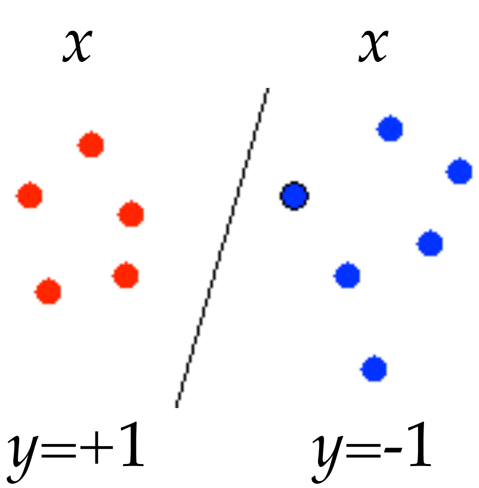
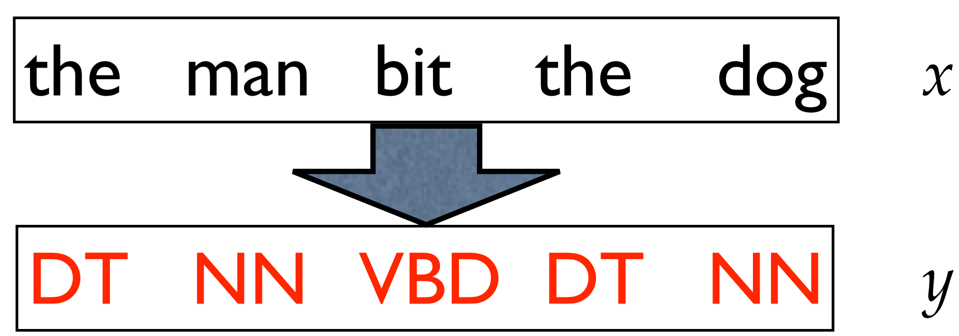
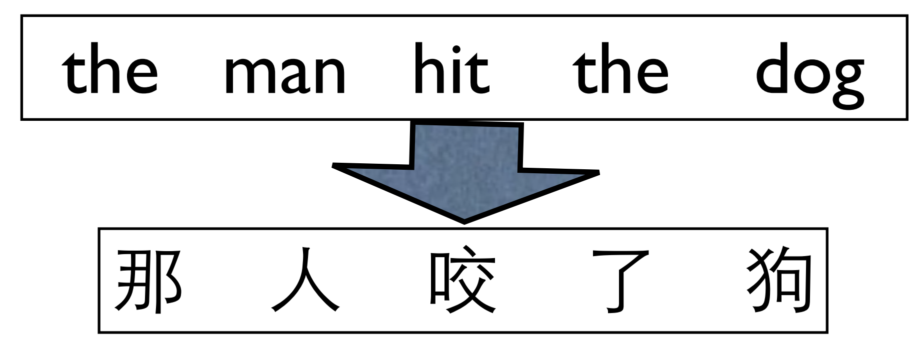
Structured Prediction with Perceptron: Theory and Algorithms





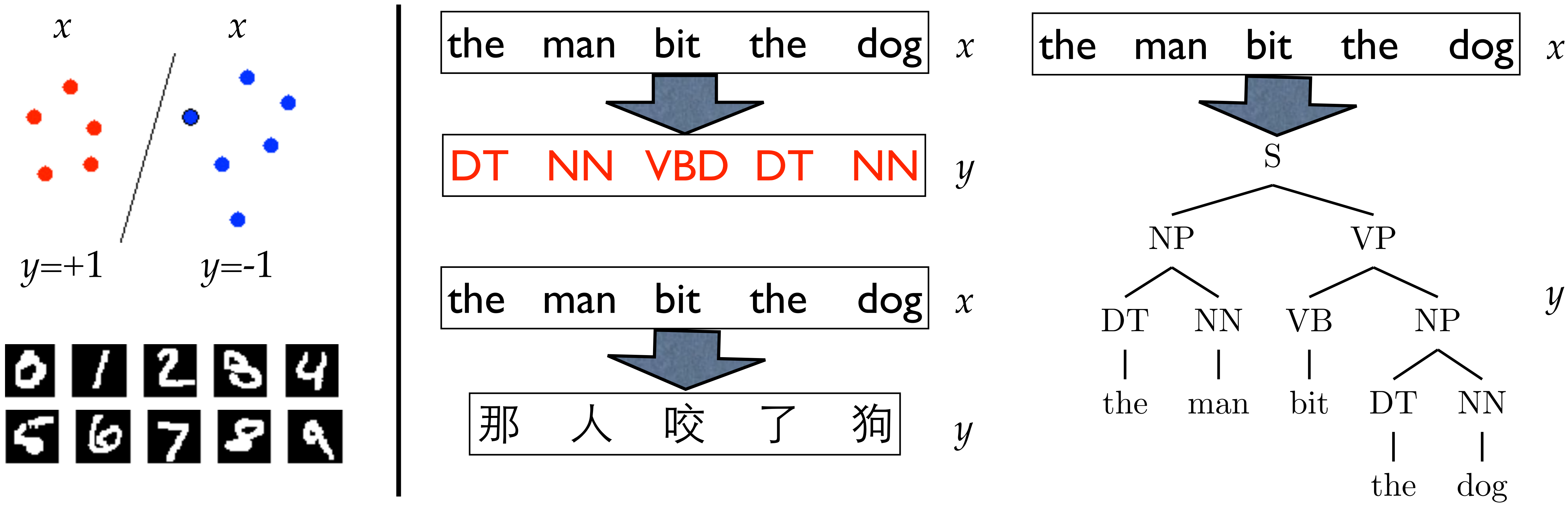


Kai Zhao   Liang Huang

Graduate Center, the City University of New York

slides at: http://kaizhao.me

What is Structured Prediction?

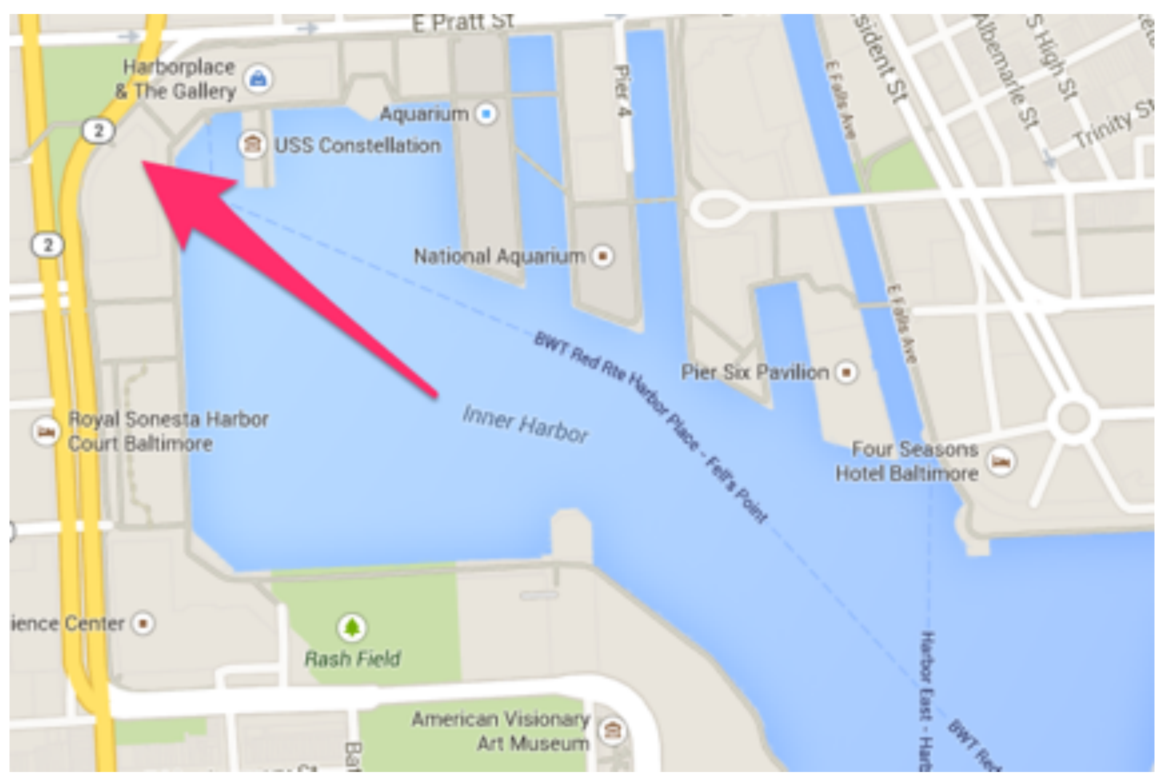


- bin
- mu

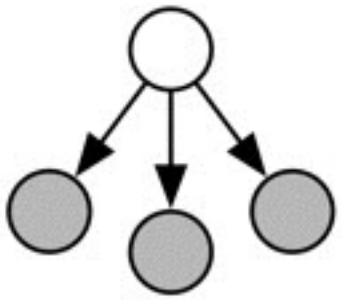
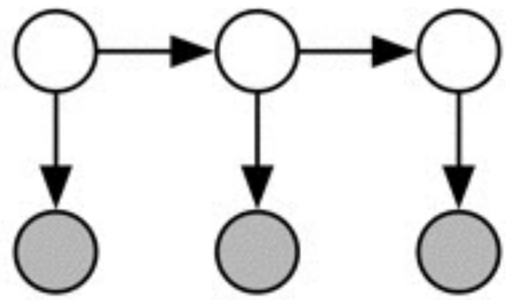
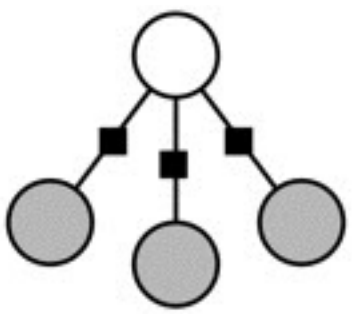
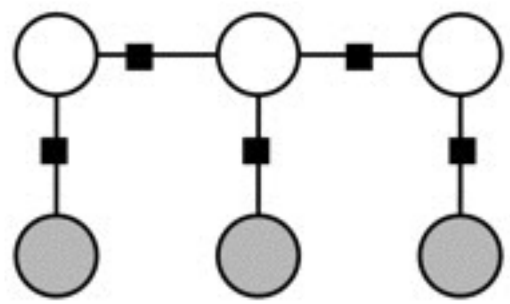
NLP is all about structured prediction!

- structured classification: output is a structure (seq., tree, graph)
  - part-of-speech tagging, parsing, summarization, translation
  - exponentially many classes: *search* (inference) efficiency is crucial!

An Example of Bad Structured Prediction

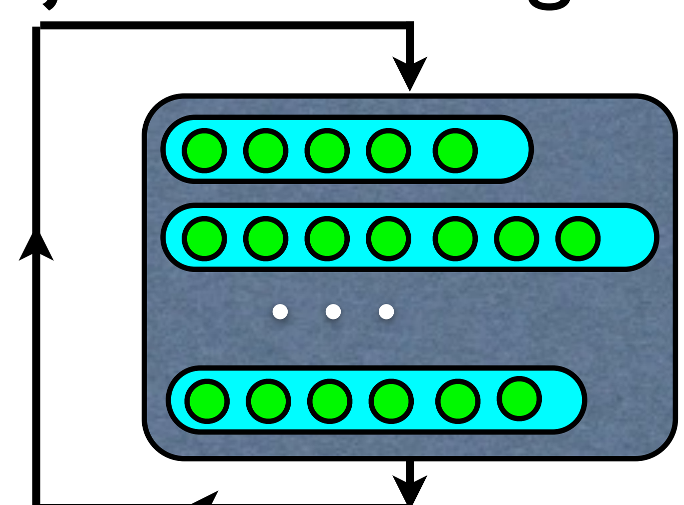


Learning: Unstructured vs. Structured

binary/multiclass		structured learning		
 naive bayes	→	 HMMs	generative	(count & divide)
↓ Conditional		↓ Conditional	discriminative	
 logistic regression (maxent)	→	 CRFs		(expectations)
↓ Online+ Viterbi		↓ Online+ Viterbi		
perceptron	→	structured perceptron		(argmax)
↓ max margin		↓ max margin		
SVM	→	structured SVM		(loss-augmented argmax)

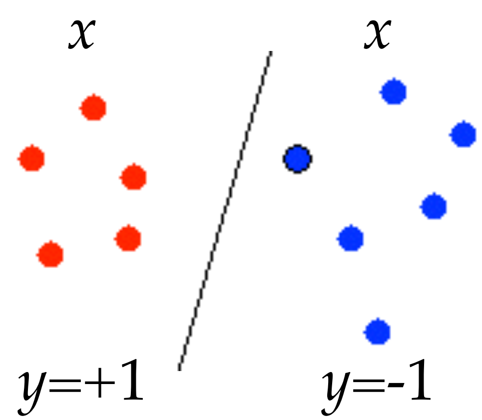
Why Perceptron (Online Learning)?

- because we want scalability on big data!
- learning time has to be linear in the number of examples
  - can make only constant number of passes over training data
  - only online learning (perceptron/MIRA) can guarantee this!
  - SVM scales between $O(n^2)$ and $O(n^3)$; CRF no guarantee
- and inference on each example must be super fast
  - another advantage of perceptron: just need argmax

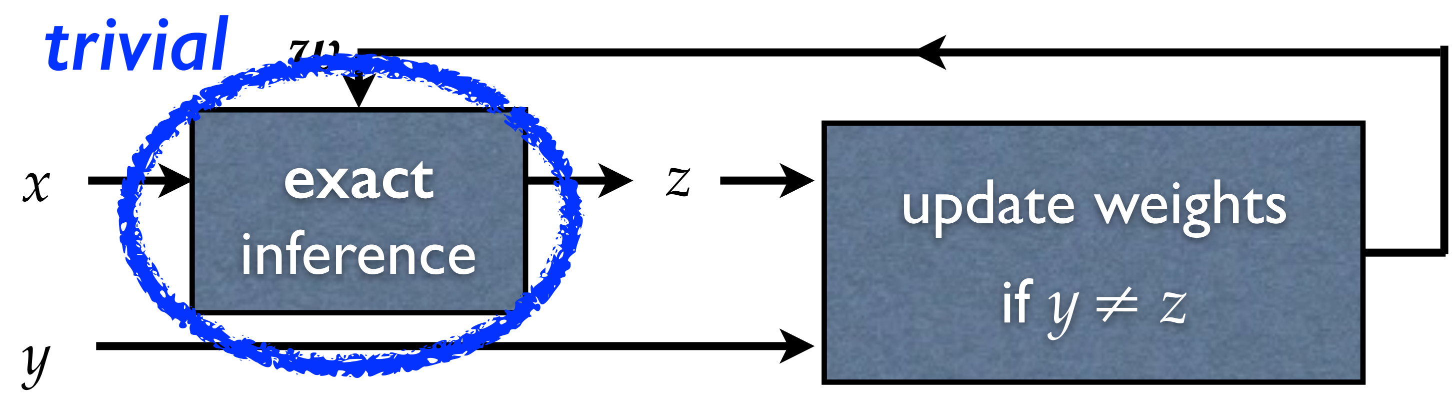


Perceptron: from binary to structured

binary perceptron
(Rosenblatt, 1959)

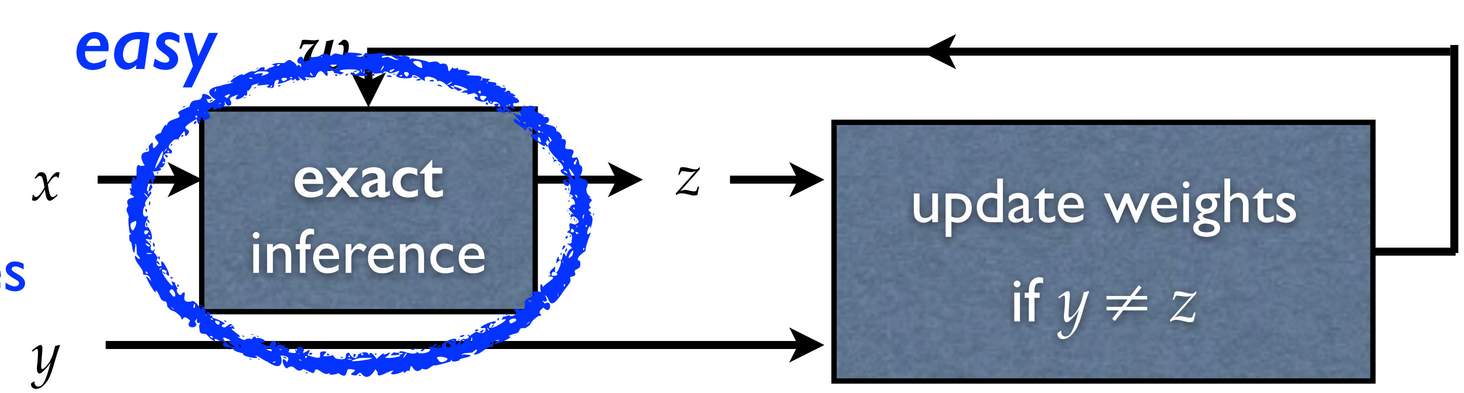


2 classes



multiclass perceptron
(Freund/Schapire, 1999)

constant
# of classes



structured perceptron
(Collins, 2002)

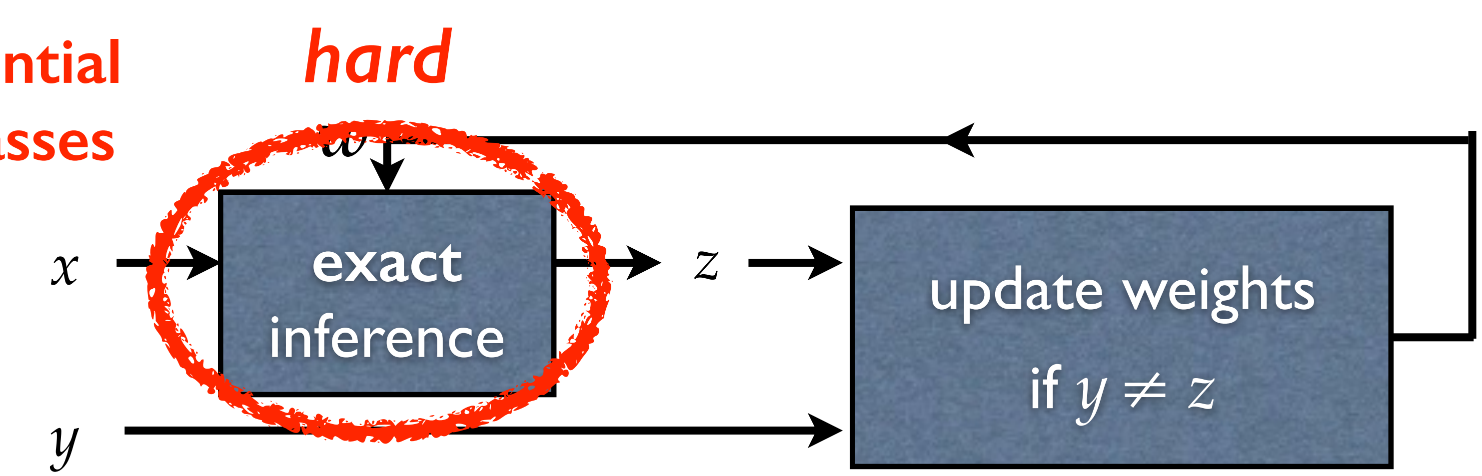
the	man	bit	the	dog

x

那	人	咬	了	狗

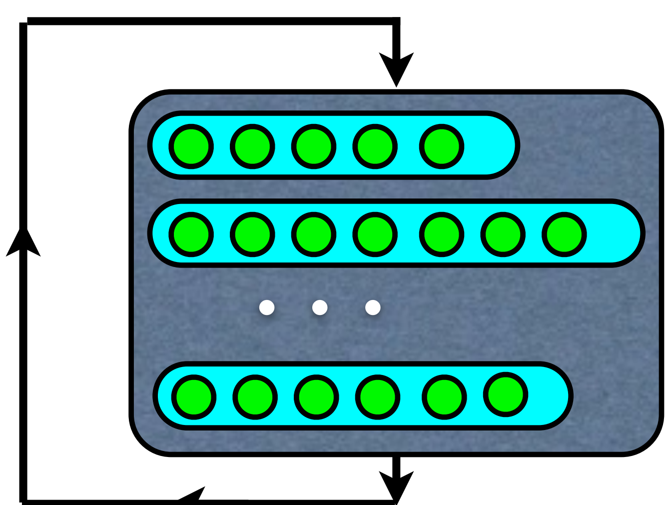
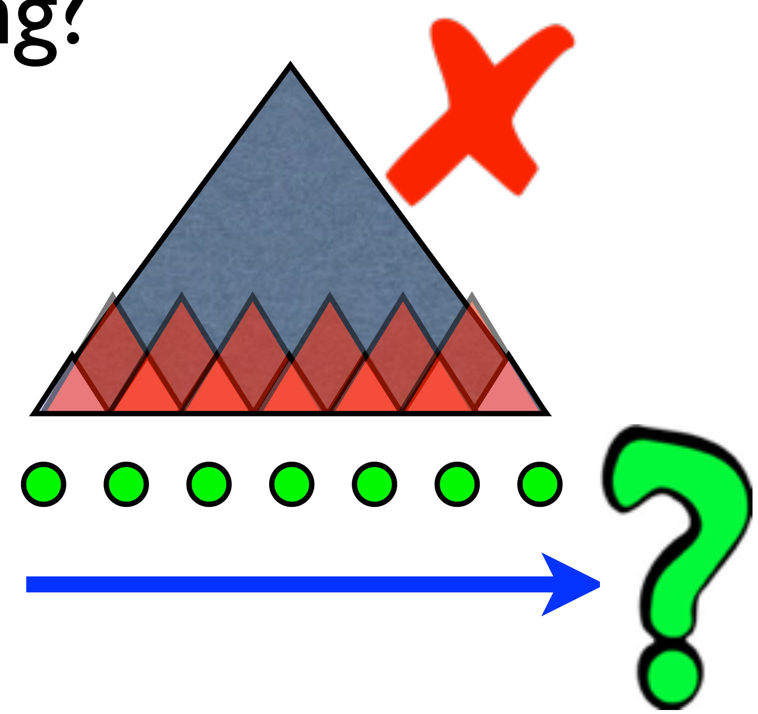
y

exponential
# of classes



Scalability Challenges

- inference (on one example) is too slow (even w/ DP)
  - can we sacrifice search exactness for faster learning?
  - would inexact search interfere with learning?
  - if so, how should we modify learning?

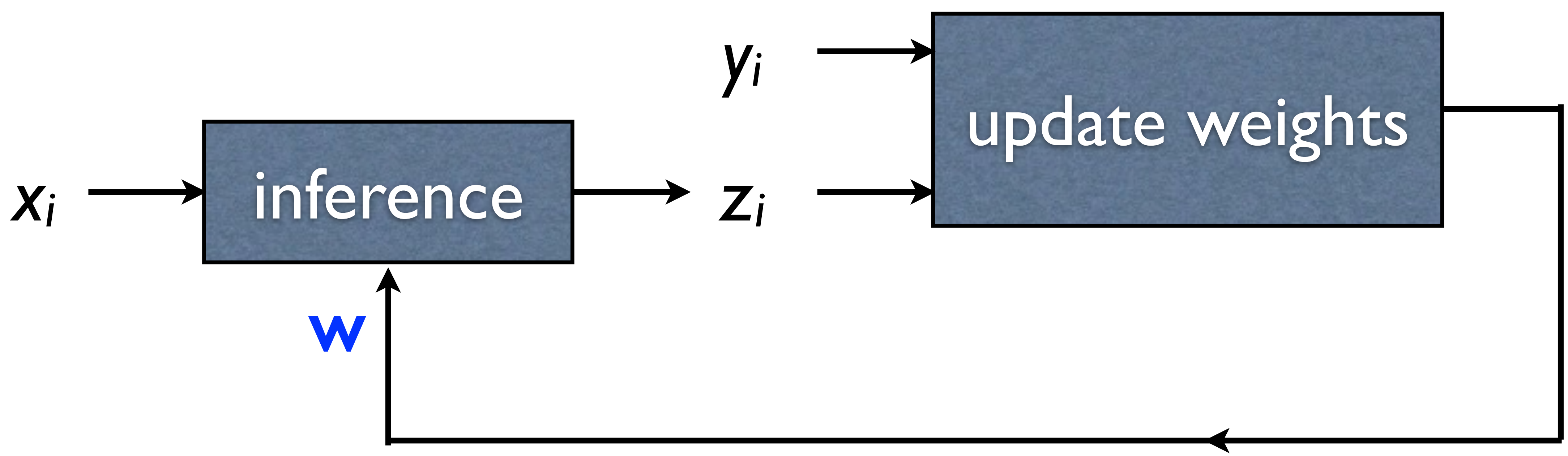



Outline

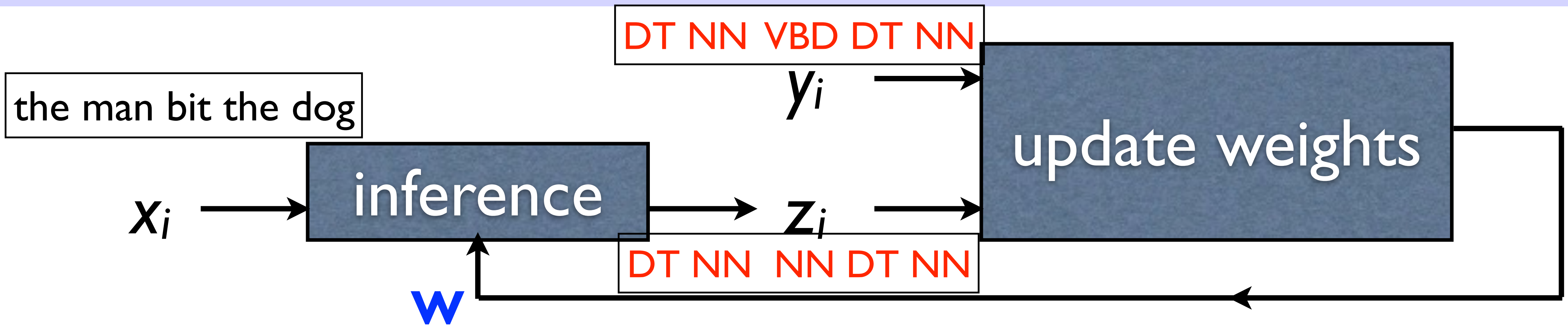
- Overview of Structured Learning
  - Challenges in Scalability
- Structured Perceptron
  - convergence proof
- Structured Perceptron with Inexact Search
- Latent-Variable Structured Perceptron

Generic Perceptron

- perceptron is the simplest machine learning algorithm
- online-learning: one example at a time
- learning by doing
  - find the best output under the current weights
  - update weights at mistakes



Structured Perceptron



Inputs: Training set (x_i, y_i) for $i = 1 \dots n$

Initialization: $\mathbf{W} = 0$

Define: $F(x) = \boxed{\operatorname{argmax}_{y \in \mathbf{GEN}(x)}} \mathbf{\Phi}(x, y) \cdot \mathbf{W}$

Algorithm: For $t = 1 \dots T, i = 1 \dots n$

$z_i = F(x_i)$

If $(z_i \neq y_i)$ $\boxed{\mathbf{W} \leftarrow \mathbf{W} + \mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, z_i)}$

Output: Parameters $\mathbf{W}$

Example: POS Tagging

- gold-standard: DT NN VBD DT NN y
  - the man bit the dog x

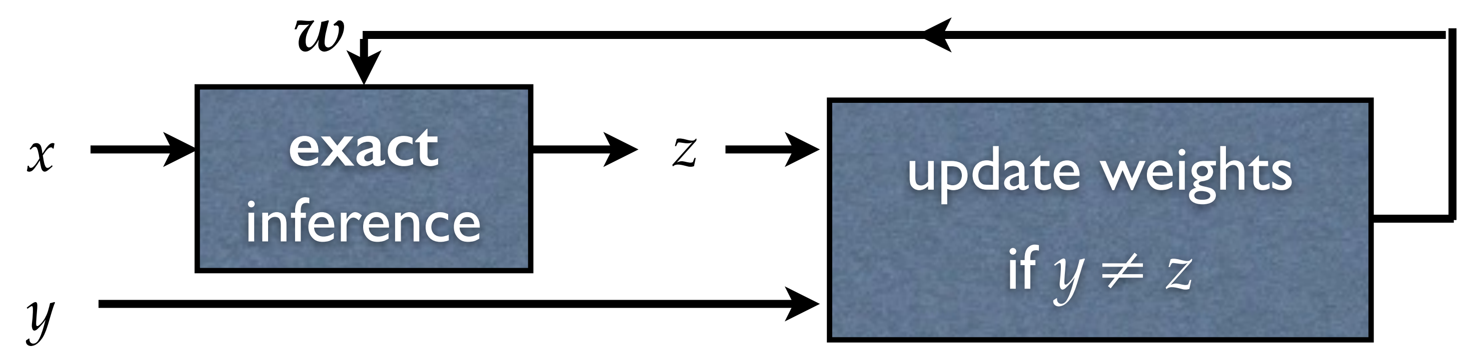
$\Phi(x, y)$

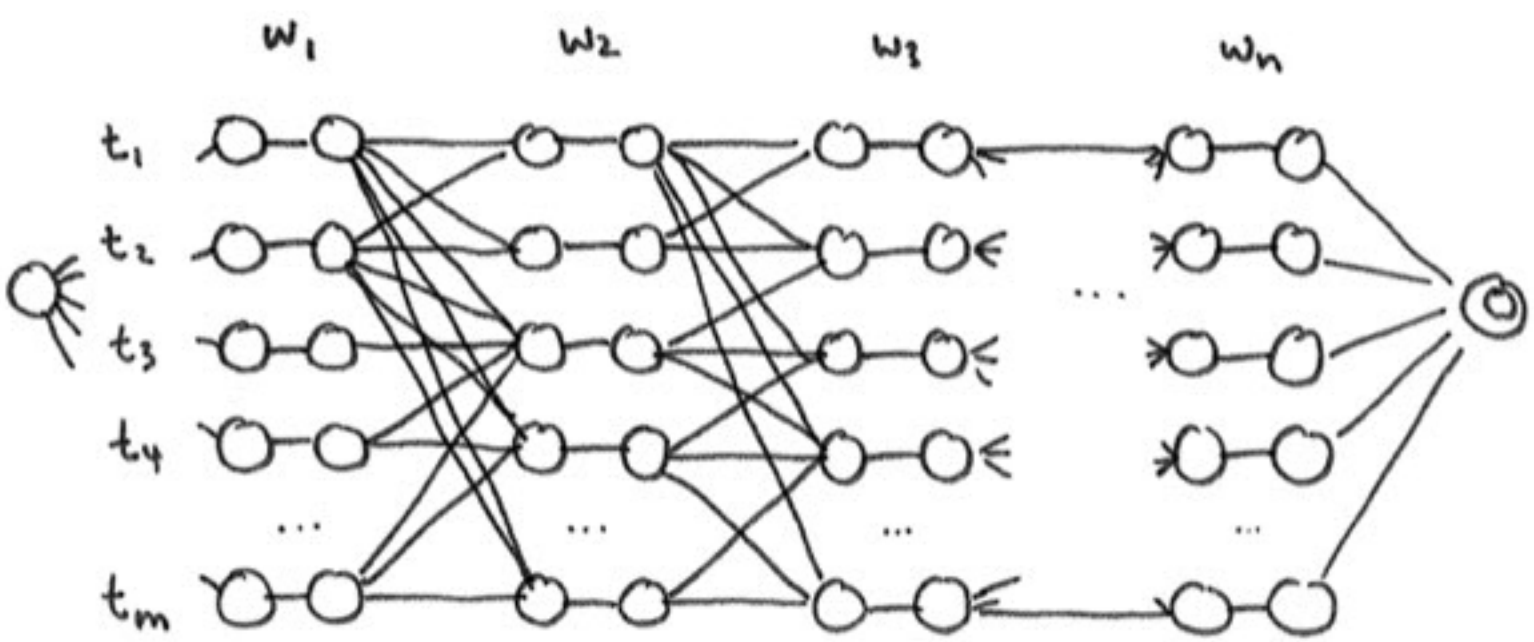
- current output: DT NN NN DT NN z
  - the man bit the dog x

$\Phi(x, z)$

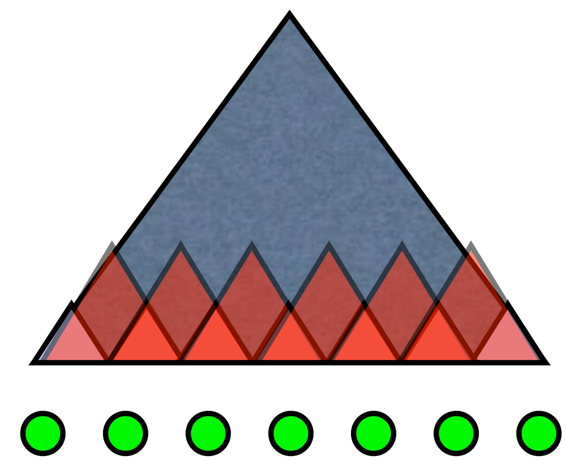
- assume only two feature classes
  - tag bigrams t_{i-1} t_i
  - word/tag pairs w_i
- weights ++: (NN, VBD) (VBD, DT) (VBD → bit)
- weights −−: (NN, NN) (NN, DT) (NN → bit)

Inference: Dynamic Programming





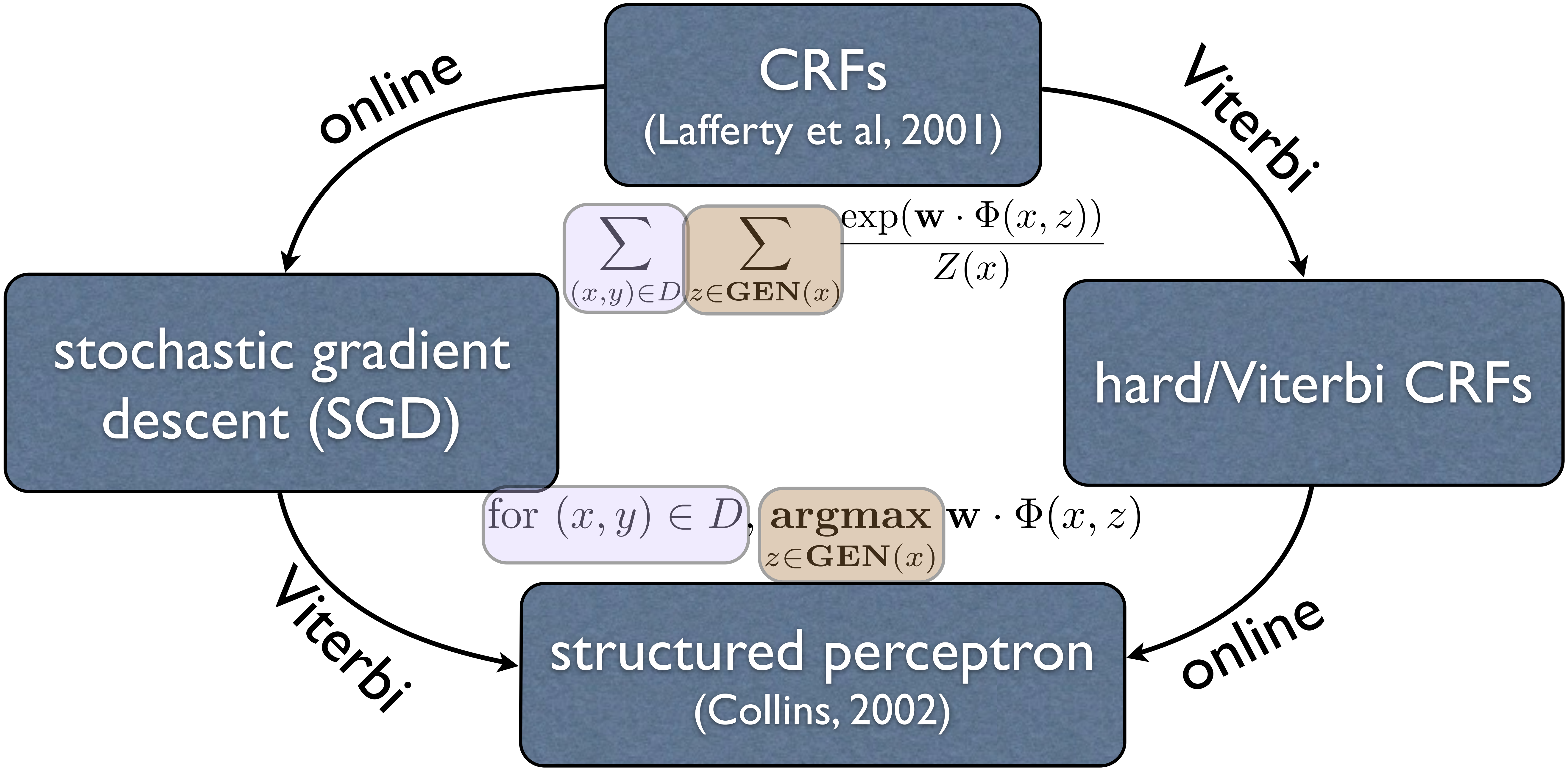
tagging: $O(nT^3)$



CKY parsing: $O(n^3)$

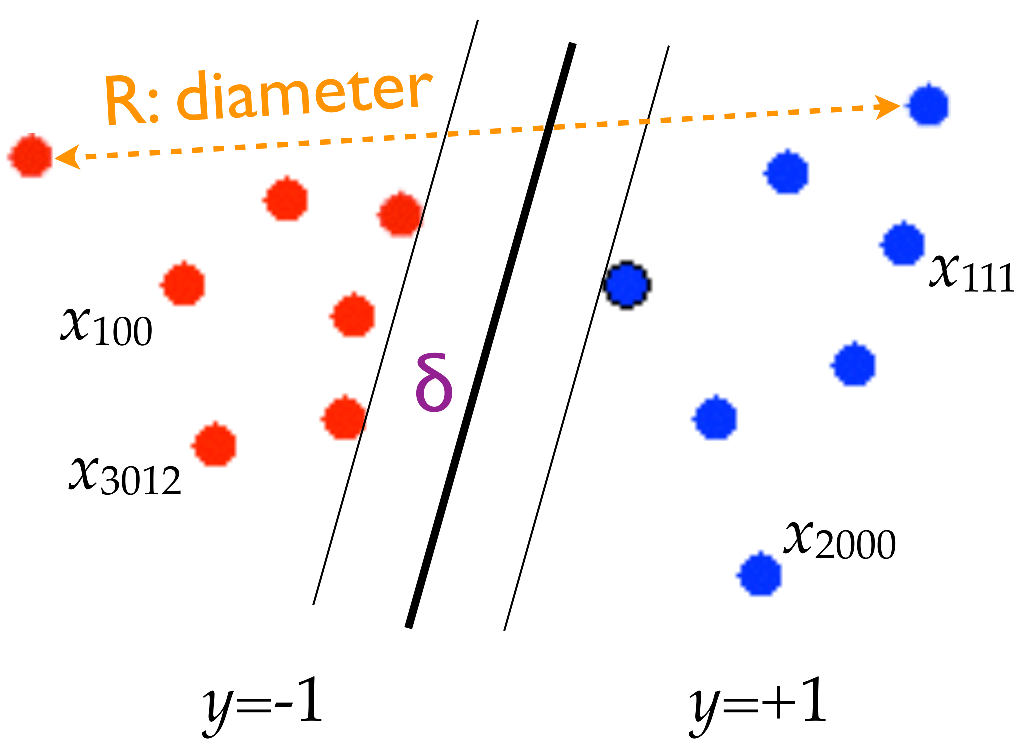
Perceptron vs. CRFs

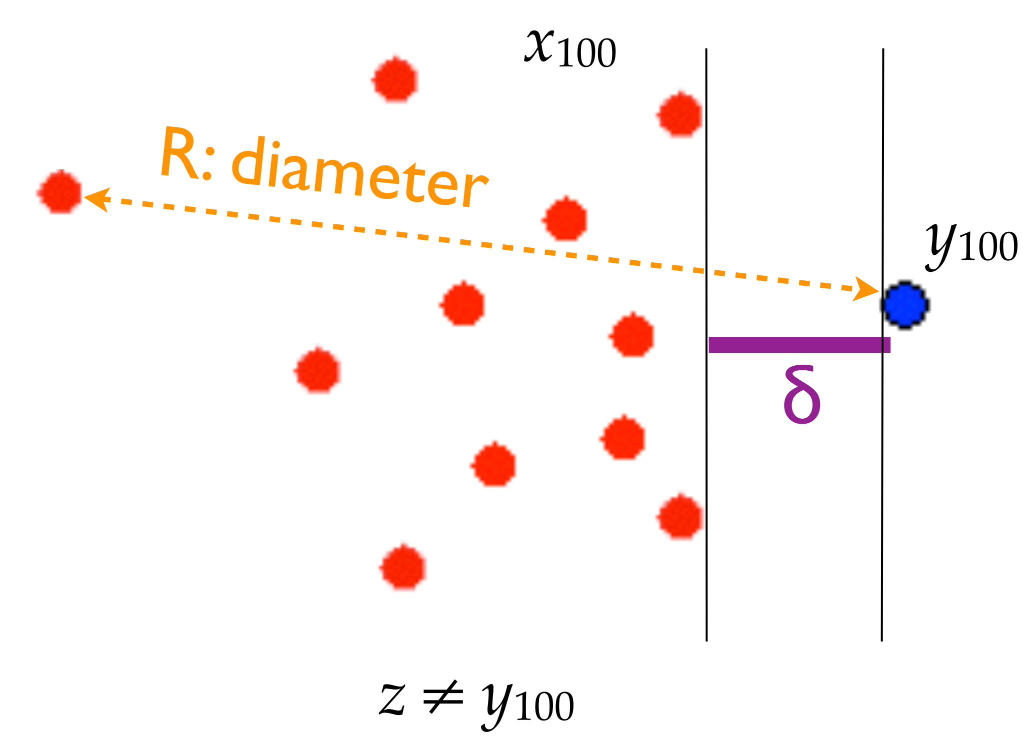
- perceptron is online and Viterbi approximation of CRF
- simpler to code; faster to converge; ~same accuracy



Perceptron Convergence Proof

- binary classification: converges iff. data is separable
- structured prediction: converges iff. data is separable
  - there is an oracle vector that correctly labels all examples
  - one vs the rest (correct label better than all incorrect labels)
- theorem: if separable, then **# of updates ≤** R^2 / δ^2 R: diameter

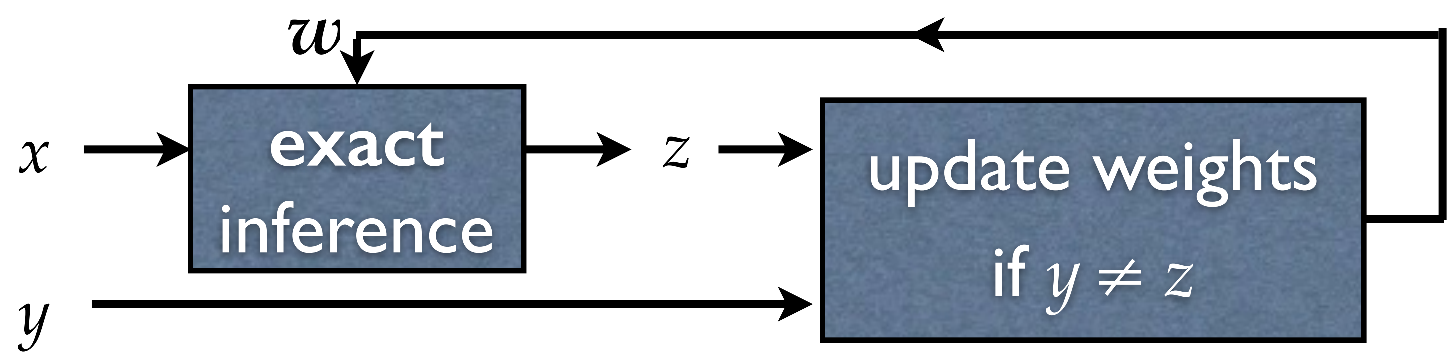


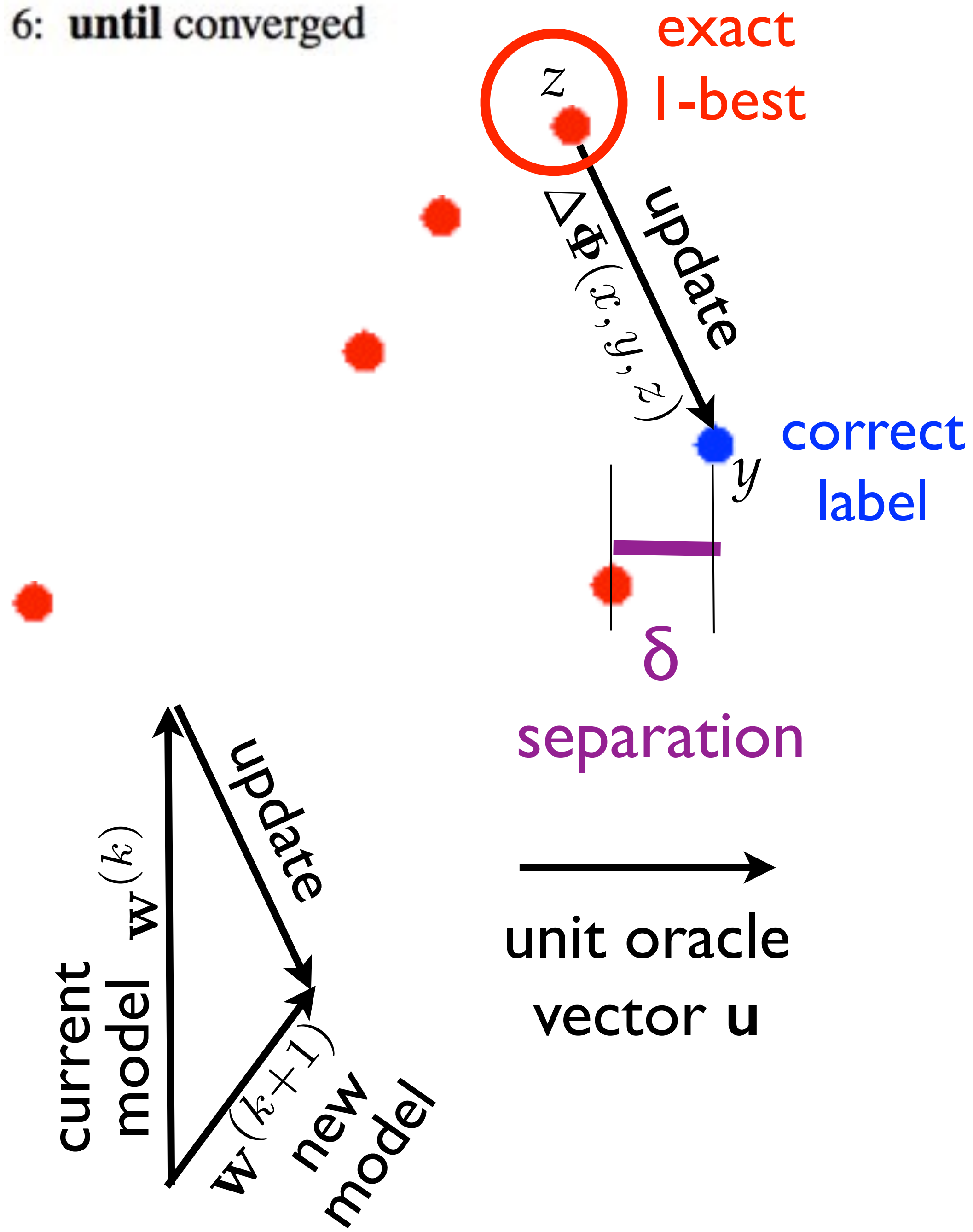


Novikoff => Freund & Schapire => Collins
1962 1999 2002

Geometry of Convergence Proof pt 1

repeat
  for each example (x, y) **in** D **do**
    $z \leftarrow \textsc{Exact}(x, \mathbf{w})$
    if $z \neq y$ **then**
      $\mathbf{w} \leftarrow \mathbf{w} + \Delta\mathbf{\Phi}(x, y, z)$
until converged





perceptron update:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \Delta\mathbf{\Phi}(x, y, z)$$

$$\mathbf{u} \cdot \mathbf{w}^{(k+1)} = \mathbf{u} \cdot \mathbf{w}^{(k)} + \boxed{\underbrace{\mathbf{u} \cdot \Delta\mathbf{\Phi}(x, y, z)}_{\geq \delta \quad \text{margin}}}$$

$$\mathbf{u} \cdot \mathbf{w}^{(k+1)} \geq k\delta \qquad \text{(by induction)}$$

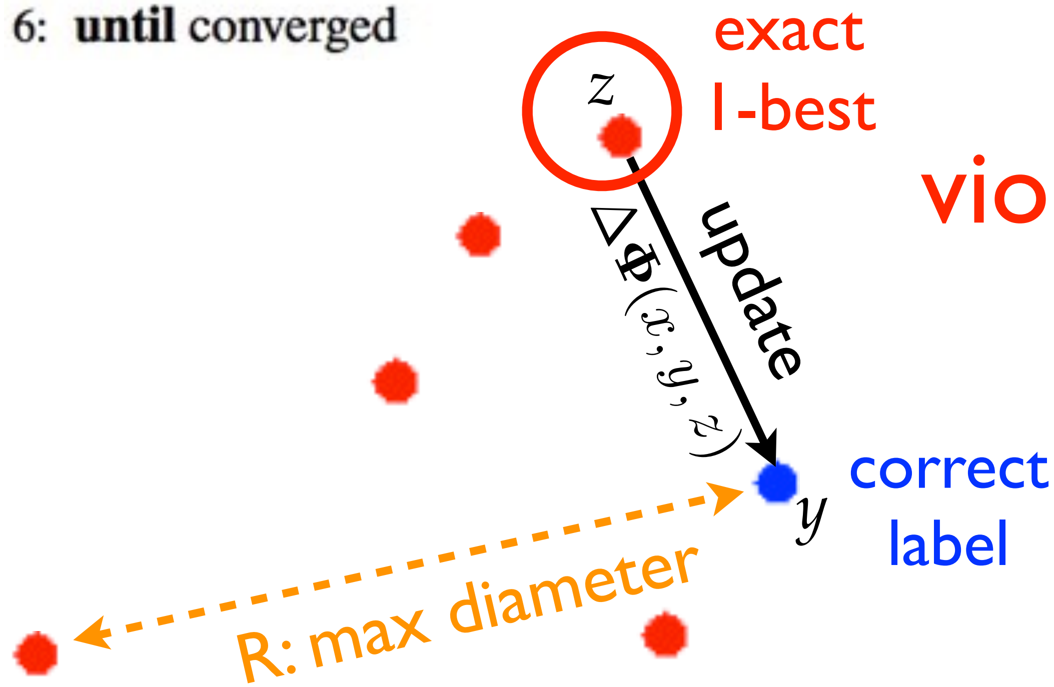
$$\|\mathbf{w}^{k+1}\| \geq k\delta \qquad \text{(part 1: lowerbound)}$$

Geometry of Convergence Proof pt 2

```
repeat
    for each example (x, y) in D do
        z ← EXACT(x, w)
        if z ≠ y then
            w ← w + ΔΦ(x, y, z)
until converged
```

summary: the proof uses 3 facts:

1. separation (margin)
2. diameter (always finite)
3. violation (guaranteed by exact search)



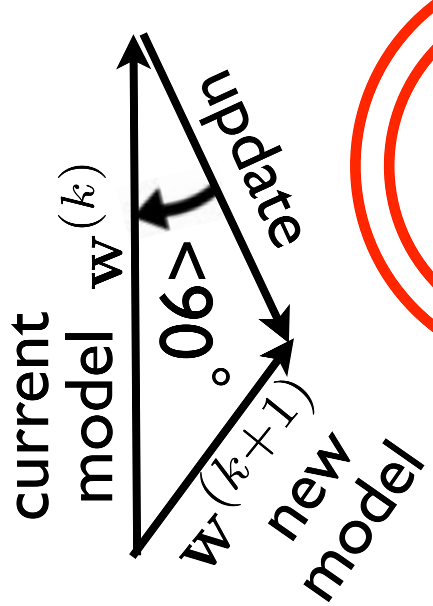
violation: incorrect label scored higher

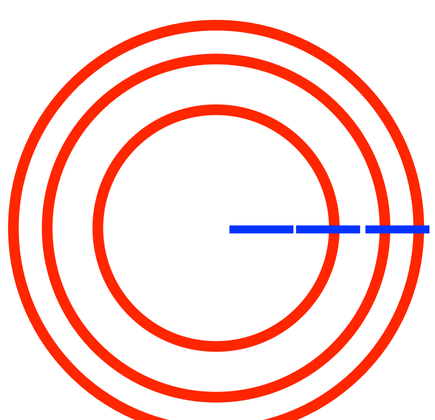
perceptron update:

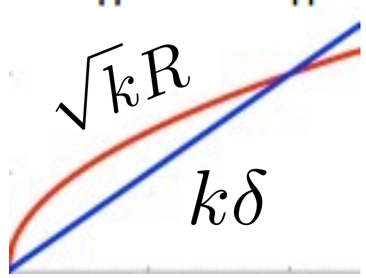
$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \Delta\mathbf{\Phi}(x,y,z)$$

$$\|\mathbf{w}^{(k+1)}\|^2 = \|\mathbf{w}^{(k)} + \Delta\mathbf{\Phi}(x,y,z)\|^2$$

$$= \|\mathbf{w}^{(k)}\|^2 + \boxed{\underbrace{\|\Delta\mathbf{\Phi}(x,y,z)\|^2}_{\leq R^2 \text{ diameter}}} + \underbrace{2\,\mathbf{w}^{(k)}\cdot\Delta\mathbf{\Phi}(x,y,z)}_{\leq 0 \text{ violation}}$$







by induction: $\|\mathbf{w}^{k+1}\|^2 \leq kR^2$ (part 2: upperbound)

$\|\mathbf{w}^{k+1}\| \leq \sqrt{k}R$

combine with: $\|\mathbf{w}^{k+1}\| \geq k\delta$ (part 1: lowerbound)

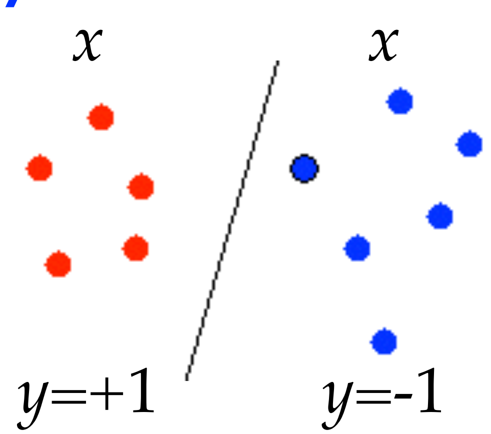
bound on # of updates: $\boxed{k \leq R^2/\delta^2}$

Outline

- Overview of Structured Learning
  - Challenges in Scalability
- Structured Perceptron
  - convergence proof
- Structured Perceptron with Inexact Search
- Latent-Variable Perceptron

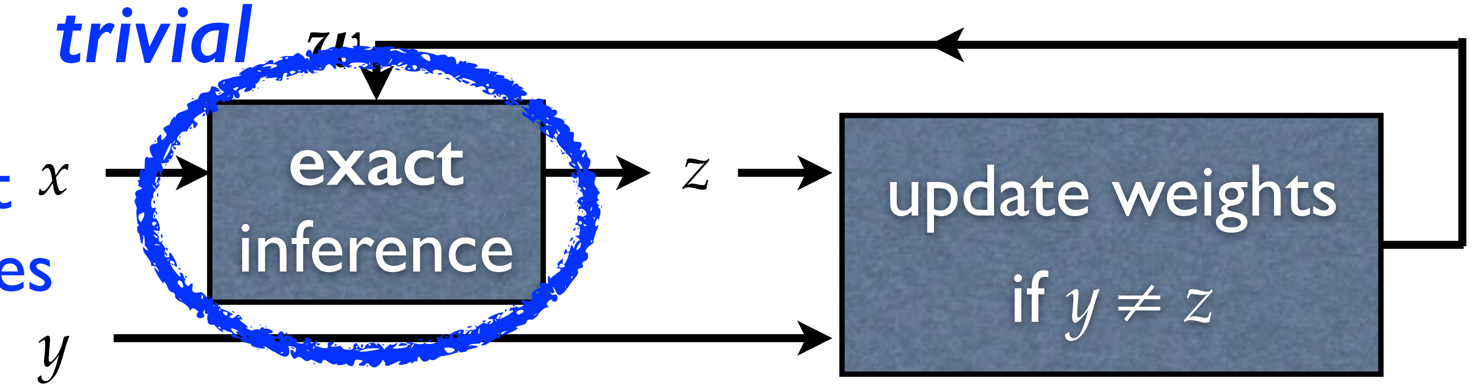
Scalability Challenge 1: Inference

binary classification



trivial

constant # of classes

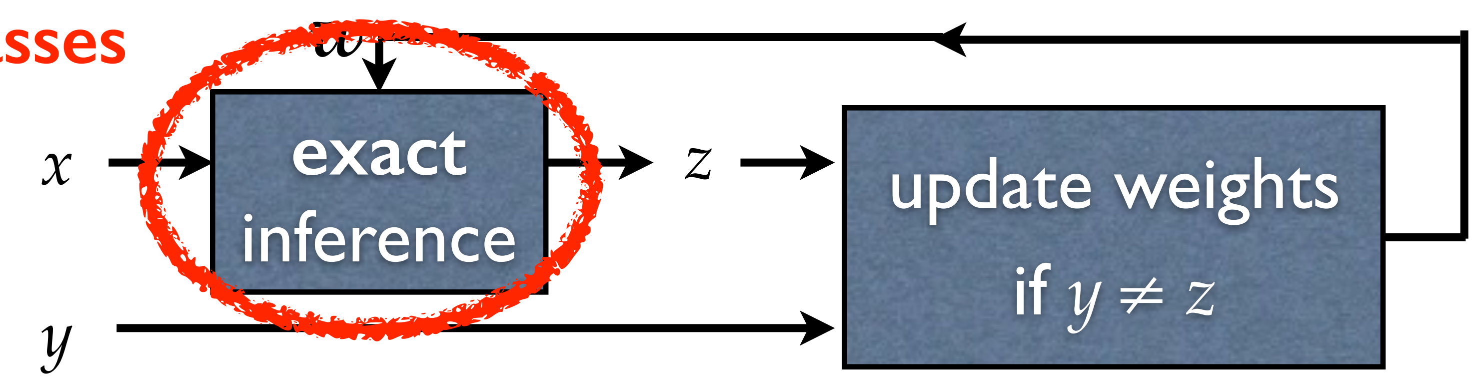


structured classification

exponential # of classes

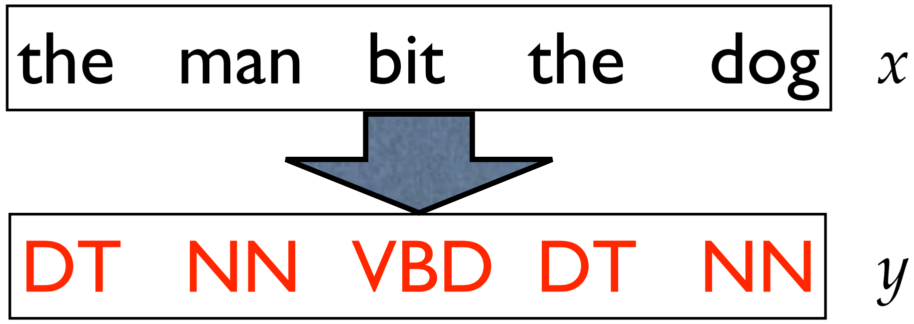
hard

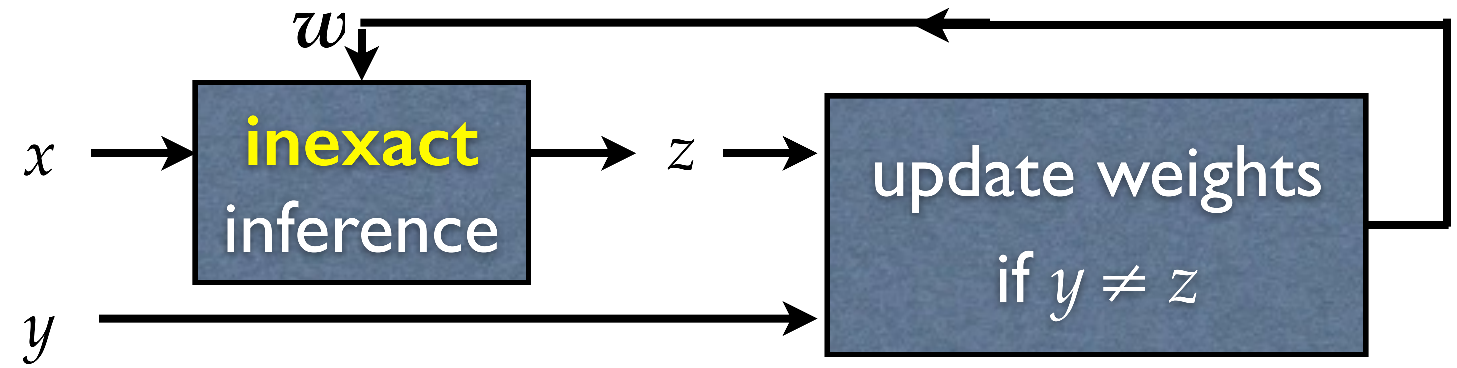
the	man	bit	the	dog	x
DT	NN	VBD	DT	NN	y

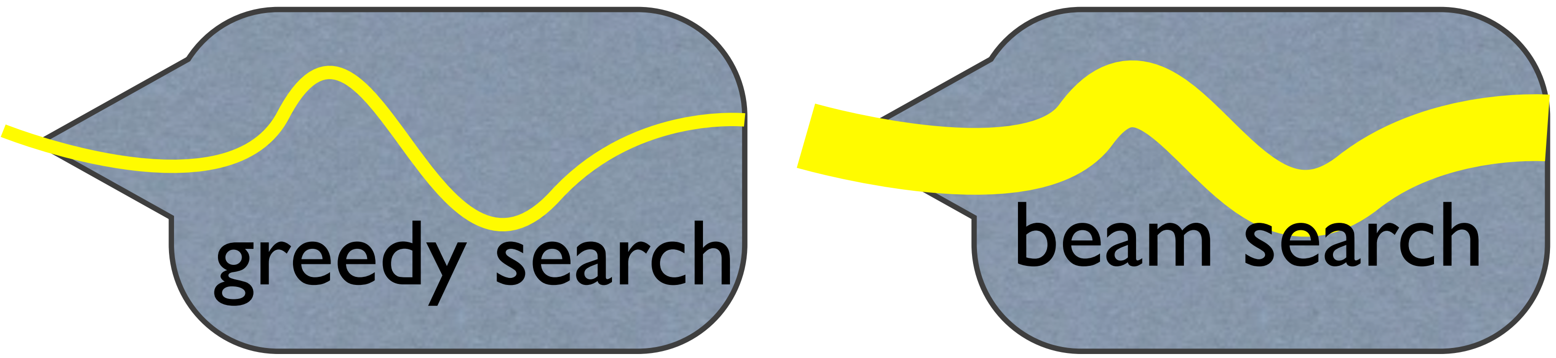


- challenge: search efficiency (exponentially many classes)
  - often use dynamic programming (DP)
  - but DP is still too slow for repeated use, e.g. parsing $O(n^3)$
  - Q: can we sacrifice search exactness for faster learning?

Perceptron w/ Inexact Inference



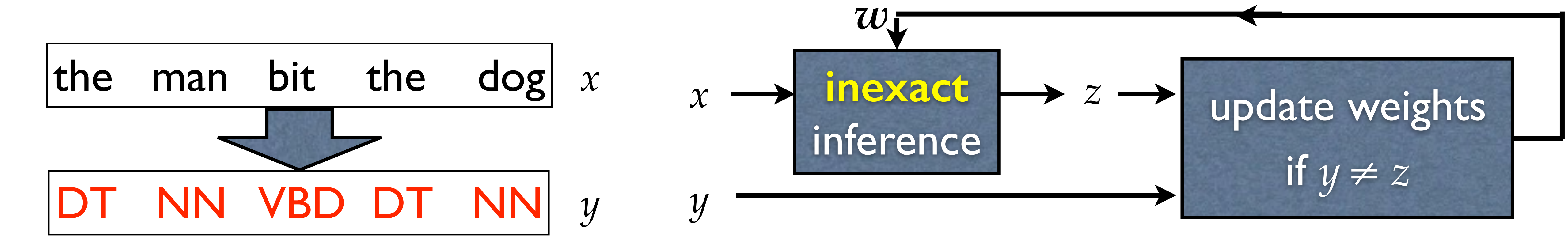


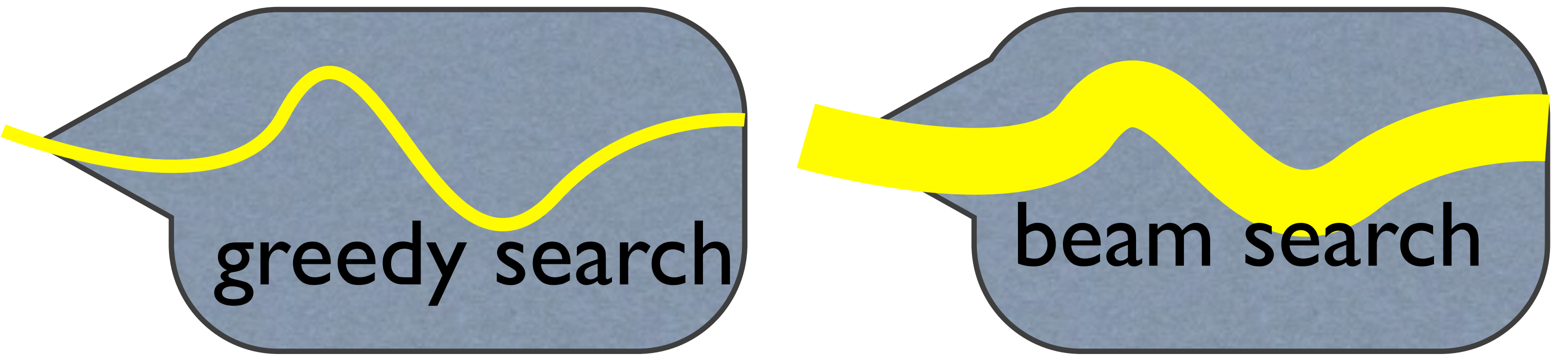


Q: does perceptron still work???

- routine use of inexact inference in NLP (e.g. beam search)
- how does structured perceptron work with inexact search?
  - so far most structured learning theory assume exact search
  - would search errors break these learning properties?
  - if so how to modify learning to accommodate inexact search?

Bad News and Good News

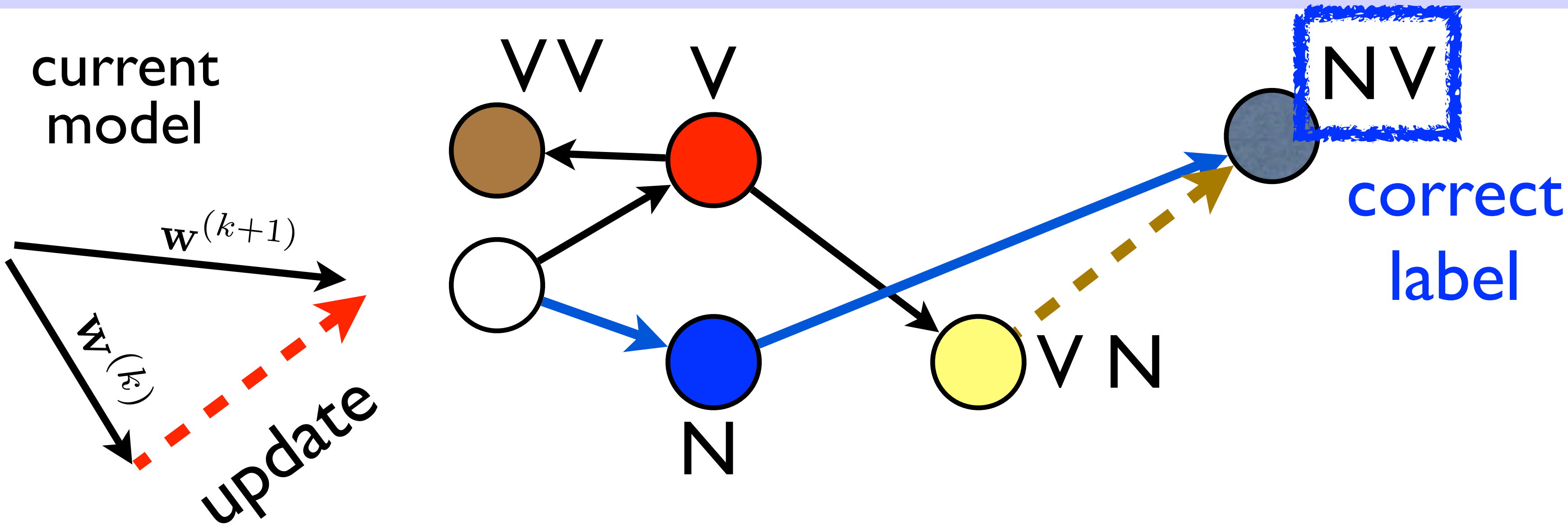




A: it no longer works as is, but we can make it work by some magic.

- bad news: no more guarantee of convergence
  - in practice perceptron degrades a lot due to search errors
- good news: new update methods guarantee convergence
  - new perceptron variants that "live with" search errors
  - in practice they work really well w/ inexact search

Convergence with Exact Search



training example

time flies

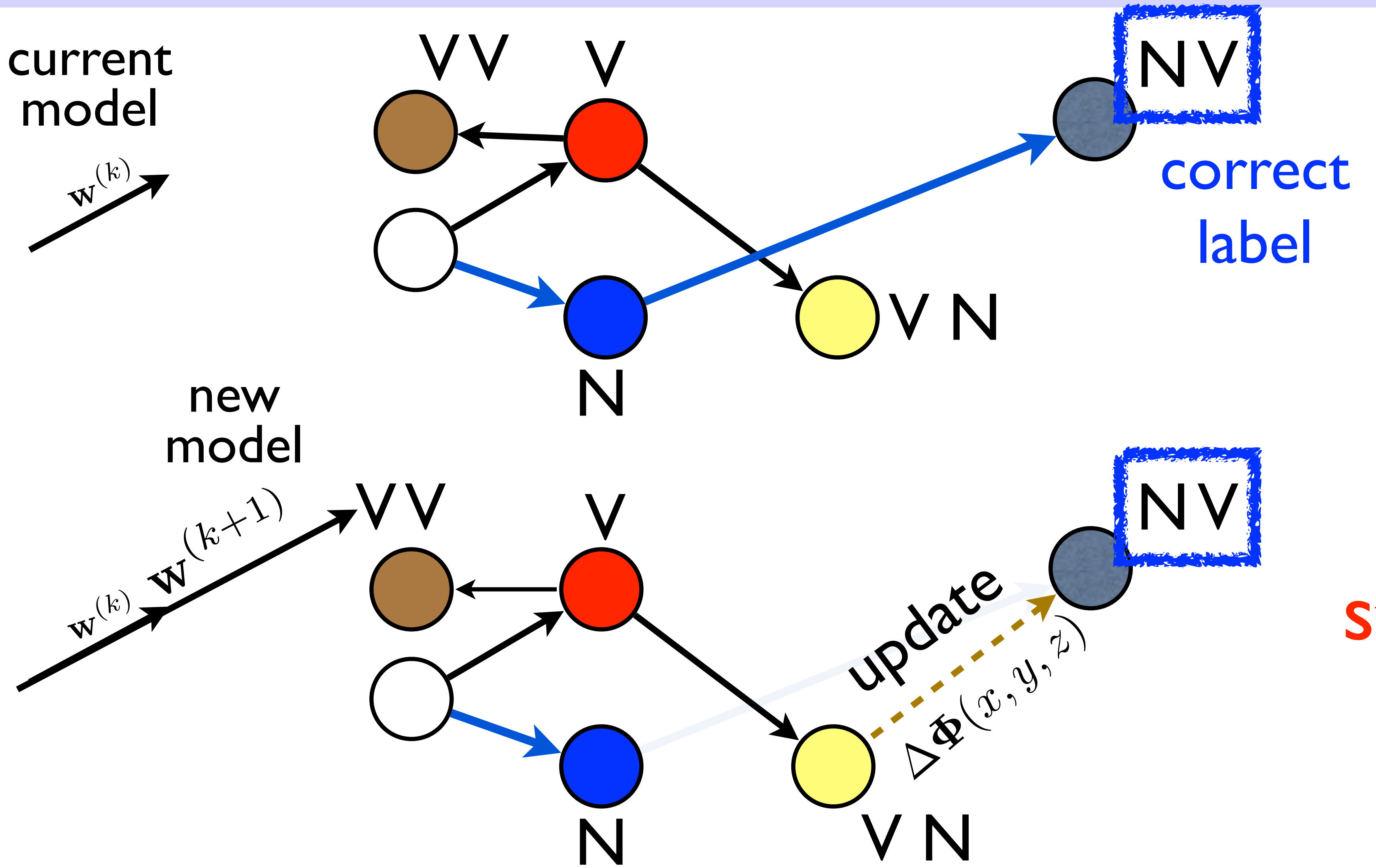
N V

output space

{N,V} x {N,V}

structured perceptron
converges with
exact search

No Convergence w/ Greedy Search



training example

time	flies
N	V

output space
{N,V} x {N,V}

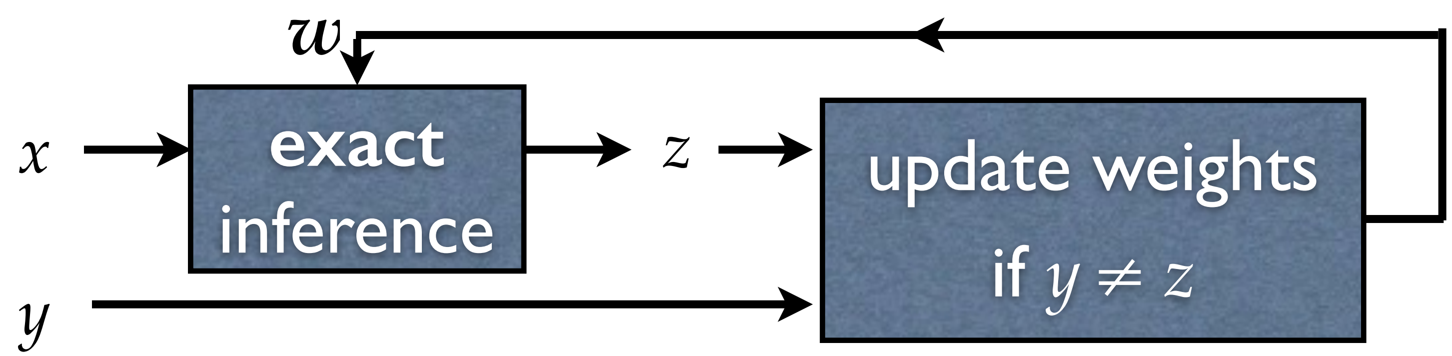
structured perceptron does not converge with greedy search

Which part of the convergence proof no longer holds?

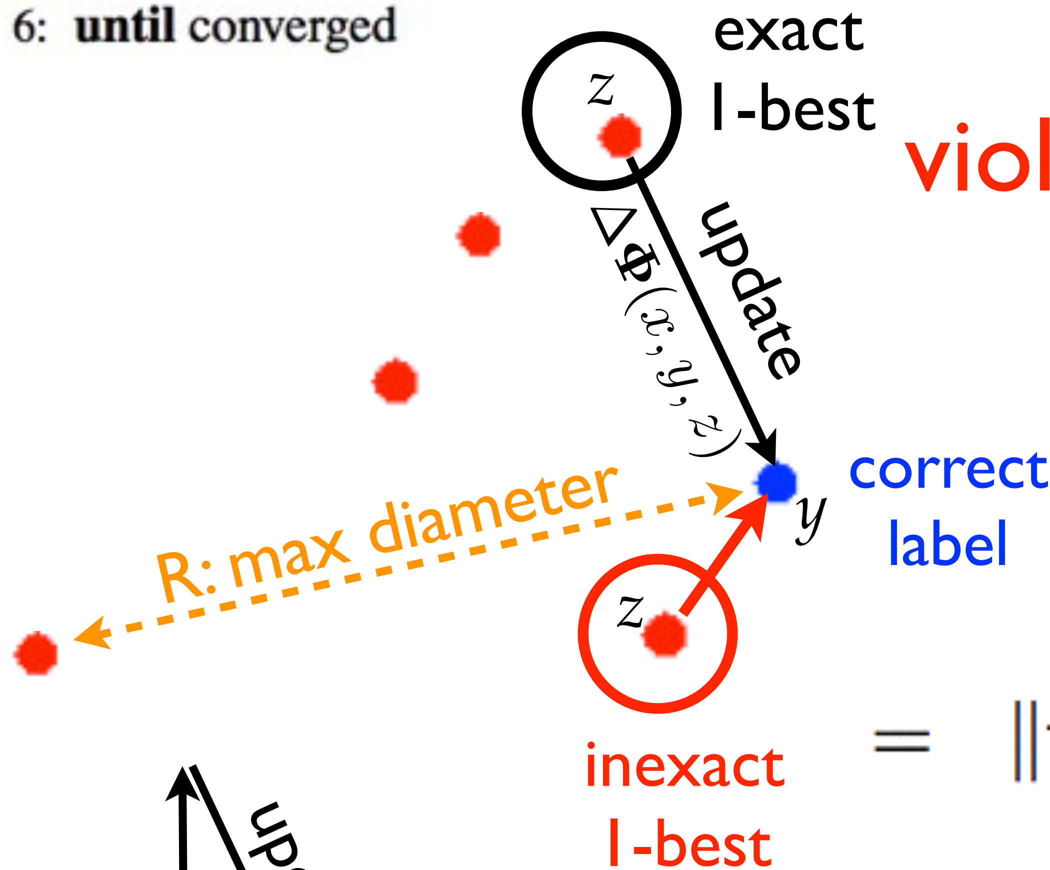
the proof only uses 3 facts:
1. separation (margin)
2. diameter (always finite)
3. violation (guaranteed by exact search)

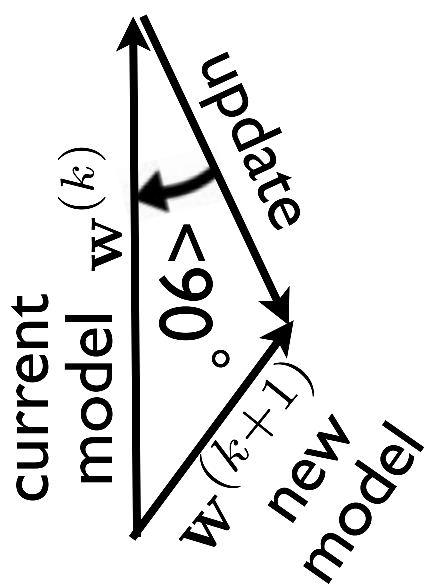
Geometry of Convergence Proof pt 2

```
repeat
    for each example (x, y) in D do
        z ← EXACT(x, w)
        if z ≠ y then
            w ← w + ΔΦ(x, y, z)
until converged
```



violation: incorrect label scored higher





perceptron update:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \Delta\Phi(x, y, z)$$

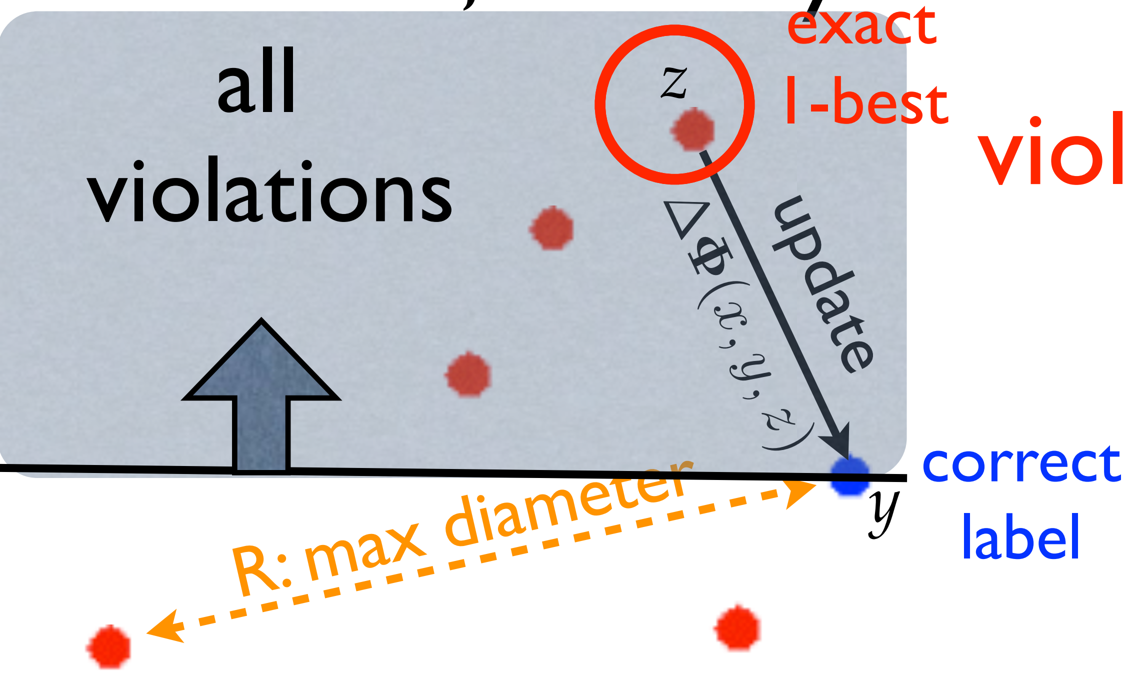
$$\|\mathbf{w}^{(k+1)}\|^2 = \|\mathbf{w}^{(k)} + \Delta\Phi(x, y, z)\|^2$$

$$= \|\mathbf{w}^{(k)}\|^2 + \underbrace{\|\Delta\Phi(x, y, z)\|^2}_{\leq R^2 \ \text{diameter}} + 2\,\underbrace{\mathbf{w}^{(k)} \cdot \Delta\Phi(x, y, z)}_{\leq 0 \ \text{violation}}$$

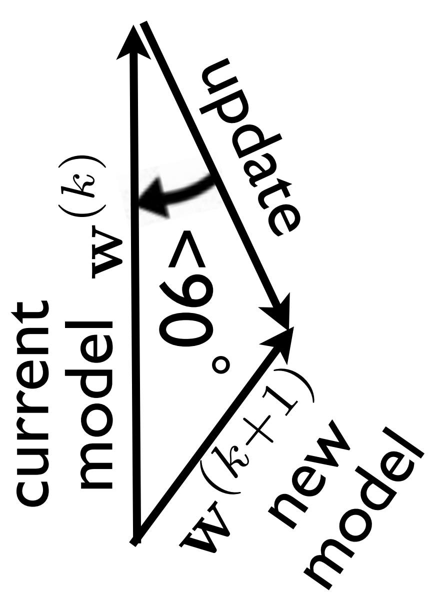
inexact search doesn't guarantee violation!

Observation: Violation is all we need!

- exact search is **not** really required by the proof
  - rather, it is only used to ensure violation!



violation: incorrect label scored higher

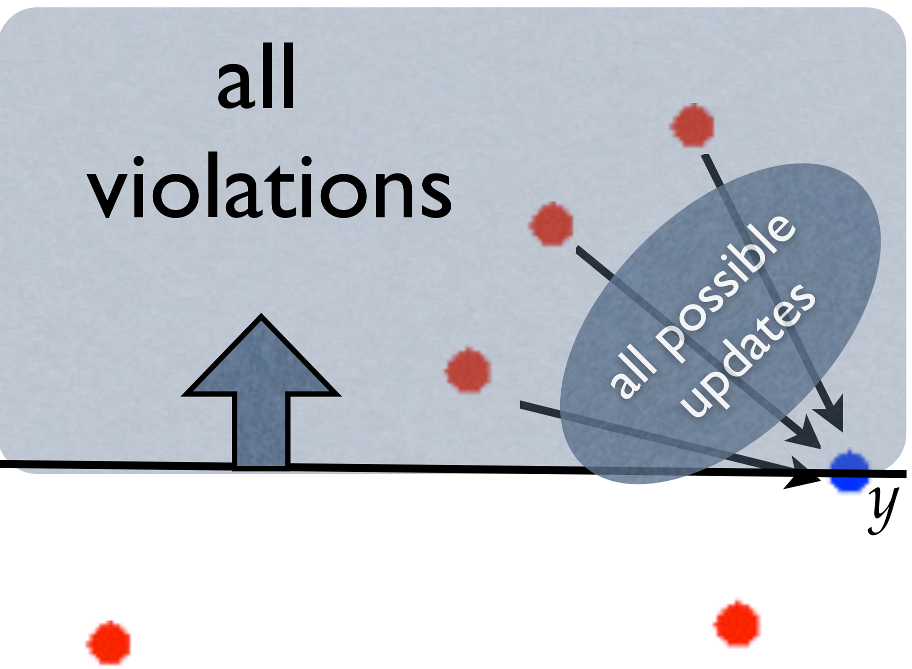


the proof only uses 3 facts:

1. separation (margin)
2. diameter (always finite)
3. violation (but no need for exact)

Violation-Fixing Perceptron

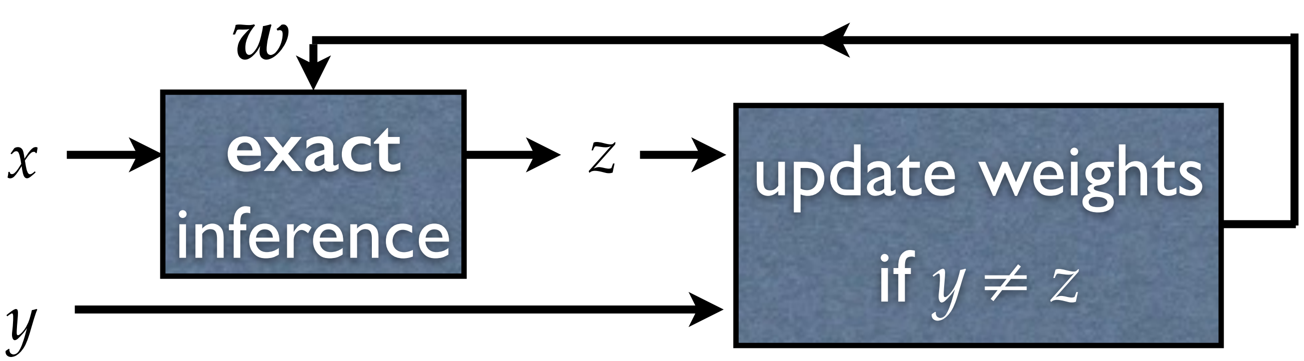
- if we guarantee violation, we don't care about exactness!
  - violation is good b/c we can at least fix a mistake



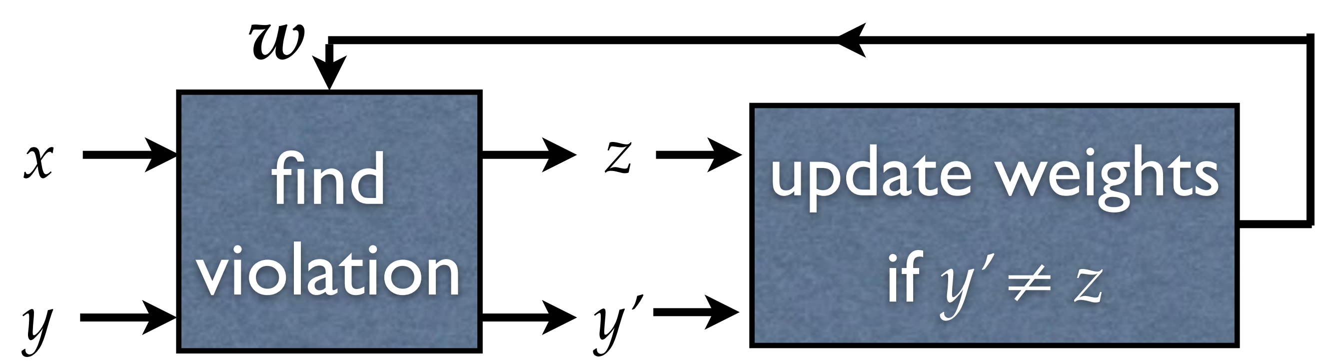
same mistake bound as before!

repeat
    for each example (x, y) **in** D **do**
        $(x, y', z) = \textsc{FindViolation}(x, y, \mathbf{w})$
        if $z \neq y$ **then** $\triangleright$ (x, y', z) is a viol
            $\mathbf{w} \leftarrow \mathbf{w} + \Delta\mathbf{\Phi}(x, y', z)$
until converged

standard perceptron

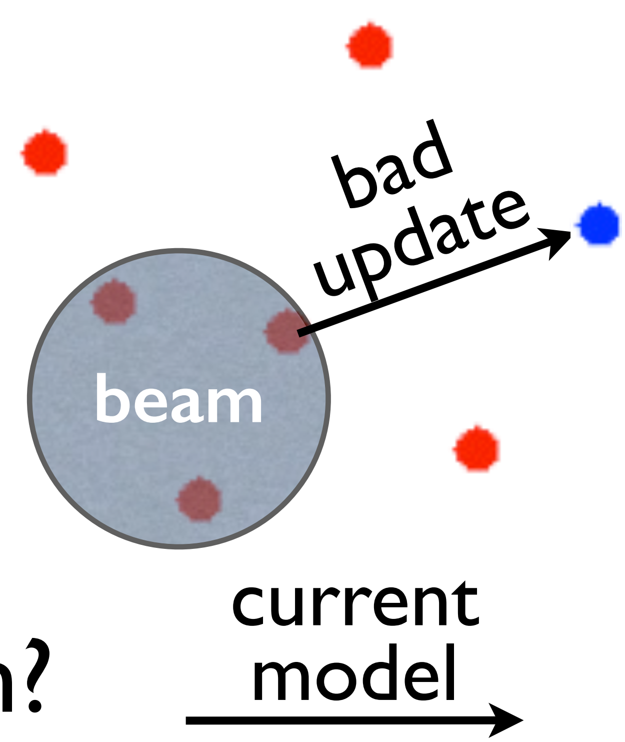


violation-fixing perceptron

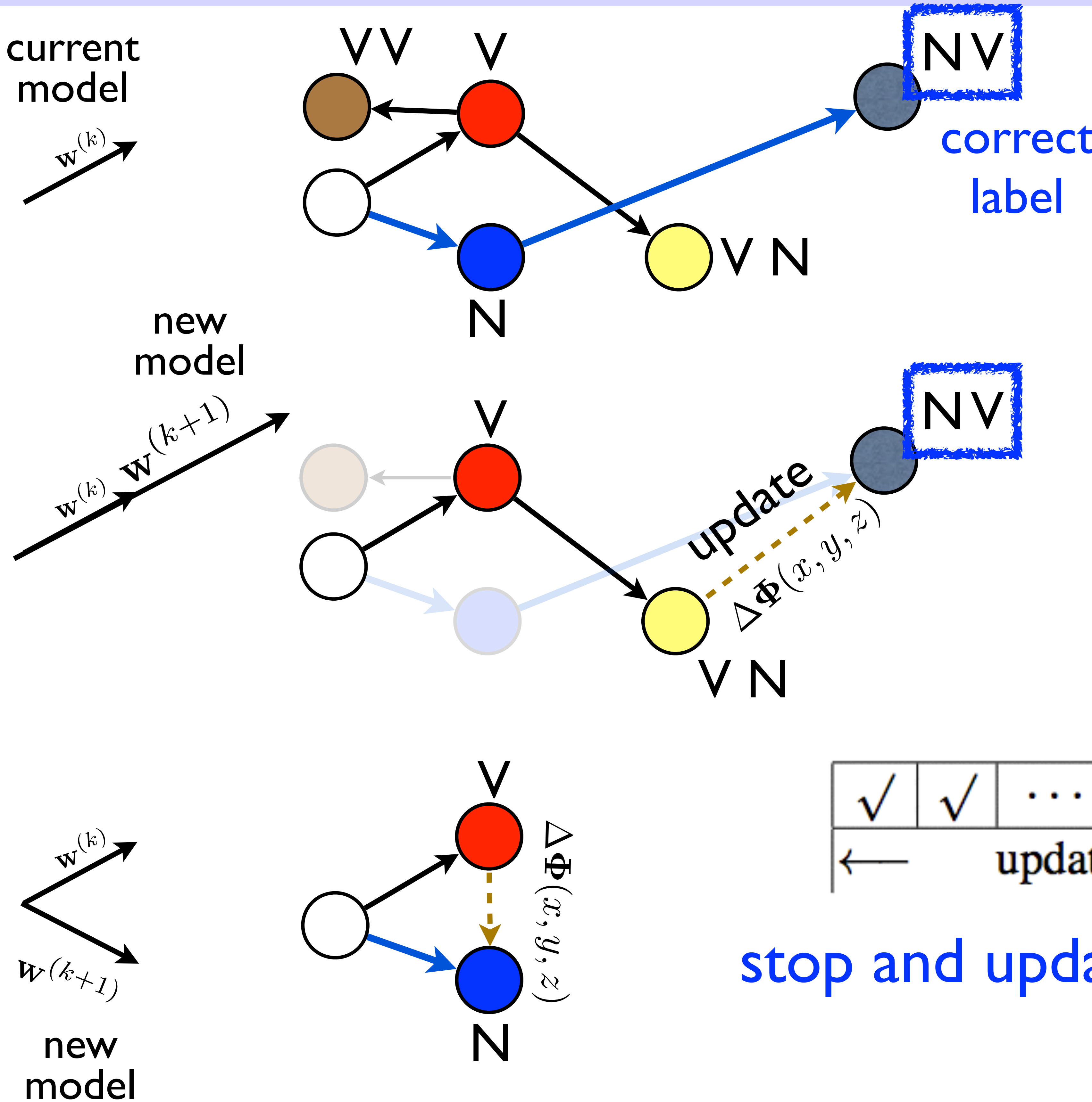


What if can't guarantee violation

- this is why perceptron doesn't work well w/ inexact search
  - because not every update is guaranteed to be a violation
  - thus the proof breaks; no convergence guarantee
- example: beam or greedy search
  - the model might prefer the correct label (if exact search)
  - but the search prunes it away
  - such a non-violation update is "bad" because it doesn't fix any mistake
  - the new model still misguides the search
- Q: how can we always guarantee violation?



Solution 1: Early update (Collins/Roark 2004)



training example

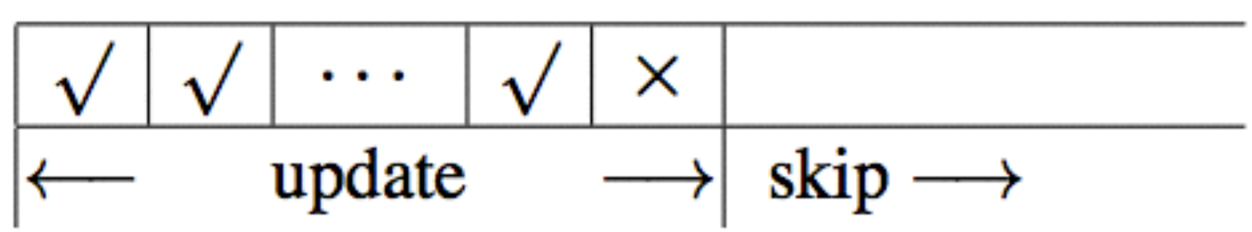
time flies

N V

output space

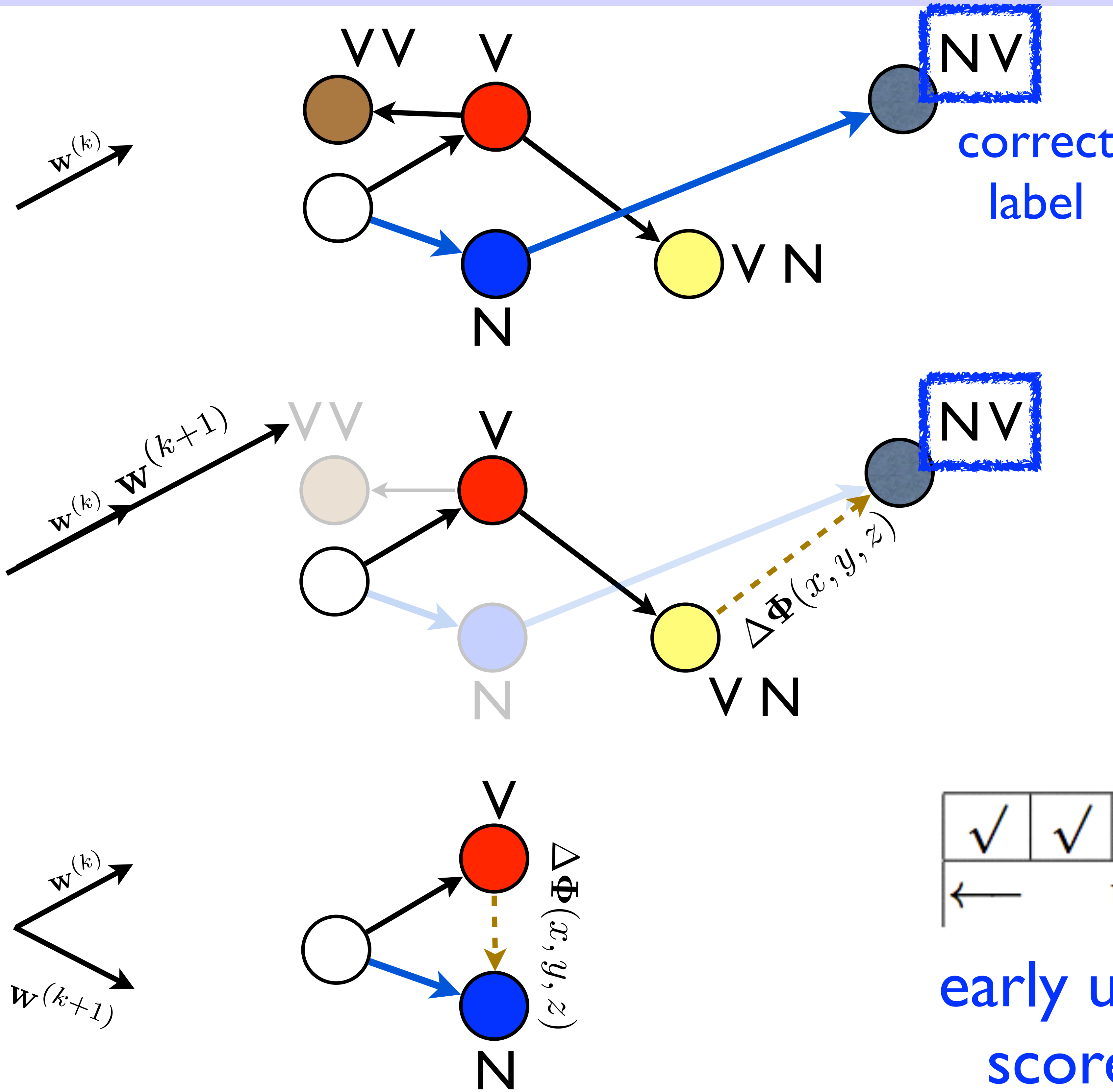
{N,V} x {N,V}

standard perceptron
does not converge
with greedy search



stop and update at the first mistake

Early Update: Guarantees Violation



training example

time flies

N V

output space

{N,V} x {N,V}

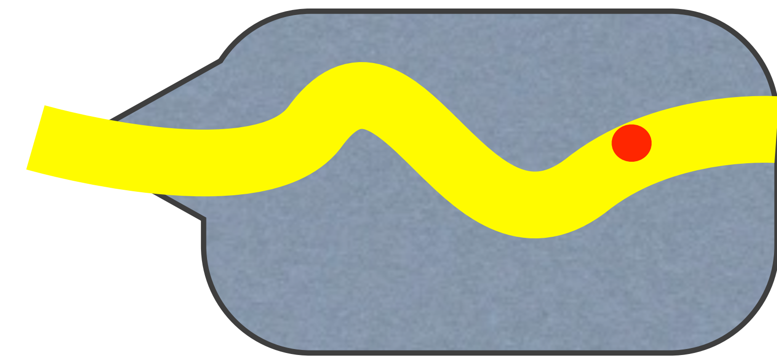
standard update doesn't converge b/c it doesn't guarantee violation

√	√	···	√	×	
←	update			→	skip →

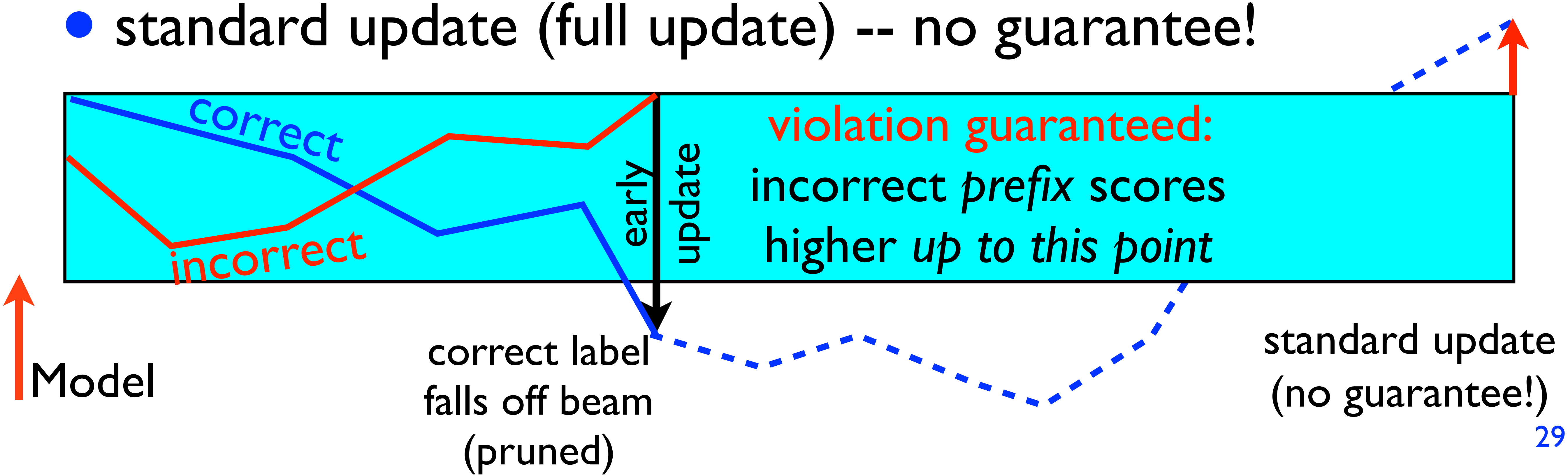
early update: incorrect prefix scores higher: a violation!

Early Update: from Greedy to Beam

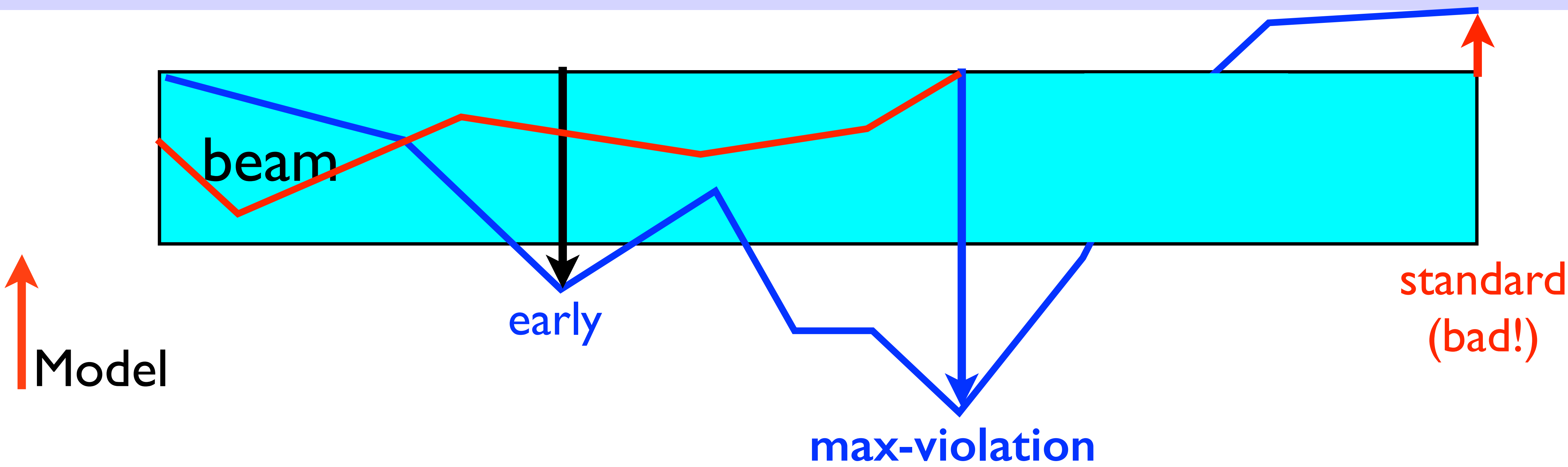
- beam search is a generalization of greedy (where b=1)
  - at each stage we keep top b hypothesis
  - widely used: tagging, parsing, translation...



- early update -- when correct label first falls off the beam
  - up to this point the incorrect prefix should score higher
- standard update (full update) -- no guarantee!



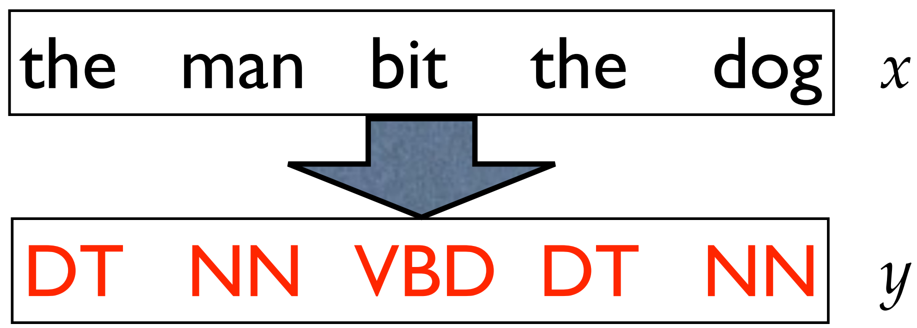
Solution 2: Max-Violation (Huang et al 2012)



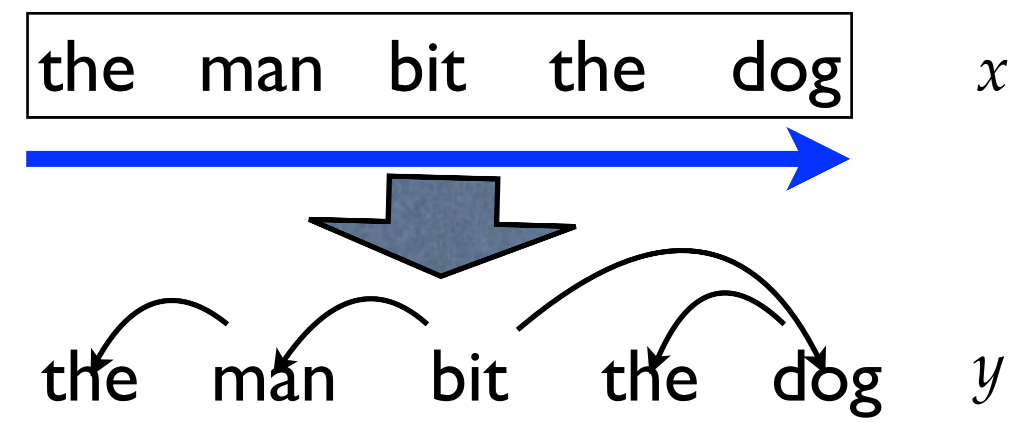
- we now established a theory for early update (Collins/Roark)
- but it learns too slowly due to partial updates
- max-violation: use the prefix where violation is maximum
  - "worst-mistake" in the search space
- all these update methods are violation-fixing perceptrons

Four Experiments

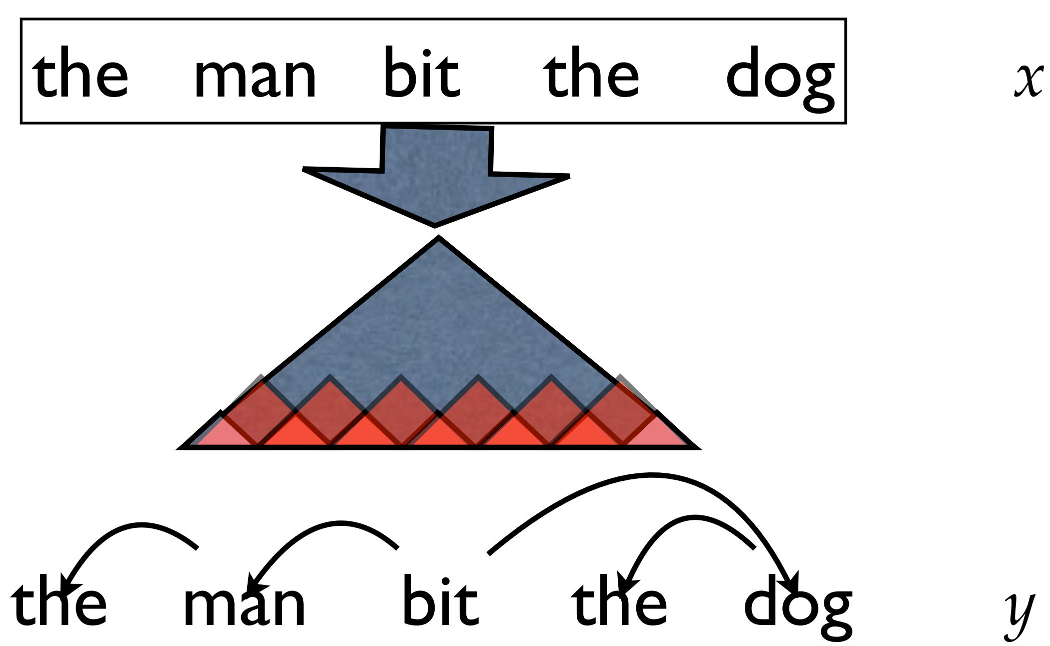
part-of-speech tagging



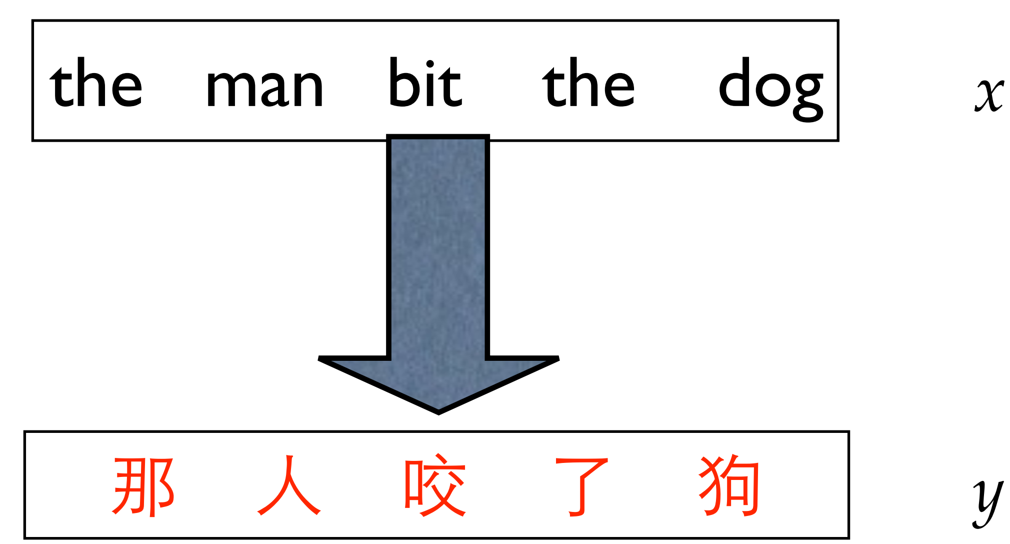
incremental parsing



bottom-up parsing w/ cube pruning

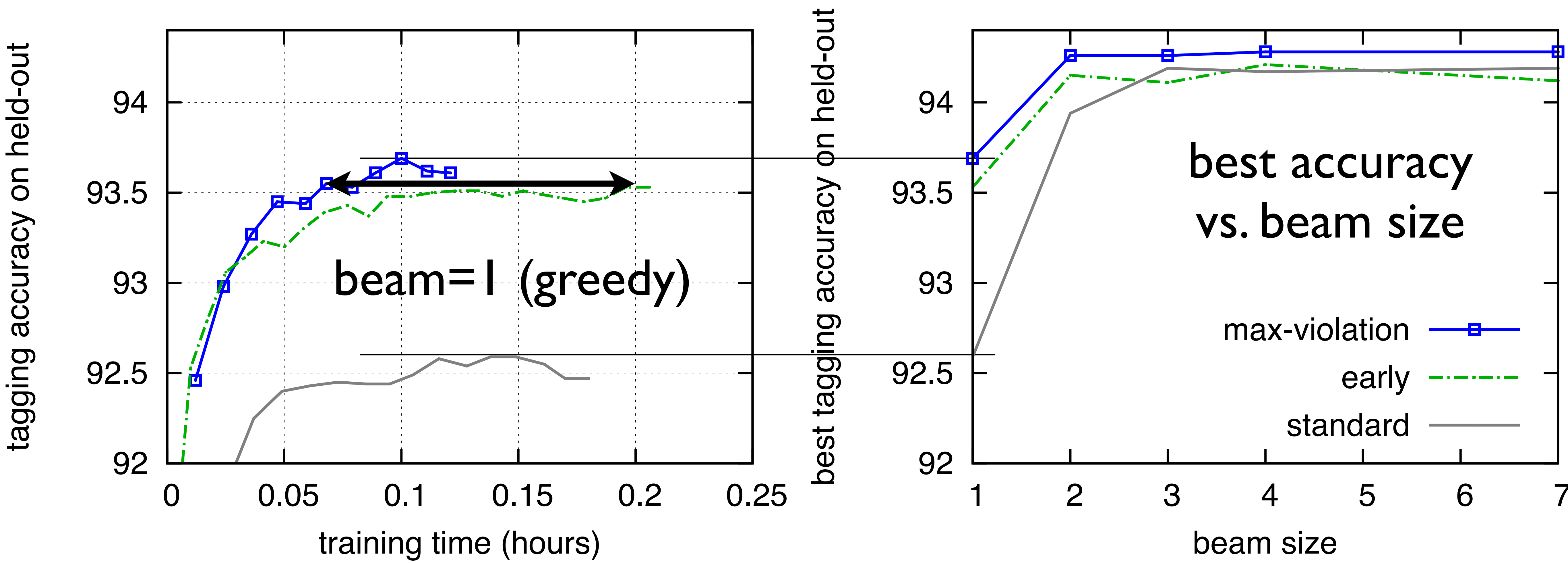


machine translation



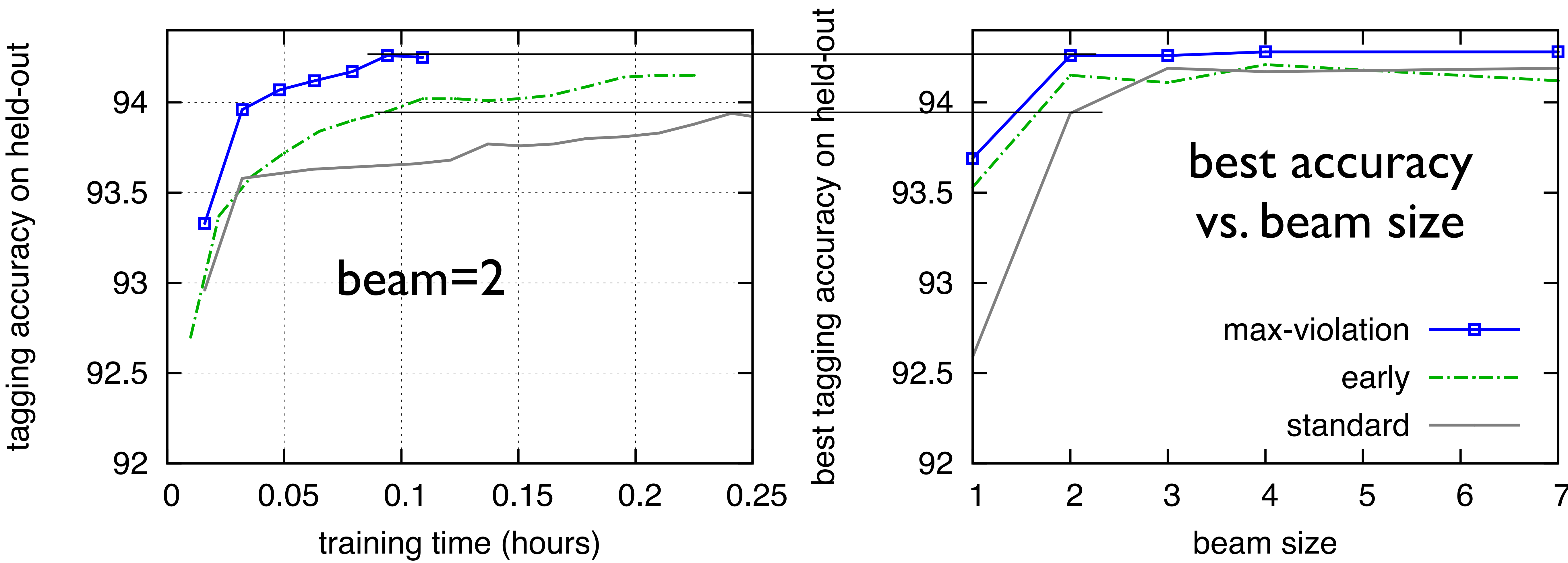
Max-Violation > Early >> Standard

- exp 1 on part-of-speech tagging w/ beam search (on CTB5)
- early and max-violation >> standard update at smallest beams
  - this advantage shrinks as beam size increases
  - max-violation converges faster than early (and slightly better)



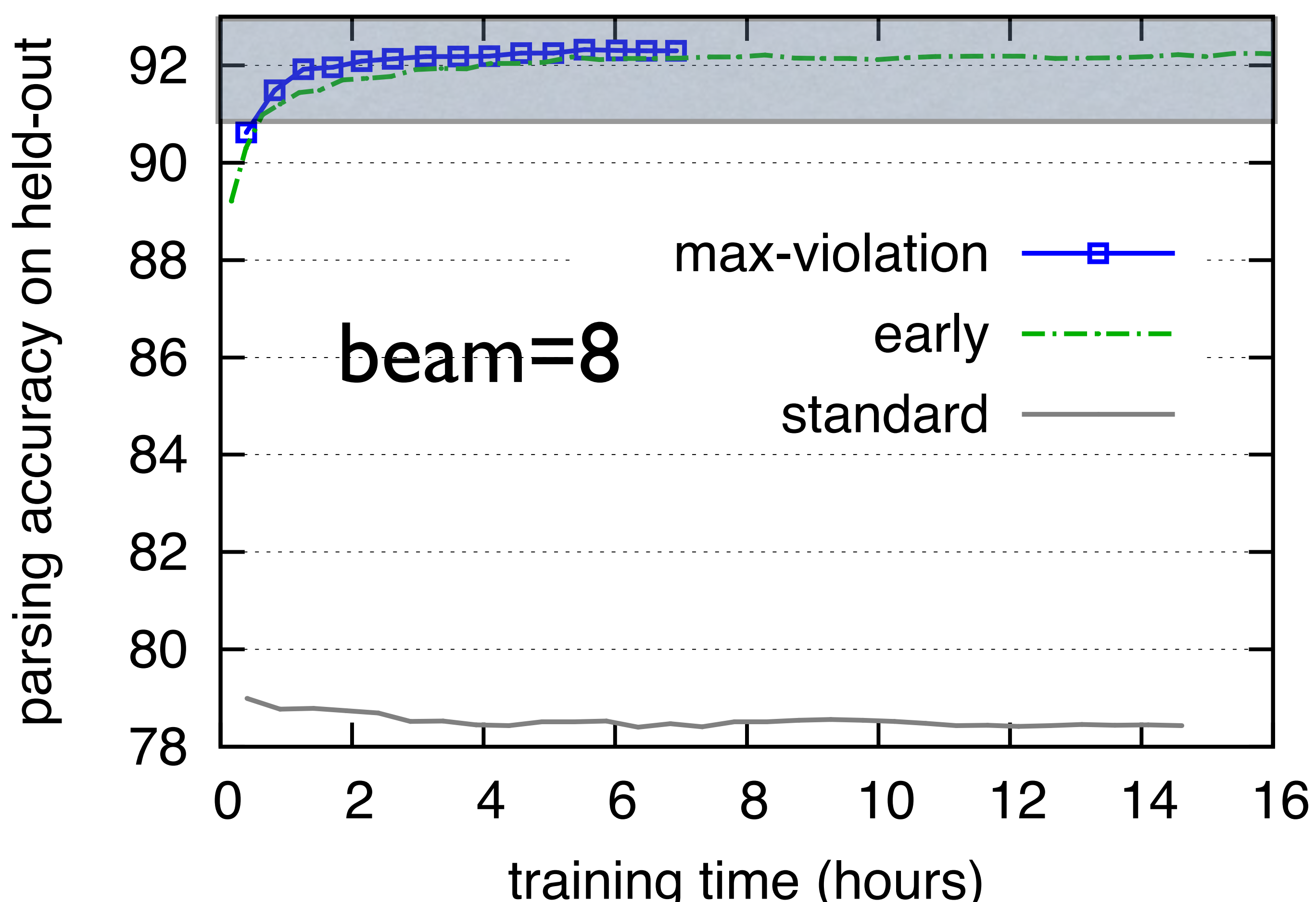
Max-Violation > Early >> Standard

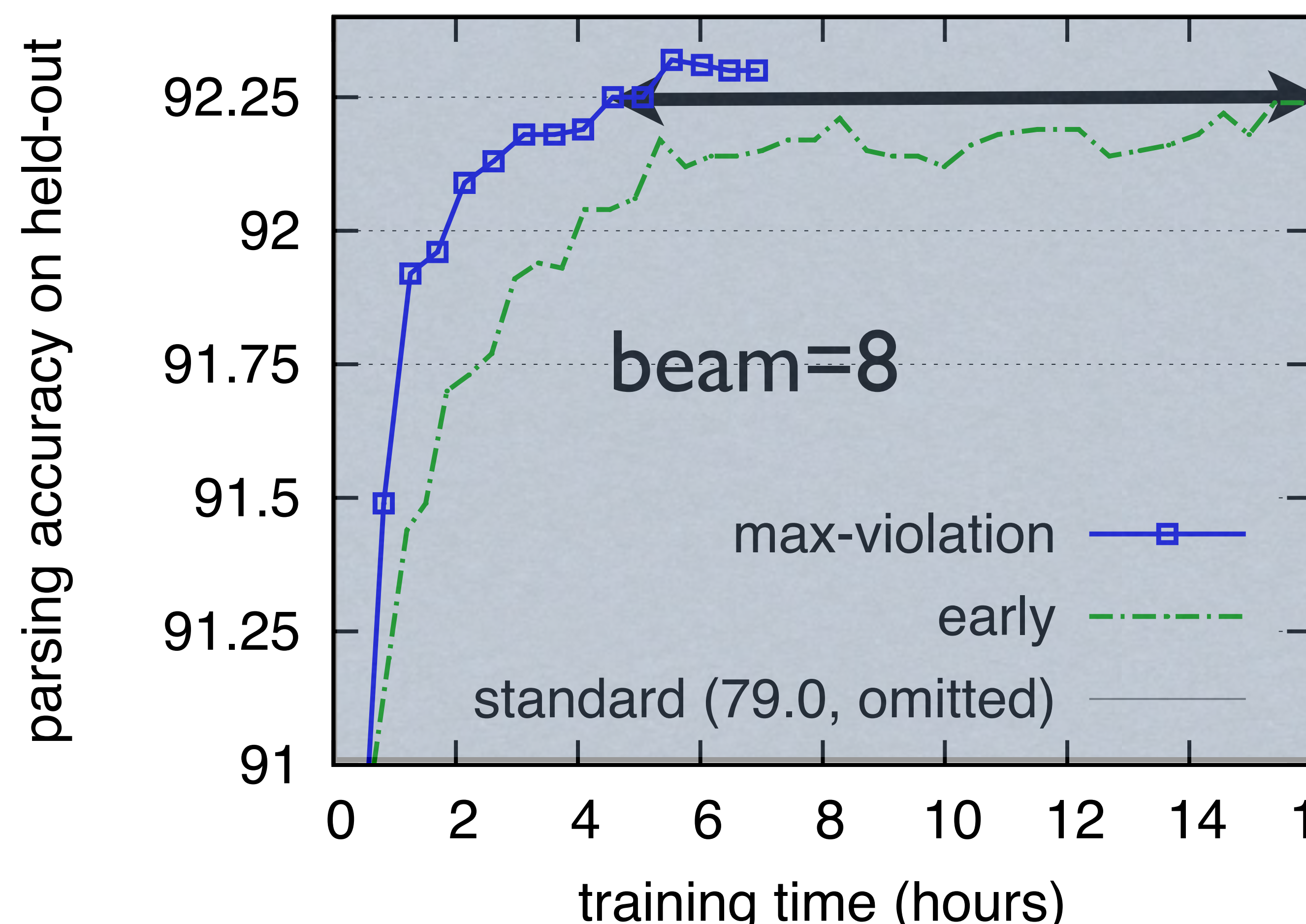
- exp 1 on part-of-speech tagging w/ beam search (on CTB5)
- early and max-violation >> standard update at smallest beams
  - this advantage shrinks as beam size increases
  - max-violation converges faster than early (and slightly better)



Max-Violation > Early >> Standard

- exp 2 on incremental dependency parser (Huang & Sagae 10)
- standard update is horrible due to search errors
- early update: 38 iterations, 15.4 hours (92.24)
- max-violation: 10 iterations, 4.6 hours (92.25)

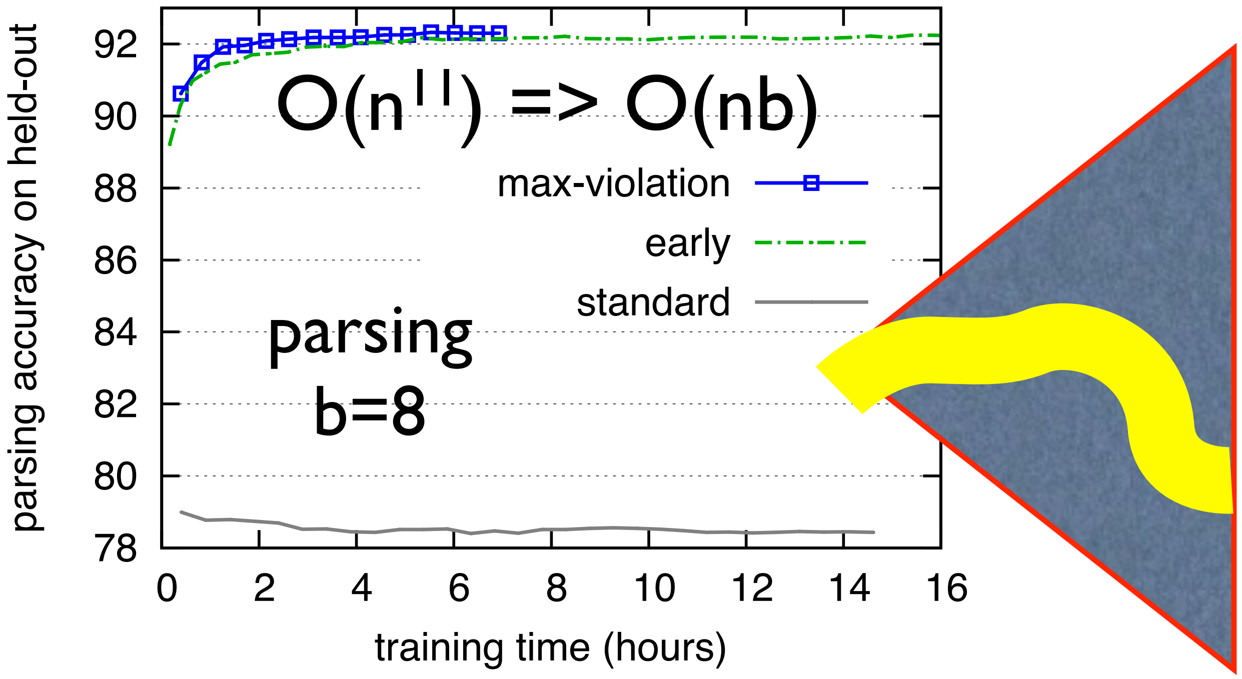


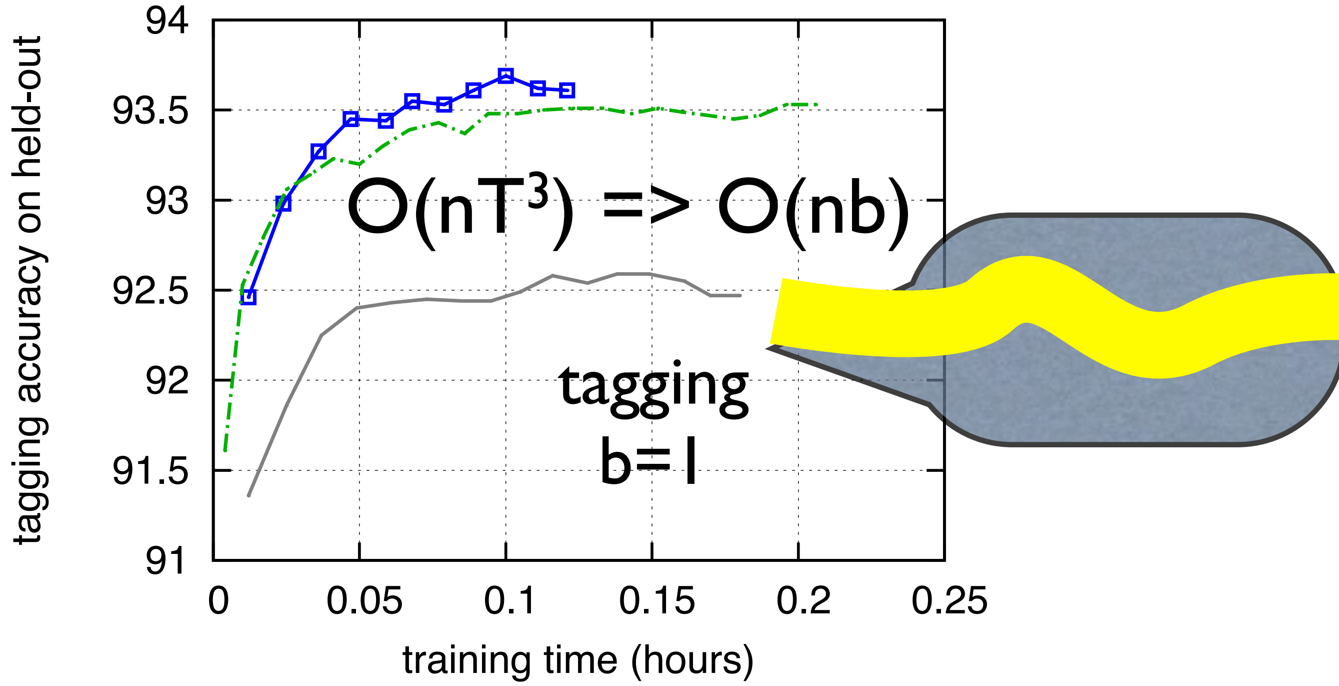


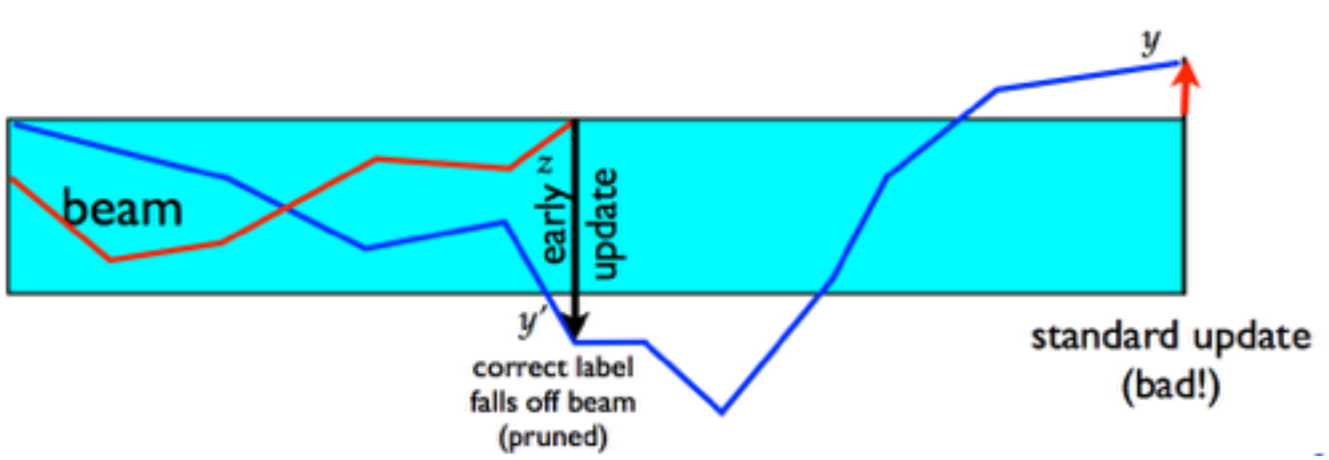
max-violation is 3.3x faster than early update

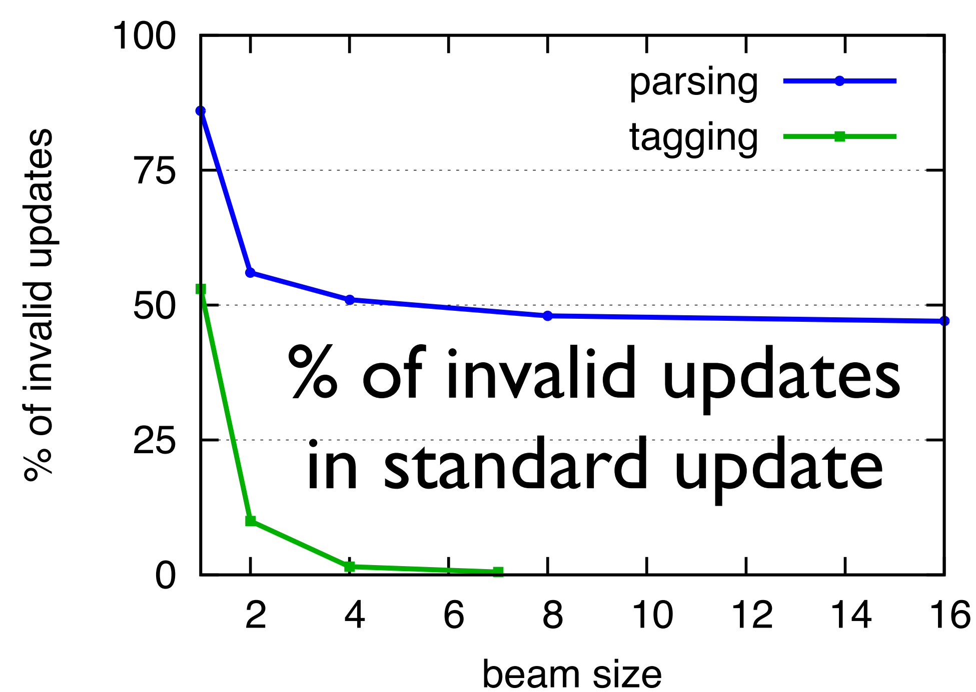
Why standard update so bad for parsing

- standard update works horribly with severe search error
    - due to large number of *invalid* updates (non-violation)



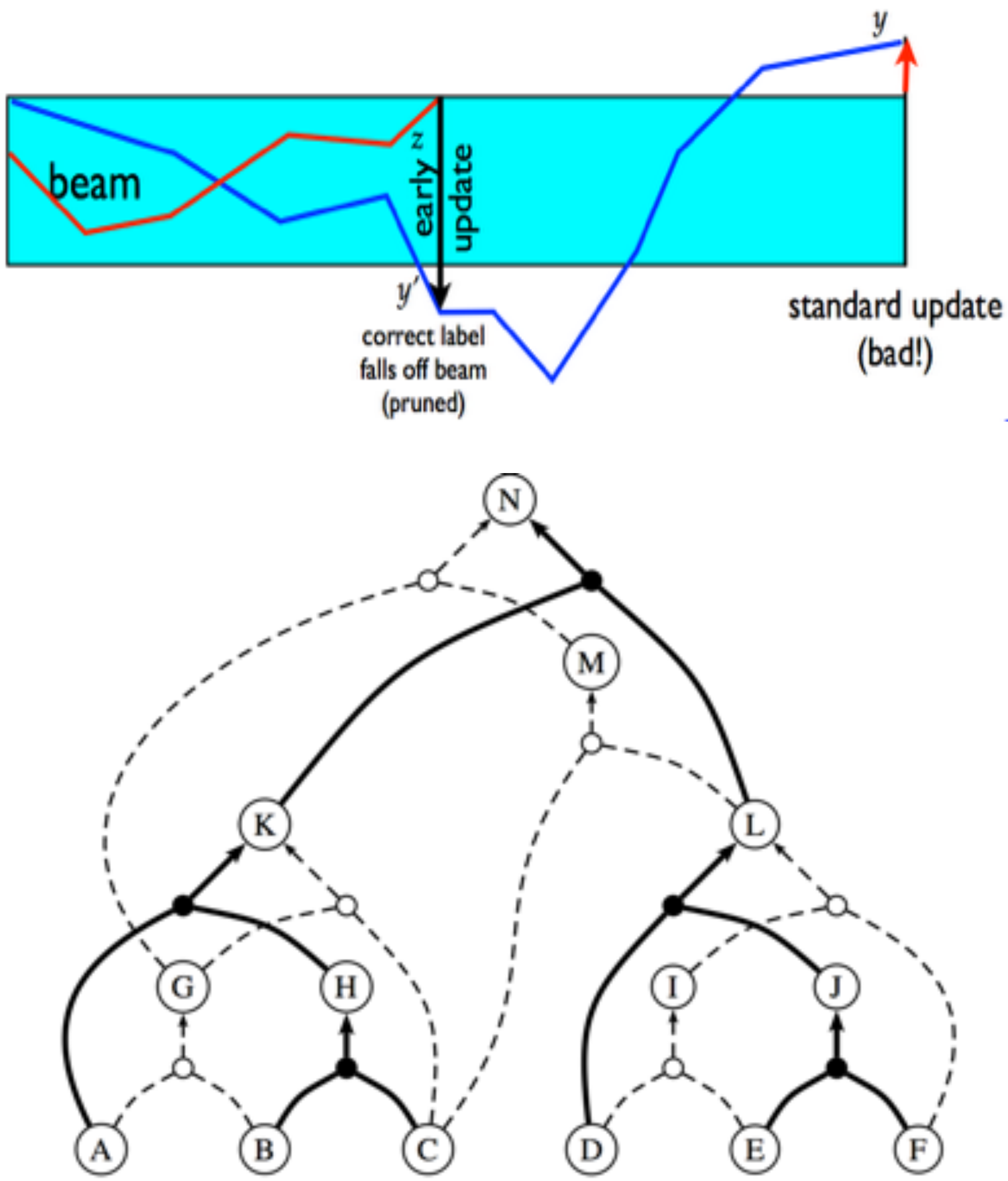


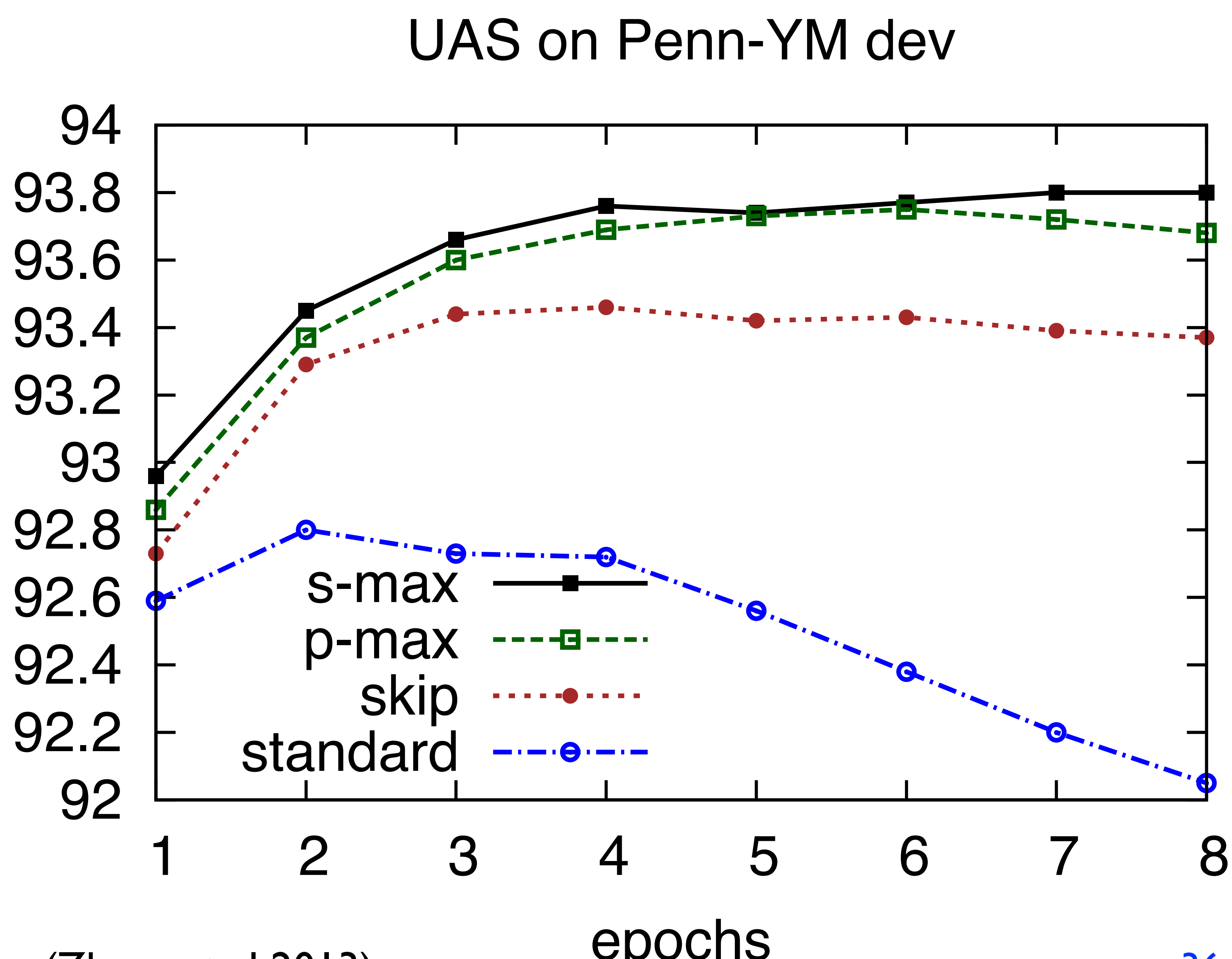




Exp 3: Bottom-up Parsing

- CKY parsing with cube pruning for higher-order features
- we extended our framework from graphs to hypergraphs

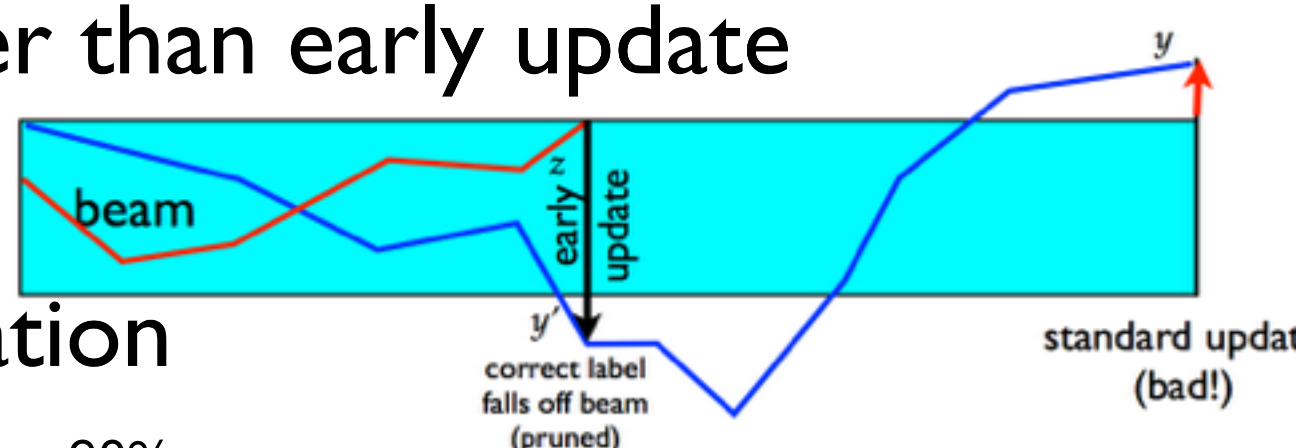


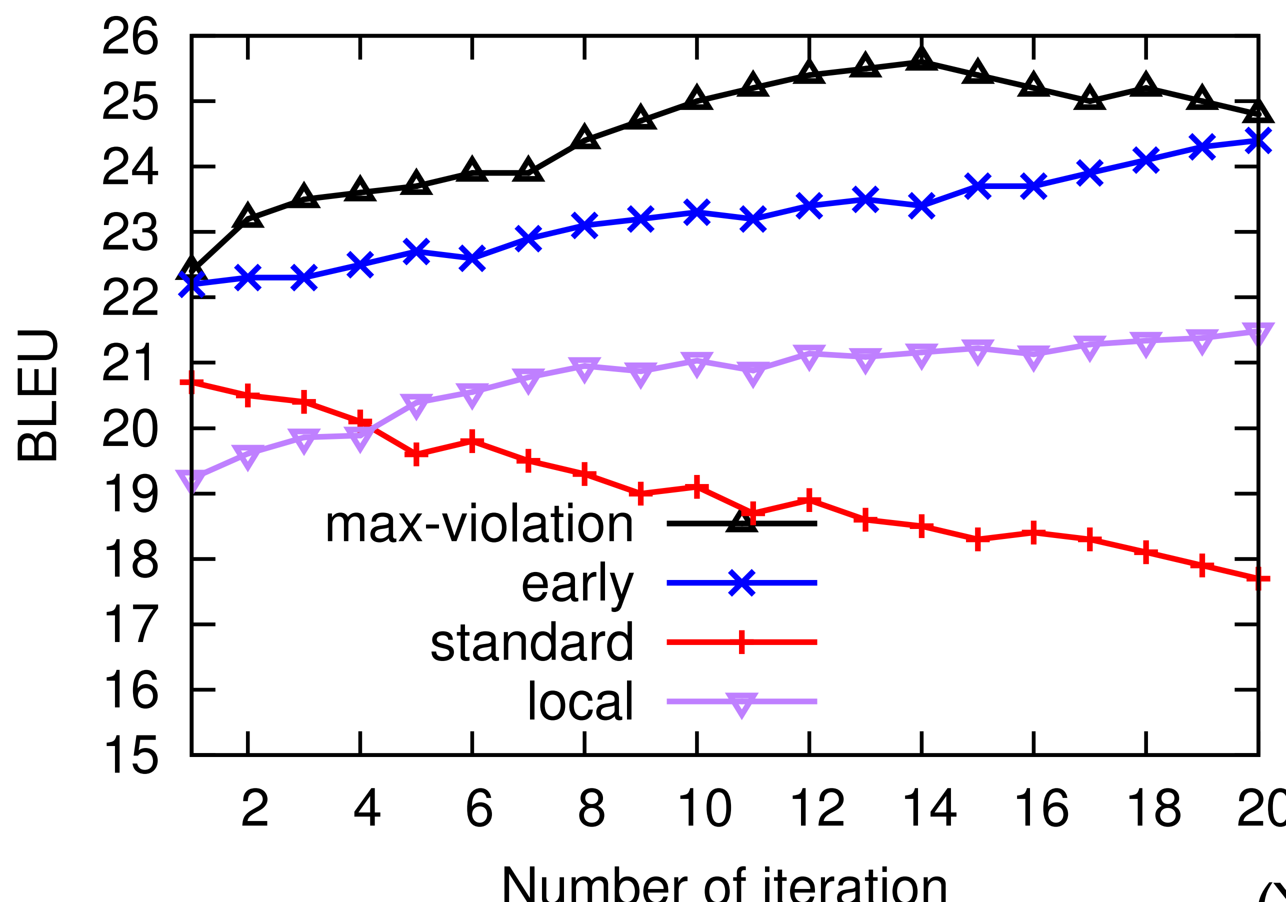


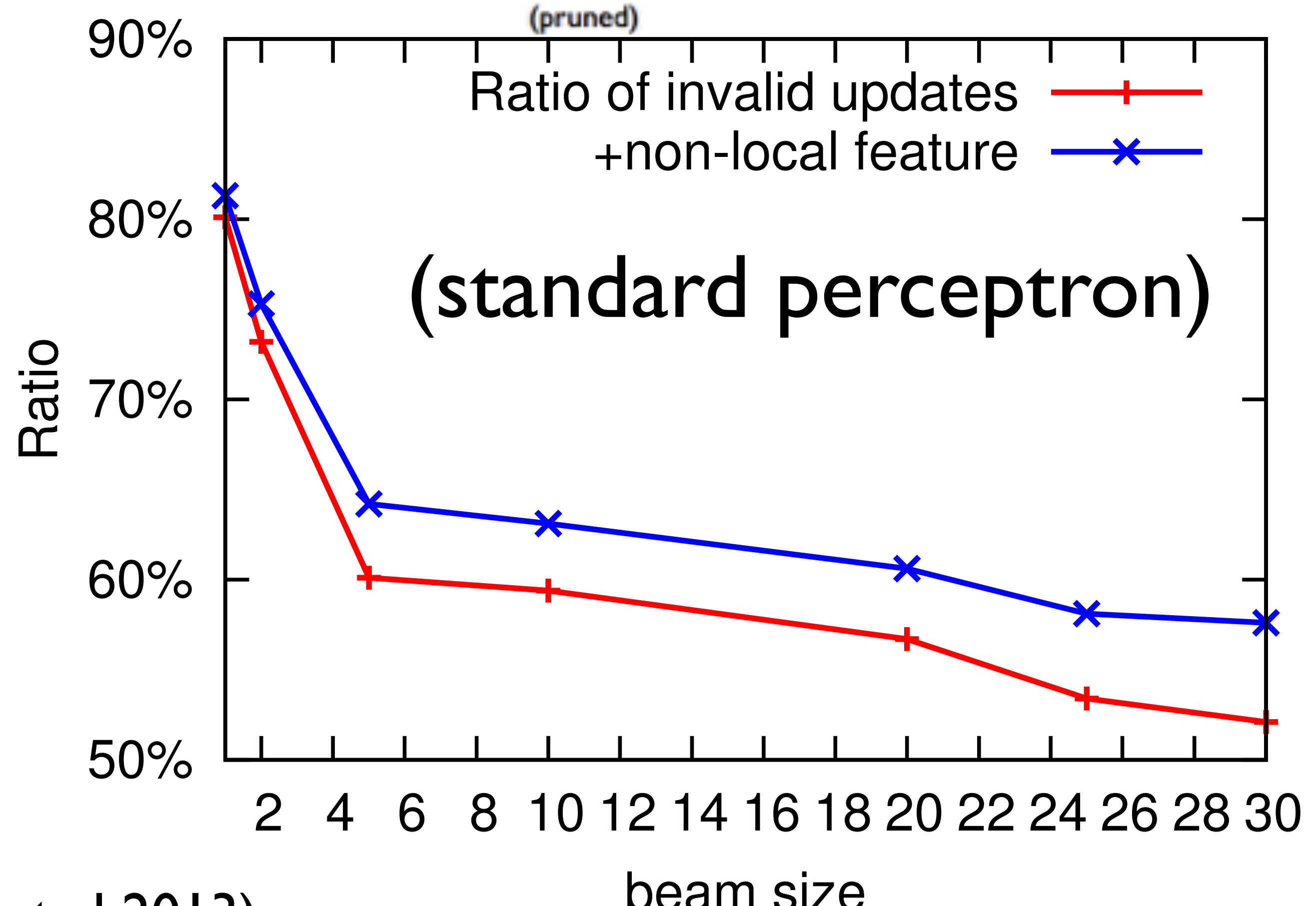
(Zhang et al 2013)

Exp 4: Machine Translation

- standard perceptron works poorly for machine translation
    - b/c invalid update ratio is very high (search quality is low)
- max-violation converges faster than early update
- first truly successful effort in large-scale training for translation



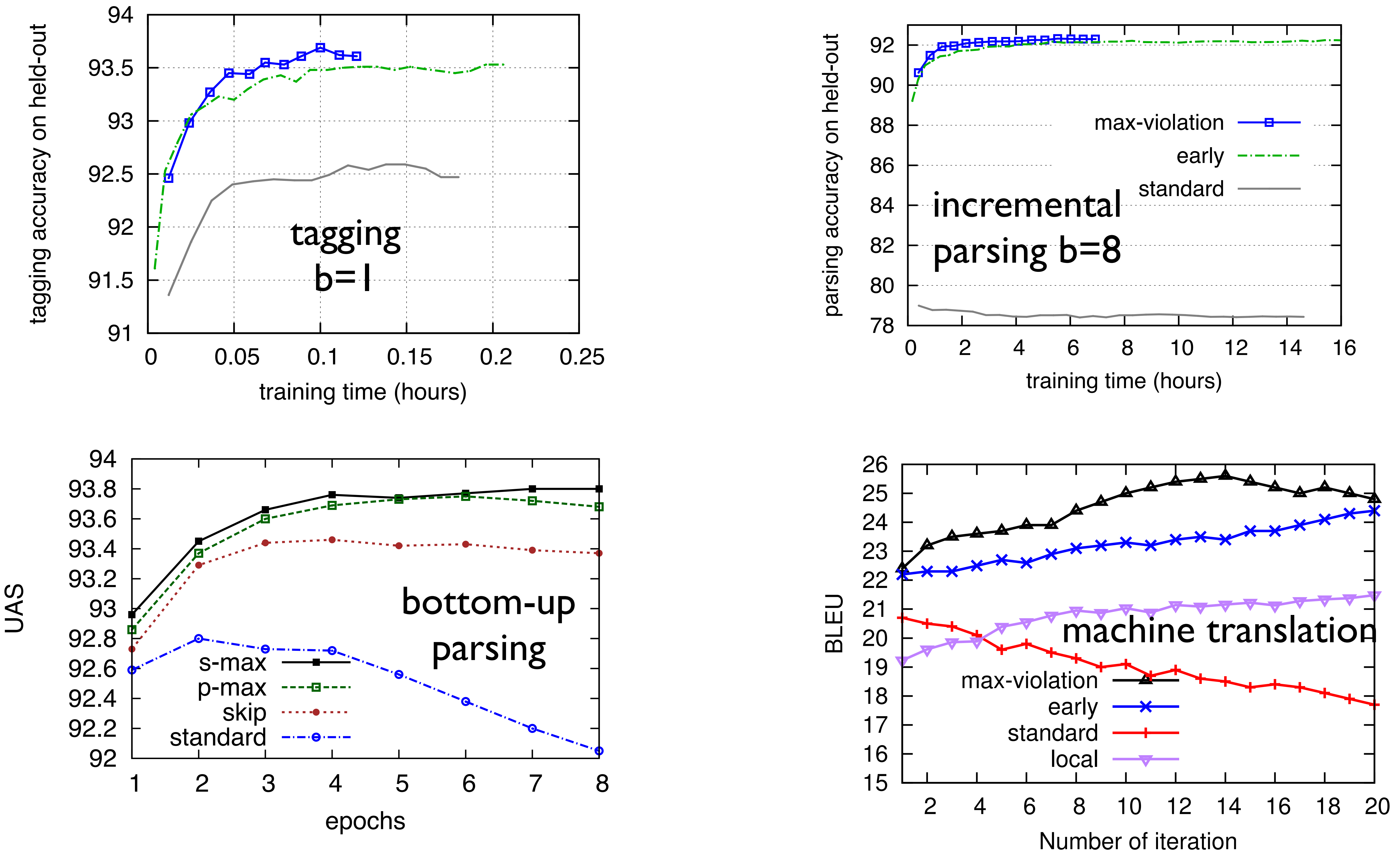




(Yu et al 2013)

Comparison of Four Exps

- the harder your search, the more advantageous



Outline

- Overview of Structured Learning
  - Challenges in Scalability
- Structured Perceptron
  - convergence proof
- Structured Perceptron with Inexact Search
- Latent-Variable Perceptron

Learning with Latent Variables

- aka "weakly-supervised" or "partially-observed" learning
- learning from "natural annotations"; more scalable
- examples: translation, transliteration, semantic parsing...

布什 与 沙龙 会谈 x

latent derivation

Bush talked with Sharon y

parallel text

(Liang et al 2006;
Yu et al 2013;
Xiao and Xiong 2013)

computer

latent derivation

コンピューター
ko n py u : ta :

transliteration

(Knight & Graehl, 1998;
Kondrak et al 2007, etc.)

What is the largest state?

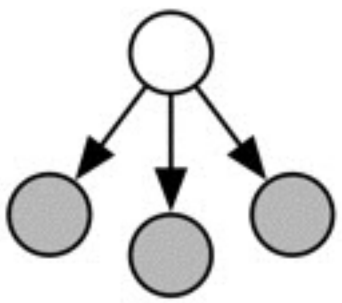
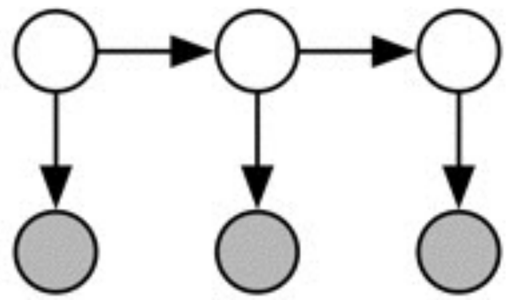
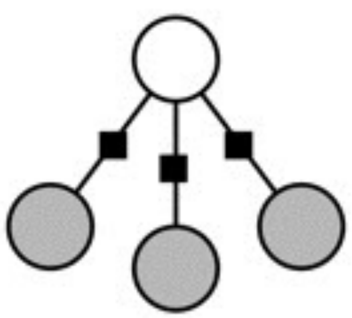
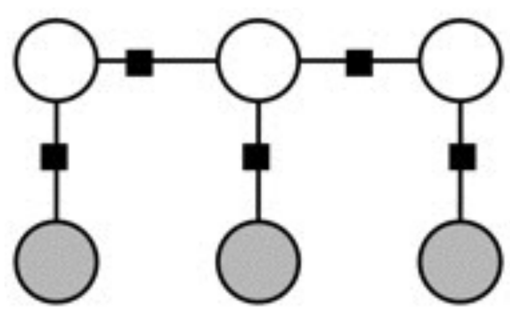
argmax(state, size)

Alaska

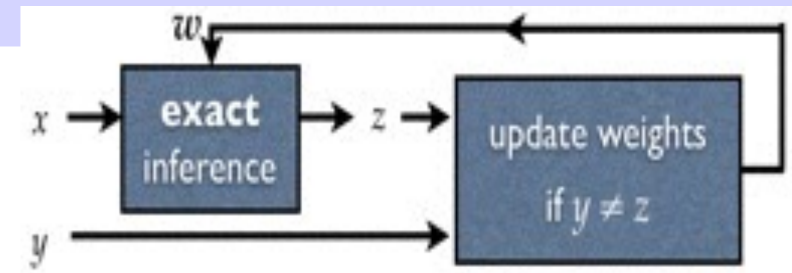
QA pairs

(Clark et al 2010;
Liang et al 2013;
Kwiatkowski et al 2013)

Learning Latent Structures

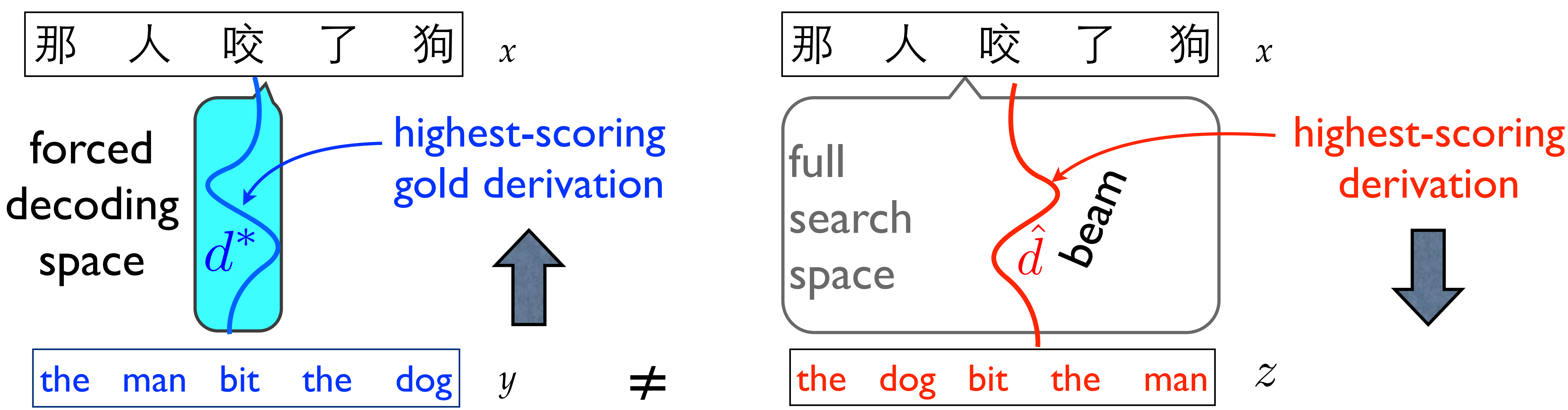
binary/multiclass		structured learning		latent structures
 naive bayes	→	 HMMs		EM (forward-backward)
↓ Conditional		↓ Conditional		
 logistic regression (maxent)	→	 CRFs		latent CRFs
↓ Online+ Viterbi		↓ Online+ Viterbi		
perceptron	→	structured perceptron		latent perceptron
↓ max margin		↓ max margin		
SVM	→	structured SVM		latent structured SVM

Latent Structured Perceptron



- no explicit positive signal
  - hallucinate the "correct" derivation by current weights

training example *during online learning...*



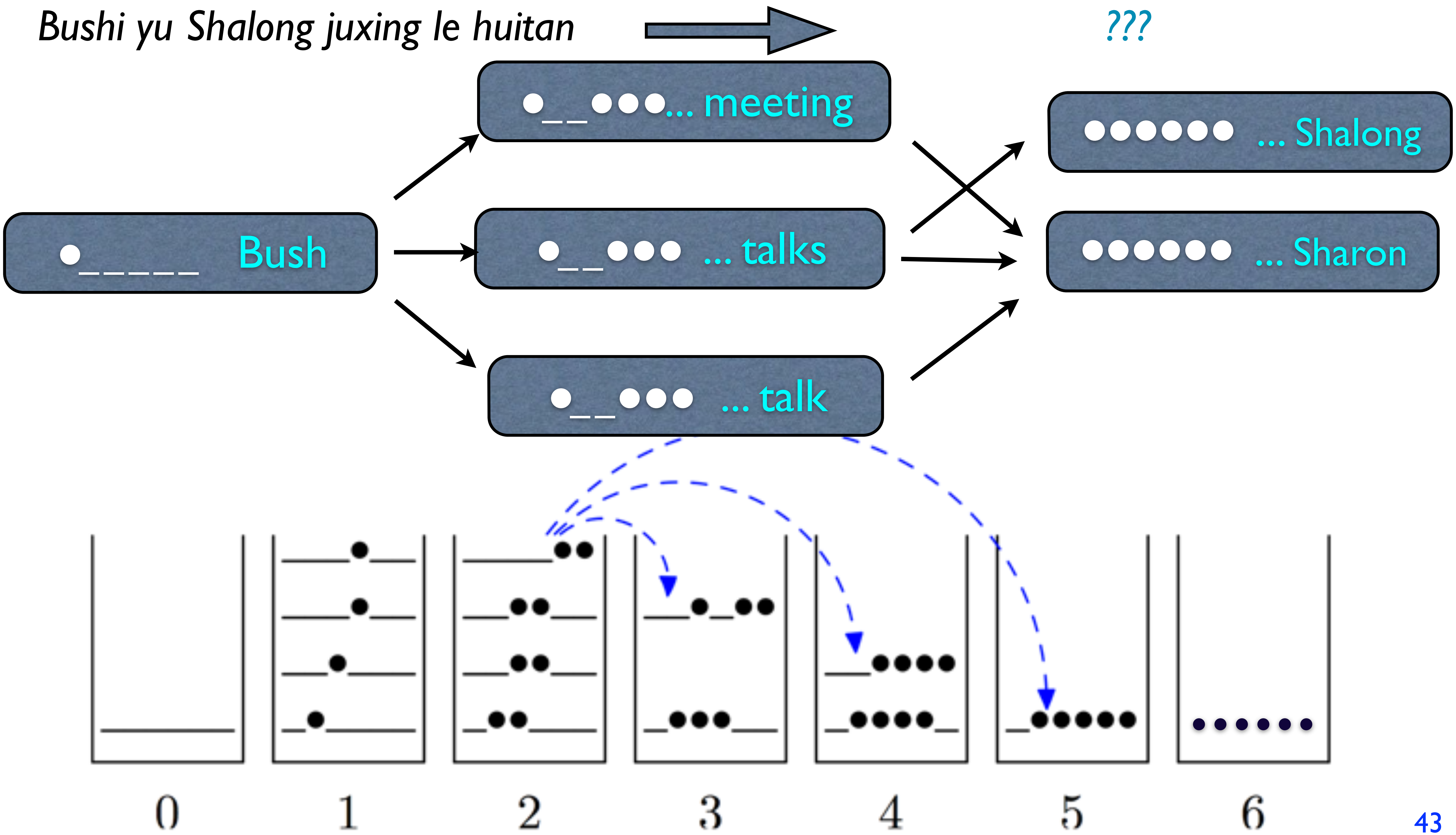
$$\mathbf{w} \leftarrow \mathbf{w} + \Phi(x, d^*) - \Phi(x, \hat{d})$$

reward correct penalize wrong

(Liang et al 2006; Yu et al 2013)

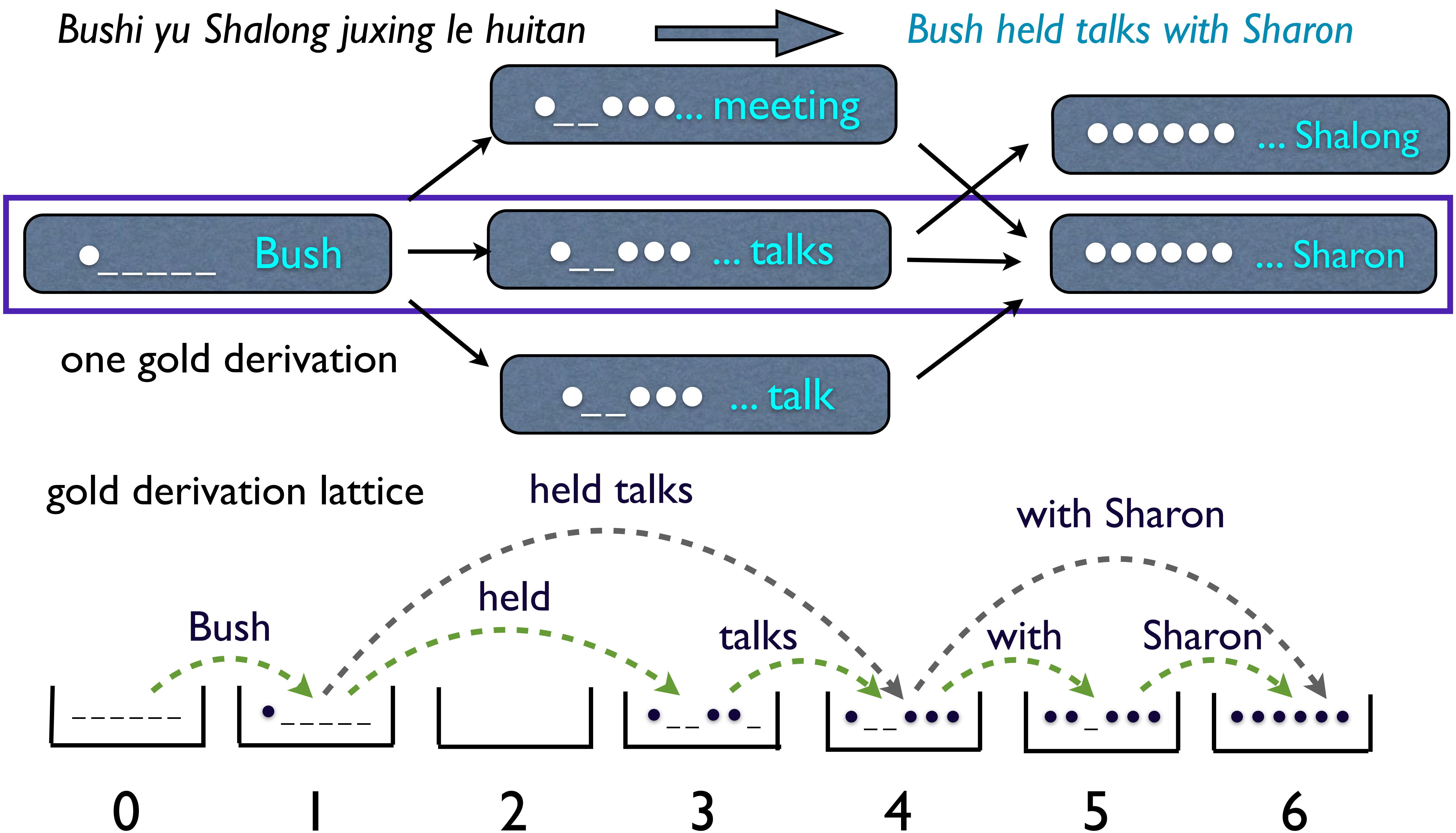
Unconstrained Search

- example: beam search phrase-based decoding

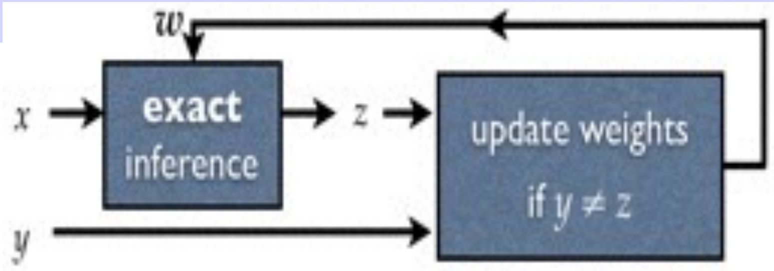


Constrained Search

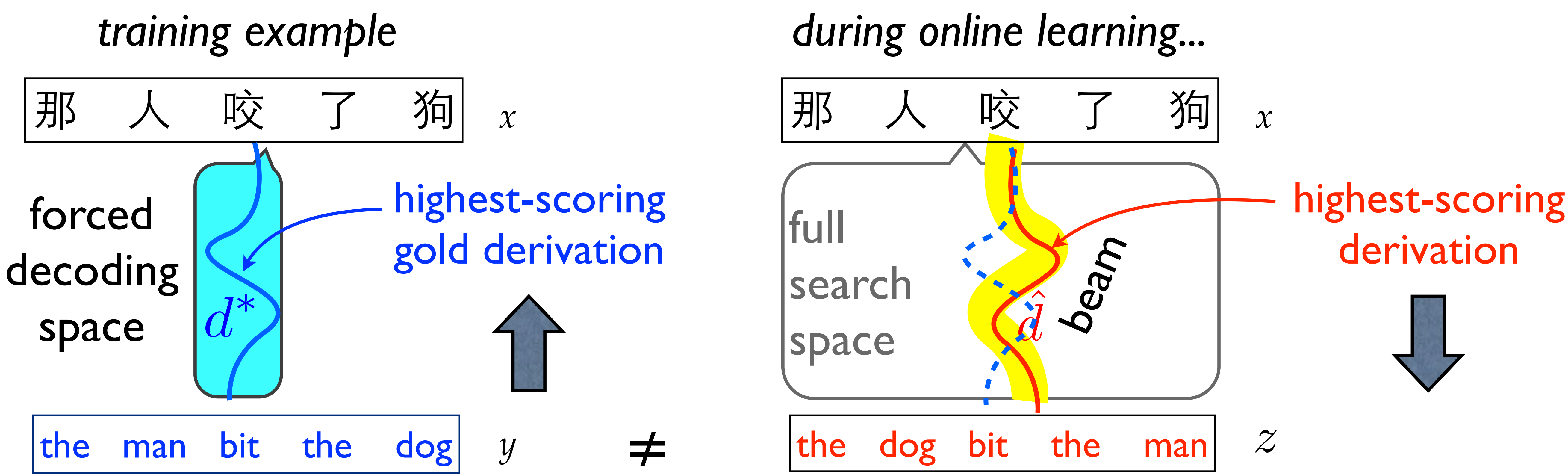
- forced decoding: must produce the exact reference translation



Search Errors in Decoding



- no explicit positive signal
  - hallucinate the "correct" derivation by current weights



$$\mathbf{w} \leftarrow \mathbf{w} + \Phi(x, d^*) - \Phi(x, \hat{d})$$

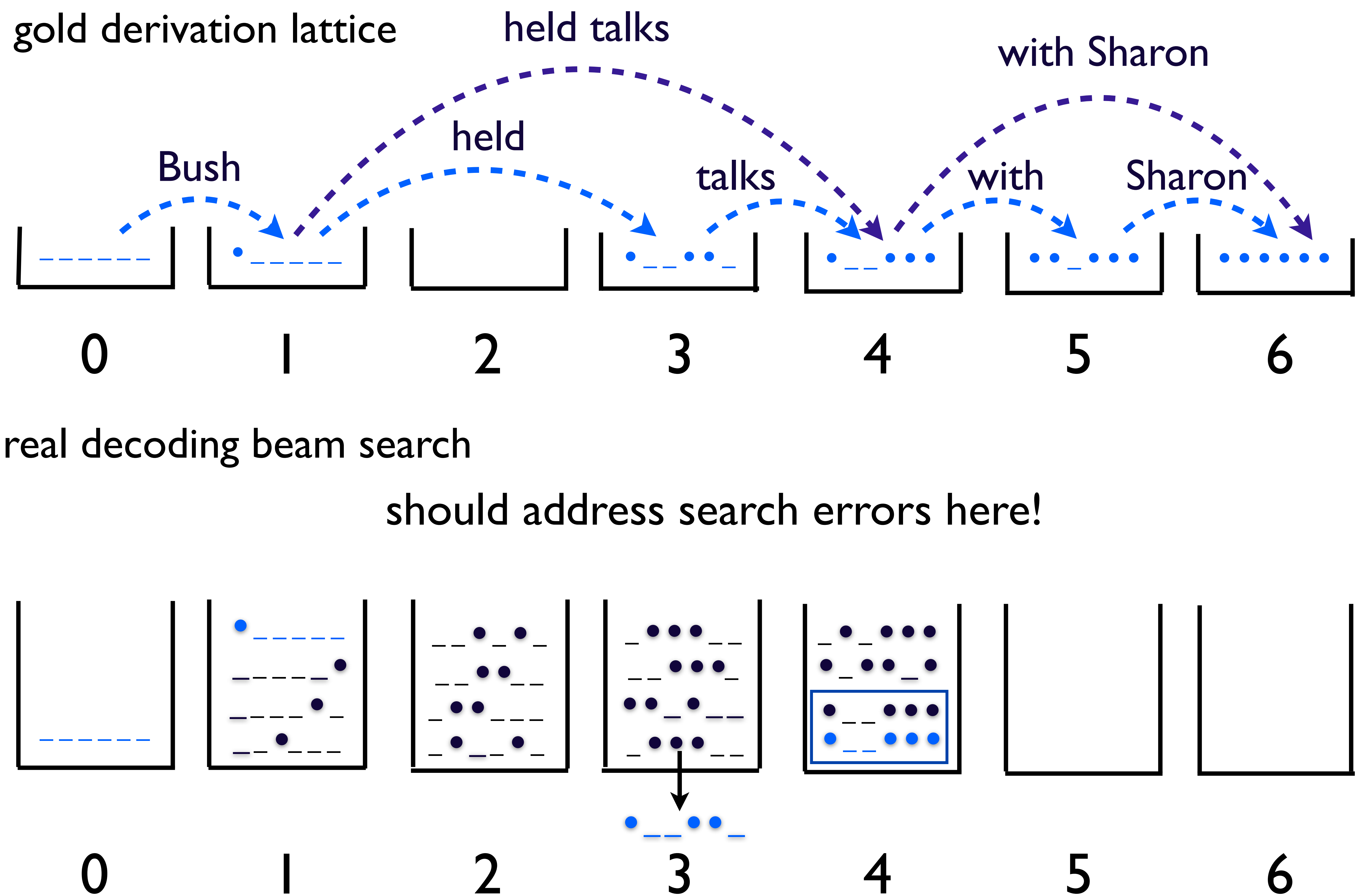
reward correct

penalize wrong

problem: search errors

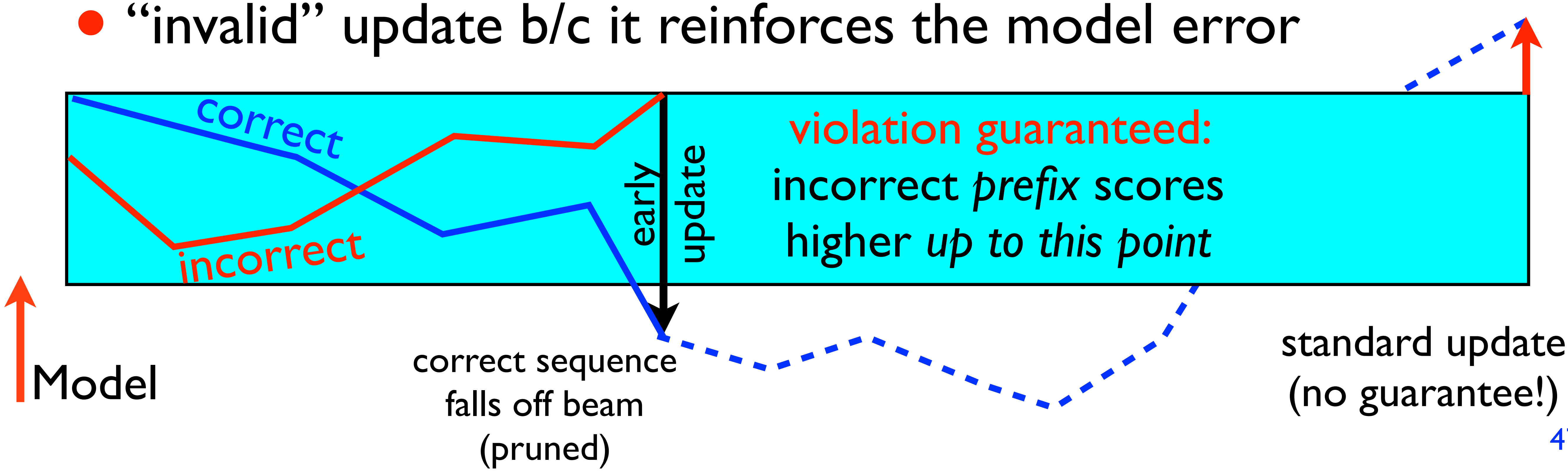
(Liang et al 2006; Yu et al 2013)

Search Error: Gold Derivations Pruned



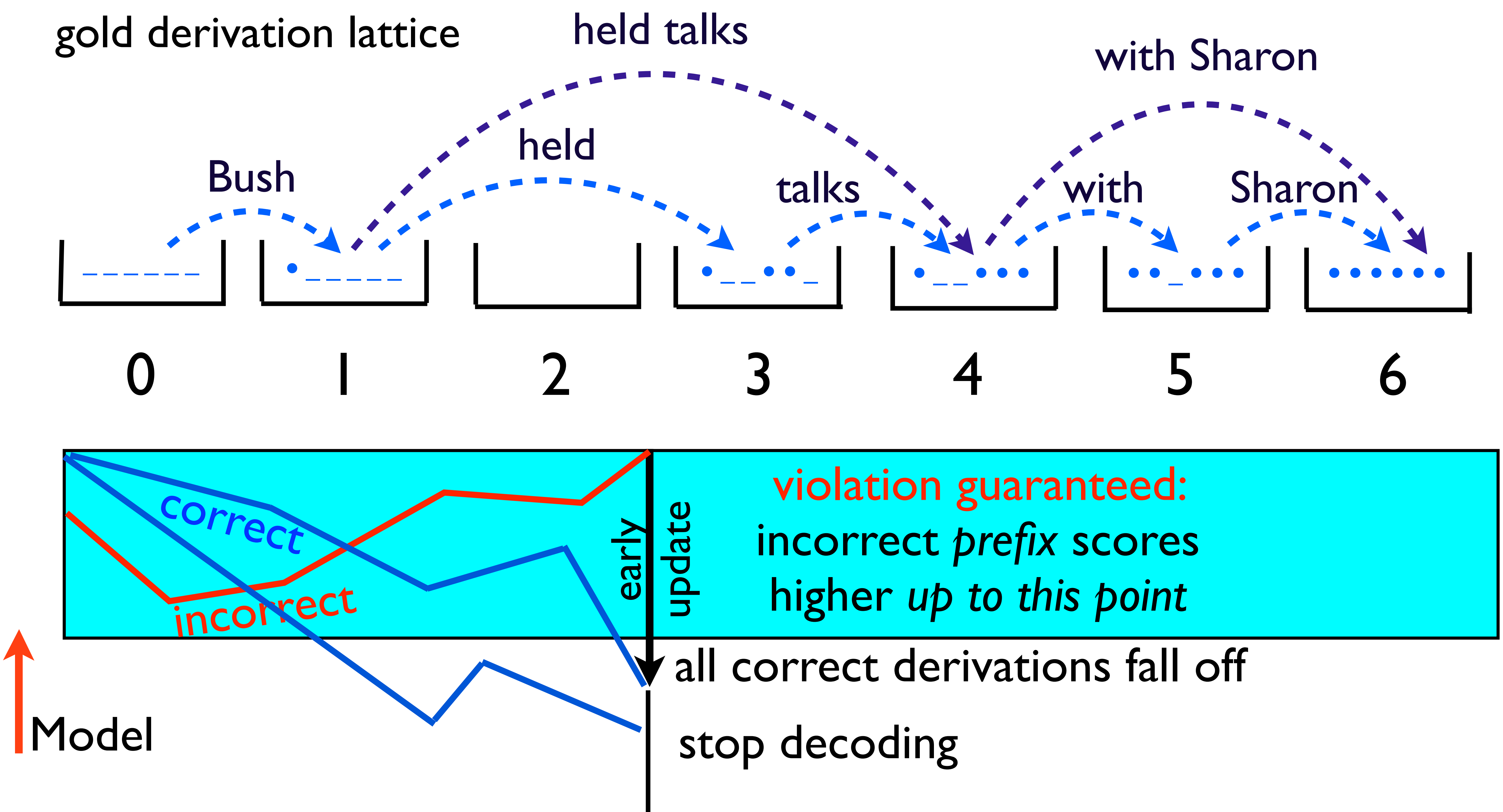
Fixing Search Error 1: Early Update

- early update (Collins/Roark'04) when the correct falls off beam
  - up to this point the incorrect prefix should score higher
  - that's a "violation" we want to fix; proof in (Huang et al 2012)
- standard perceptron does not guarantee violation
  - the correct sequence (pruned) might score higher at the end!
  - "invalid" update b/c it reinforces the model error

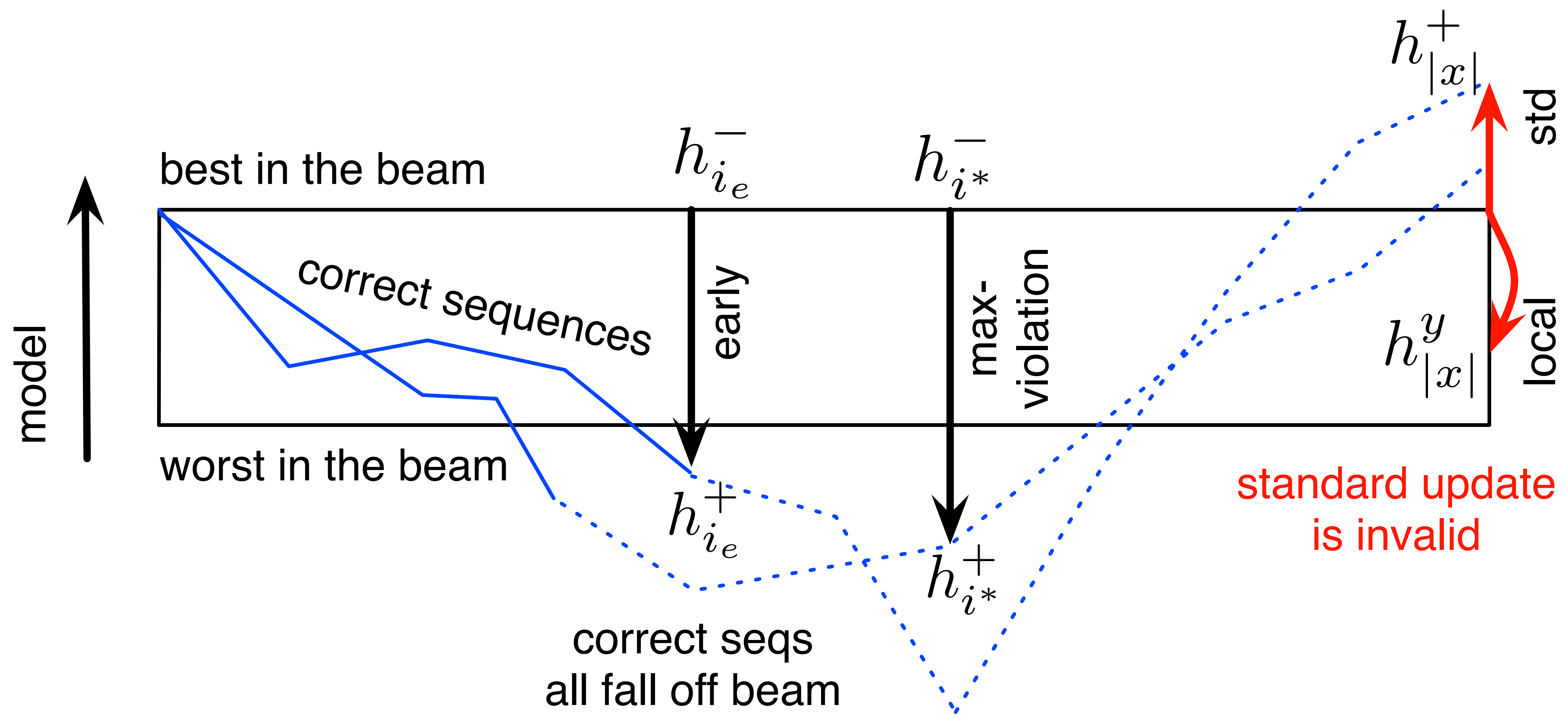


Early Update w/ Latent Variable

- the gold-standard derivations are ***not*** annotated
  - we treat any reference-producing derivation as good

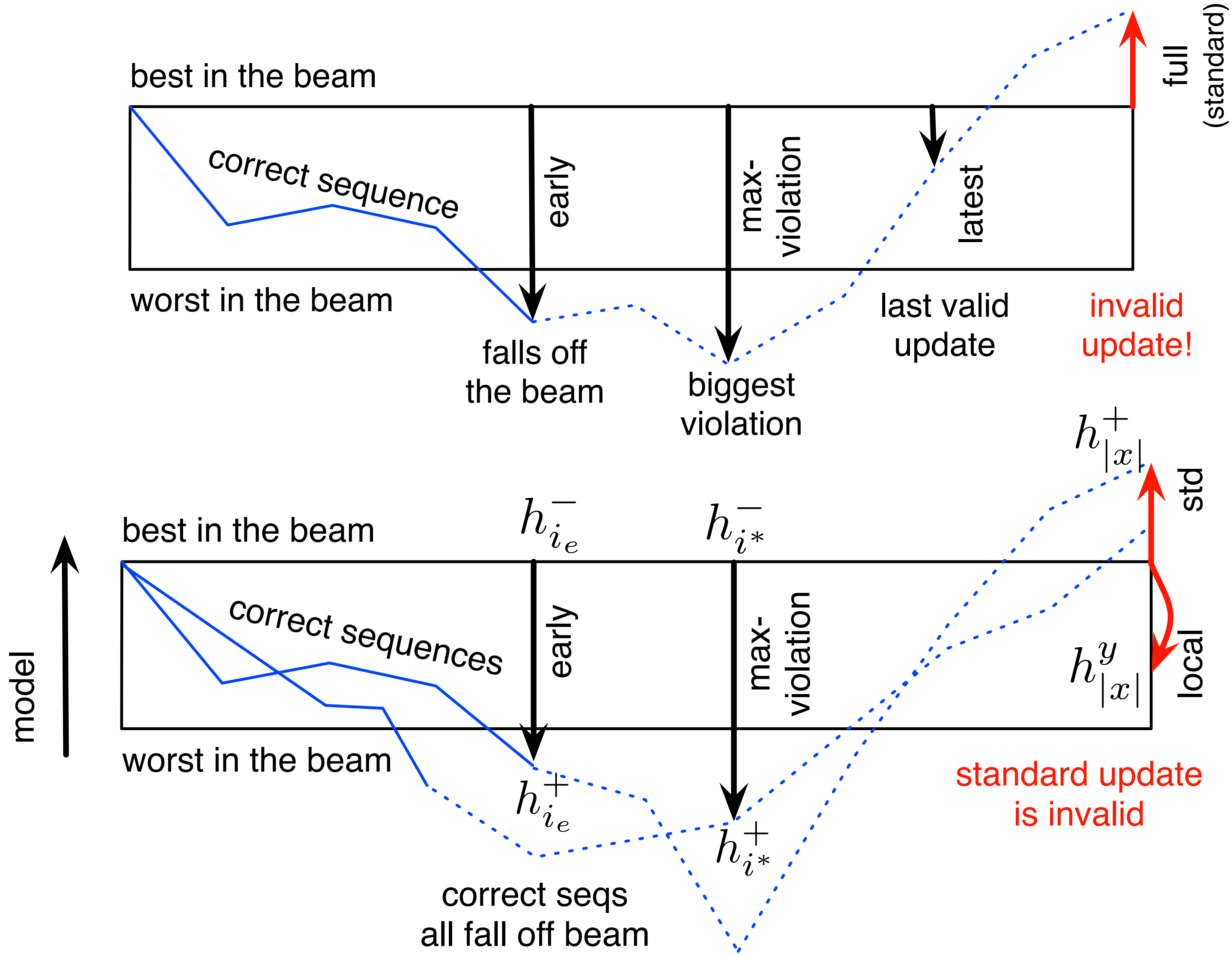


Fixing Search Error 2: Max-Violation

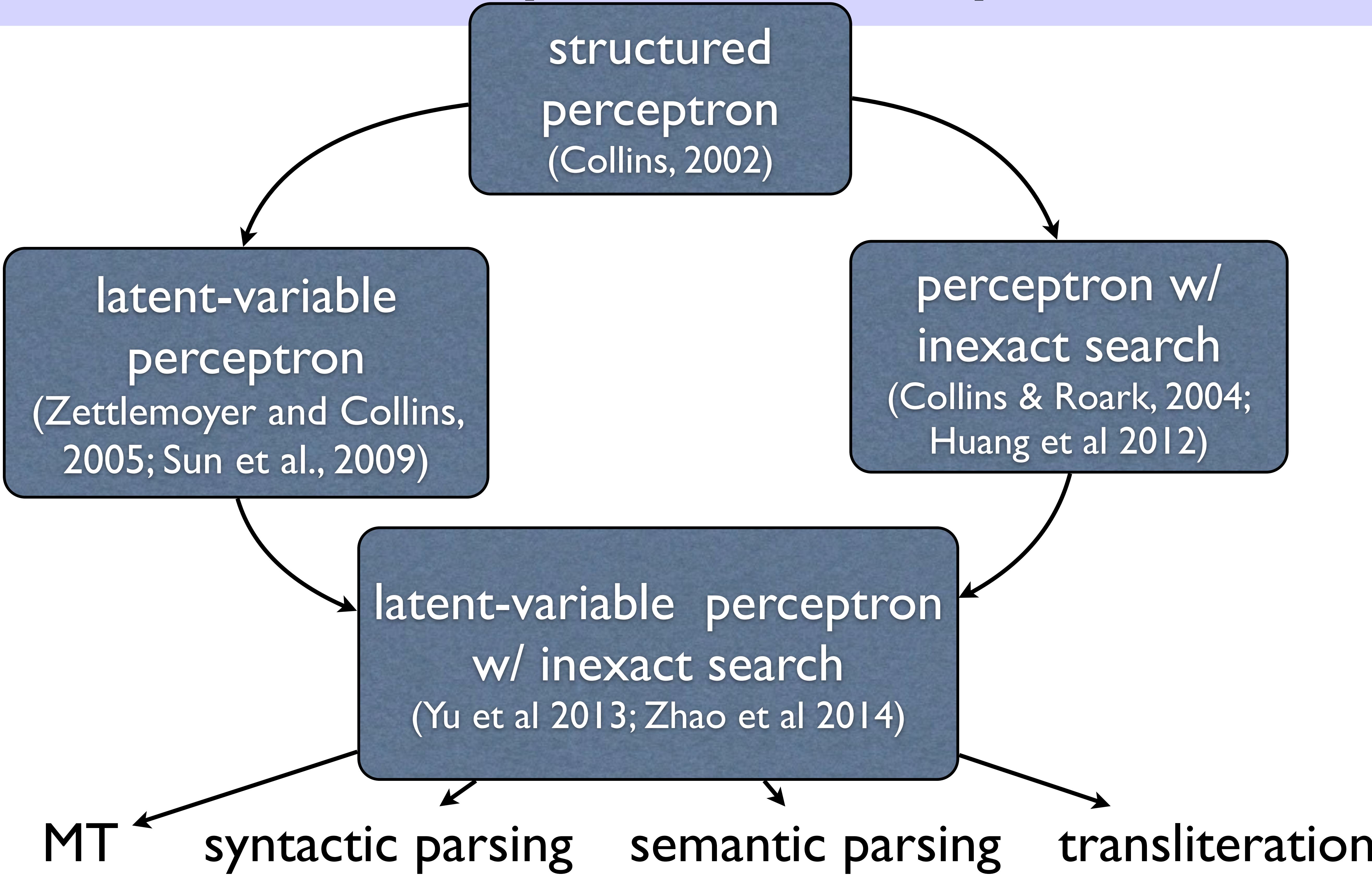


- early update works but learns slowly due to partial updates
- max-violation: use the prefix where violation is maximum
  - "worst-mistake" in the search space
  - now extended to handle latent-variable

Latent-Variable Perceptron

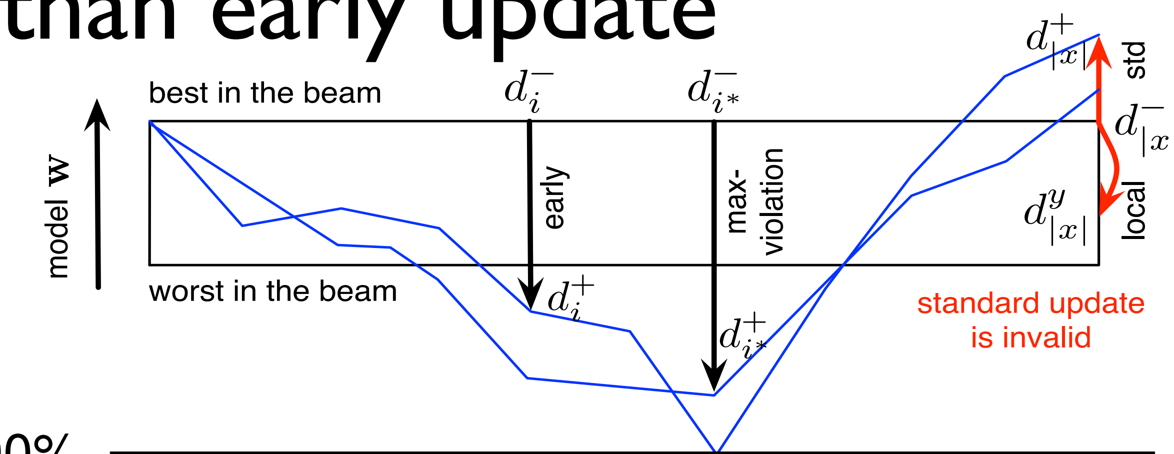


Roadmap of Techniques

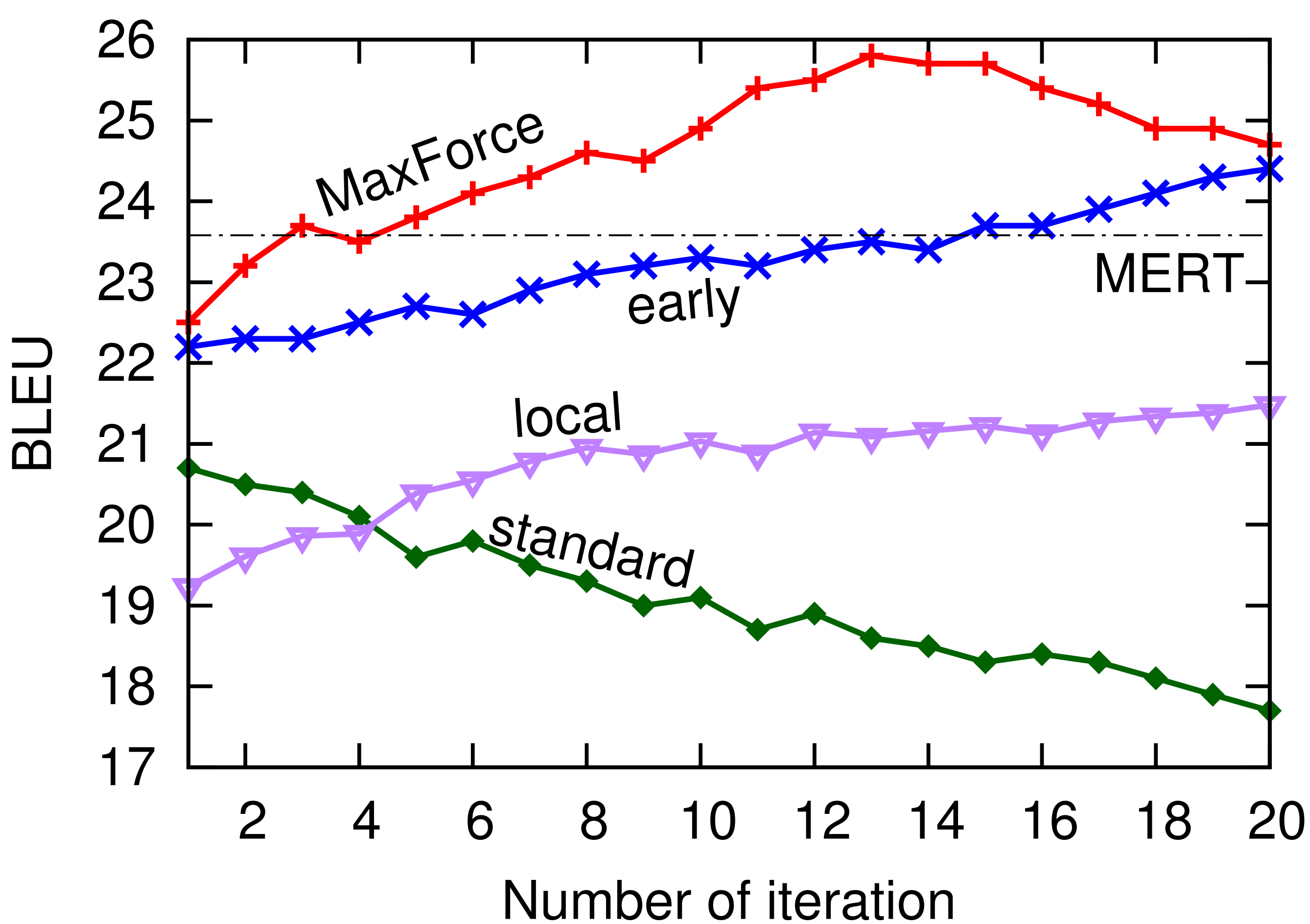


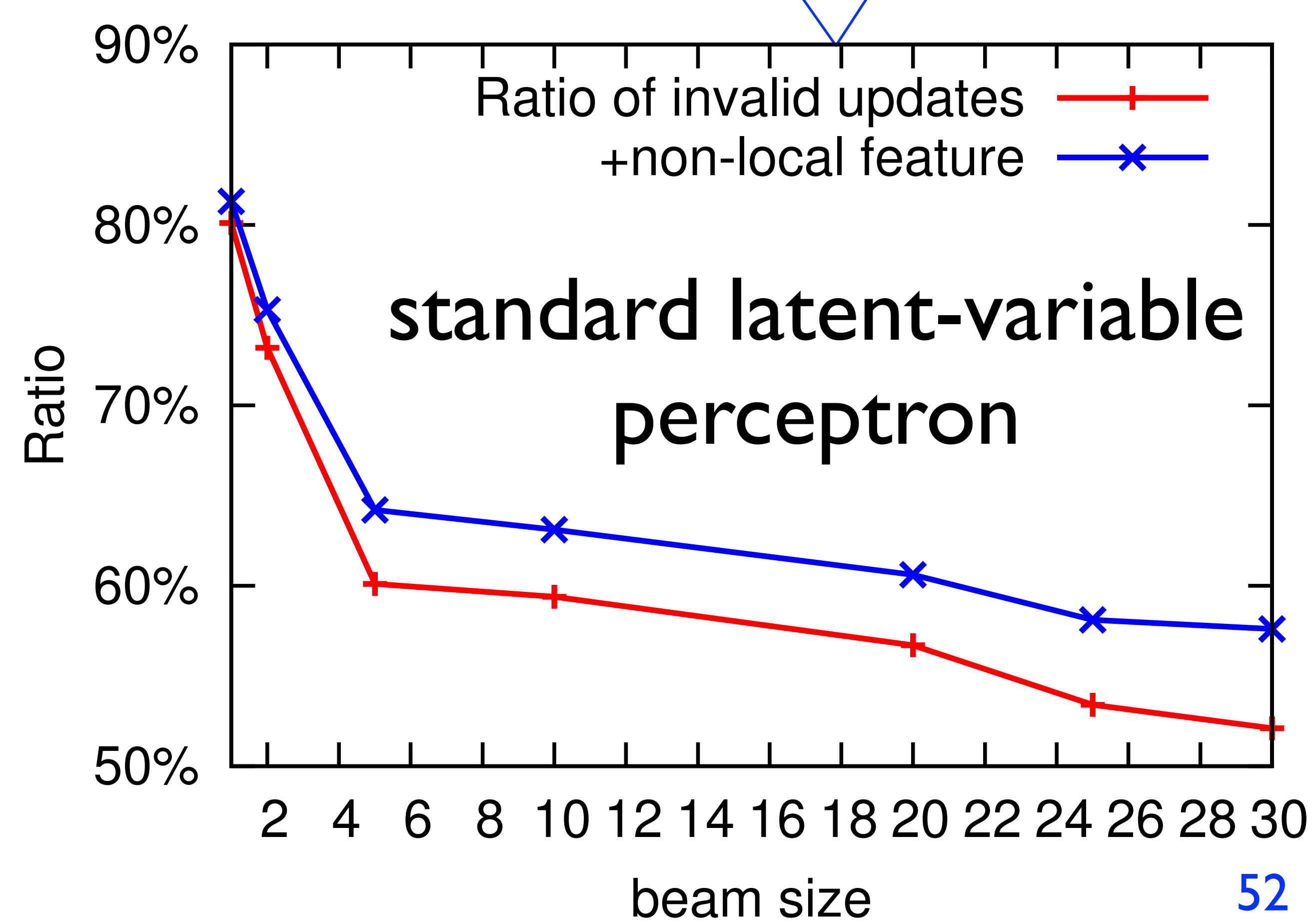
Experiments: Discriminative Training for MT

- standard update (Liang et al's "bold") works poorly
  - b/c invalid update ratio is very high (search quality is low)
- max-violation converges faster than early update



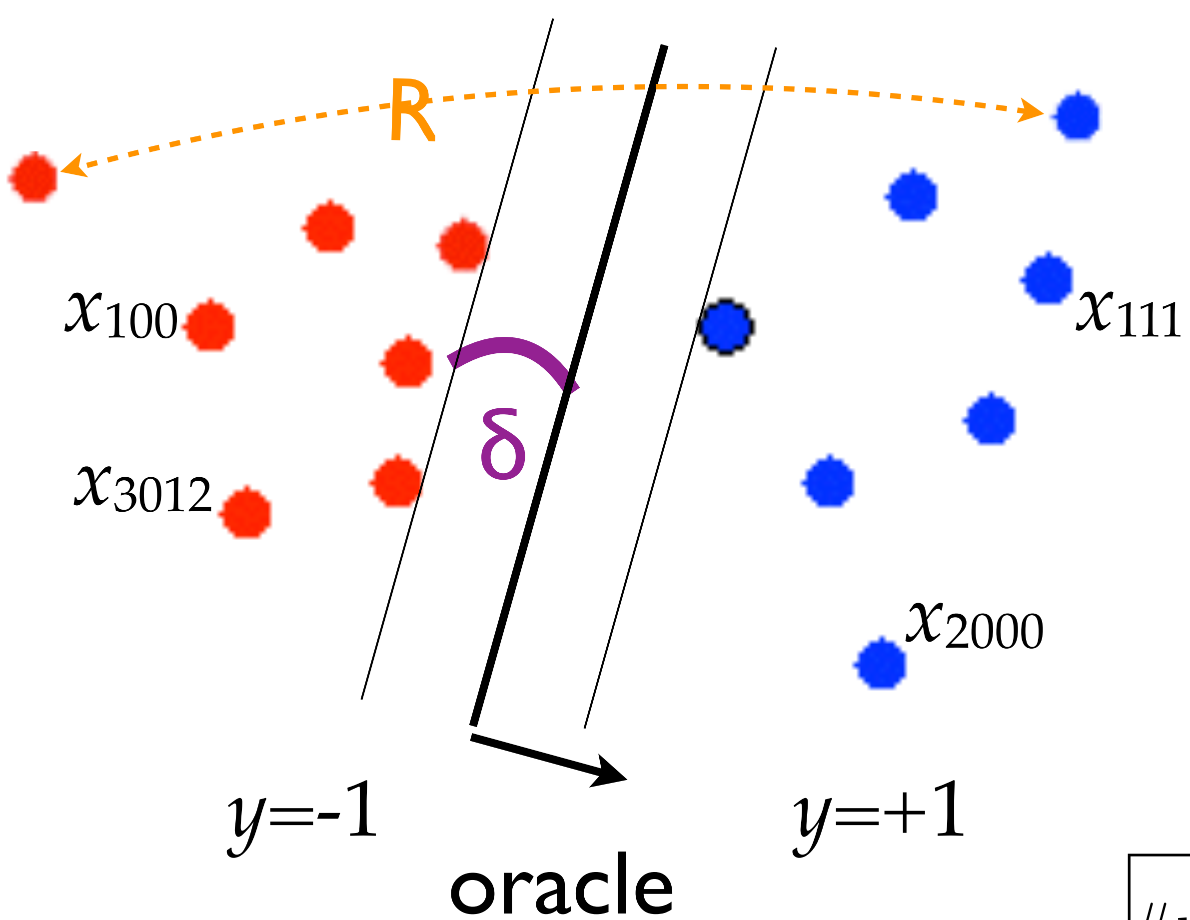
this explains why Liang et al '06 failed
std ~ "bold"; local ~ "local"





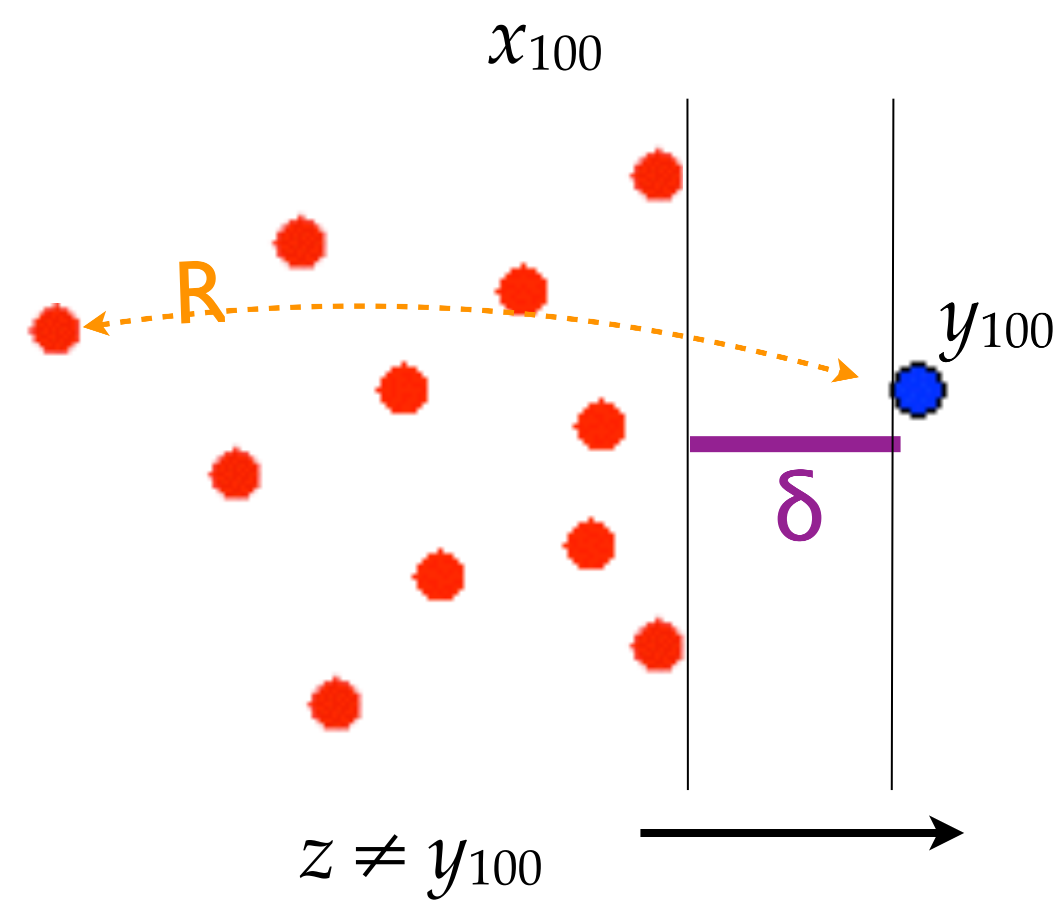
Open Problems in Theory

- latent-variable structured perceptron:
  - does it converge? under what conditions?
  - special case: POS tagging (Sun et al., 2009)
- latent-variable structured perceptron with inexact search
  - does it converge? under what conditions?



$$\#\text{updates} \leq \frac{R^2}{\delta^2}$$

(Novikoff, 1962)

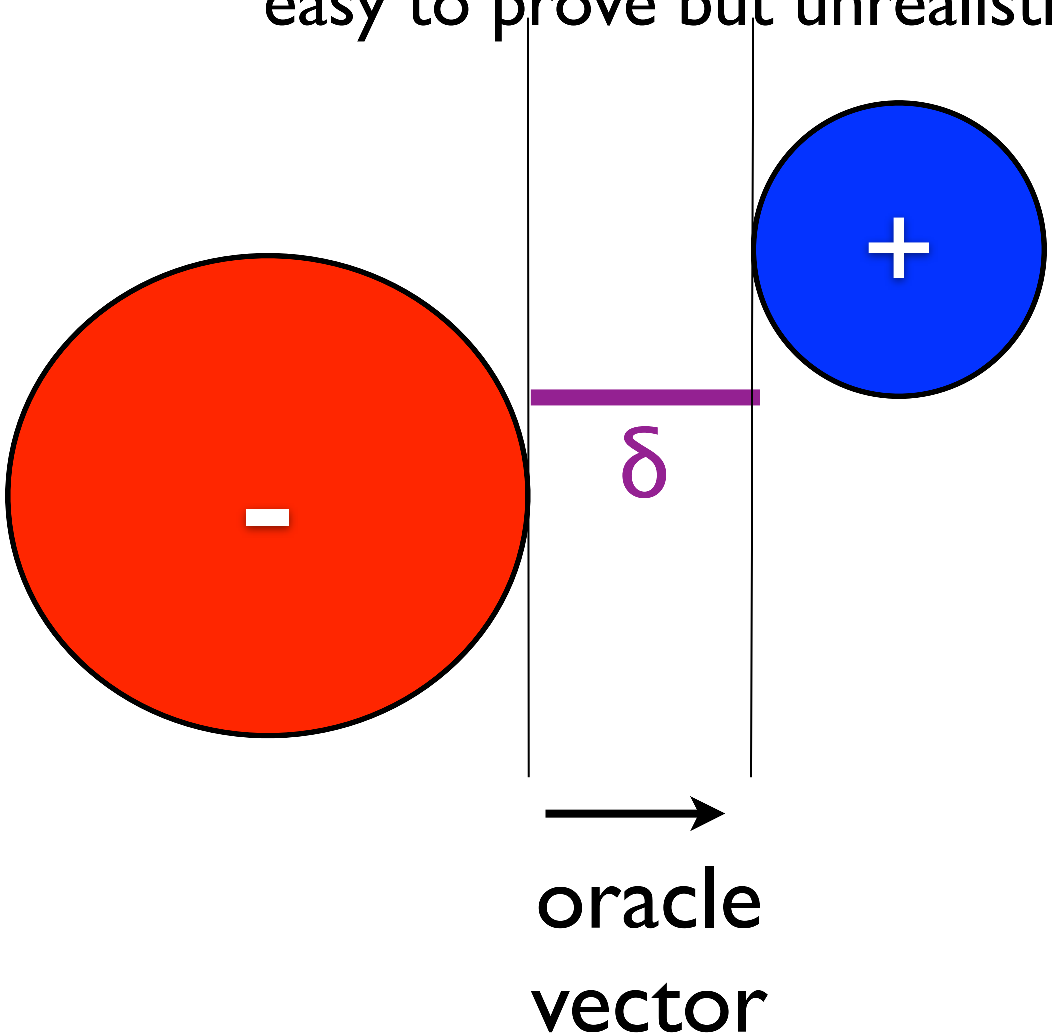


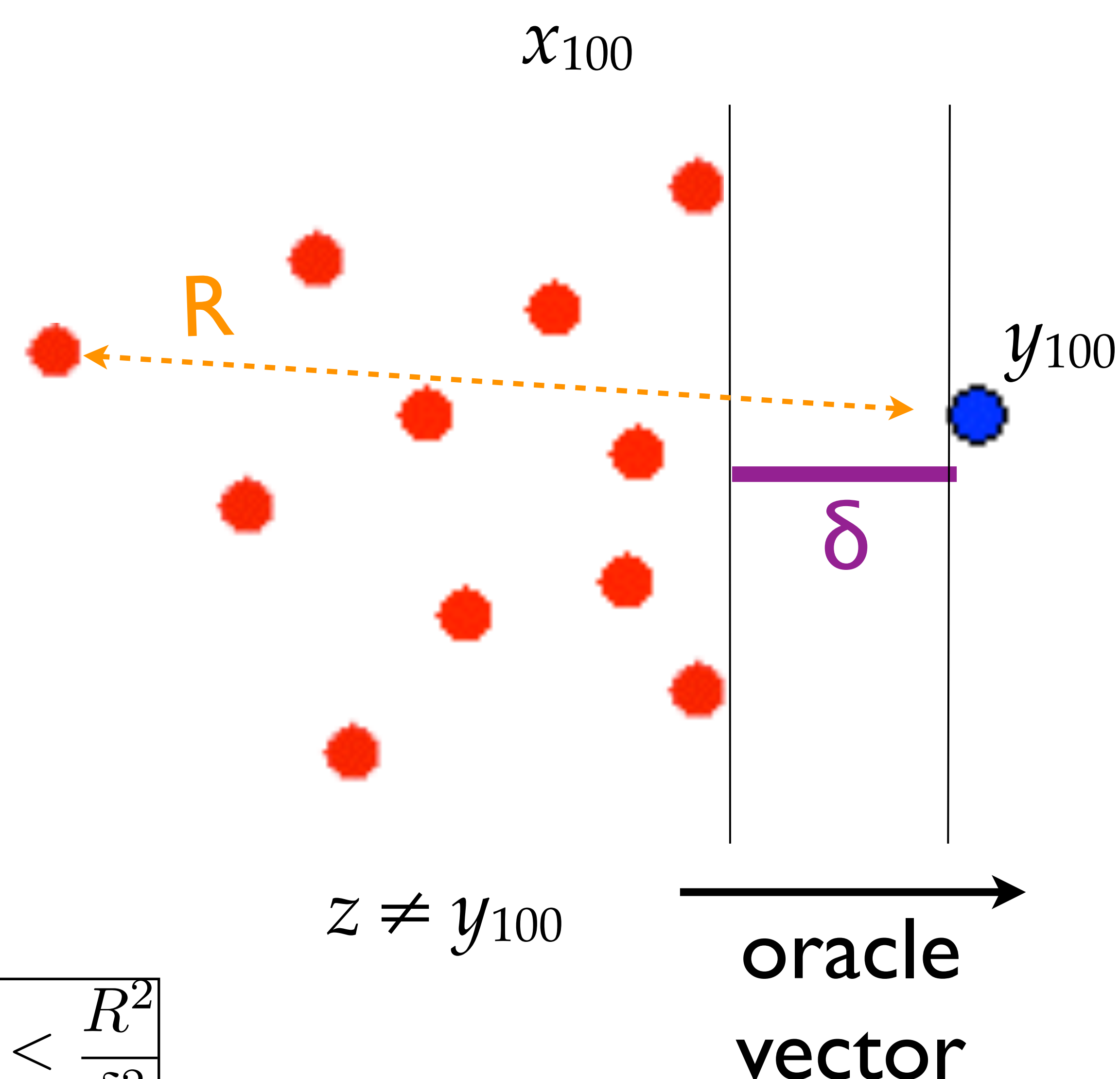
(Collins, 2002)

Open Problems in Theory

- latent-variable structured perceptron:
  - does it converge? under what conditions?
- latent-variable structured perceptron with inexact search
  - does it converge? under what conditions?

easy to prove but unrealistic





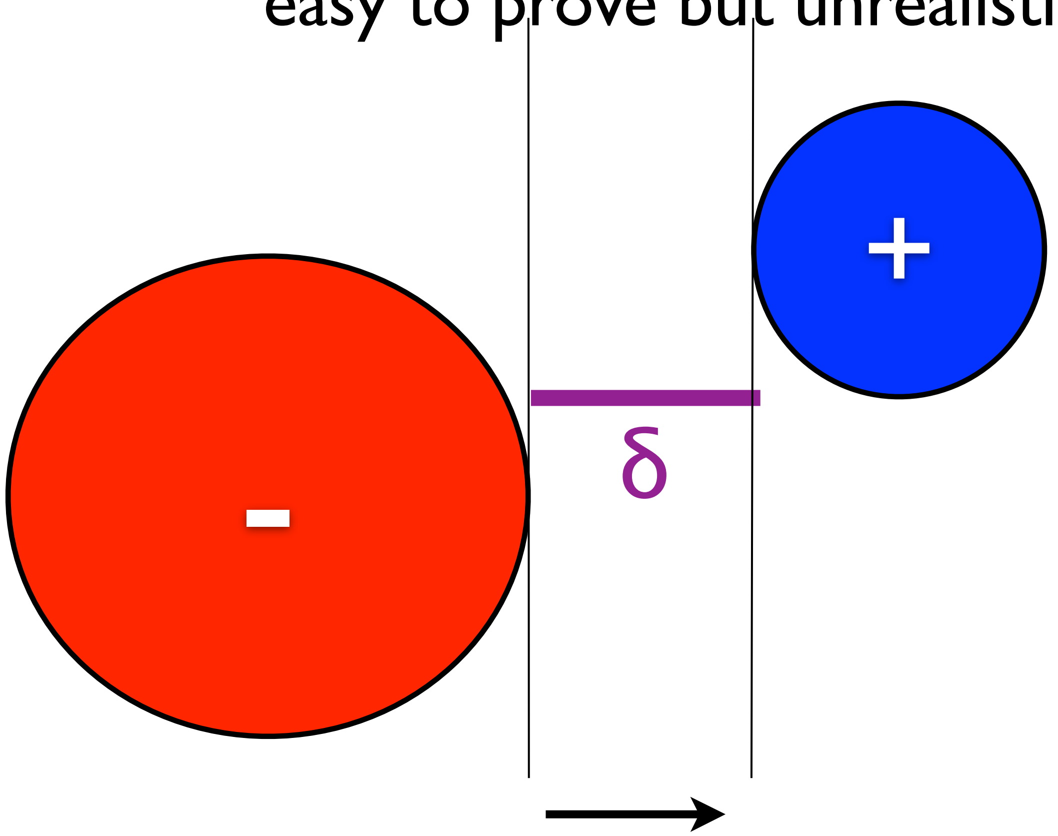
$$\boxed{\#\text{updates} \leq \frac{R^2}{\delta^2}}$$

(Collins, 2002)

Open Problems in Theory

- latent-variable structured perceptron:
  - does it converge? under what conditions?
- latent-variable structured perceptron with inexact search
  - does it converge? under what conditions?

easy to prove but unrealistic

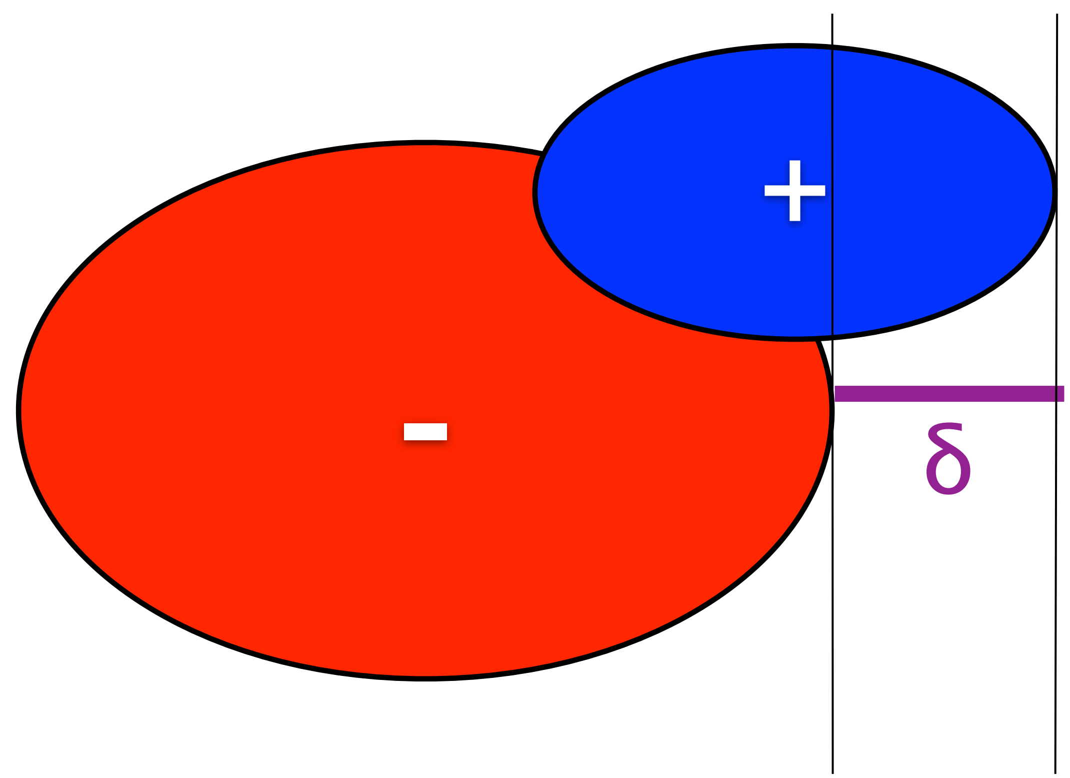


oracle vector

(Sun et al, 2009)

$$\boxed{\#\text{updates} \leq \frac{R^2}{\delta^2}}$$

ideal situation but hard to prove



???

Final Conclusions

- online structured learning is simple and powerful
- search efficiency is the key challenge
- search errors do interfere with learning
  - but we can use violation-fixing perceptron w/ inexact search
- we can extend perceptron to learn latent structures