A FIRST ATTEMPT AT POLYPHONIC SOUND EVENT DETECTION USING CONNECTIONIST TEMPORAL CLASSIFICATION

Yun Wang and Florian Metze
{yunwang, fmetze}@cs.cmu.edu
Carnegie Mellon University
Language Technologies Institute

Introduction

Task: Sound event detection – detect the (type, starting time, ending time) of each occurrence

Conventional solution: Recurrent neural networks

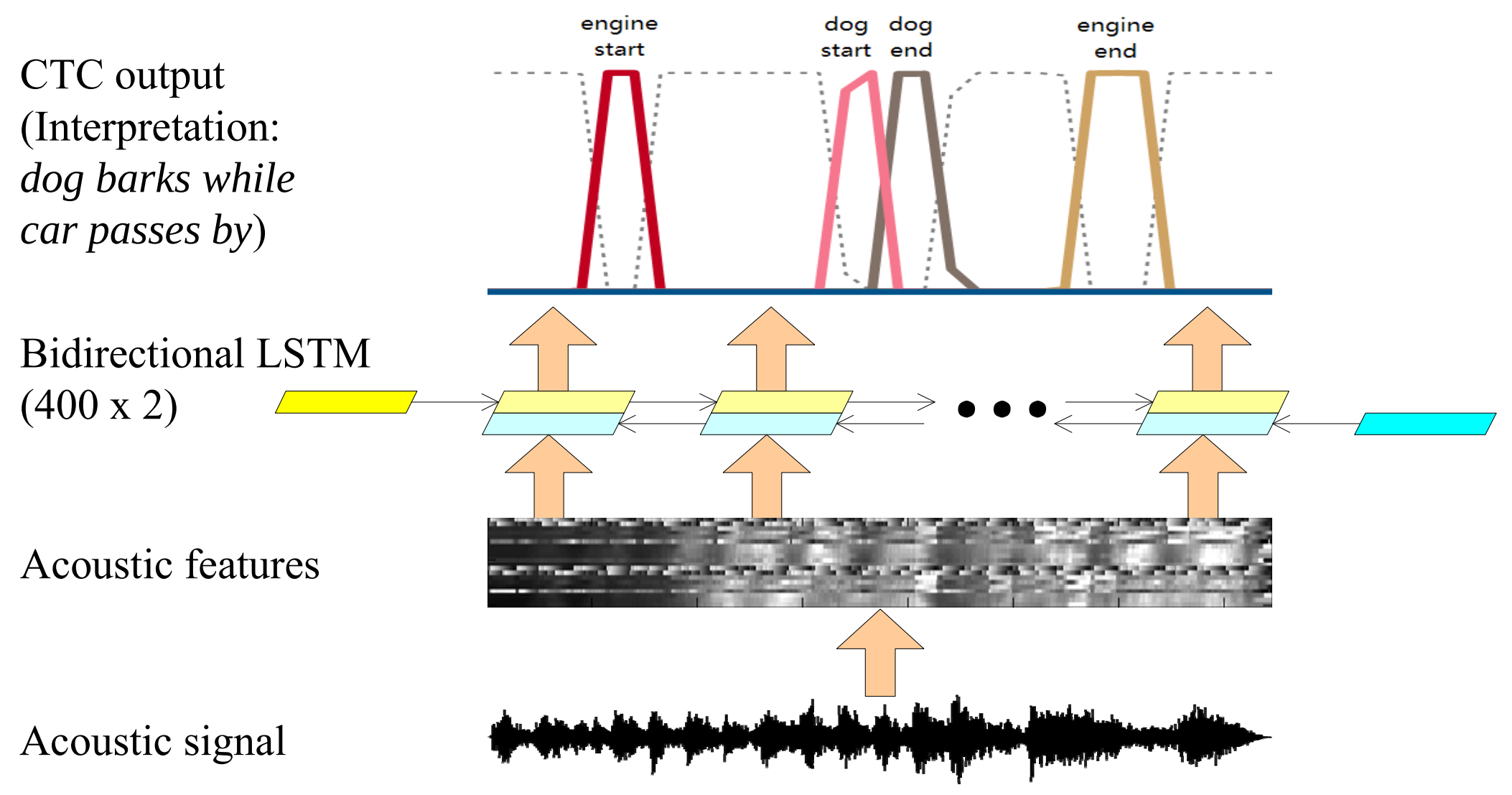
Problems:

1. Polyphony – multiple events may overlap
2. Inexact timing – labeling the starting and ending times of each event can be tedious, and these boundaries can be ill-defined

Proposed solution:

- Detect the sequence of event onsets and offsets with CTC, and expect to generate peaks near the true locations of event boundaries

System Architecture



CTC vocabulary: {start, end} for each event + {blank}

References

[1] S. Burger, *et al.*, “Noisemes: manual annotation of environmental noise in audio streams”, technical report CMU-LTI-12-07, 2012.

[2] Y. Wang, L. Neves, and F. Metze, “Audio-based multimedia event detection using deep recurrent neural networks”, ICASSP 2016.

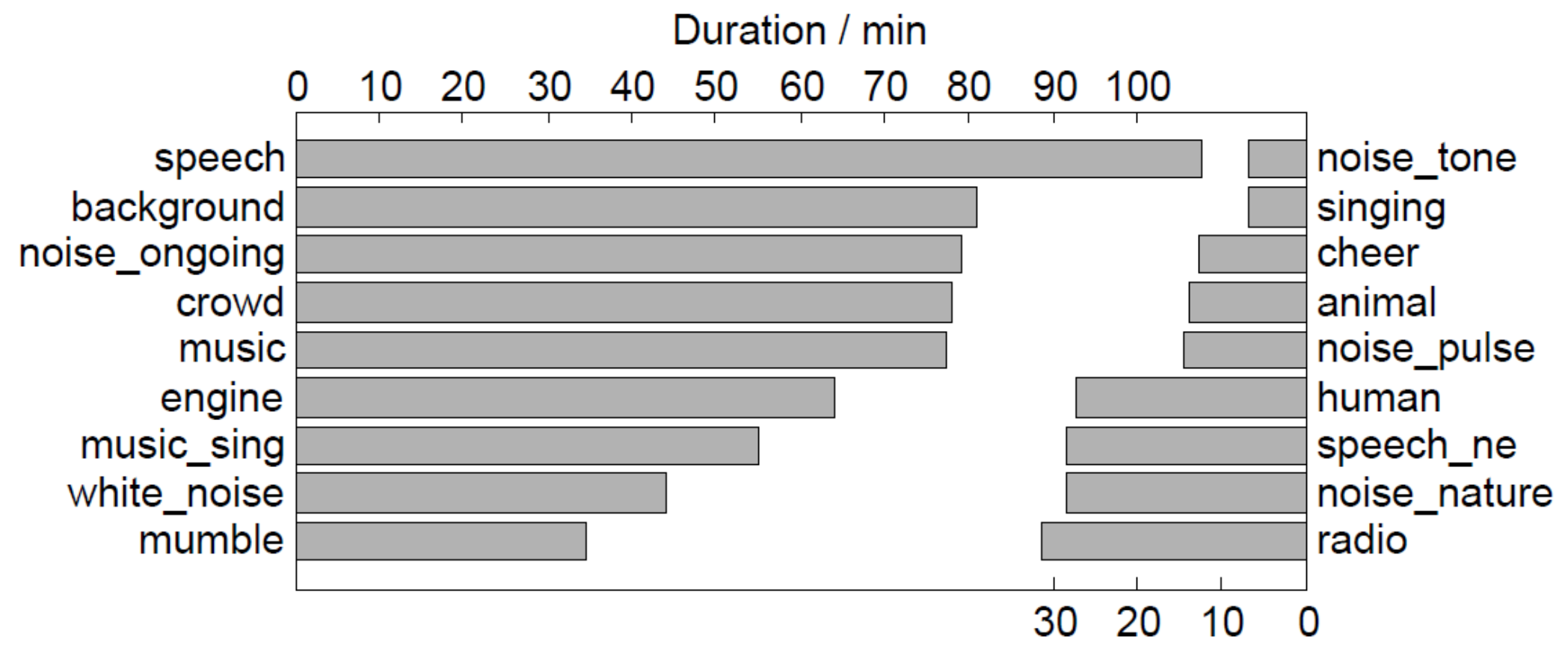
[3] Y. Aytar, *et al.*, “SoundNet: Learning sound representations from unlabeled video”, NIPS 2016.

This work was supported in part by a gift award from Robert Bosch LLC.
This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by NSF grant number OCI-1053575.

Training the CTC-RNN

Corpus:

- Noiseme corpus [1], expanded
- 464 recordings, 9.6 hrs (60% train, 40% test)
- 48 sound events, merged to 17
- Average polyphony: 1.44



Feature extraction:

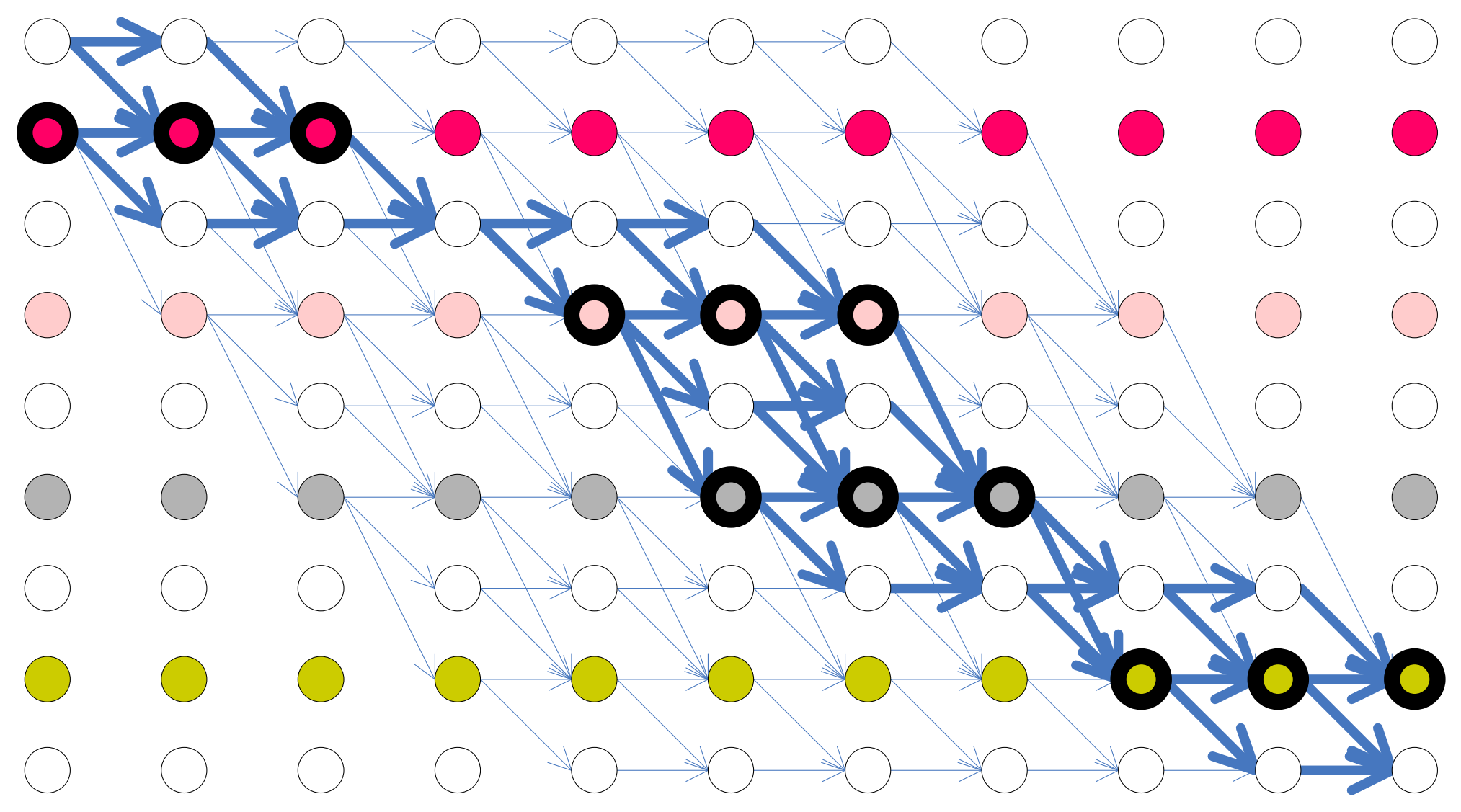
- MFCC, F0, etc extracted with OpenSMILE, 100 frames / second
- 6,669-dim statistics over 2-second windows, 10 windows / second
- Reduced to 50 dims with PCA

Training method:

- Objective: Per-frame negative log-likelihood
- Batch size: 5 segments of 500 frames
- Optimizer: SGD, Nesterov moment = 0.9
- Learning rate: 0.3 until 200 epochs; decay by 0.99 until 500 epochs

Tricks for training:

- Pre-training with a framewise event detector (improved from [2], frame accuracy 55.5%)
- Gradient clipping at 0.001
- Alignment hinting: Each token must occur within *k* frames of the ground truth

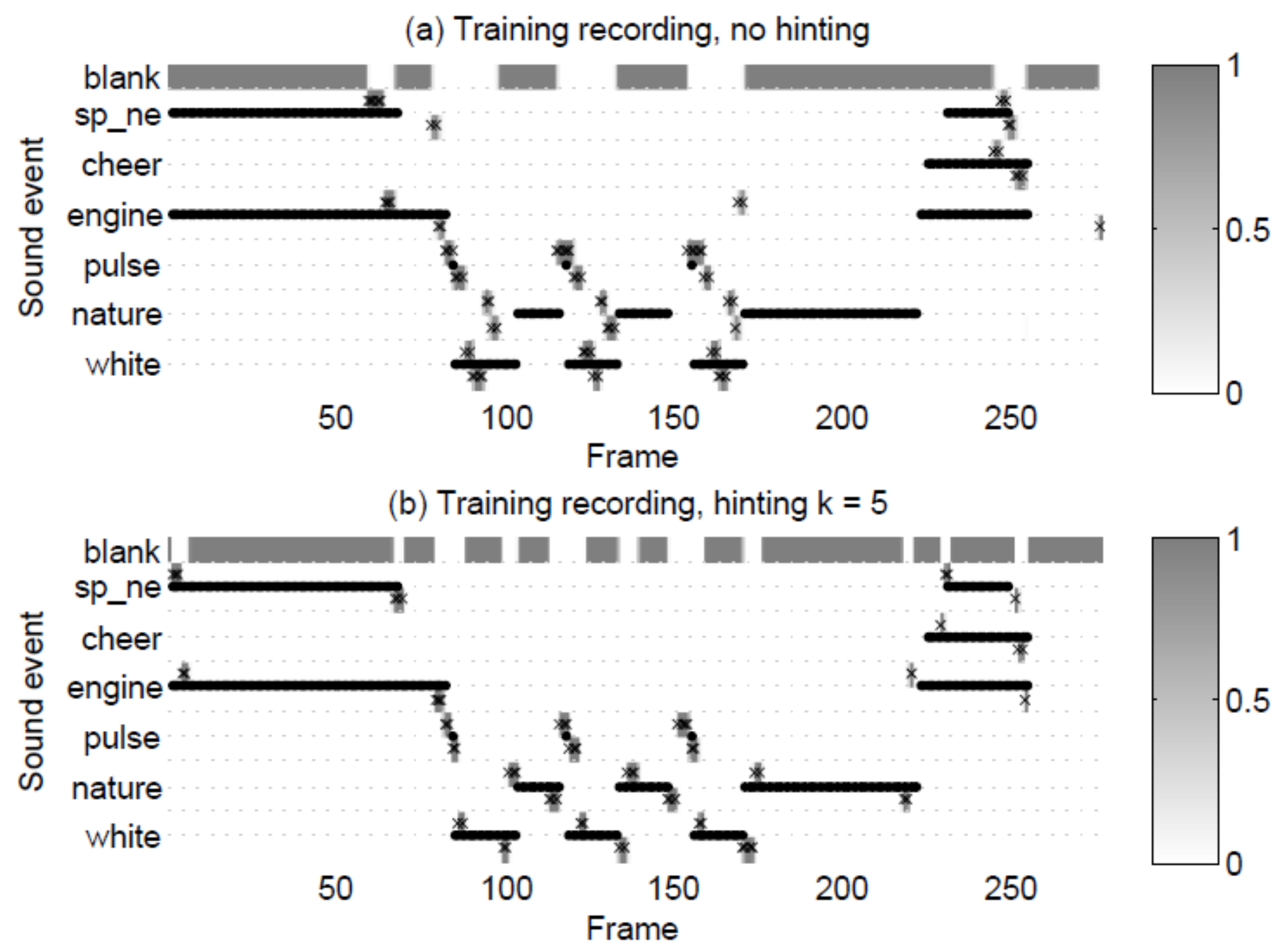


Example of alignment hinting:
k = 1, only thick circles and arrows are allowed

Qualitative Analysis

On training data:

- The sequence of event boundaries can be almost perfectly recovered
- But the boundaries tend to cluster together
- Alignment hinting makes the peaks fall in the right places

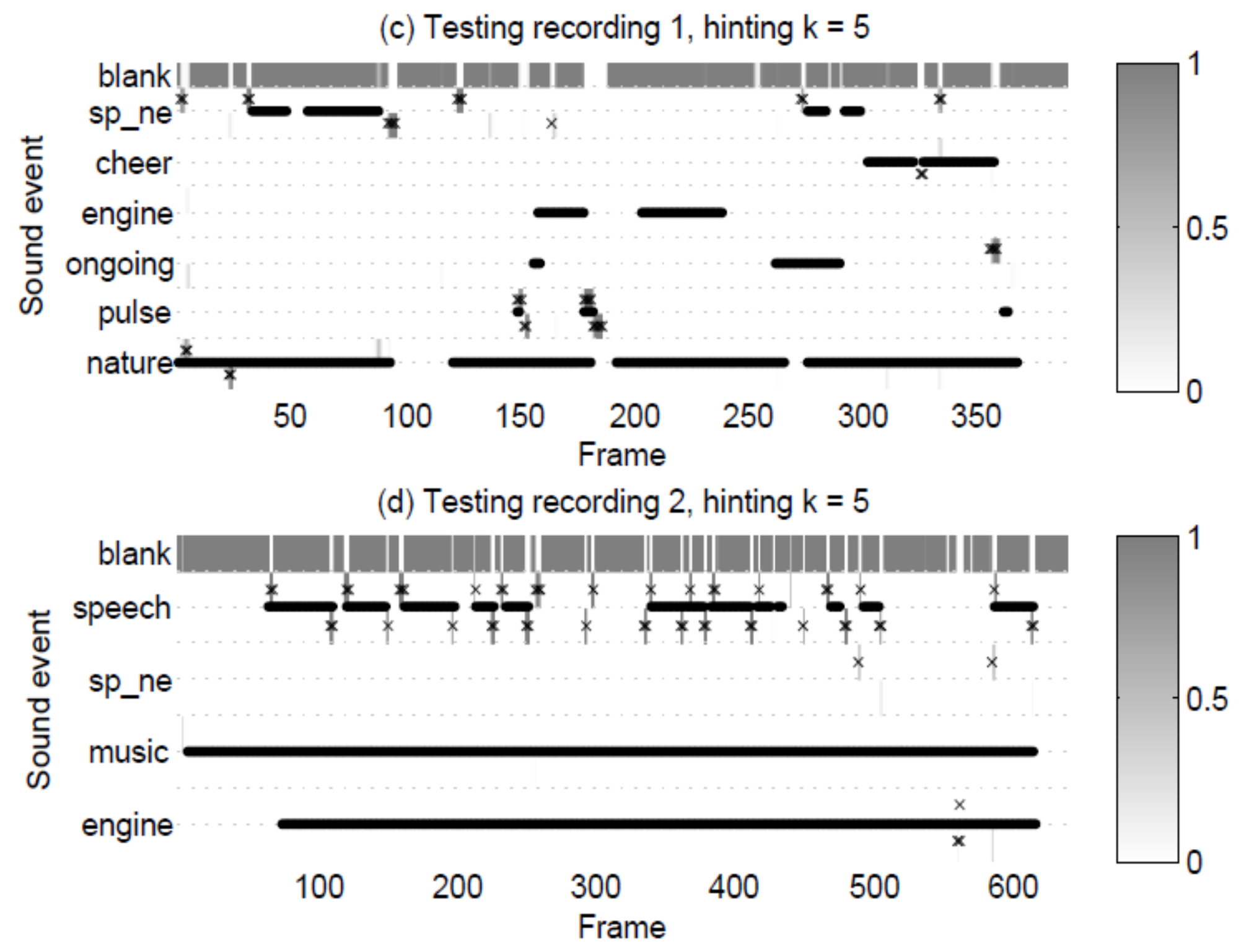


Legend:

- Thick horizontal lines: ground truth events
- Shades: Predicted token probabilities (above line: start; below line: end)
- Crosses: Framewise argmax of token prob.

On test data:

- Speech segments are well detected
    - Notably, speech and non-English speech can be distinguished
- Some short events (*e.g.* pulse) are detected
- Many long events are missed



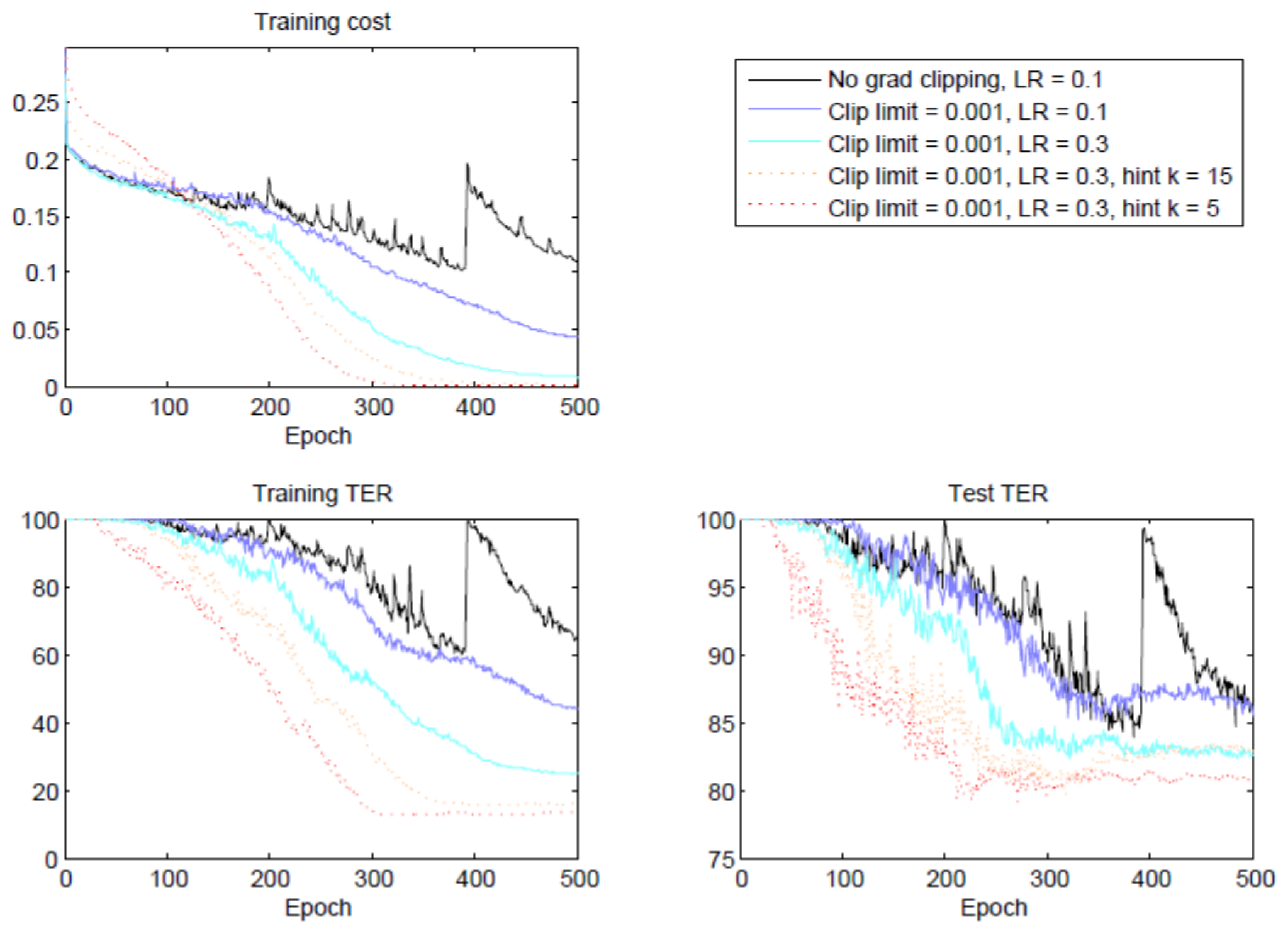
Quantitative Evaluation

Decoding and evaluation:

- Best path decoding (no prefix search)
- Evaluation metric: Token error rate (TER)

Observations:

- Gradient clipping avoids surges and allows a larger learning rate
- Alignment hinting speeds up convergence
- Overfitting (training TER 13%, test 81%)



Conclusion

CTC network for sound event detection:

- Relaxes the need for exact annotation of event boundaries
- Can detect short, transient sound events, which are conventionally hard

Lots of problems to solve:

- Poor generalization to test data
- Alignment hinting necessary

Solutions?

- No data is like more data
    - Hand-labeling, data augmentation
- Regularization

Prospect: Use SoundNet [3] as a feature extractor

- Transfer learning: predict visual objects and scenes from audio
- Big data: trained on 1 year of Flickr videos
- Going deep: 5 layers of convolution