Backpropagating through Structured Argmax using a SPIGOT

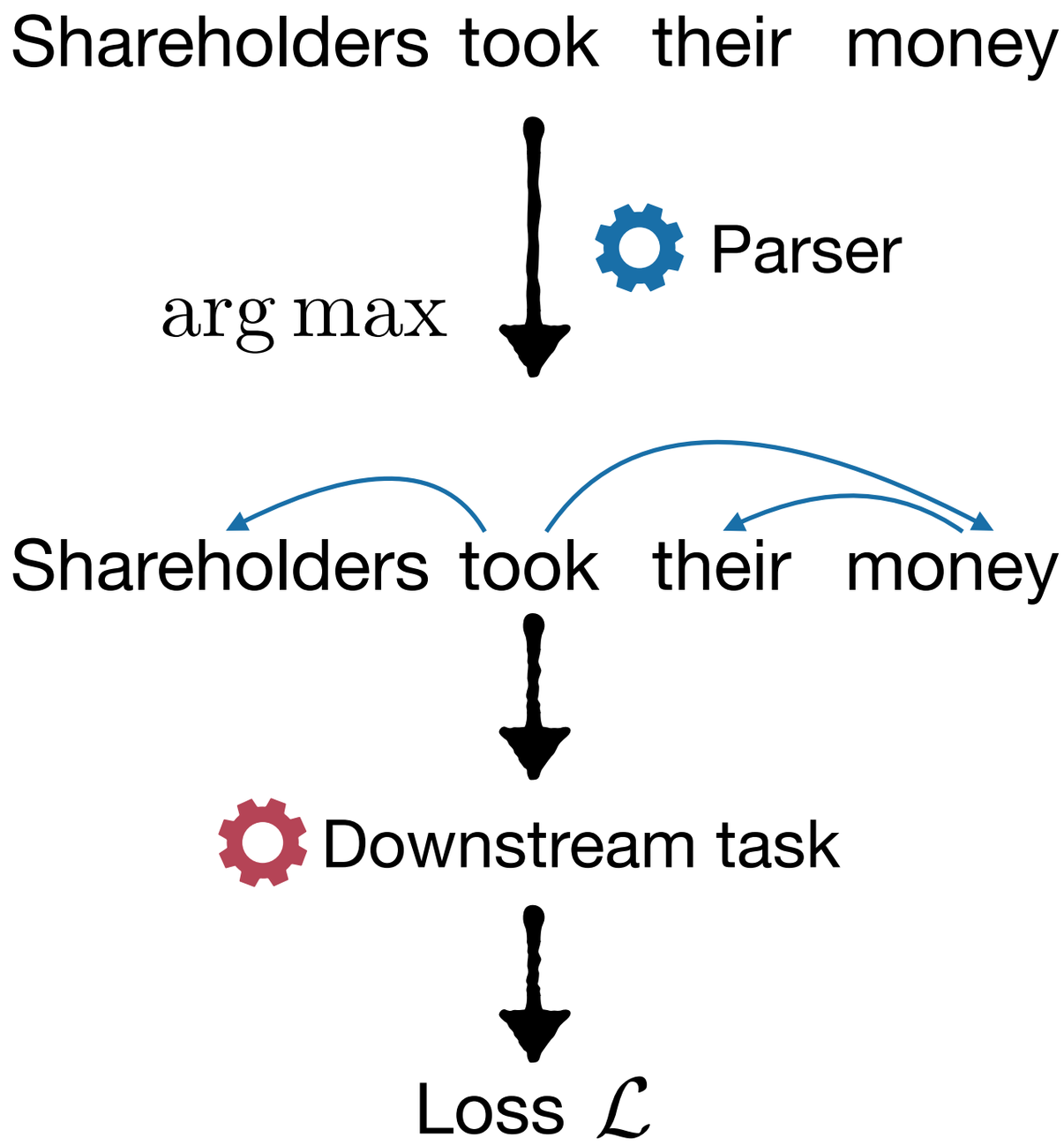
Hao Peng, Sam Thomson, Noah A. Smith

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

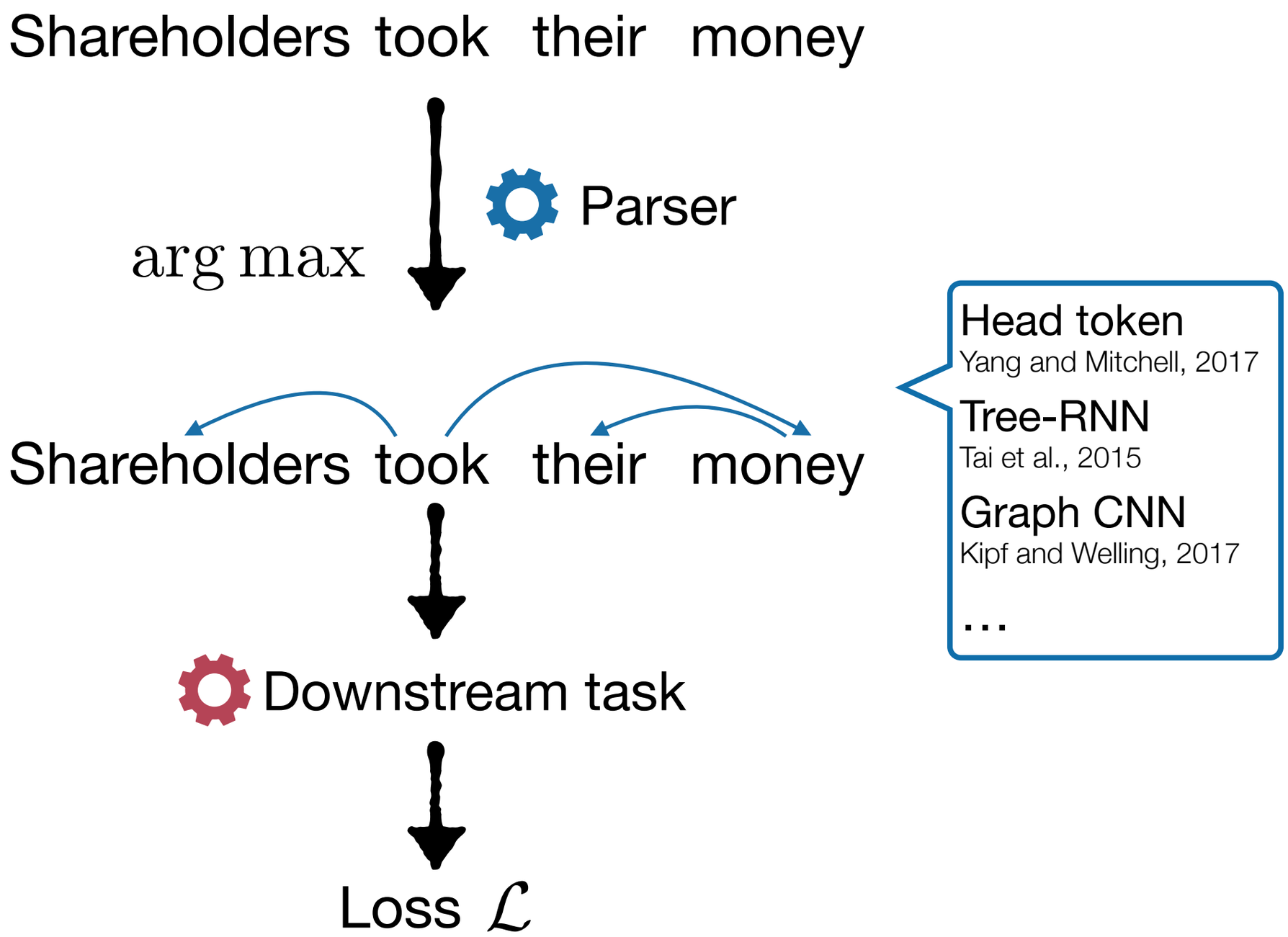
Language
Technologies
Institute

@ACL
July 17, 2018

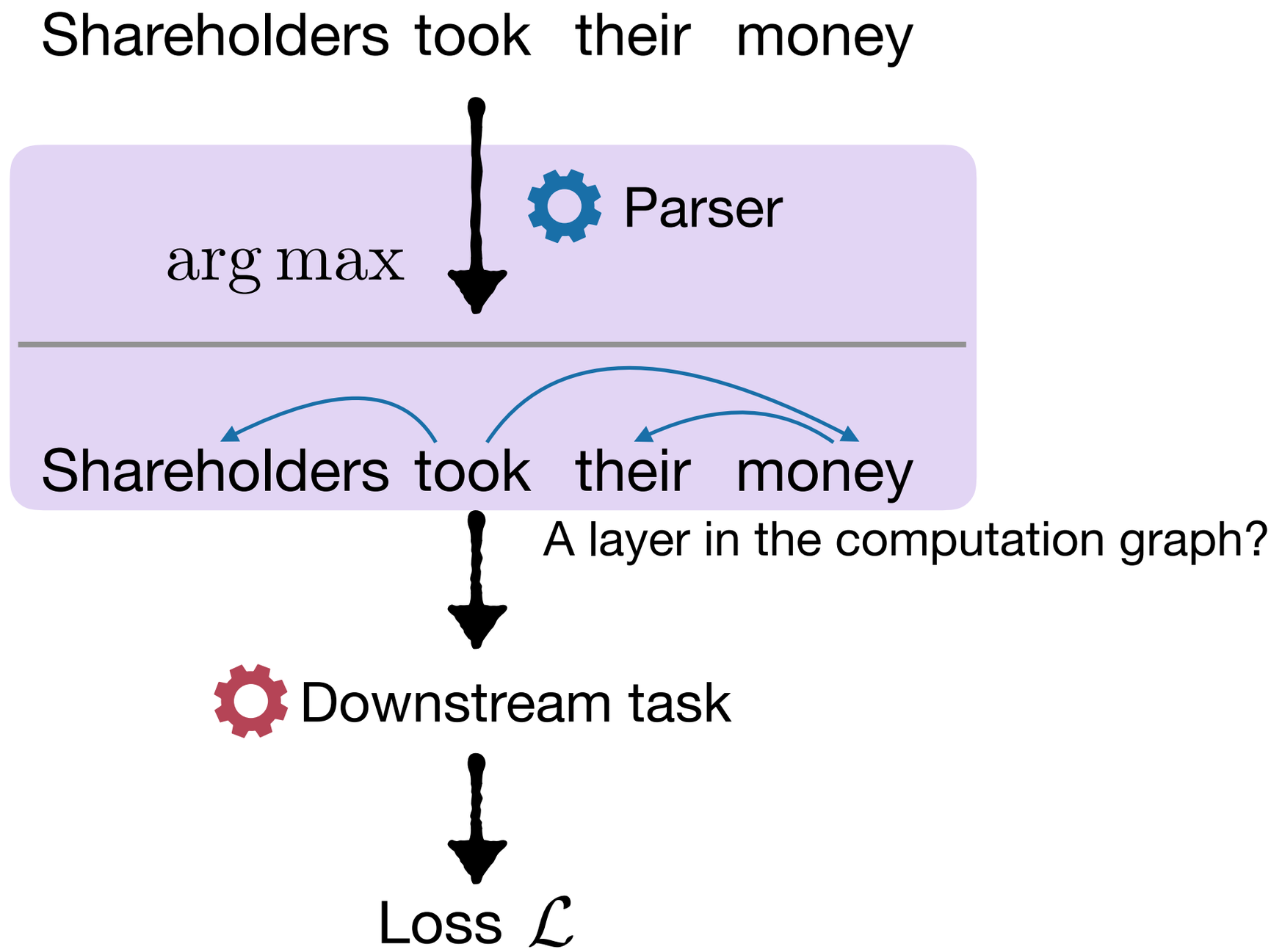
Overview


Shareholders took their money
arg max
Parser
Shareholders took their money
Downstream task
Loss $\mathcal{L}$

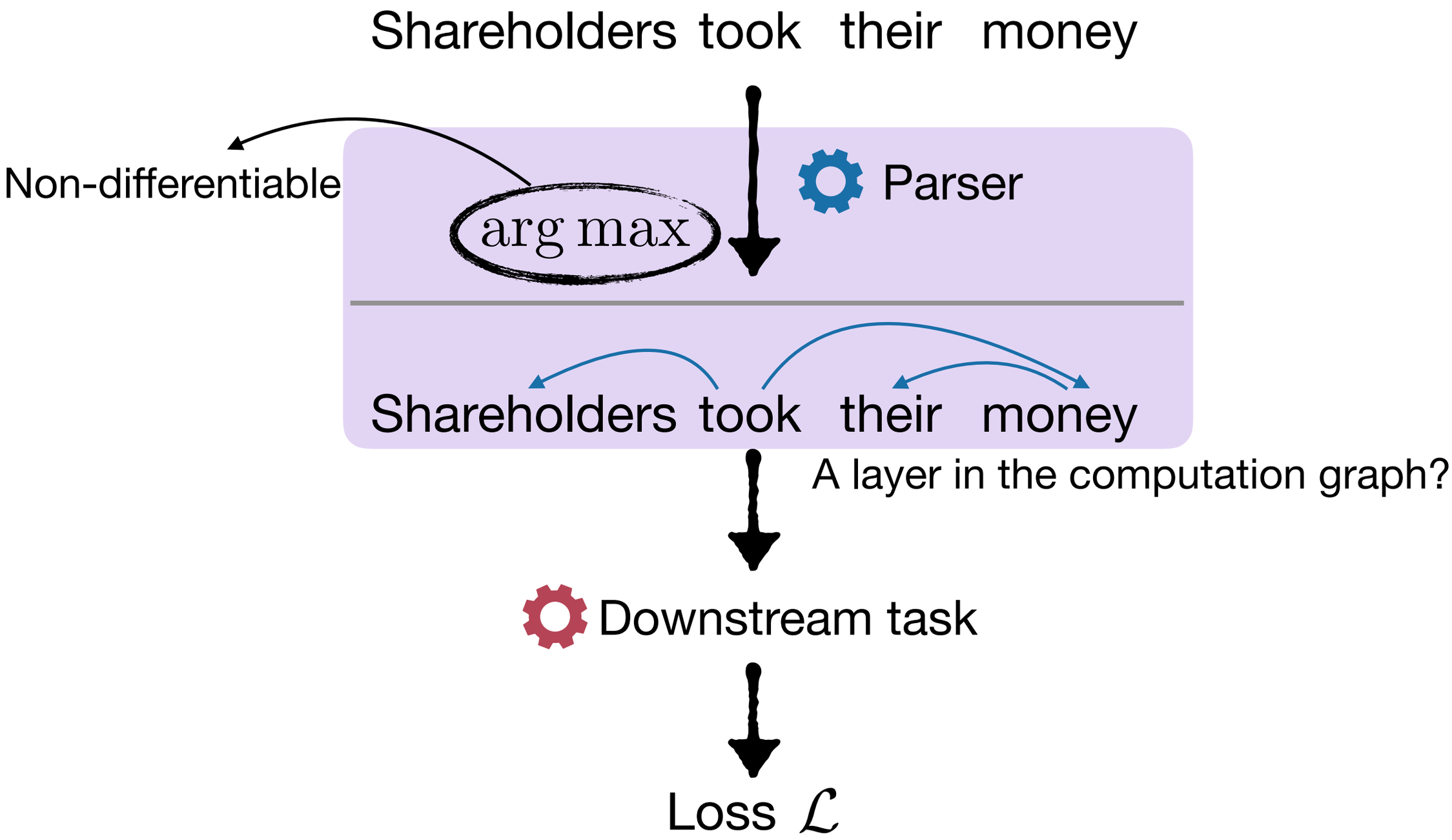
Overview


Shareholders took their money
arg max
Parser
Head token
Yang and Mitchell, 2017
Tree-RNN
Tai et al., 2015
Graph CNN
Kipf and Welling, 2017
...
Shareholders took their money
Downstream task
Loss $\mathcal{L}$

Overview


Shareholders took their money
arg max
Parser
Shareholders took their money
A layer in the computation graph?
Downstream task
Loss $\mathcal{L}$

Overview


Shareholders took their money
Non-differentiable
arg max
Parser
Shareholders took their money
A layer in the computation graph?
Downstream task
Loss $\mathcal{L}$

Overview

Aim

- Structured prediction as a layer.

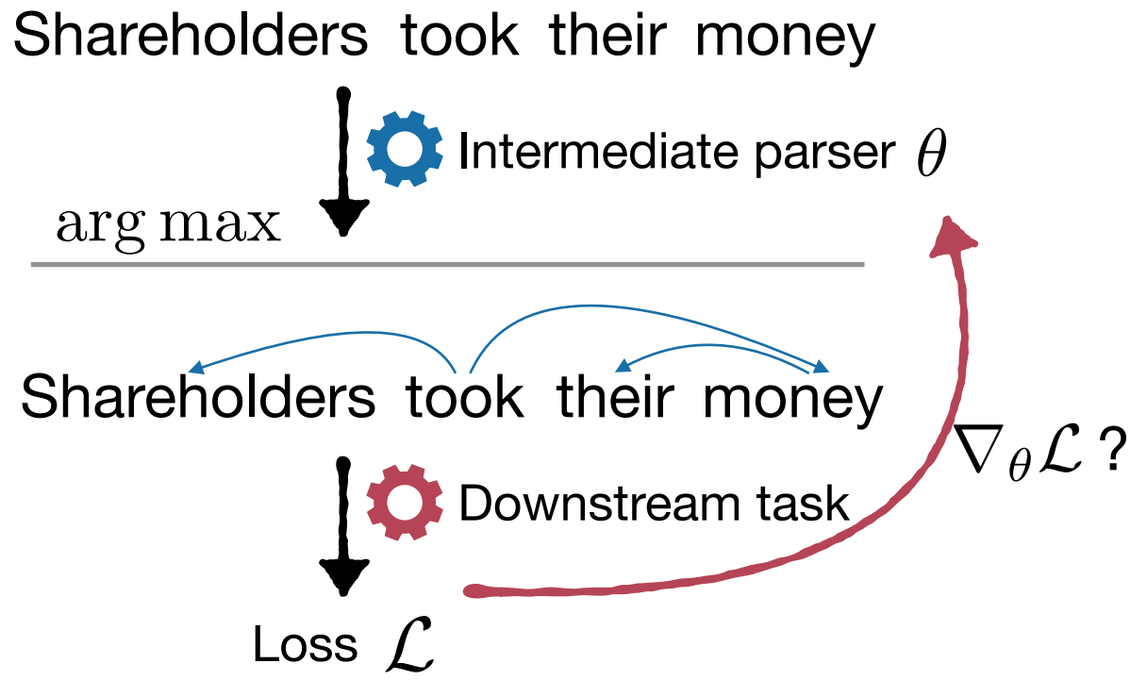
Motivation

- Structures help.

  Ji and Smith, 2017; Oepen et al., 2017

- Linguistic structures may not be universally optimal.

  Williams, 2017



Overview

Aim

- Structured prediction as a layer.

Motivation

- Structures help.

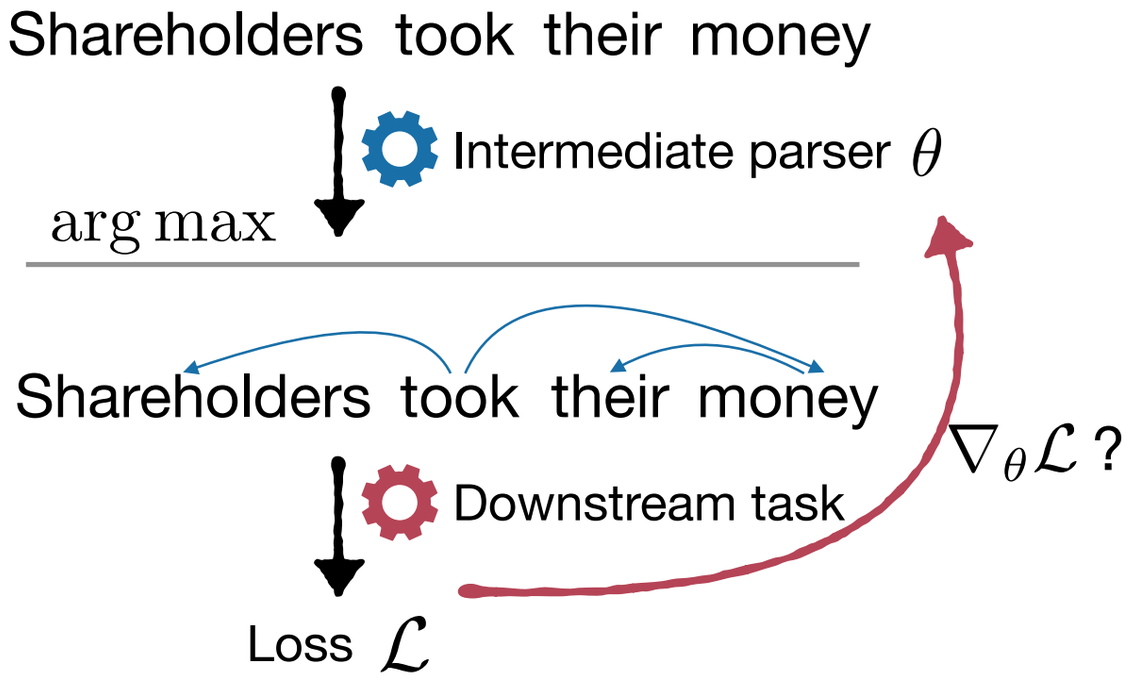
  Ji and Smith, 2017; Oepen et al., 2017

- Linguistic structures may not be universally optimal.

  Williams, 2017

Challenges

- argmax is non-differentiable.



Overview

Aim

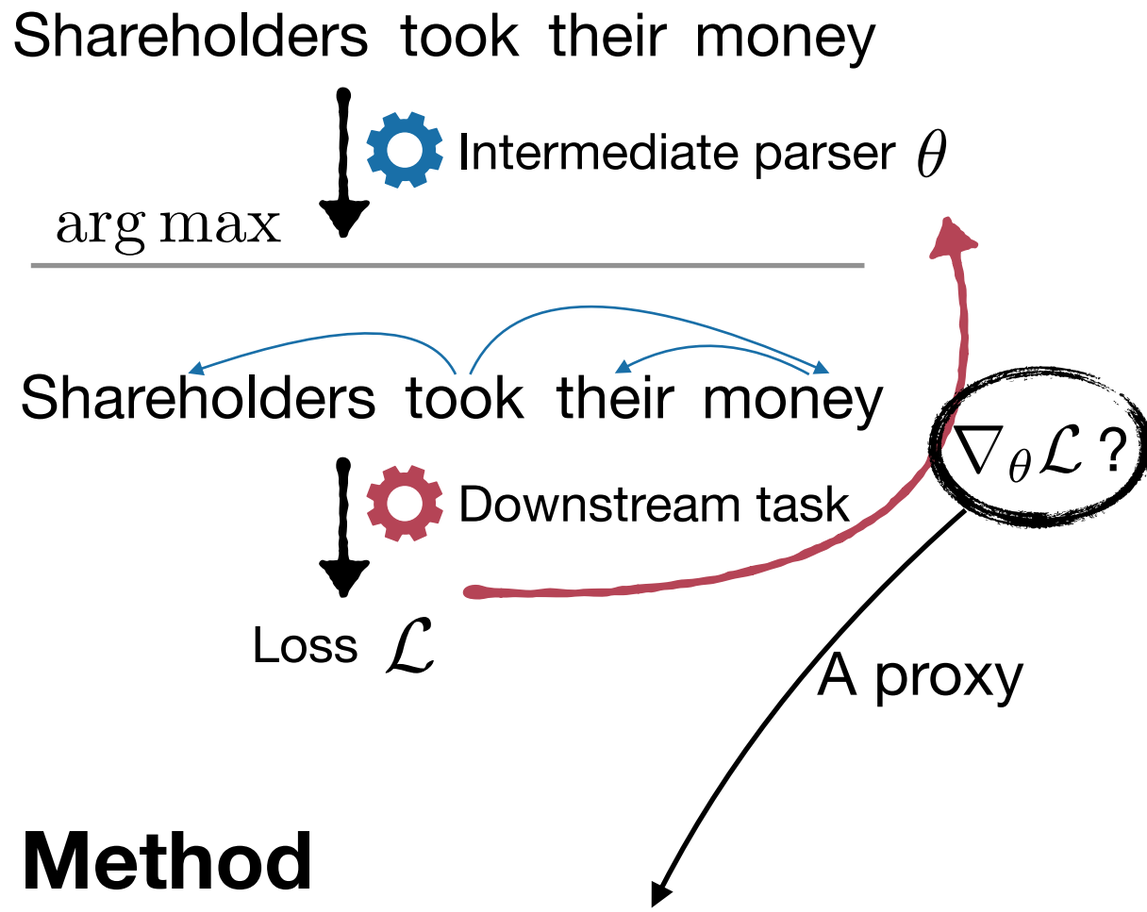
- Structured prediction as a layer.

Motivation

- Structures help.
  Ji and Smith, 2017; Oepen et al., 2017
- Linguistic structures may not be universally optimal.
  Williams, 2017

Challenges

- argmax is non-differentiable.



Method

Structured **P**rediction **I**ntermediate
Gradients **O**ptimization **T**echnique

SPIGOT

Outline

- **Background: structured prediction as linear programs**
- Method: SPIGOT algorithm
- Experiments

Structured Prediction Reviewed

Input

Shareholders took their money

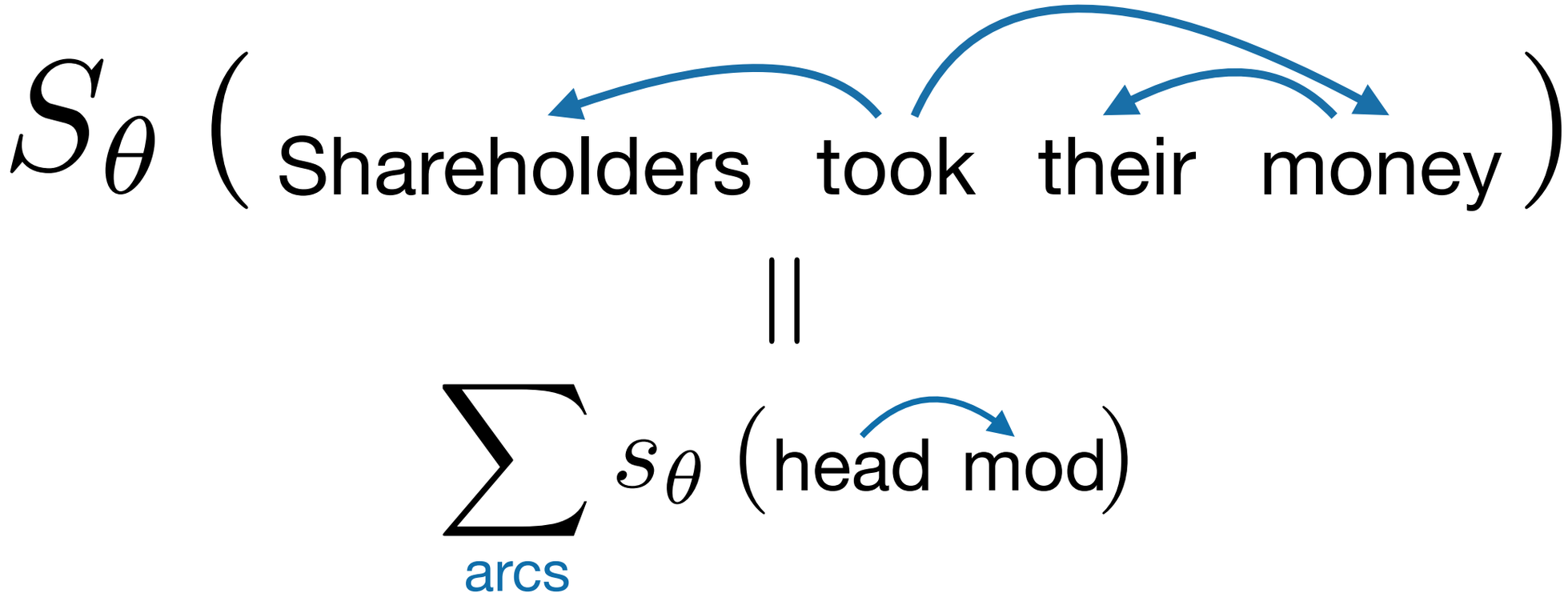
Output

Structured Prediction Reviewed

Input

Shareholders took their money

Score



Structured Prediction Reviewed

Input

Shareholders took their money

Score

$$\mathbf{s}_\theta = \left[s_\theta\left(\text{their} \leftarrow \text{money}\right), s_\theta\left(\text{took} \leftarrow \text{their}\right), s_\theta\left(\text{took} \rightarrow \text{money}\right), \ldots, s_\theta\left(\text{their} \rightarrow \text{took}\right) \right]^\top$$

$$\mathbf{z} = [\quad 1?, \quad 0?, \quad 1?, \quad \ldots, \quad 0? \quad]^\top$$

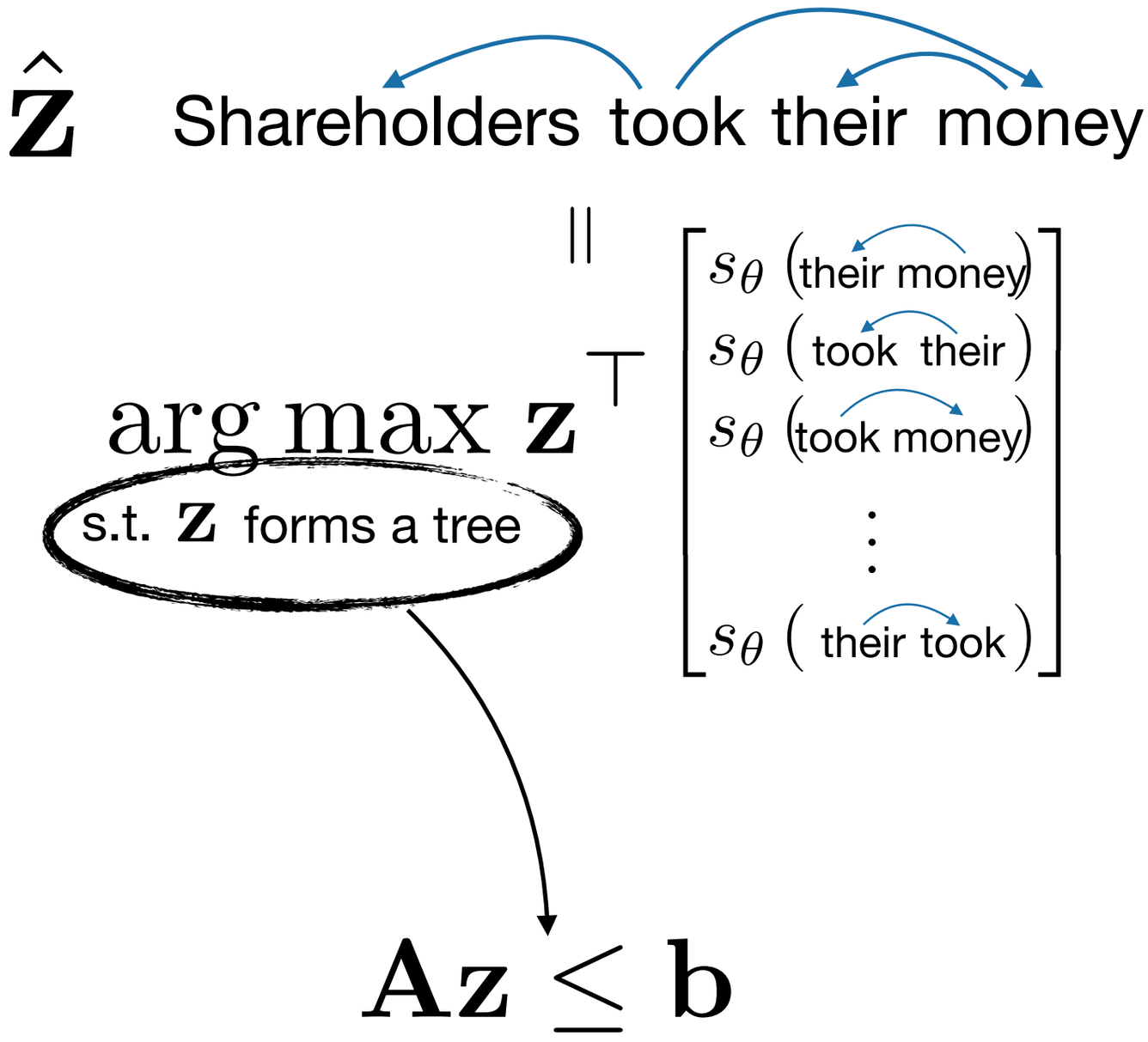
Output

$$\arg\max \ \mathbf{z}^\top \mathbf{s}_\theta$$

s.t. $\mathbf{z}$ forms a tree

$\hat{\mathbf{z}}$

Linear Programming Formulation



Roth and Yih, 2004; Martins et al., 2009

Linear Programming Formulation

$\hat{\mathbf{z}}$ Shareholders took their money

$$= \arg\max \mathbf{z}^\top \begin{bmatrix} s_\theta(\text{their money}) \\ s_\theta(\text{took their}) \\ s_\theta(\text{took money}) \\ \vdots \\ s_\theta(\text{their took}) \end{bmatrix}$$

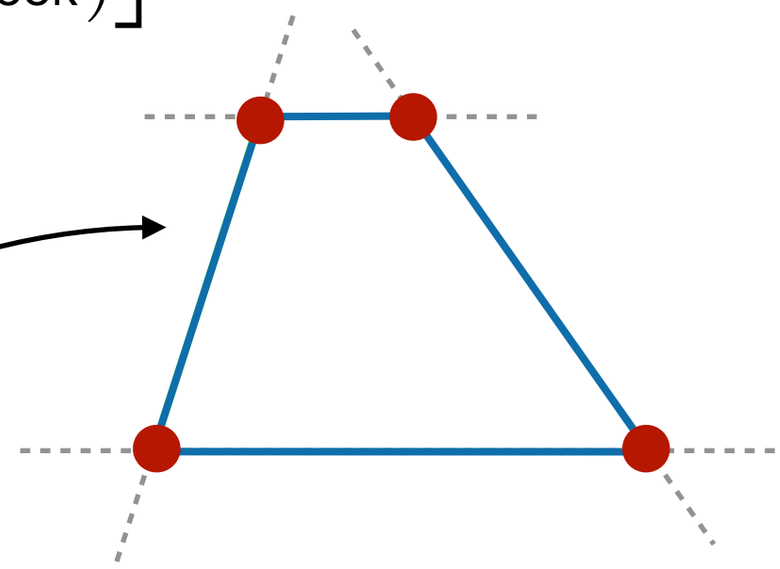
s.t. $\mathbf{z}$ forms a tree

$z_i \in \{0, 1\}$

relaxation

$z_i \in [0, 1]$

$$\mathbf{Az} \leq \mathbf{b}$$



Roth and Yih, 2004; Martins et al., 2009

Outline

- Background: structured prediction as linear programs
- **Method: SPIGOT algorithm**
- Experiments

Backprop

$$\arg\max \mathbf{z}^\top \begin{bmatrix} s_\theta(\text{their money}) \\ s_\theta(\text{took their}) \\ s_\theta(\text{took money}) \\ \vdots \\ s_\theta(\text{their took}) \end{bmatrix}$$

s.t. $\mathbf{Z}$ forms a tree

$\nabla_\theta \mathcal{L}$

$\hat{\mathbf{z}}$ Shareholders took their money

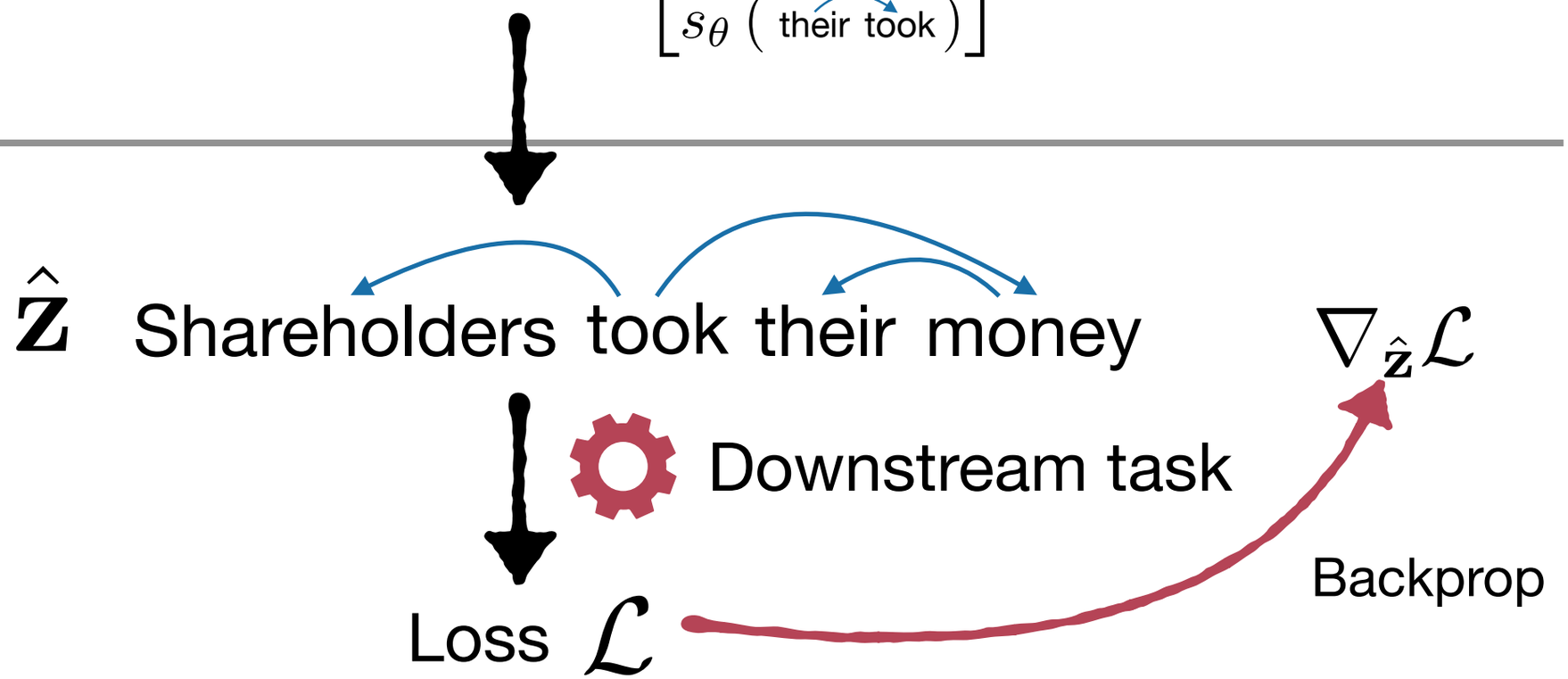
Downstream task

Loss $\mathcal{L}$

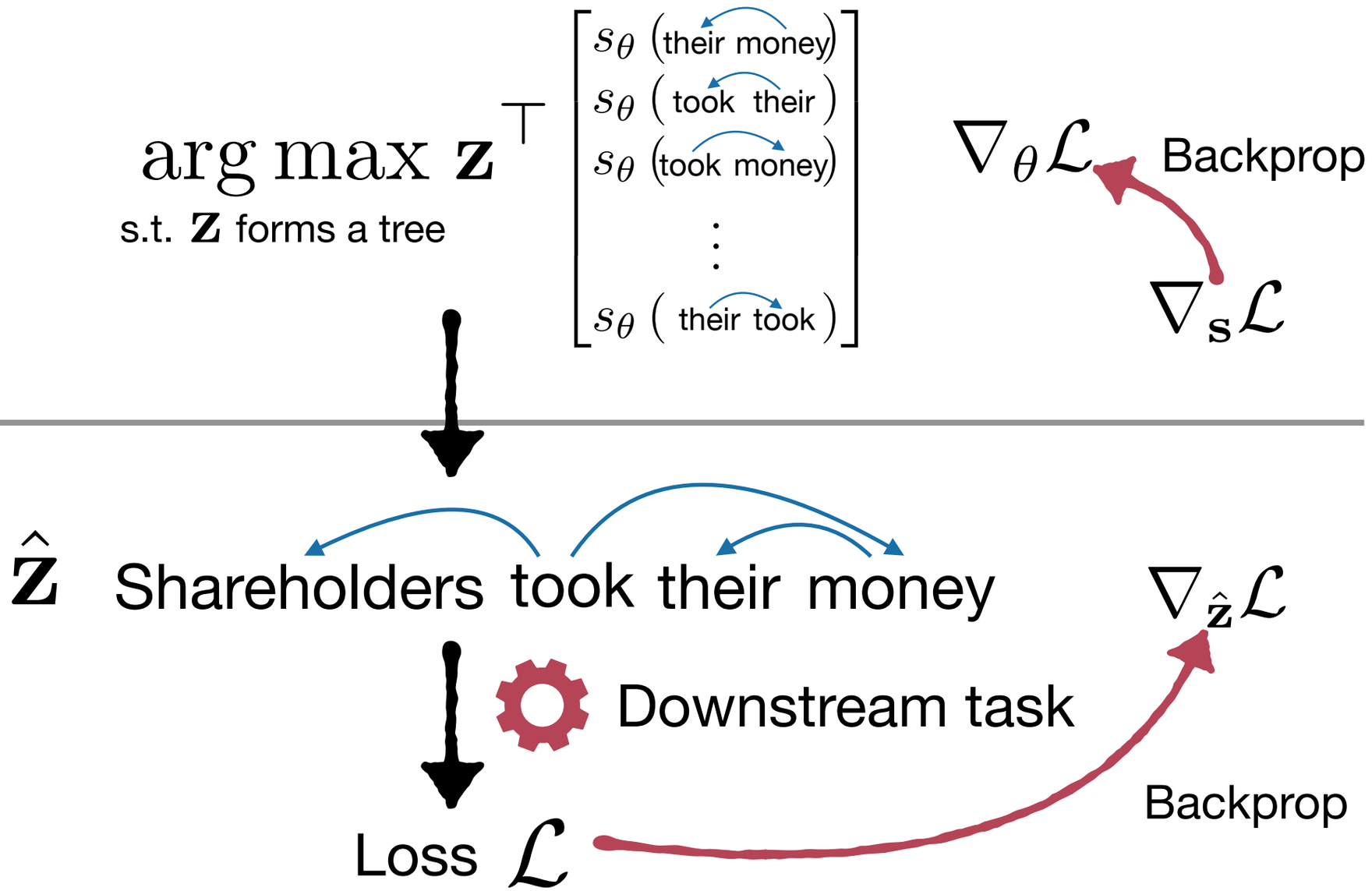
Backprop

$$\arg\max \mathbf{z}^\top \begin{bmatrix} s_\theta(\text{their money}) \\ s_\theta(\text{took their}) \\ s_\theta(\text{took money}) \\ \vdots \\ s_\theta(\text{their took}) \end{bmatrix} \qquad \nabla_\theta \mathcal{L}$$

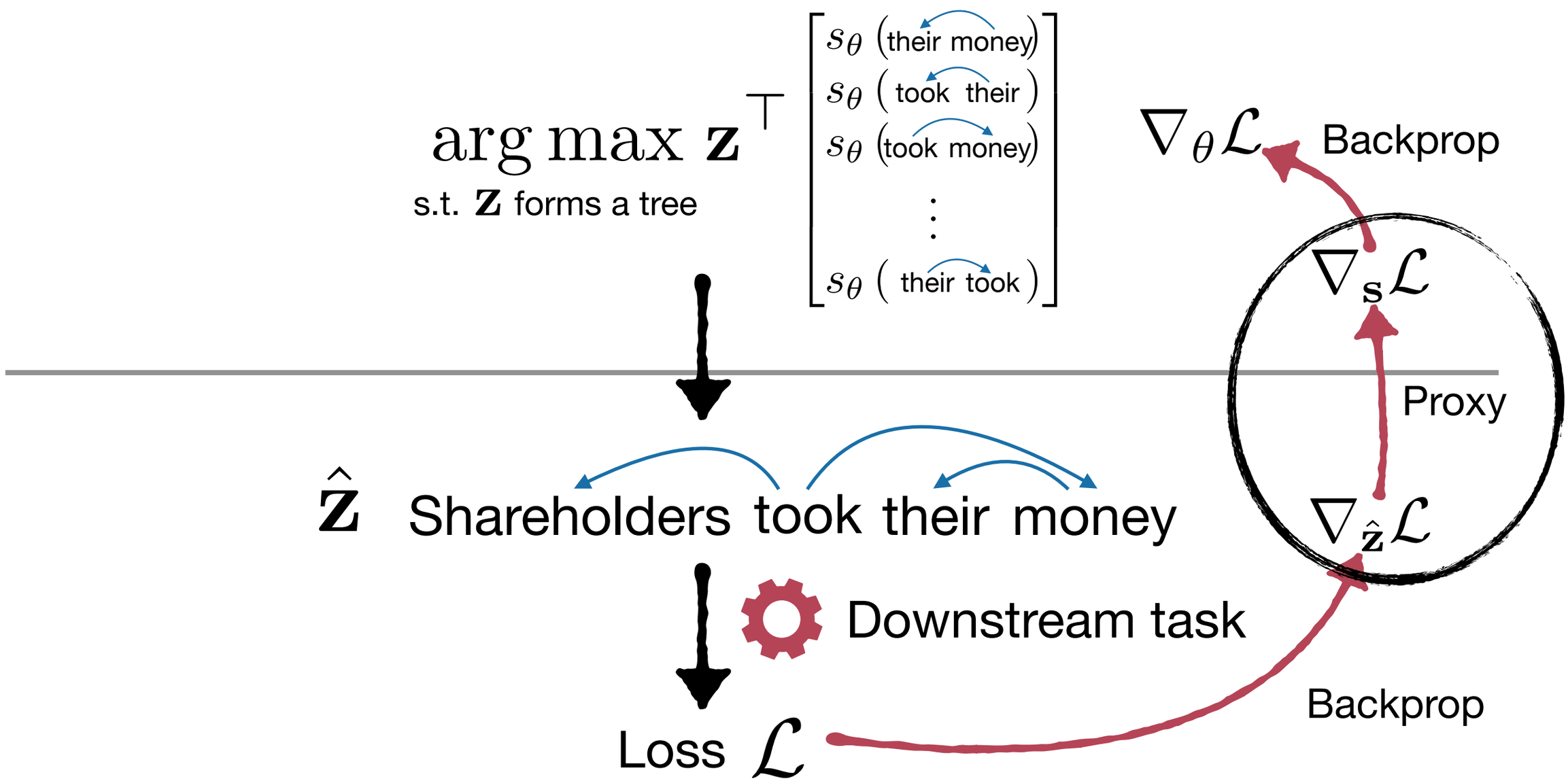
s.t. $\mathbf{z}$ forms a tree



Backprop


$\arg\max \mathbf{z}^\top$
s.t. $\mathbf{Z}$ forms a tree
s_θ (their money)
s_θ (took their)
s_θ (took money)
s_θ (their took)
$\nabla_\theta \mathcal{L}$
Backprop
$\nabla_{\mathbf{s}} \mathcal{L}$
$\hat{\mathbf{z}}$
Shareholders took their money
$\nabla_{\hat{\mathbf{z}}} \mathcal{L}$
Downstream task
Backprop
Loss $\mathcal{L}$

Backprop


$\arg\max \mathbf{z}^\top$
s.t. $\mathbf{Z}$ forms a tree
s_θ (their money)
s_θ (took their)
s_θ (took money)
s_θ (their took)
$\nabla_\theta \mathcal{L}$
Backprop
$\nabla_{\mathbf{s}} \mathcal{L}$
Proxy
$\nabla_{\hat{\mathbf{z}}} \mathcal{L}$
$\hat{\mathbf{z}}$
Shareholders took their money
Downstream task
Loss $\mathcal{L}$
Backprop

Backprop

We have: $\nabla_{\hat{\mathbf{z}}} \mathcal{L}$ We need: $\nabla_{\mathbf{s}} \mathcal{L}$

Backprop

We have: $\nabla_{\hat{\mathbf{z}}}\mathcal{L}$ **We need:** $\nabla_{\mathbf{s}}\mathcal{L}$

Leibniz, 1676

$$\nabla_{\mathbf{s}}\mathcal{L} = \text{“}\mathbf{J}\text{”}\nabla_{\hat{\mathbf{z}}}\mathcal{L}$$

Backprop

We have: $\nabla_{\hat{\mathbf{z}}}\mathcal{L}$ **We need:** $\nabla_{\mathbf{s}}\mathcal{L}$

Leibniz, 1676

$$\nabla_{\mathbf{s}}\mathcal{L} = \text{“}\mathbf{J}\text{”}\nabla_{\hat{\mathbf{z}}}\mathcal{L}$$ ✗

$$\hat{\mathbf{z}} = \arg\max_{\mathbf{Z}} \mathbf{Z}^\top \mathbf{s}_\theta$$

s.t. $\mathbf{Z}$ forms a tree

Jacobian not defined

Backprop

We have: $\nabla_{\hat{\mathbf{z}}}\mathcal{L}$ **We need:** $\nabla_{\mathbf{s}}\mathcal{L}$

Leibniz, 1676

$$\nabla_{\mathbf{s}}\mathcal{L} = \text{“}\mathbf{J}\text{”}\nabla_{\hat{\mathbf{z}}}\mathcal{L}$$ 

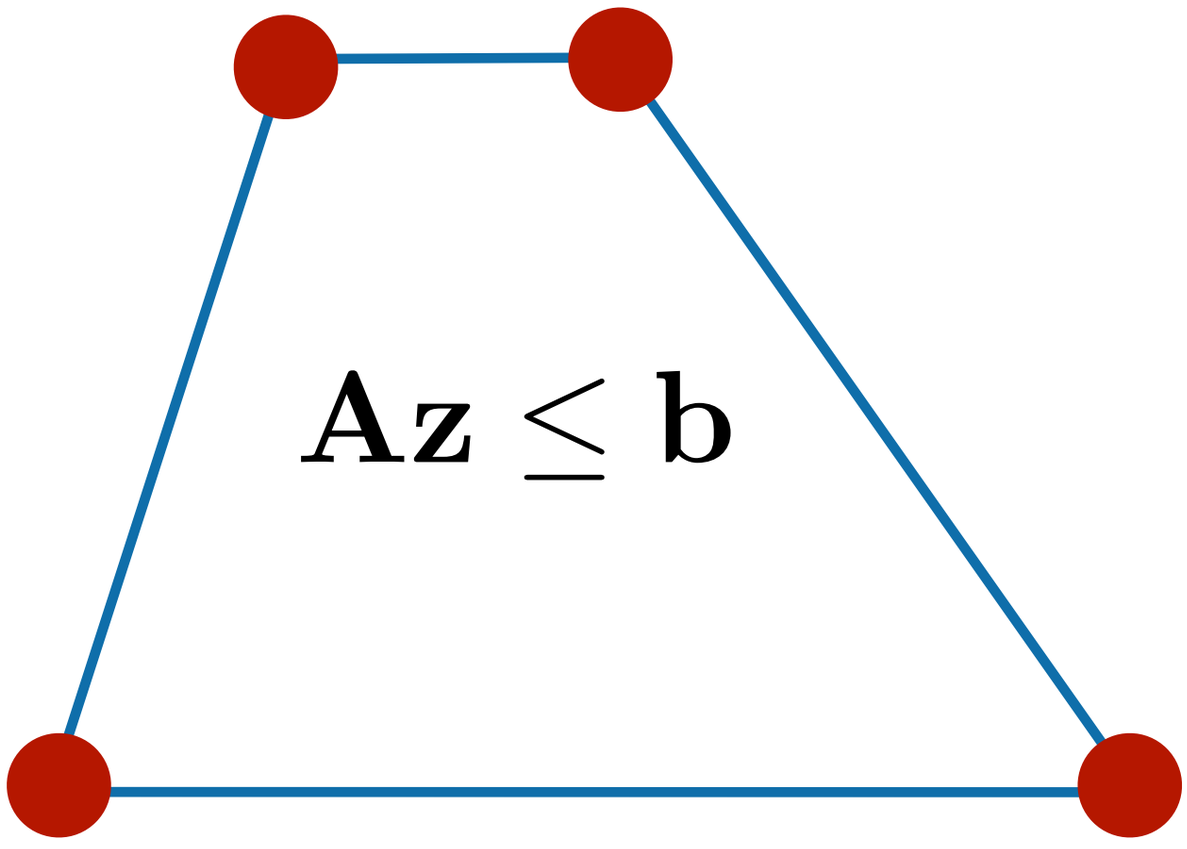
Straight-through Estimator (STE)

Hinton, 2012; Bengio et al., 2013

$$\nabla_{\mathbf{s}}\mathcal{L} \triangleq \nabla_{\hat{\mathbf{z}}}\mathcal{L}$$

Some Geometry…

Straight-through Estimator (STE): $\nabla_{\mathbf{s}}\mathcal{L} \triangleq \nabla_{\hat{\mathbf{z}}}\mathcal{L}$

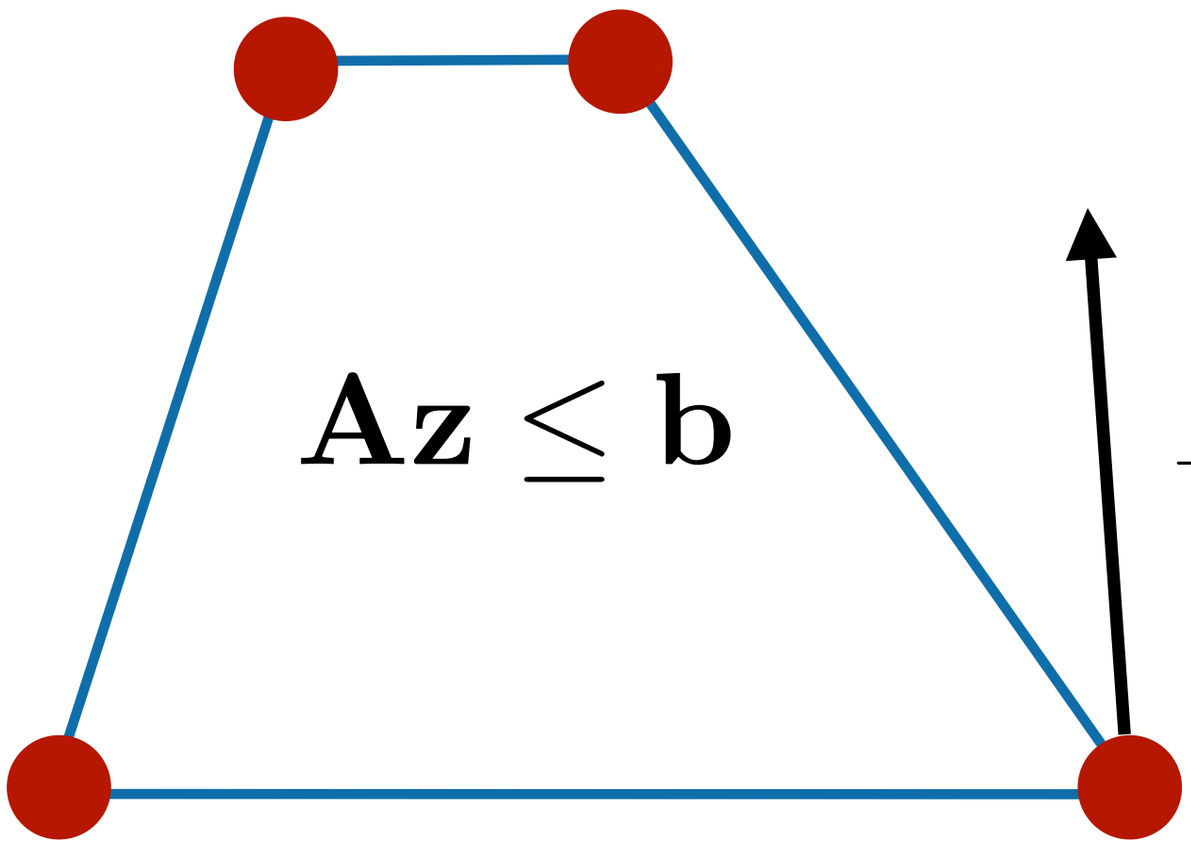


$\hat{\mathbf{z}} = [1, 0, 1, \cdots, 0]^\top$

Shareholders took their money

Some Geometry…

Straight-through Estimator (STE): $\nabla_{\mathbf{s}}\mathcal{L} \triangleq \nabla_{\hat{\mathbf{z}}}\mathcal{L}$



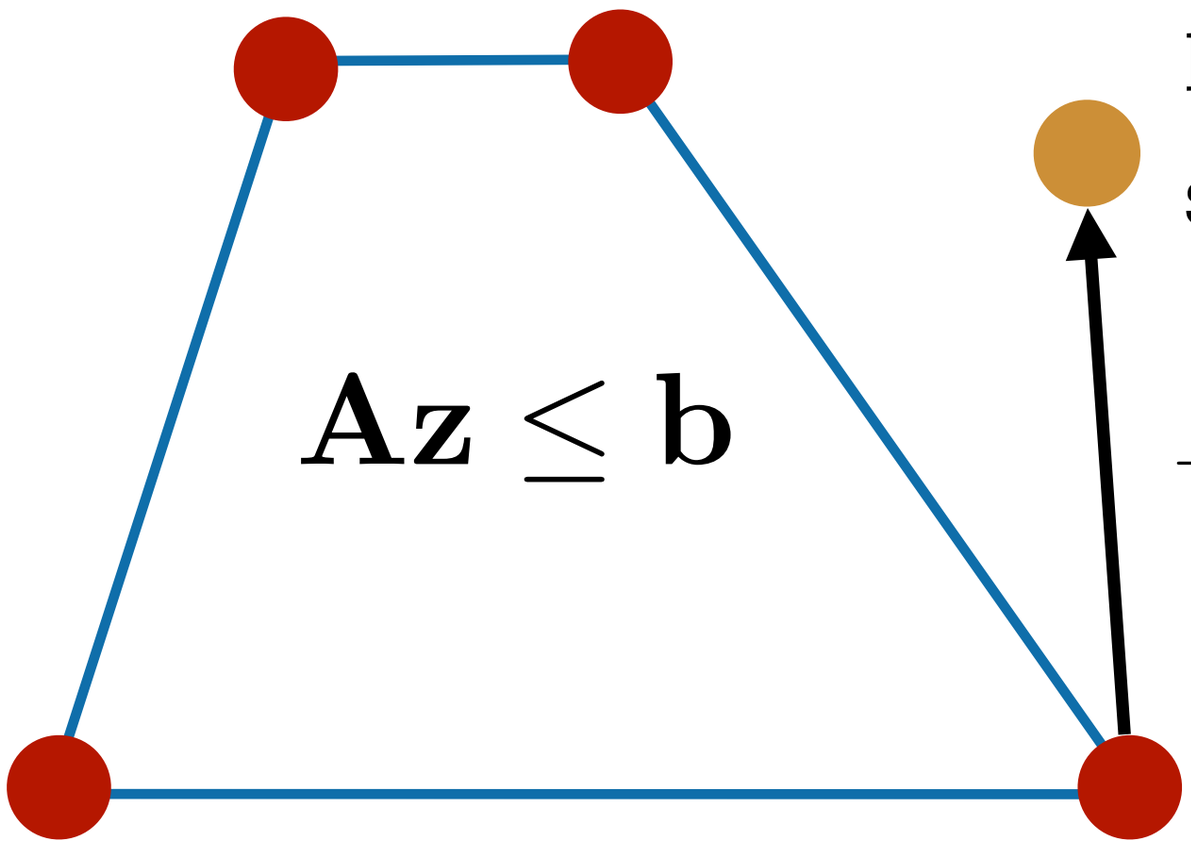
$-\nabla_{\hat{\mathbf{z}}}\mathcal{L} = [-0.3, 0.5, 0.4, \ldots, 0.2]$

$\hat{\mathbf{z}} = [1, 0, 1, \cdots, 0]^{\top}$

Shareholders took their money

Some Geometry…

Straight-through Estimator (STE): $\nabla_{\mathbf{s}}\mathcal{L} \triangleq \nabla_{\hat{\mathbf{z}}}\mathcal{L}$



$$\mathbf{p} = \hat{\mathbf{z}} - \nabla_{\hat{\mathbf{z}}}\mathcal{L}$$

Shareholders took their money

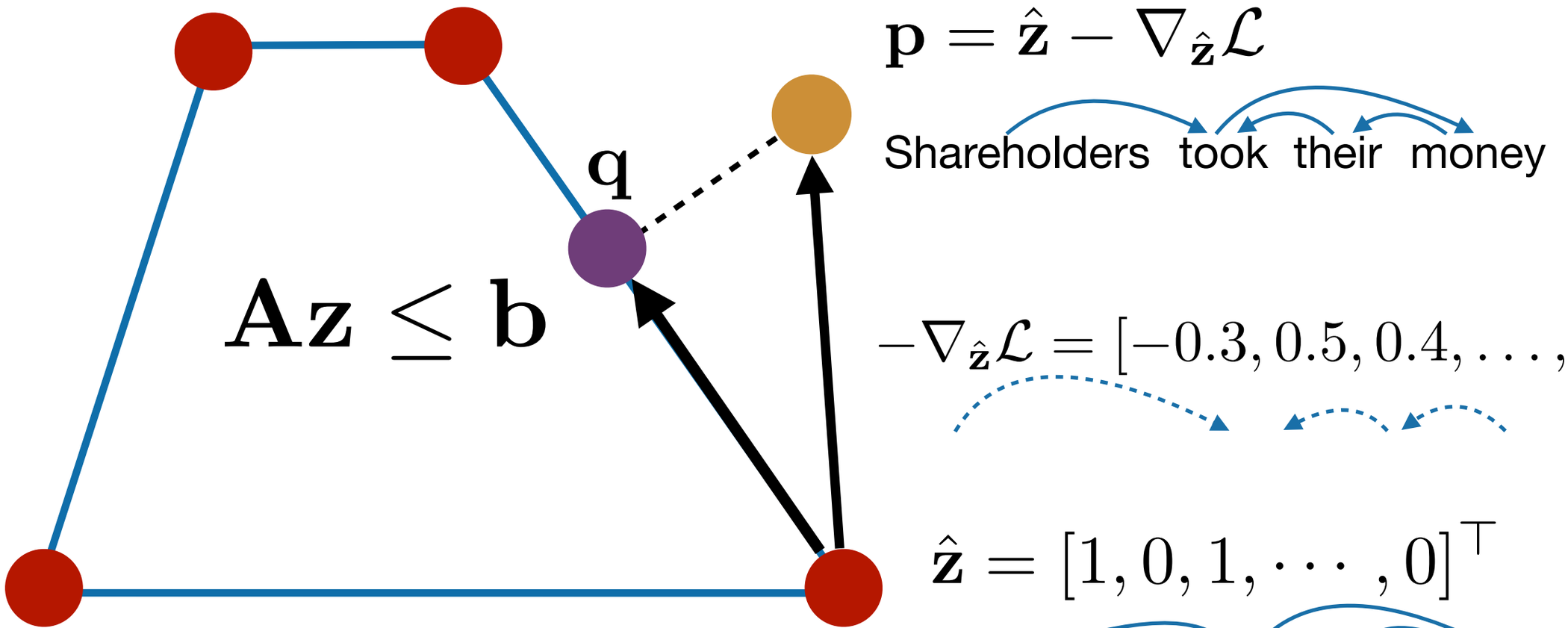
$$-\nabla_{\hat{\mathbf{z}}}\mathcal{L} = [-0.3, 0.5, 0.4, \ldots, 0.2]$$

$$\hat{\mathbf{z}} = [1, 0, 1, \cdots, 0]^{\top}$$

Shareholders took their money

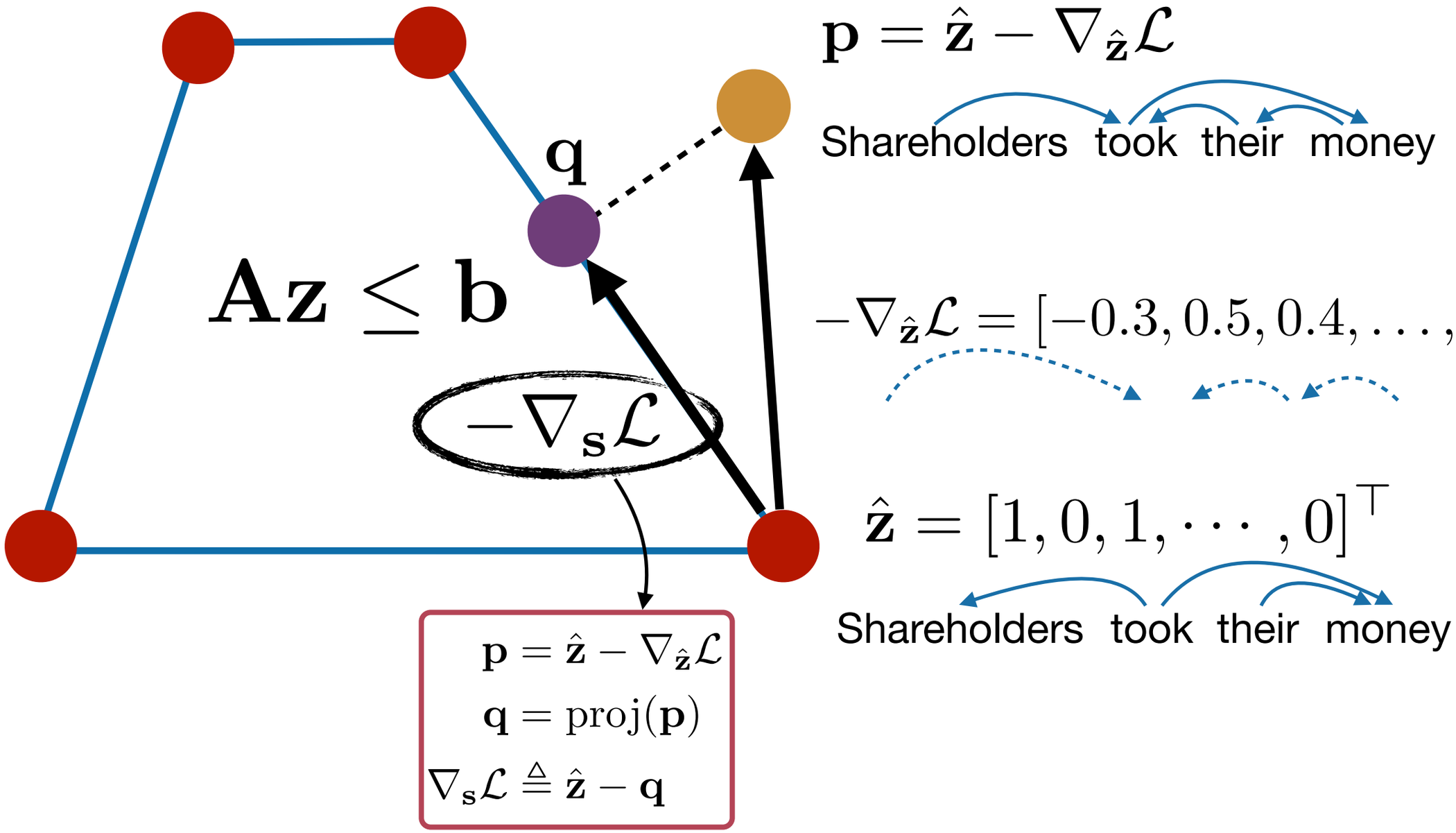
Some Geometry…

SPIGOT



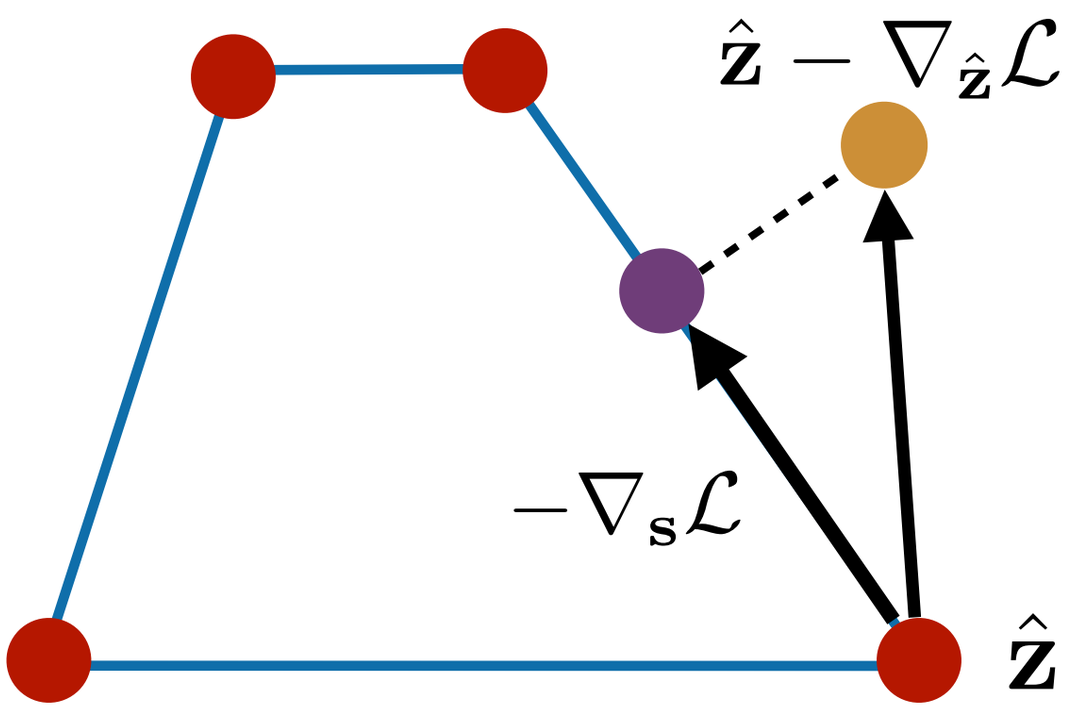
Some Geometry…

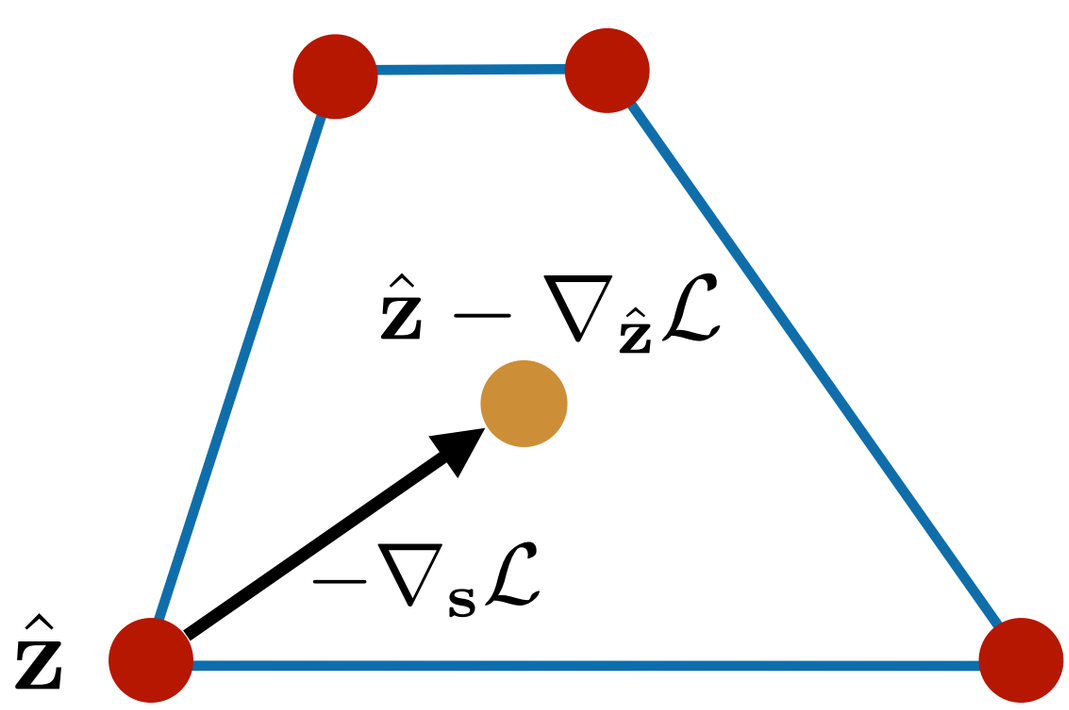
SPIGOT



Some Geometry…

SPIGOT





Algorithm

Input

Shareholders took their money

Parser θ

$$\arg\max_{\mathbf{z}} \mathbf{z}^\top \begin{bmatrix} s_\theta(\text{their money}) \\ s_\theta(\text{took their}) \\ s_\theta(\text{took money}) \\ \vdots \\ s_\theta(\text{their took}) \end{bmatrix} \quad \text{s.t. } \mathbf{z} \text{ forms a tree}$$

$\hat{\mathbf{z}}$

Algorithm

Input

Shareholders took their money

Parser θ

$$\arg\max_{\mathbf{z}} \mathbf{z}^\top \begin{bmatrix} s_\theta(\text{their money}) \\ s_\theta(\text{took their}) \\ s_\theta(\text{took money}) \\ \vdots \\ s_\theta(\text{their took}) \end{bmatrix} \quad \text{s.t. } \mathbf{z} \text{ forms a tree}$$

$\hat{\mathbf{z}}$

Downstream task ϕ

Loss $\mathcal{L}$

Algorithm

Input

Shareholders took their money

Parser θ

$$\arg\max_{\text{s.t. } \mathbf{Z} \text{ forms a tree}} \mathbf{z}^\top \begin{bmatrix} s_\theta(\text{their money}) \\ s_\theta(\text{took their}) \\ s_\theta(\text{took money}) \\ \vdots \\ s_\theta(\text{their took}) \end{bmatrix}$$

$\hat{\mathbf{z}}$ Shareholders took their money $\nabla_{\hat{\mathbf{z}}} \mathcal{L}$

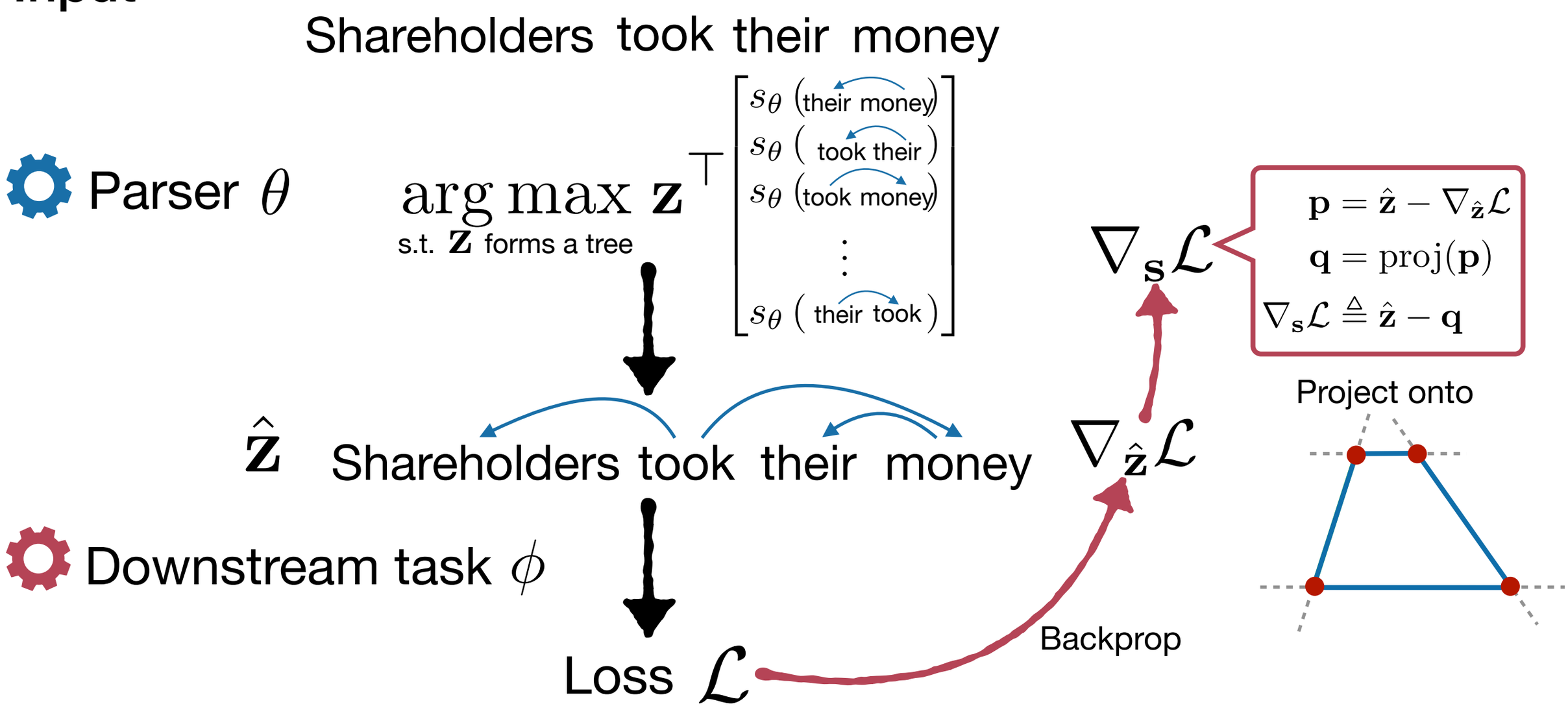
Downstream task ϕ

Loss $\mathcal{L}$

Backprop

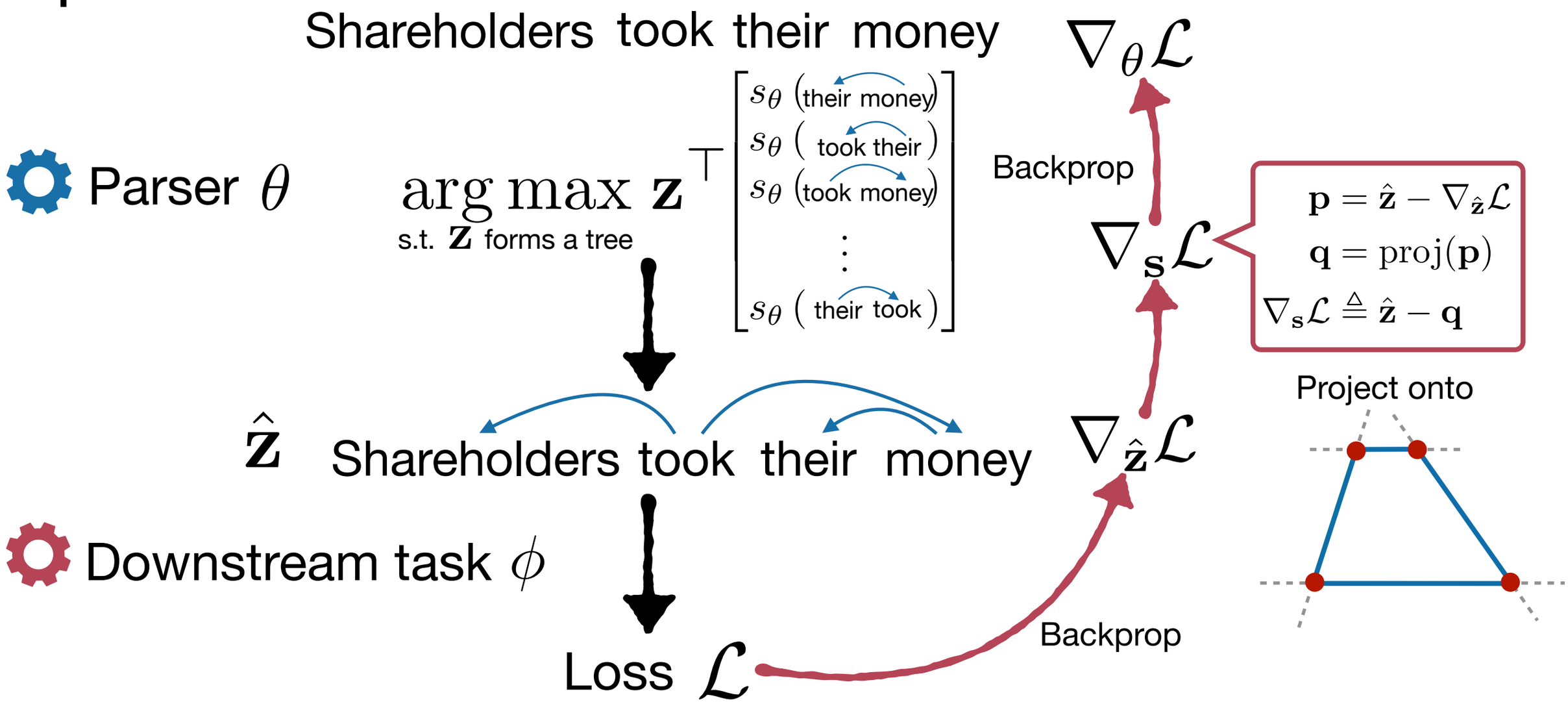
Algorithm

Input



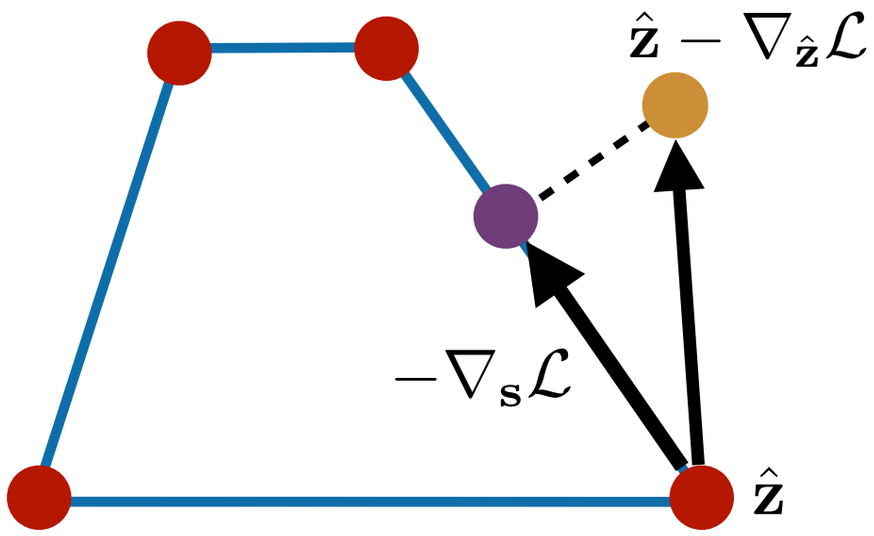
Algorithm

Input

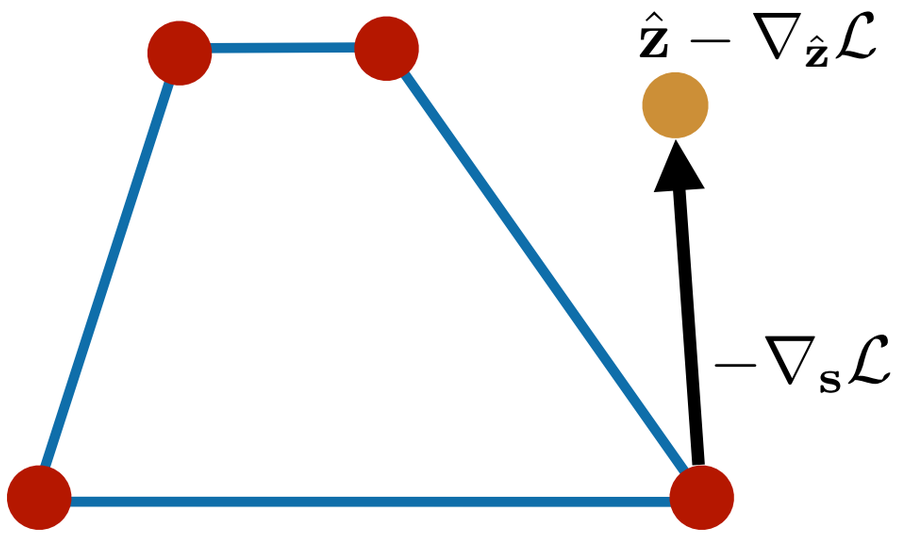


Connections to Related Work

SPIGOT



STE

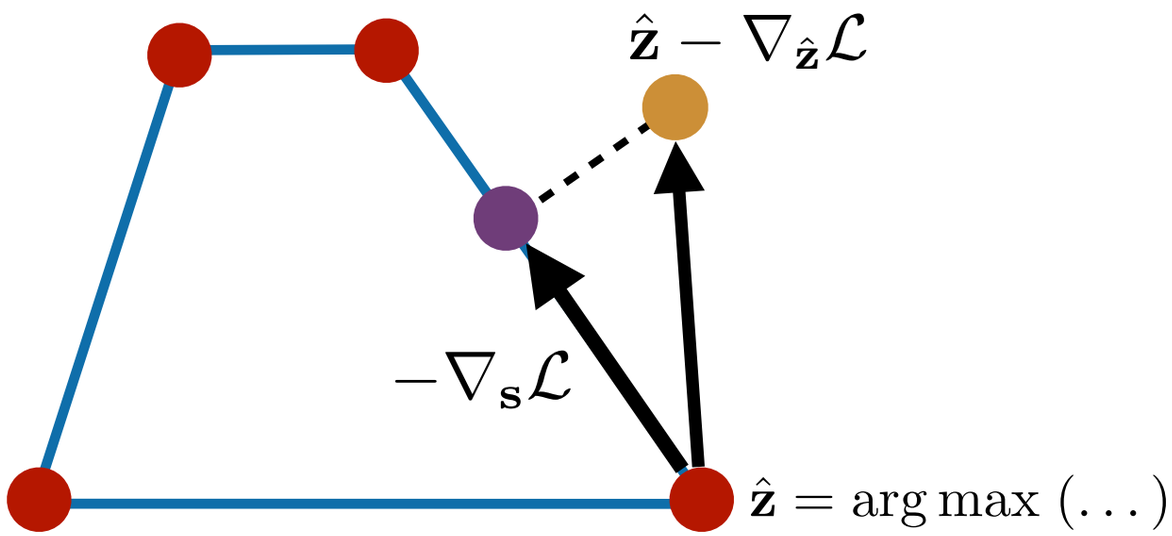


	Pipeline	STE	Structured Att.	SPIGOT
Hard decision on $\hat{\mathbf{z}}$	✓	✓		✓
Backprop		✓	✓	✓
Marginal			✓	
Projection				✓

Structured Attention: Kim et al., 2017

Connections to Related Work

SPIGOT



Structured Attention

$$\hat{\mathbf{z}} = \mathrm{softmax}(\dots)$$

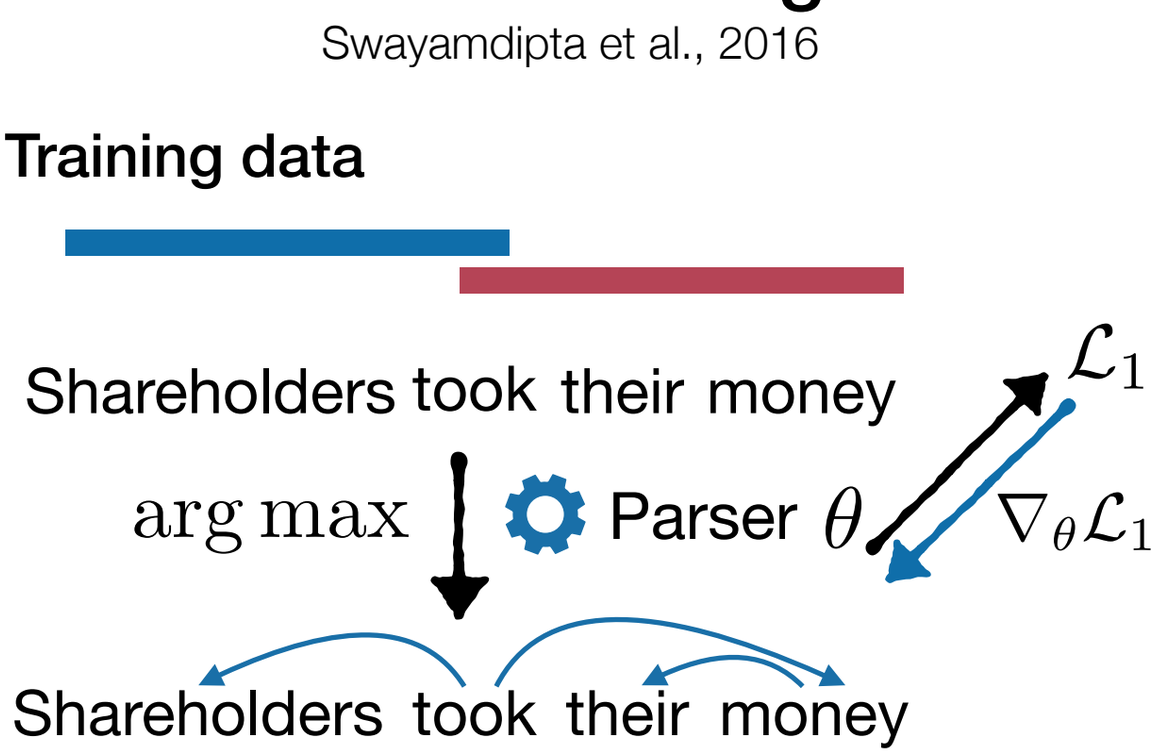
	Pipeline	STE	Structured Att.	SPIGOT
Hard decision on $\hat{\mathbf{z}}$	✓	✓		✓
Backprop		✓	✓	✓
Marginal			✓	
Projection				✓

Structured Attention: Kim et al., 2017

Applications

Joint learning

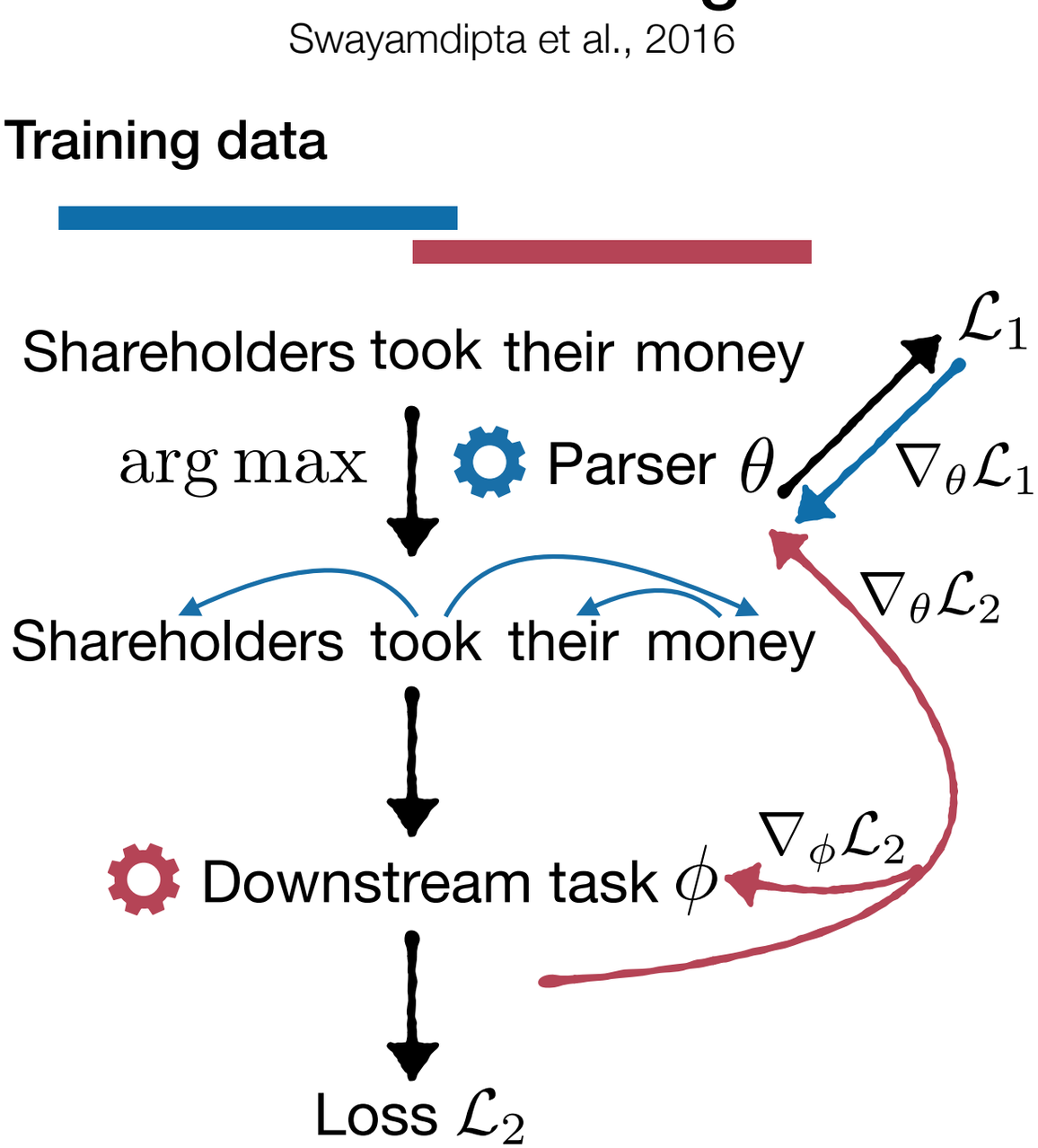
Swayamdipta et al., 2016



Applications

Joint learning

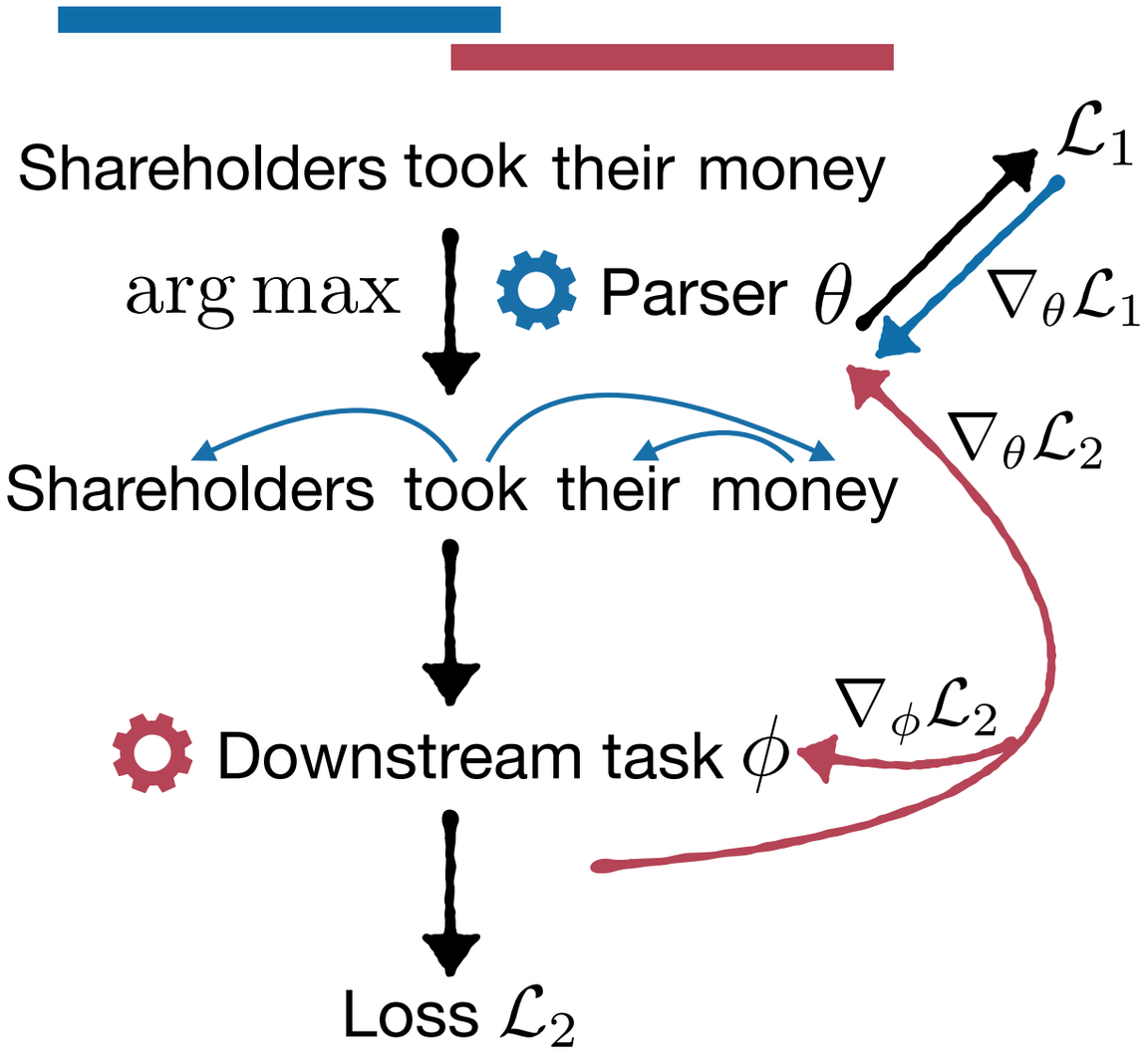
Swayamdipta et al., 2016



Applications

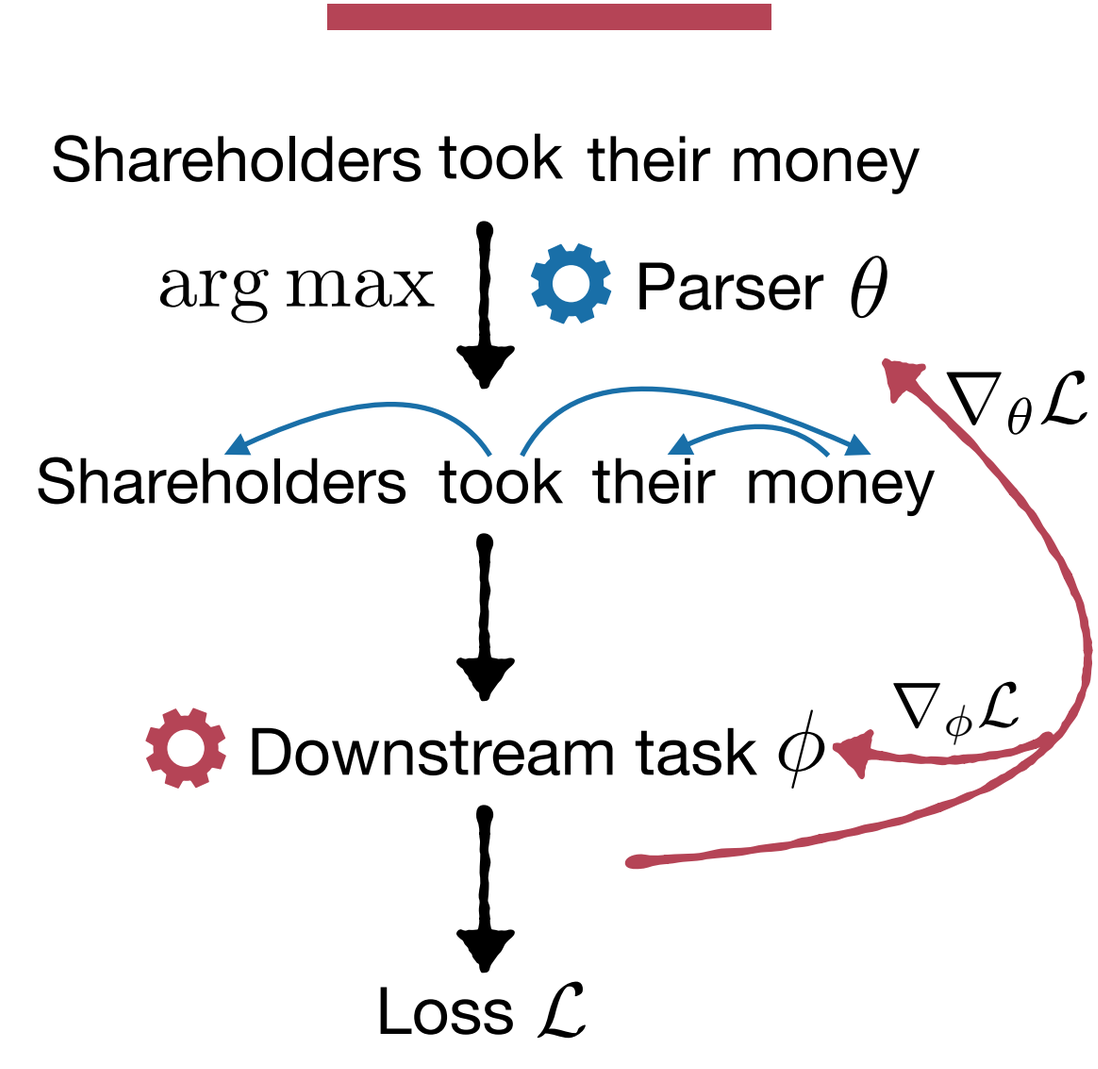
Joint learning

Swayamdipta et al., 2016



Induce latent structures

Yogatama et al., 2017; Williams et al., 2017



Outline

- Background: structured prediction as linear programs
- Method: SPIGOT algorithm
- **Experiments**

Experiments: Syntactic-then-semantic Parsing

Input

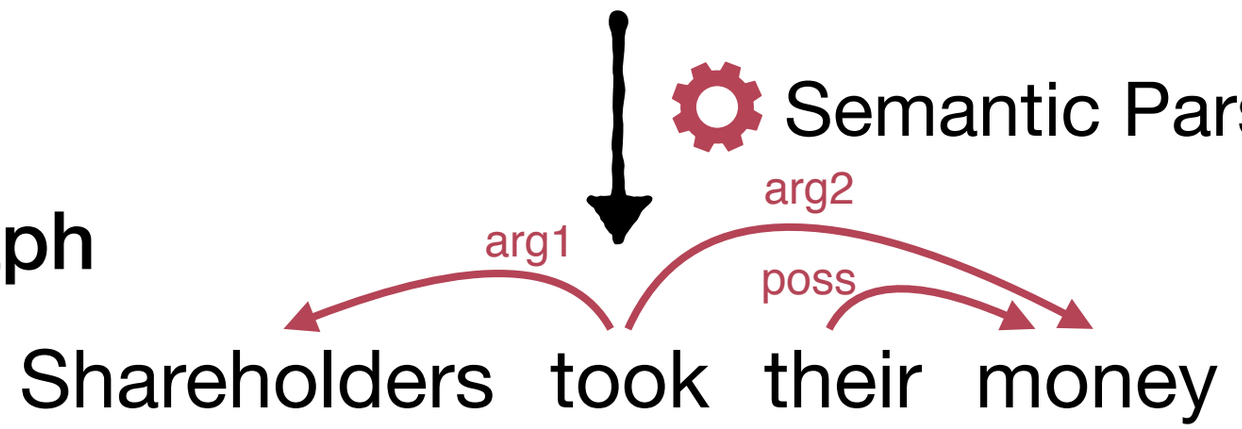
Shareholders took their money

$\arg\max$ ⚙ Syntactic Parser θ

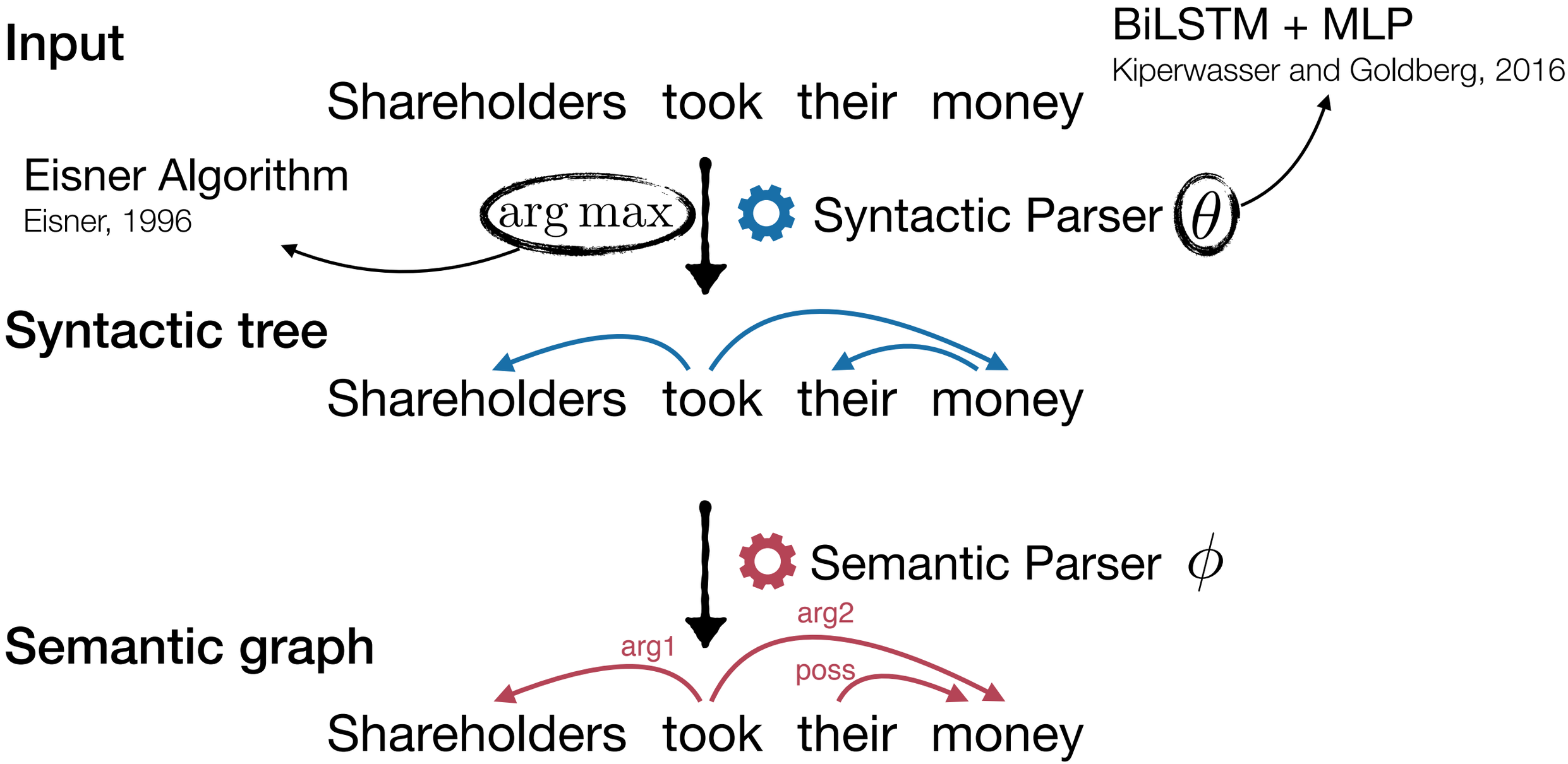
Syntactic tree

⚙ Semantic Parser ϕ

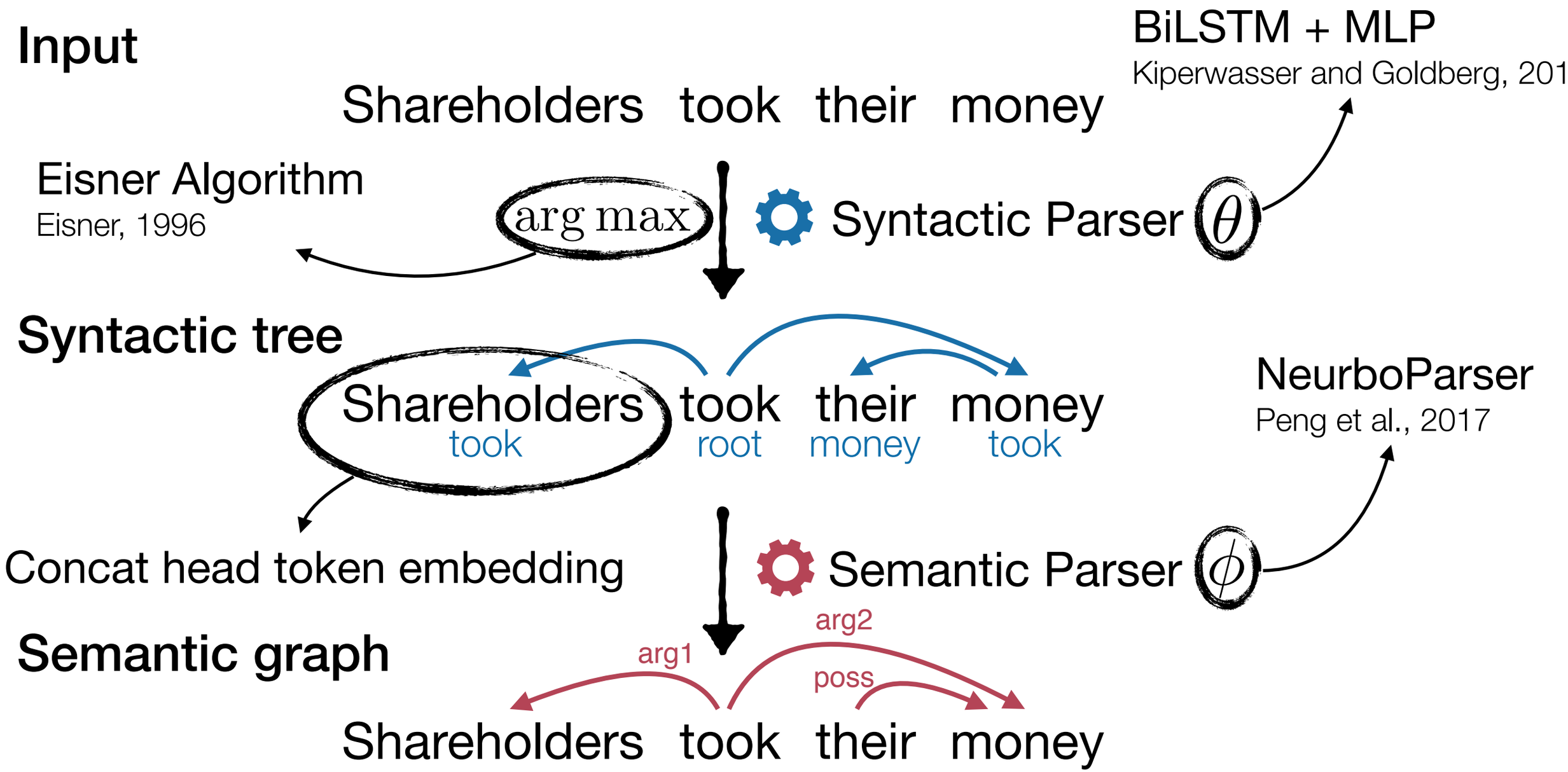
Semantic graph



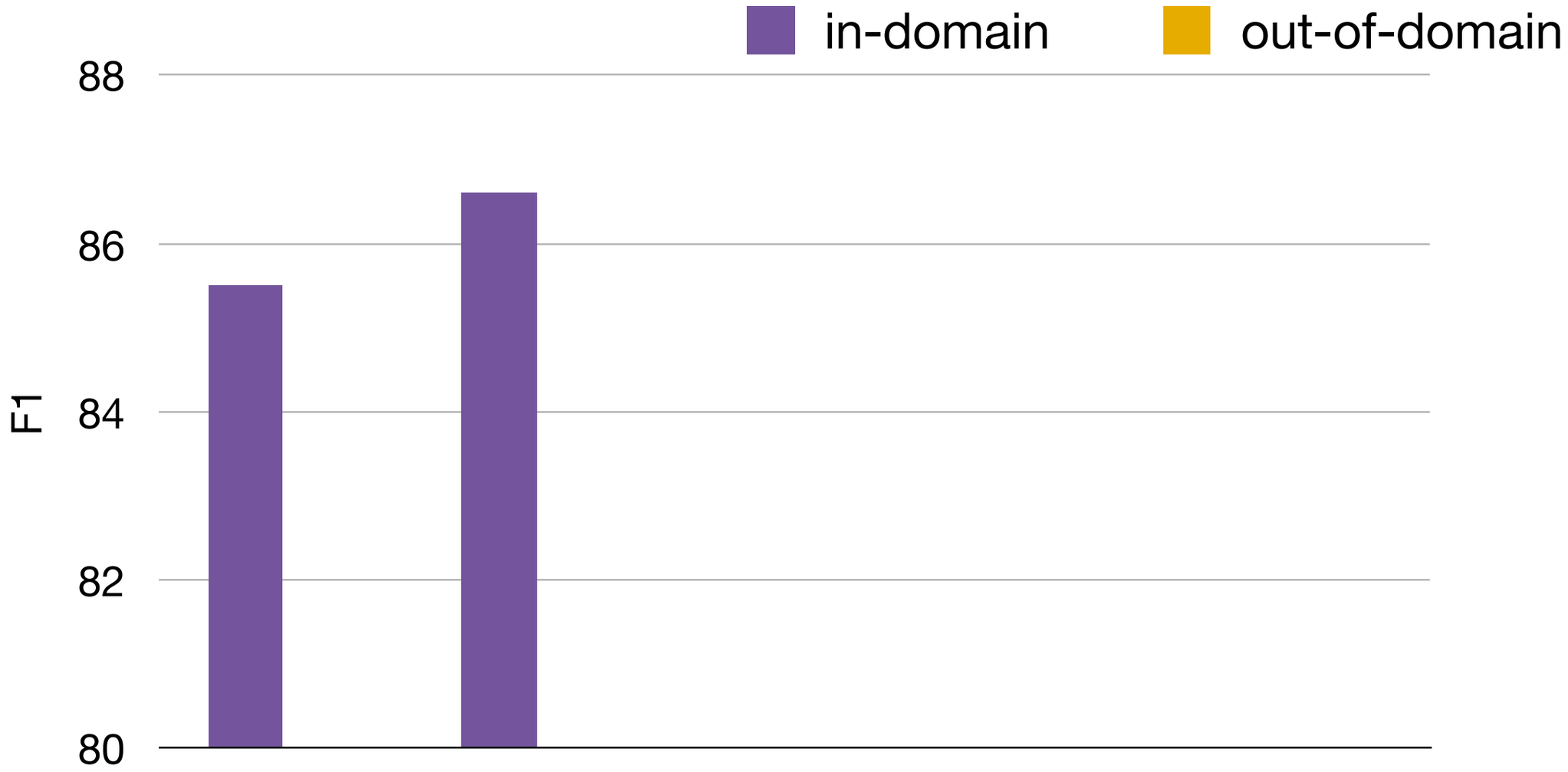
Experiments: Syntactic-then-semantic Parsing



Experiments: Syntactic-then-semantic Parsing



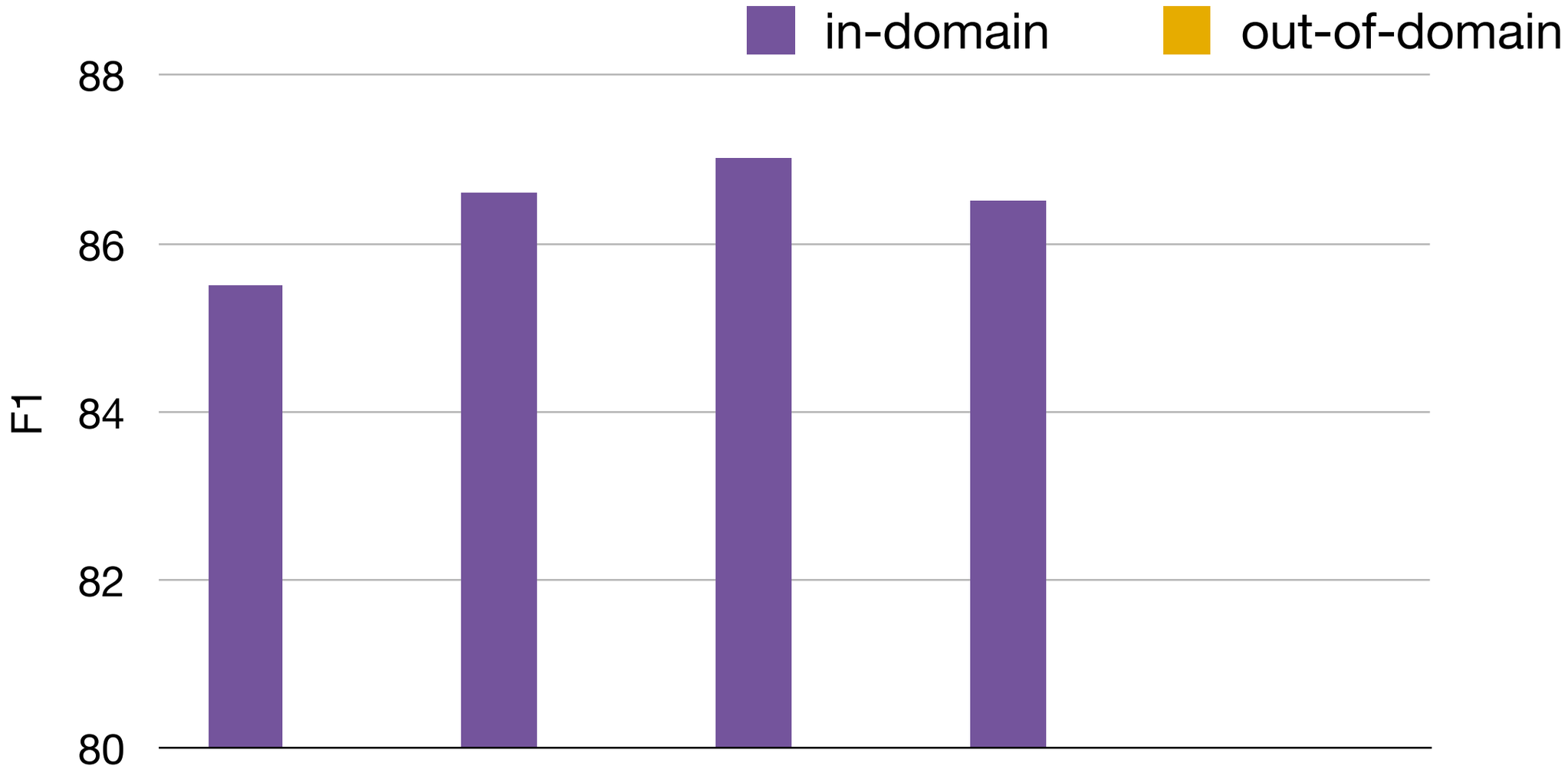
SemEval ’15. Micro-averaged labeled F_1



	Neurbo	Pipeline
Syntax		✓
Backprop	N/A	
Hard decision $\hat{\mathbf{Z}}$	N/A	✓
Projection	N/A	

Neurbo: Peng et al., 2017

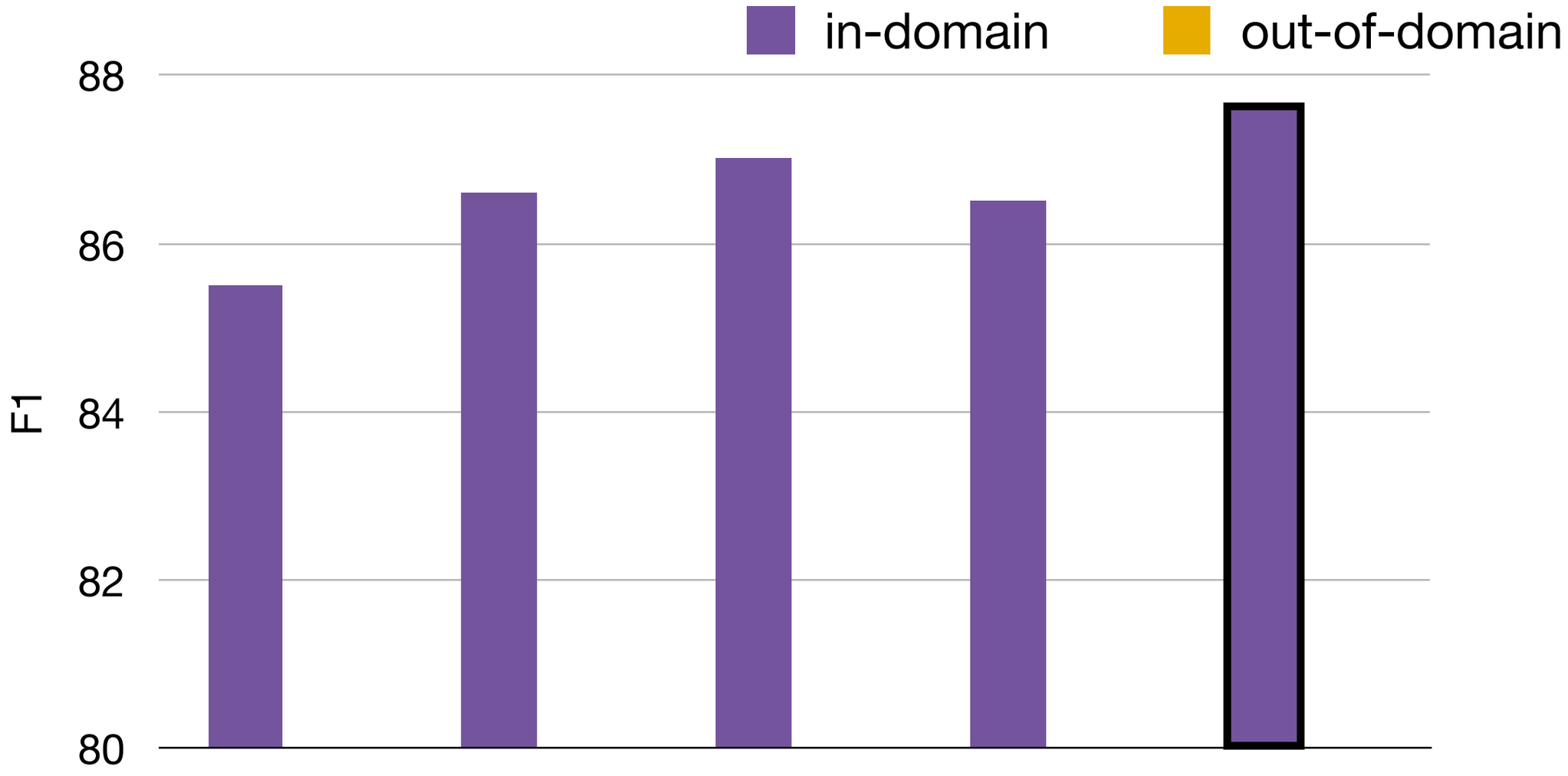
SemEval ’15. Micro-averaged labeled F_1



	Neurbo	Pipeline	STE	Structured Att.
Syntax		✓	✓	✓
Backprop	N/A		✓	✓
Hard decision $\hat{\mathbf{Z}}$	N/A	✓	✓	
Projection	N/A			

Neurbo: Peng et al., 2017

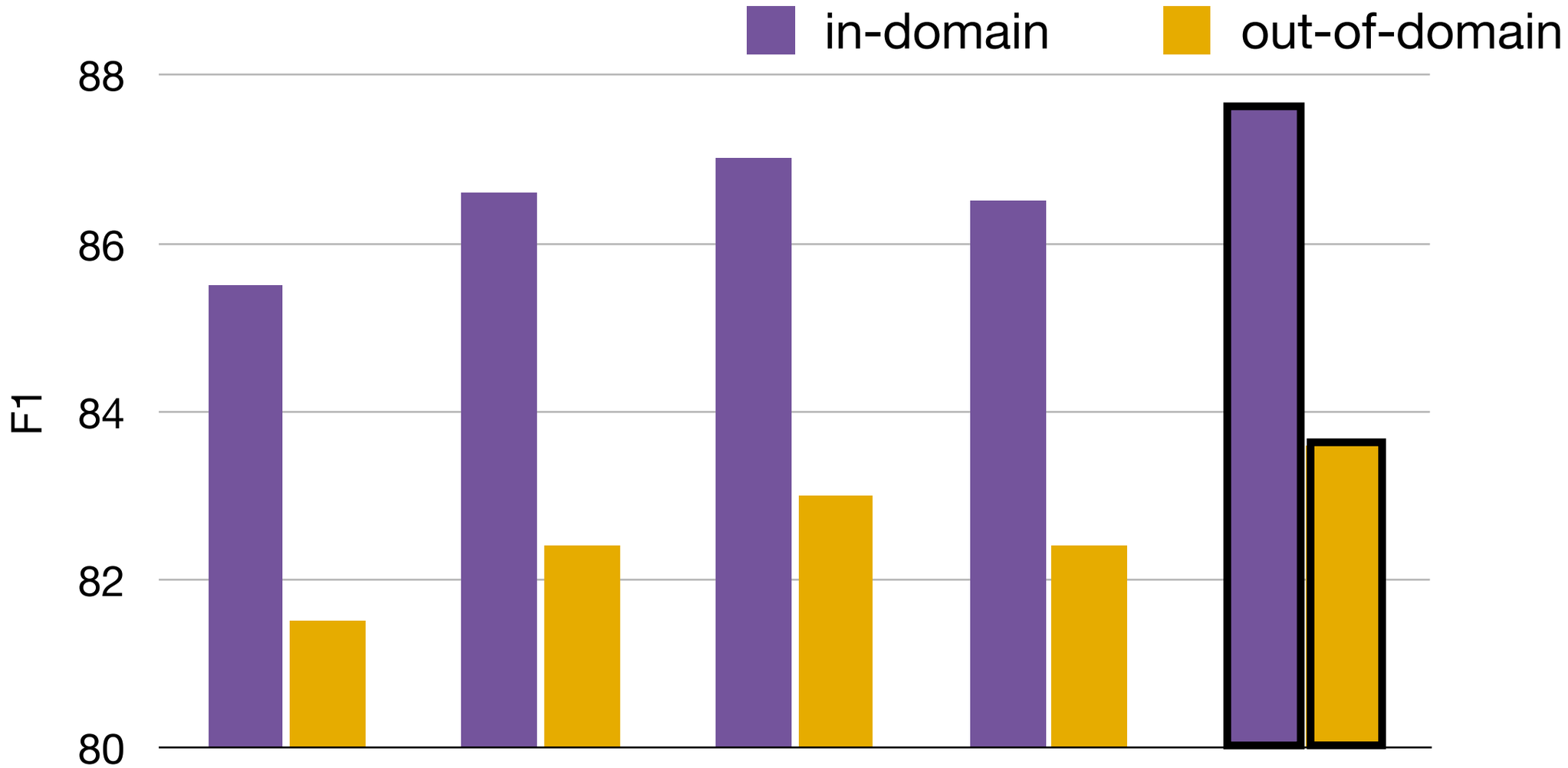
SemEval ’15. Micro-averaged labeled F_1



	Neurbo	Pipeline	STE	Structured Att.	SPIGOT
Syntax		✓	✓	✓	✓
Backprop	N/A		✓	✓	✓
Hard decision $\hat{\mathbf{Z}}$	N/A	✓	✓		✓
Projection	N/A				✓

Neurbo: Peng et al., 2017

SemEval ’15. Micro-averaged labeled F_1



	Neurbo	Pipeline	STE	Structured Att.	SPIGOT
Syntax		✓	✓	✓	✓
Backprop	N/A		✓	✓	✓
Hard decision $\hat{\mathbf{z}}$	N/A	✓	✓		✓
Projection	N/A				✓

Neurbo: Peng et al., 2017

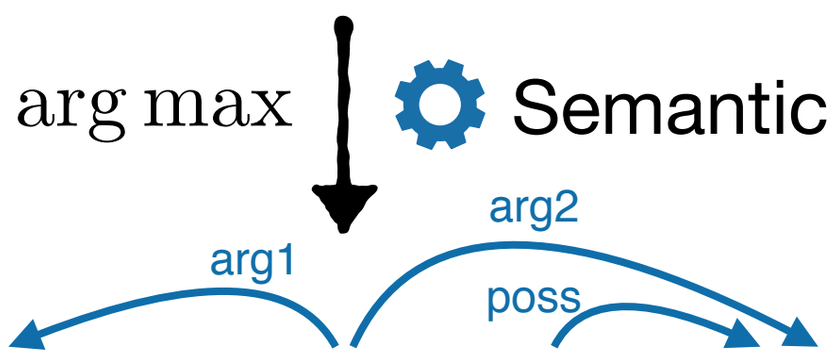
Semantic Parsing for Sentiment Classification

Input

Shareholders took their money

$\arg\max$ Semantic Parser θ

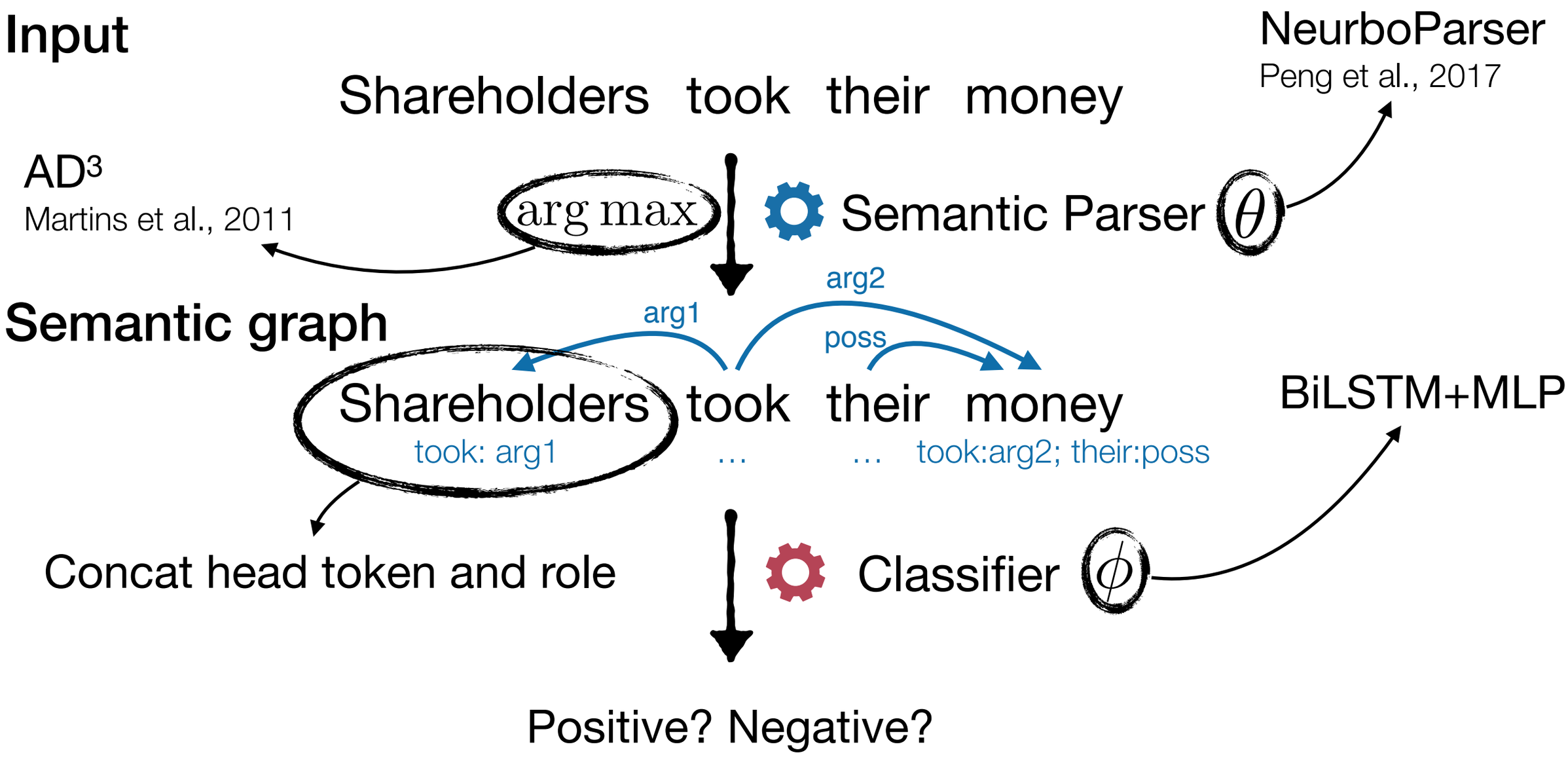
Semantic graph

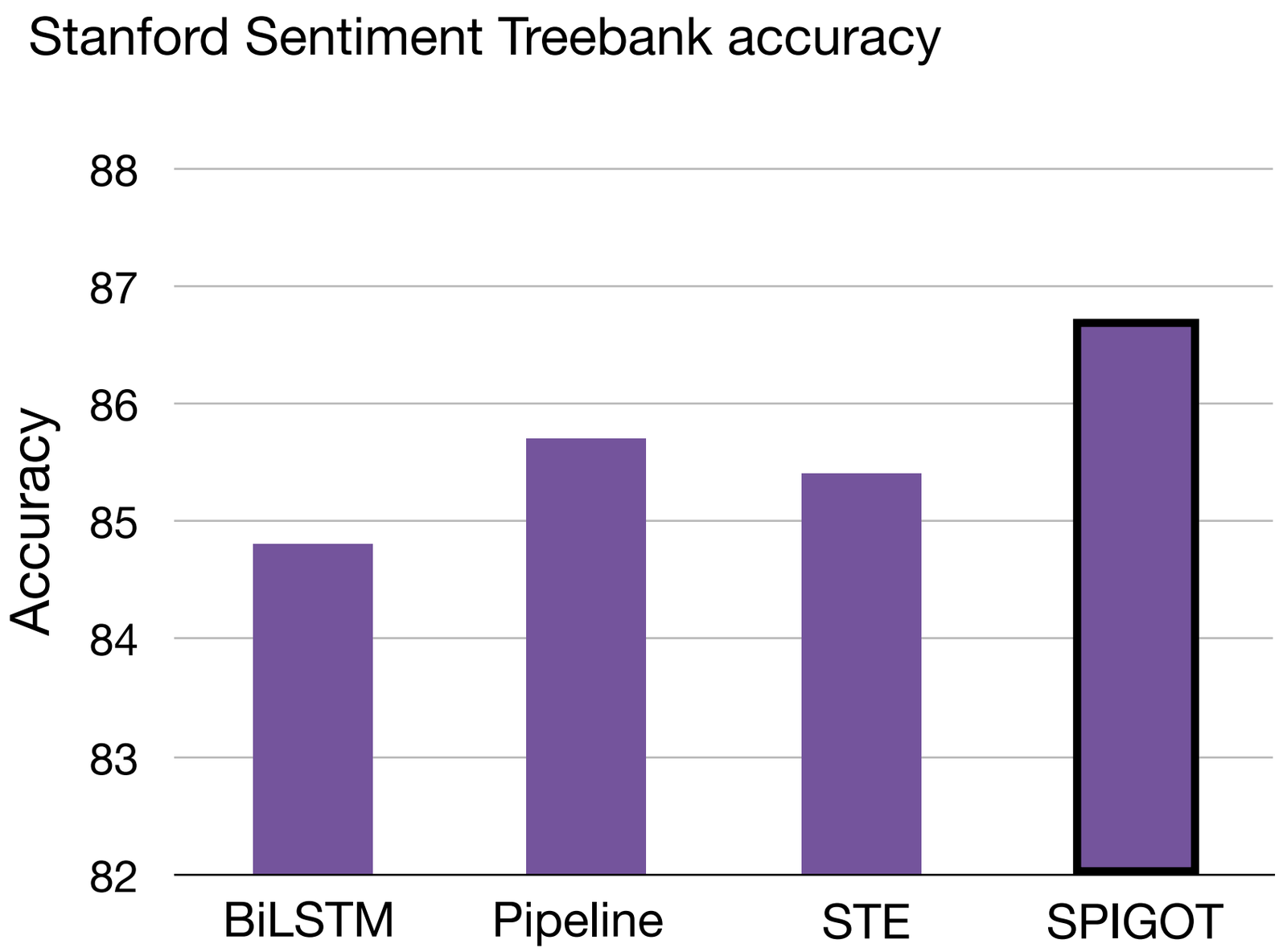


Classifier ϕ

Positive? Negative?

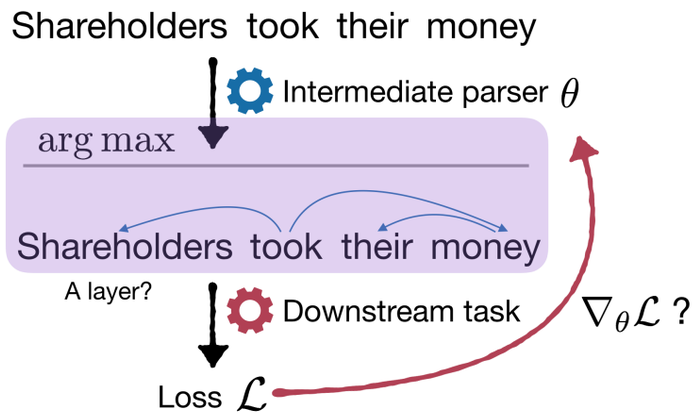
Semantic Parsing for Sentiment Classification




Stanford Sentiment Treebank accuracy
Accuracy
88
87
86
85
84
83
82
BiLSTM
Pipeline
STE
SPIGOT

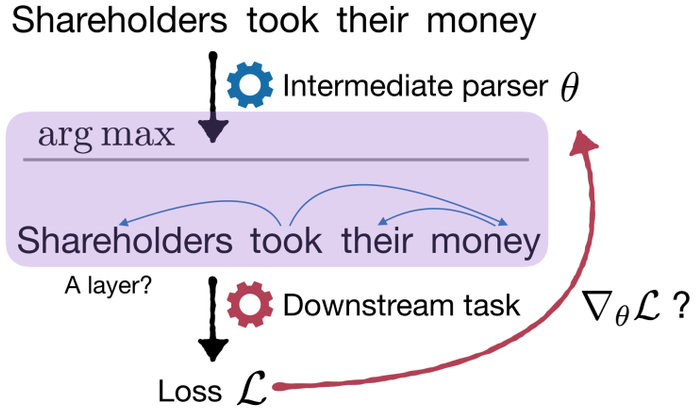
Conclusion

Problem



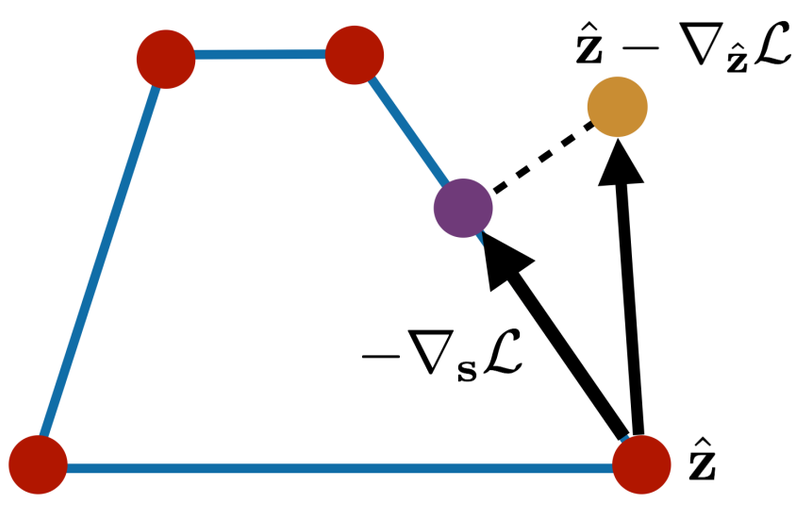
Conclusion

Problem



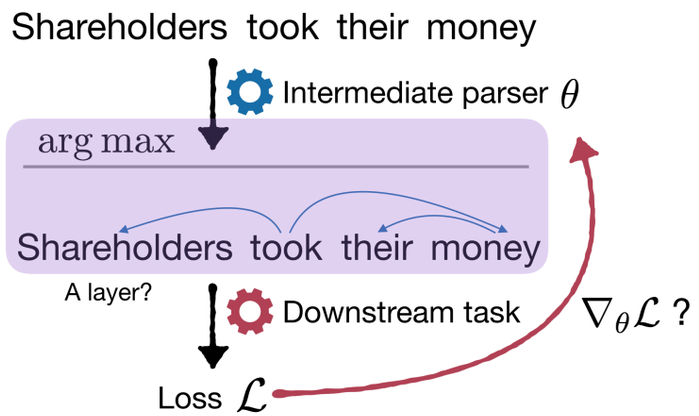
Method

SPIGOT



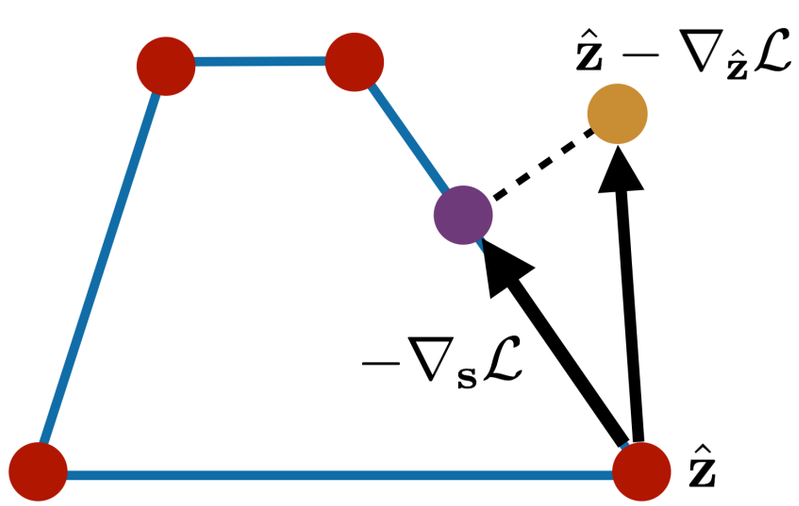
Conclusion

Problem



Method

SPIGOT



Results

Thank you!