Modeling Training Efficiency and Return on Investment in GIFT

Gregory A. Goodwin[1], Jong Kim[1], & James Niehaus[2]
U.S. Army Research Laboratory[1], Charles River Analytics[2]

GIFTSym 5, Orlando Fl
May 11, 2017

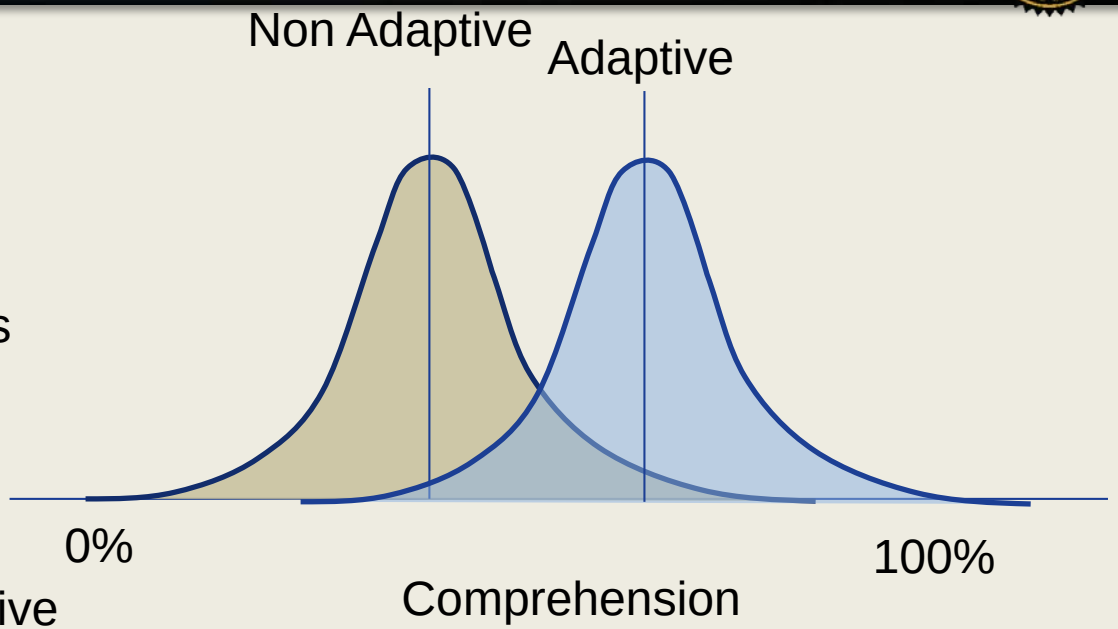
Overview

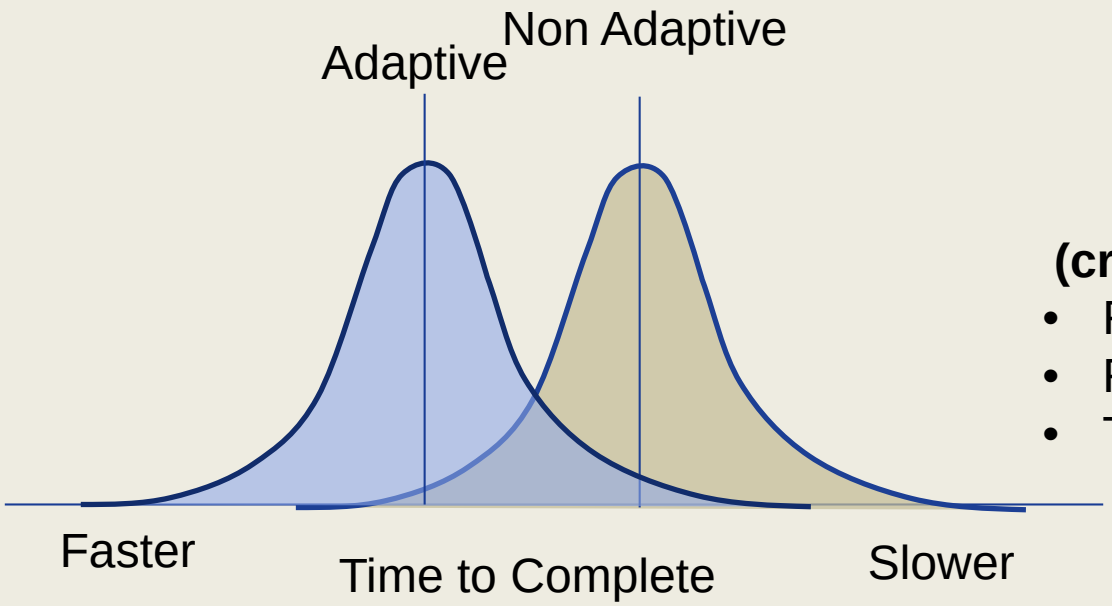
- Probabilistic Programming for Anticipated Simulation Training (PAST) Time
- Benefits of adaptive training
- Modeling time to train in GIFT
- Model Predictions
- Maximizing ROI
- Conclusions

Benefits of Adaptive Training

Training Effectiveness (time held constant)

- Accurate diagnosis of errors
- Targeted remediation
- Tailored training methods





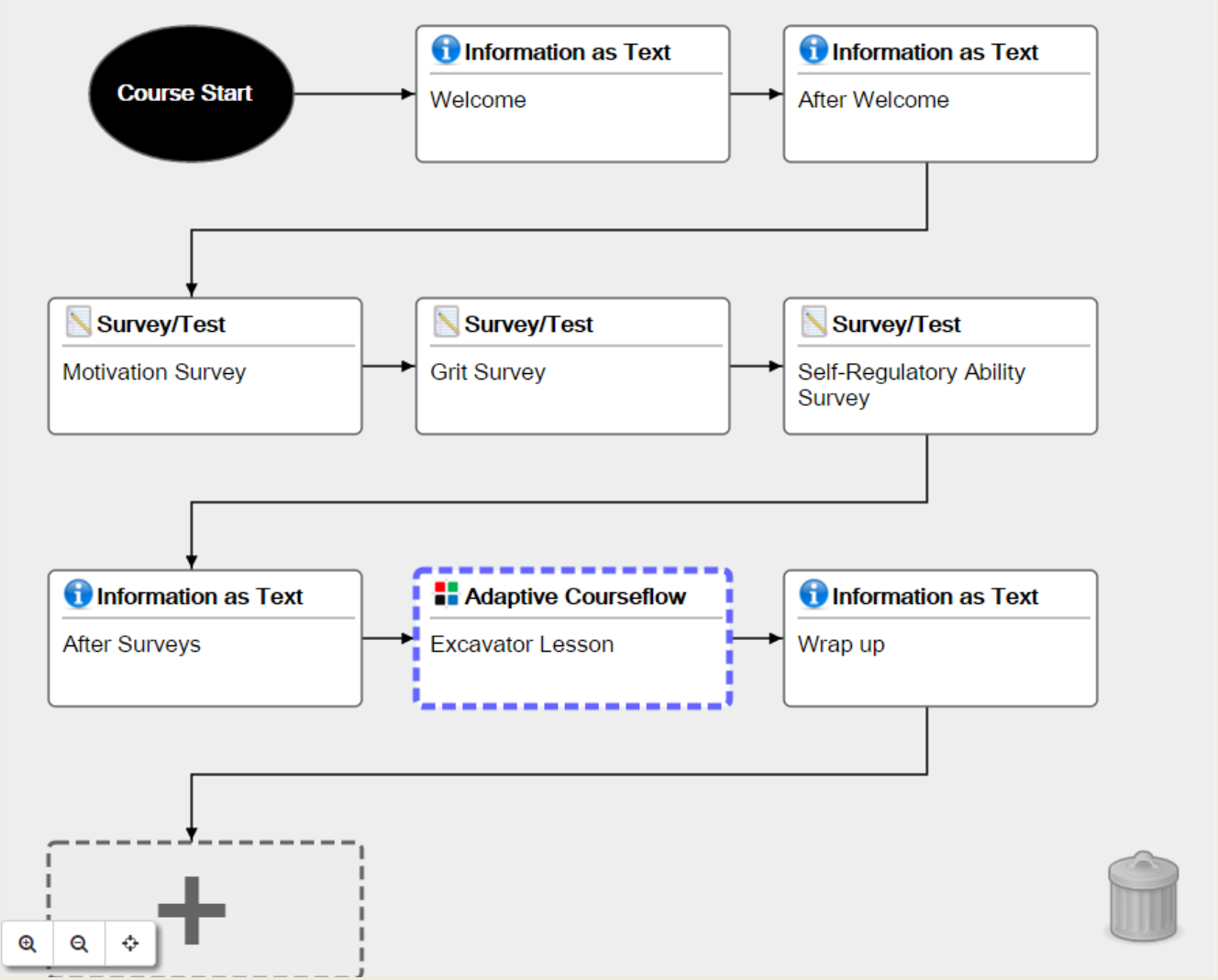
Training Efficiency (criterion held constant)

- Rapid diagnosis of errors
- Rapid remediation
- Tailored training content

Factors that Impact Training Time

- **Learner Factors**: Aptitude, reading speed, reading comprehension level, prior knowledge & experience
- **Content Factors**: Number of words, number of images, content difficulty, test characteristics, etc.
- **Instructional Factors**: training methods and techniques

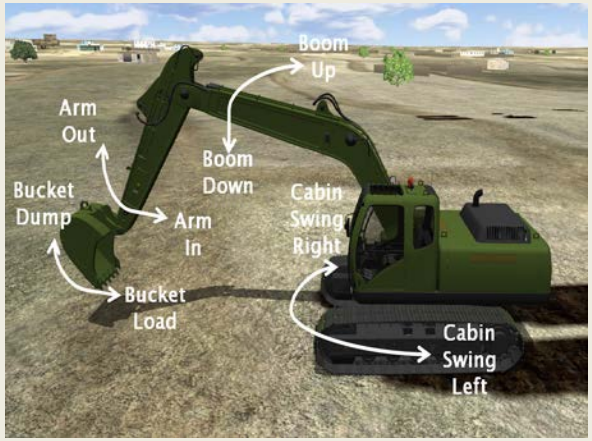
Excavator Trainer Course Map



Adaptive Courseflow Object

Rules

Here are the excavator components and their movements



Examples

This is how you move the bucket.

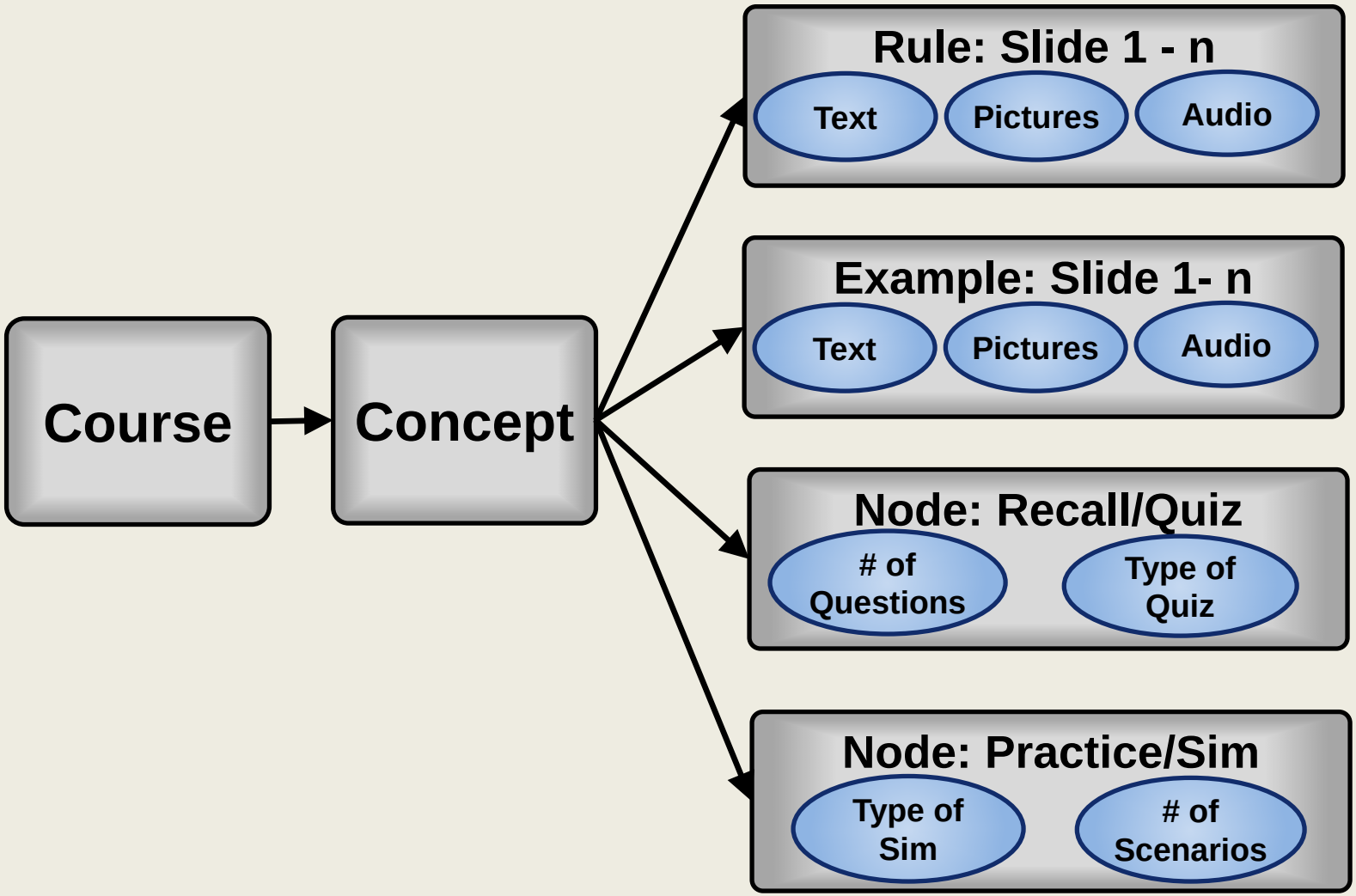
Recall

Which control is labeled “D” on the Excavator?

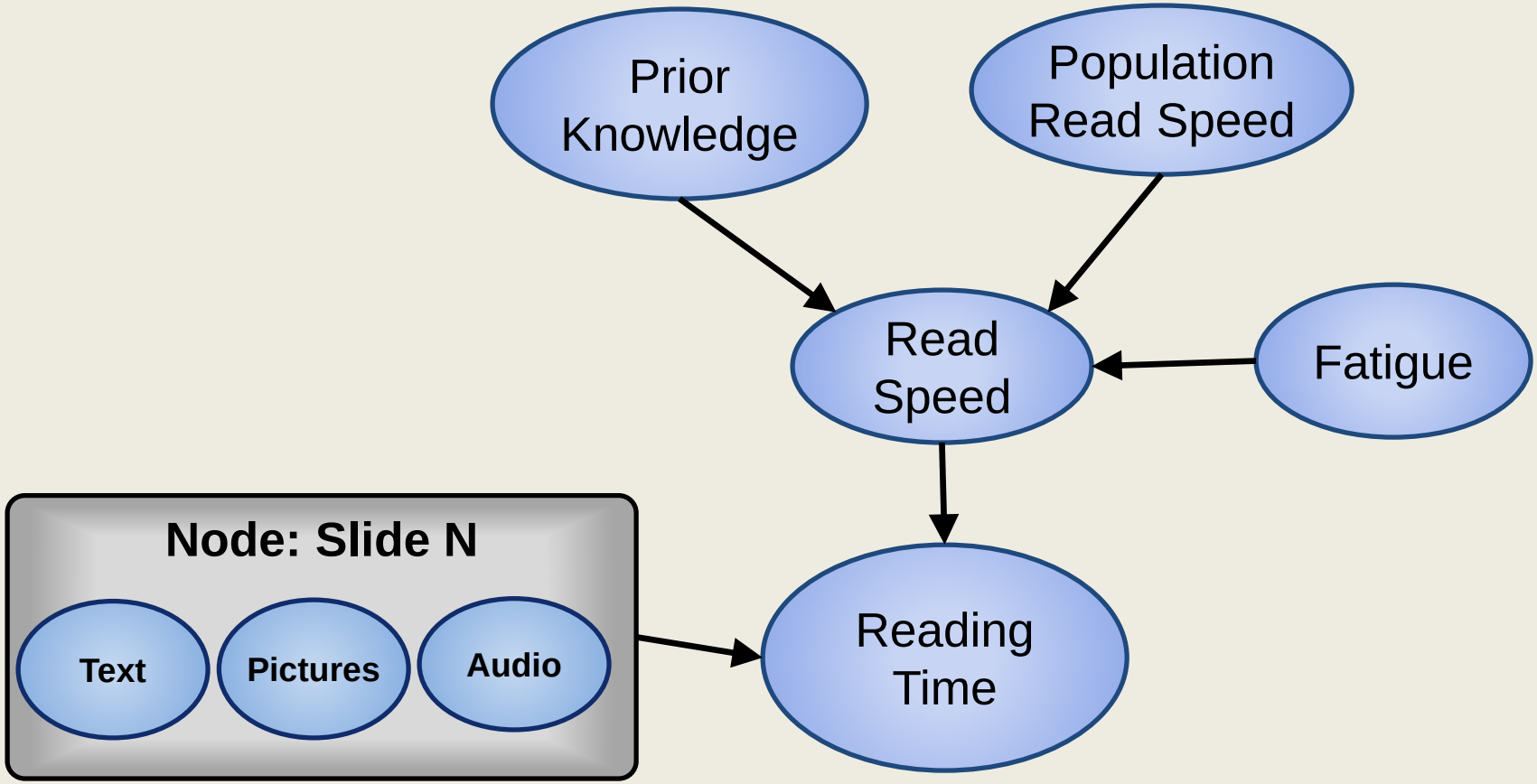
Practice

Complete tasks in virtual environment

Representation of Course Content and Factors that Influence Completion Time



Simple Model for Modeling Time to Read



Figaro Probabilistic Programming Language

1,000 characters of text

0.2 probability of prior knowledge

Population reading speed: normally distributed @100 characters/min

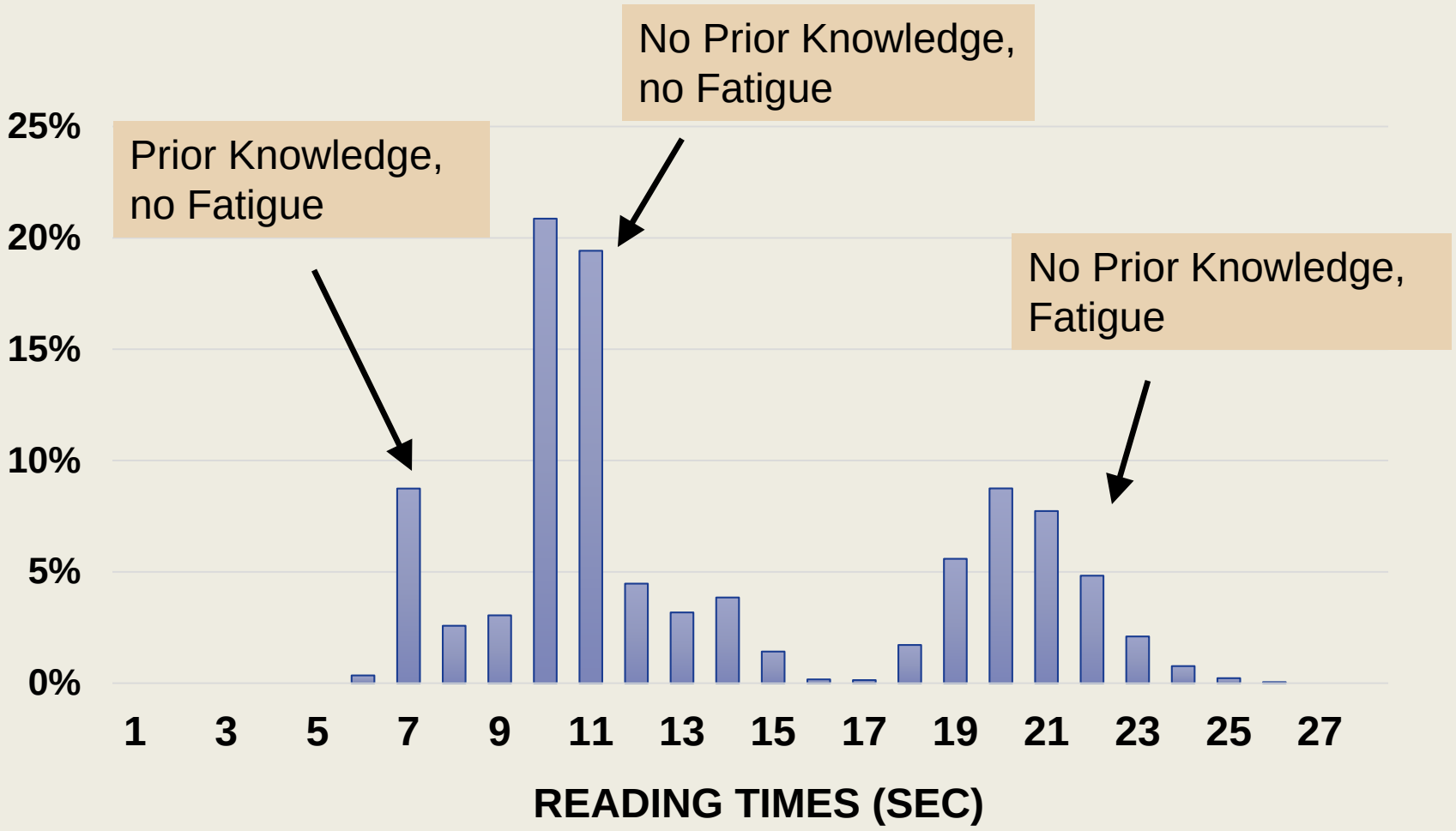
```
val text = Constant(1000.0)
  val priorKnowledge = Flip(0.2)
  val populationReadSpeed = Normal(100.0, 50.0)
  val readSpeed = If(priorKnowledge,
    populationReadSpeed ++ Normal(50.0, 25.0), populationReadSpeed)
  val fatigued = Flip(0.4)
  val readingTime = If(fatigued,
    text / (readSpeed * Constant(0.5)), text / readSpeed)

  val algorithm = Importance(10000, readingTime)
  algorithm.start
  println(algorithm.distribution(readingTime))
```

Prior knowledge increases around 50 char/min

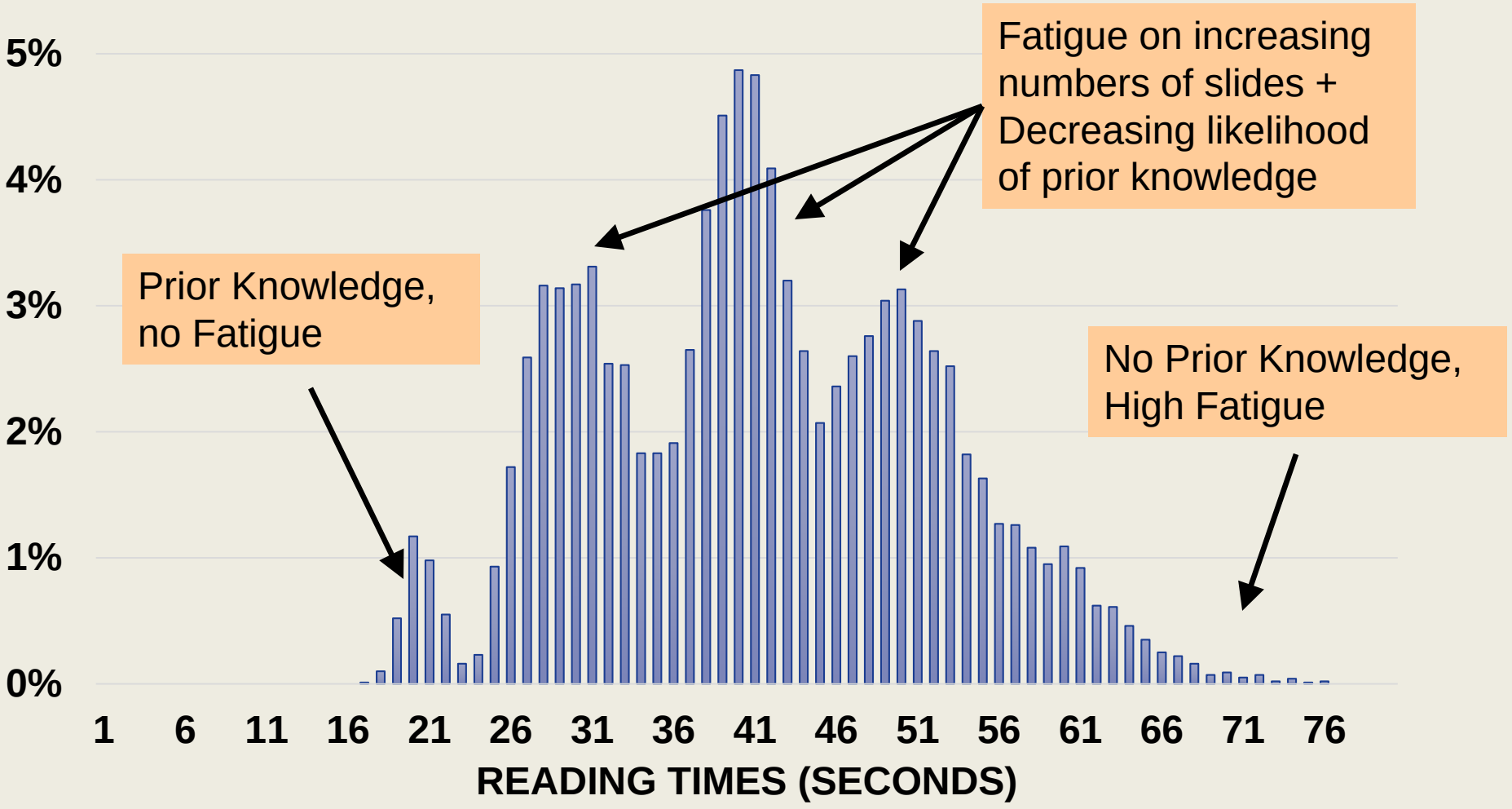
Learner reading speed

Effect of fatigue – decrease speed by 50%

Reading Time 2 Slides

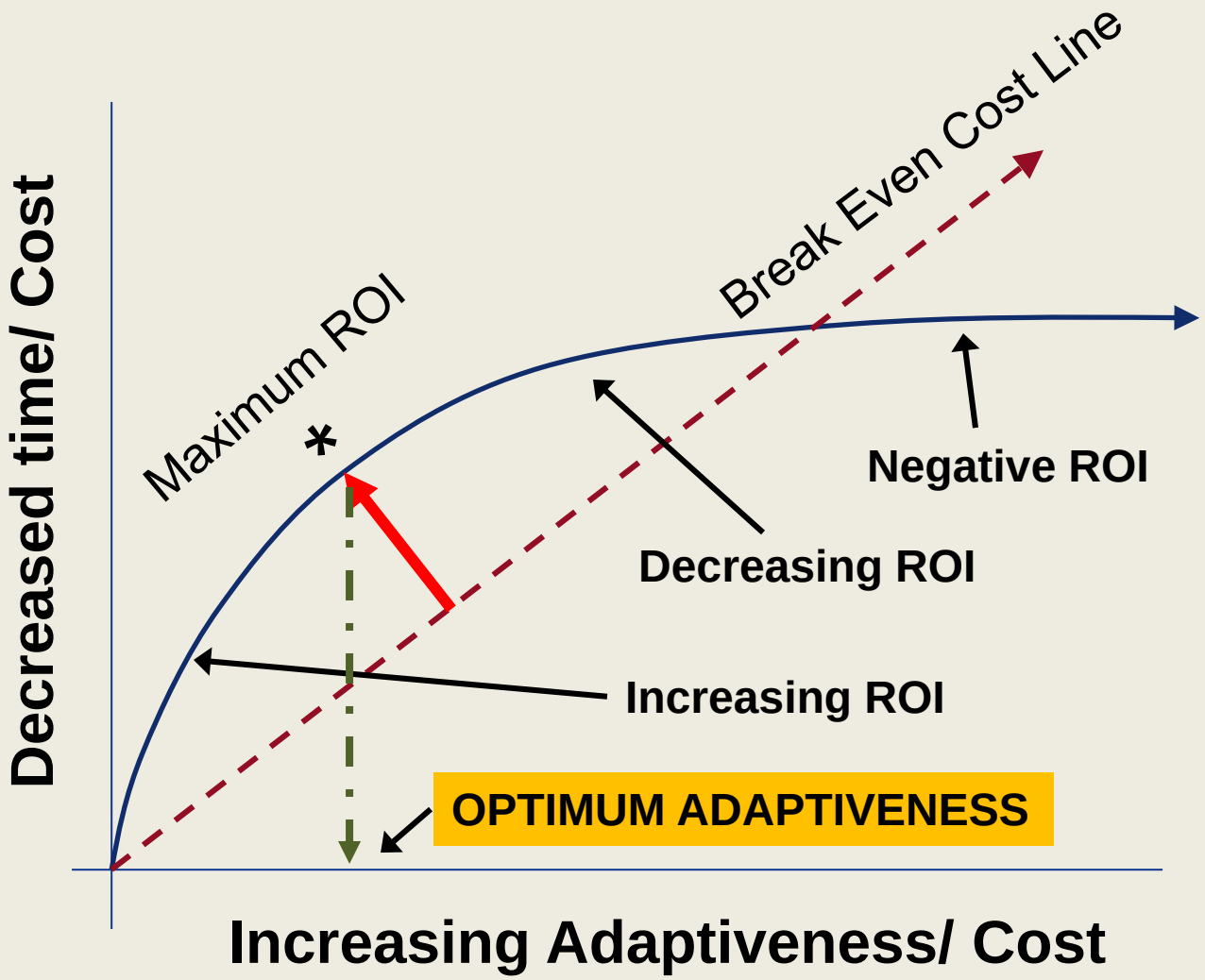


Reading Time: 3 Slides



Probability of fatigue increases 5% with each slide

Maximizing ROI



Conclusions

- Calculating ROI is difficult for training – examining time saved is a simple but easy metric to consider focused on the cost of training delivery.
- Future work will validate the predictive model with learner data.
- Other applications of the model include :
  - Authoring: predict the benefit of additional adaptation
  - Run time monitoring: identify anomalous students in need of intervention.