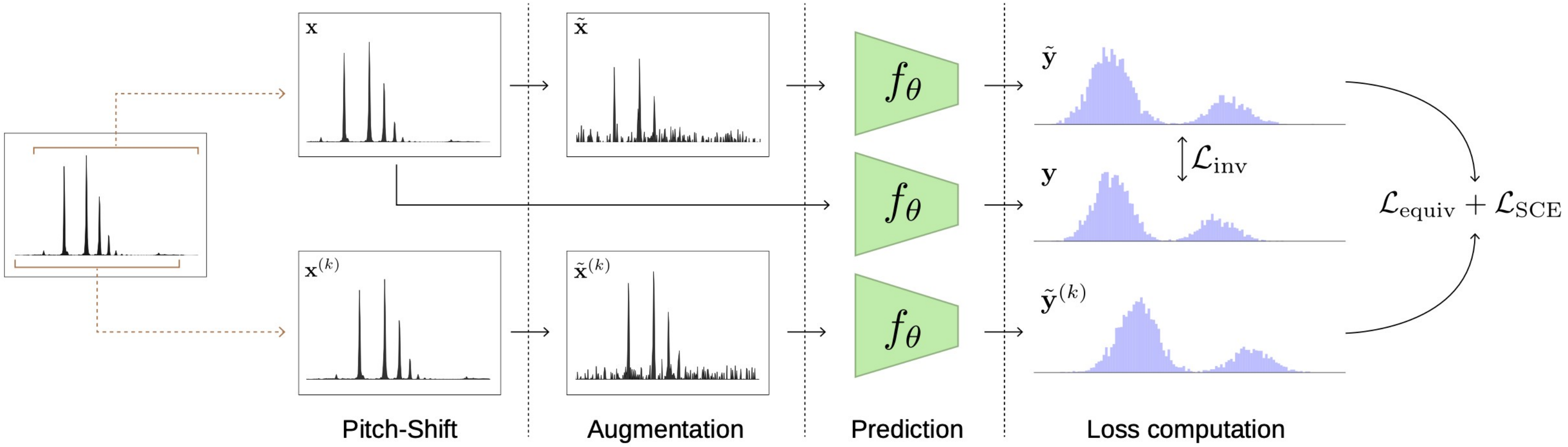


$$y_1, y_2, \ldots, y_n$$

$$y_1 + k, y_2 + k, \ldots, y_n + k$$

Pitch-shift mechanism

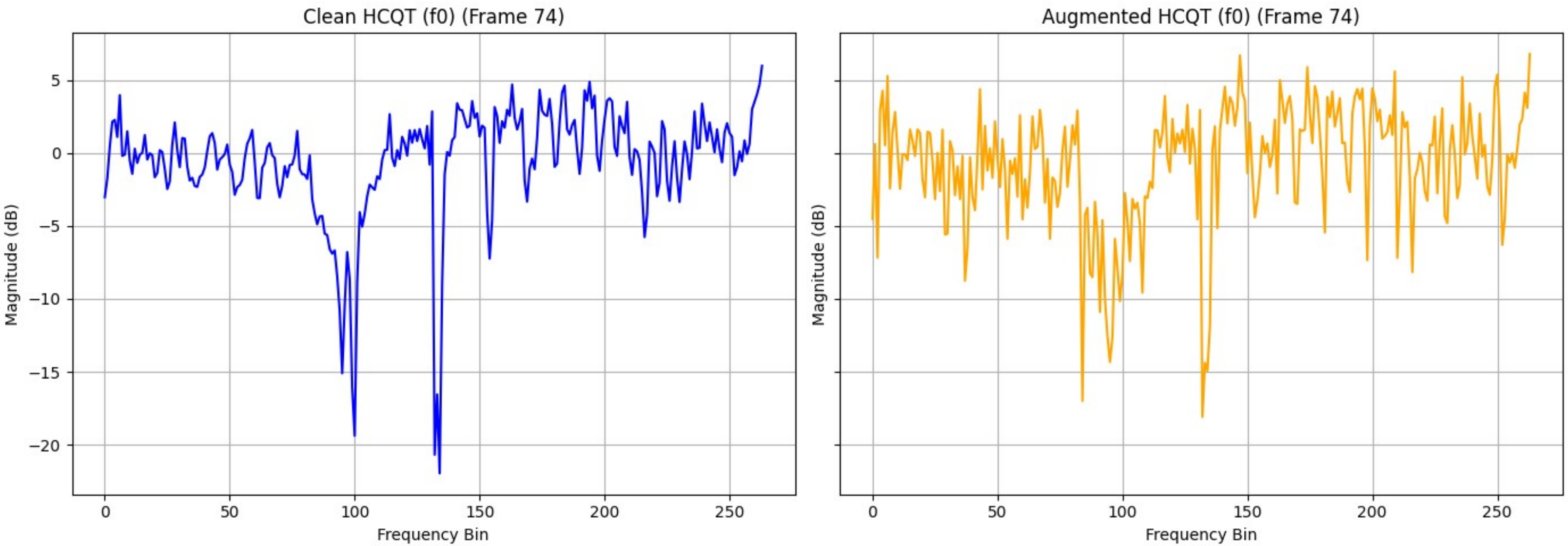
PESTO Framework Overview



$\mathcal{L}_{inv}$: Invariance (stable under augmentation)\n$\mathcal{L}_{equiv}$: Equivariance (consistent with pitch-shift)\n$\mathcal{L}_{SCE}$: Classification (guided by pseudo-labels)

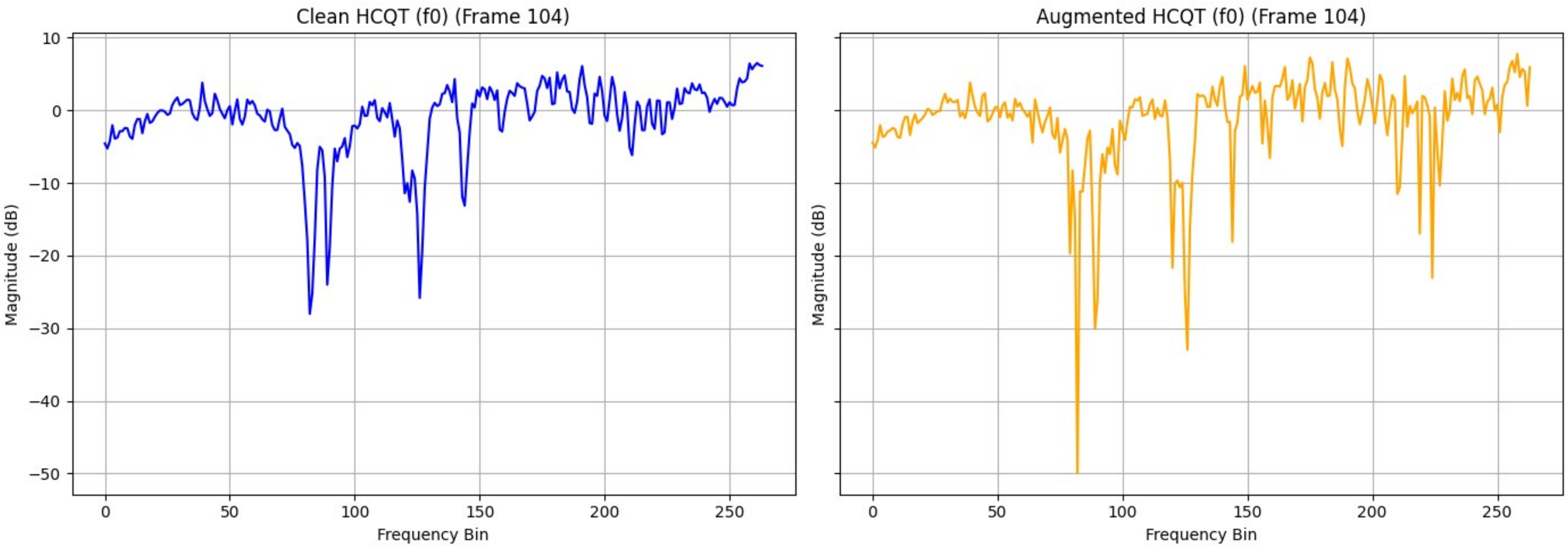
$$\begin{aligned}\mathcal{L}(\mathbf{y}, \tilde{\mathbf{y}}, \tilde{\mathbf{y}}^{(k)}, k) = \lambda_{\text{inv}} \; & \mathcal{L}_{\text{inv}}(\mathbf{y}, \tilde{\mathbf{y}}) \\ + \lambda_{\text{equiv}} \; & \mathcal{L}_{\text{equiv}}(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}^{(k)}, k) \\ + \lambda_{\text{SCE}} \; & \mathcal{L}_{\text{SCE}}(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}^{(k)}, k)\end{aligned}$$

Problem Focus



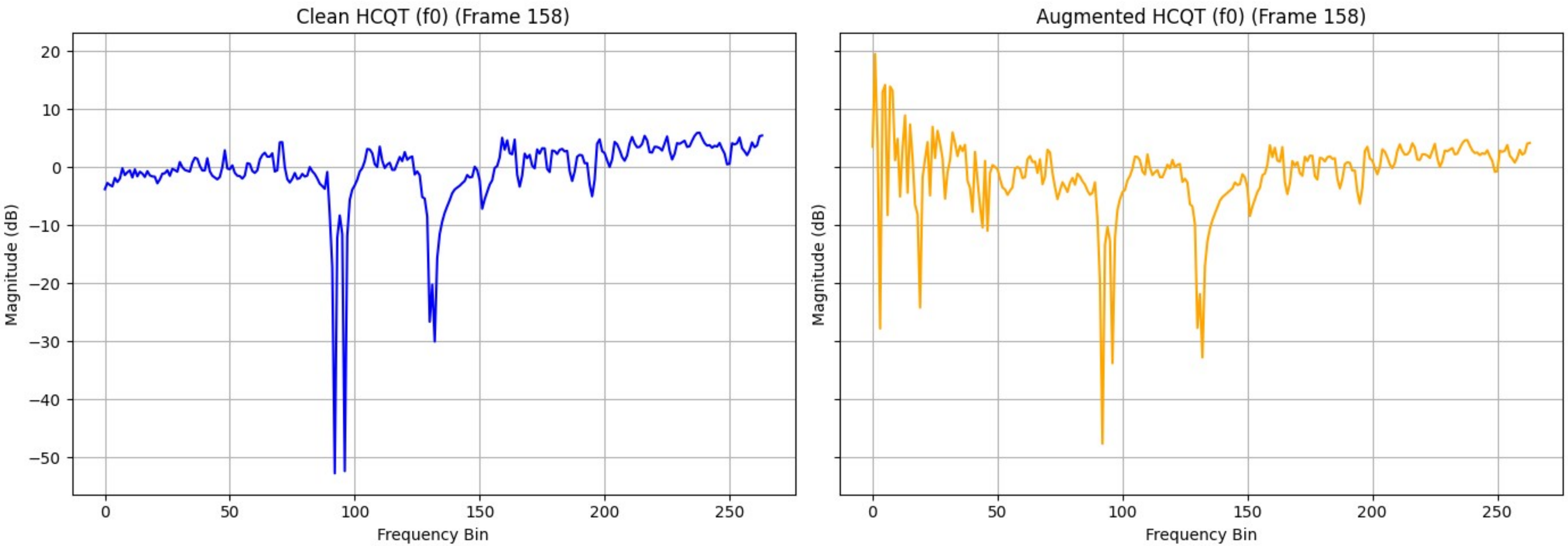
Spectrogram comparison under white noise

Problem Focus



Spectrogram comparison under blue noise

Problem Focus



Spectrogram comparison under pink noise

Design Choices & Constraints

Real-time

processes each frame independently, without temporal context

Constraint

cannot use smoothing or recurrent refinement (would break real-time property)

Feature Choice

CQT is fixed, since its logarithmic frequency axis naturally matches semitone shifts in PESTO’s pitch-shift mechanism

Strategy

design choices focus on noise injection, progressive scheduling, and invariance weighting

Multiple Noise Injection

Intuitive Approach

Training with noise directly improves test-time robustness

Why Feature-domain Noise?

Adding noise in time-domain makes SNR an external factor → not suitable for dynamic weighting

One-noise-per-utterance reduces diversity; frame-level noise too complex for alignment

Feature-domain injection allows per-batch random noise type & intensity, easy to control inside model

Multiple Noise Injection

Noise Modeling

Realistic generation: start from complex Gaussian noise, with spherical sampling for plausibility

Balanced spectrum: apply power normalization, avoid silent bands, enforce spectral correlation

Beyond amplitude-only: add phase perturbation and band-wise variations for richer distortion

Progressive Noise Scheduling

Start simple

early epochs use only white noise to stabilize training and avoid collapse

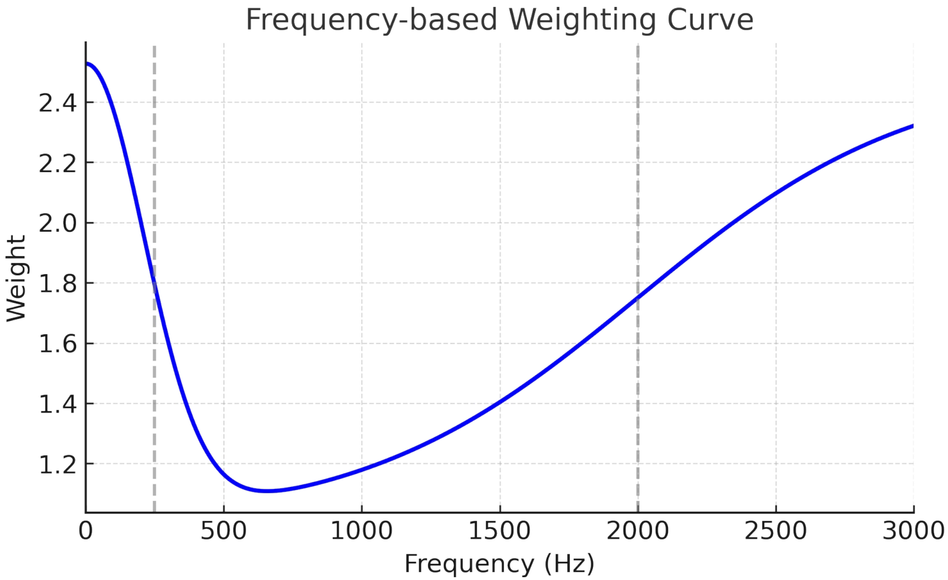
Increase challenge

gradually introduce low-frequency and stronger noises as training progresses

Dynamic Invariance Loss Weighting

Frequency-based weighting

smoothly emphasize both very low (<250 Hz) and very high (>2000 Hz) frames



Dynamic Invariance Loss Weighting

SNR-based weighting

noisier samples weighted higher

frequency + SNR weights combined and clipped

frequency + SNR weights combined and clipped

$$w = \mathrm{clip}(w_{\mathrm{freq}} \cdot w_{\mathrm{SNR}})$$

Experimental Setup

Comparison

original PESTO vs. improved version with noise robustness

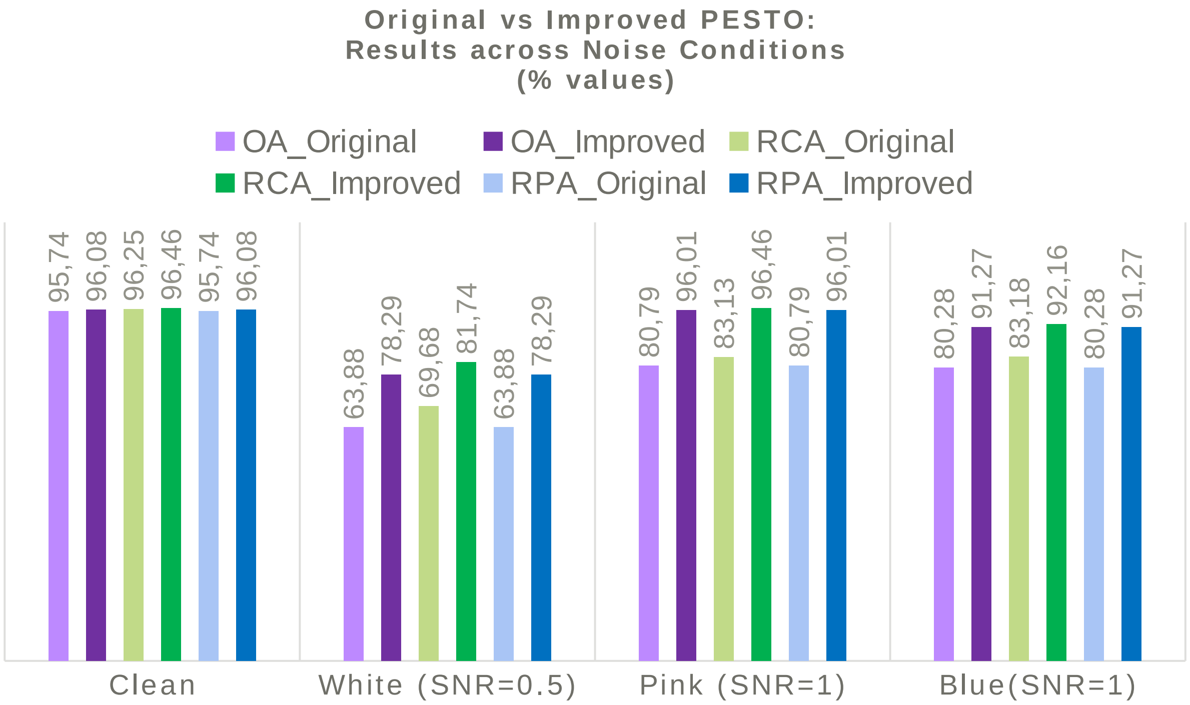
Evaluation conditions

tested under clean speech, white noise, pink noise, and blue noise environments

Metrics

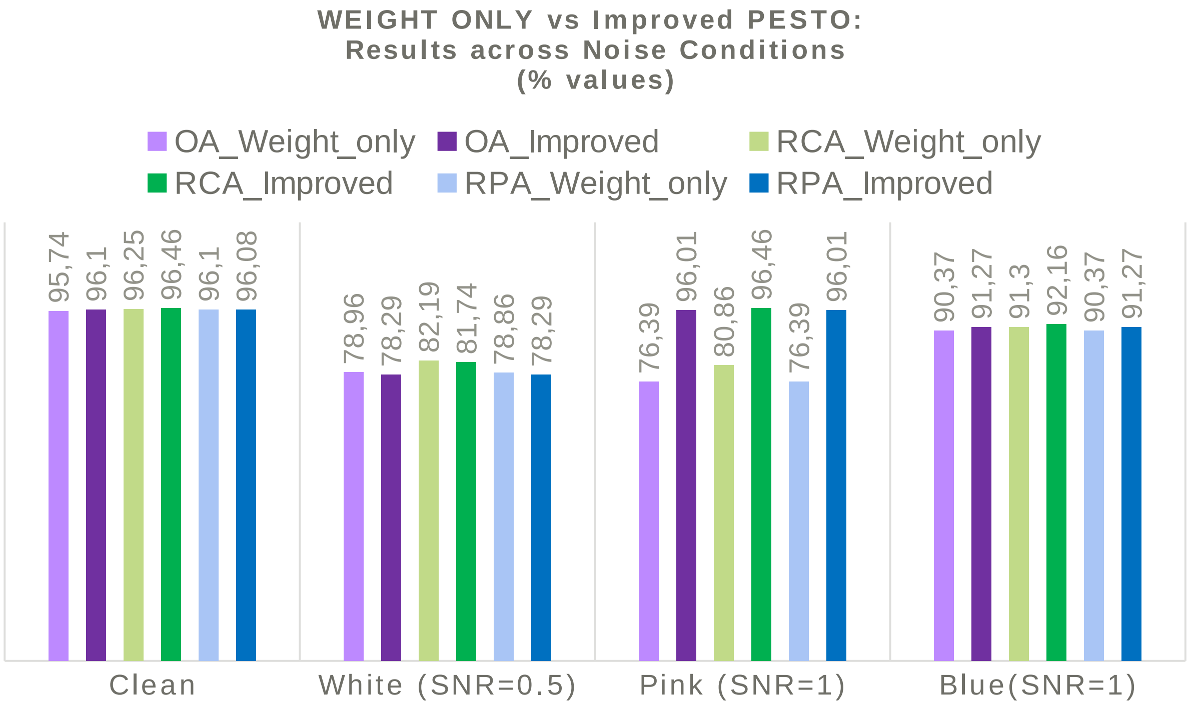
measured with OA (Overall Accuracy), RCA (Raw Chroma Accuracy), and RPA (Raw Pitch Accuracy)

Results

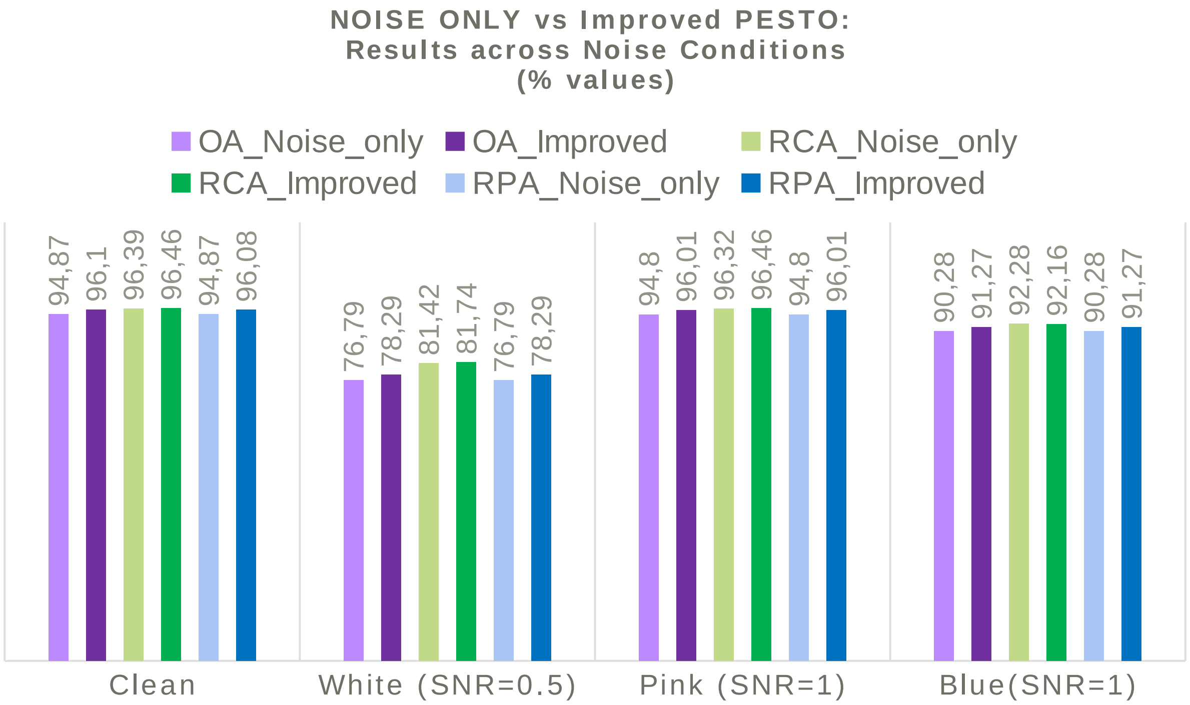


Original vs Improved PESTO:
Results across Noise Conditions
(% values)
OA_Original
OA_Improved
RCA_Original
RCA_Improved
RPA_Original
RPA_Improved
95,74
96,08
96,25
96,46
95,74
96,08
63,88
78,29
69,68
81,74
63,88
78,29
80,79
96,01
83,13
96,46
80,79
96,01
80,28
91,27
83,18
92,16
80,28
91,27
Clean
White (SNR=0.5)
Pink (SNR=1)
Blue(SNR=1)

Ablation Study



Ablation Study



Ablation Study

White noise (extreme case)

Robustness improvement mainly from Dynamic Weighting

Multiple Noise Injection contributes, but less significantly

Low-frequency noise

Improvement primarily from Multiple Noise Injection

Dynamic Weighting alone has little effect

High-frequency noise

Both Dynamic Weighting and Multiple Noise Injection provide substantial gains

Conclusion & Future Work

Conclusion

Introduced Multiple Noise Injection, Progressive Scheduling, and Dynamic Invariance Weighting

Achieved robustness gains without sacrificing clean performance or real-time efficiency

Future Work

Extend to more diverse noise types and real-world datasets

Investigate integration with temporal models without losing real-time property

Thank You !

Q&A

Yifei WANG – DataAI M2