Perceptual debugging of speech synthesis

Using speech from the future to pinpoint the design decisions that matter most

Gustav Eje Henter

ghe@kth.se

Department of Speech, Music and Hearing,
KTH Royal Institute of Technology, Stockholm, Sweden

2018-06-27

Synopsis

- To efficiently improve speech technology, we need to know which parts that work well, and which parts that are flawed

Synopsis

- To efficiently improve speech technology, we need to know which parts that work well, and which parts that are flawed
- *Perceptual debugging* provides the answers, e.g.,
  - Why DNN-based synthesis beats HMM-based synthesis
  - What major bottlenecks affect statistical parametric speech synthesis

Synopsis

- To efficiently improve speech technology, we need to know which parts that work well, and which parts that are flawed
- *Perceptual debugging* provides the answers, e.g.,
  - Why DNN-based synthesis beats HMM-based synthesis
  - What major bottlenecks affect statistical parametric speech synthesis
- Methodological innovation: Manipulating *repeated speech*
  - We can listen to synthesisers that do not exist yet(!)
  - This can answer questions previously considered unanswerable

Overview

Overview

Design choices in speech synthesis

Practical statistical speech synthesis involves many design decisions:

- Training corpus
- Text processing
- Speech parameter representation (waveform-level or vocoder, MGCs, etc.)
- Probabilistic models
- Output generation method

Inappropriate design choices degrade (for example) output naturalness

- Statistical parametric speech synthesis (SPSS) is not convincingly natural

Perceptual debugging

"Isolating the perceptual effects of design decisions in speech synthesis"

Perceptual debugging

"Isolating the perceptual effects of design decisions in speech synthesis"

- Software debugging: Tying behavioural flaws to the code segments that cause them
  - Finding faults is necessary to eliminate them

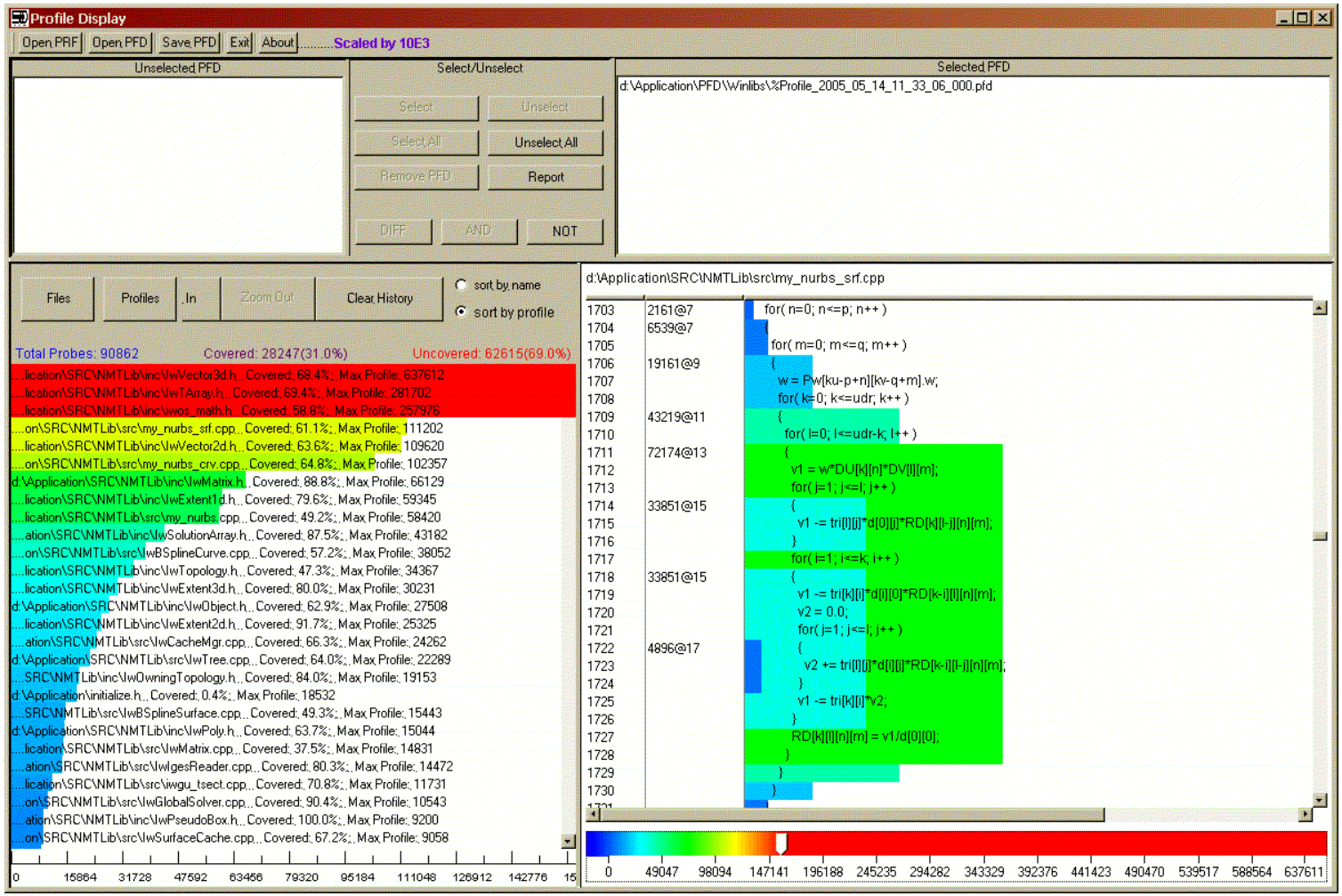
Perceptual debugging

"Isolating the perceptual effects of design decisions in speech synthesis"

- Software debugging: Tying behavioural flaws to the code segments that cause them
  - Finding faults is necessary to eliminate them
- Perceptual debugging: Tying perceptual flaws to the design decisions that cause them
  - Fault-finding for model assumptions rather than code

Similar to software profiling

Highlight hotspots of poor performance in design decisions, not code



Why perceptual debugging?

- Guide speech synthesis research and development
  - Which innovations are most important for observed perceptual improvements?
  - Which design aspects limit current synthesis performance?
  - How much better could synthesis get if specific bottlenecks were eliminated?
- Shed light on human perception
  - Which properties of speech do humans attend to?

Great perceptual debugging is. . .

- Quantitative, comprehensive, and precise
  - Numerically quantifies degradation severities
  - Can partition the entire gap between natural and synthetic speech, attributing each piece to a specific design decision

Great perceptual debugging is. . .

- Quantitative, comprehensive, and precise
  - Numerically quantifies degradation severities
  - Can partition the entire gap between natural and synthetic speech, attributing each piece to a specific design decision
- Different from correlating human judgements and low-level signal properties (e.g., correlating naturalness and global variance)
  - Perceptual debugging finds problems not with the signal, but with the design
  - Perceptual debugging establishes causation, not just correlation

Two debugging approaches

1. Listen to existing synthesisers
    - Build and compare synthesisers with minimal differences
        - Special cases: Exploring enhancements; ablation analysis
    - Traditional fault-finding approach
    - Constructive lower bound on performance

Two debugging approaches

1. Listen to existing synthesisers
   - Build and compare synthesisers with minimal differences
     - Special cases: Exploring enhancements; ablation analysis
   - Traditional fault-finding approach
   - Constructive lower bound on performance
2. Simulate hypothetical future synthesisers
   - Possible to do using repeated speech
   - Novel, model-free approach!
   - Non-constructive *upper* bounds on performance

Two debugging approaches

1. Listen to existing synthesisers
   - Build and compare synthesisers with minimal differences
     - Special cases: Exploring enhancements; ablation analysis
   - Traditional fault-finding approach
   - Constructive lower bound on performance
2. Simulate hypothetical future synthesisers
   - Possible to do using repeated speech
   - Novel, model-free approach!
   - Non-constructive *upper* bounds on performance

We will look at an example of each approach in turn

Overview

Co-author credit

This section is based on:

Watts, O., Henter, G. E., Merritt, T., Wu, Z., and King, S. (2016).

From HMMs to DNNs: where do the improvements come from?
In *Proc. ICASSP*, pages 5505–5509

Benefits of perceptual debugging

- Guide speech synthesis research and development
  - Which innovations are most important for observed perceptual improvements?
  - Which design aspects limit current synthesis performance?
  - How much better could synthesis get if specific bottlenecks were eliminated?
- Shed light on human perception
  - Which properties of speech do humans attend to?

Whence the recent improvements?

- DNN-based statistical speech synthesisers are more natural than older HMM/decision-tree systems
  - (Zen et al., 2013; Qian et al., 2014; Wu et al., 2015; Hashimoto et al., 2015)
- Many factors differ between systems
  - It is not just DNNs/RNNs vs. decision trees!

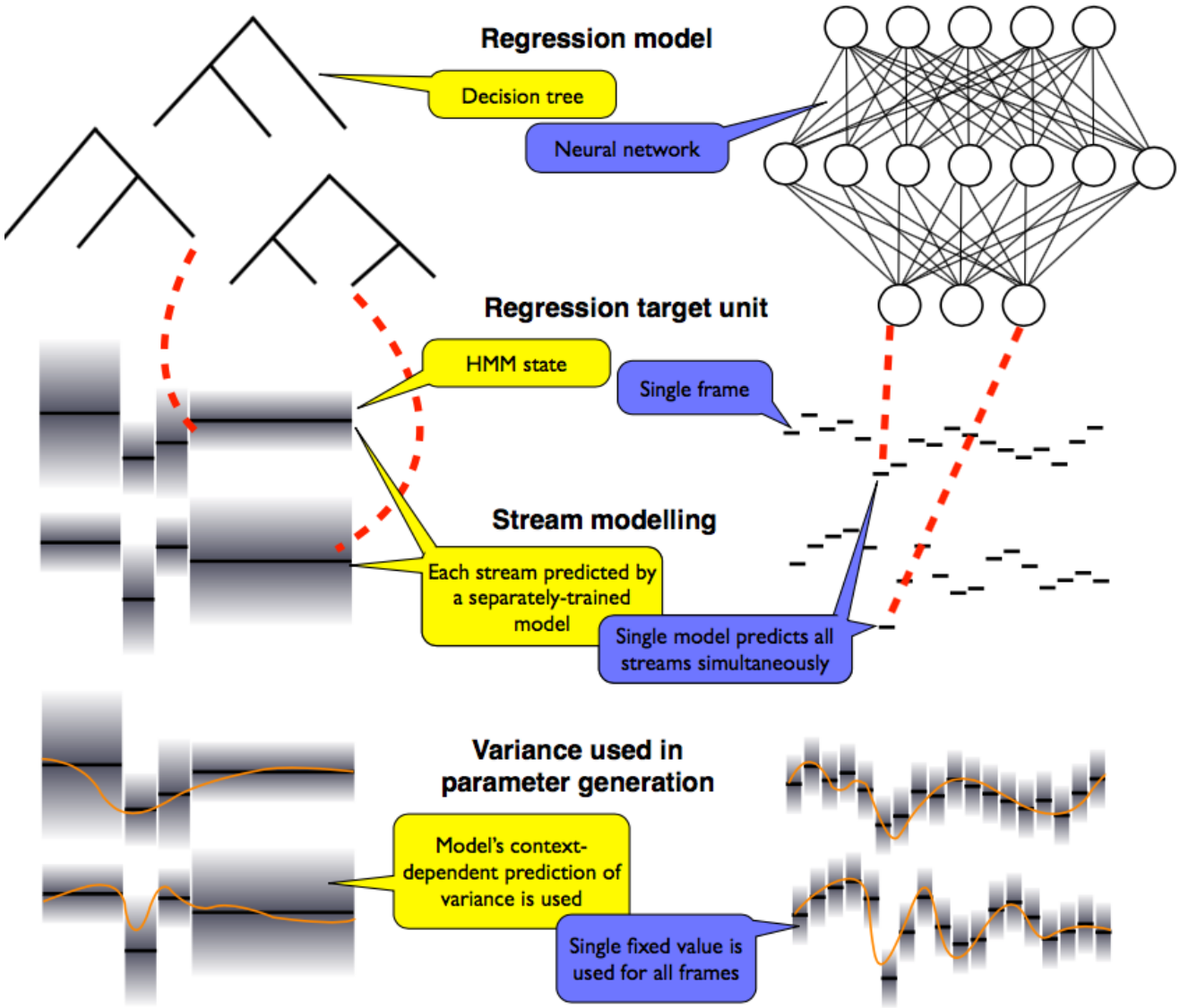
Whence the recent improvements?

- DNN-based statistical speech synthesisers are more natural than older HMM/decision-tree systems
  - (Zen et al., 2013; Qian et al., 2014; Wu et al., 2015; Hashimoto et al., 2015)
- Many factors differ between systems
  - It is not just DNNs/RNNs vs. decision trees!
- Idea: Compare different acoustic models by stepping gradually from HTS (Zen et al., 2007) to Merlin (Wu et al., 2016)
  - Measure output naturalness side by side
  - What are the important factors for the recent improvements?

Differences between HTS and Merlin

- Regression tree or deep/recurrent neural network
  - (Zen et al., 2013; Merritt et al., 2015)
- State-level or frame-level granularity
  - (Zen et al., 2013)
- Parameter streams modelled separately or jointly
  - (Chen et al., 2015)
- Context-dependent or fixed variance
  - (Zen and Senior, 2014)
- Duration-independent or dependent acoustics
  - (Zen et al., 2013; Wu et al., 2015)
- Full or reduced question set
- Multi-space or interpolated F0
  - (Chen and Yu, 2014)

Graphical comparison of HTS and Merlin



Overview

Setup in brief

- **Data:**
  - Hurricane corpus: 2400 studio-recorded (48 kHz) newspaper sentences read by "Nick" (native British male RP speaker)
  - Evaluate on 60 Harvard sentences (6 balanced sets)
- **Features (HTS/Merlin):**
  - 2926/863 binary + 1/9 continuous linguistic features based on Festvox
  - 60×3 MCCs, 25×3 baps, 1×3 log F0 acoustic features based on STRAIGHT (Kawahara, 2006)
  - 5 ms frame shift
- **HTS setup:** 5 sub-phone states, MDL penalty factor 1.0
- **Merlin setup:** 6 FF tanh layers with 1024 units, MDN output

Systems built

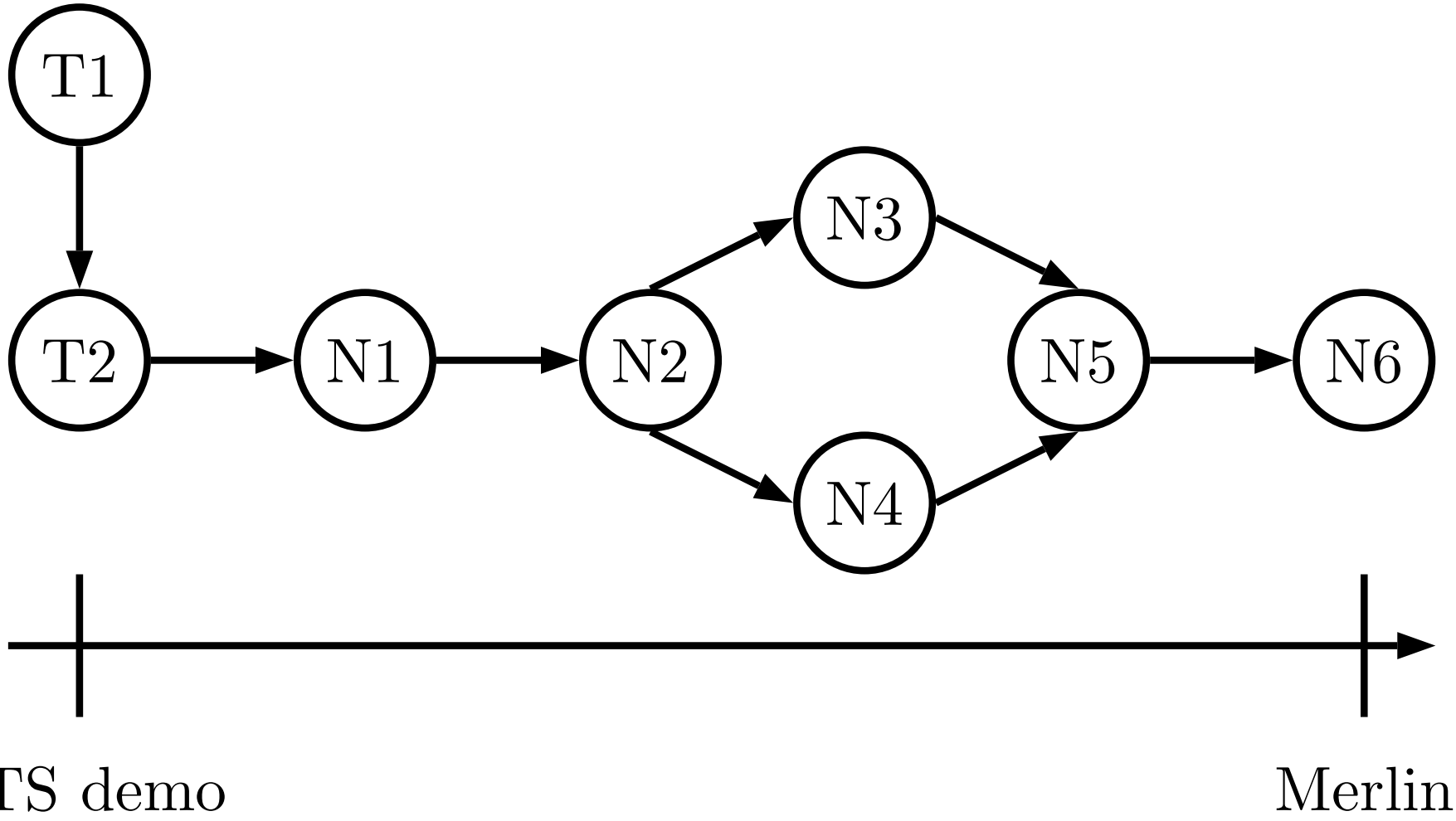
Numerous synthesisers were built with slight variations between each:

ID	Reg.	Gran.	Streams	Var.	Dur. feats.	Enhance
T1	RT	state	sep.	context-dep.	no (1)	GV
T2	RT	state	sep.	context-dep.	no (1)	PF
N1	DNN	state	sep.	context-dep.	no (1)	PF
N2	DNN	state	sep.	fixed	no (1)	PF
N3	DNN	state	joint	fixed	no (1)	PF
N4	DNN	frame	sep.	fixed	no (2)	PF
N5	DNN	frame	joint	fixed	no (2)	PF
N6	DNN	frame	joint	fixed	yes (9)	PF

The listening test also included vocoded natural speech (V)

A spectrum of models

Studied acoustic models step gradually from HTS demo to Merlin



Listening examples

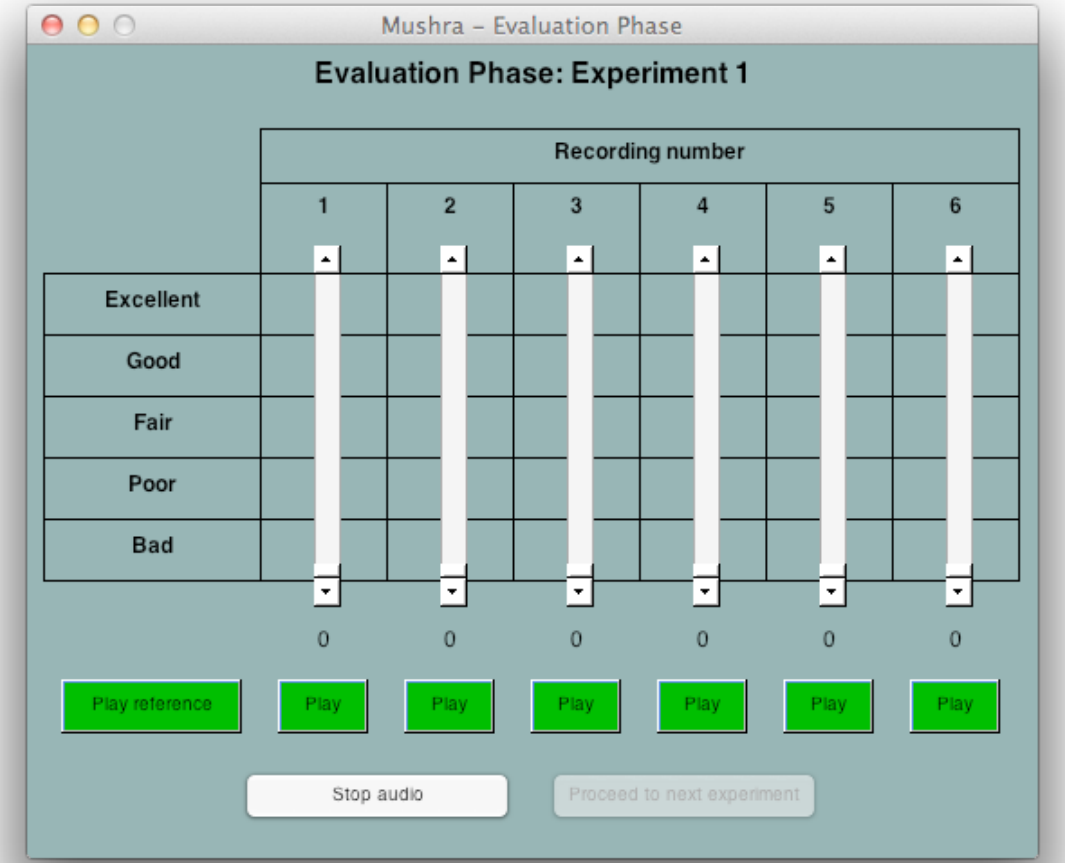
V:	Hvd. 651	Hvd. 652
T1:	Hvd. 651	Hvd. 652
T2:	Hvd. 651	Hvd. 652
N1:	Hvd. 651	Hvd. 652
N2:	Hvd. 651	Hvd. 652
N3:	Hvd. 651	Hvd. 652
N4:	Hvd. 651	Hvd. 652
N5:	Hvd. 651	Hvd. 652
N6:	Hvd. 651	Hvd. 652

(More at dx.doi.org/10.7488/ds/1316)

Naturalness test

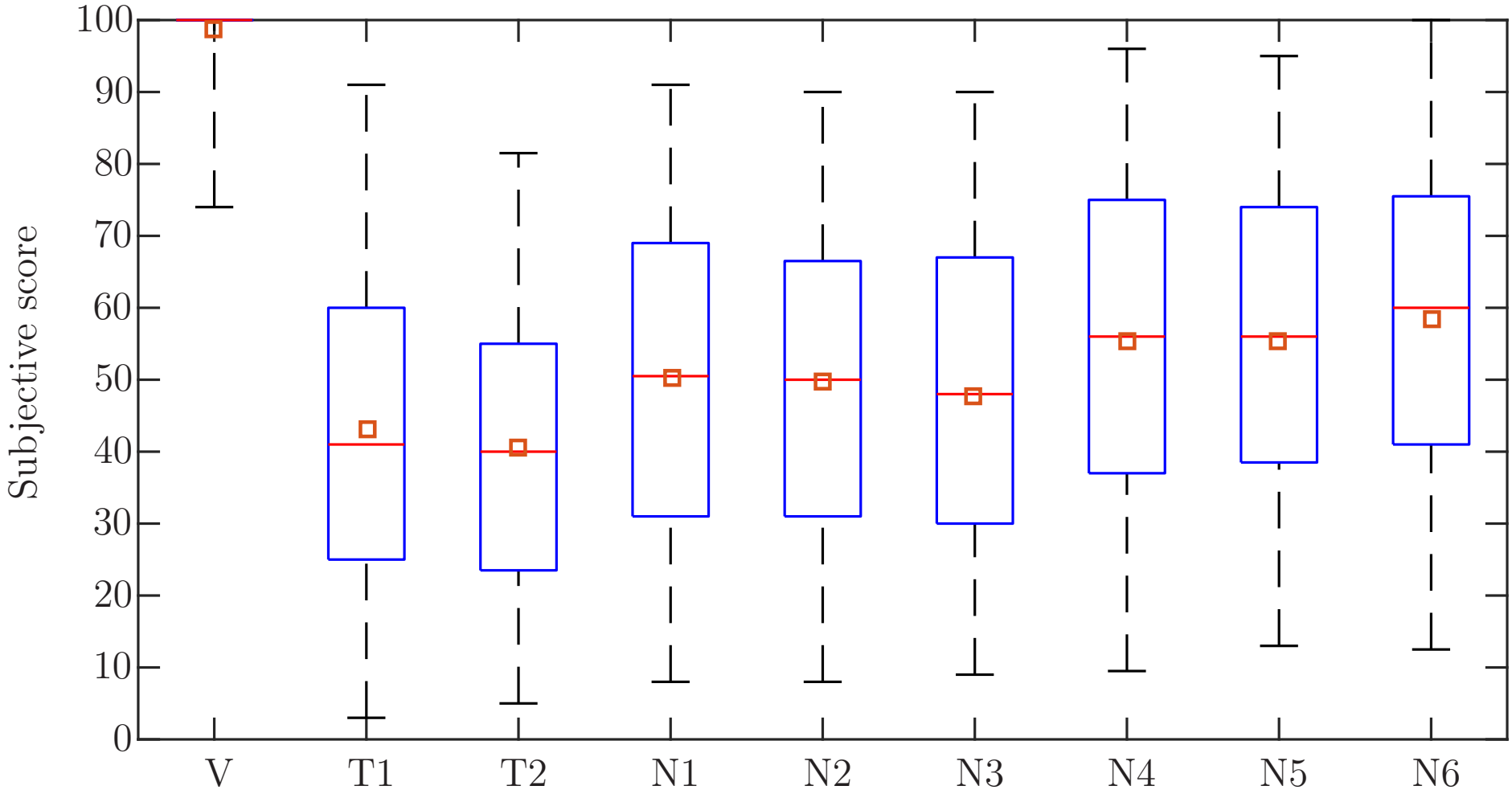
MUSHRA test for parallel, fine-grained naturalness assessment

- ITU standard; better resolution than MOS (Ribeiro et al., 2015)



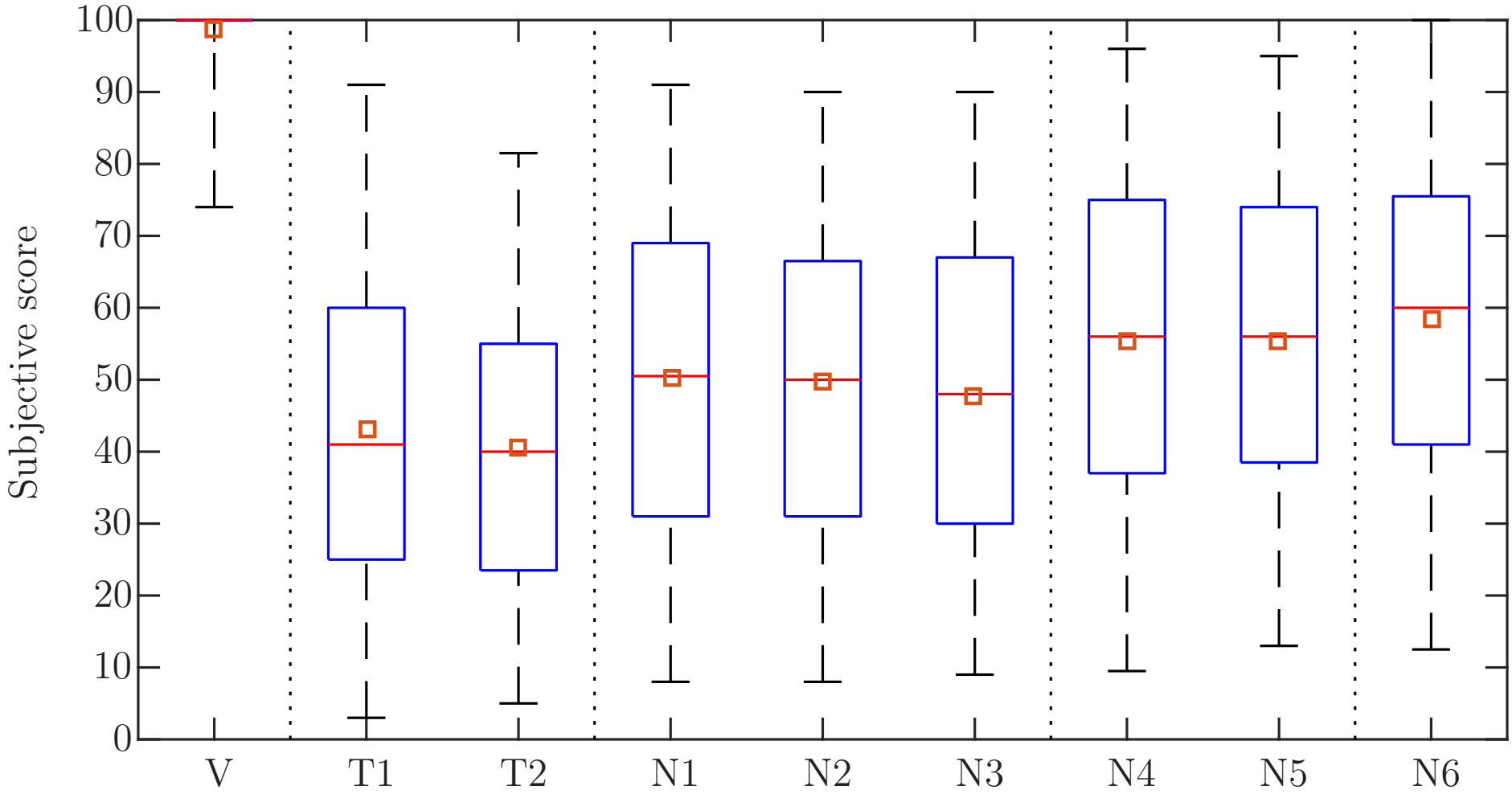
Naturalness results

Box plot of 400 MUSHRA comparisons (20 subjects):



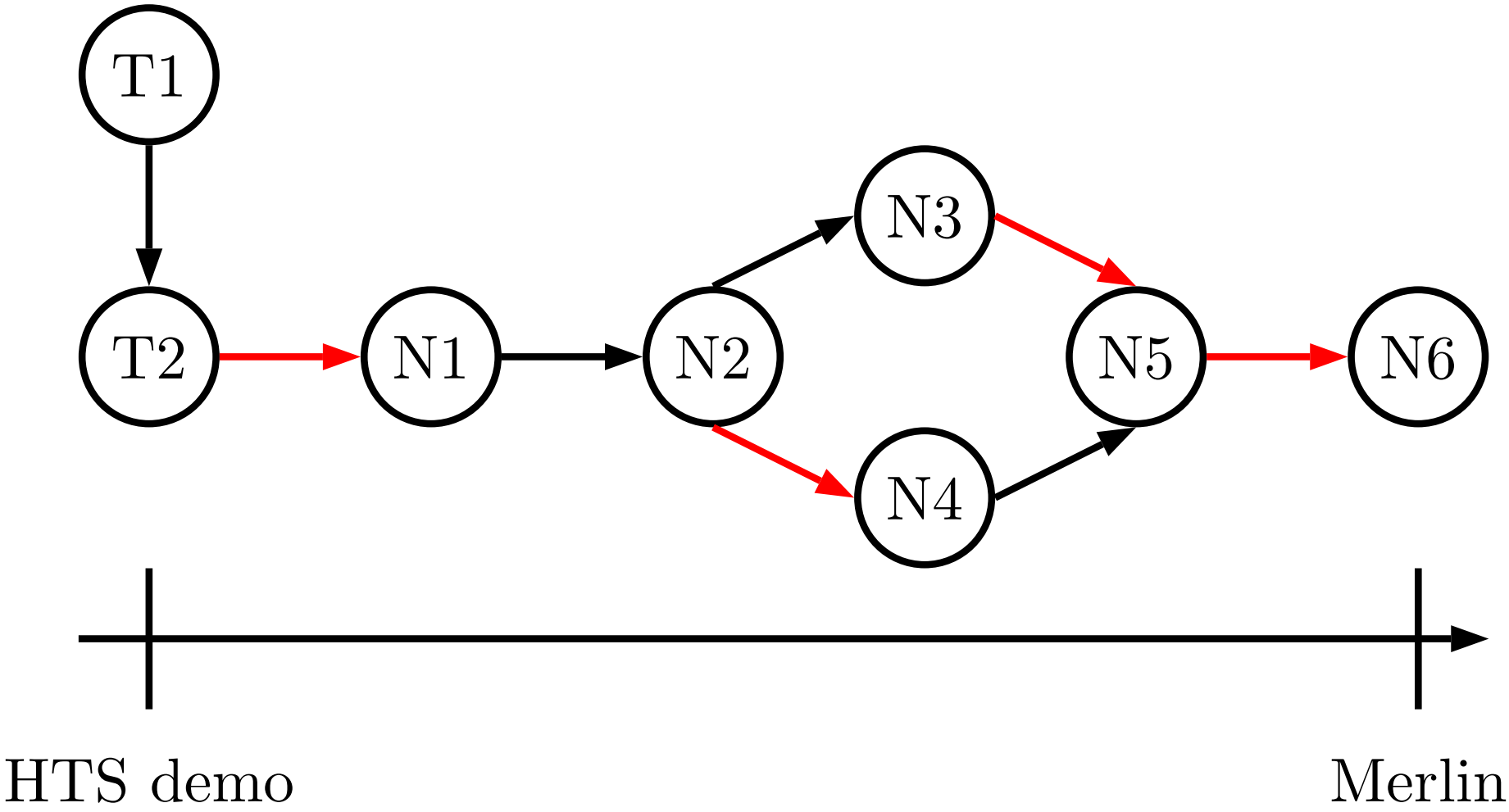
Naturalness results

Pairwise Wilcoxon signed-rank tests ($\alpha = 0.05$) separate five groups



Differences identified

Red arrows highlight significant differences



Findings

Several factors significantly improved rated naturalness:

1. Regression tree $\rightarrow$ deep neural network (big effect)
2. State-level $\rightarrow$ frame-level granularity (big effect)
3. Duration-dependent acoustics (smaller effect)
    - Only evaluated using oracle (not predicted) durations

Other factors did not have a significant effect

- Enhancement method, context-dependent variance, separate or joint stream modelling

Overview

Co-author credit

This section is based on:

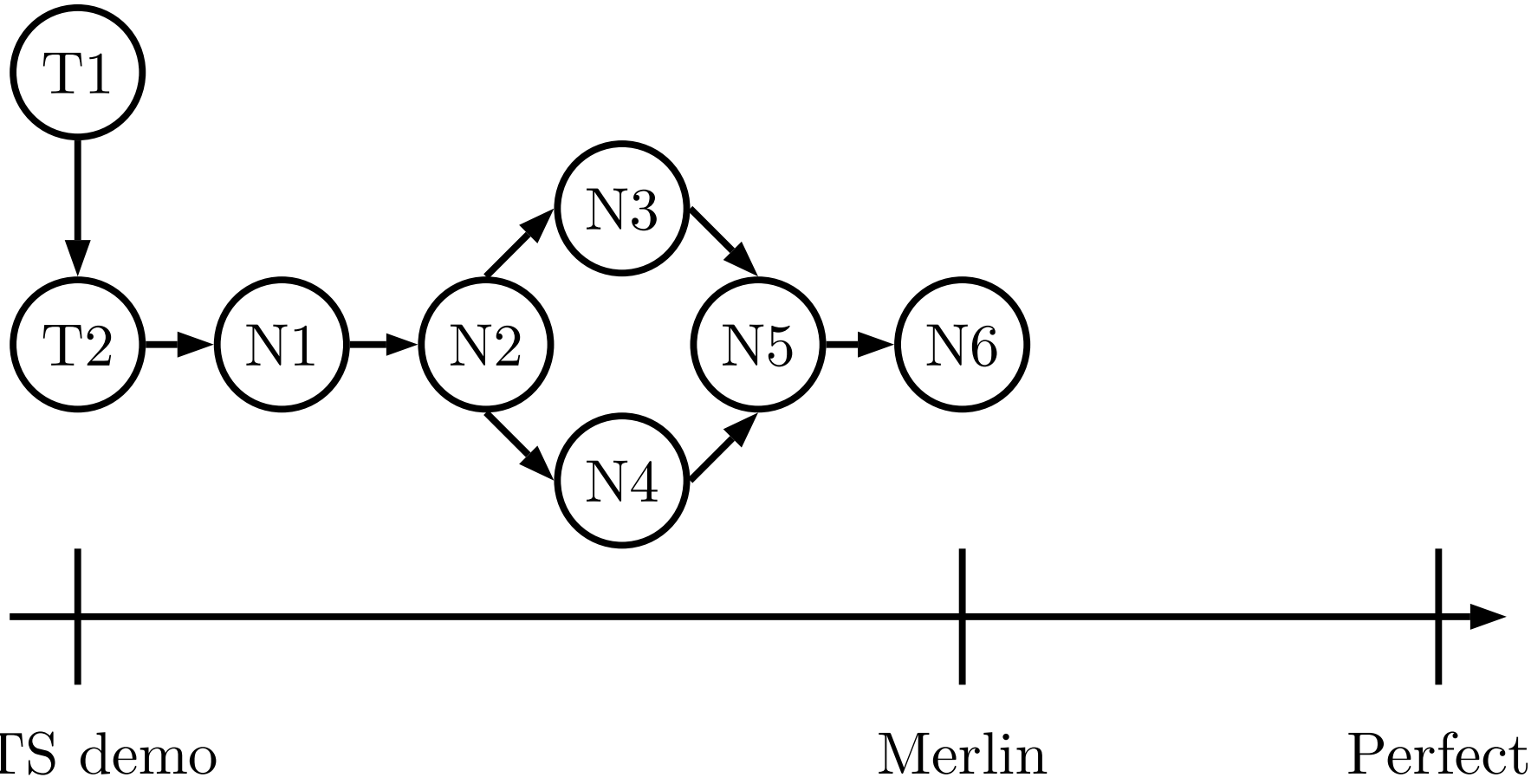
Henter, G. E., Merritt, T., Shannon, M., Mayo, C., and King, S. (2014). Measuring the perceptual effects of modelling assumptions in speech synthesis using stimuli constructed from repeated natural speech.
In *Proc. Interspeech*, pages 1504–1508

Benefits of perceptual debugging

- Guide speech synthesis research and development
  - Which innovations are most important for observed perceptual improvements?
  - Which design aspects limit current synthesis performance?
  - How much better could synthesis get if specific bottlenecks were eliminated?
- Shed light on human perception
  - Which properties of speech do humans attend to?

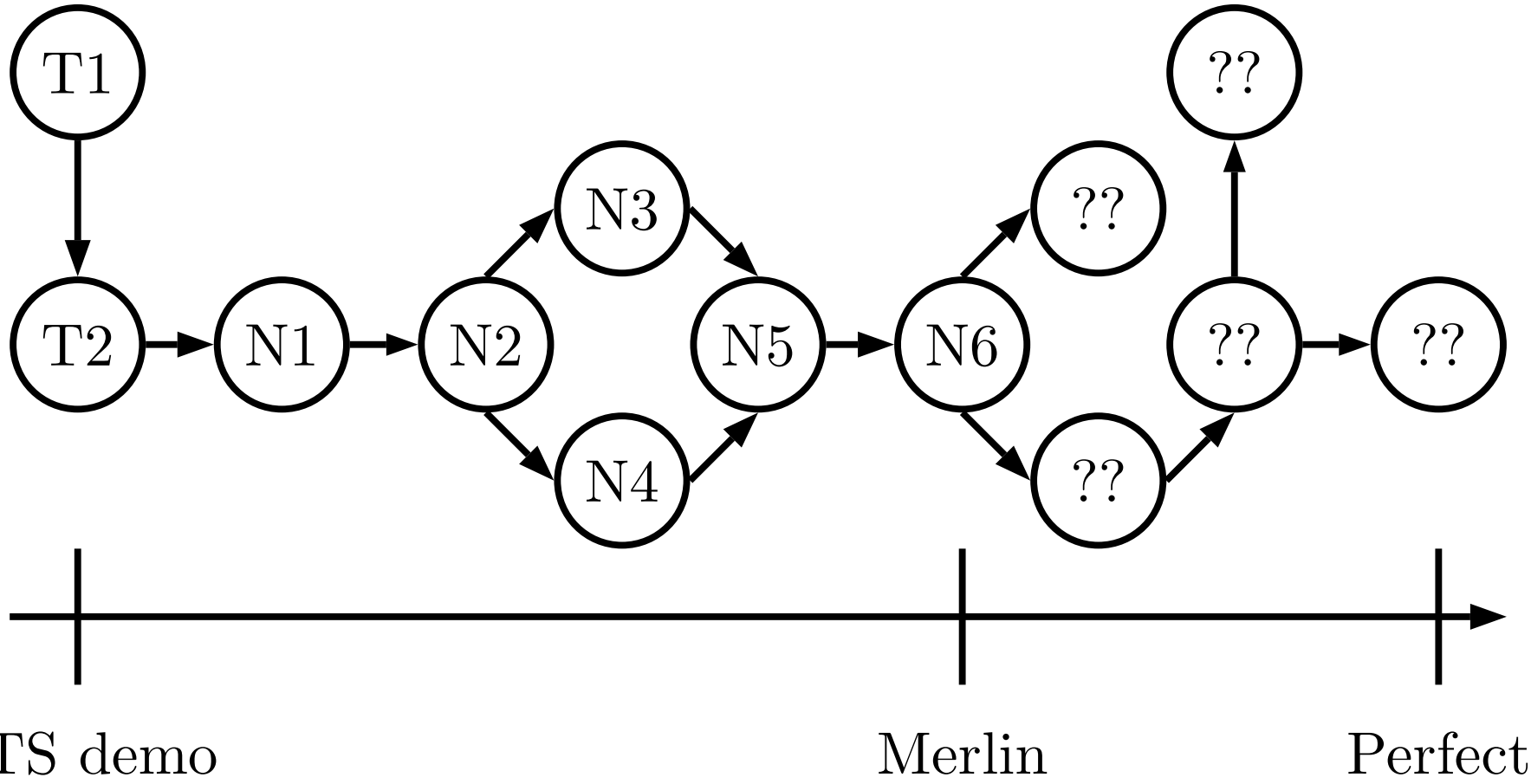
An impossible(?) dream

Wouldn't it be cool to extend the model-stepping approach to evaluate models that are better than those we have right now?



An impossible(?) dream

Wouldn't it be cool to extend the model-stepping approach to evaluate models that are better than those we have right now?



Speech from the future

It turns out that we can listen to speech synthesisers before we build them!

Speech from the future

It turns out that we can listen to speech synthesisers before we build them!

- We can simulate the output of hypothetical, future synthesisers
- Not science fiction, but science *fact!*

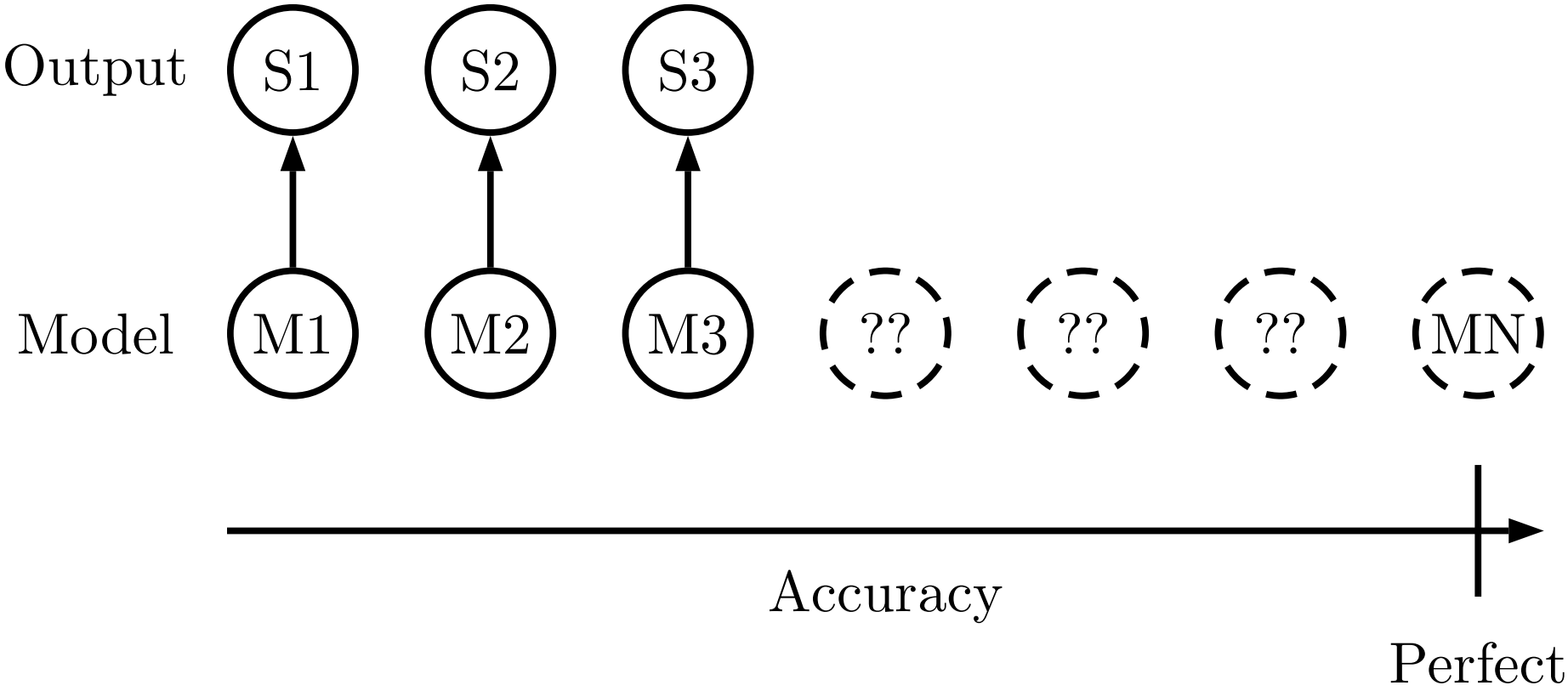
Speech from the future

It turns out that we can listen to speech synthesisers before we build them!

- We can simulate the output of hypothetical, future synthesisers
- Not science fiction, but science *fact!*
- Unlike traditional fault finding, comparisons occur in the context of an otherwise highly-accurate model
  - The perceptual effects of different design choices are no longer masked by other imperfections

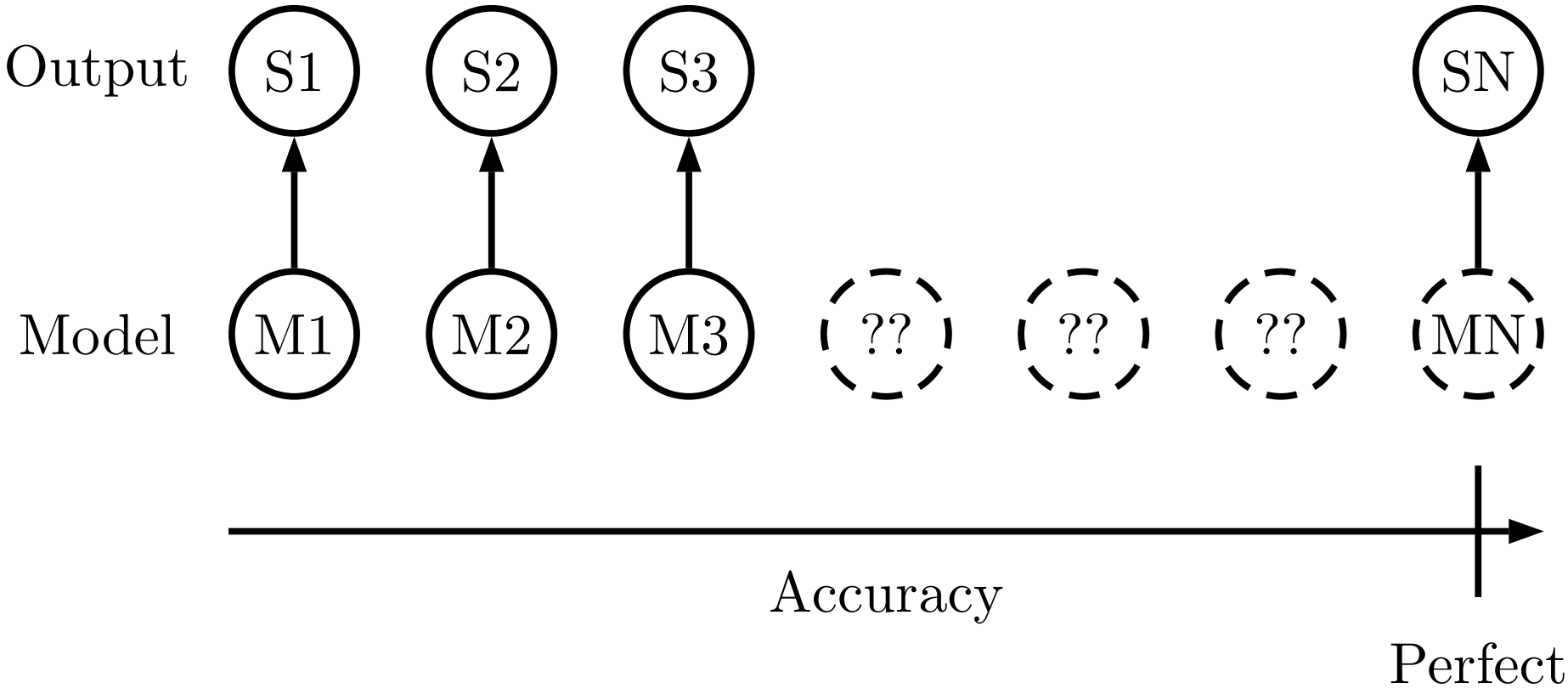
Our insight

Synthesisers are statistical models we can listen to, but there are many better models we do not have access to



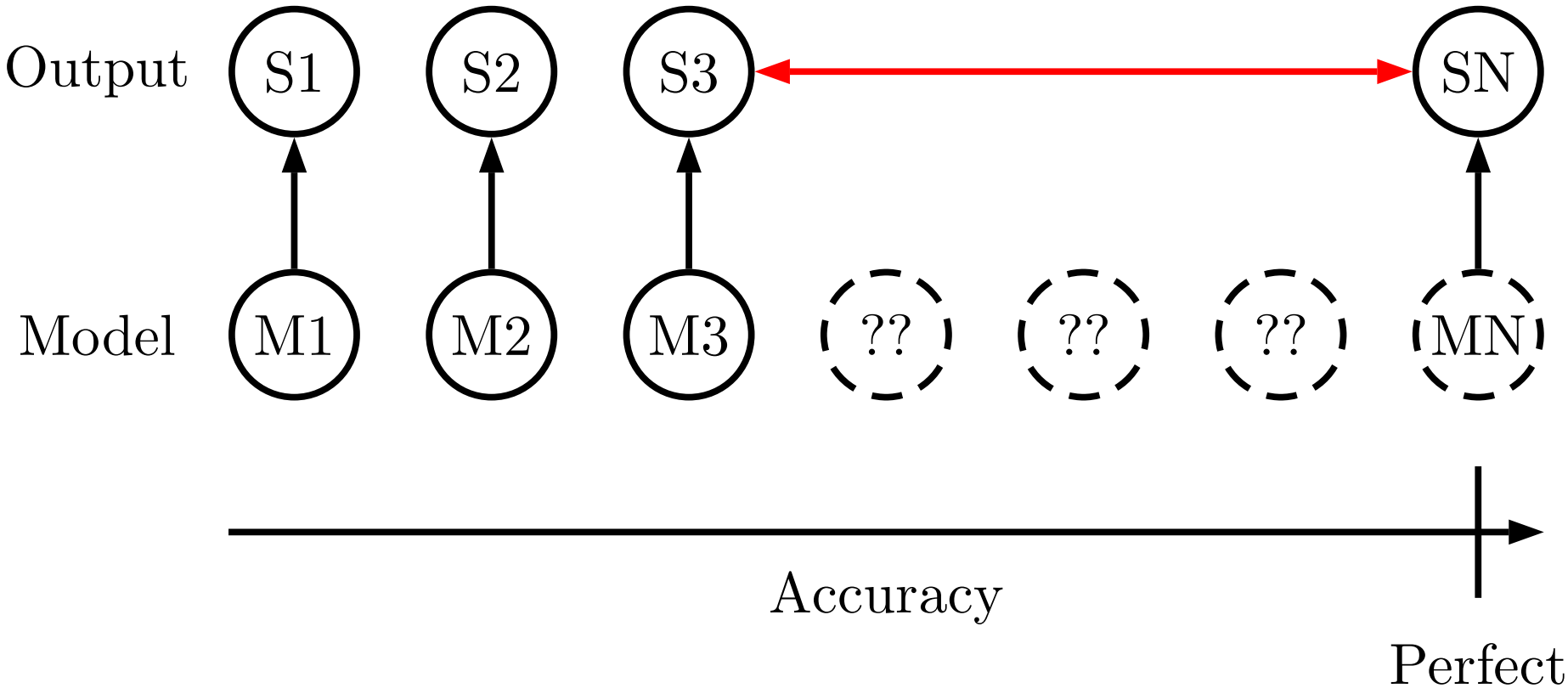
Our insight

We can, however, listen to the output of the true speech model



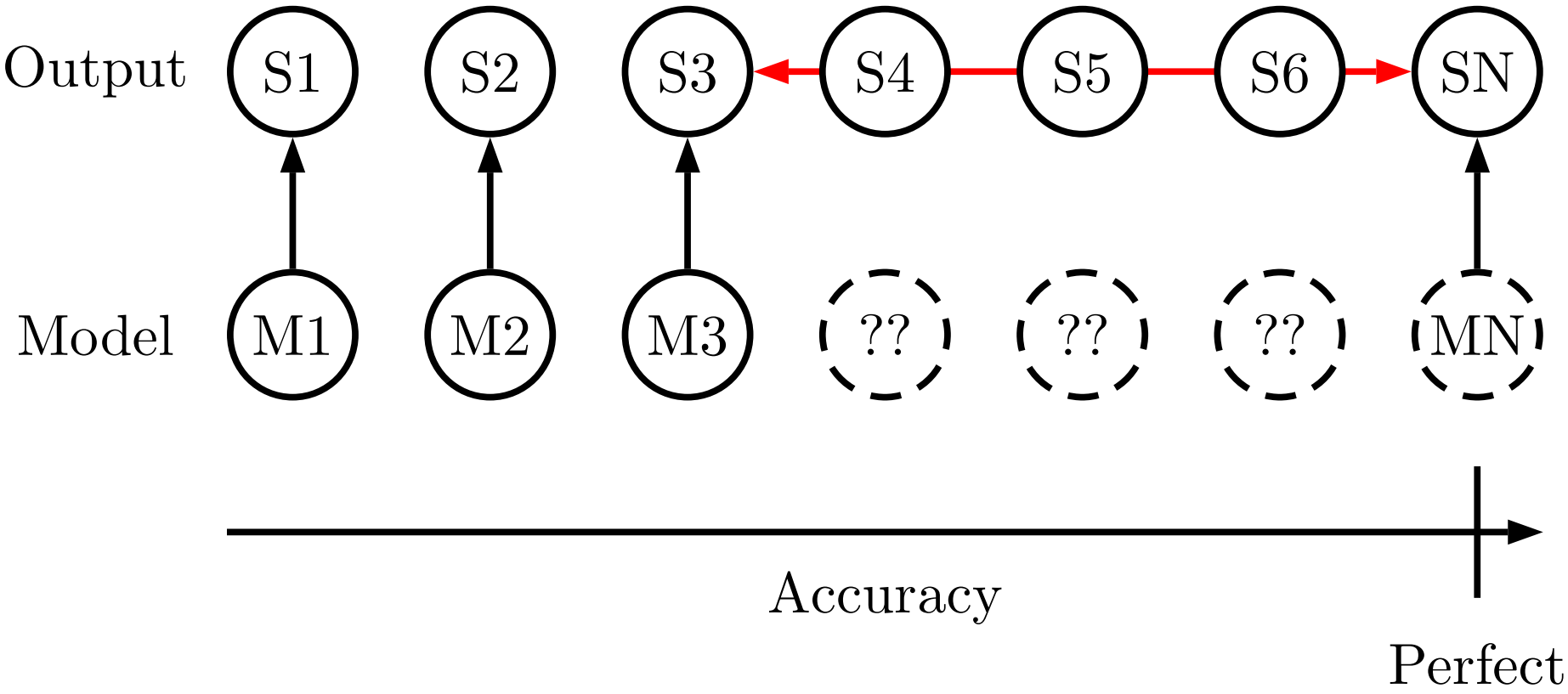
Our insight

This makes it possible to simulate the output of models in between the two extremes



Our insight

This makes it possible to simulate the output of models in between the two extremes



More specifically

- Natural speech is a sample from the true acoustic model

More specifically

- Natural speech is a sample from the true acoustic model
- By manipulating *repeated natural speech* we can simulate output from
  - Better models than those we have today
    - That only incorporate certain, specific modelling assumptions
    - And otherwise are close to perfect

More specifically

- Natural speech is a sample from the true acoustic model
- By manipulating *repeated natural speech* we can simulate output from
  - Better models than those we have today
    - That only incorporate certain, specific modelling assumptions
    - And otherwise are close to perfect
  - Given a particular parameter representation (including vocoder)
  - And a particular output generation method

Overview

Repeated speech

Even when controlling for context, the same text can be realised acoustically in many different ways

Repeated speech

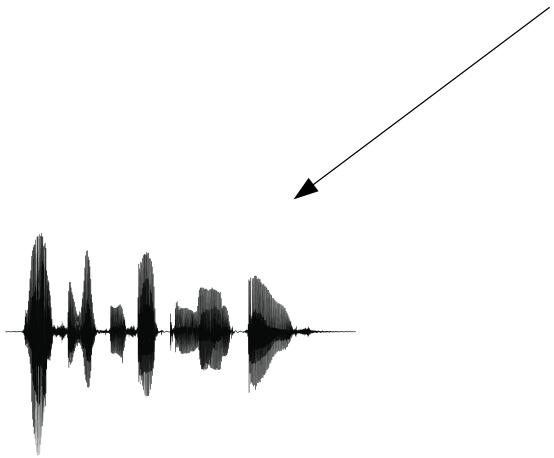
Even when controlling for context, the same text can be realised acoustically in many different ways

"Rice is often served in round bowls"

Repeated speech

Even when controlling for context, the same text can be realised acoustically in many different ways

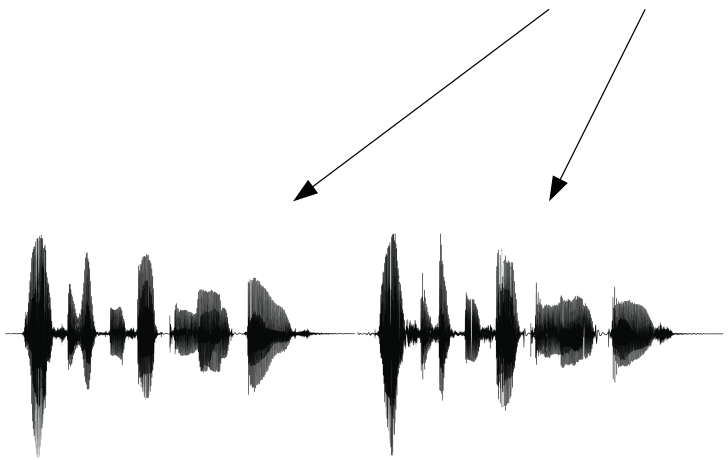
"Rice is often served in round bowls"



Repeated speech

Even when controlling for context, the same text can be realised acoustically in many different ways

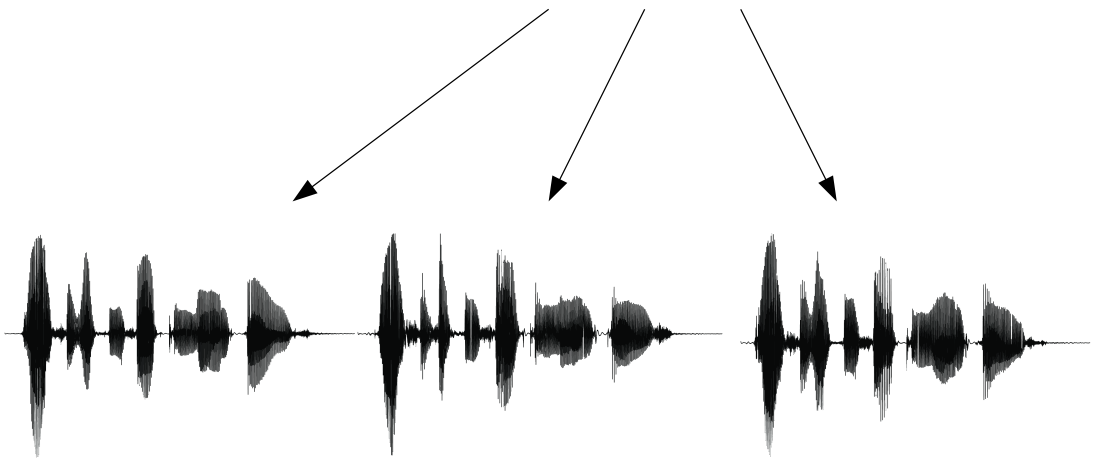
"Rice is often served in round bowls"



Repeated speech

Even when controlling for context, the same text can be realised acoustically in many different ways

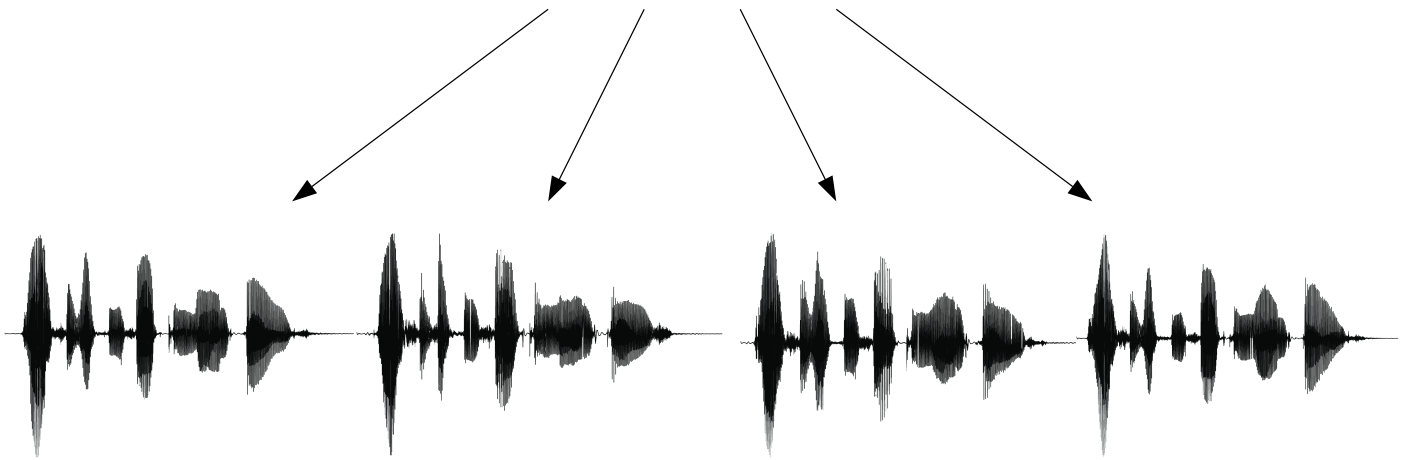
"Rice is often served in round bowls"



Repeated speech

Even when controlling for context, the same text can be realised acoustically in many different ways

"Rice is often served in round bowls"



The value of repeated speech

Repeated speech under carefully controlled conditions reflects the inexorable acoustic variation of natural speech:

- Same talker
- Same text
- Same place
- Same time
- Same equipment
- Same manner of speaking
- . . . just different realisations

The value of repeated speech

Repeated speech under carefully controlled conditions reflects the inexorable acoustic variation of natural speech:

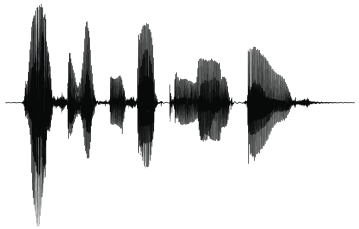
- Same talker
- Same text
- Same place
- Same time
- Same equipment
- Same manner of speaking
- . . . just different realisations

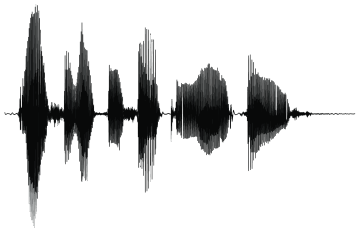
REHASP 0.5 corpus

- "REpeated HArvard Sentence Prompts"
- Female native British English speaker "Lucy"
- 30 Harvard sentence prompts
- Each read aloud 40 times
  - Presented in random order
- Recorded at 16 bit 96 kHz
- Publicly available under a permissive license
  - datashare.is.ed.ac.uk/handle/10283/561

In pictures

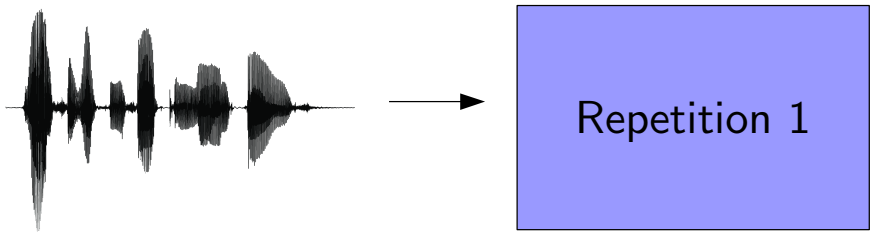
0. Start with natural speech repetitions:





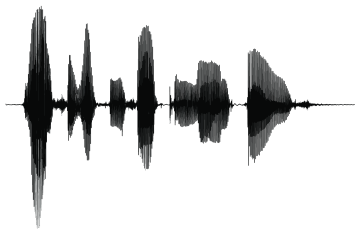
In pictures

1. Extract parameters:



In pictures

1. Extract parameters:



Repetition 1

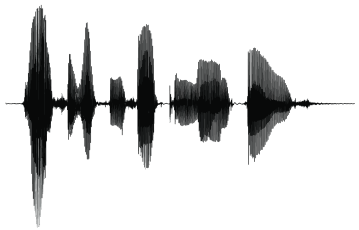
Speech representation

Standard parametric speech representation used for experiments:

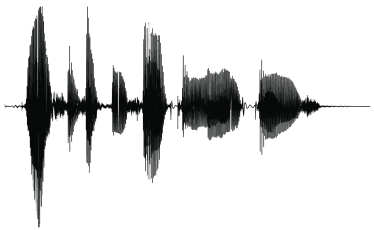
- 16 kHz sampling frequency
- Matlab STRAIGHT for parameter extraction
- 46-dimensional parameter vector with three streams:
  - 40 MCEPs (0–39), representing filter coefficients
  - log F0
  - 5 band aperiodicities (baps)
- 5 ms frame shift

In pictures

1.b. Resynthesise (baseline “V”):

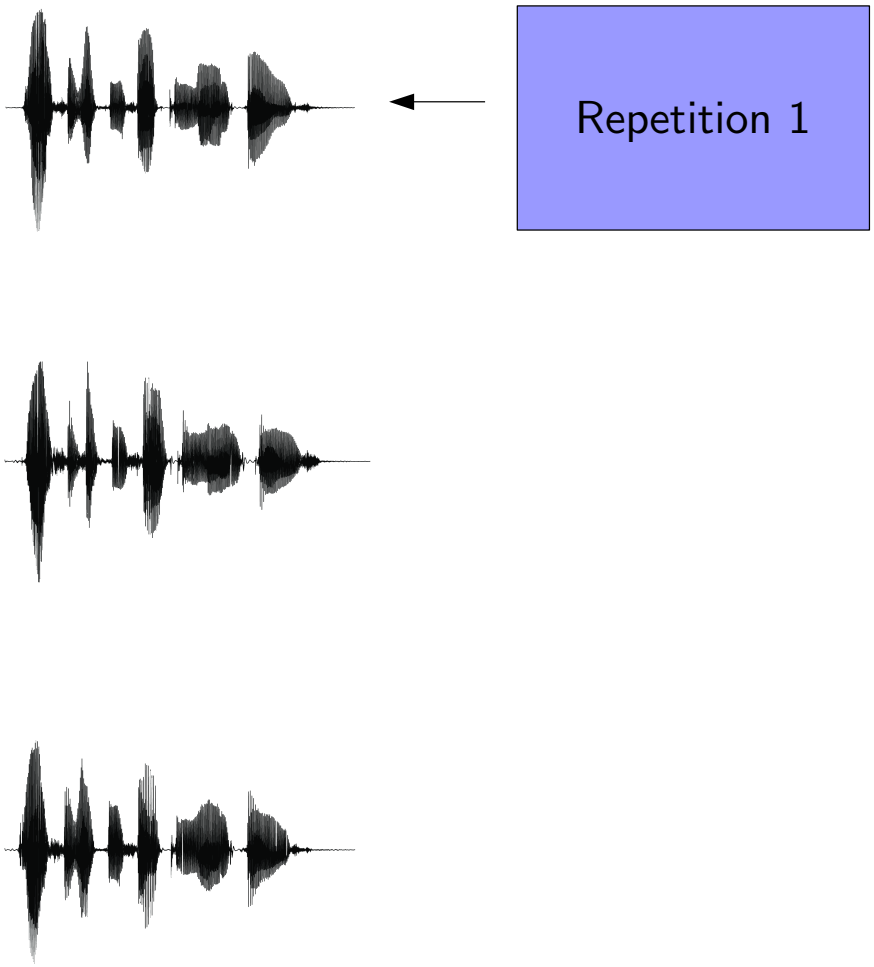


Repetition 1



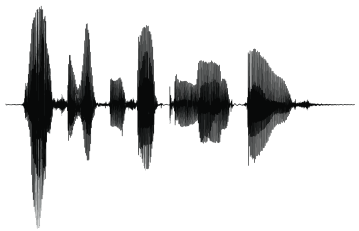
In pictures

1.b. Resynthesise (baseline “V”):

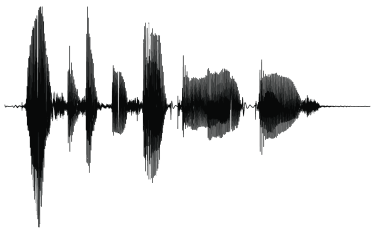


In pictures

1. Extract parameters:

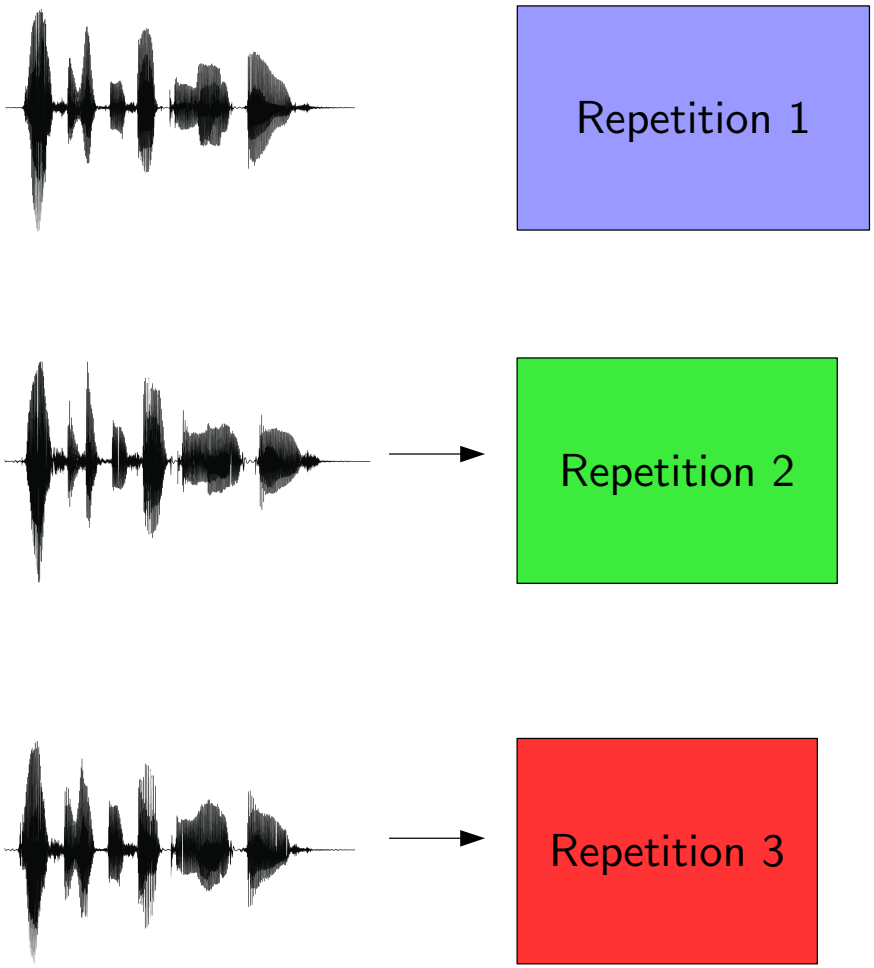


Repetition 1



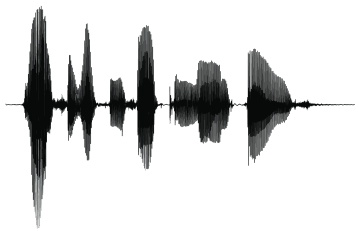
In pictures

1. Extract parameters:

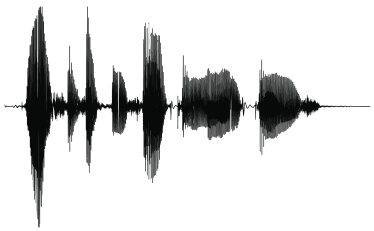


In pictures

1. Extract parameters:



Repetition 1

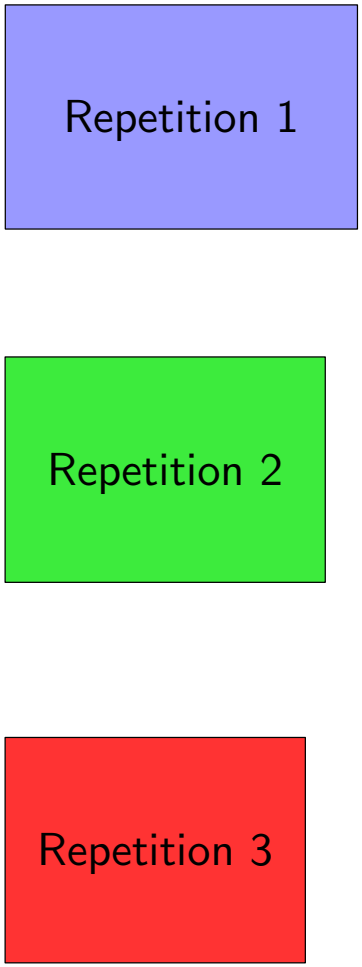


Repetition 2

Repetition 3

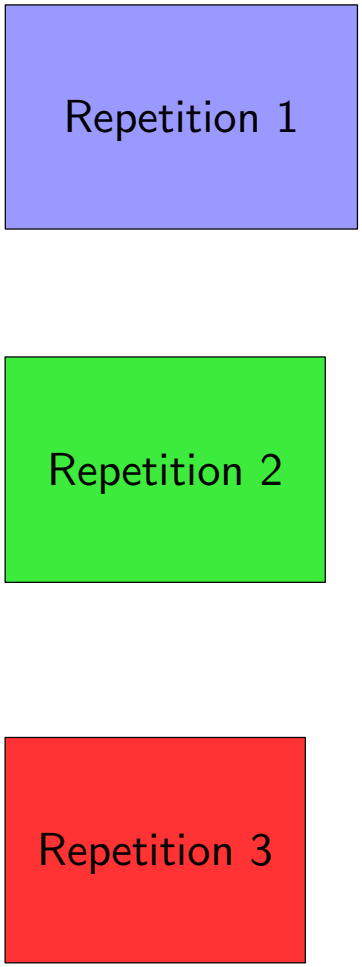
In pictures

1. Extract parameters:



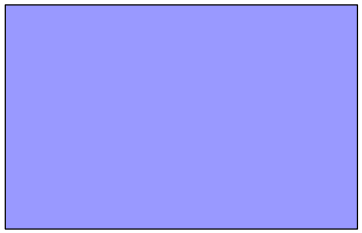
Match timings

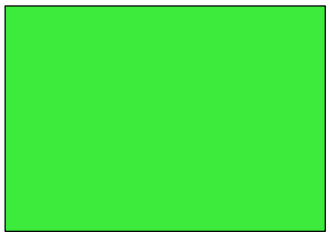
2.a. Match frames:

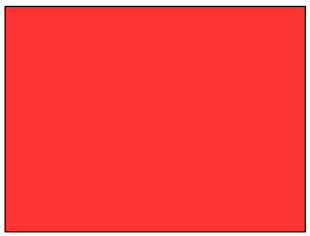


Match timings

2.a. Match frames:

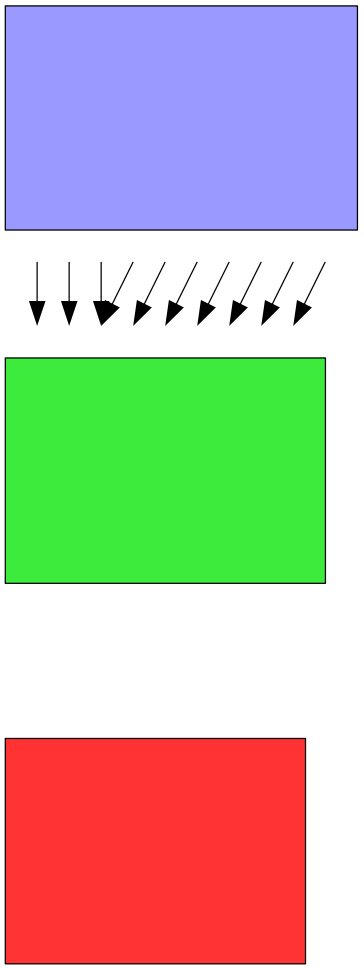






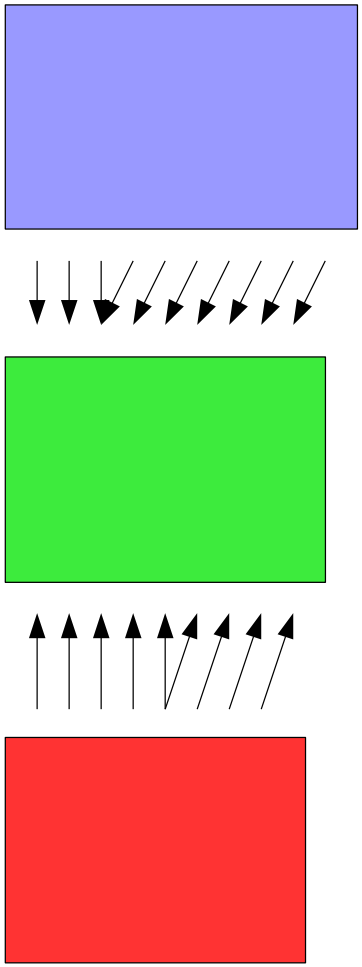
Match timings

2.a. Match frames:



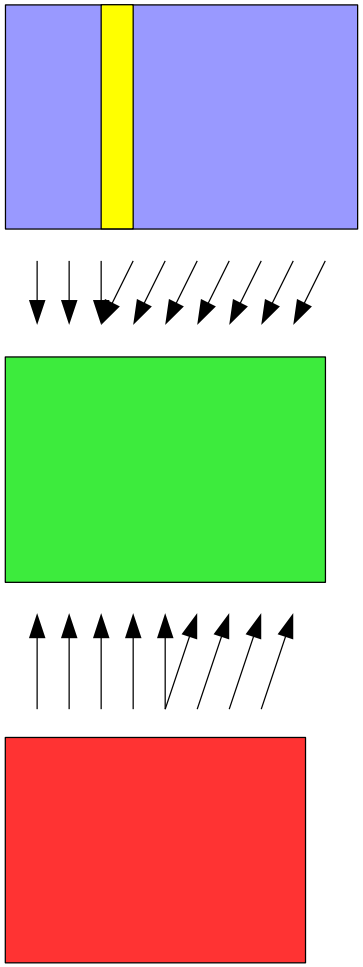
Match timings

2.a. Match frames:



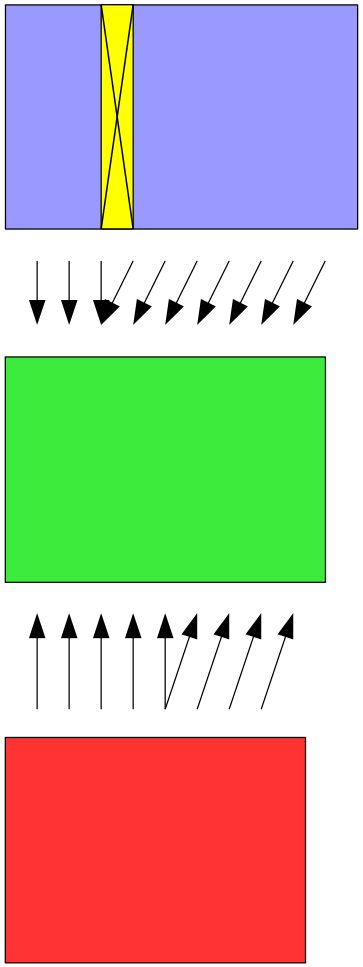
Match timings

2.b. Warp timings:



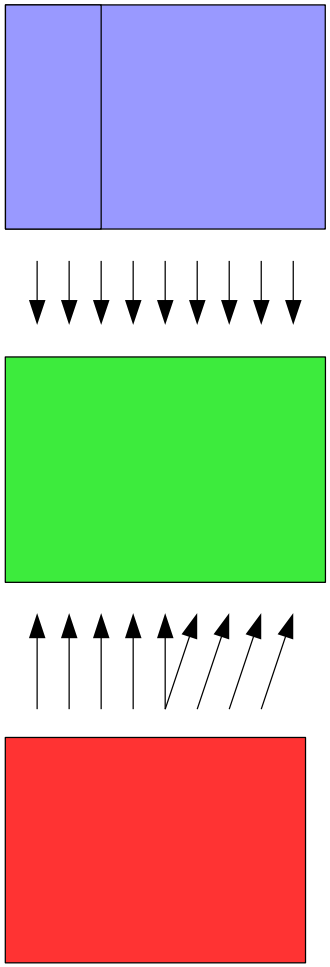
Match timings

2.b. Warp timings:



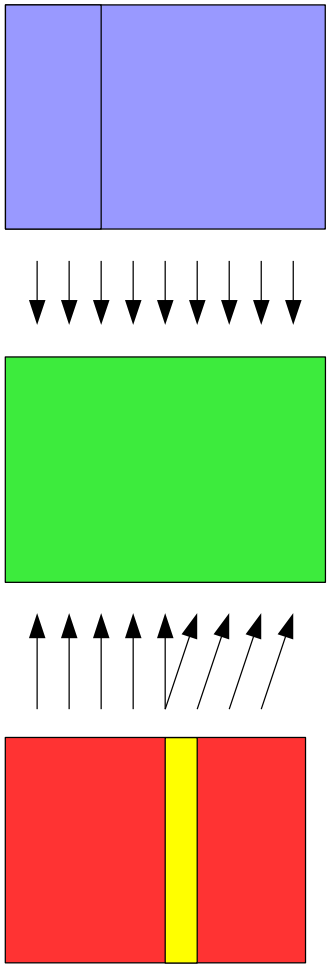
Match timings

2.b. Warp timings:



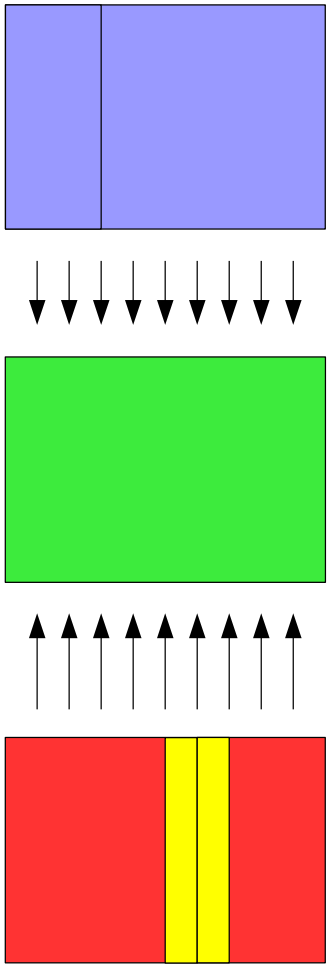
Match timings

2.b. Warp timings:



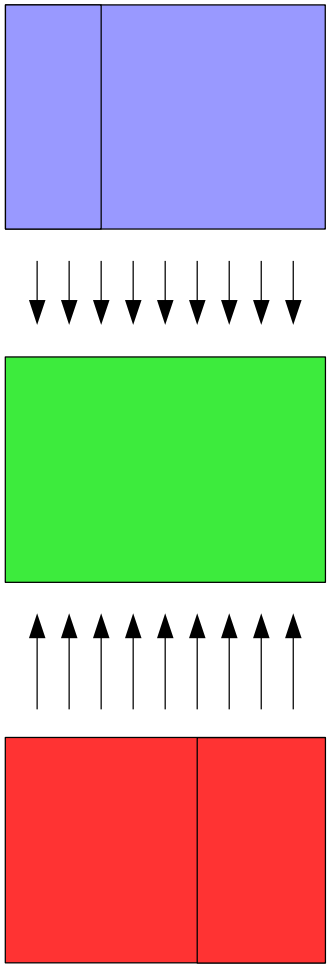
Match timings

2.b. Warp timings:



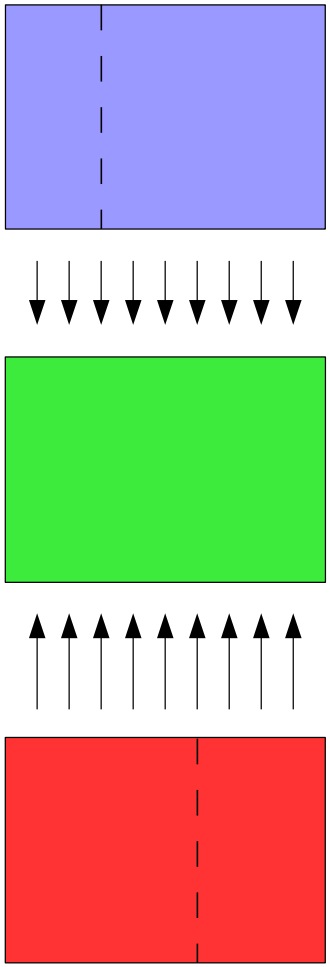
Match timings

2.b. Warp timings:



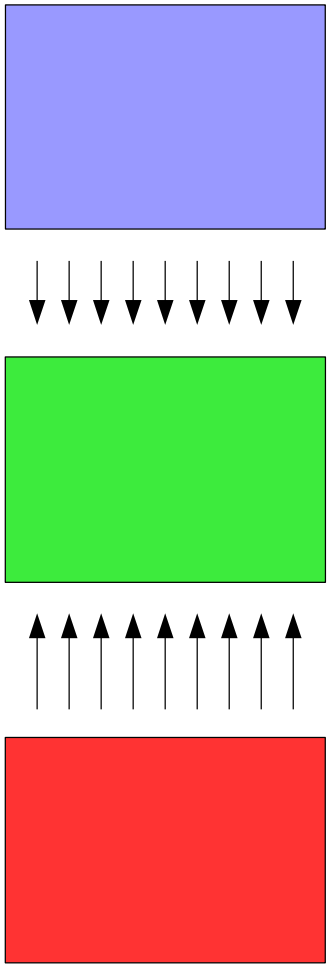
Match timings

2.b. Warp timings:



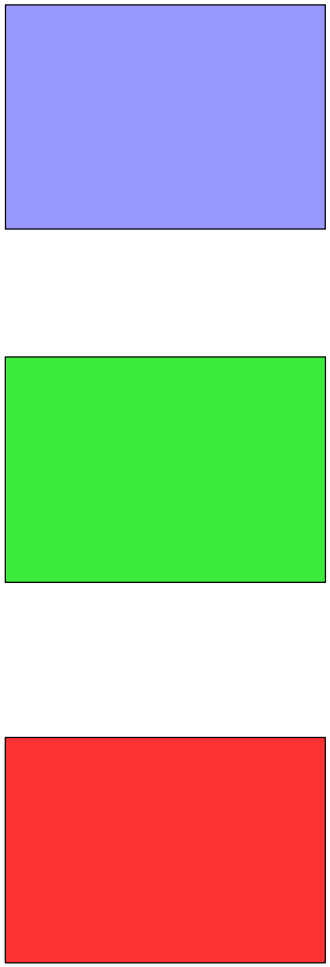
Match timings

2.b. Warp timings:



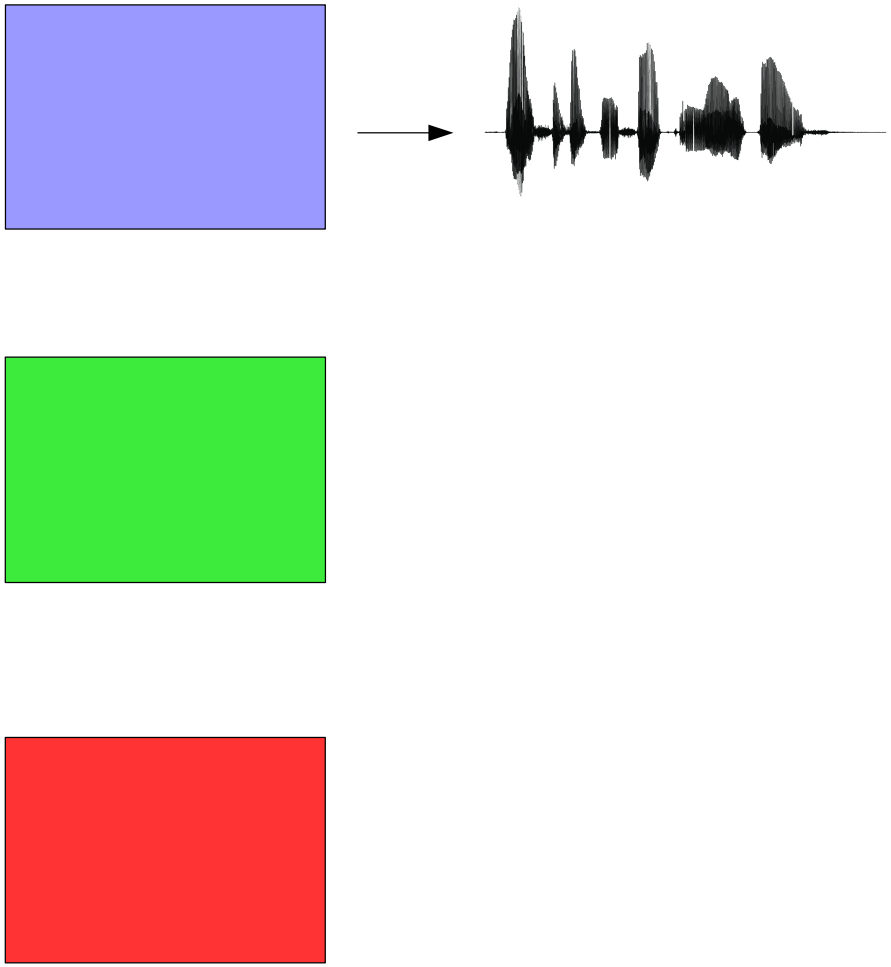
Match timings

2.b. Warp timings:



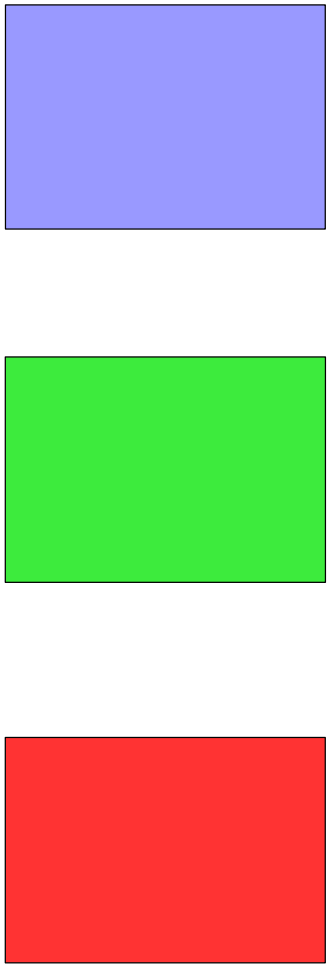
Match timings

2.c. Resynthesise (baseline “D”):



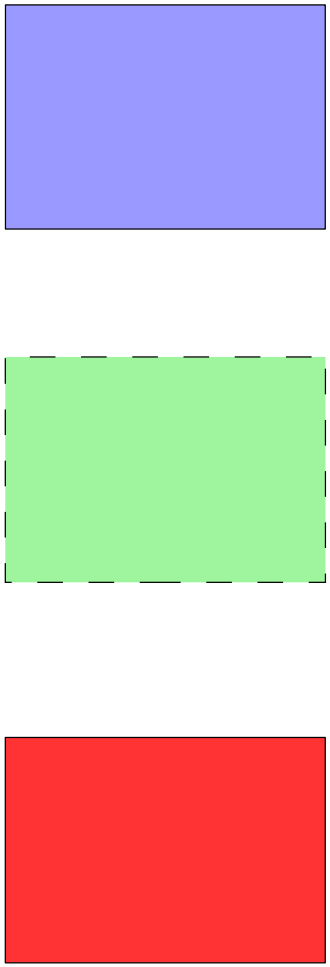
Match timings

2.d. Optionally remove reference:



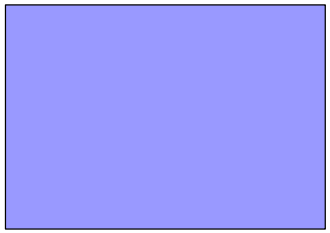
Match timings

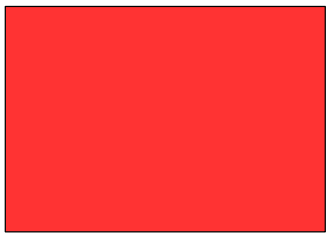
2.d. Optionally remove reference:



Match timings

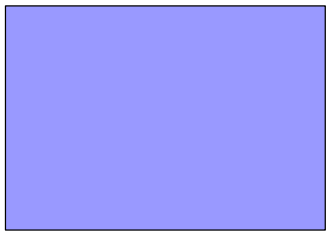
2.d. Optionally remove reference:

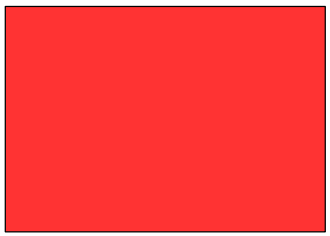




Match timings

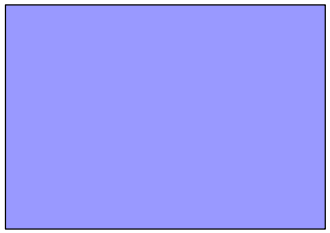
We now have "LEGO pieces" of aligned repetitions

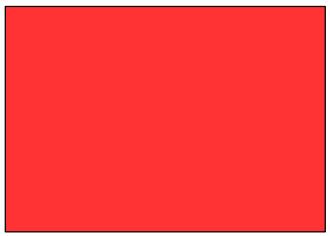




Create chimeric speech

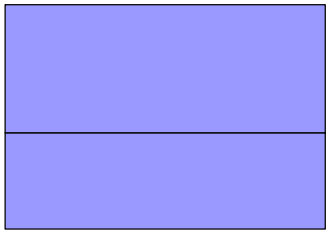
3.a. Combine parameters from independent repetitions:





Create chimeric speech

3.a. Combine parameters from independent repetitions:



Create chimeric speech

3.a. Combine parameters from independent repetitions:

Create chimeric speech

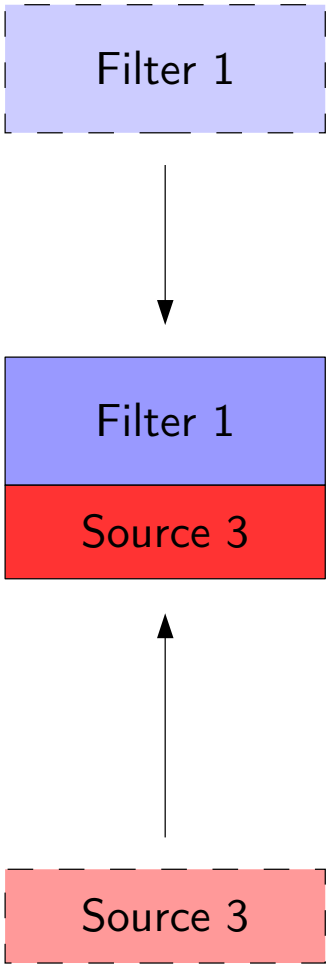
3.a. Combine parameters from independent repetitions:

Create chimeric speech

3.a. Combine parameters from independent repetitions:

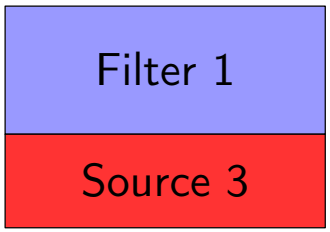
Create chimeric speech

3.a. Combine parameters from independent repetitions:



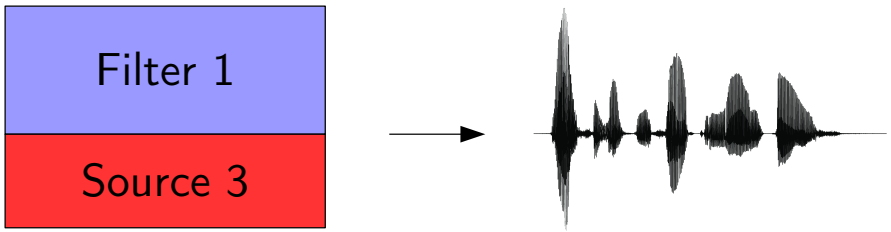
Create chimeric speech

3.a. Combine parameters from independent repetitions:



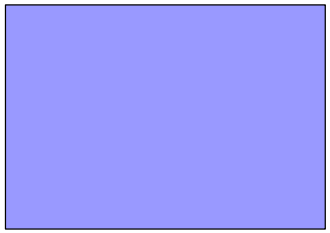
Create chimeric speech

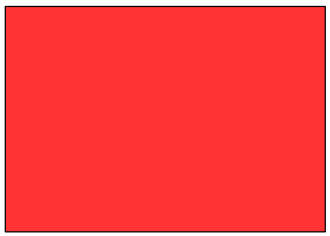
3.a. Resynthesise chimeric speech (here condition “SF”):



Create mean speech

3.b. Take the mean of all repetitions:



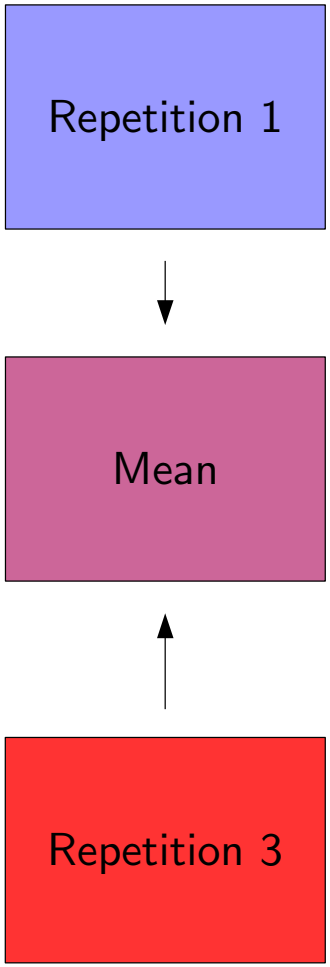


Create mean speech

3.b. Take the mean of all repetitions:

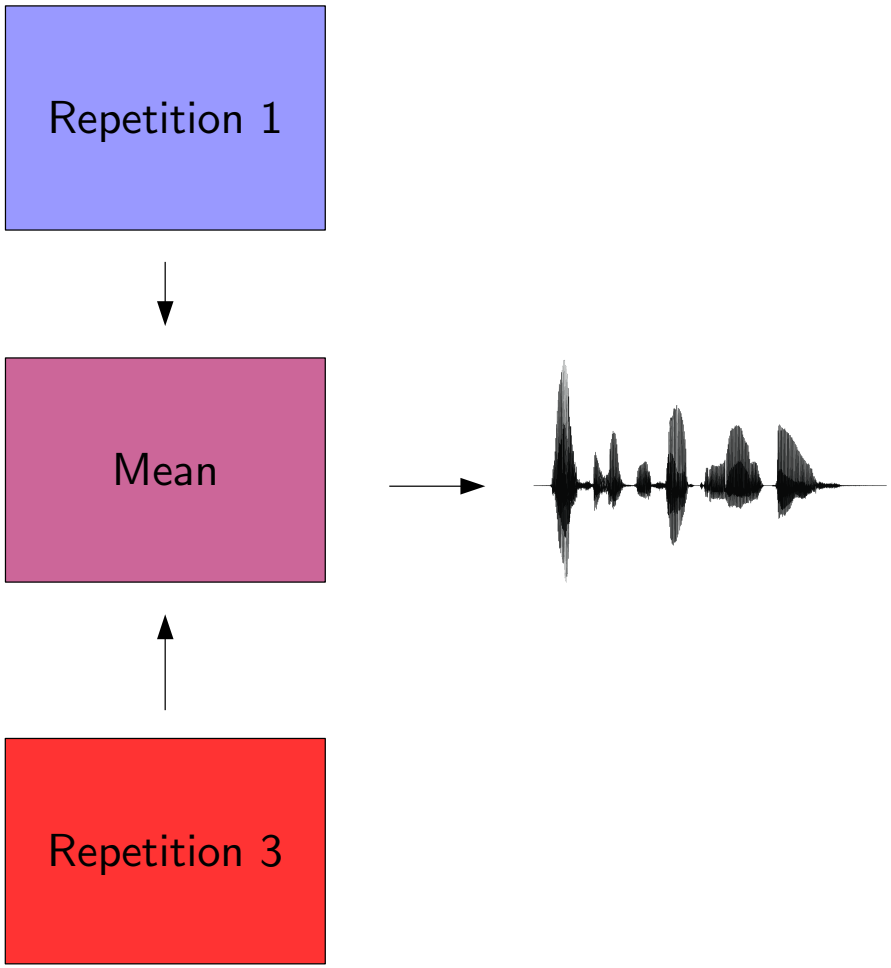
Create mean speech

3.b. Take the mean of all repetitions:



Create mean speech

3.b. Resynthesise mean speech (condition "M"):



Interpretation

More than methodology: Our manipulations are meaningful because they have deep and important mathematical interpretations

Interpretation

More than methodology: Our manipulations are meaningful because they have deep and important mathematical interpretations

- Repeated speech $\approx$ independent samples from a "perfect" acoustic model
- Chimeric speech $\approx$ samples from a model making certain independence assumptions but no distributional assumptions
- Mean speech $\approx$ the mean of a probabilistic model
  - Same mean regardless of feature independence assumptions

Overview

First investigation

- Two model-assumption classes:
  1. Stream independence assumptions
     - 1.1 Source and filter parameters conditionally independent
     - 1.2 Filter, pitch, aperiodicities conditionally independent
  2. Conditional independence among filter coefficients

First investigation

- Two model-assumption classes:
  1. Stream independence assumptions
     - 1.1 Source and filter parameters conditionally independent
     - 1.2 Filter, pitch, aperiodicities conditionally independent
  2. Conditional independence among filter coefficients
- Two output-generation methods:
  1. Random sampling from probability distribution
  2. Mean parameter generation

First investigation

- Two model-assumption classes:
  1. Stream independence assumptions
     - 1.1 Source and filter parameters conditionally independent "SF"
     - 1.2 Filter, pitch, aperiodicities conditionally independent "SI"
  2. Conditional independence among filter coefficients "I" etc.
- Two output-generation methods:
  1. Random sampling from probability distribution
  2. Mean parameter generation "M"

= 12 *conditions* (4 baselines: "N", "VU", "V", "D")

- For each of our 30 Harvard sentences

Listening examples

Sampling-based generation:

Database examples:	3	7	26	32	
Baselines:	N	VU	V	D	
Stream independence:	SF	SI			
Filter coefficient independence:	L1	L2	H1	H2	I

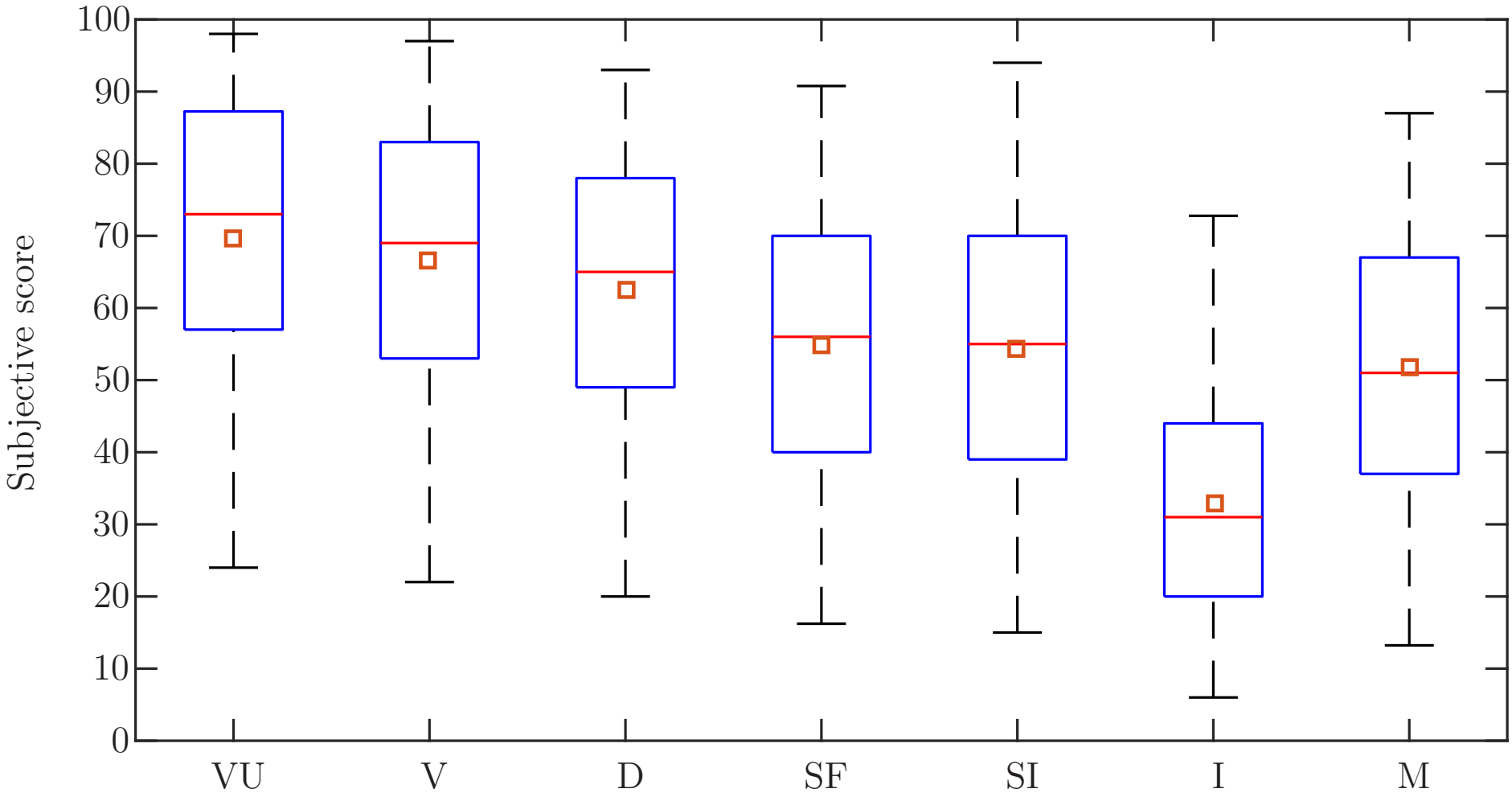
Mean-based generation:

Averaging:	M

(More at homepages.inf.ed.ac.uk/ghenter)

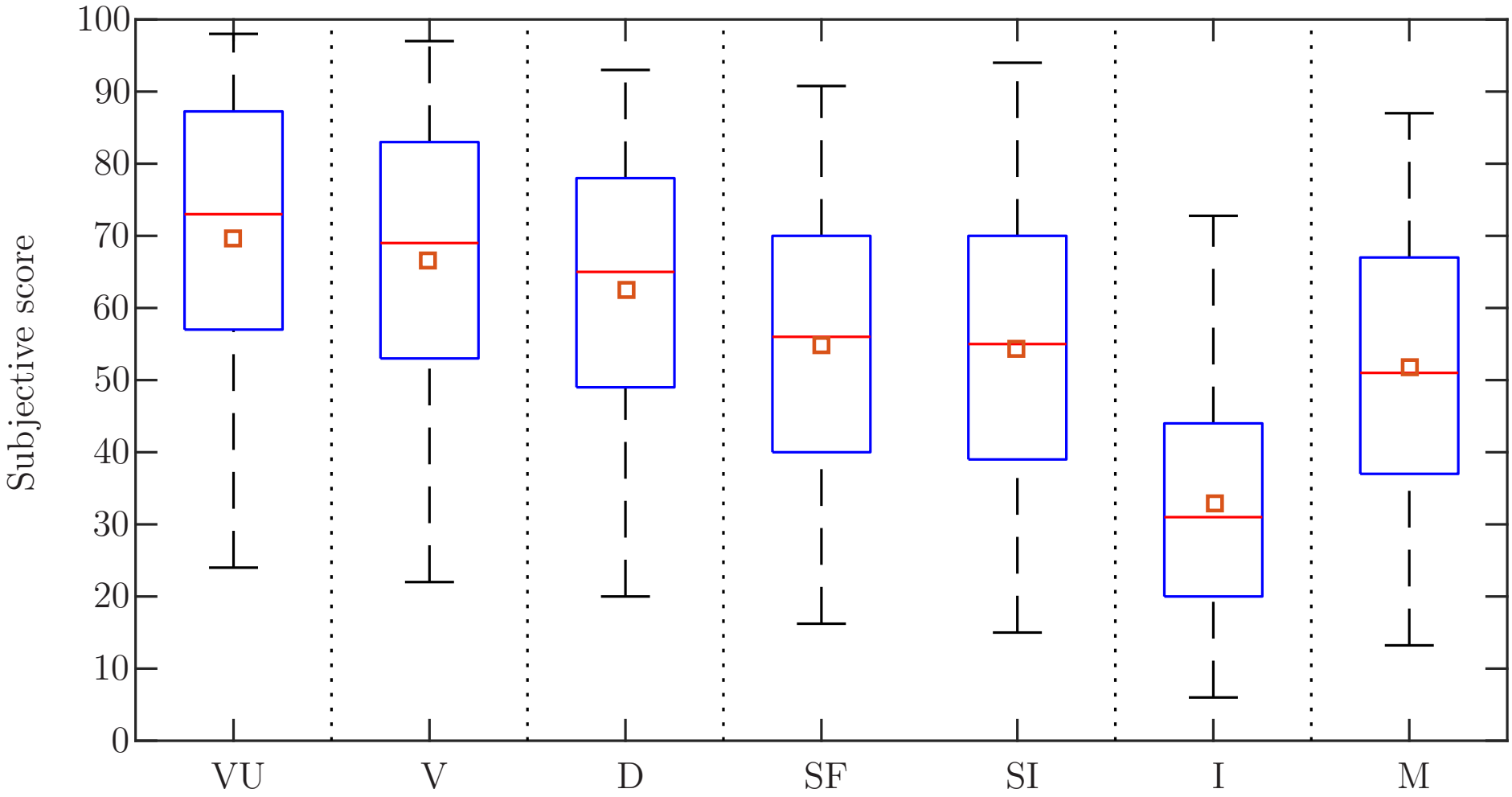
Naturalness results

Box plot of 549 MUSHRA comparisons rating natural speech at 100:



Naturalness results

Wilcoxon signed-rank tests ($\alpha = 0.01$) separate most conditions



Findings

- When sampling from models:
  1. Source-filter independence assumption reduces naturalness
  2. Independence assumptions among filter coefficients further reduce naturalness

Findings

- When sampling from models:
  1. Source-filter independence assumption reduces naturalness
  2. Independence assumptions among filter coefficients further reduce naturalness
- Using mean-based parameter generation:
  1. Better than sampling for poor models
  2. Less natural than sampling for accurate models

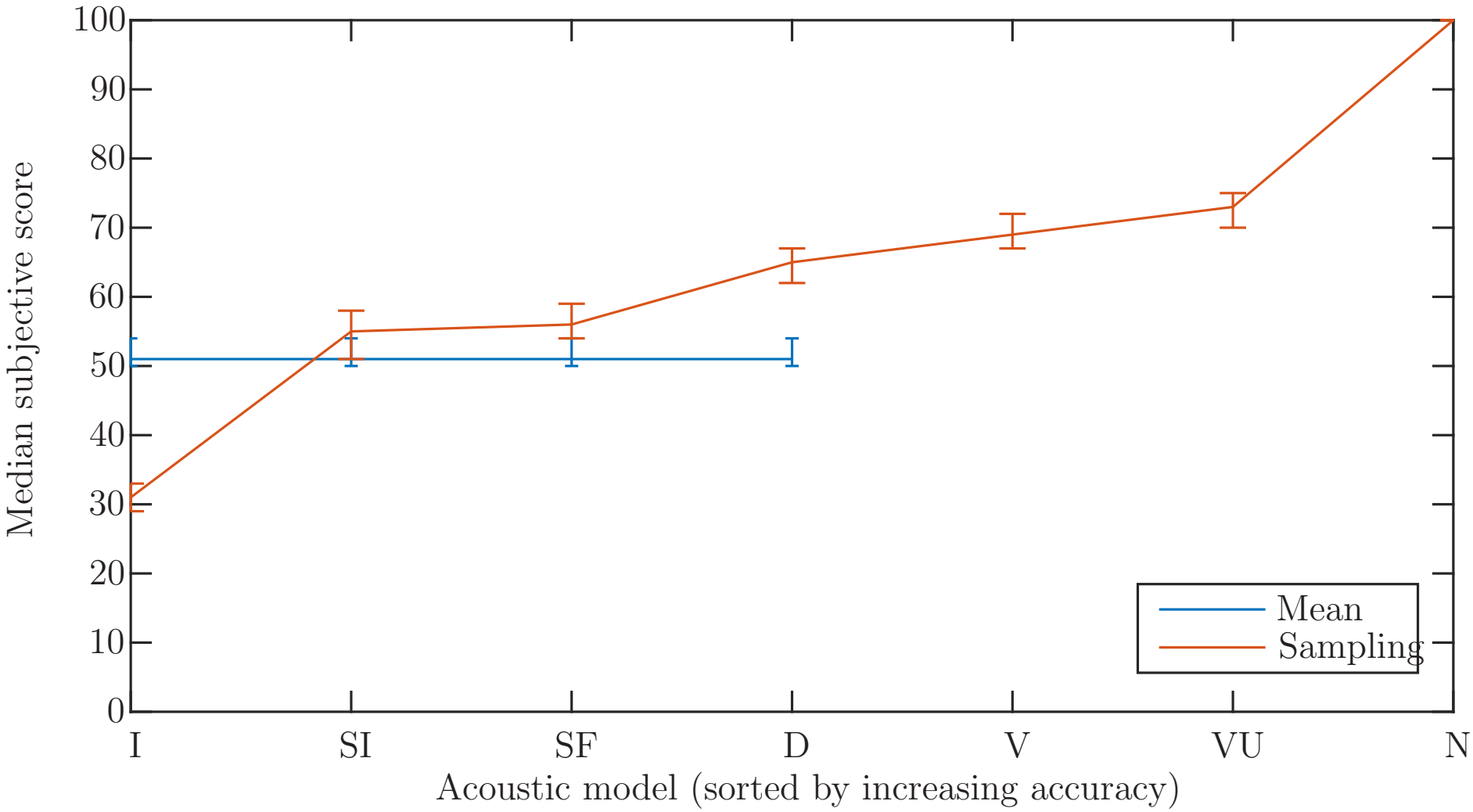
Findings

- When sampling from models:
  1. Source-filter independence assumption reduces naturalness
  2. Independence assumptions among filter coefficients further reduce naturalness
- Using mean-based parameter generation:
  1. Better than sampling for poor models
  2. Less natural than sampling for accurate models

Note: Since $\widehat{x}_{\mathrm{MMSE}} = \mathrm{argmin}_{\widehat{x}} \, \mathbb{E}\,(X - \widehat{x})^2 = \mathbb{E}\,(X)$ for any X-distribution, any approach that successfully minimises the squared error of the predicted speech features will sound like M

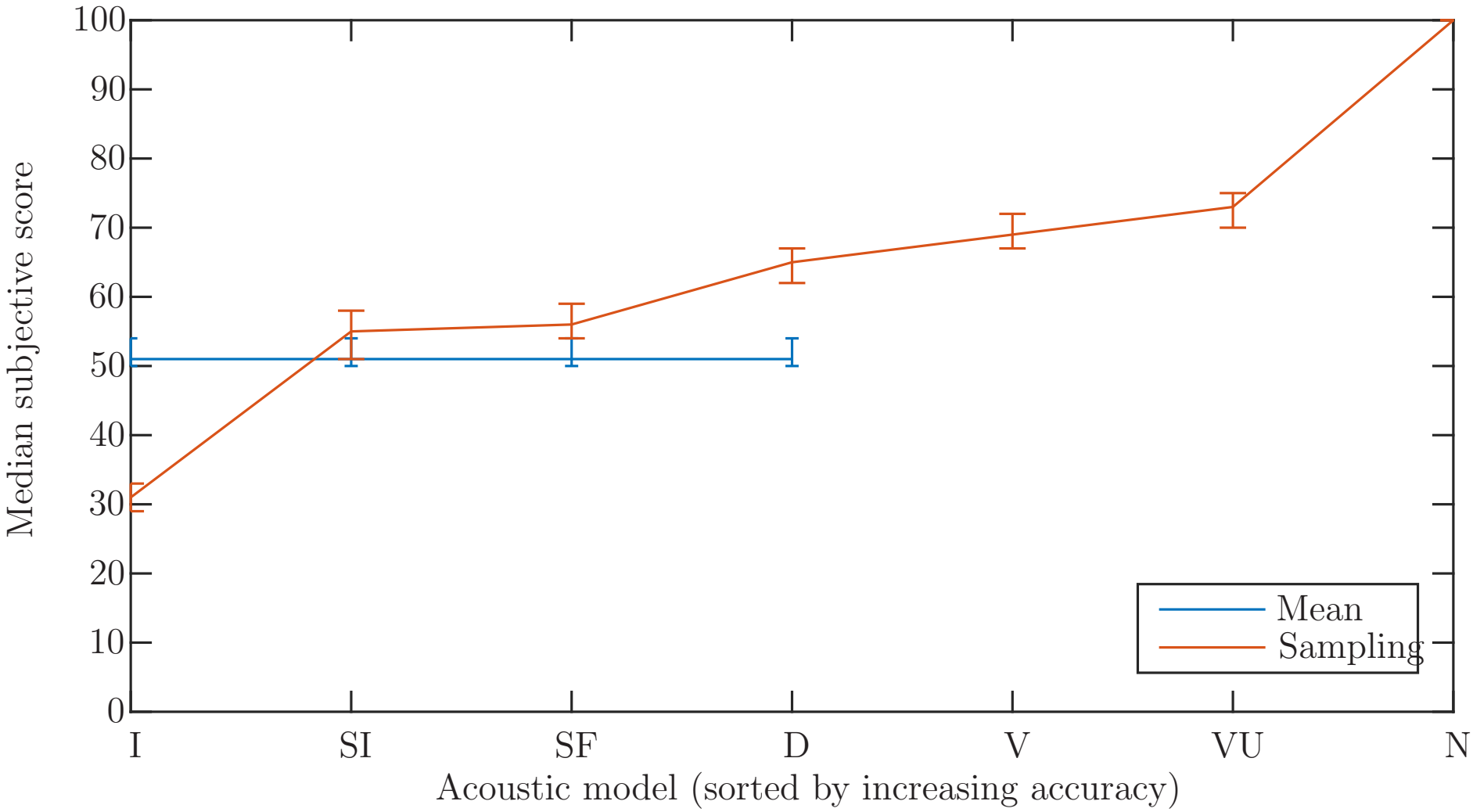
Mean vs. sampling generation

Mean-based generation (=MMSE) may never sound perfectly natural



Mean vs. sampling generation

Not until WaveNet (van den Oord et al., 2016) did crossover occur



Study limitations

Study conclusions not directly applicable to:

- Other speakers
- Other speech representations
  - Different vocoders, waveform-level synthesis, ...
- Other parameter generation methods
  - Postfiltering, global variance modelling (GV), modulation spectrum, ...

...but the same methodology is still applicable

Methodological limitations

The repeated-speech methodology has broad applicability

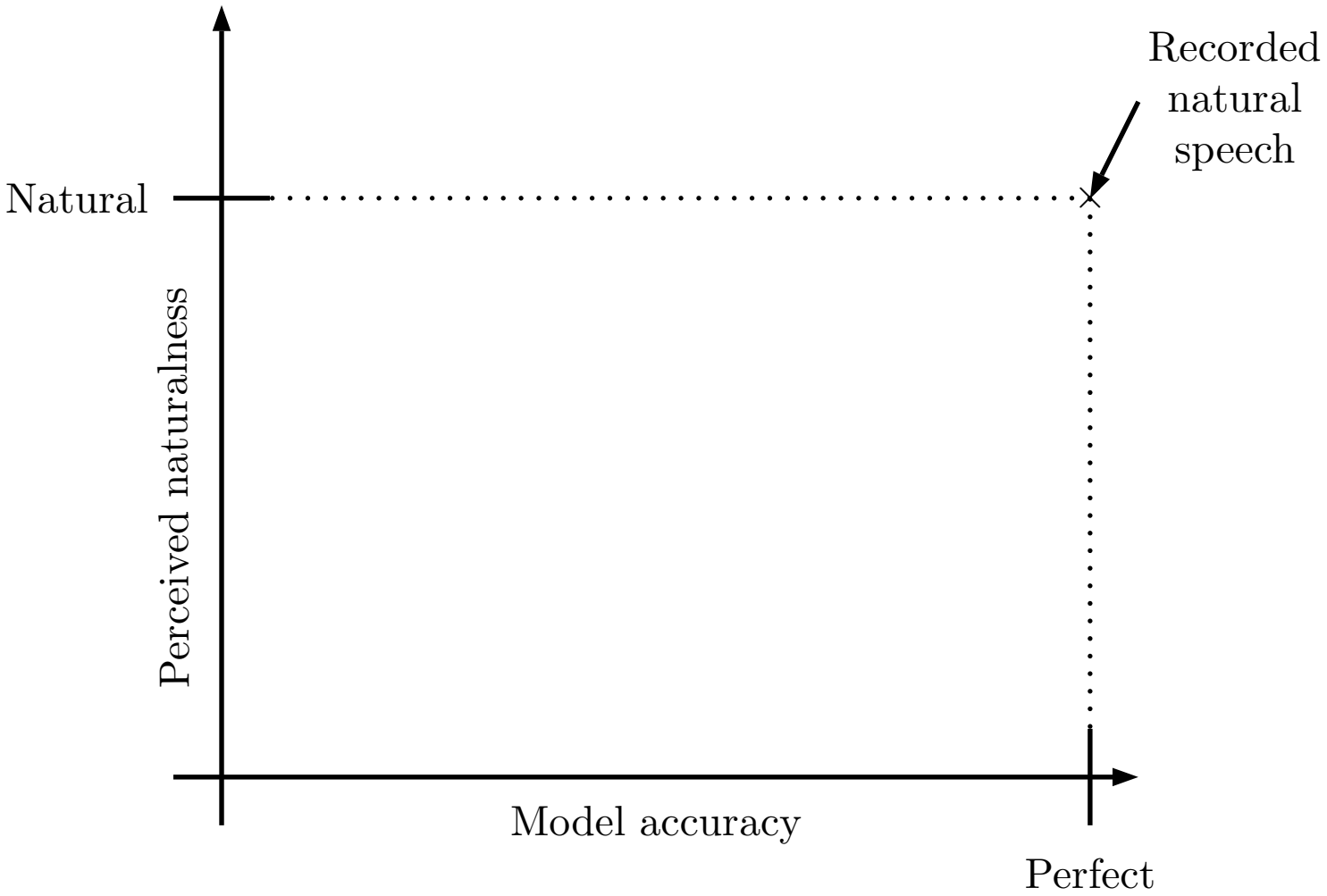
- Both positive and negative results are of interest

Key limitations:

- *Can only speak pre-recorded prompts*
  - Not a general-purpose TTS system
- Needs accurate time-alignment of repetitions
- Manipulation artefacts must be kept in check
- Applies in the limit of accurate models
  - Higher modelling accuracy $\not\Rightarrow$ perceptually preferred
  - E.g., GV gives a less accurate model, but better-sounding mean speech (Shannon and Byrne, 2013)

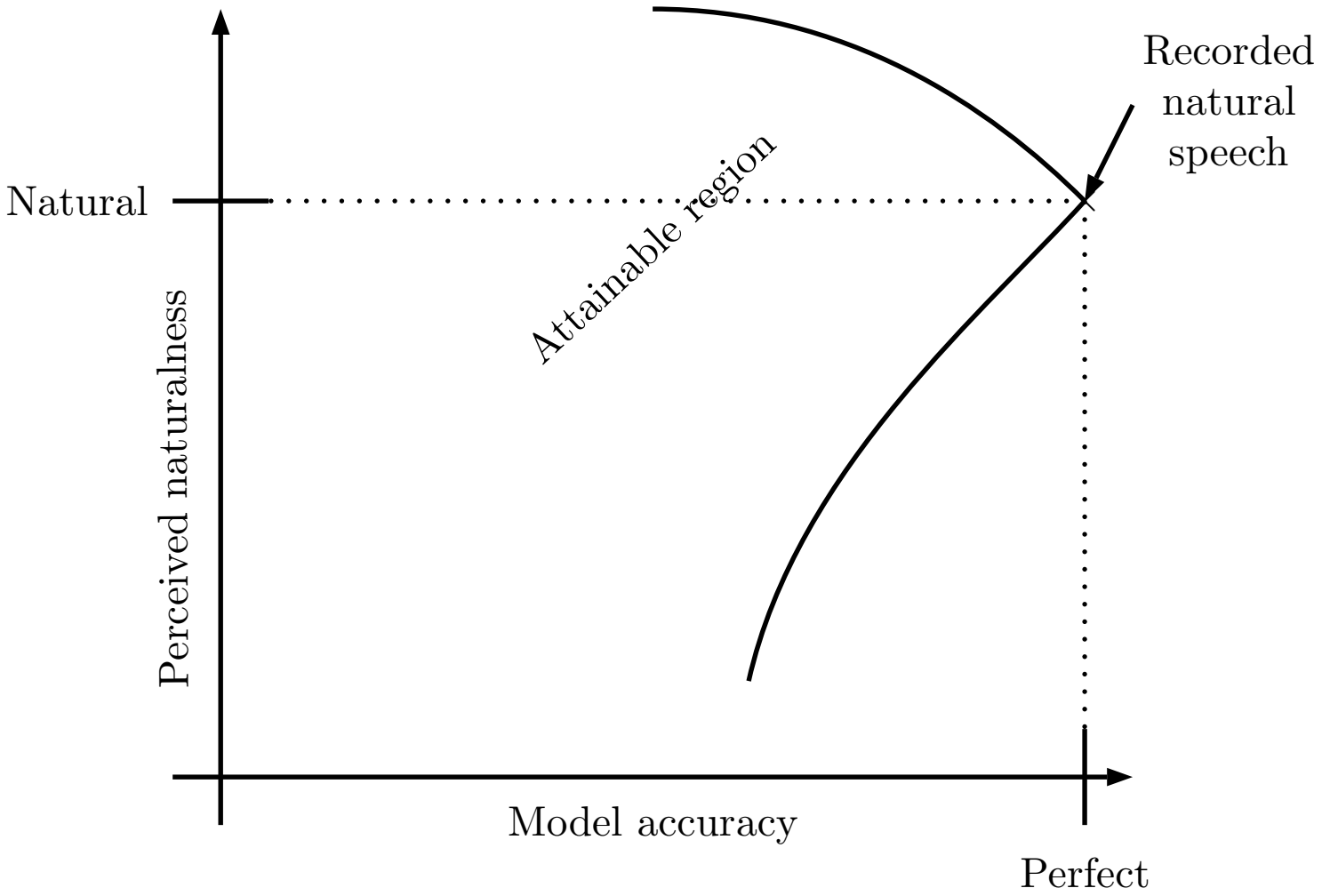
Limitations of accurate models

Not all accuracy-naturalness combinations are attainable



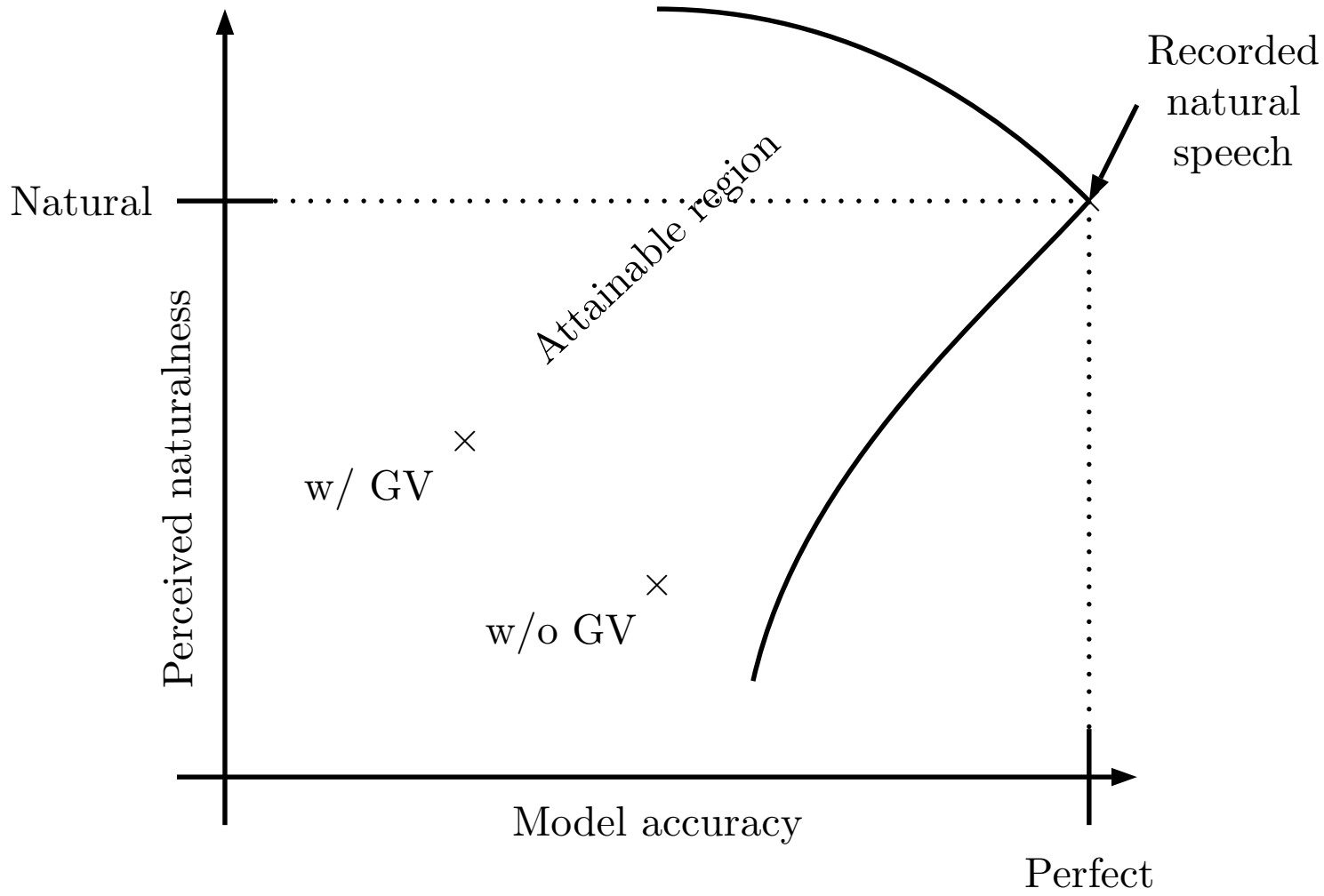
Limitations of accurate models

Well-produced audio can be preferred over raw natural recordings



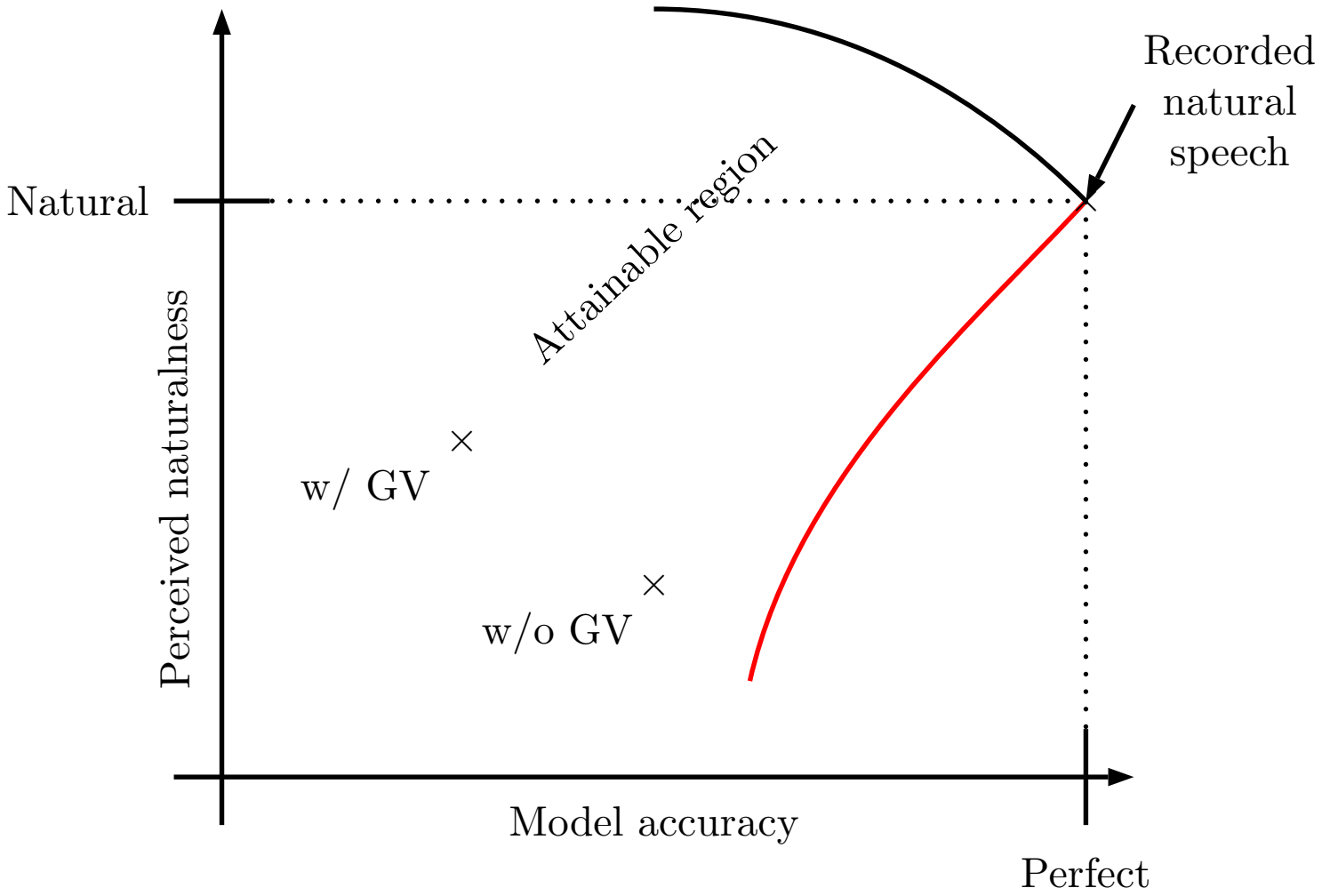
Limitations of accurate models

TTS with GV is exaggerated (less accurate), but sounds better



Limitations of accurate models

Not clear if repeated speech can explore models above the dotted line



Overview

Outlook

Starting point: Perceptual debugging is not isolated papers, but an entire line of research!

- Many things left to explore
- The use of repeated speech, especially, allows new questions to be answered
- Let me know if you are interested!
  - We have an unreleased database (REHASP 1.0) with 40×100 repetitions + newspaper sentences

Future directions: Modelled speech

- Extended synthesiser comparison
  - RNNs as well as DNNs and HMMs
  - Include question set and F0 handling
  - Evaluate with predicted durations
  - Include GV-based output generation

Future directions: Modelled speech

- Extended synthesiser comparison
  - RNNs as well as DNNs and HMMs
  - Include question set and F0 handling
  - Evaluate with predicted durations
  - Include GV-based output generation
- Which factors are necessary for DNN success?
  - Opinions differ on why DNNs work so well
  - Dataset size, computational resources, depth, pretraining?

Future directions: Repeated speech I

- Finding speech signals that optimise objective measures
  - Mean speech approx. minimises mel-cepstral distortion (MCD)
  - Iterate alignment and minimisation, similar to EM-algorithm
  - Estimate the expected MCD of sampled and mean speech
  - What is the MCD gap to modelled speech?
  - What does minimum-MCD speech sound like?

Future directions: Repeated speech I

- Finding speech signals that optimise objective measures
  - Mean speech approx. minimises mel-cepstral distortion (MCD)
  - Iterate alignment and minimisation, similar to EM-algorithm
  - Estimate the expected MCD of sampled and mean speech
  - What is the MCD gap to modelled speech?
  - What does minimum-MCD speech sound like?
- Directly approximating the most likely speech parameters
  - Standard SPSS parameter generation (Tokuda et al., 2000) is derived from seeking the most probable output
  - The peak is different from the mean if the distribution is skewed
  - Apply mode-seeking algorithms

Future directions: Repeated speech I

- Finding speech signals that optimise objective measures
  - Mean speech approx. minimises mel-cepstral distortion (MCD)
  - Iterate alignment and minimisation, similar to EM-algorithm
  - Estimate the expected MCD of sampled and mean speech
  - What is the MCD gap to modelled speech?
  - What does minimum-MCD speech sound like?
- Directly approximating the most likely speech parameters
  - Standard SPSS parameter generation (Tokuda et al., 2000) is derived from seeking the most probable output
  - The peak is different from the mean if the distribution is skewed
  - Apply mode-seeking algorithms
- Measuring human consistency
  - How precisely do humans control speech signal properties?
  - What is consistent, and what is not?

Future directions: Repeated speech II

- Perceptual debugging beyond naturalness
  - How do design decisions affect intelligibility in noise, speaker similarity, or completion rates and times on complex task?
  - Example: Transcription of chimeric/mean stimuli in noise

Future directions: Repeated speech II

- Perceptual debugging beyond naturalness
  - How do design decisions affect intelligibility in noise, speaker similarity, or completion rates and times on complex task?
  - Example: Transcription of chimeric/mean stimuli in noise
- Perceptual debugging of other TTS paradigms
  - Hybrid synthesis (trajectory tiling) parameter generation
  - Waveform-level alignments for waveform-level synthesis

Future directions: Repeated speech II

- Perceptual debugging beyond naturalness
  - How do design decisions affect intelligibility in noise, speaker similarity, or completion rates and times on complex task?
  - Example: Transcription of chimeric/mean stimuli in noise
- Perceptual debugging of other TTS paradigms
  - Hybrid synthesis (trajectory tiling) parameter generation
  - Waveform-level alignments for waveform-level synthesis
- Perceptual debugging of TTS prosody
  - Transplanting prosody between man and machine

Future directions: Repeated speech II

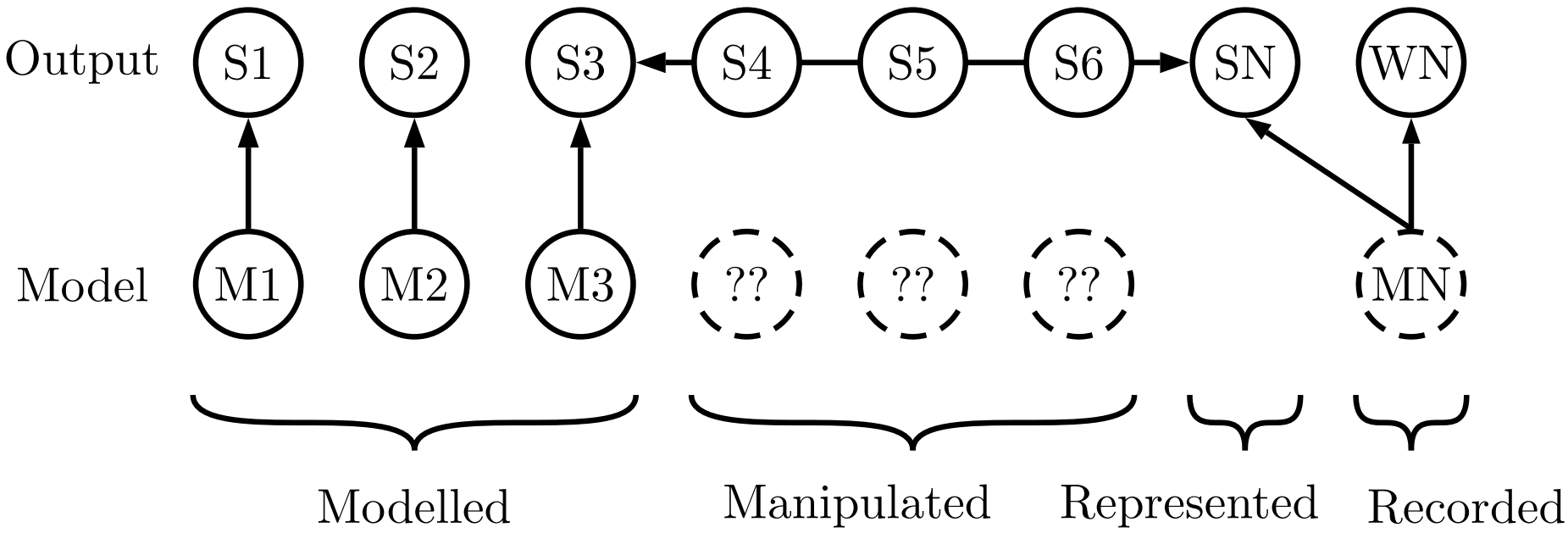
- Perceptual debugging beyond naturalness
  - How do design decisions affect intelligibility in noise, speaker similarity, or completion rates and times on complex task?
  - Example: Transcription of chimeric/mean stimuli in noise
- Perceptual debugging of other TTS paradigms
  - Hybrid synthesis (trajectory tiling) parameter generation
  - Waveform-level alignments for waveform-level synthesis
- Perceptual debugging of TTS prosody
  - Transplanting prosody between man and machine
- Multimodal data, e.g., lip/body motion synthesis...
  - Use repeated recordings to test independence assumptions beyond TTS

Future directions: Both

- Partition the entire perceptual gap between natural and synthetic speech
  - Text analysis
  - Speech representation
    - Vocoder; MGCs vs. PCA dimensionality reduction
  - Independence assumptions
  - Distribution assumptions
  - . . . all under different generation methods
    - Sampling, mean generation (Tokuda et al., 2000), state-of-the-art GV compensation (Nose, 2016)

A unified approach

We can score the entire spectrum of design decisions in TTS:



The end

The end

Thank you for listening!

The end

Any questions?

References I

Chen, B., Chen, Z., Xu, J., and Yu, K. (2015).
An investigation of context clustering for statistical speech synthesis with deep neural network.
In *Proc. Interspeech*, pages 2212–2216.

Chen, Z. and Yu, K. (2014).
An investigation of implementation and performance analysis of dnn based speech synthesis system.
In *Proc. ICSP*, pages 577–582.

Hashimoto, K., Oura, K., Nankaku, Y., and Tokuda, K. (2015).
The effect of neural networks in statistical parametric speech synthesis.
In *Proc. ICASSP*, pages 4455–4459.

Henter, G. E., Merritt, T., Shannon, M., Mayo, C., and King, S. (2014).
Measuring the perceptual effects of modelling assumptions in speech synthesis using stimuli constructed from repeated natural speech.
In *Proc. Interspeech*, pages 1504–1508.

References II

Kawahara, H. (2006).
STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds.
Acoust. Sci. Technol., 27(6):349–353.

Merritt, T., Latorre, J., and King, S. (2015).
Attributing modelling errors in hmm synthesis by stepping gradually from natural to modelled speech.
In *Proc. ICASSP*, pages 4220–4224.

Nose, T. (2016).
Efficient implementation of global variance compensation for parametric speech synthesis.
IEEE/ACM T. Audio Speech, 24(10):1694–1704.

Qian, Y., Fan, Y., Hu, W., and Soong, F. K. (2014).
On the training aspects of deep neural network (DNN) for parametric TTS synthesis.
In *Proc. ICASSP*, pages 3829–3833.

References III

Ribeiro, M. S., Yamagishi, J., and Clark, R. A. J. (2015).
A perceptual investigation of wavelet-based decomposition of f0 for text-to-speech synthesis.
In *Proc. Interspeech*, pages 1586–1590.

Shannon, M. and Byrne, W. (2013).
Fast, low-artifact speech synthesis considering global variance.
In *Proc. ICASSP*, pages 7869–7873.

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R., Saurous, R. A., Agiomyrgiannakis, Y., and Wu, Y. (2018).
Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions.
In *Proc. ICASSP*, pages 4779–4783.

Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., and Kitamura, T. (2000).
Speech parameter generation algorithms for HMM-based speech synthesis.
In *Proc. ICASSP*, pages 1315–1318.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016).
WaveNet: A generative model for raw audio.

References IV

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agiomyrgiannakis, Y., Clark, R., and Saurous, R. A. (2017).

Tacotron: Towards end-to-end speech synthesis.
In *Proc. Interspeech*, pages 4006–4010.

Watts, O., Henter, G. E., Merritt, T., Wu, Z., and King, S. (2016).
From HMMs to DNNs: where do the improvements come from?
In *Proc. ICASSP*, pages 5505–5509.

Wu, Z., Valentini-Botinhao, C., Watts, O., and King, S. (2015).
Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis.
In *Proc. ICASSP*, pages 4460–4464.

Wu, Z., Watts, O., and King, S. (2016).
Merlin: An open source neural network speech synthesis system.
In *Proc. SSW9*, pages 218–223.

Zen, H., Nose, T., Yamagishi, J., Sako, S., Masuko, T., Black, A., and Tokuda, K. (2007).
The HMM-based speech synthesis system (HTS) version 2.0.
In *Proc. SSW6*, pages 294–299.

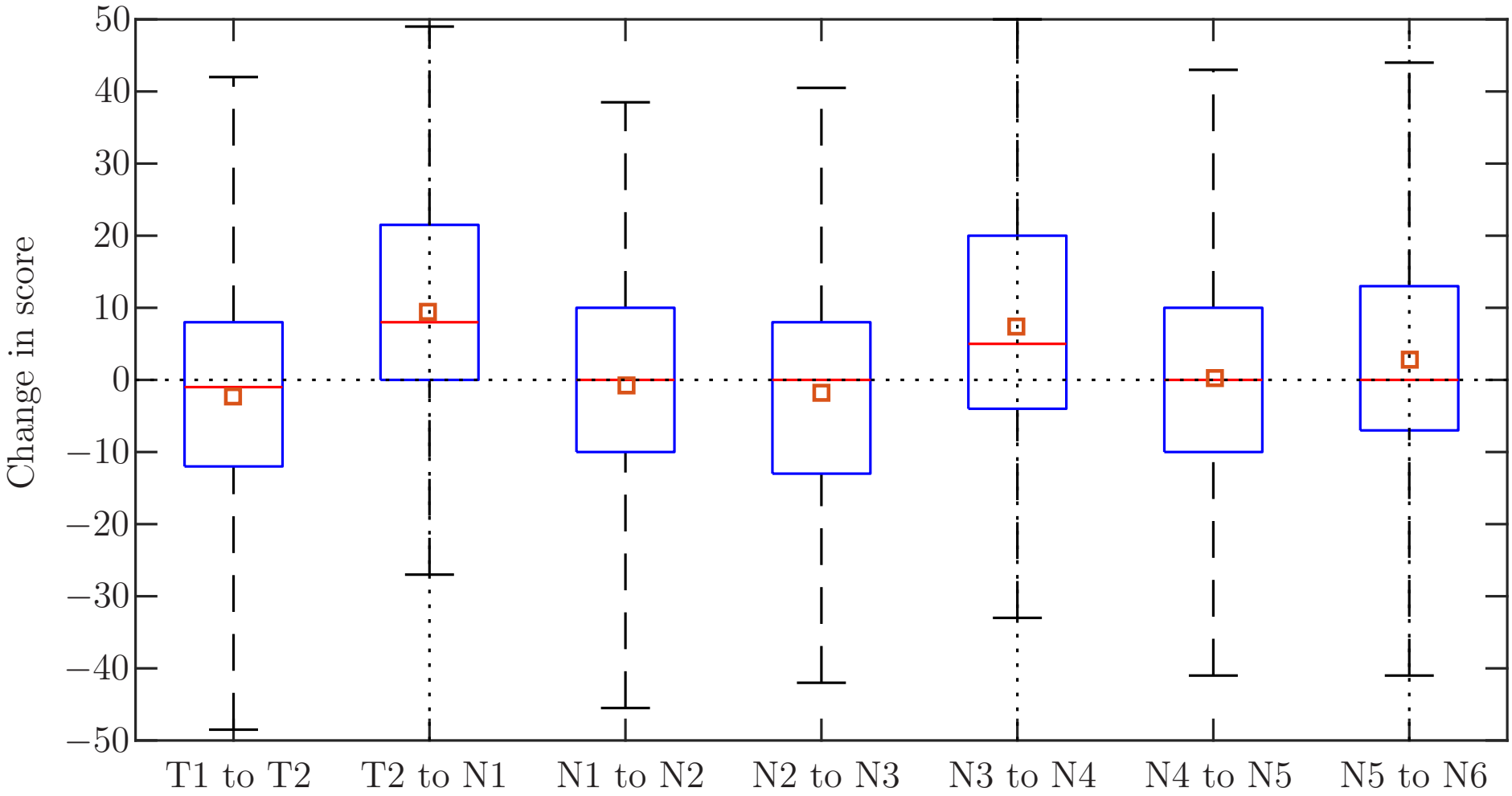
References V

Zen, H. and Senior, A. (2014).
Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis.
In *Proc. ICASSP*, pages 3872–3876.

Zen, H., Senior, A., and Schuster, M. (2013).
Statistical parametric speech synthesis using deep neural networks.
In *Proc. ICASSP*, pages 7962–7966.

Pairwise differences

Box plot of score differences when stepping from HTS to Merlin:



Interpretation

- Repeated speech $\approx$ independent samples from a "perfect" acoustic model

$$\mathcal{D} = \{\boldsymbol{x}_n\} \quad \boldsymbol{X}_n \sim f_{\boldsymbol{X}_n \mid \boldsymbol{L}}(\boldsymbol{x} \mid \boldsymbol{l}) = f_{\boldsymbol{X} \mid \boldsymbol{L}}(\boldsymbol{x} \mid \boldsymbol{l}) \quad \boldsymbol{X} = \begin{bmatrix} \boldsymbol{S} \\ \boldsymbol{F} \end{bmatrix}$$

- Chimeric speech $\approx$ samples from a model making certain independence assumptions but no distributional assumptions

$$f_{\boldsymbol{X} \mid \boldsymbol{L}}(\boldsymbol{x} \mid \boldsymbol{l}) = f_{\boldsymbol{S} \mid \boldsymbol{L}}(\boldsymbol{s} \mid \boldsymbol{l})\, f_{\boldsymbol{F} \mid \boldsymbol{L}}(\boldsymbol{f} \mid \boldsymbol{l})$$

$$f_{\boldsymbol{S}_1 \mid \boldsymbol{L}}(\boldsymbol{s}_1 \mid \boldsymbol{l})\, f_{\boldsymbol{F}_1 \mid \boldsymbol{L}}(\boldsymbol{f}_1 \mid \boldsymbol{l}) \cdot f_{\boldsymbol{S}_2 \mid \boldsymbol{L}}(\boldsymbol{s}_2 \mid \boldsymbol{l})\, f_{\boldsymbol{F}_2 \mid \boldsymbol{L}}(\boldsymbol{f}_2 \mid \boldsymbol{l})$$
$$= f_{\boldsymbol{S}_1 \mid \boldsymbol{L}}(\boldsymbol{s}_1 \mid \boldsymbol{l})\, f_{\boldsymbol{F}_1 \mid \boldsymbol{L}}(\boldsymbol{f}_2 \mid \boldsymbol{l}) \cdot f_{\boldsymbol{S}_2 \mid \boldsymbol{L}}(\boldsymbol{s}_2 \mid \boldsymbol{l})\, f_{\boldsymbol{F}_2 \mid \boldsymbol{L}}(\boldsymbol{f}_1 \mid \boldsymbol{l})$$

- Mean speech $\approx$ the mean of a probabilistic model

$$(\mathbb{E}(\boldsymbol{X} \mid \boldsymbol{l}))_d = \int x_d f_{\boldsymbol{X} \mid \boldsymbol{L}}(\boldsymbol{x} \mid \boldsymbol{l})\, d\boldsymbol{x} = \int x_d f_{X_d \mid \boldsymbol{L}}(x_d \mid \boldsymbol{l})\, dx_d$$

REHASP 1.0 corpus

- "REpeated HArvard Sentence Prompts"
- Professional male native British English speaker "Nick"
- 40 Harvard sentence prompts
- Each read aloud 100 times
  - Presented in random order
- 24 bit 96 kHz
- To be made available under a permissive license
- Also includes ≈3000 news prompts

REHASP 1.0 corpus

- "REpeated HArvard Sentence Prompts"
- Professional male native British English speaker "Nick"
- 40 Harvard sentence prompts
- Each read aloud 100 times
  - Presented in random order
- 24 bit 96 kHz
- To be made available under a permissive license
- Also includes ≈3000 news prompts

REHASP 1.0 corpus

- "REpeated HArvard Sentence Prompts"
- Professional male native British English speaker "Nick"
- 40 Harvard sentence prompts
- Each read aloud 100 times
  - Presented in random order
- 24 bit 96 kHz
- To be made available under a permissive license
- Also includes ≈3000 news prompts

REHASP 1.0 corpus

- "REpeated HArvard Sentence Prompts"
- Professional male native British English speaker "Nick"
- 40 Harvard sentence prompts
- Each read aloud 100 times
  - Presented in random order
- 24 bit 96 kHz
- To be made available under a permissive license
- Also includes ≈3000 news prompts

REHASP 1.0 corpus

- "REpeated HArvard Sentence Prompts"
- Professional male native British English speaker "Nick"
- 40 Harvard sentence prompts
- Each read aloud 100 times
  - Presented in random order
- 24 bit 96 kHz
- To be made available under a permissive license
- Also includes ≈3000 news prompts

Recent synthesis breakthroughs

- WaveNet (van den Oord et al., 2016)
  - Naturalistic generative signal model
- Tacotron (Wang et al., 2017)
  - Forced alignment (decision tree) $\rightarrow$ jointly learned attention (neural network)
  - No need for a duration model
- Tacotron 2, others
  - WaveNet as a signal generator for SPSS
- Tacotron 2 (Shen et al., 2018)
  - Faster, simpler, 16-bit WaveNet
  - Learns to map speech parameters with SPSS artefacts to naturalistic signals, thus circumventing the bound we found