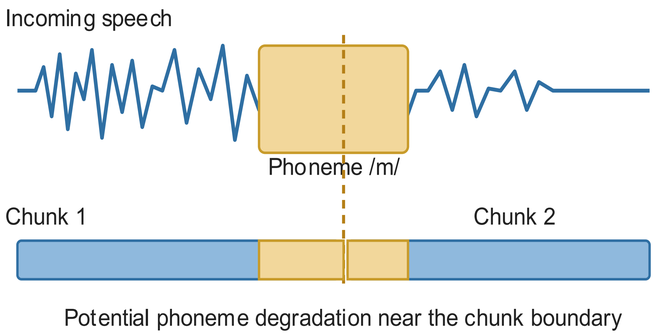
Multi-View Context Processing (MVCP)

Optimization and Phoneme-Level Evaluation of a Bounded-Context Compensation Strategy for ASR

Jason Stuckless, MASc Software Engineering
Supervisors: Dr. Masoud Makrehchi & Dr. Sanaa Alwidian

PROBLEM AND MOTIVATION

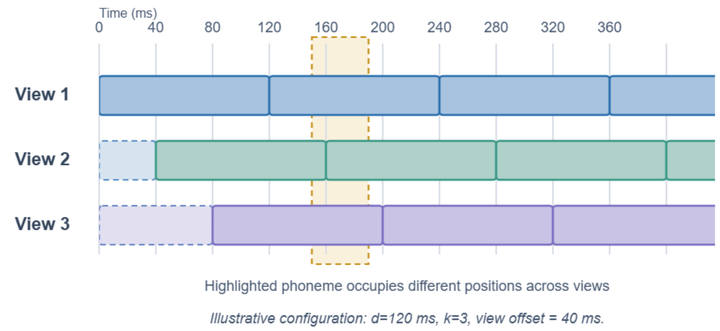
Streaming Automatic Speech Recognition (ASR) processes audio in short chunks with limited context. Phonemes near chunk boundaries may be omitted, misidentified, or assigned inaccurate onset and offset times. This degradation may not be captured by word-level metrics. Bounded-Context Compensation Strategies (BCCSs) seek to mitigate these effects while balancing phoneme preservation, finalization latency, and computational requirements.



MULTI-VIEW CHUNK SCHEDULING

Multi-View Context Processing (MVCP) divides incoming audio into multiple offset chunk schedules, called views. Each view processes bounded-context chunks independently using the same pretrained phoneme recognizer. By shifting chunk boundaries between views, a phoneme near a boundary in one view may receive more complete context in another.

Chunk Duration (d): Determines the nominal duration of each audio chunk.
Number of Views (k): Determines how many offset chunk schedules process the same incoming audio.



The same phoneme is observed under different chunk-boundary conditions, providing complementary predictions that MVCP combines into a single phoneme sequence.

RESEARCH QUESTIONS

RQ1: Phoneme Preservation
Can MVCP improve phoneme preservation and reduce phoneme-level degradation relative to uncompensated bounded-context recognition?

RQ2: Configuration Optimization
How do the five MVCP parameters (d,k,c,p,i) interact to affect phoneme omission, insertion-operation rate, and finalization latency under computational constraints, and what trade-offs emerge between these objectives?

RQ3: Generalization
How well do MVCP configurations optimized on controlled speech recordings generalize to LibriSpeech test-clean and different phoneme recognition architectures?

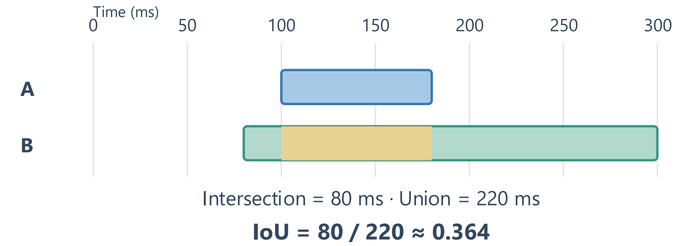
RQ4: BCCS Comparison
How do MVCP's phoneme preservation, insertion behaviour, duration distortion, finalization latency, and computational requirements compare with those of overlap, buffering, and lookahead?

CROSS-VIEW PHONEME CORRESPONDENCE

Predictions from different views may represent the same phoneme event but disagree in identity, onset, or offset times. MVCP uses interval Intersection over Union (IoU) to measure correspondence between predicted phoneme intervals:

$$IoU(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

IoU Correspondence Threshold (i): Predictions satisfy the correspondence similarity criterion when IoU(A,B) ≥ i. IoU distinguishes interval estimates with substantially different extents, even when one interval contains another.



Predictions that do not satisfy the correspondence criterion may form separate provisional correspondence groups rather than being discarded.

COVERAGE ELIGIBILITY AND CONSENSUS

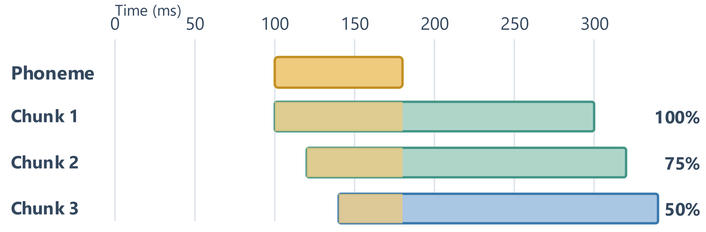
Phoneme Coverage Threshold (p)
Predictions contribute to consensus only when their source chunks cover a sufficient fraction of the provisional phoneme interval:

$$\text{Coverage} = \frac{|I_p \cap I_c|}{|I_p|} \geq p$$

Eligibility is recalculated when the provisional interval changes.

Consensus Formation
The winning phoneme label is selected by supporting-view count, mean confidence, mean boundary-relative reliability, and deterministic tie-breaks. Onset and offset times are estimated using arithmetic means, prioritizing coverage-eligible predictions that do not touch chunk boundaries.

Minimum Support (c)
A consensus event requires at least c supporting views before Competing-Event Reconciliation. This does not permit early finalization.

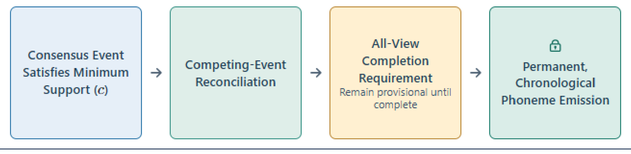


Illustrative p = 0.75: Chunks 1 and 2 satisfy coverage eligibility

RECONCILIATION AND FINALIZATION

Competing-Event Reconciliation
Separate correspondence groups may represent the same phoneme event. Candidates satisfying Minimum Support (c) are reconciled by supporting-view count, mean confidence, and deterministic tie-breaks to prevent duplicate phoneme emission.

Causal Progressive Finalization
A phoneme is finalized only after every view has completed all chunks that could contribute relevant evidence. Emission occurs when the phoneme is locked, after processing completes. Once emitted, a phoneme's identity, interval, and sequence position must remain immutable. The algorithm is designed to preserve chronological, non-overlapping output and online prefix consistency.



FIVE-VARIABLE OPTIMIZATION

MVCP will use constrained multi-objective Bayesian optimization (BoTorch) to jointly optimize five interacting parameters:

$$\boldsymbol{\theta} = (d, k, c, p, i)$$

Optimization Objectives and Constraint
Constrained multi-objective Bayesian optimization minimizes three measures while enforcing computational feasibility:

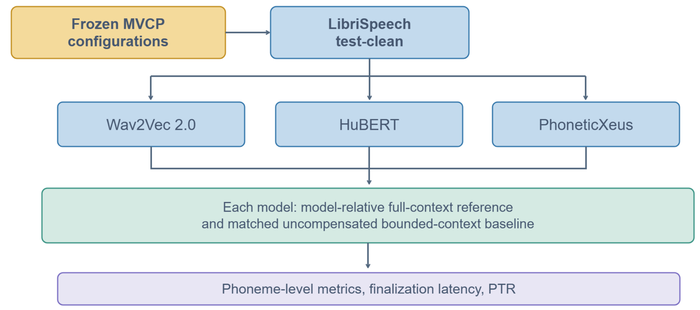
- **POR**: Phoneme Omission Rate, the proportion of reference phonemes omitted.
- R_{ins}: Phoneme Insertion-Operation Rate, relative to the reference phoneme count.
- L_{95}: 95th-Percentile Phoneme Finalization Latency, measured from phoneme occurrence to finalization.
- **PTR**: Processing Time Ratio, the total processing time relative to audio duration. Configurations must satisfy PTR ≤ 1.

$$\min_{\boldsymbol{\theta}} (\text{POR}, R_{\text{ins}}, L_{95}) \quad \text{subject to PTR} \leq 1$$

Optimization Dataset
30 controlled recordings (10 phoneme-targeted scripts × 3 takes) will be used to identify configurations balancing the competing objectives.

MVCP EVALUATION AND BCCS COMPARISON

Independent MVCP Evaluation
Frozen MVCP configurations will be evaluated on LibriSpeech test-clean using Wav2Vec 2.0, HuBERT, and PhoneticXeus, without further optimization or model-specific tuning. Each recognizer will use its own full-context phoneme output as a model-relative reference and a matched uncompensated bounded-context baseline.



Phoneme-Level Evaluation Metrics
POR, PCR, insertion rate, TCI, TEI, ATDI, PLI, and Phoneme Vulnerability Profiles (PVPs), alongside finalization latency and PTR.

Alternative BCCS Comparison
MVCP will subsequently be compared against three Bounded-Context Compensation Strategies (BCCSs):

- **Overlap**: Uses overlapping audio chunks to provide additional context near chunk boundaries, without inherently requiring MVCP's cross-view correspondence and consensus mechanism.
- **Buffering**: Retains contextual audio before recognition or output commitment, allowing additional speech context to inform predictions or finalization decisions.
- **Lookahead**: Incorporates a controlled amount of future audio context before finalization, potentially improving phoneme predictions at the cost of latency.

Comparisons will assess phoneme omission, insertions, duration distortion, finalization latency, and computational requirements under appropriately matched conditions.

CURRENT RESEARCH STATUS

Current Progress
The MVCP algorithm is undergoing implementation and correctness verification, with emphasis on causal processing, incremental consensus, competing-event reconciliation, and irreversible phoneme finalization. Optimization and evaluation experiments are pending, and no validated performance results are currently available.

Correctness Verification
Testing will establish whether finalized phonemes remain immutable as additional audio arrives, preserving chronological output and online prefix consistency. Processing latency and computational requirements must also be measured accurately.

Next Steps
Following implementation and verification, the research will proceed through five-variable Bayesian optimization, independent MVCP evaluation, and comparison with alternative Bounded-Context Compensation Strategies (BCCSs).