Speech Recognition on Cell Broadband Engine

UCRL-PRES-223890

Yang Liu, Holger Jones, John Johnson, Sheila Vaidya
(Lawrence Livermore National Laboratory)

Michael Perrone, Borivoj Tydlitat, Ashwini Nanda
(IBM)

Daniel May
(Mississippi State)

GAIA

Graphics Architectures for Intelligence Applications

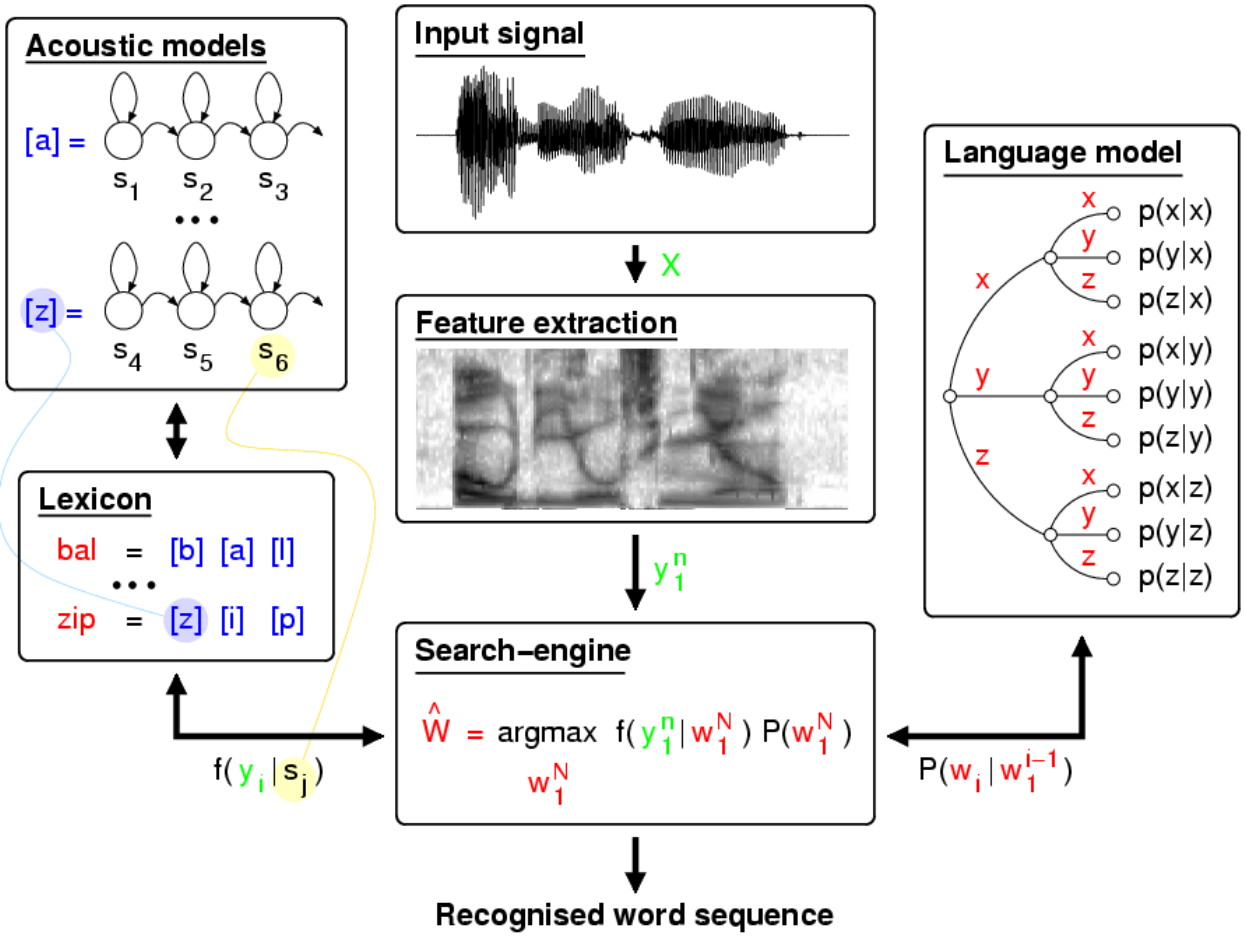
Multi-Channel Speech Recognition

- Motivation
  - Concurrent processing for thousands of speech channels!
  - Each CPU can process only a few (10-30) channels in real-time.
  - **Real-time speech processing of high volume voice traffic**
    - **Can streaming architectures help achieve this goal?**
- Approach
  - Started with Mississippi State open source ISIP toolkit,
    …ported feature extraction onto GPU, and then Cell,
    …implemented isolated digit decoder for Cell,
    …then connected digit decoder for Cell,
    …**and now recently connected phones**.

Outline

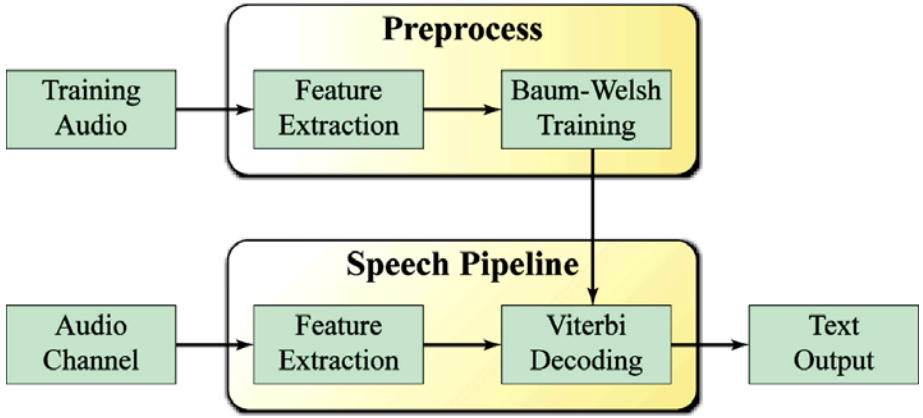
- Speech system
  - Feature extraction
  - Speech decoding
- Cell implementation
  - Isolated digits (Viterbi decoder)
  - Connected digits (Level Building Algorithm)
  - Phones (lexicon / grammar)
- Results
- Future work

Speech System Overview



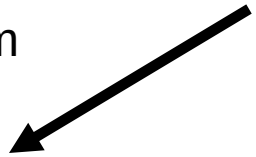
Speech System Components

- **Feature extraction**
  - Encode audio channel into a sequence of feature vectors
    - Characterize spectral and temporal aspects of speech
    - Mel Frequency Cepstral Coefficients (MFCC) features model human auditory response
- **Viterbi decoding**
  - Hidden Markov Model (HMM) based pattern-matching
    - Model digits directly using HMMs
    - Dynamic programming
- Baum-Welch training
  - HMM parameter estimation
    - Require training corpus



Feature Extraction

- Twelve signal processing stages:
  - Window Extraction
  - Zero Mean
  - Energy Computation
  - Preemphasis Filter
  - Hamming Window
  - Spectrum Computation (FFT)
  - Mel Frequency Computation
  - Cepstrum Computation
  - Discrete Cosine Transform
  - Lifter (Cepstral Filter)
  - **Cepstrum Energy Norm**
  - First and Second Derivatives

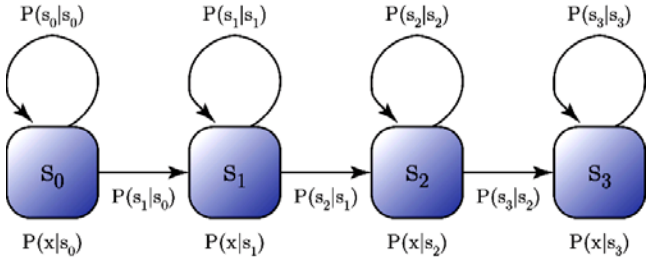


- MFCC feature vector frame:
  - Represents 25 ms of audio w/ 15 ms overlap (10 ms / frame)
  - 250 floats per 25 ms of audio (Pad to 256 for FFT/DCT)
  - Reduce to 39 coefficients
  - Real-time processing requires 100 frames per channel per second
  - Normalize Cepstrum variance for speaker-independence

Viterbi Decoding

- Hidden Markov Models
  - Finite state machine that characterizes observation (MFCC) sequences
    - Transitional probabilities assigned to each directed edge
    - Each state generates an observation drawn from its PDF
  - Stochastically model the process of human speech synthesis

- Viterbi algorithm
  - How are HMMs used to recognize digits from an observation sequence?
    - Sum up probability of all possible paths through the HMM that could have generated the observation sequence
    - Too expensive, instead approximate this with the maximum likelihood path
  - Use dynamic programming to cache intermediate path probabilities



Cell Implementation

Decoding Isolated Digits

- Easy problem
  - Digit word models
  - Exactly one digit per channel
  - Generalizes to more complex speech recognition
- First experiment
  - Very fast prototyping
  - Estimate best performance on Cell
- Components
  - Feature extraction
  - Viterbi decoding

Feature Extraction

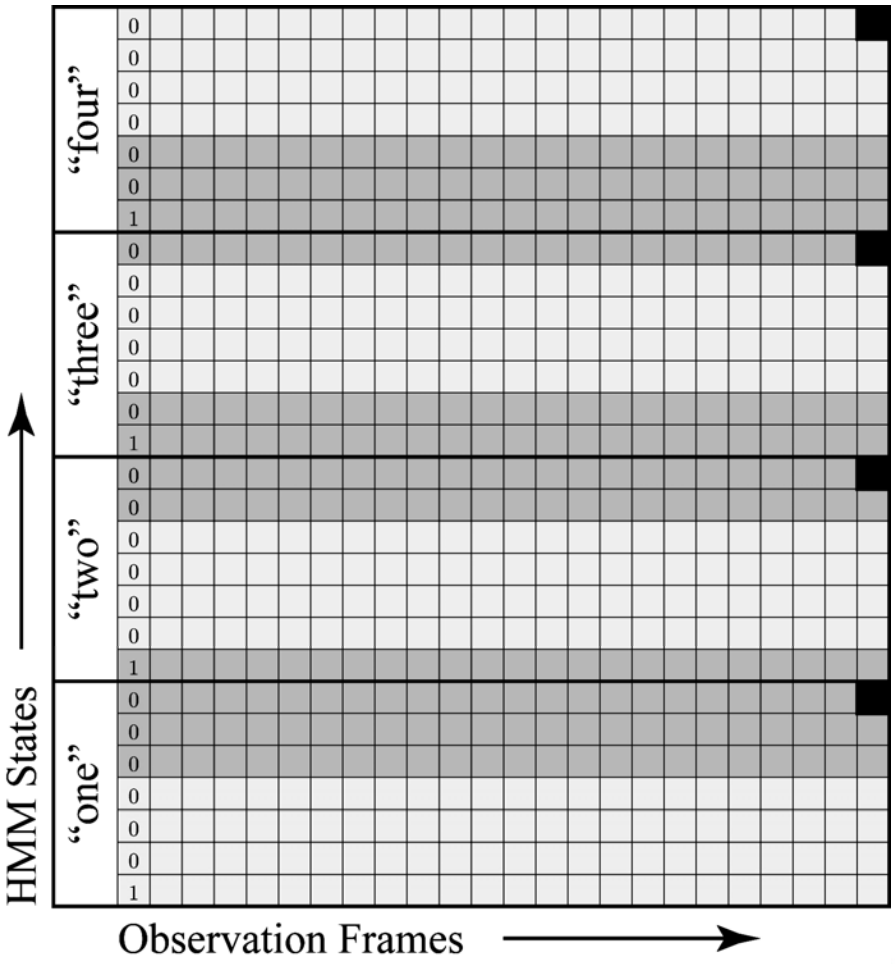
- Straightforward parallel implementation
  - Process 8 signals concurrently (packed into 2 arrays of quad-words)
- FFT is the most computationally intense
  - Take advantage of IBM's extremely optimized Cell FFT library
  - Data format conversion required to interface with FFT library
- Hybrid implementation
  - Process audio channel in single pass
  - Data reduction computed on SPEs
  - First and second derivatives computed on PPE
- Enforce a streaming model
  - Pipeline data communication, and avoid branches

Viterbi Decoding

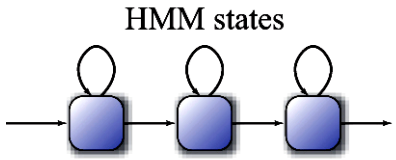
- Computational bottleneck!
  - Data-independent processing → opportunity to leverage parallelism
  - Profile and identify kernels
- Simplifying assumptions:
  - First order HMM
  - Strict left–right model, only allow self & forward transitions
  - PDF approximated by linear combination of four Gaussian kernels (Gaussian Mixture Model – GMM)
- For each HMM state:
  - Calculate observational probability (evaluate GMM)
  - Select maximum likelihood (between self and forward transition)

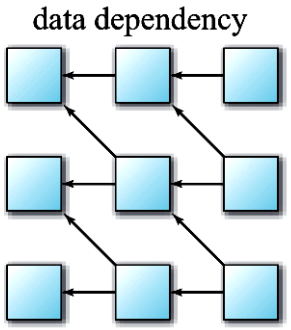
Viterbi Decoding

- Data representation
  - Store HMM states in quad-word vectors
    - SIMD parallelism
  - Transition probabilities across model boundaries are zeroed out to prevent “data contamination”
  - Store only two columns; perform computation “in place”
- Isolated digit recognition
  - Seed with initial probability
  - Decode all HMMs in one pass
  - Extract final probabilities and select maximum



Computation Kernels





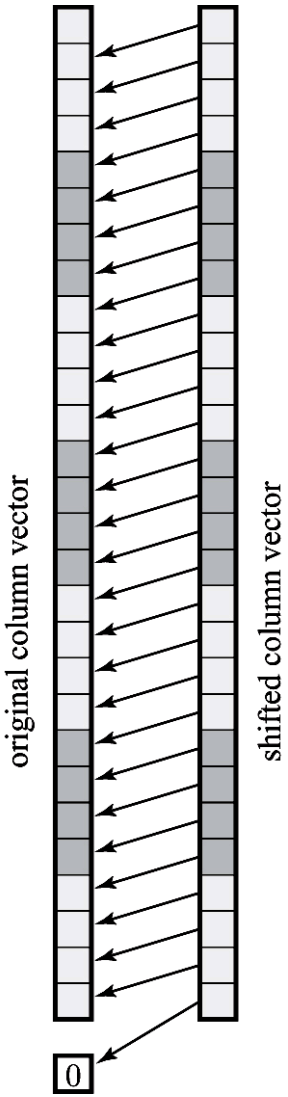
- Observation probability
  - Evaluate feature vector frame at each state:

$$\mathrm{obs}(\mathbf{s}_i) = \mathrm{P}(\mathbf{x} \mid \mathbf{s}_i)$$

- Maximum likelihood
  - Find likelihood of best path through each HMM:

$$\mathrm{L}^{k}(\mathbf{s}_i) = \mathrm{obs}(\mathbf{s}_i) + \max \begin{cases} \mathrm{L}^{k-1}(\mathbf{s}_i) + \mathrm{slf}(\mathbf{s}_i) \\ \mathrm{L}^{k-1}(\mathbf{s}_{i-1}) + \mathrm{fwd}(\mathbf{s}_{i-1}) \end{cases}$$

  - Must shift column vector up by one element to align data access



SPE Programs

- **`spe_mfcc_frontend`**
  - Convert audio sequence to MFCC frames
- **`spe_decode_obs`**
  - Convert MFCC frames to observation probability
- **`spe_decode_max`**
  - Maximum likelihood computation

Decoding Connected Digits

- Multiple utterances in a frame sequence:
  - How many digits are there?
  - Where does one digit end and another begin?
  - What if the speaker talks too fast? (co-articulation)
    (e.g. “two-oh”, “three-eight-two”, “nine-nine”, “six-seven-nine”)

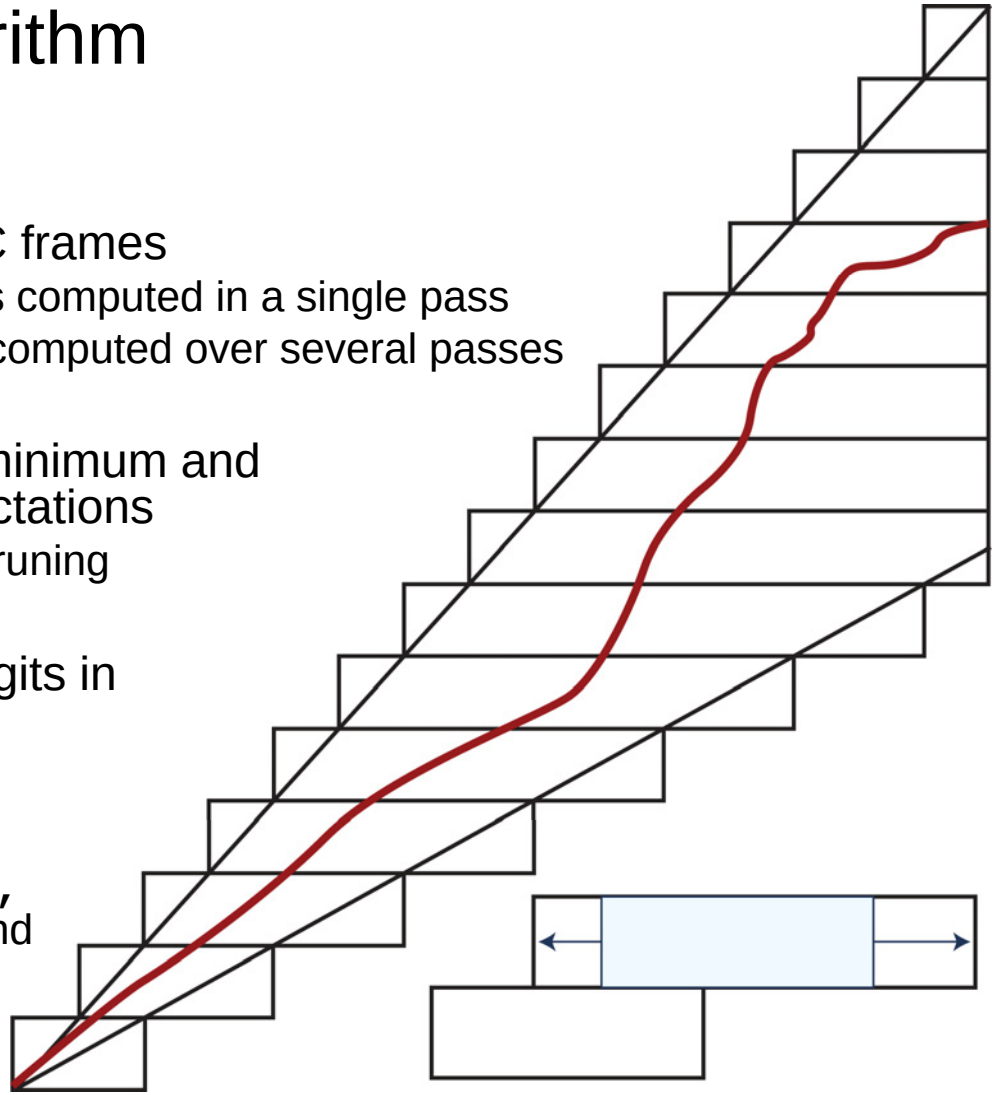
- One naïve approach:
  - Construct large HMM for every possible digit string
    (e.g. “zero-zero-zero”, “zero-zero-one”)
    - Too many combinatoric possibilities!
    - Duplicate computation!

Decoding Connected Digits

- Level Building Algorithm
  - Explore all possible combinatoric arrangements of HMMs…
    …but **cache intermediate computations**
- Basic idea:
  - First digit starts at frame 0, but may terminate anywhere
  - Decode second digit at each possible ending frame for the first digit
  - Now the second digit can end anywhere
  - Decoding results overlap, so select the **best probability at each frame**
  - Each frame also needs to maintain its starting frame pointer and label of HMM with best probability
  - Organize computation into **levels**
    - Decode at most one digit per level

Level Building Algorithm

- Multiple passes over MFCC frames
  - Observational probabilities computed in a single pass
  - Maximum-likelihood path computed over several passes
- Bound computation using minimum and maximum digit length expectations
  - Opportunity for dynamic pruning
- Traceback step decodes digits in reverse
- PPE implementation
  - Drives **`spe_decode_max`**, maintains bookkeeping, and performs traceback



Decoding Phones

- Linguistic phone
  - Atomic unit of speech
  - Support arbitrary vocabulary size
  - More efficient to decode than word models
  - Triphone models provide left / right context
- Implementation
  - Not so different from word models
  - Fewer states per phone model → fewer HMM state pruning opportunities
  - Apply lexicon / language grammar to decode valid words from phones (more later)

Preliminary Results

Experiment Setup

- Vocabulary
  - Recognize digit phones
  - 3 state phone HMMs
  - 20 classes, 57 total states
- Gaussian Mixture Model
  - 4 Gaussians per mixture
  - One mixture per state
- Corpus
  - TIDIGITS
- Connected phone recognition
  - Level Building Algorithm
  - Phones between 5 to 20 frames
- Training
  - HMMs trained off-line using MS ISIP's Baum-Welch
- Platforms
  - 3.2 GHz Pentium (CPU)
  - 4.0 GHz Cell simulator (SIM)
  - 2.0 GHz Cell beta hardware (CELL)

System Performance

	Real Time Channels (100 frames / sec)		
	CPU (3.2 GHz)	SIM (4.0 GHz)	CELL (2.0 GHz)
Feature Extraction	500	45000	13235
Connected Phones	6	1625	1098
Total (FE + CP)	**5**	**1568**	**1013**

- Timing methodology:
  - Ignored initialization and network I/O
  - `gettimeofday()` wrapper around critical code
  - Timed for single SPE, extrapolated to all 8

- Gained at least **two orders of magnitude** of performance on Cell, but does not include lexicon / grammar / pruning optimizations!

Sample Run

```
[root@(none) speech]# ./decode.ppe
Allocating levels...
Loading Gaussian means...
Loading Gaussian inverse variances...
Loading Gaussian weight factors...
Loading Gaussian scale factors...
Loading transitional probabilities...
Loading MFCC file...242 frames.
Computing observational probabilities...
Decoding maximum likelihood...
nLevels = 48
min_levels = 9
prob = -17111.398438
Decoded: [sil] f ay v n ay n ah ay n s eh v ih n s w ih k s [sil]
Obs time: 0.002181 seconds.
Max time: 0.002976 seconds.
Tot time: 0.005157 seconds.
Cleaning up...
[root@(none) speech]# _
```

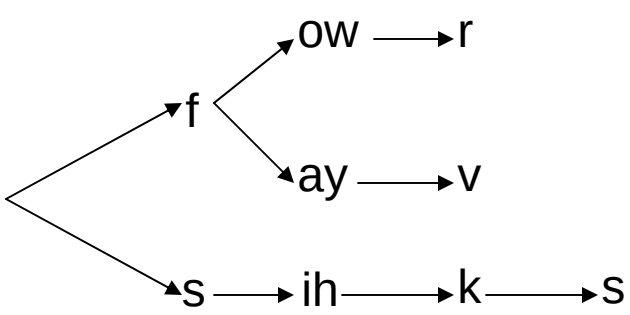
Next Steps…

Sample Decode

- Speech decode using only acoustic phone models:
  - **[sil] f ay v n ay n ah ay n s eh v ih n s w ih k s [sil]**
    ([sil] five nine nine seven six [sil])
- What is “swihks”?
- Apply lexicon constraints to decode actual words from phones
  - Implement lexicon at level boundaries in LBA on PPE

Lexicon

- Phone transcriptions
  - FOUR – f ow r
    FIVE – f ay v
    SIX – s ih k s
- Lexical graph



- Implementation
  - Must maintain state for each decode
  - Managed by Level Building Algorithm on PPE
  - Opportunity to also apply phone bi-gram / tri-gram
  - Difficult to leverage SIMD parallelism

Larger Corpora

- TIDIGITS (Early Prototyping)
  - 326 speakers (111 men, 114 women, 50 boys, and 51 girls)
  - Each pronounce 77 digit sequences, 11 words, no grammar
- TIMIT (Develop Phonetic System)
  - 630 speakers (representing 8 major dialects of US)
  - Each speak 10 sentences ~6000 words, limited grammar
- Wall Street Journal (WSJ)
  - Read from WSJ news text
  - Also includes spontaneous dictation
- Switchboard (Real Speech Problem)
  - 2430 spontaneous conversations from over 500 speakers across telephone
  - Noisy data, speech vs. non-speech (stuttering, incomplete words, etc)
  - Unknown vocabulary and/or language grammar

Future Work

- Larger vocabulary
  - More challenging corpora (WSJ, Switchboard) for performance / accuracy tradeoff study
  - **Lexicon** / language grammar (bigram / trigram)
- PPE optimizations
  - Model / word level hierarchical pruning
  - **SPE system load balancing**
- SPE optimizations
  - **Chain feature extraction kernels** (pending DMA issues)
  - Tied Gaussian mixtures
  - HMM-level pruning

Questions?

- Contact:
  - Yang Liu (liu24@llnl.gov)
  - Holger Jones (jones19@llnl.gov)