Learning Musicianship for Automatic Accompaniment

Gus (Guangyu) Xia

Roger Dannenberg

School of Computer Science

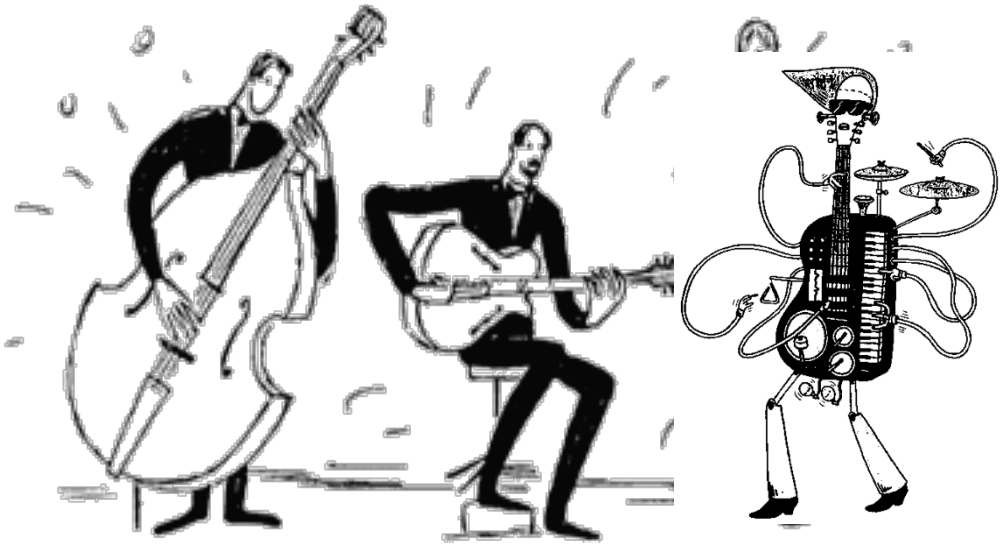
Carnegie Mellon University

Introduction: Musical background

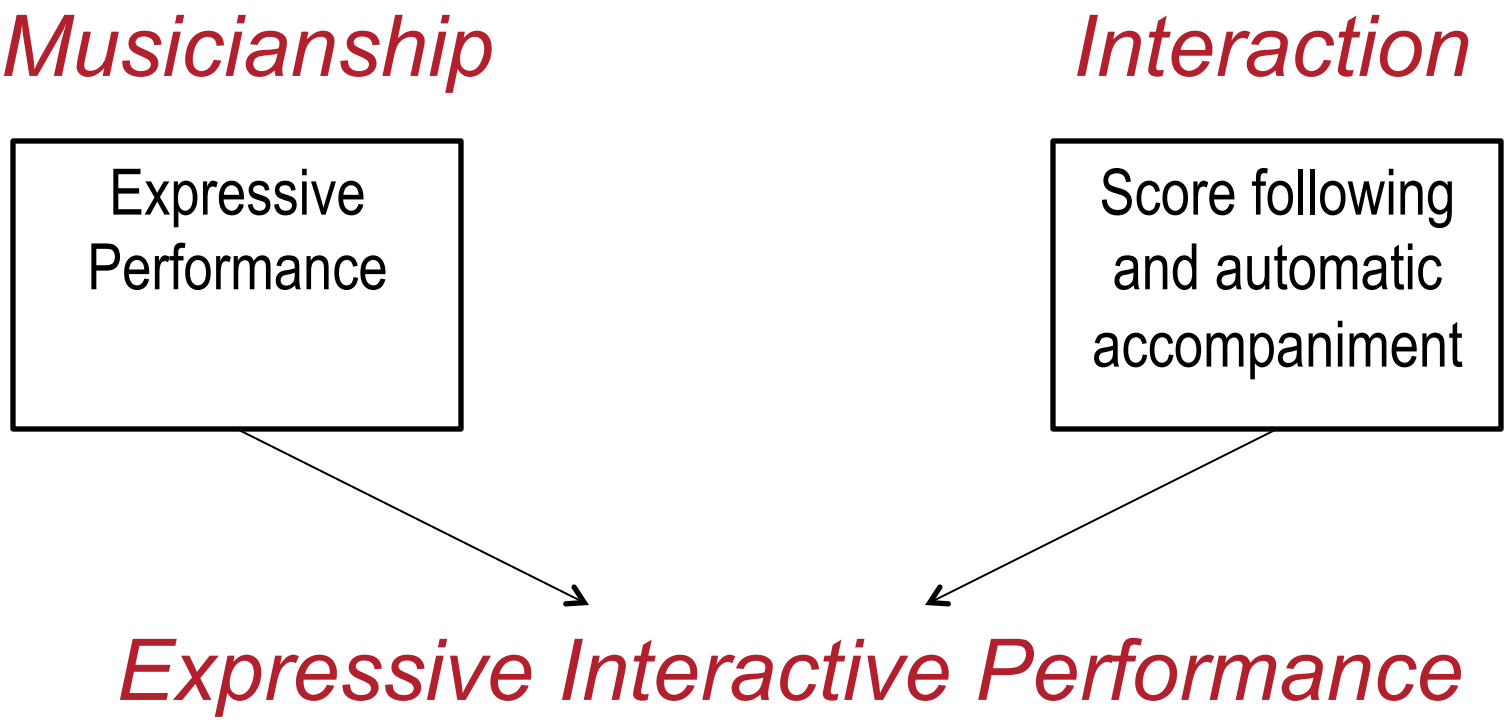
Interaction

Expression

Rehearsal



Introduction: Technical background



Introduction: Problem definition

For interactive music performance, how can we build artificial performers that automatically improve their ability to sense and coordinate with human musicians' expression with rehearsal experience?

- How to interpret the music based on the expression of human musicians?
- How to distill models from rehearsals?
- What are the limits of validity of the learned models?
- How many rehearsals are needed?

We start from piano duets, focusing on expressive timing and expressive dynamics.

Outline

- Introduction
- Data Collection
- Methods
- Demos
- Conclusion & Future Work

Current Data Collection

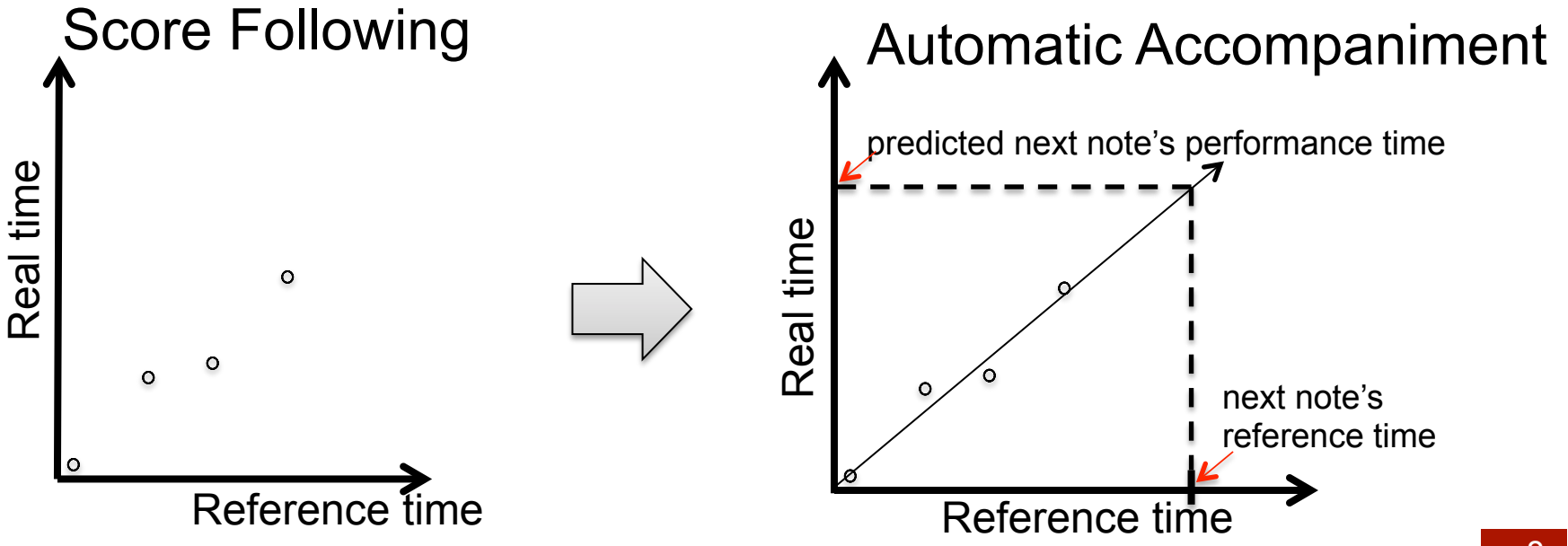
- Musicians:
  - 10 music master students play duet pieces in 5 pairs.
- Music pieces:
  - 3 pieces of music are selected, *Danny boy*, *Serenade* (by *Schubert*), and *Ashokan Farewell.*
  - Each pair performs every piece of music 7 times.
- Recording settings:
  - Recorded by electronic pianos with MIDI output.

Outline

- Introduction
- Data Collection
- **Methods**
- **Demos**
- **Conclusion & Future Work**

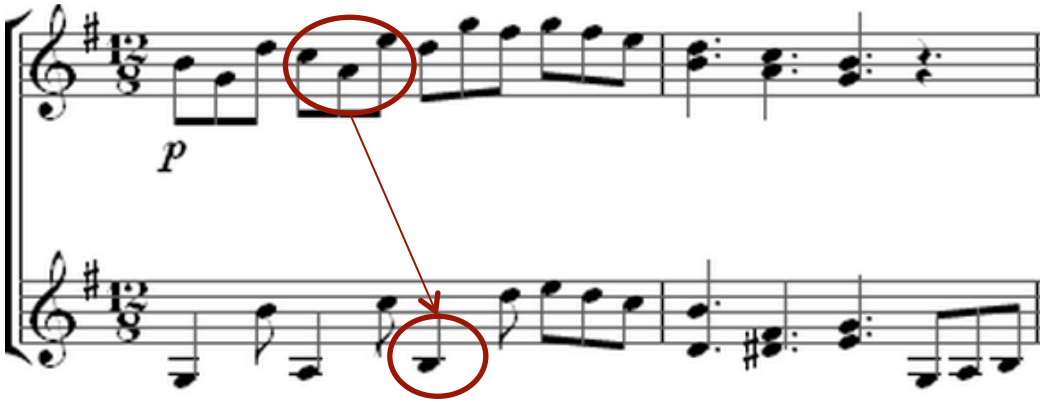
Method Overview

- From “local” to “general”
  - Local: low-dimensional feature space, only apply to certain notes
  - General: high-dimensional feature space, apply to the whole piece of music
- Base line:



Method (1): Note-specific approach

- Idea:
  - Expressive timings of the notes are linearly correlated.
  - Predict the expressive timing of 2nd piano by the expressive timing of 1st piano.



$$X = [x_1, x_2, \ldots, x_N]$$

$$Y = [y_1, y_2, \ldots, y_M]$$

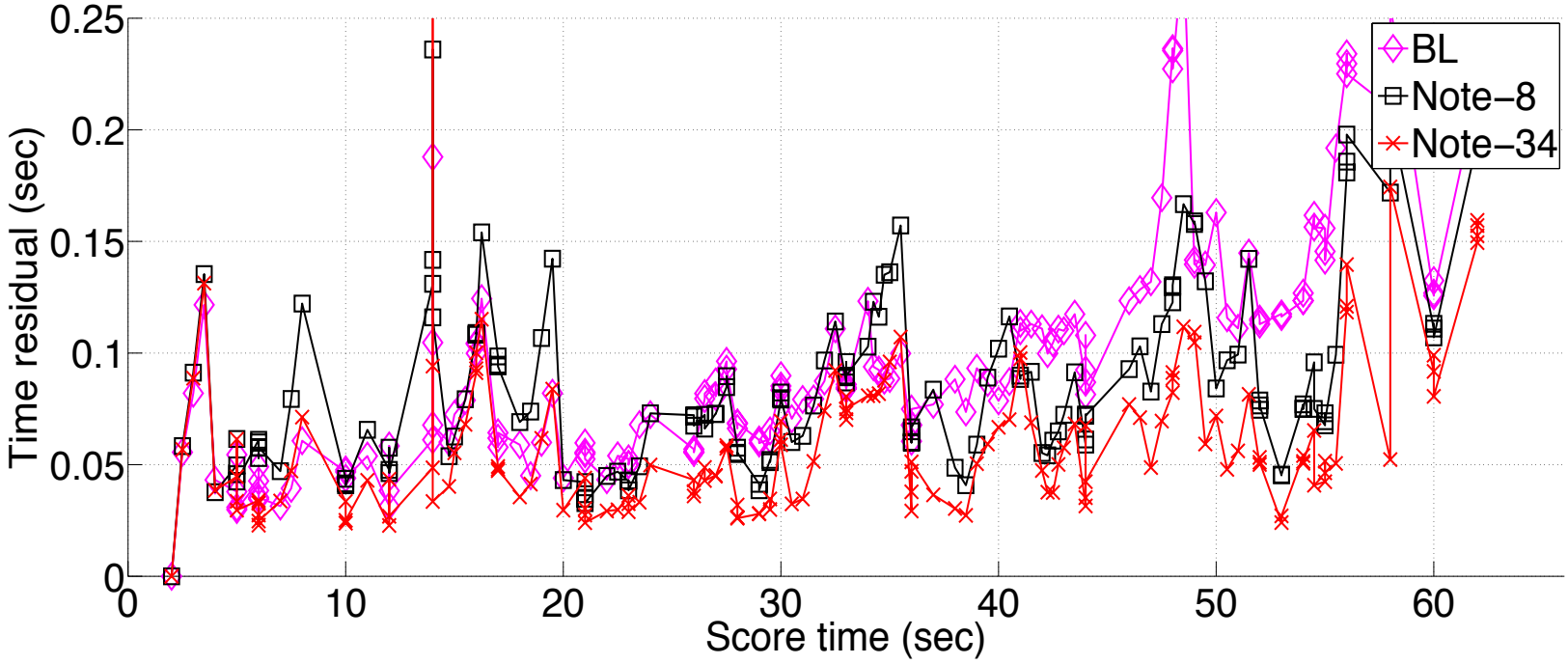
- Model:

$$y_i = \beta_0^{y_i} + \sum_{j=1}^{p} \beta_j^{y_i} x_{over(y_i)-j}$$

Result: Note-specific approach

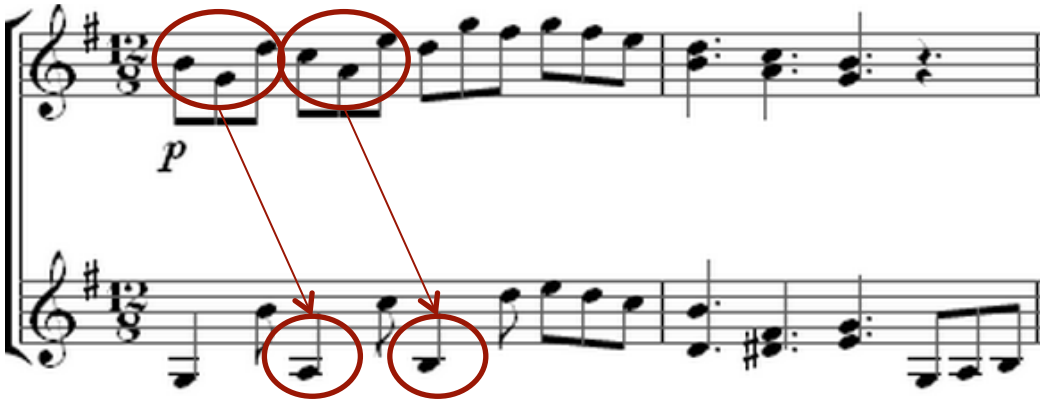
Mean Absolute Error:

BL: 0.098 Note-8: 0.087 Note-34: 0.060



Method (2): Rhythm-specific approach

- Idea:
  - Notes with same score rhythm context share parameters.
  - Introduces an extra dummy variable to encode the score rhythm context of each note .



$$X = [x_1, x_2, \ldots, x_N]$$

$$Y = [y_1, y_2, \ldots, y_M]$$

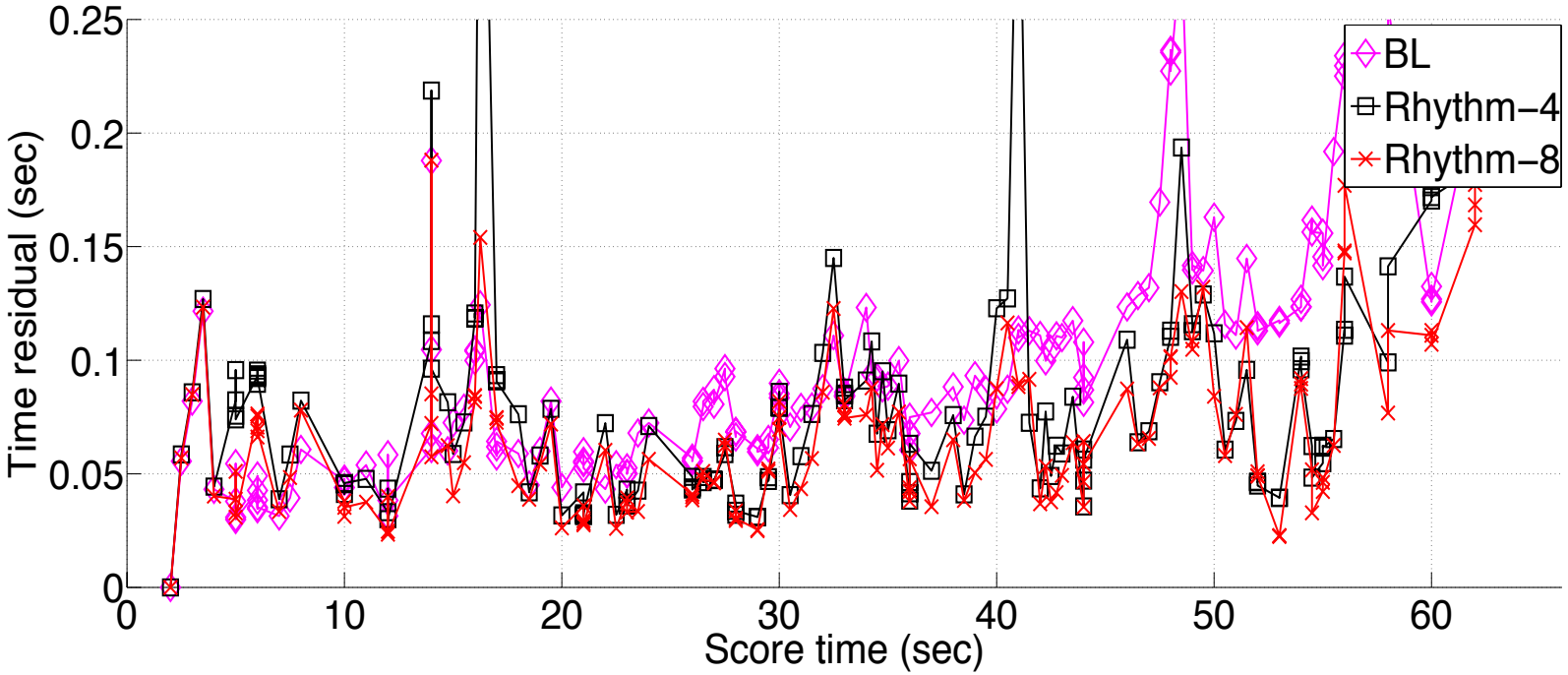
- Model:

$$y_i = \beta_0^{rhythm(y_i,q)} + \sum_{j=1}^{p} \beta_j^{rhythm(y_i,q)} x_{over(y_i)-j}$$

Result: Rhythm-specific approach

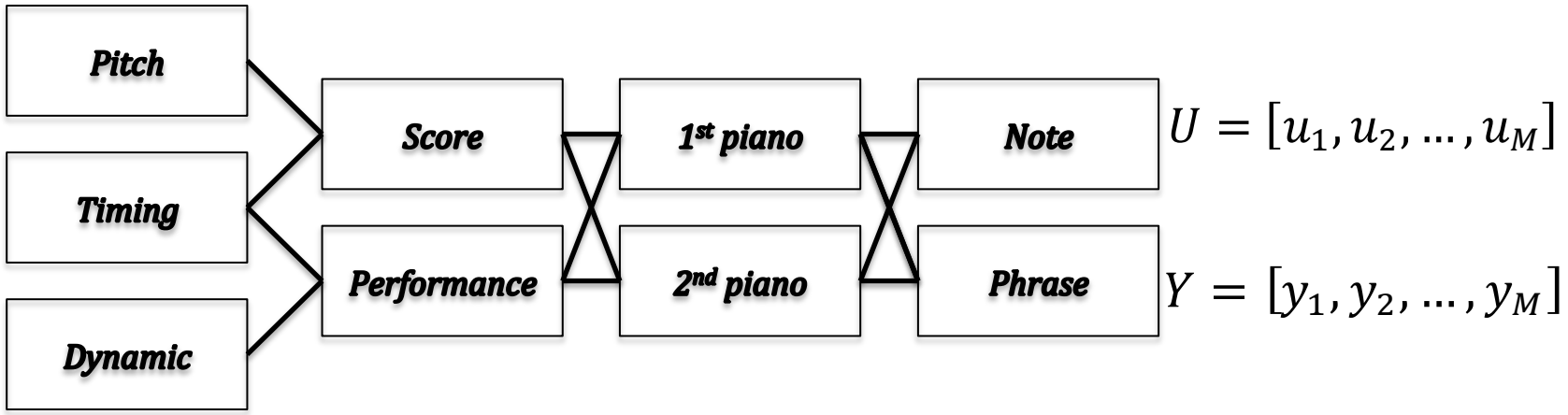
Mean Absolute Error:

BL: 0.098 Rhythm-4: 0.084 Rhythm-8: 0.067



Method (3): General feature approach

- Idea:
  - Make the model more general.
  - Predict the expressive timing by considering more than score rhythm context .



- Model:

$$Y = BU$$

Regularization: Group Lasso

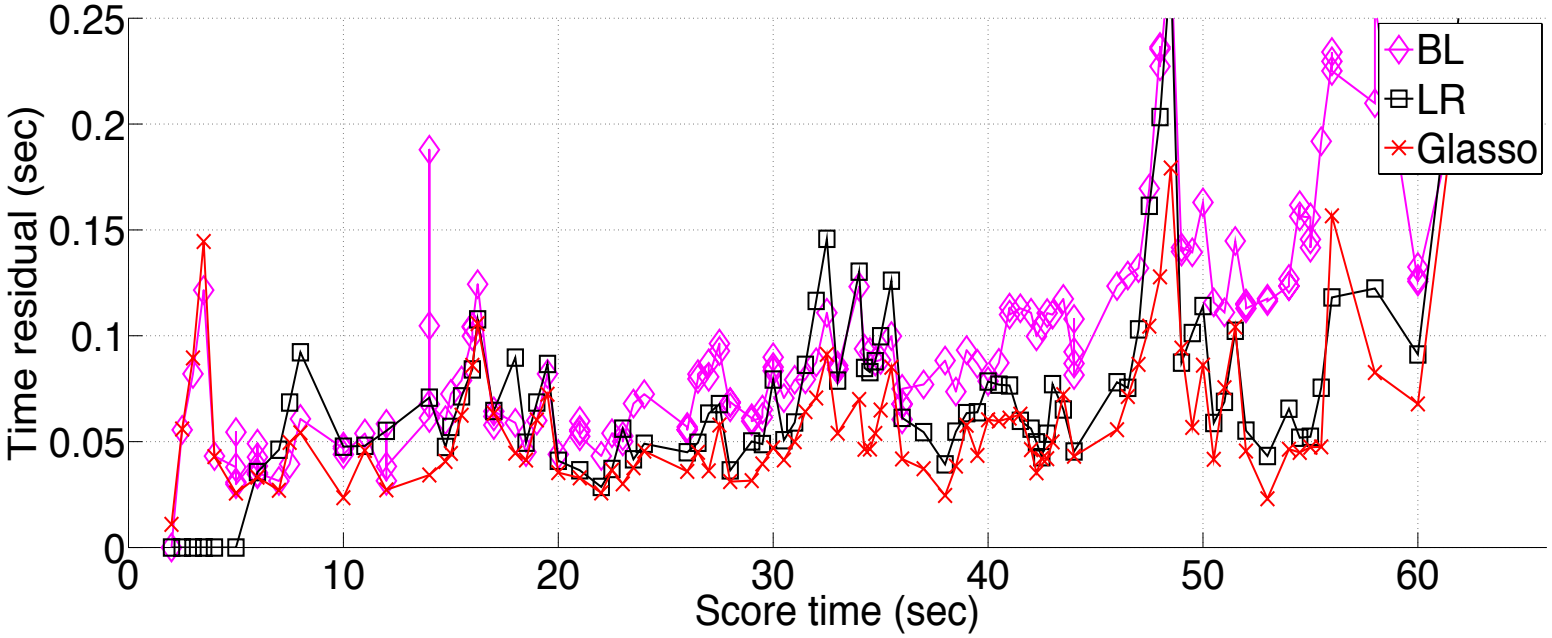
- Idea:
  - Reduces the burden for training.
  - Discover the dominant features that could predict the expressive timings.
- Solve:

$$\min_{B \in R^P} \left(\|Y - BU\|_2^2 + \lambda \sum_{l=1}^{L} \sqrt{p_l} \|B_l\|_2 \right)$$

Result: General feature approach

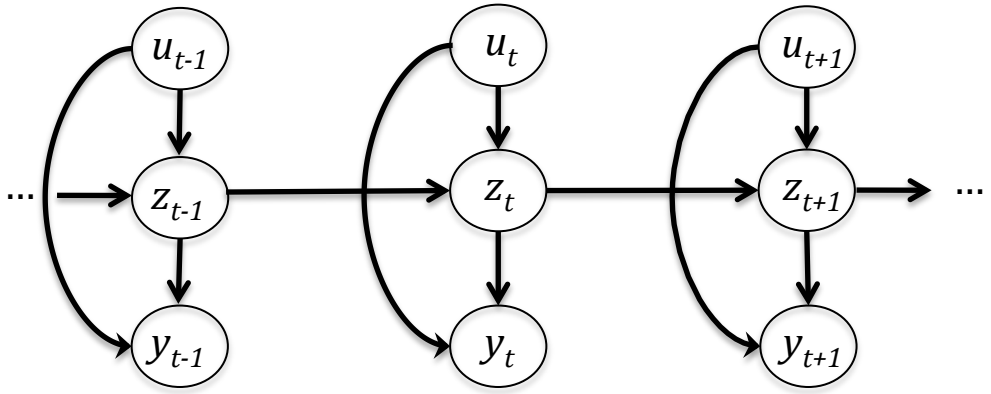
Mean Absolute Error: (ONLY 4 training pieces!)

BL: 0.098 LR: 0.072 Glasso: 0.059



Method (4): LDS approach

- Idea:
  - Add another regularization by adjacent notes.
  - Lower dimensional hidden mental states that control the expressive timings.



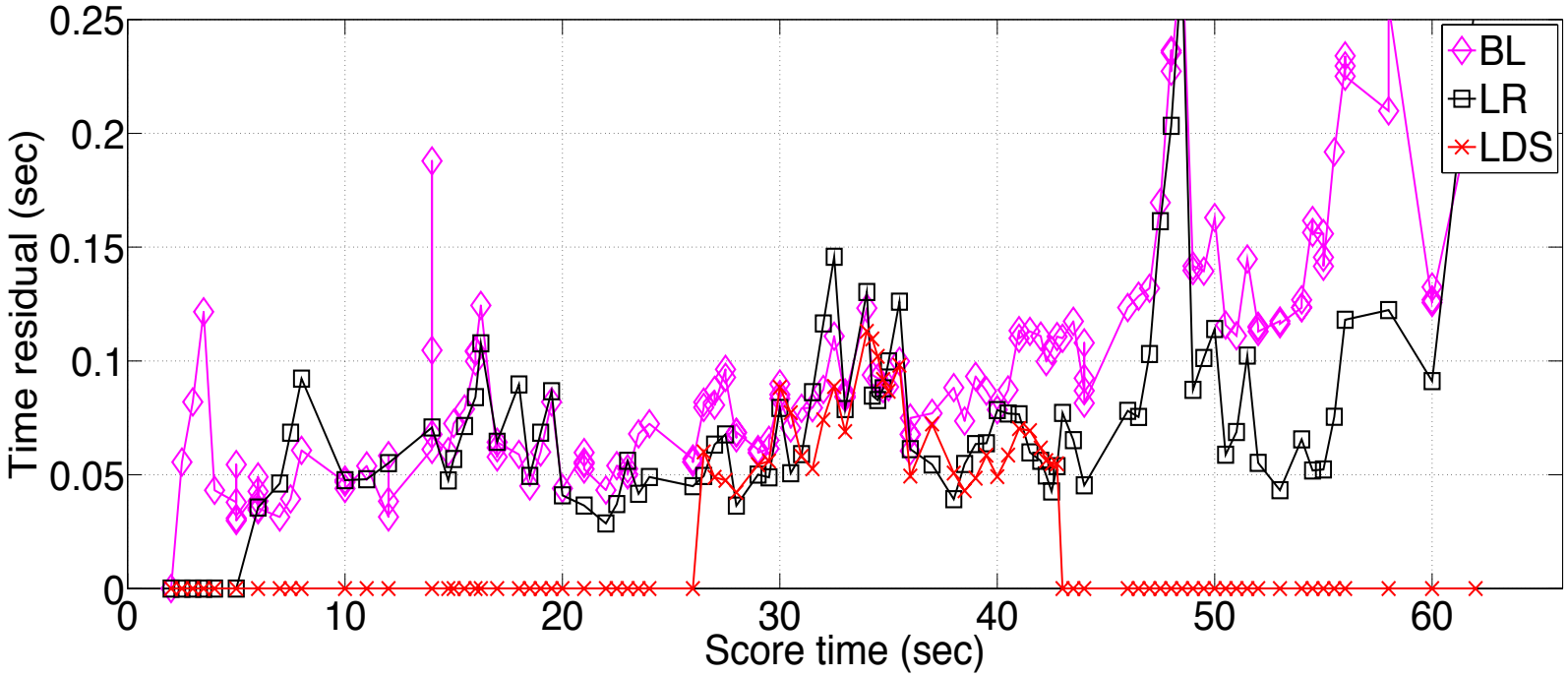
- Model:

$$z_t = Az_{t-1} + Bu_t + w_t \quad w_t \sim \mathcal{N}(0, Q)$$
$$y_t = Cz_t + Du_t + v_t \quad v_t \sim \mathcal{N}(0, R)$$

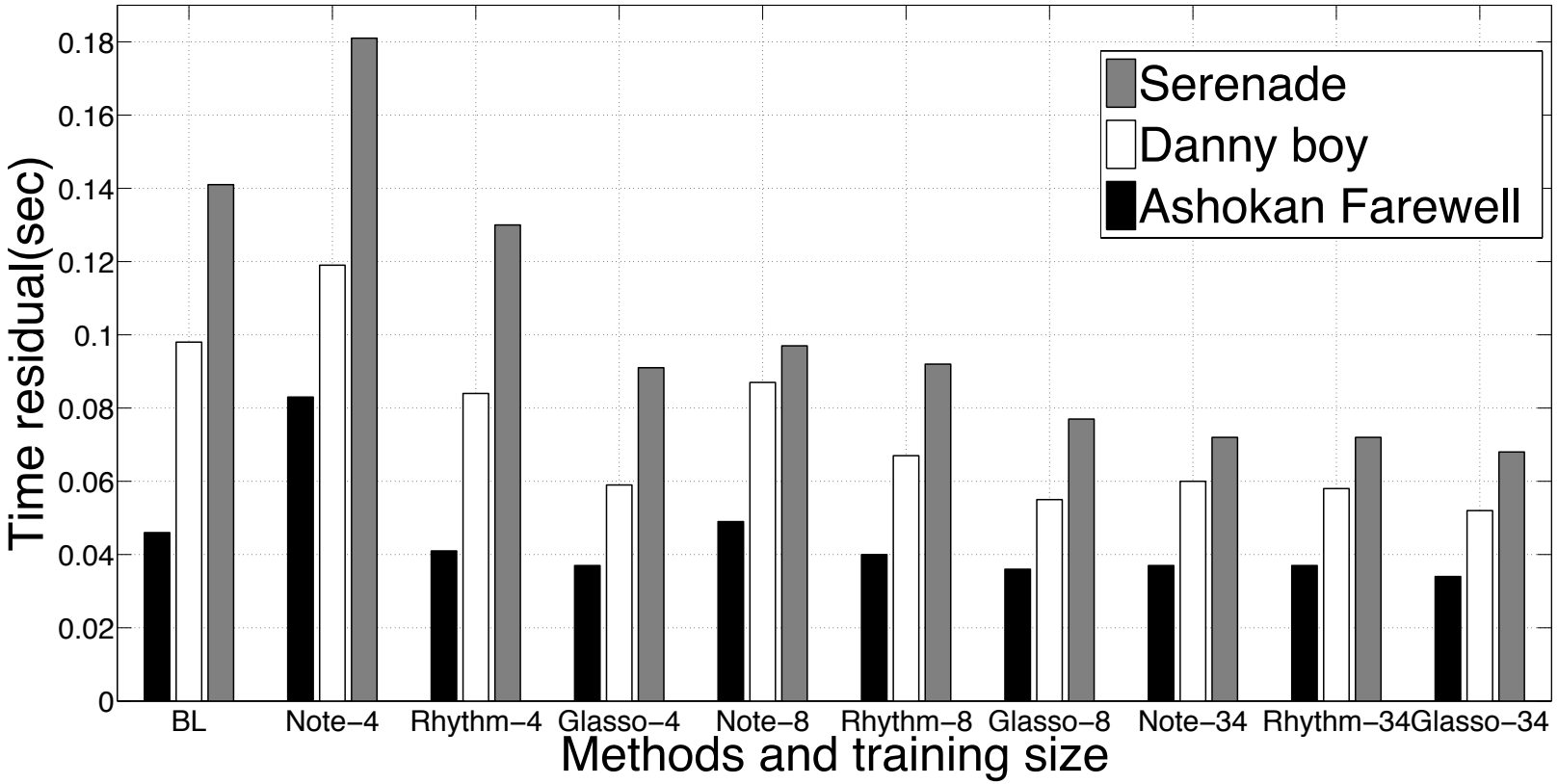
Result: LDS (horizontal regularization)

Mean Absolute Error: (ONLY 4 training pieces!)

BL: 0.085 LR: 0.072 LDS: 0.067



A Global View



Outline

- Introduction
- Data Collection
- Methods
- **Demos**
- **Conclusion & Future Work**

Some Initial Audio Demo

- Base Line:

- Note-specific approach: 34 training examples

- General feature approach, group lasso: 4 training examples

Future Work

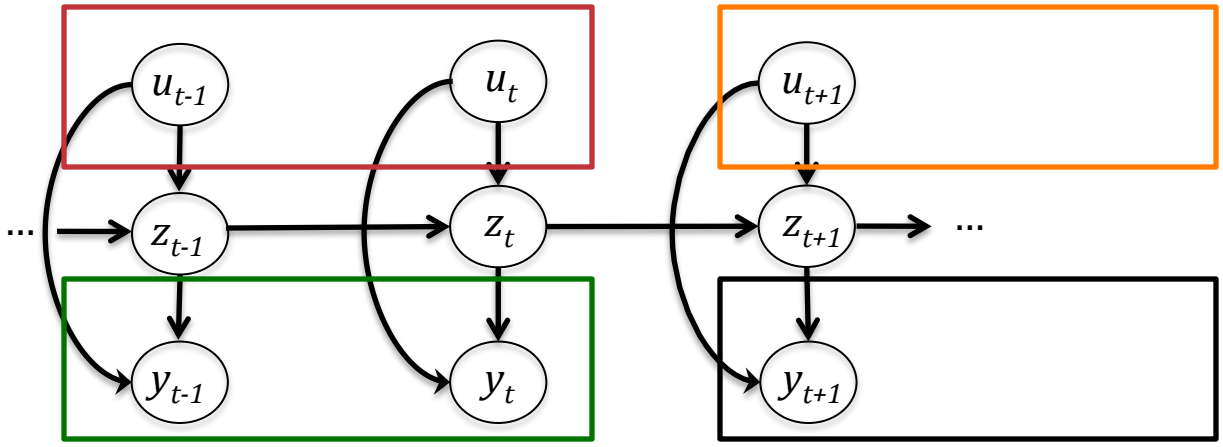
- Cross-piece models
- Performer-specific models
- Online learning and decoding
- Plugin with music robots

Conclusion

- An artificial performer for interactive performance
- Learn musicianship from rehearsal experience
- A combination of expressive performance and automatic accompaniment
- Much better prediction just based on 4 rehearsals

Q&A

Spectral Learning(1): Oblique projections

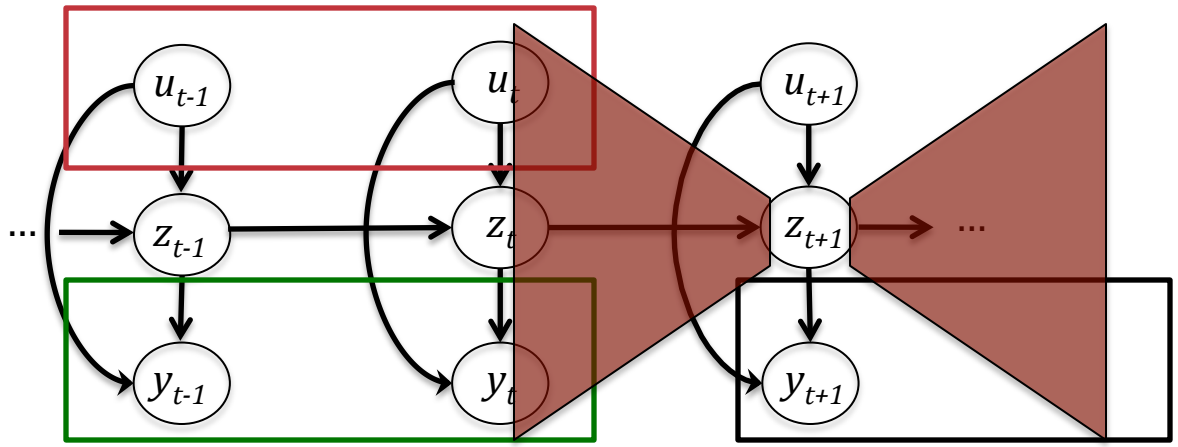


$$\mathbb{E}(Y_F) = [\beta_{Y_H}\ \beta_{U_H}\ \beta_{U_F}] \begin{bmatrix} Y_H \\ U_H \\ U_F \end{bmatrix}$$

- We don't know the future.
- Partially explain future observations based on the history

$$\hat{Y}_F \stackrel{\text{def}}{=} [\hat{\beta}_{Y_H}\ \hat{\beta}_{U_H}\ 0] \begin{bmatrix} Y_H \\ U_H \\ 0 \end{bmatrix}$$

Spectral Learning(2): state estimation



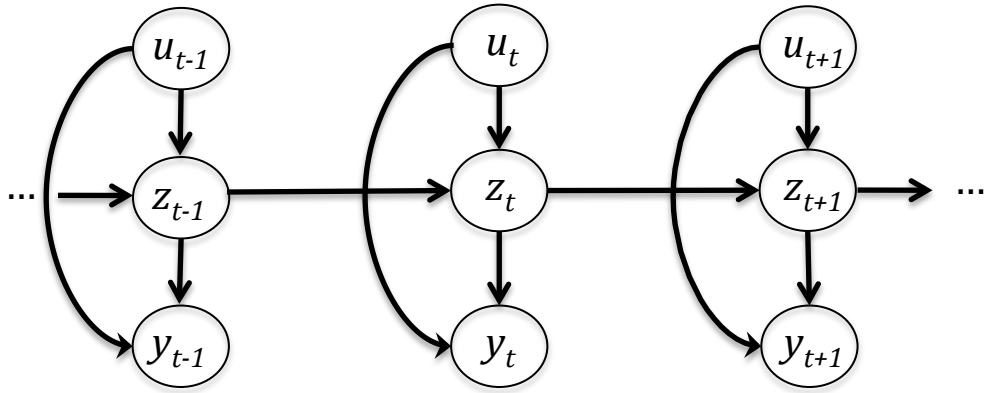
$$\mathcal{Y}_F = \Gamma_f Z_f \stackrel{\text{def}}{=} \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix} Z_f$$

- States estimation by SVD

$$\mathcal{Y}_F = \mathcal{U}\Sigma\mathcal{V}^T = (\mathcal{U}\Sigma^{\frac{1}{2}})(\Sigma^{\frac{1}{2}}\mathcal{V}^T)$$

- Moreover, enforce a bottleneck by throwing out near-zero singular values and corresponding columns in U and V.

Spectral Learning(3): Estimate parameter



$$z_{t+1} = Az_t + Bu_t + w_t$$
$$w_t \sim \mathcal{N}(0, Q)$$
$$y_t = Cz_t + Du_t + v_t$$
$$v_t \sim \mathcal{N}(0, R)$$

- Based on estimated hidden states, the parameters could be estimated from the following equation:

$$\begin{bmatrix} \hat{Z}_f^- \\ Y_f \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{Z}_f \\ U_f \end{bmatrix} + \begin{bmatrix} e_w \\ e_v \end{bmatrix}$$