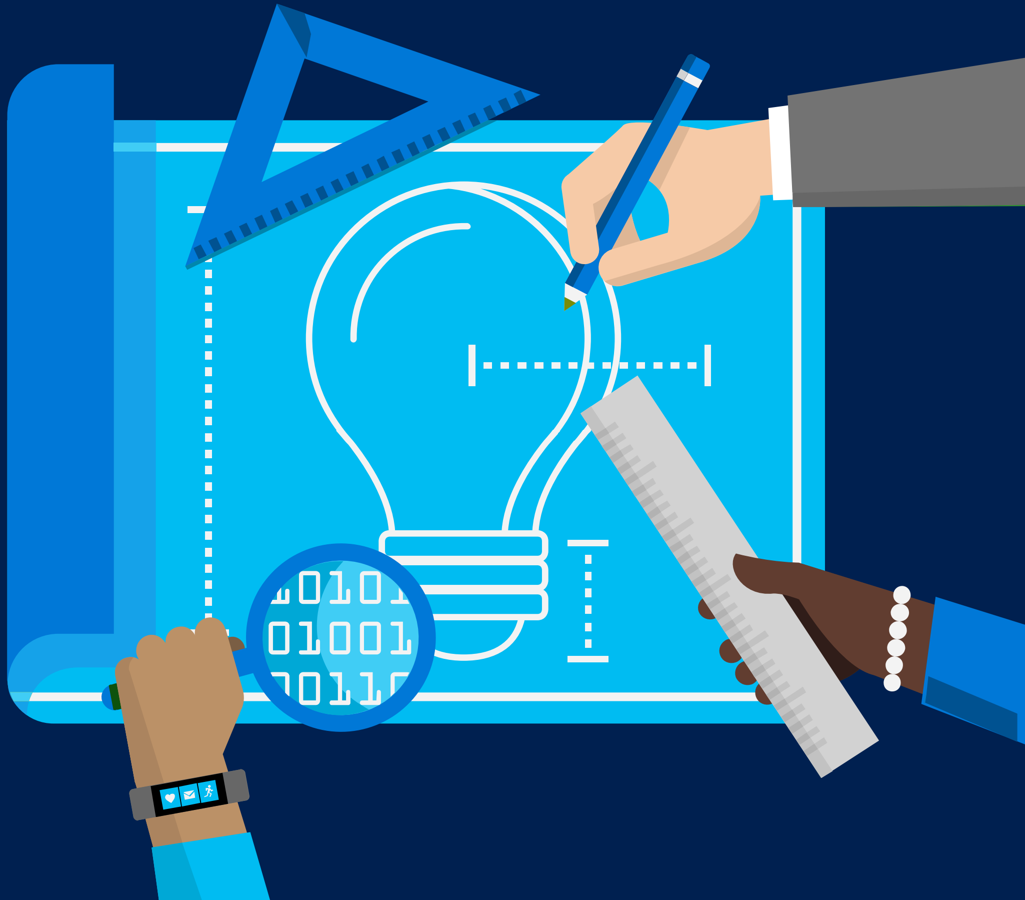
Learning to Understand Source Code with Machine Learning

Miltos Allamanis

Microsoft Research, Cambridge



Code as…

- Data → Software Engineering (SE) Tools
- Machine Learning (ML) component → Artificial Intelligence (AI) Tool

Code Understanding

Code Synthesis

Rule-based Expert Systems

Statistical NLP with hand-coded features

Deep Learning

translator

Source Code is Bimodal

Research in ML+Code

- Infer latent intent
- Ambiguous information
- Learned heuristics

ml4code.github.io

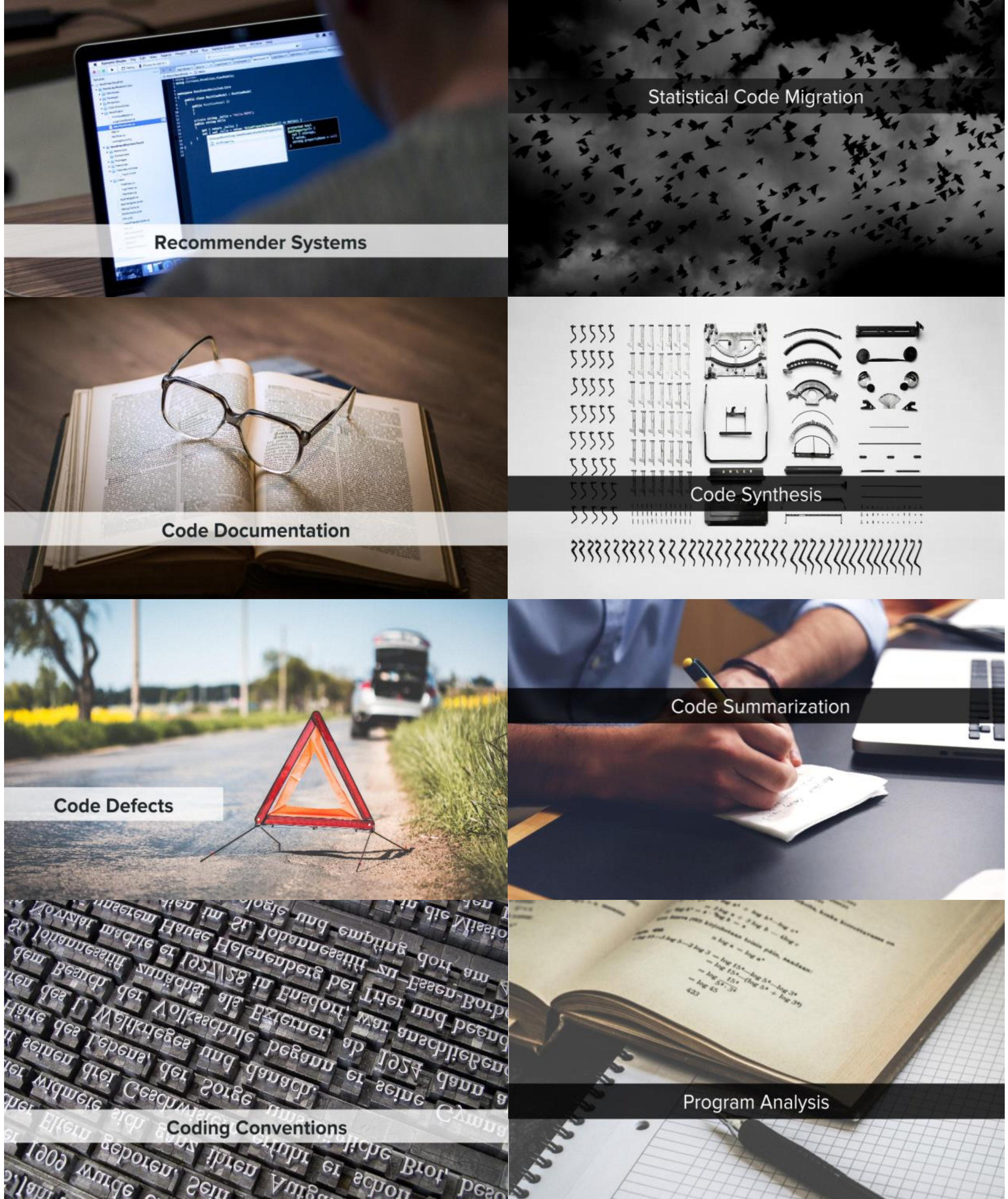
1

A Survey of Machine Learning for Big Code and Naturalness

MILTIADIS ALLAMANIS, Microsoft Research
EARL T. BARR, University College London
PREMKUMAR DEVANBU, University of California, Davis
CHARLES SUTTON, University of Edinburgh and The Alan Turing Institute

Research at the intersection of machine learning, programming languages, and software engineering has recently taken important steps in proposing learnable probabilistic models of source code that exploit code's abundance of patterns. In this article, we survey this work. We contrast programming languages against natural languages and discuss how these similarities and differences drive the design of probabilistic models. We present a taxonomy based on the underlying design principles of each model and use it to navigate the literature. Then, we review how researchers have adapted these models to application areas and discuss cross-cutting and application-specific challenges and opportunities.

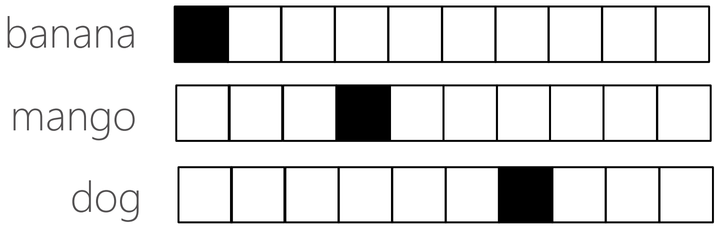
CCS Concepts: • Computing methodologies → Machine learning; Natural language processing; • Soft-



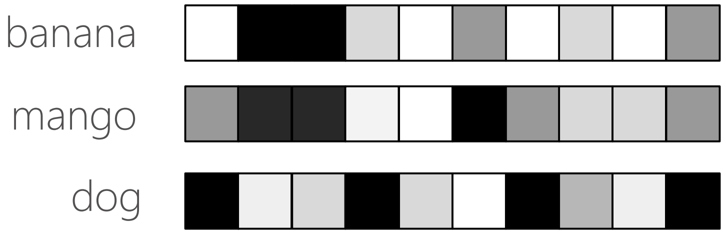
Outline

- Detecting Bugs with ML
- Practical Tips for Machine Learning
- Continuous [Code] Semantics?

Vector Space Representations



Local representation

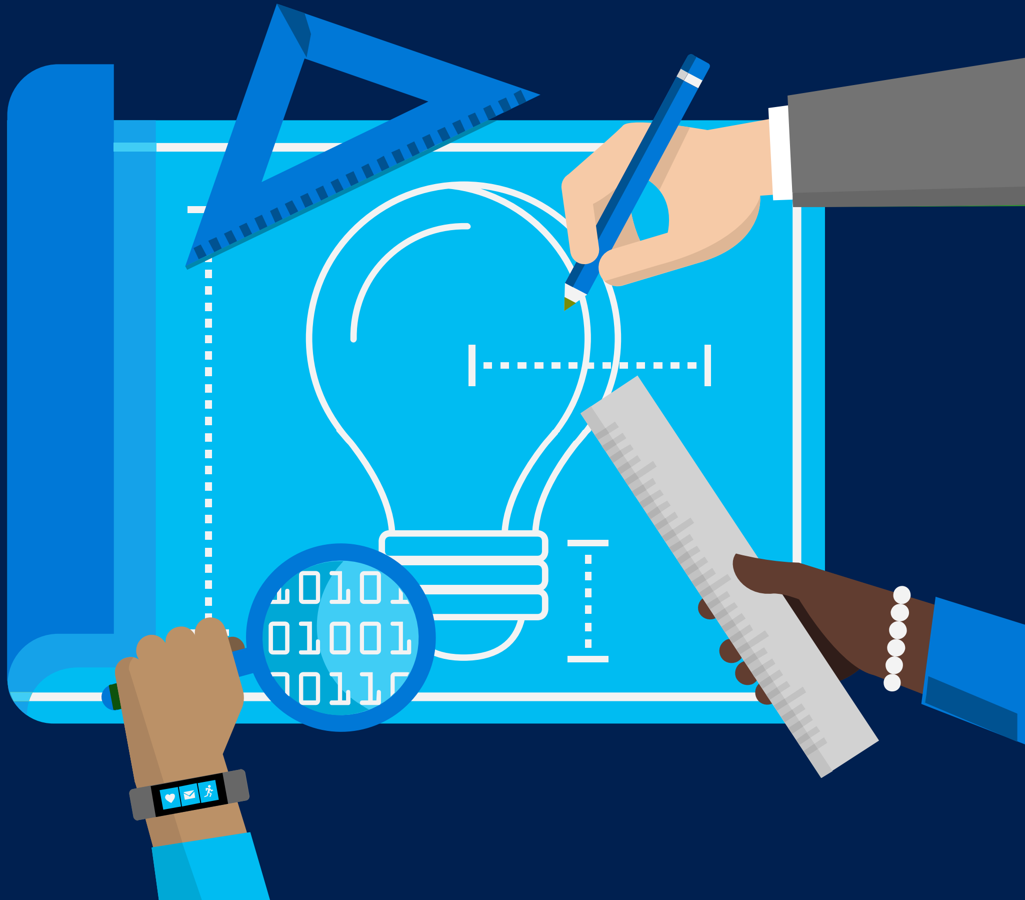


Distributed representation

Microsoft

Detecting Variable Misuses

with Graph Neural Networks



Joint work with
Marc Brockschmidt, Mahmoud Khademi

Target Task

```
var clazz=classTypes["Root"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull(clazz);

var first=classTypes["RecClass"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull([redacted]);

Assert.Equal("string", first.Properties["Name"].Name);
Assert.False(clazz.Properties["Name"].IsArray);
```

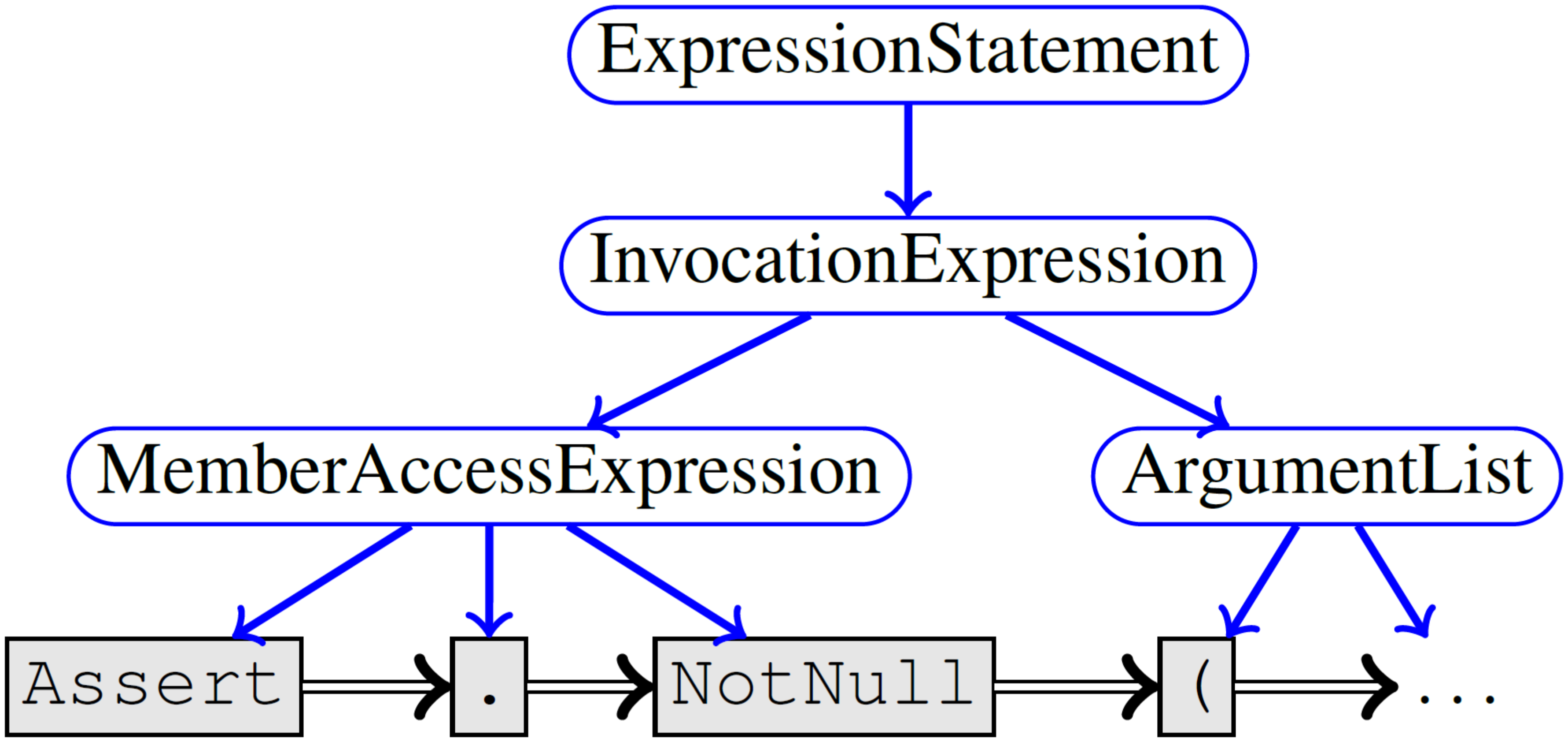
Possible type-correct options: `clazz, first`

Not easy to catch with static analysis tools.

Learning to Reason about Code

```
int SumPositive(int[] arr, int lim) {
  int sum = 0;
  for (int i = 0; i < lim; i++)
    if (arr[i]>0)
        sum += arr[i];

  return sum;
}
```

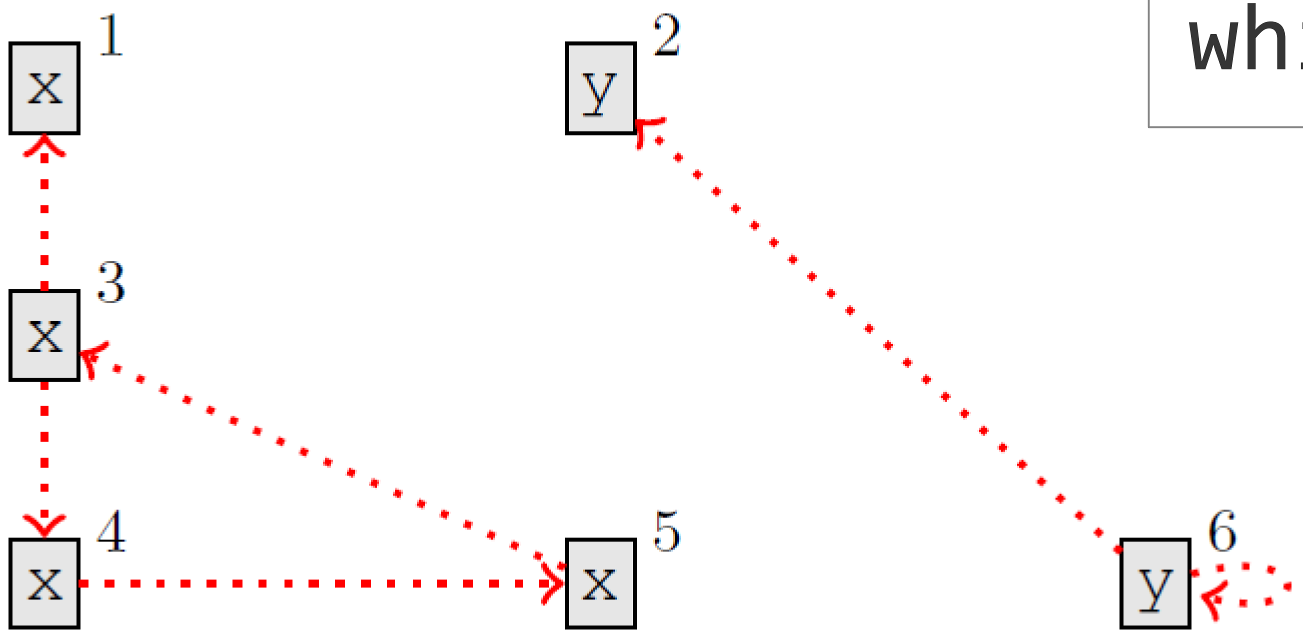
Representing Program Structure as a Graph




```
var clazz=classTypes["Root"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull(clazz);

var first=classTypes["RecClass"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull( clazz );

Assert.Equal("string", first.Properties["Name"].Name);
Assert.False(clazz.Properties["Name"].IsArray);
```

Representing Program Structure as a Graph

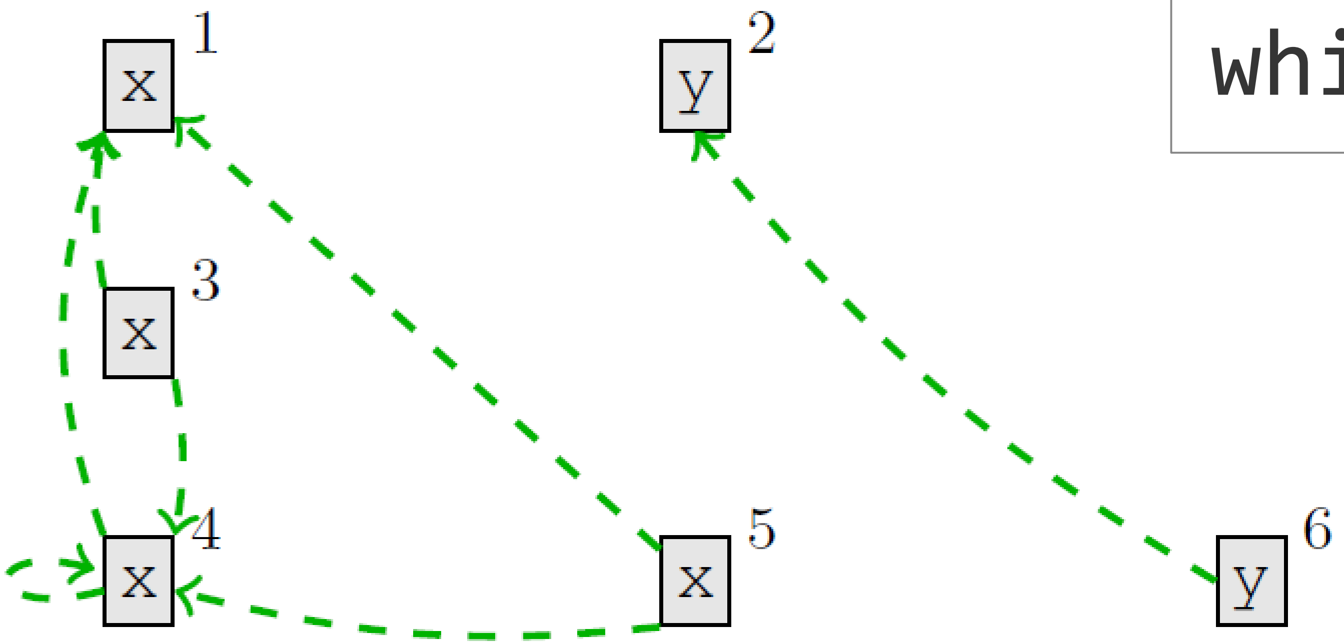
```
(x¹,y²) = Foo();
while (x³>0) x⁴=x⁵+y⁶
```



LastUse

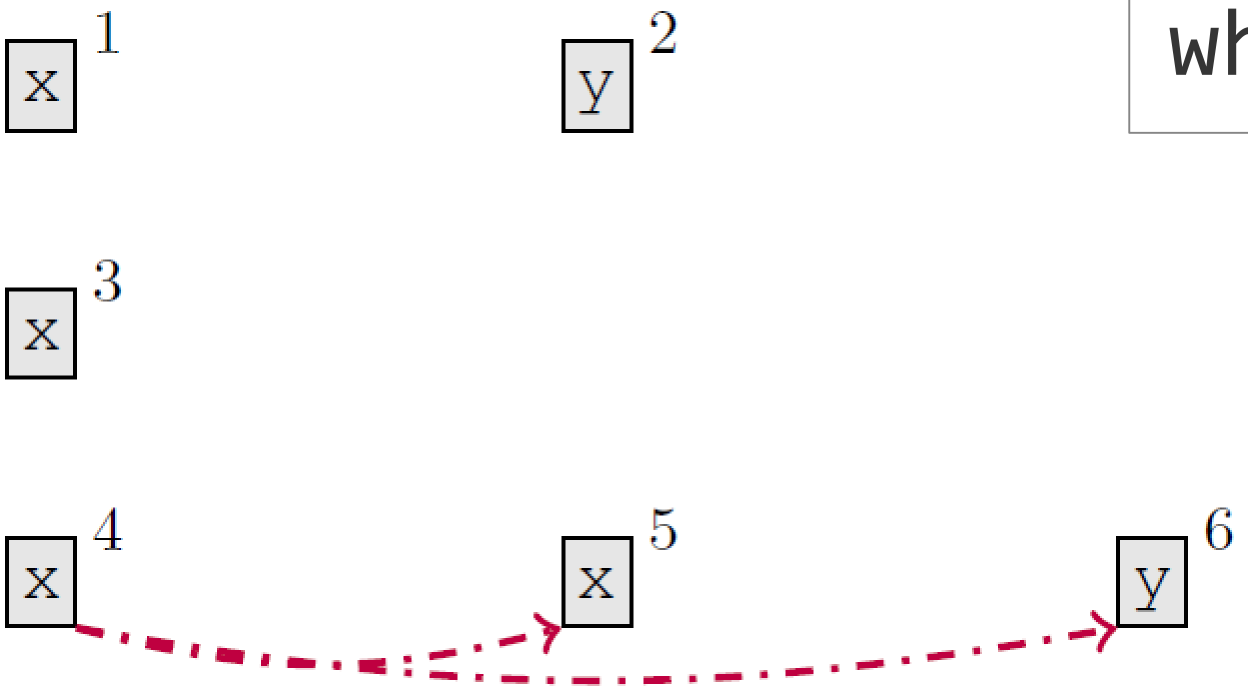
Representing Program Structure as a Graph

(x^1,y^2) = Foo();
while $(x^3>0)$ $x^4=x^5+y^6$



LastWrite

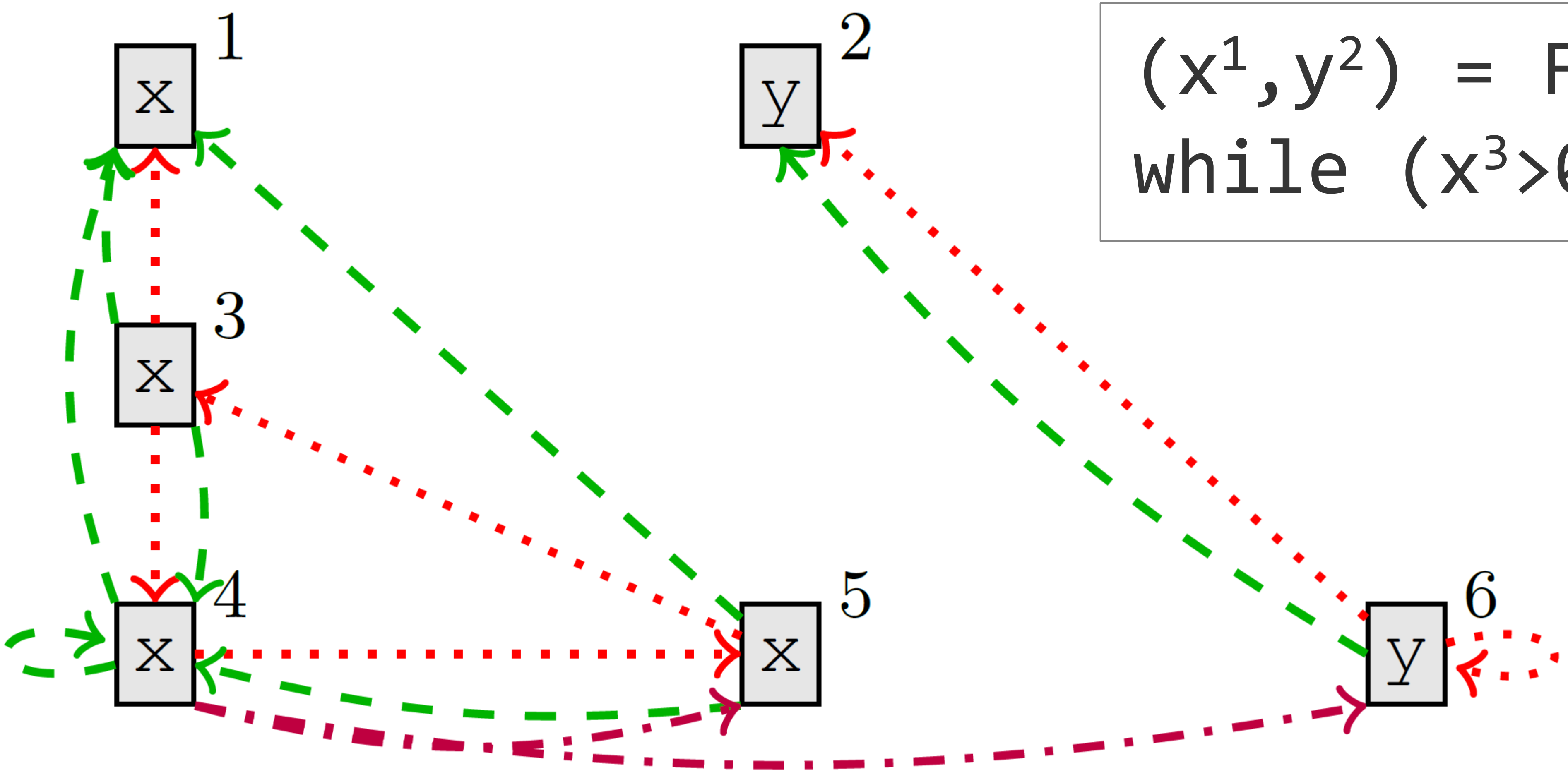
Representing Program Structure as a Graph



(x^1, y^2) = Foo();
while $(x^3 > 0)$ $x^4 = x^5 + y^6$

ComputerFrom

Representing Program Structure as a Graph



$(x^1, y^2) = \texttt{Foo}();$
$\texttt{while}\ (x^3 > 0)\ x^4 = x^5 + y^6$

Representing Program Structure as a Graph

Additional Edge Types:

- ReturnsTo

```
int foo(int sum)
{
    …
    return x;
}
```

Representing Program Structure as a Graph

Additional Edge Types:

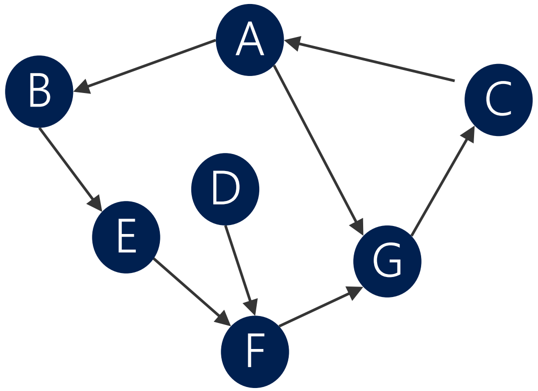
- ReturnsTo
- FormalArgName

```
b = foo(result);
```

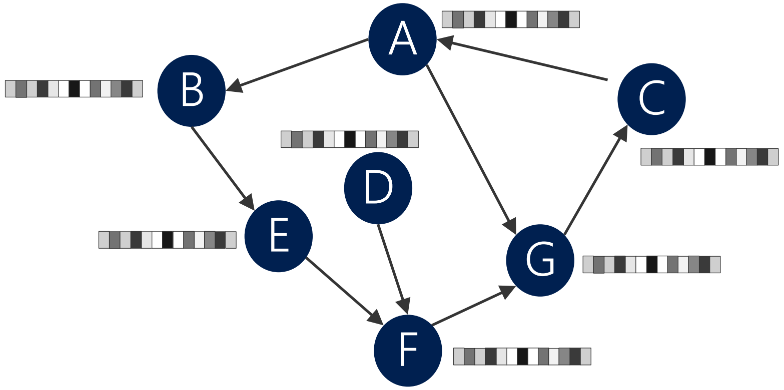
sum

```
void foo(int sum) { … }
```

Graph Neural Networks



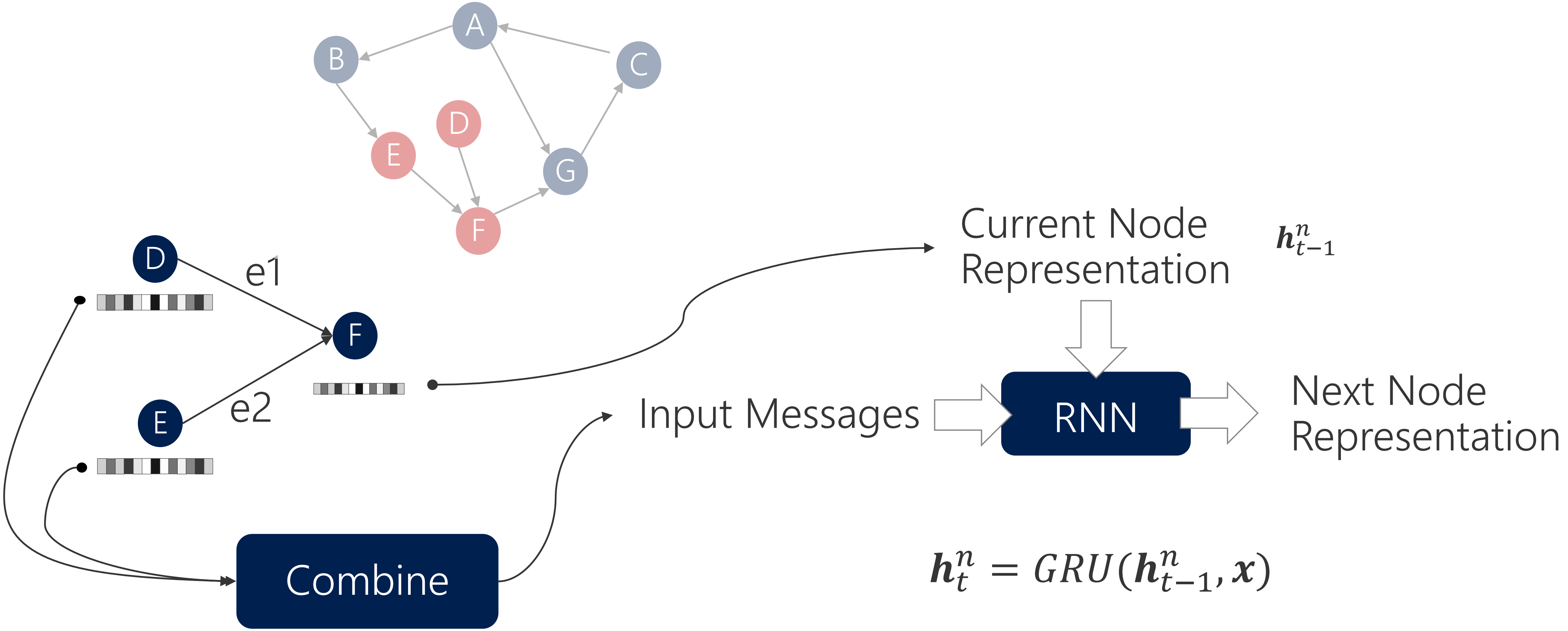
Graph Representation
of Problem



Initial Representation
of each node

Li et al (2015). Gated Graph Sequence Neural Networks.

Graph Neural Networks: Message Propagation

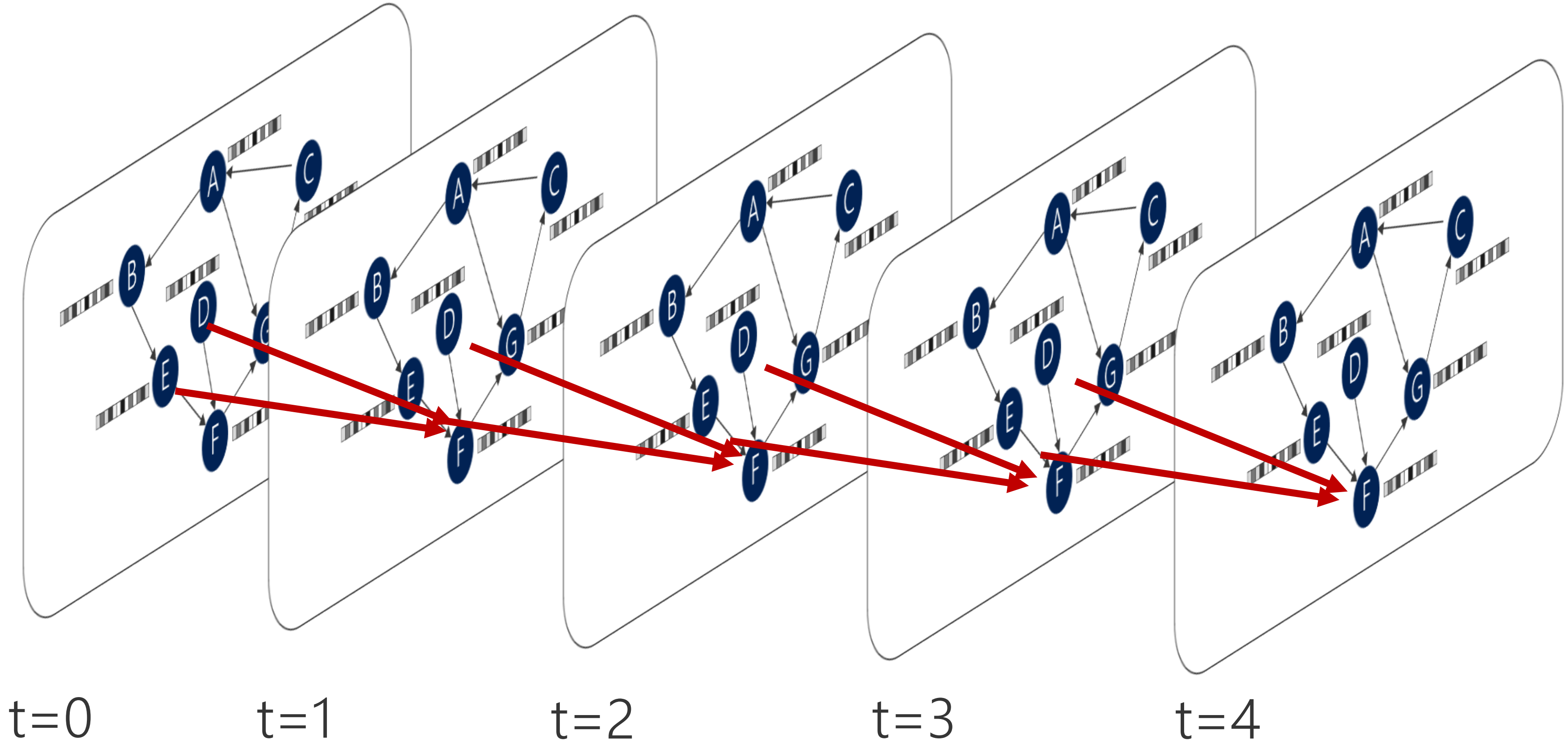


$$h_t^n = GRU(h_{t-1}^n, x)$$

$$x = \sum_{n' \in neig(n)} E_{\tau(n' \to n)} h_{t-1}^{n'} + b_{\tau(n' \to n)}$$

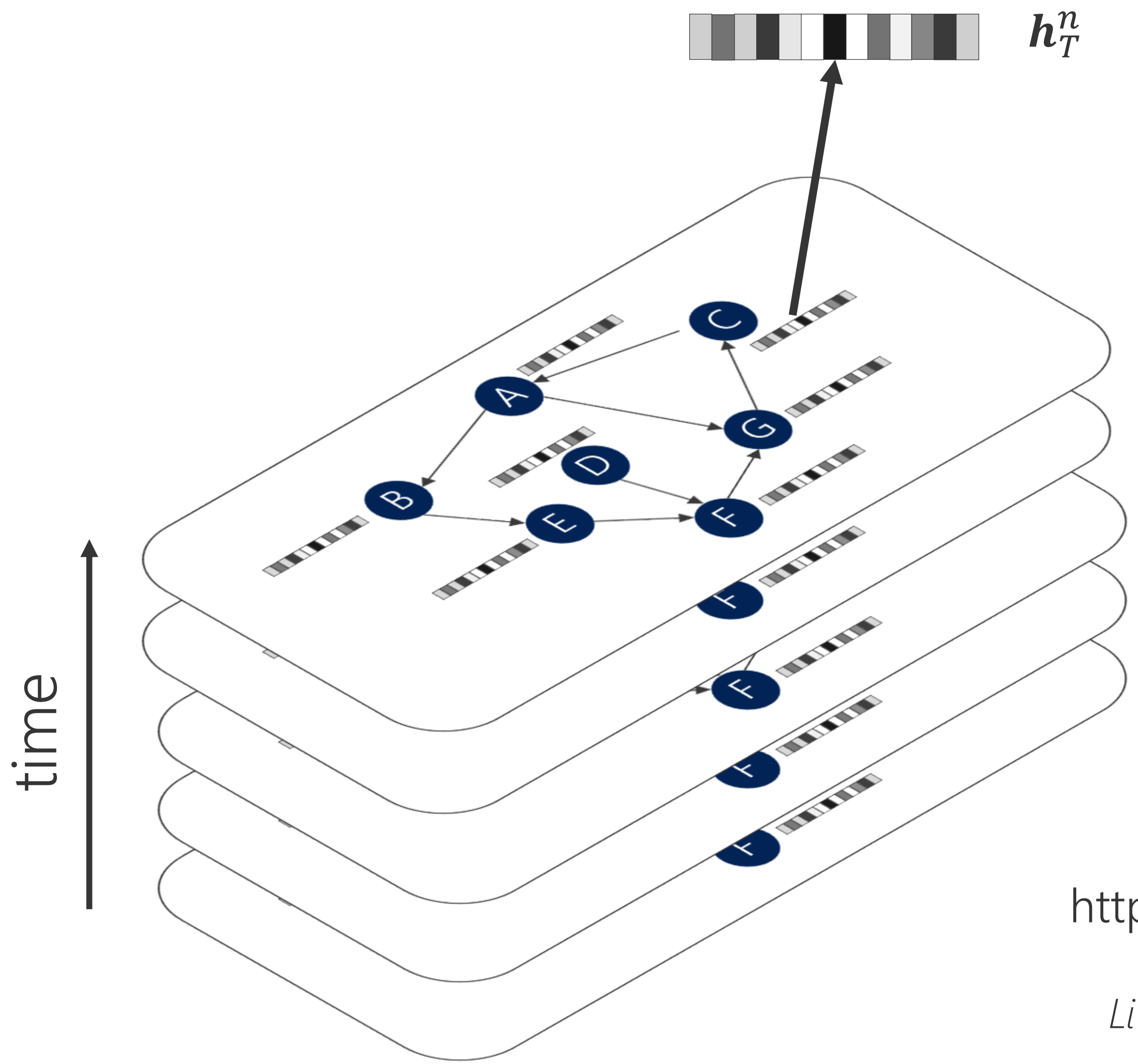
Li et al (2015). Gated graph sequence neural networks.

Graph Neural Networks: Unrolling



Li et al (2015). Gated graph sequence neural networks.

Graph Neural Networks: Unrolling



Node Classification: $y_i = \mathrm{MLP}(\boldsymbol{h}_T^n)$

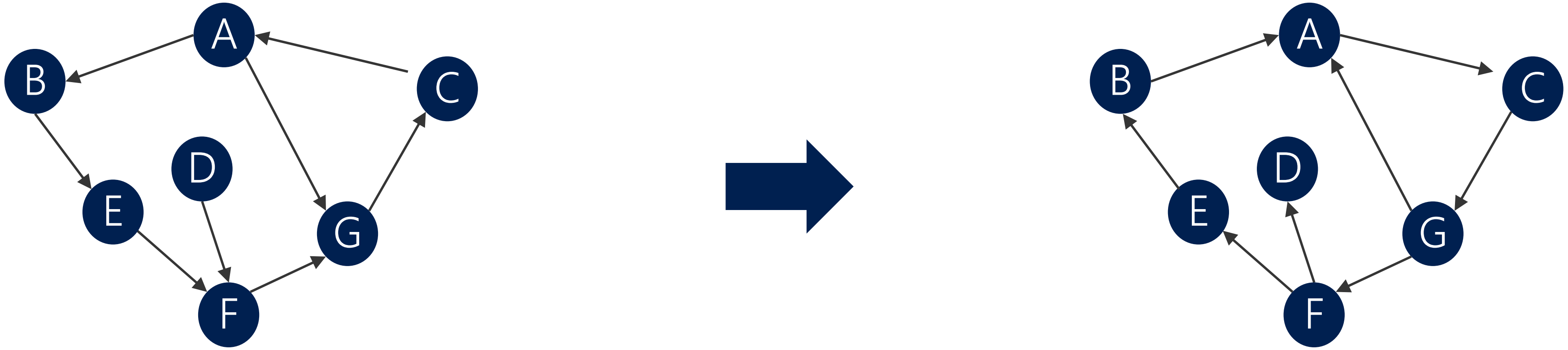
Node Selection: $\boldsymbol{y} = \mathrm{softmax}(\mathrm{f}(\{\boldsymbol{h}_T^n\}))$

Graph Classification : $\boldsymbol{y} = \mathrm{g}(\{\boldsymbol{h}_T^n\})$

https://github.com/Microsoft/gated-graph-neural-network-samples

Li et al (2015). Gated Graph Sequence Neural Networks.

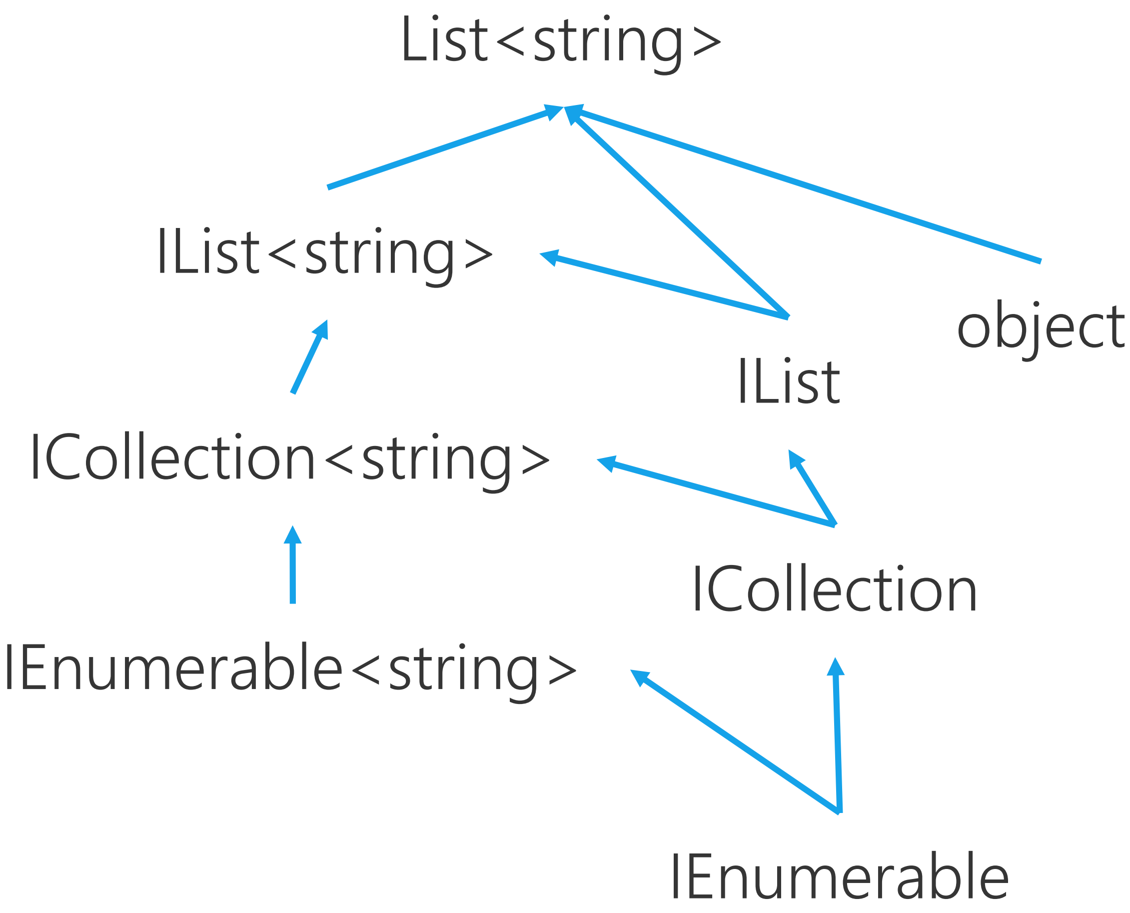
Backwards Edges



Graph Representation
of Problem

Li et al (2015). Gated Graph Sequence Neural Networks.

Representing Variable Type Information



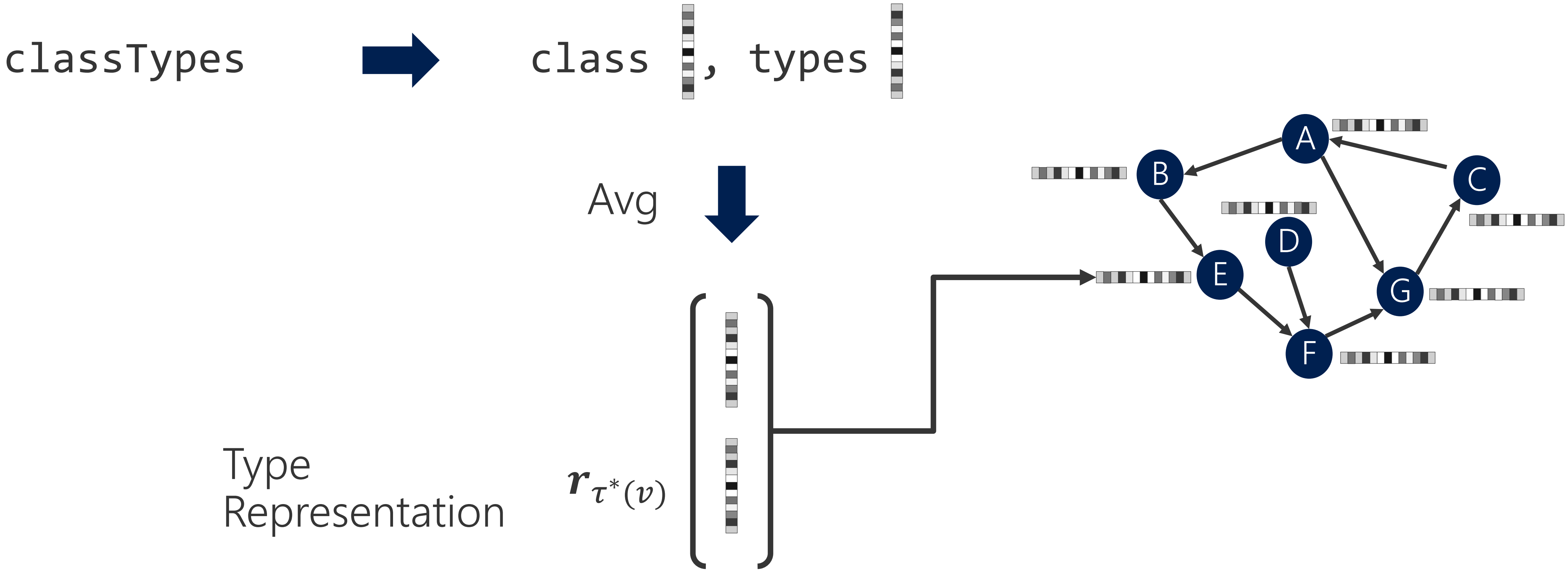
$$\tau^*(v) = \{\tau_{List<string>}, \tau_{IList}, \tau_{object}, \dots\}$$

$$\boldsymbol{r}_{List<string>} \quad \boldsymbol{r}_{IList} \quad \boldsymbol{r}_{object}$$

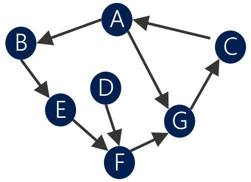
Elementwise Max

$$\boldsymbol{r}_{\tau^*(v)}$$

Representing Nodes



Graph Representation for Variable Misuse




```
var clazz=classTypes["Root"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull(clazz);

var first=classTypes["RecClass"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull([redacted]);

Assert.Equal("string", first.Properties["Name"].Name);
Assert.False(clazz.Properties["Name"].IsArray);
```

Possible type-correct options: `clazz, first`

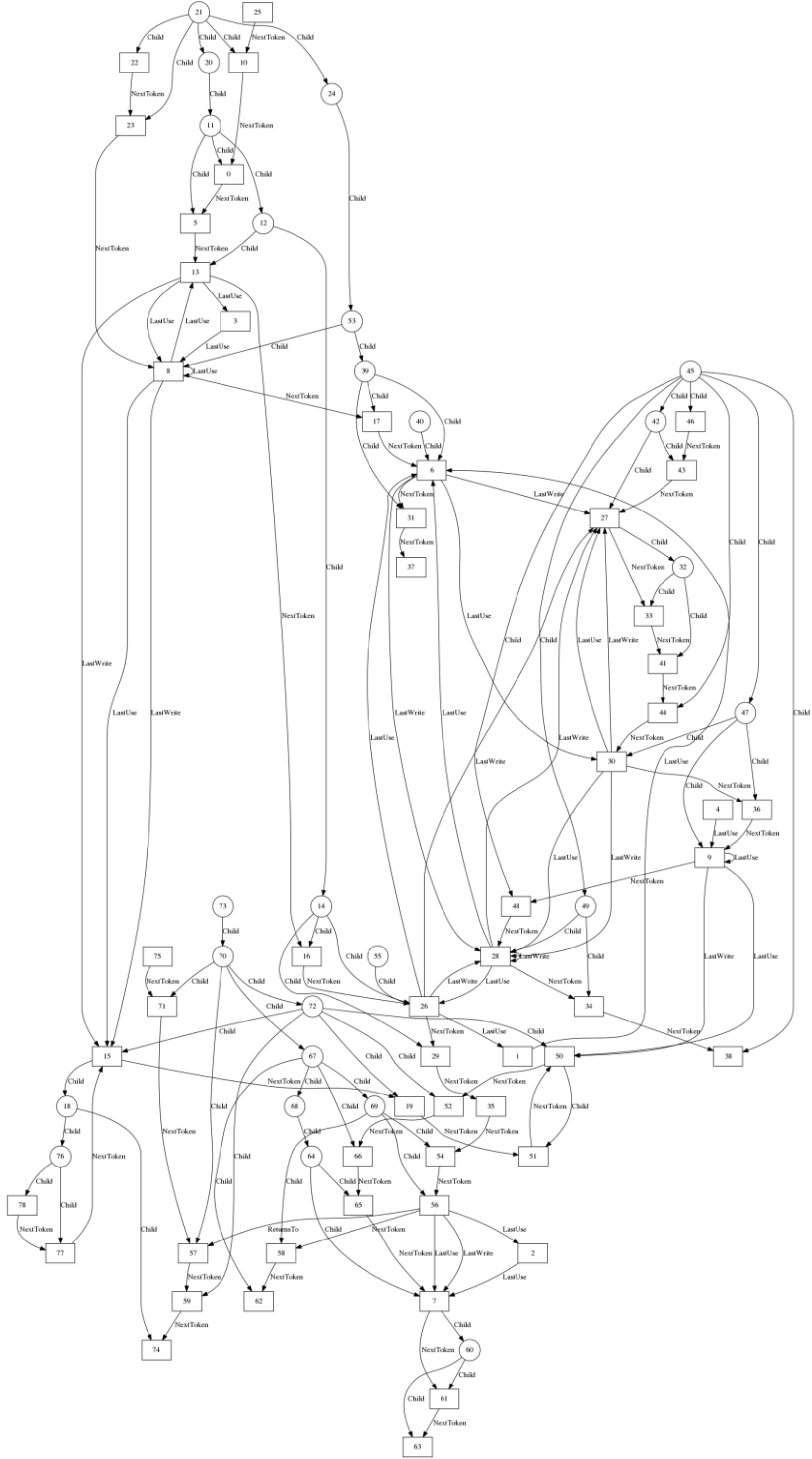
Graph Representation for Variable Misuse

```
var clazz=classTypes["Root"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull(clazz);

var first=classTypes["RecClass"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull( SLOT );        first        clazz

Assert.Equal("string", first.Properties["Name"].Name);
Assert.False(clazz.Properties["Name"].IsArray);
```

Goal: make the representation of **SLOT** as close as possible to the representation of the correct candidate node

From Code to Graphs

```
int SumPositive(int[] arr, int lim) {
  int sum = 0;
        for (int i=0; <; ++)
          if ([ ]>0)
             +=[ ];
  return sum;
}
```



~900 nodes/graph ~8k edges/graph

Implementation

- Sparse TensorFlow implementation
- 16 edge types (forward + backward)

Stats

- 65 graphs/s during training
- 272 graphs/s during testing

https://github.com/Microsoft/gated-graph-neural-network-samples

Dataset

2.9MLOC

Name	Git SHA	kLOCs	Slots	Vars	Description
Akka.NET	719335a1	240	51.3k	51.2k	Actor-based Concurrent & Distributed Framework
AutoMapper	2ca7c2b5	46	3.7k	10.7k	Object-to-Object Mapping Library
BenchmarkDotNet	1670ca34	28	5.1k	6.1k	Benchmarking Library
BotBuilder	190117c3	44	6.4k	8.7k	SDK for Building Bots
choco	93985688	36	3.8k	5.2k	Windows Package Manager
commandline†	09677b16	11	1.1k	2.3k	Command Line Parser
CommonMark.NET[Dev]	f3d54530	14	2.6k	1.4k	Markdown Parser
Dapper	931c700d	18	3.3k	4.7k	Object Mapper Library
EntityFramework	fa0b7ec8	263	33.4k	39.3k	Object-Relational Mapper
Hangfire	ffc4912f	33	3.6k	6.1k	Background Job Processing Library
Humanizer†	cc11a77e	27	2.4k	4.4k	String Manipulation and Formatting
Lean†	f574bfd7	190	26.4k	28.3k	Algorithmic Trading Engine
Nancy	72e1f614	70	7.5k	15.7	HTTP Service Framework
Newtonsoft.Json	6057d9b8	123	14.9k	16.1k	JSON Library
Ninject	7006297f	13	0.7k	2.1k	Code Injection Library
NLog	643e326a	75	8.3k	11.0k	Logging Library
Opserver	51b032e7	24	3.7k	4.5k	Monitoring System
OptiKey	7d35c718	34	6.1k	3.9k	Assistive On-Screen Keyboard
orleans	e0d6a150	300	30.7k	35.6k	Distributed Virtual Actor Model
Polly	0afdbc32	32	3.8k	9.1k	Resilience & Transient Fault Handling Library
quartznet	b33e6f86	49	9.6k	9.8k	Scheduler
ravendb[Dev]	55230922	647	78.0k	82.7k	Document Database
RestSharp	70de357b	20	4.0k	4.5k	REST and HTTP API Client Library
Rx.NET	2d146fe5	180	14.0k	21.9k	Reactive Language Extensions
scriptcs	f3cc8bcb	18	2.7k	4.3k	C# Text Editor
ServiceStack	6d59da75	231	38.0k	46.2k	Web Framework
ShareX	718dd711	125	22.3k	18.1k	Sharing Application
SignalR	fa88089e	53	6.5k	10.5k	Push Notification Framework
Wox	cdaf6272	13	2.0k	2.1k	Application Launcher

GitHub

Quantitative Results

Accuracy (%)	Syntax Only	Avg BiRNN	GGNN
Seen Projects	49.6	73.5	**82.6**

3.8 type-correct alternative variables per slot (median 3, σ= 2.6)

Quantitative Results

Accuracy (%)	Syntax Only	Avg BiRNN	GGNN
Seen Projects	49.6	73.5	**82.6**
Unseen Projects	36.2	59.7	**76.0**

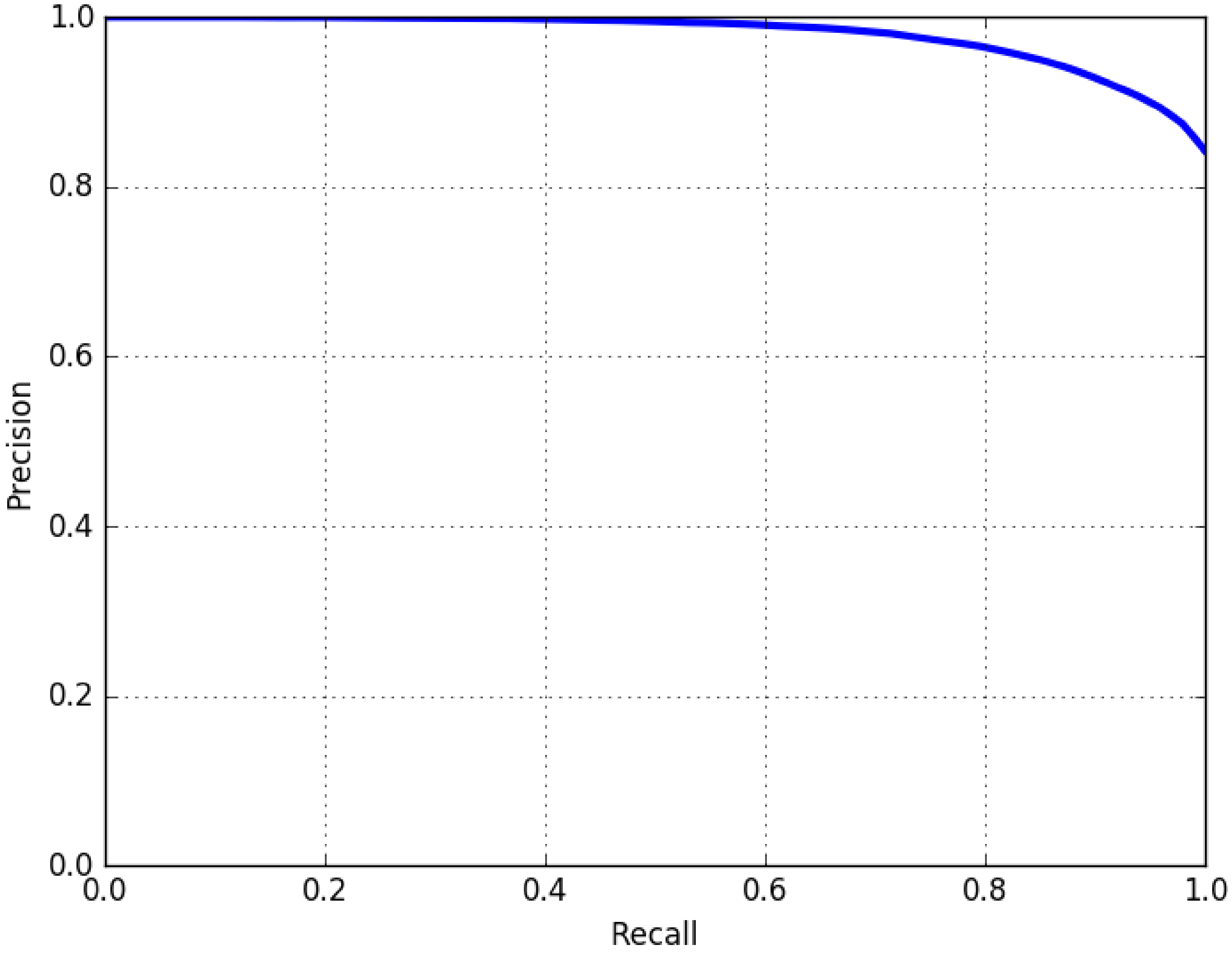
3.8 type-correct alternative variables per slot (median 3, σ= 2.6)

```
bool TryFindGlobalDirectivesFile(string baseDirectory, string fullPath, out string path) {
   baseDirectory = baseDirectory.TrimEnd(Path.DirectorySeparatorChar);
   var directivesDirectory = Path.GetDirectoryName([redacted])
                                       .TrimEnd(Path.DirectorySeparatorChar);
   while (directivesDirectory != null && directivesDirectory.Length >= baseDirectory.Length) {
      path = Path.Combine(directivesDirectory, GlobalDirectivesFileName);
      if (File.Exists(path)) return true;

      directivesDirectory = Path.GetDirectoryName(directivesDirectory)
                                     .TrimEnd(Path.DirectorySeparatorChar);
   }
   path = null;
   return false;
}
```

What the model sees...

```
bool TryFindGlobalDirectivesFile(string baseDirectory, string fullPath, out string path) {
    baseDirectory = baseDirectory.TrimEnd(Path.DirectorySeparatorChar);
    var directivesDirectory = Path.GetDirectoryName(fullPath)
                                        .TrimEnd(Path.DirectorySeparatorChar);
    while (directivesDirectory != null && directivesDirectory.Length >= baseDirectory.Length) {
        path = Path.Combine(directivesDirectory, GlobalDirectivesFileName);
        if (File.Exists(path)) return true;

        directivesDirectory = Path.GetDirectoryName(directivesDirectory)
                                        .TrimEnd(Path.DirectorySeparatorChar);
    }
    path = null;
    return false;
}
```

What the model sees...



Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Qualitative Results

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute]
public void MergeFrom(pb::CodedInputStream input) {
  uint tag;
  while ((tag = input.ReadTag()) != 0) {
    switch(tag) {
      default:
        input.SkipLastField();
        break;
      case 10: {
        #1 .AddEntriesFrom(input, _repeated_payload_codec);
        break;
      }
    }
  }
}
```

#1 Payload: 66%, payload_: 44%

Qualitative Results

```
public override bool IsDisposed
{
  get
  {
    lock ( #1 )
    {
      return  #2 ;
    }
  }
}
```

#1 _gate: 99%, _observers: 1%

#2 _isDisposed: 90%, _isStopped: 8%, HasObservers: 2%

Detecting Real-Life Bugs

```
protected void ValidateRestorePreconditions(string backupFilename) {
  if (IsValidBackup(backupFilename) == false) {
    output("Error:"+ backupLocation +" doesn’t look like a valid backup");
    output("Error: Restore Canceled");
    throw new InvalidOperationException(
        backupLocation + " doesn’t look like a valid backup");
  }
  ...
```

2 src/VisualStudio/Core/Def/Implementation/TableDataSource/AbstractTableEntriesSnapshot.cs

View

```
@@ -162,7 +162,7 @@ protected LinePosition GetTrackingLineColumn(Workspace workspace, DocumentId doc
```

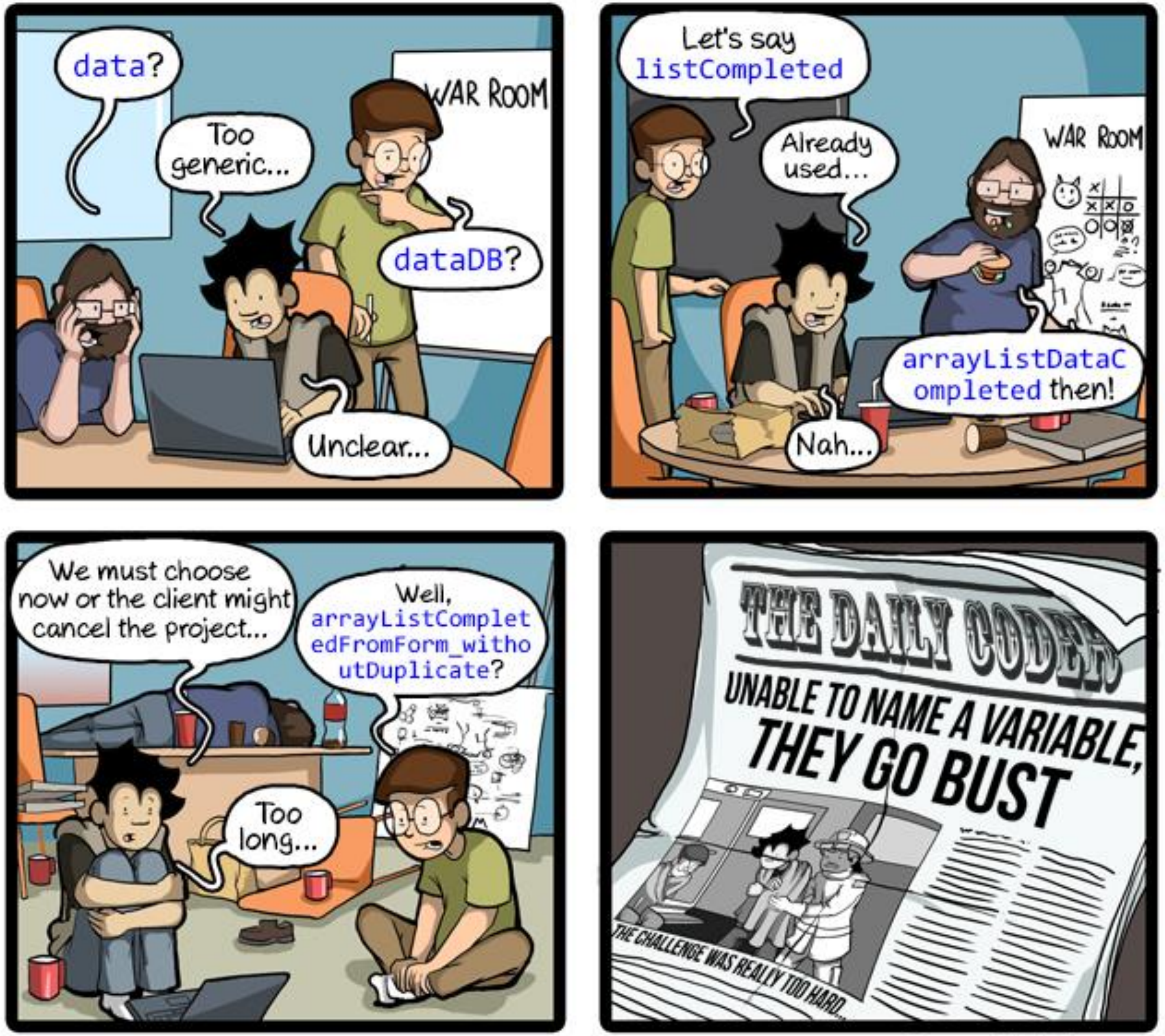
```diff
                 }
 
                 var currentSnapshot = textBuffer.CurrentSnapshot;
-                return GetLinePosition(snapshot, trackingPoint);
+                return GetLinePosition(currentSnapshot, trackingPoint);
             }
 
             private LinePosition GetLinePosition(ITextSnapshot snapshot, ITrackingPoint trackingPoint)
```

https://github.com/dotnet/roslyn/pull/23437

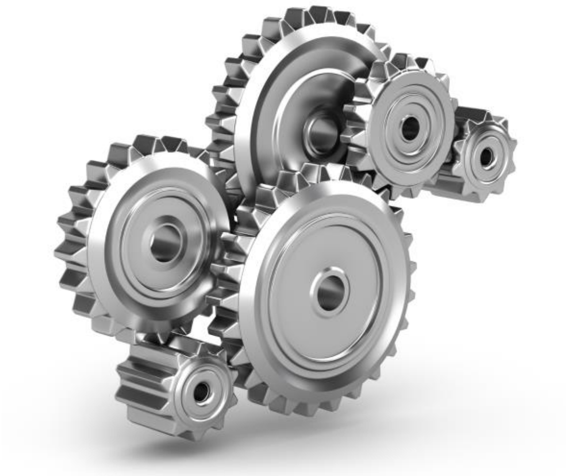
```
        private string GetTestOutputFilePath(string filepath)
        {
            string outputFilePath = @"Z:\";

            try
            {
                outputFilePath = Path.GetDirectoryName(_filePath);
            }
            catch (ArgumentException)
            {
            }

            if (string.IsNullOrEmpty(outputFilePath))
            {
                outputFilePath = @"Z:\";
            }

            return this.CompilationOptions == null ? "" : Path.Combine(outputFilePath, this.Assembly
        }
    }
}
```

https://github.com/dotnet/roslyn/pull/23437

Learning to Name Source Code



Learning to Name Source Code



A name reflects important aspects of code **functionality**.

Learning to name source code is a first step in *understanding* code through machine learning.

Encode Context, Predict Name

Fuse sources of information:

- Tokens around variable uses:

```
var [redacted] = user.GetData();
```

- Name of variable type:

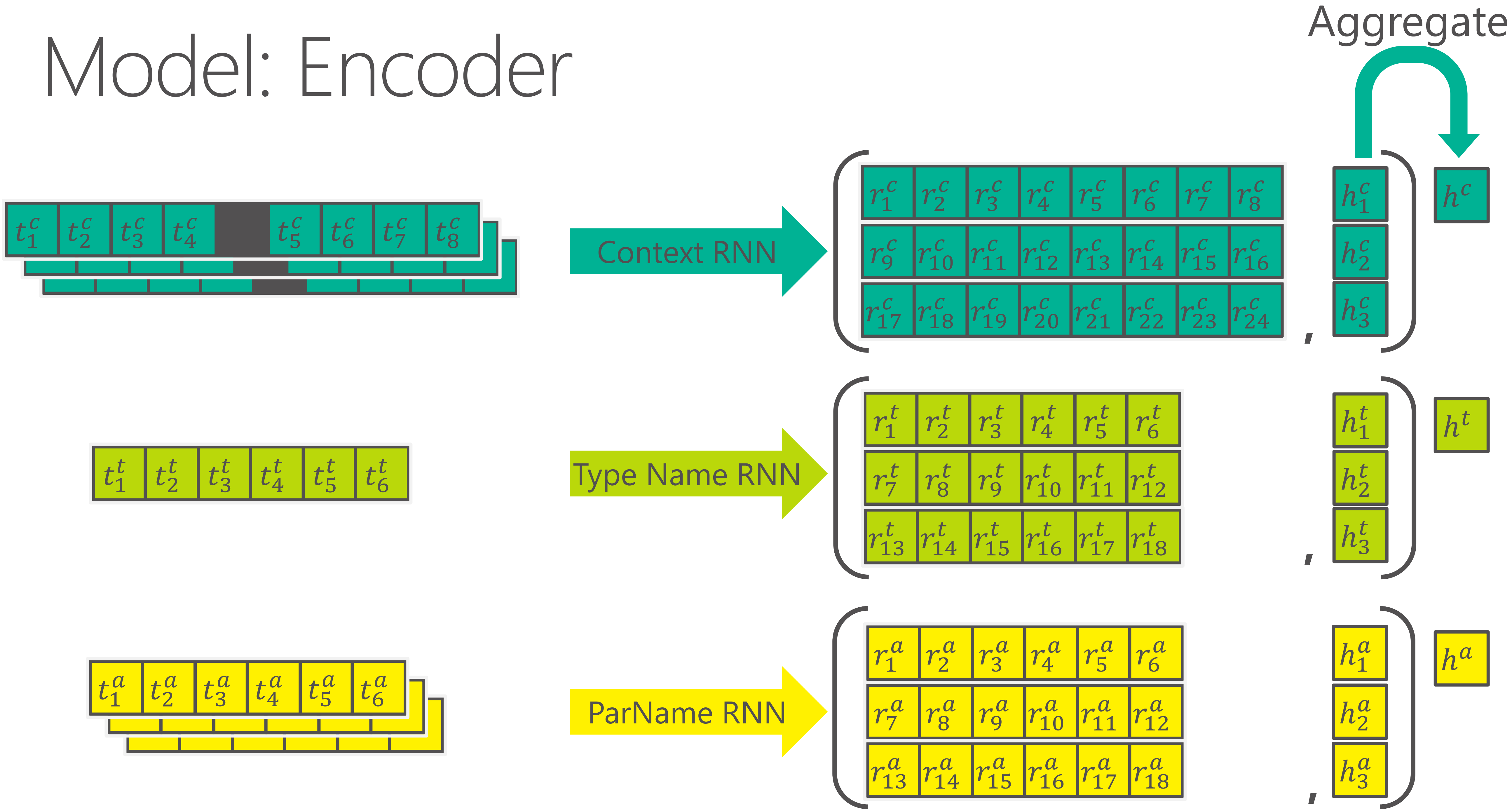
```
InputStream [redacted] = FooBar();
```

- Formal parameters it is matched to:

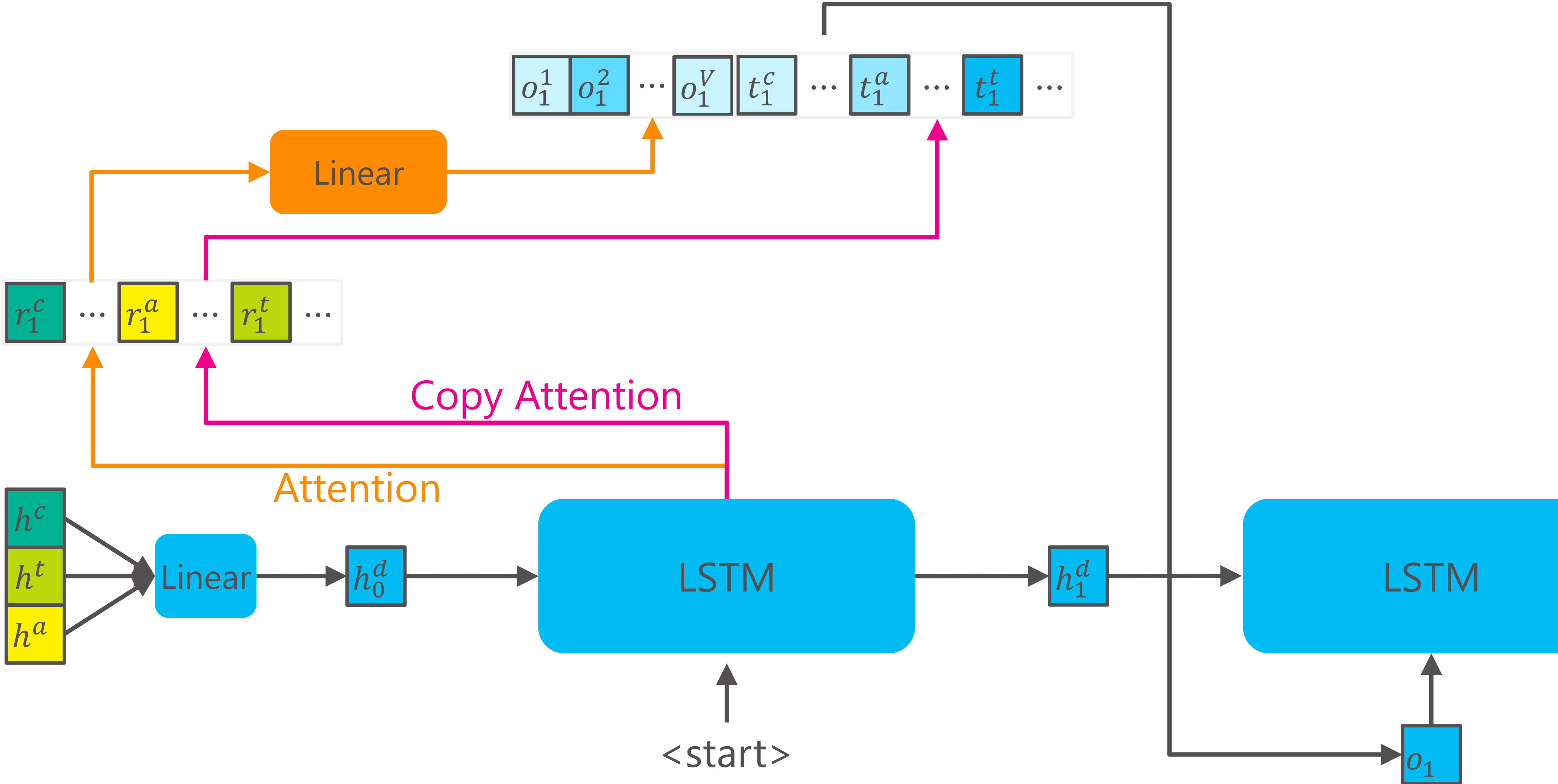
```
foo([redacted]);
…
void foo(int userNum) { … }
```

⟹ Encode Context, Predict Name

Model: Encoder



Model: Decoder



```
}
catch (ErrorResponseException e)
{
    if (e.StatusCode == HttpStatusCode.NotFound)
    {
        var text = e.ResponseString;
        if (text.Contains("maxQueryString"))
            throw new ErrorResponseException(e, text);

tion(e, "There is no index named: " + index);

}

if (HandleException(responseException))
    return null;
```

Model	Suggestions
Suggestions	message (53.87%),

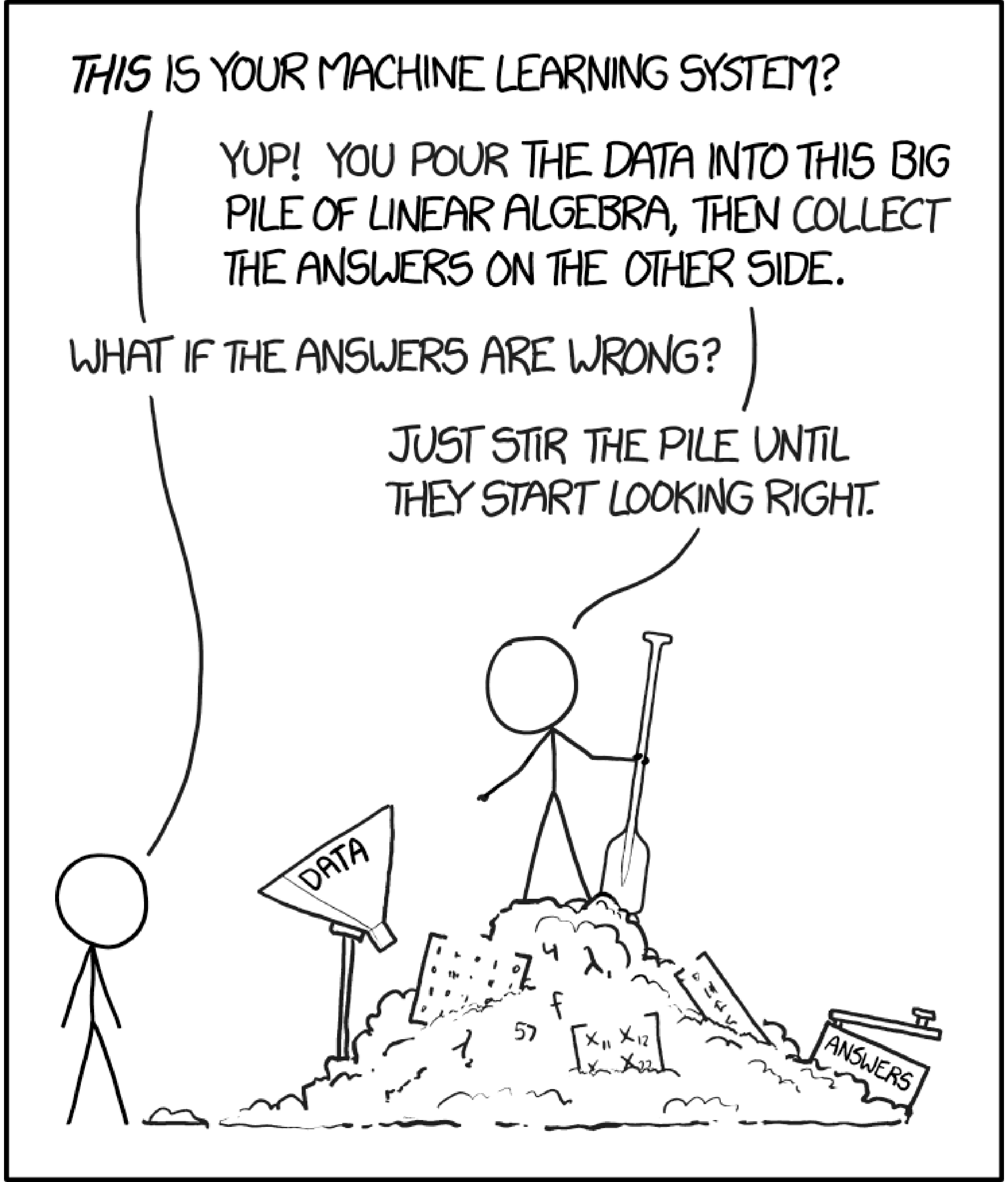
```
var field = document.GetField(Constants.ReduceValueFieldName);
if (field == null)
{
    fieldsToFetch = fieldsToFetch.CloneWith(document.GetFields().Select(x => x.Name).ToArray())
    return base.RetrieveDocument(document, fieldsToFetch, score);
}
var projection = RavenJObject.Parse(field.StringValue);
if (fieldsToFetch.FetchAllStoredFields)
{
    var fields = new HashSet<string>(document.GetFields().Select(x => x.Name));
    fields.Remove(Constants.ReduceKeyFieldName);
    var documentFromFields = new RavenJObject();
    AddFieldsToDocument(document, fields, documentFromFields);
```

Model	Suggestions
Suggestions	fieldNames (64.59%),

```
                                 lue;
}
return new IndexQueryResult
{
    Projection = projection,
    Score = score.Score,
```

Practical Considerations

- Fuse all sources of information
- Metrics
- Train-Use & Feedback Loop
- Low-Resource Environments
- Train/Use Costs
- Confidentiality


THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Practical(?) Tips on Debugging Machine Learning Models

Model Capacity (*what can the model learn?*)

- Overtrain on a small dataset
- Synthetic data

Optimization Issues (*can we make the model learn?*)

- Look at training curves
- Monitor gradient update ratios
- Hand-pick parameters for synthetic data

Other model "bugs" (*is the model doing what I want it to do?*)

- Generate samples from your model (if you can)
- Visualize learned representations (*e.g.* embeddings, nearest neighbors)
- Error analysis (examples where the model is failing, most "confident" errors)
- Simplify the problem/model
- Increase capacity, sweep hyperparameters (*e.g.* increase size of **h** in LSTM)

https://youtu.be/oMB24_ao05A

Learning Semantic Continuous Representations of Symbolic Expressions

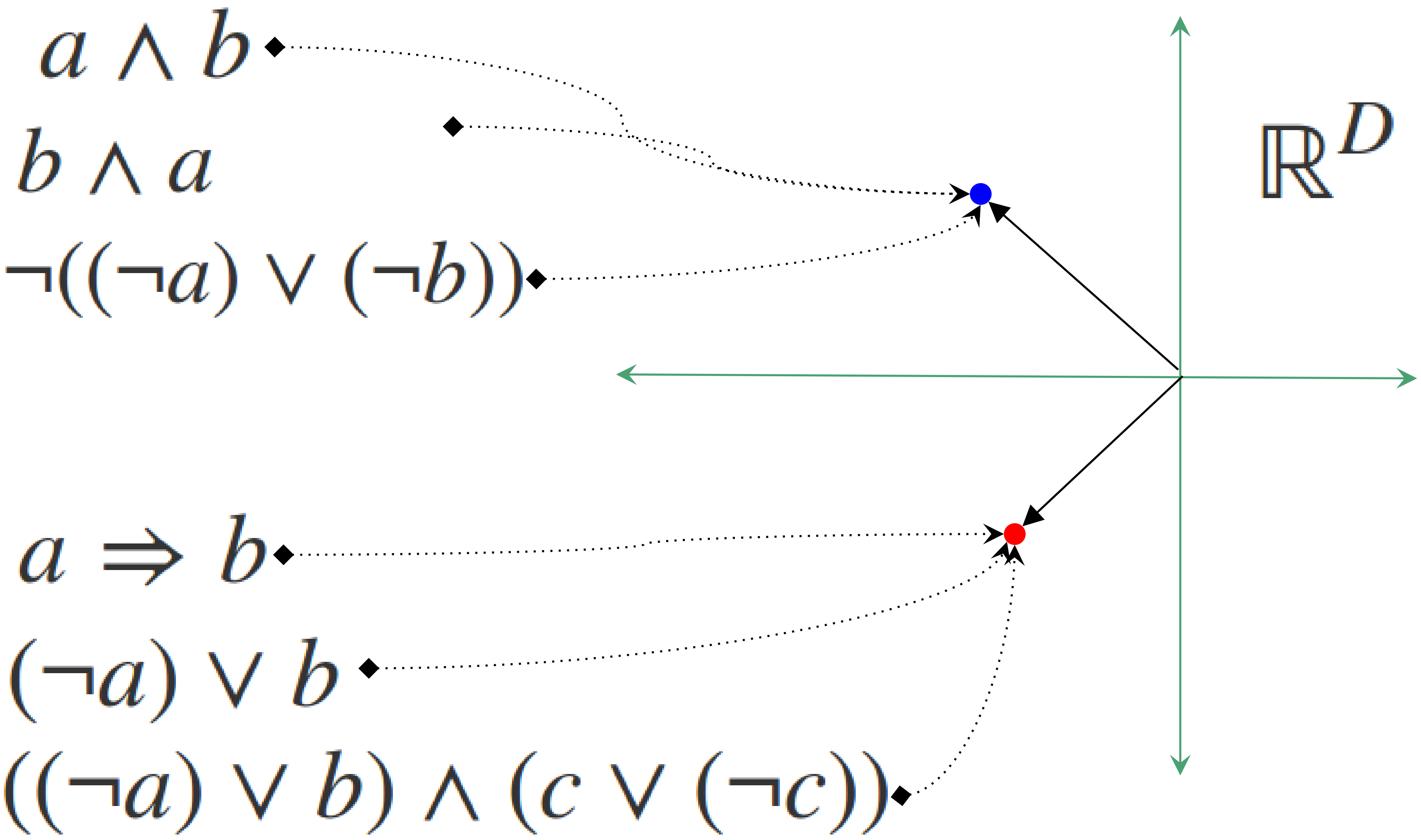
(a-b)*(b+c)+(b-b)

a*b+a*c-b*(b+c)

a*c+b*(a-b-c)

ICML 2017. Joint work with Charles Sutton, Pushmeet Kohli, Pankajan Chanthirasegaran

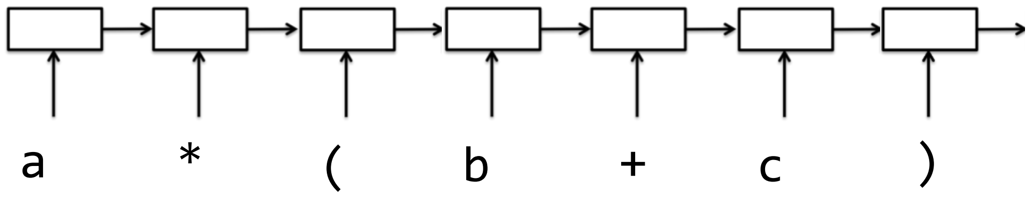
Semantic Continuous Representations



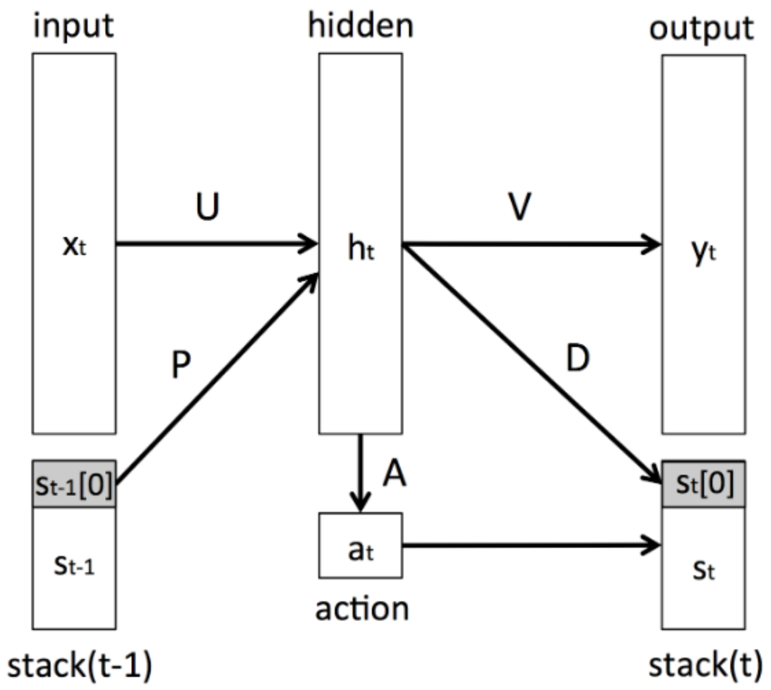
SemVecs

Syntax-driven but **small** changes in syntax can lead in **large** changes in semantics

Sequence RNNs



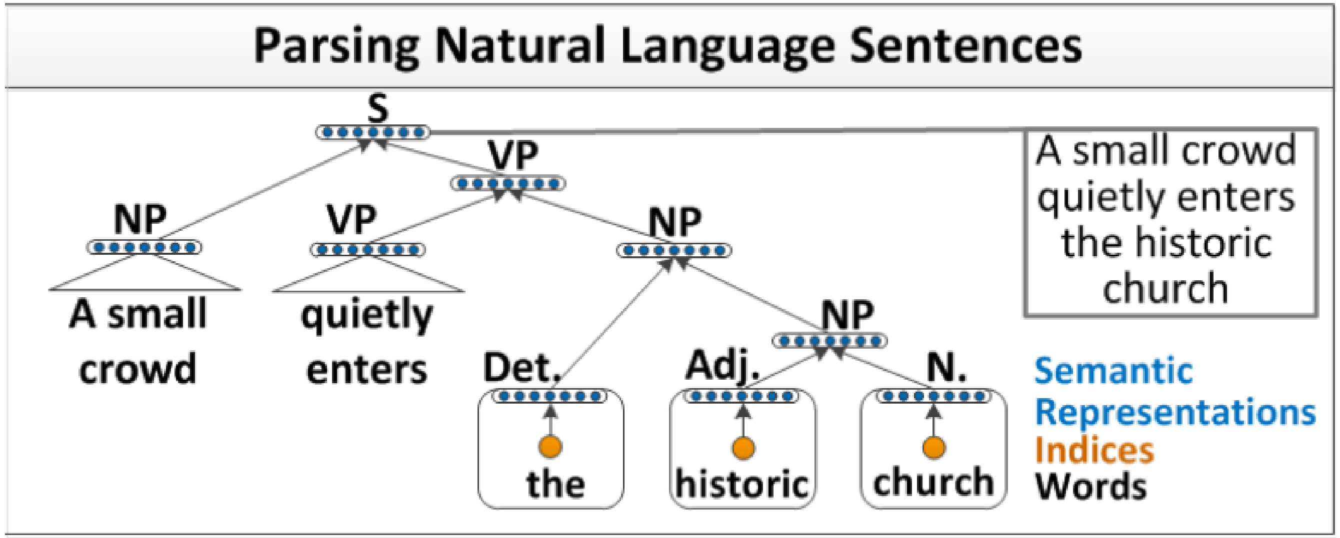
GRU

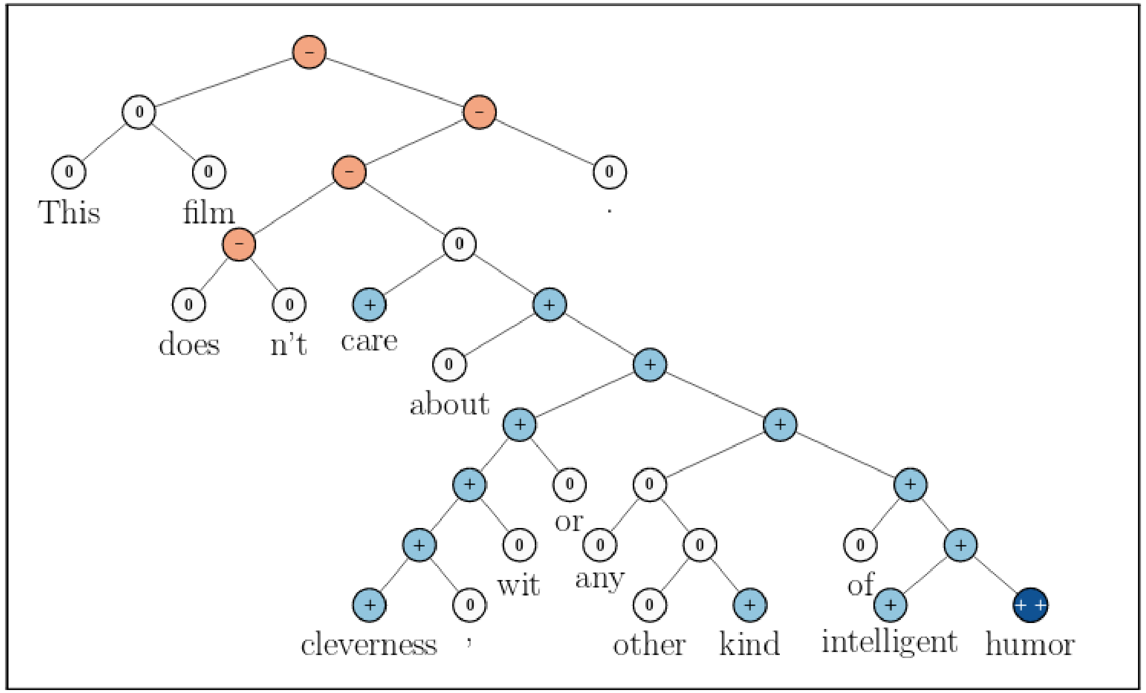


StackRNN

Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation.", 2014
“Inferring Algorithmic Patterns with Stack-Augmented Recurrent Nets” by Armand Joulin and Tomas Mikolov, 2015

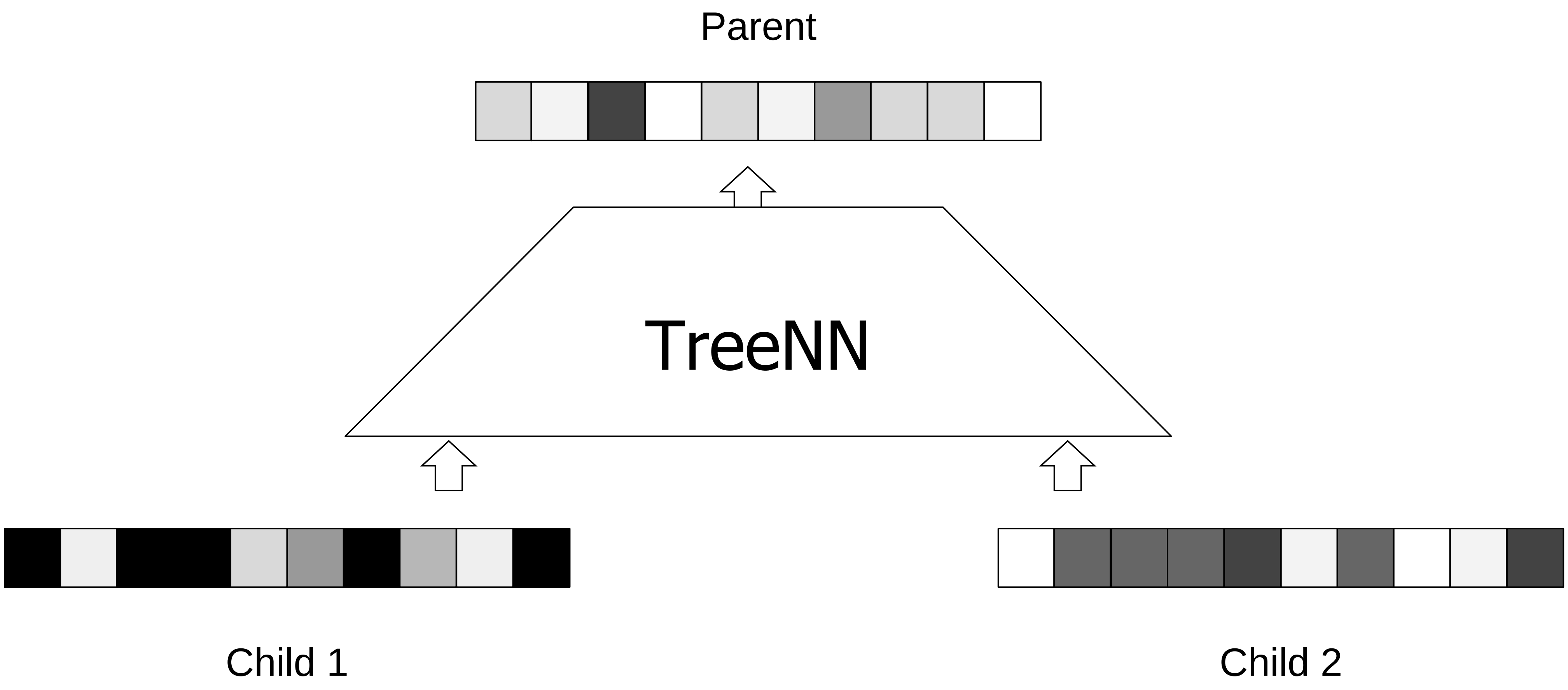
Recursive Neural Networks (TreeNN)



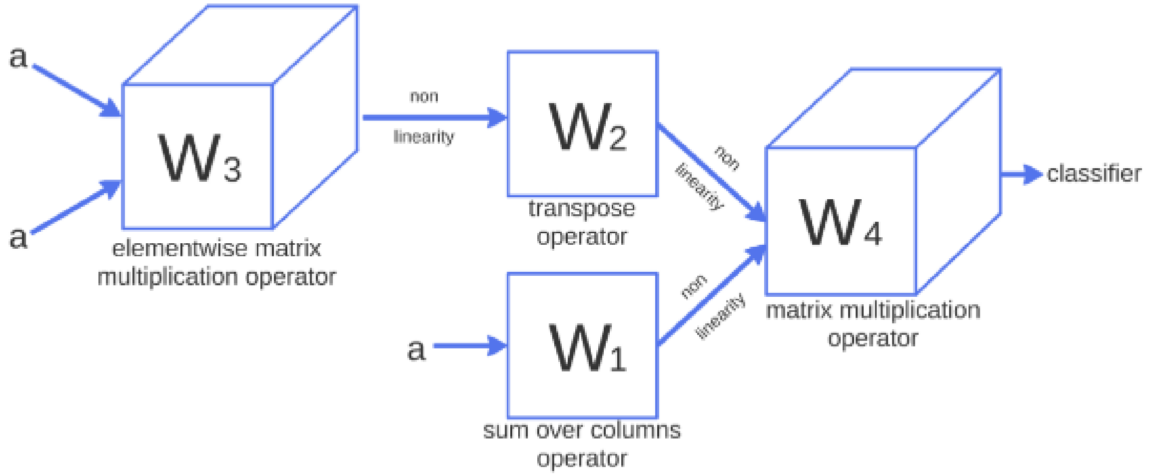


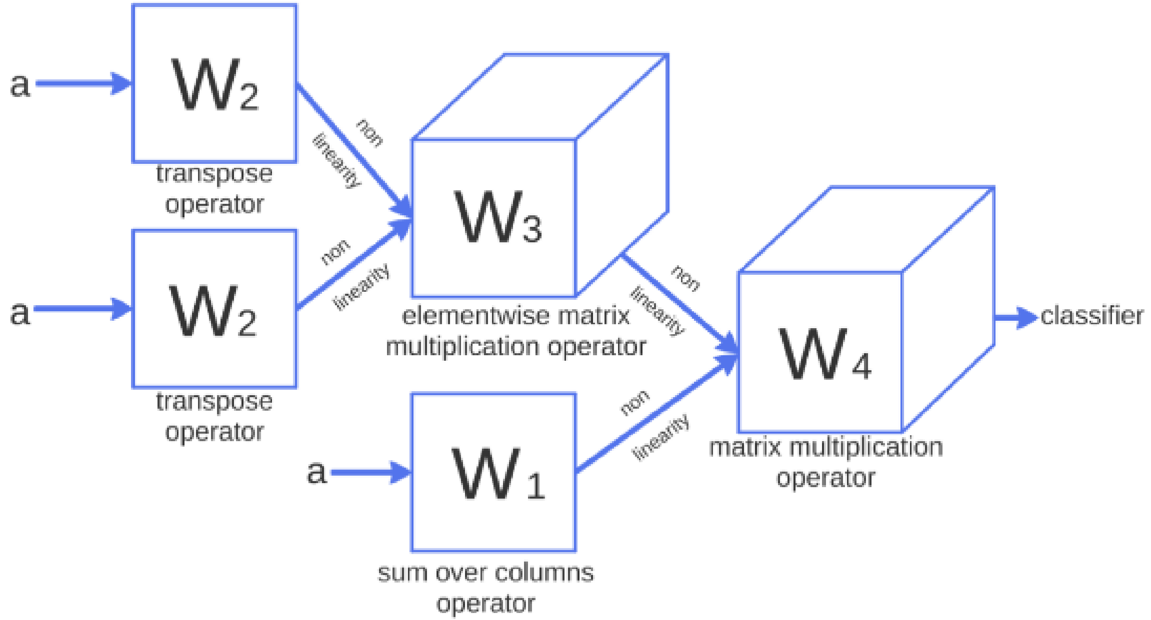
Socher *et al.* 2011, 2013

TreeNNs



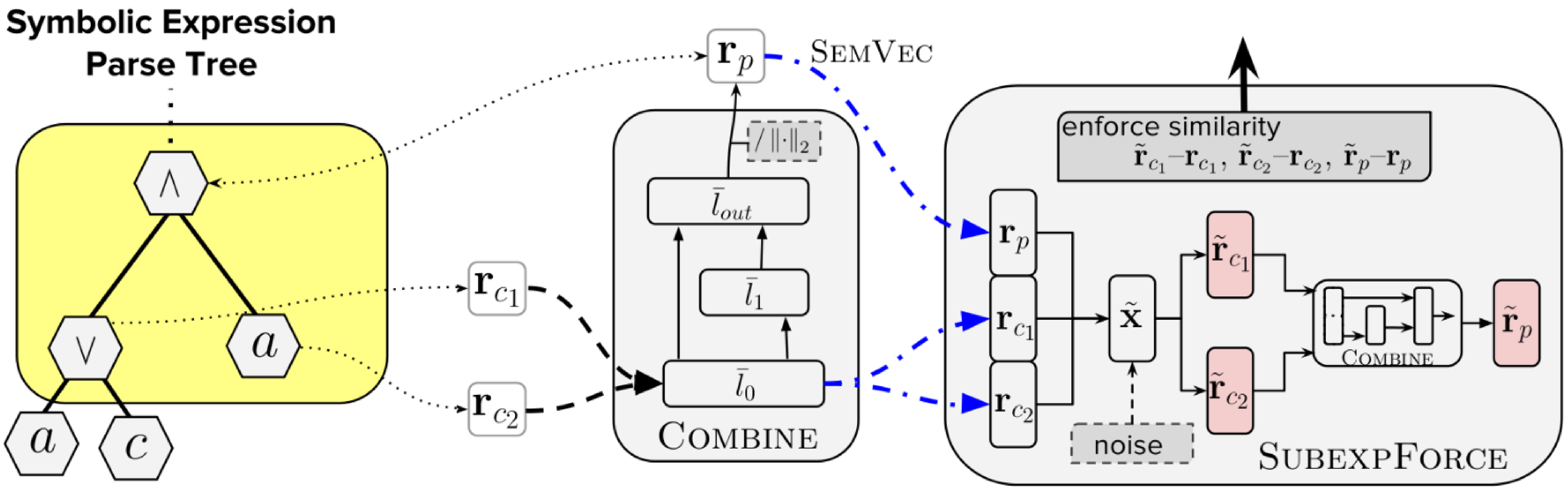
Learning to Discover Efficient Math Identities



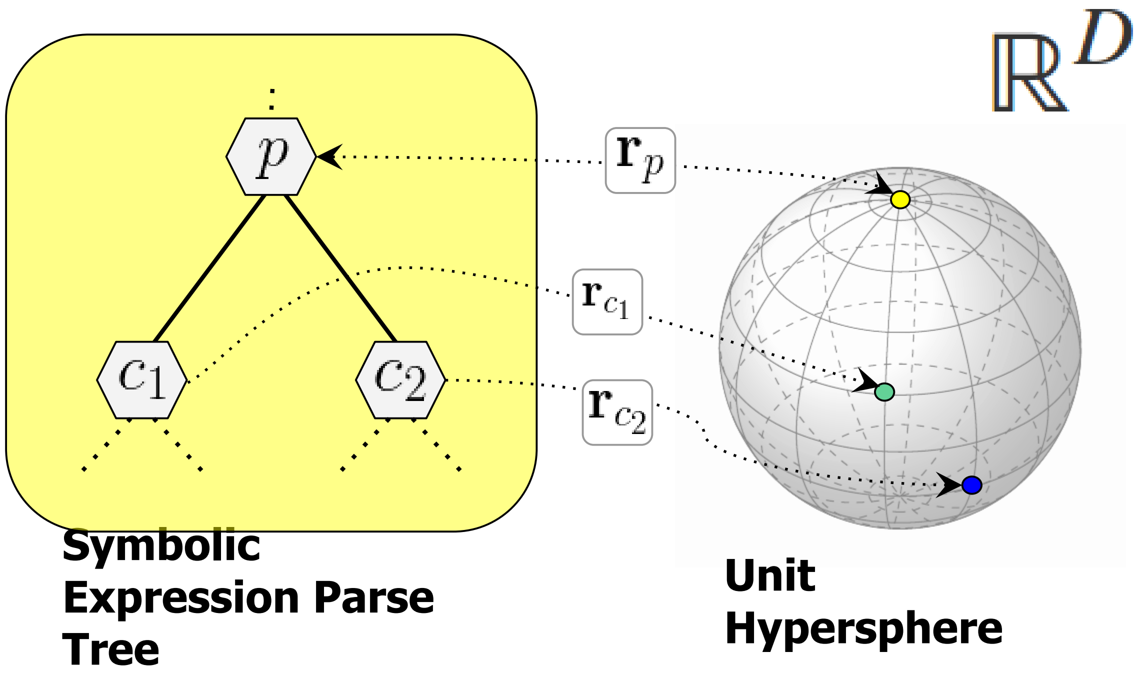


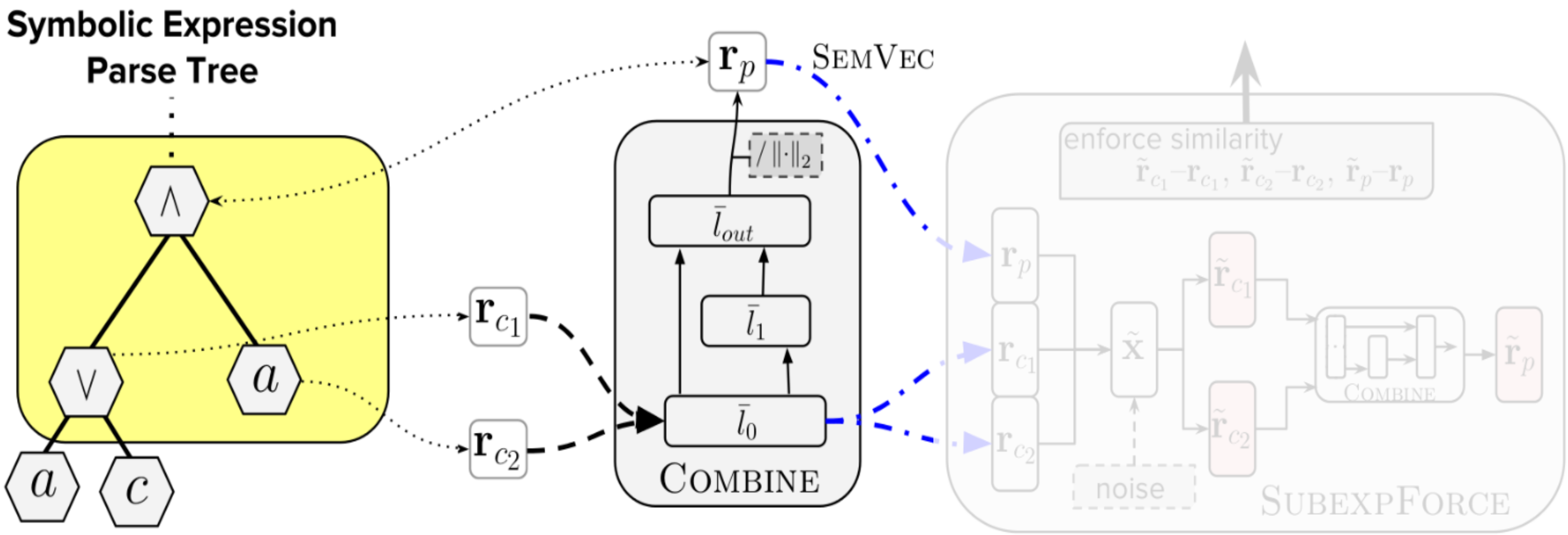
Zaremba *et al.* 2014

EqNet Architecture Overview



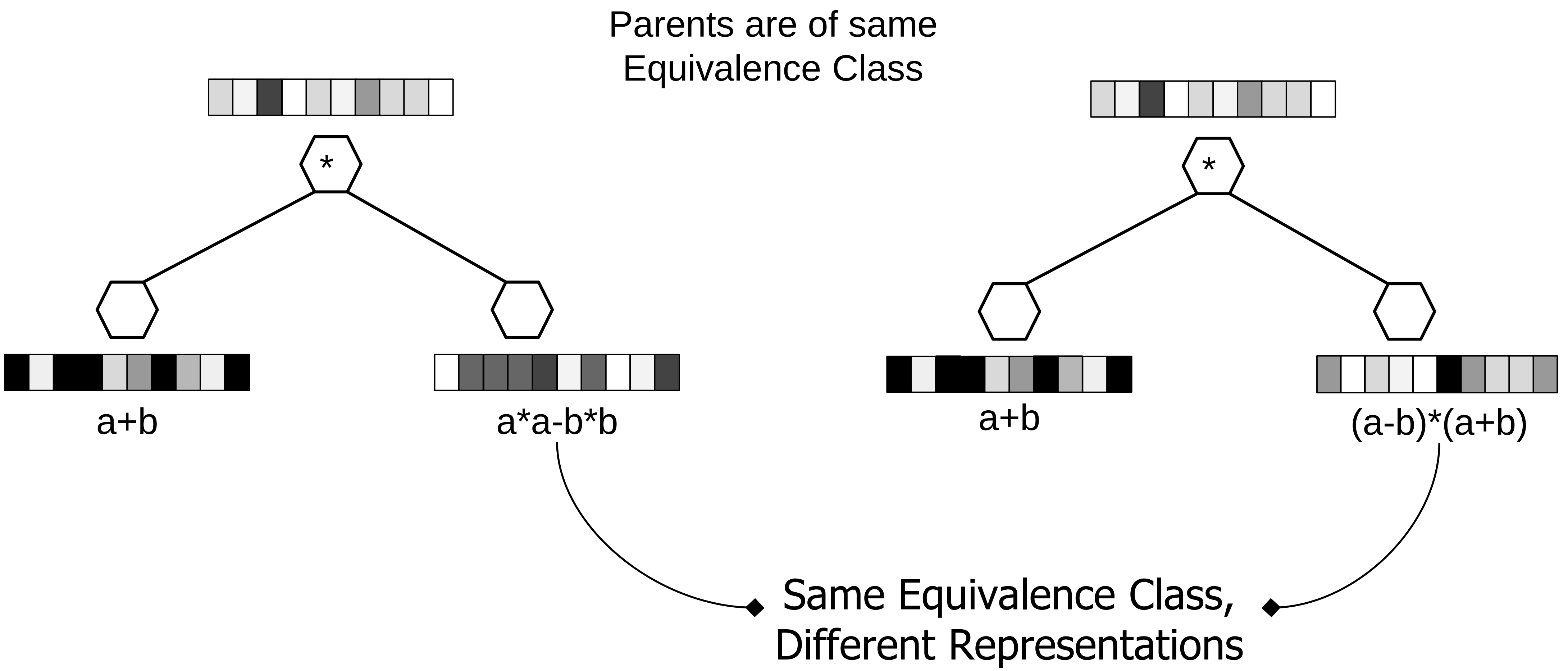
SemVecs in EqNet



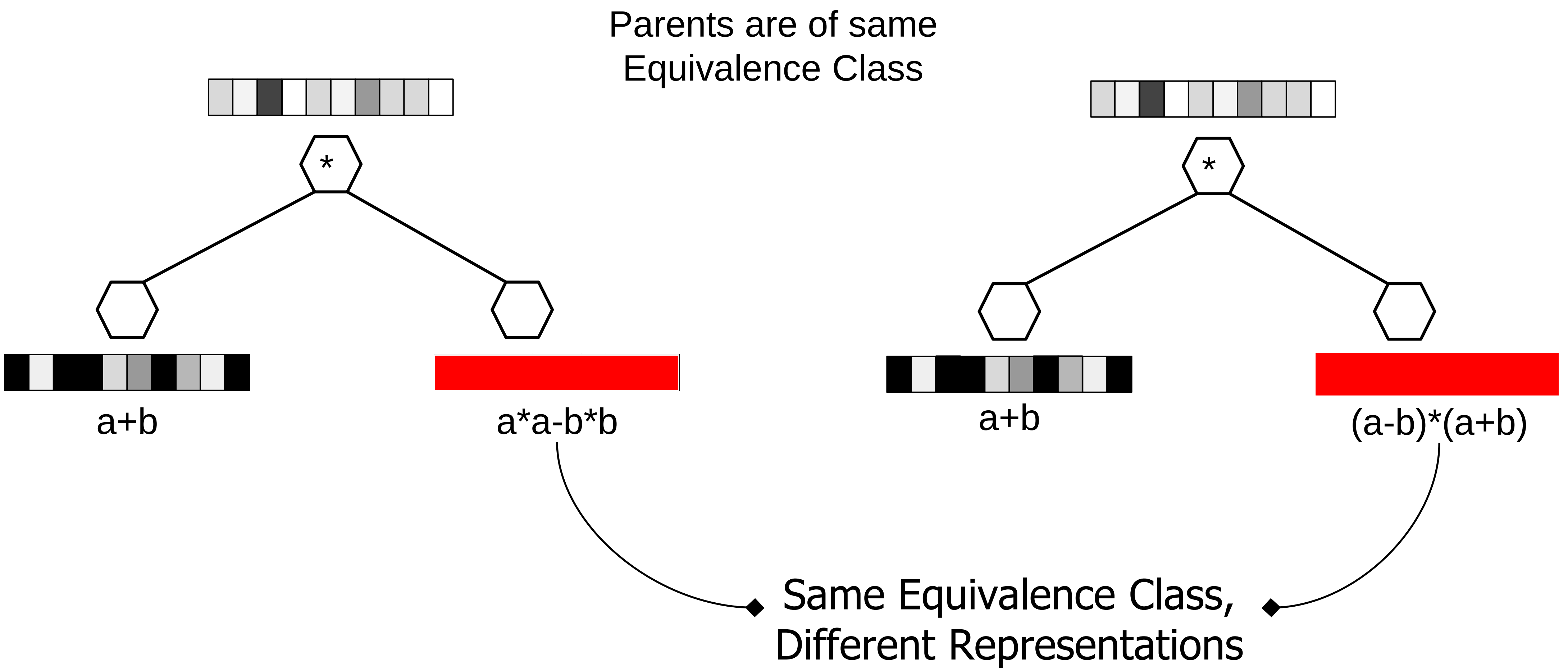


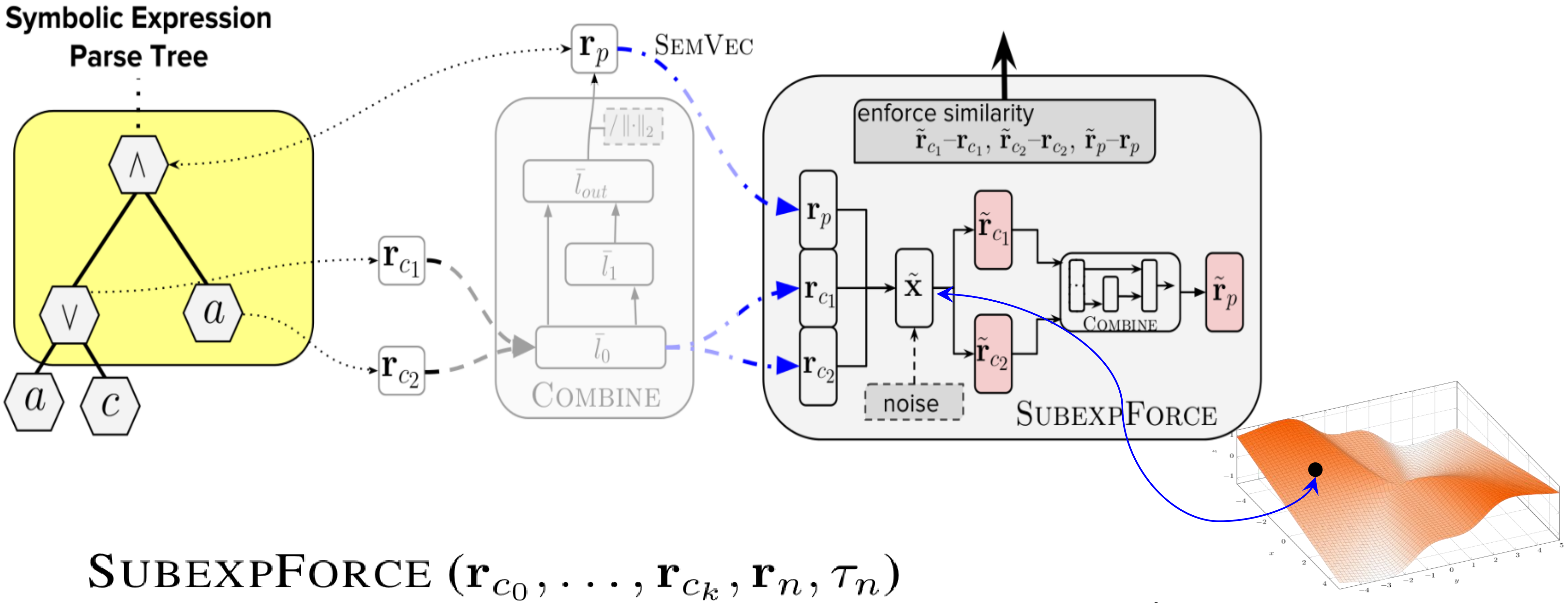
$$
\begin{aligned}
&\textsc{Combine}\left(\mathbf{r}_{c_0}, \ldots, \mathbf{r}_{c_k}, \tau_n\right)\\
&\quad \bar{l}_0 \leftarrow \left[\mathbf{r}_{c_0}, \ldots, \mathbf{r}_{c_k}\right]\\
&\quad \bar{l}_1 \leftarrow \sigma\left(W_{i,\tau_n} \cdot \bar{l}_0\right)\\
&\quad \bar{l}_{out} \leftarrow W_{o0,\tau_n} \cdot \bar{l}_0 + W_{o1,\tau_n} \cdot \bar{l}_1\\
&\quad \textbf{return}\ \ \bar{l}_{out} / \left\|\bar{l}_{out}\right\|_2
\end{aligned}
$$

Subexpression Forcing



Subexpression Forcing





$$\textsc{SubexpForce}\left(\mathbf{r}_{c_0}, \ldots, \mathbf{r}_{c_k}, \mathbf{r}_n, \tau_n\right)$$

$$\mathbf{x} \leftarrow \left[\mathbf{r}_{c_0}, \ldots, \mathbf{r}_{c_k}\right]$$

noise

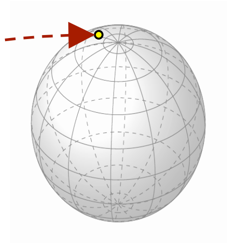
$$\tilde{\mathbf{x}} \leftarrow \tanh\left(W_d \cdot \tanh\left(W_{e,\tau_n} \cdot [\mathbf{r}_n, \mathbf{x}] \cdot \mathbf{n}\right)\right)$$

$$\tilde{\mathbf{x}} \leftarrow \tilde{\mathbf{x}} \cdot \|\mathbf{x}\|_2 \,/\, \|\tilde{\mathbf{x}}\|_2$$

$$\tilde{\mathbf{r}}_n \leftarrow \textsc{Combine}(\tilde{\mathbf{x}}, \tau_n)$$

$$\textbf{return} \ -\left(\tilde{\mathbf{x}}^\top \mathbf{x} + \tilde{\mathbf{r}}_n^\top \mathbf{r}_n\right)$$

Training Objective



$$P(e_i|T) = \frac{\exp\left(\boxed{\textsc{TreeNN}(T)}^\top \mathbf{q}_{e_i} + b_i\right)}{\sum_j \exp\left(\textsc{TreeNN}(T)^\top \mathbf{q}_{e_j} + b_j\right)}$$

Training Objective

$$\mathcal{L}_{\text{ACC}}(T, e_i) = \max\left(0, \underset{e_j \neq e_i, e_j \in \mathcal{E}}{\arg\max} \log P(e_j|T) - \log P(e_i|T) + m\right)$$

$$\mathcal{L}(T, e_i) = \mathcal{L}_{\text{ACC}}(T, e_i) + \frac{\mu}{|Q|} \sum_{n \in Q} \textsc{SubexpForce}(\text{ch}(n), n)$$

scheduled introduction

Datasets

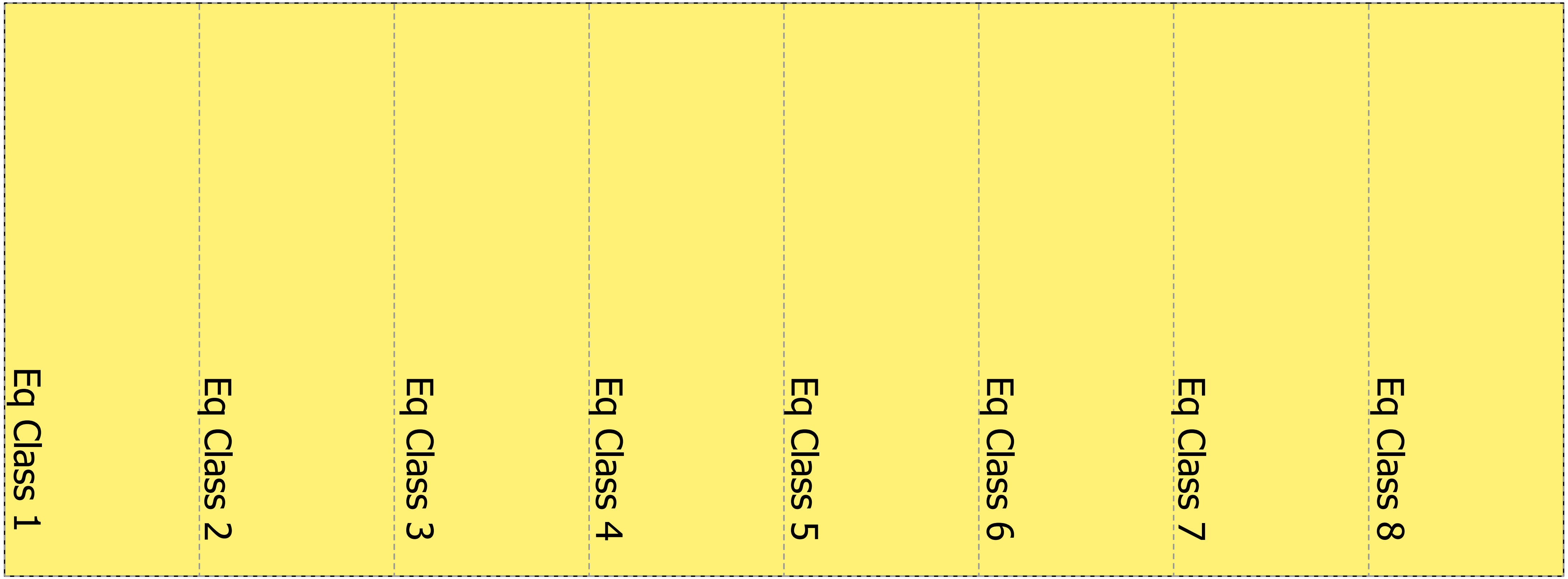
Dataset	# Vars	# Equiv Classes	# Exprs	H
SIMPBOOL8	3	120	39,048	5.6
SIMPBOOL10^S	3	191	26,304	7.2
BOOL5	3	95	1,239	5.6
BOOL8	3	232	257,784	6.2
BOOL10^S	10	256	51,299	8.0
SIMPBOOLL5	10	1,342	10,050	9.9
BOOLL5	10	7,312	36,050	11.8
SIMPPOLY5	3	47	237	5.0
SIMPPOLY8	3	104	3,477	5.8
SIMPPOLY10	3	195	57,909	6.3
ONEV-POLY10	1	83	1,291	5.4
ONEV-POLY13	1	677	107,725	7.1
POLY5	3	150	516	6.7
POLY8	3	1,102	11,451	9.0

Bool8

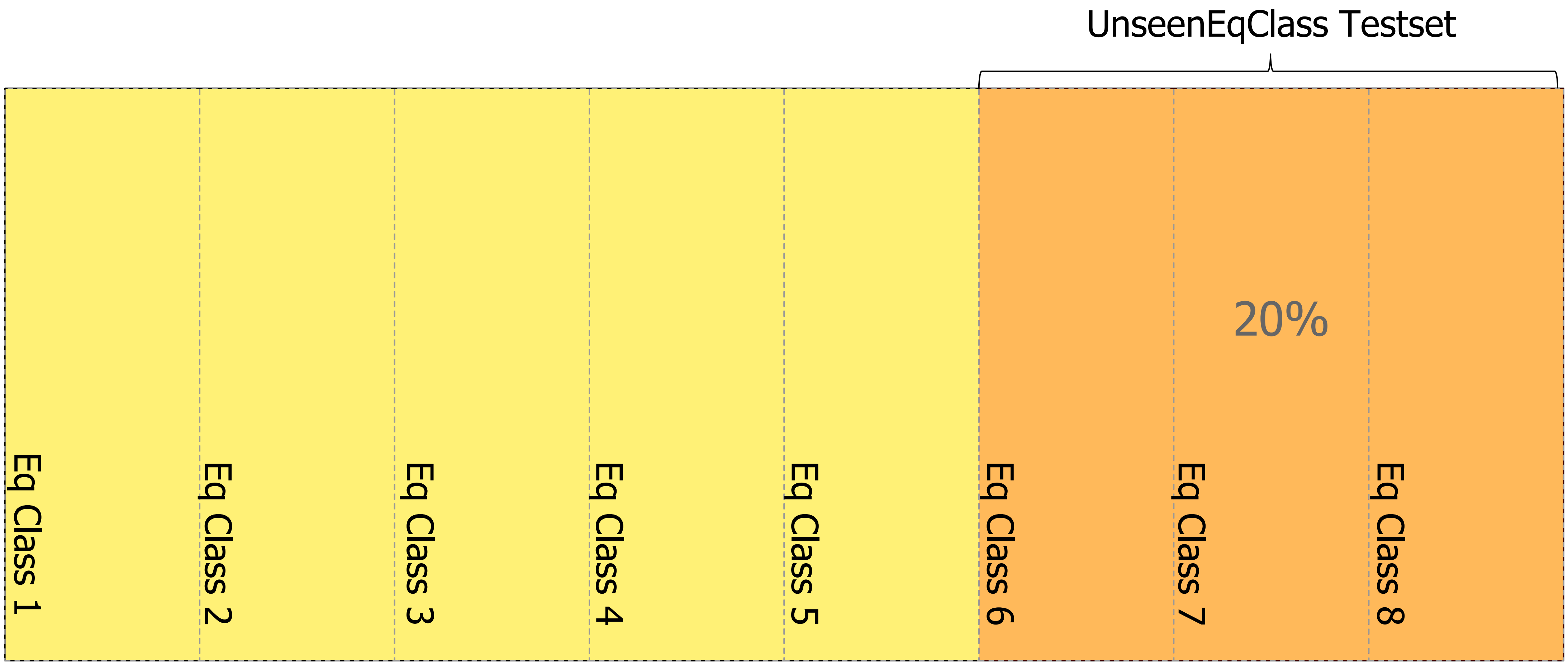
$(\neg a) \wedge (\neg b)$	$(\neg a \wedge \neg c) \vee (\neg b \wedge a \wedge c) \vee (\neg c \wedge b)$	$(\neg a) \wedge b \wedge c$
$a\neg((\neg a) \Rightarrow ((\neg a) \wedge b))$	$c \oplus (((\neg a) \Rightarrow a) \Rightarrow b)$	$\neg((\neg b) \vee ((\neg c) \vee a))$
$\neg((b \vee (\neg(\neg a))) \vee b)$	$\neg((b \oplus (b \vee a)) \oplus c)$	$((a \vee b) \wedge c) \wedge (\neg a)$
$(\neg a) \oplus ((a \vee b) \oplus a)$	$\neg((\neg(b \vee (\neg a))) \oplus c)$	$(\neg((\neg(\neg b)) \Rightarrow a)) \wedge c$
$(b \Rightarrow (b \Rightarrow a)) \wedge (\neg a)$	$((b \vee a) \oplus (\neg b)) \oplus c)$	$(c \wedge (c \Rightarrow (\neg a))) \wedge b$
$((\neg a) \Rightarrow b) \Rightarrow (a \oplus a)$	$(\neg((b \oplus a) \wedge a)) \oplus c$	$b \wedge (\neg(b \wedge (c \Rightarrow a)))$

False	$(\neg a) \wedge (\neg b) \vee (\wedge c)$	$\neg a \vee b$
$(a \oplus a) \wedge (c \Rightarrow c)$	$(a \Rightarrow (\neg c)) \oplus (a \vee b)$	$a \Rightarrow ((b \wedge (\neg c)) \vee b)$
$(\neg b) \wedge (\neg(b \Rightarrow a))$	$(a \Rightarrow (c \oplus b)) \oplus b$	$\neg(\neg((b \vee a) \Rightarrow b))$
$b \wedge ((a \vee a) \oplus a)$	$b \oplus (a \Rightarrow (b \oplus c))$	$(\neg a) \oplus (\neg(b \Rightarrow (\neg a)))$
$((\neg b) \wedge b) \oplus (a \oplus a)$	$(b \vee a) \oplus (x \Rightarrow (\neg a))$	$b \vee (\neg((\neg b) \wedge a))$
$c \wedge ((\neg(a \Rightarrow a)) \wedge c)$	$b \oplus ((\neg a) \vee (c \oplus b))$	$\neg((a \Rightarrow (a \oplus b)) \wedge a)$

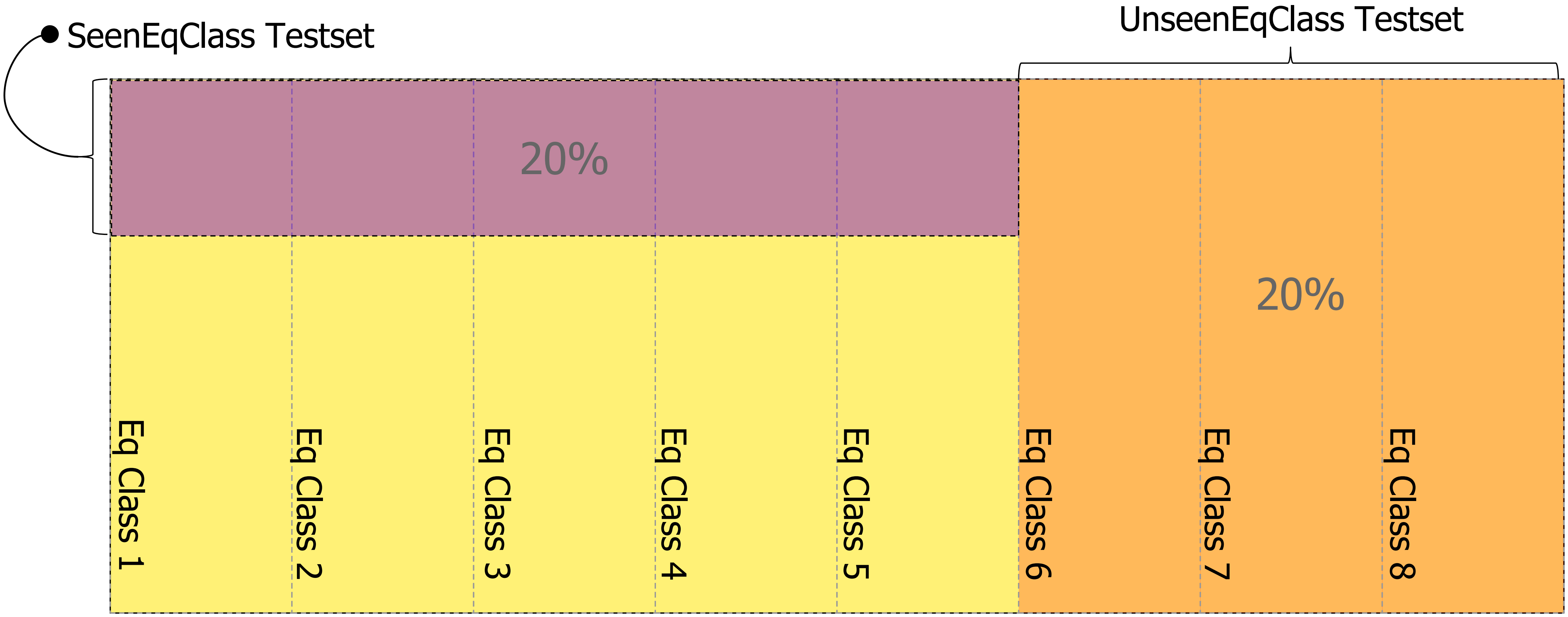
Test Datasets



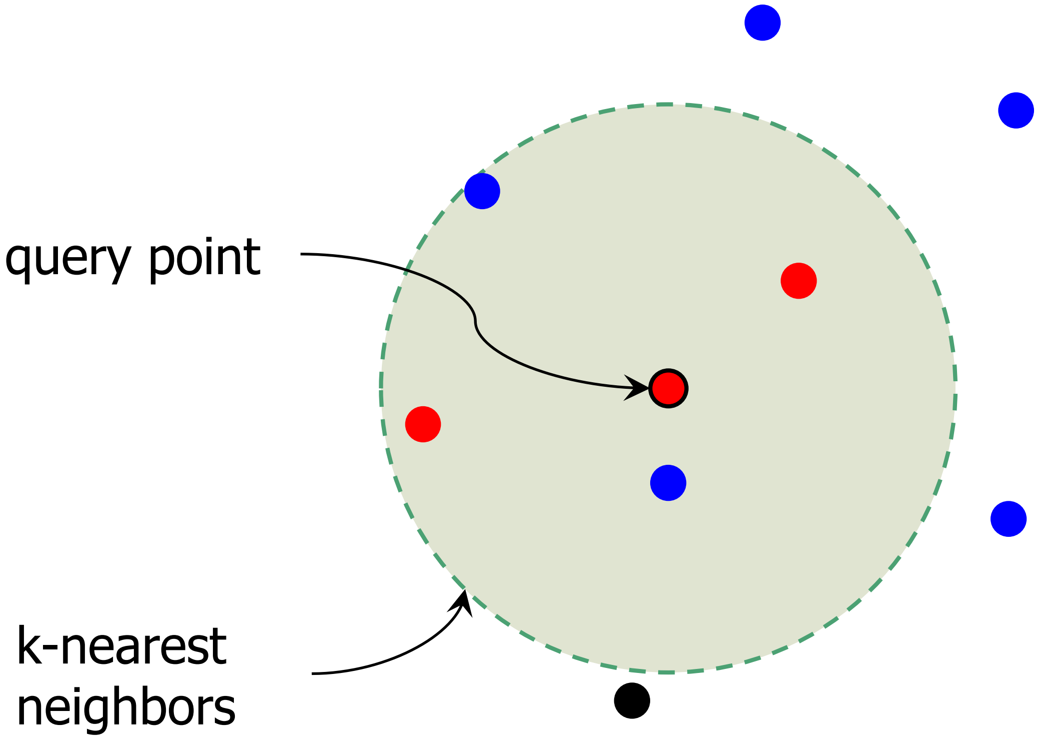
Test Datasets



Test Datasets

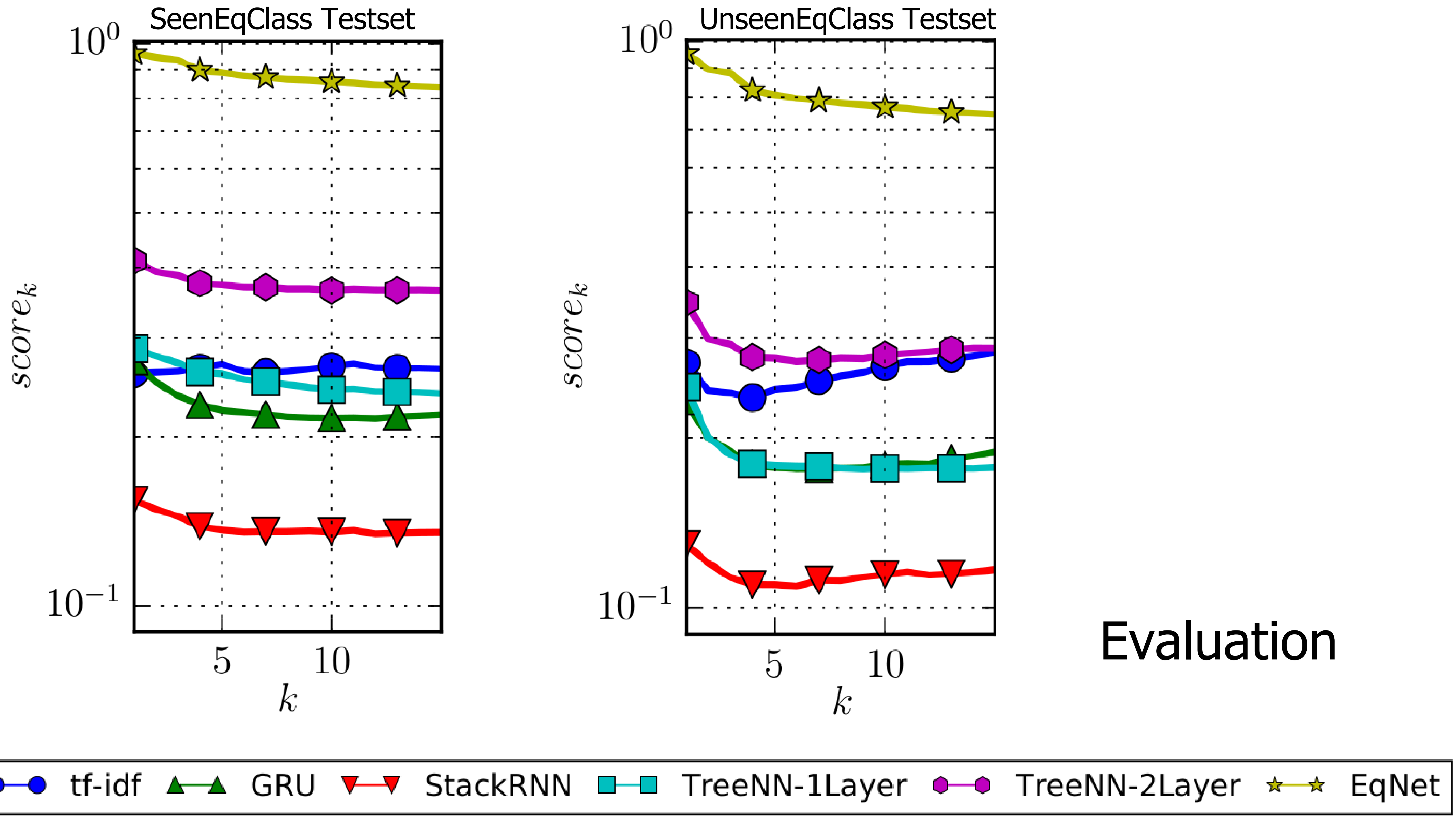


Evaluation Metric

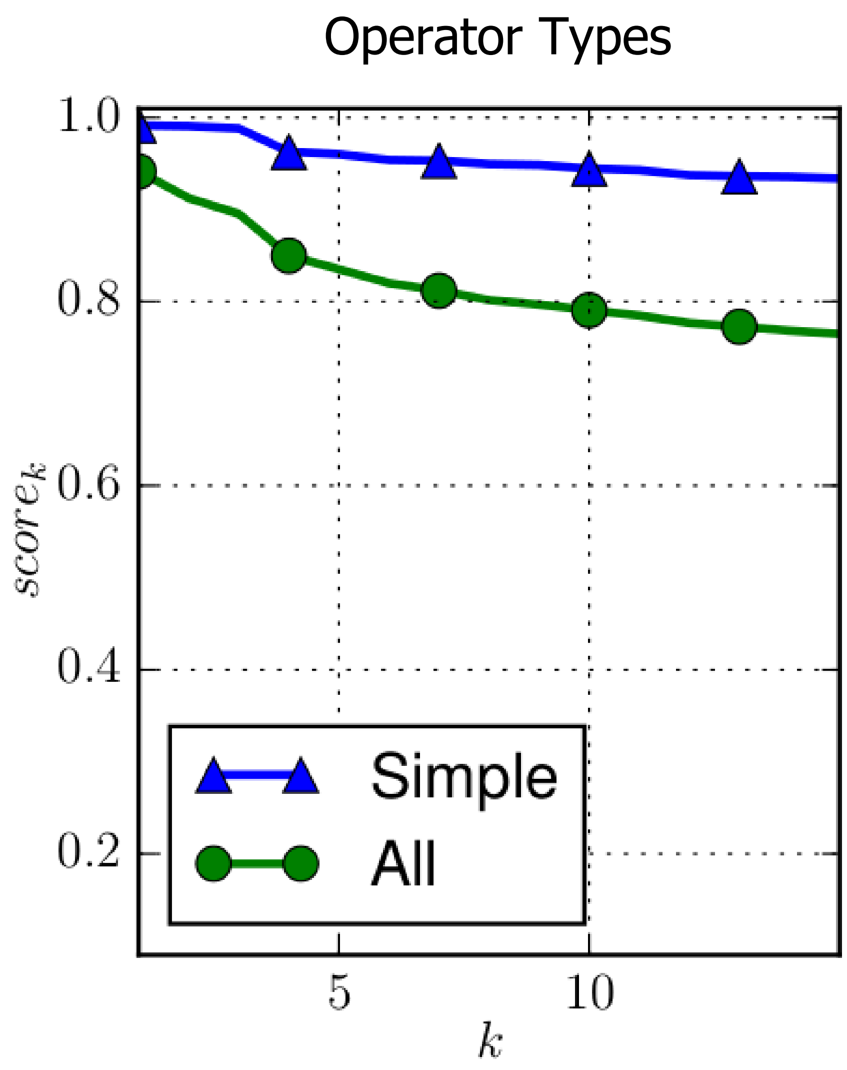


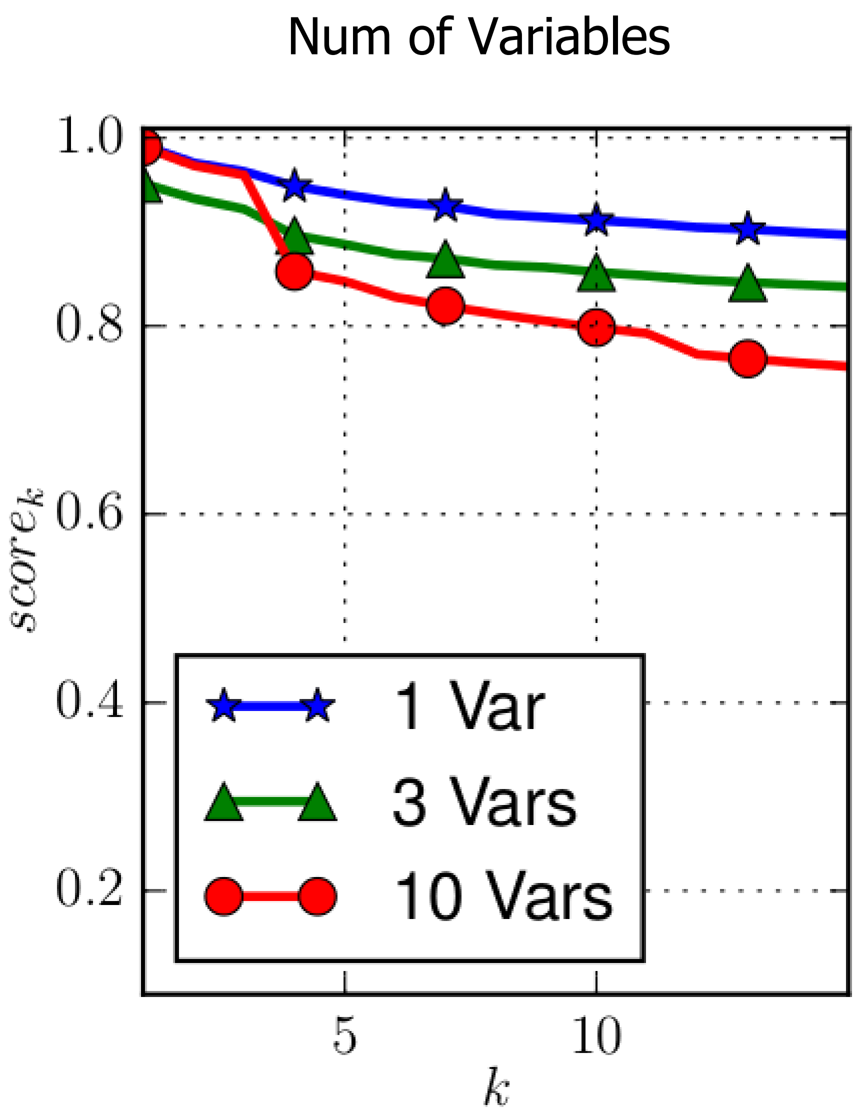
$$score_k(q) = \frac{|\mathbb{N}_k(q) \cap c|}{\min(k, |c|)}$$

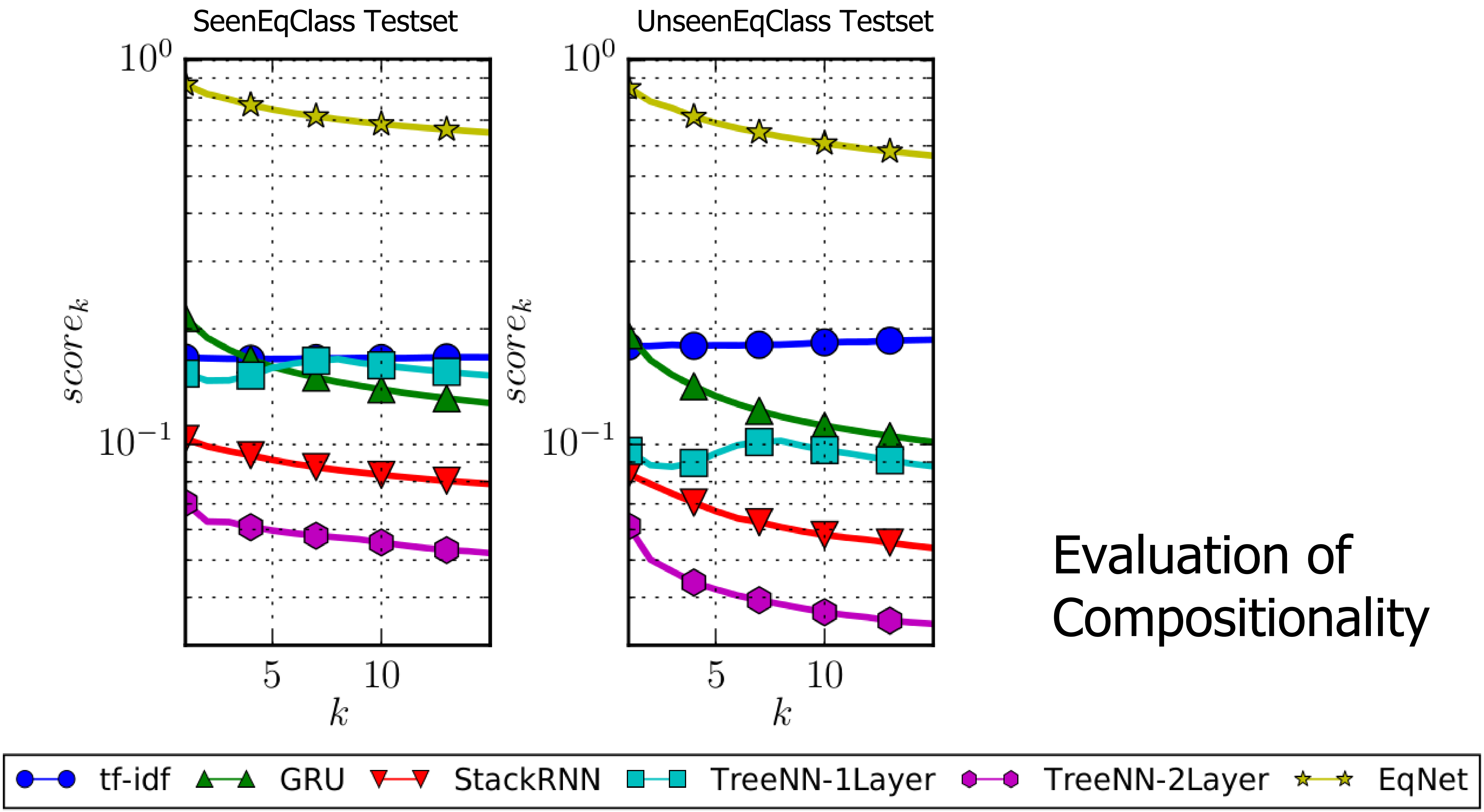
Evaluation


SeenEqClass Testset
UnseenEqClass Testset
10^0
10^{-1}
$score_k$
5
10
k
tf-idf
GRU
StackRNN
TreeNN-1Layer
TreeNN-2Layer
EqNet

EqNet Performance *vs* Dataset Characteristics

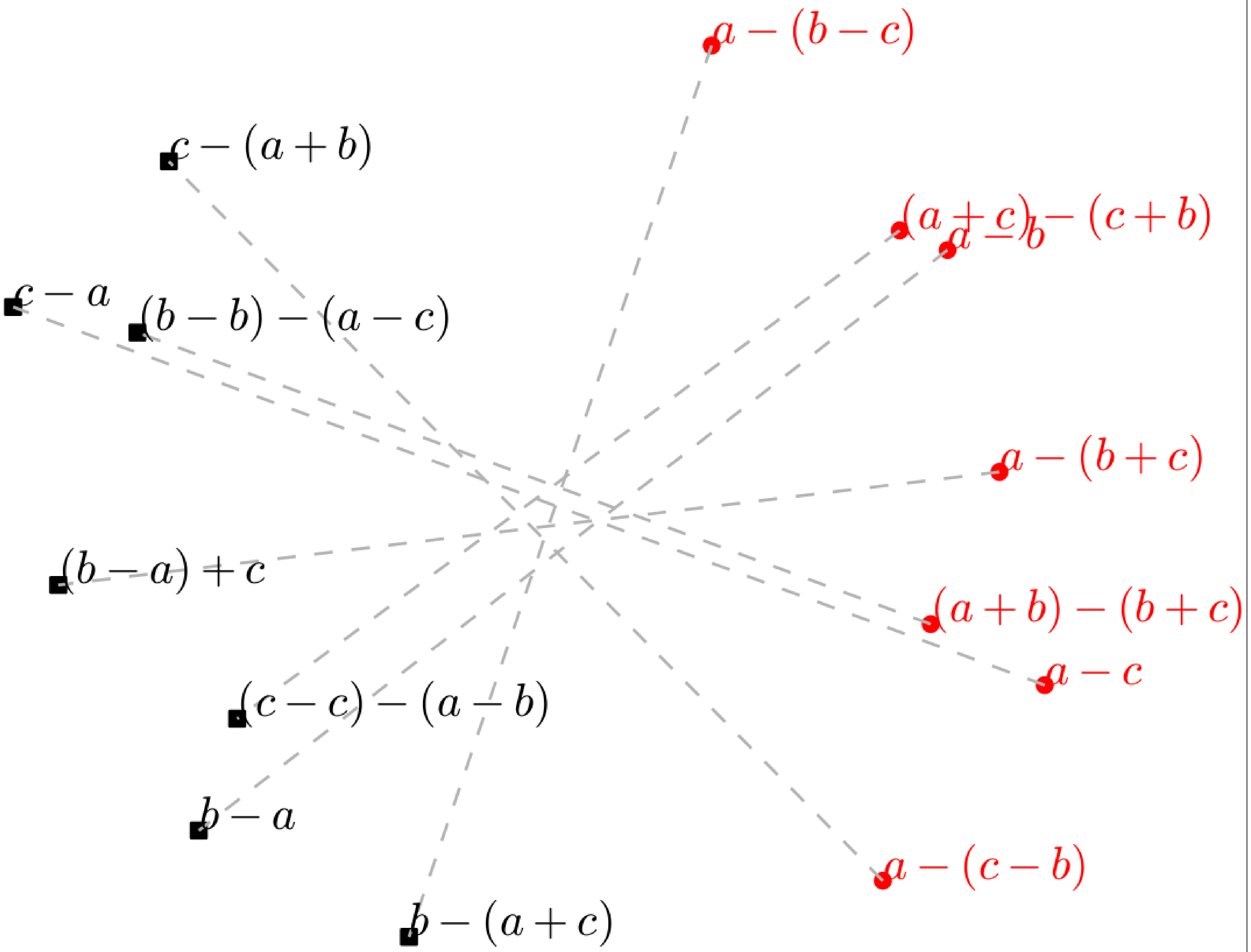




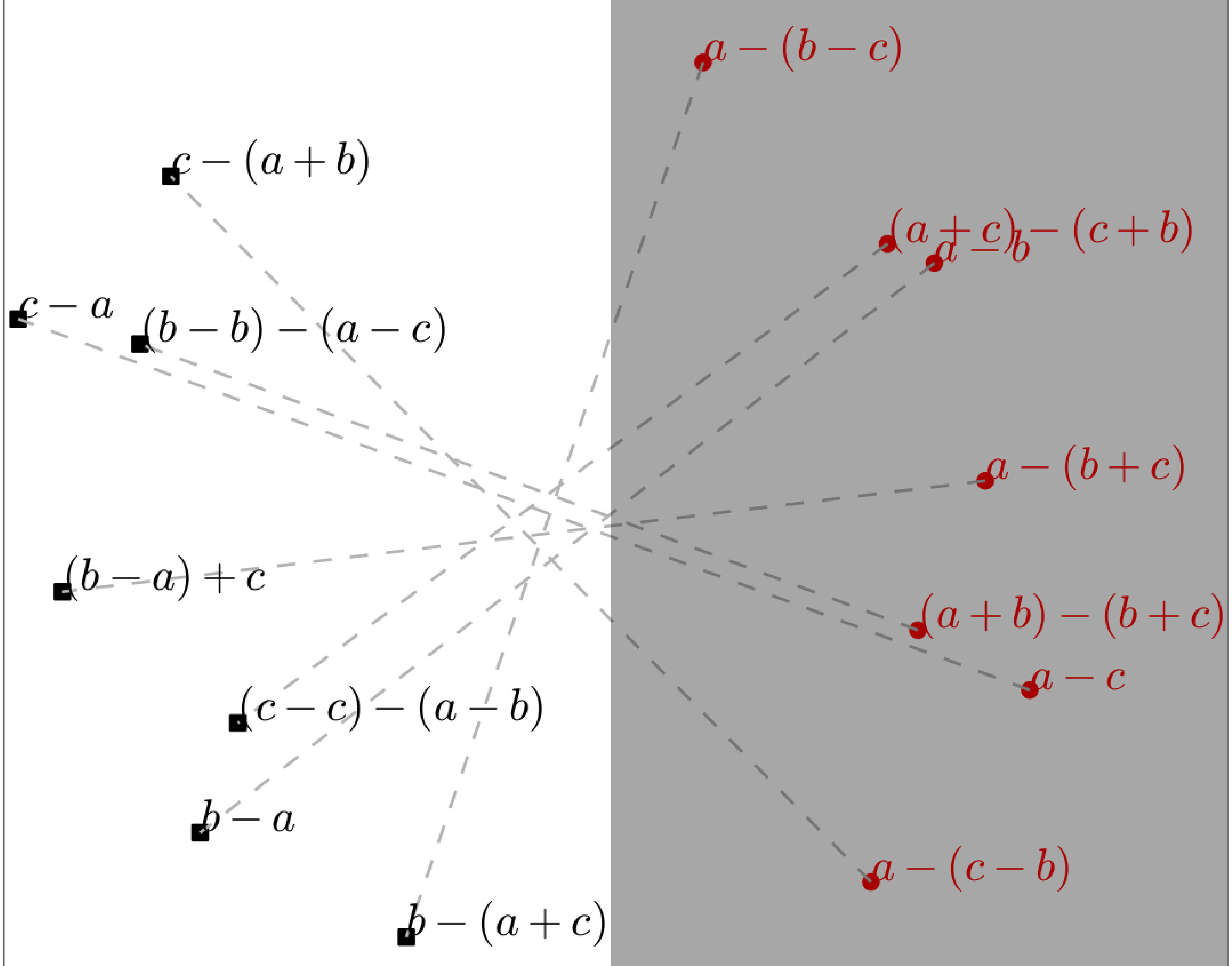


Evaluation of Compositionality

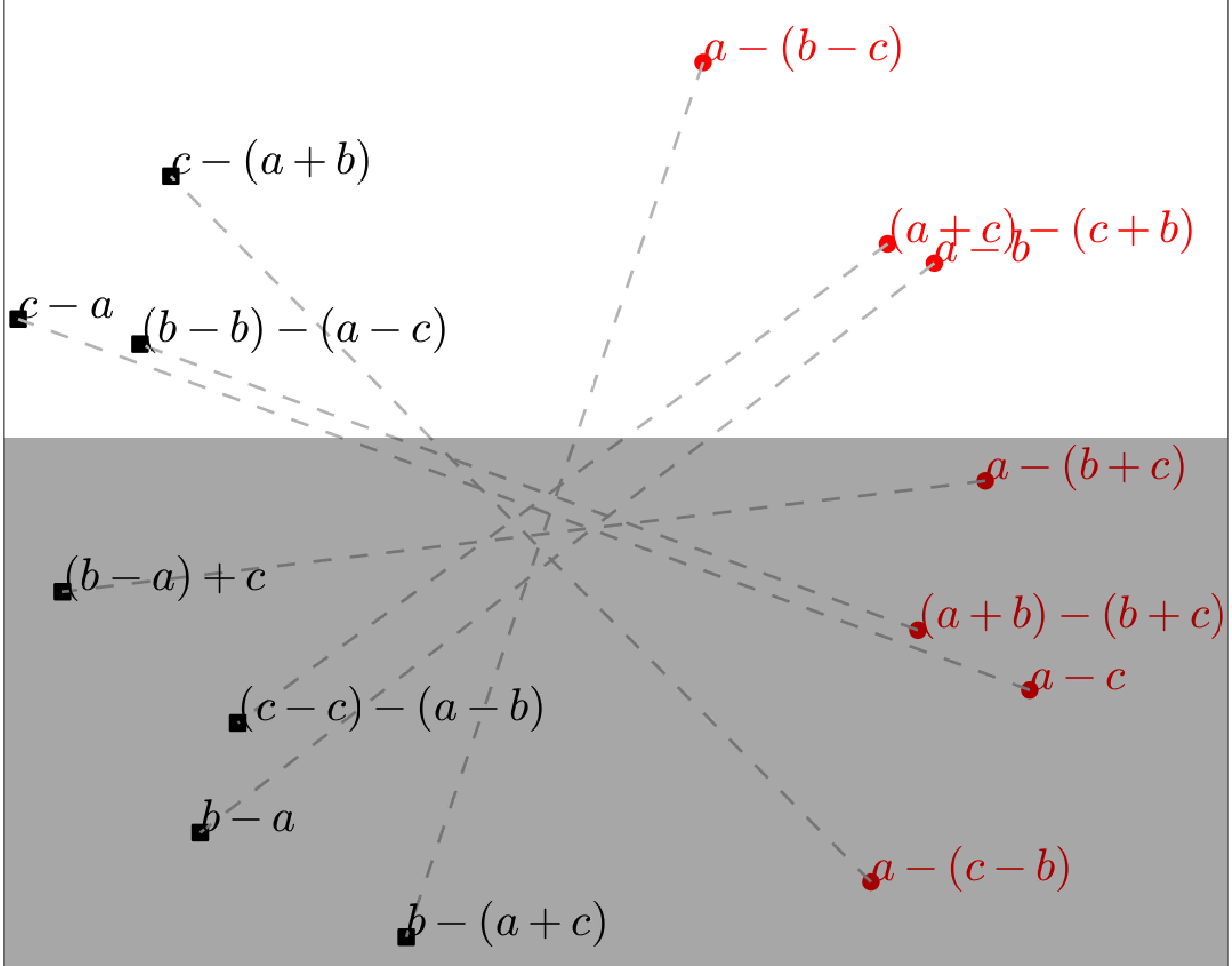
Visualizing Polynomials and their Negatives



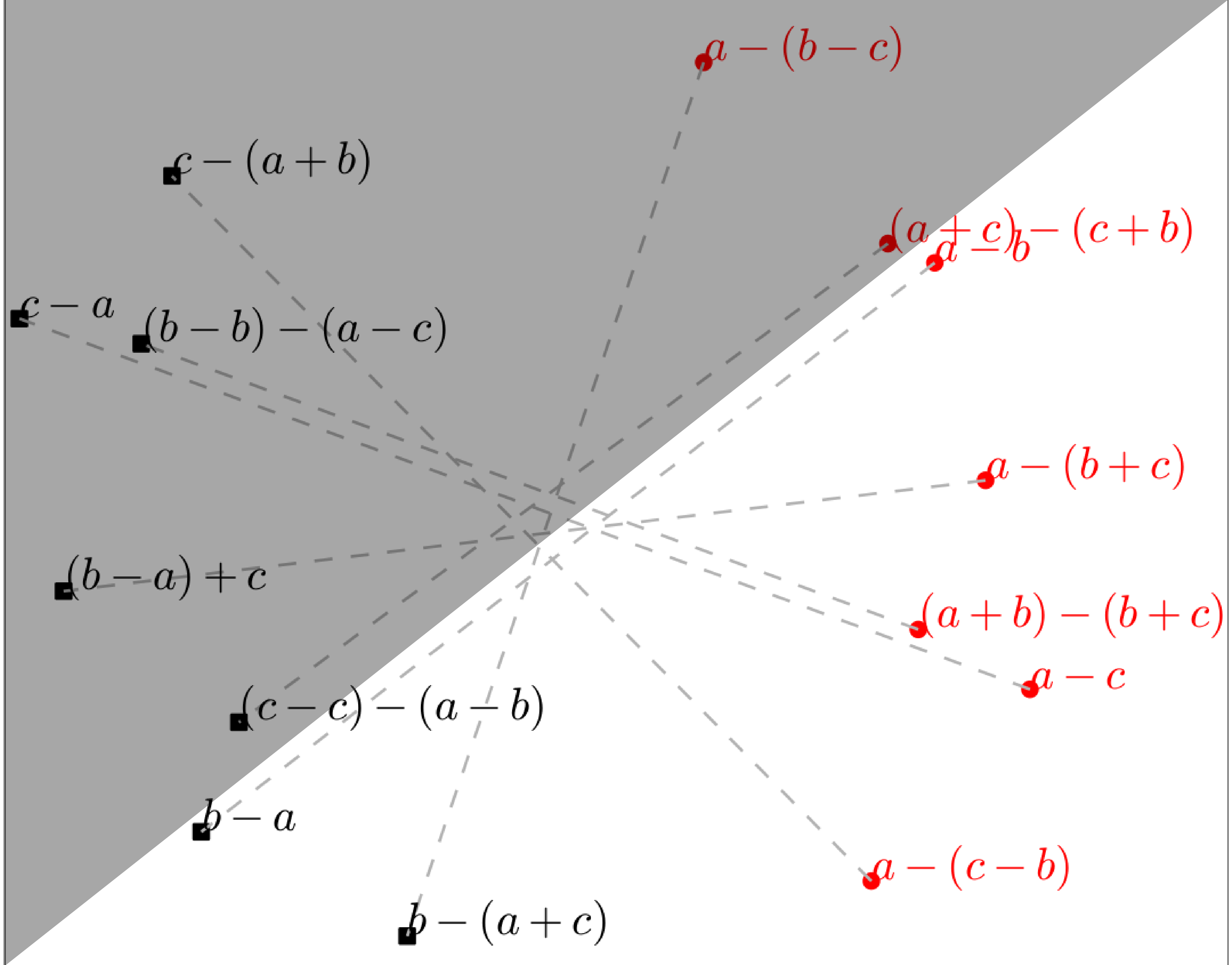
Visualizing Polynomials and their Negatives

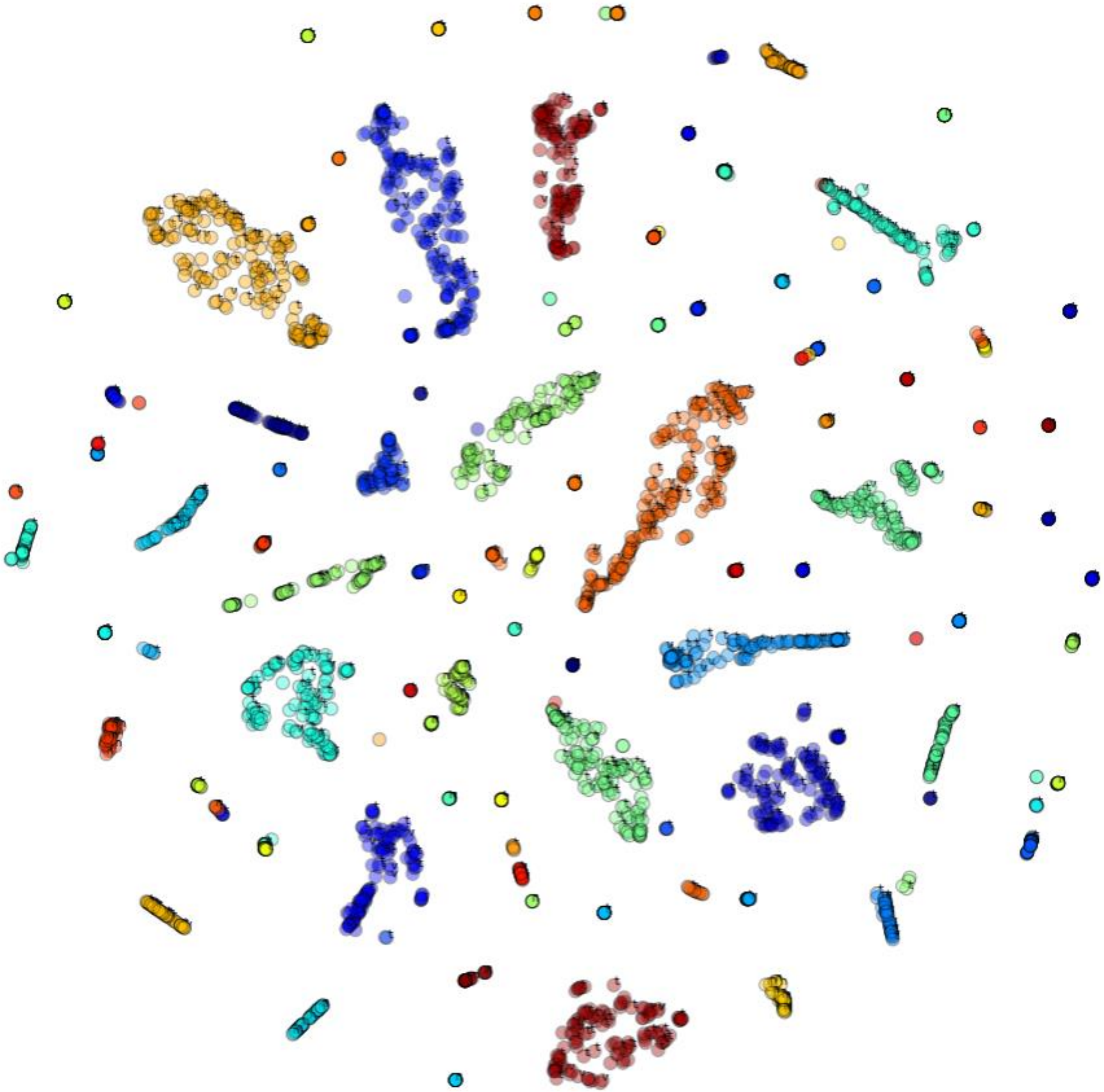


Visualizing Polynomials and their Negatives



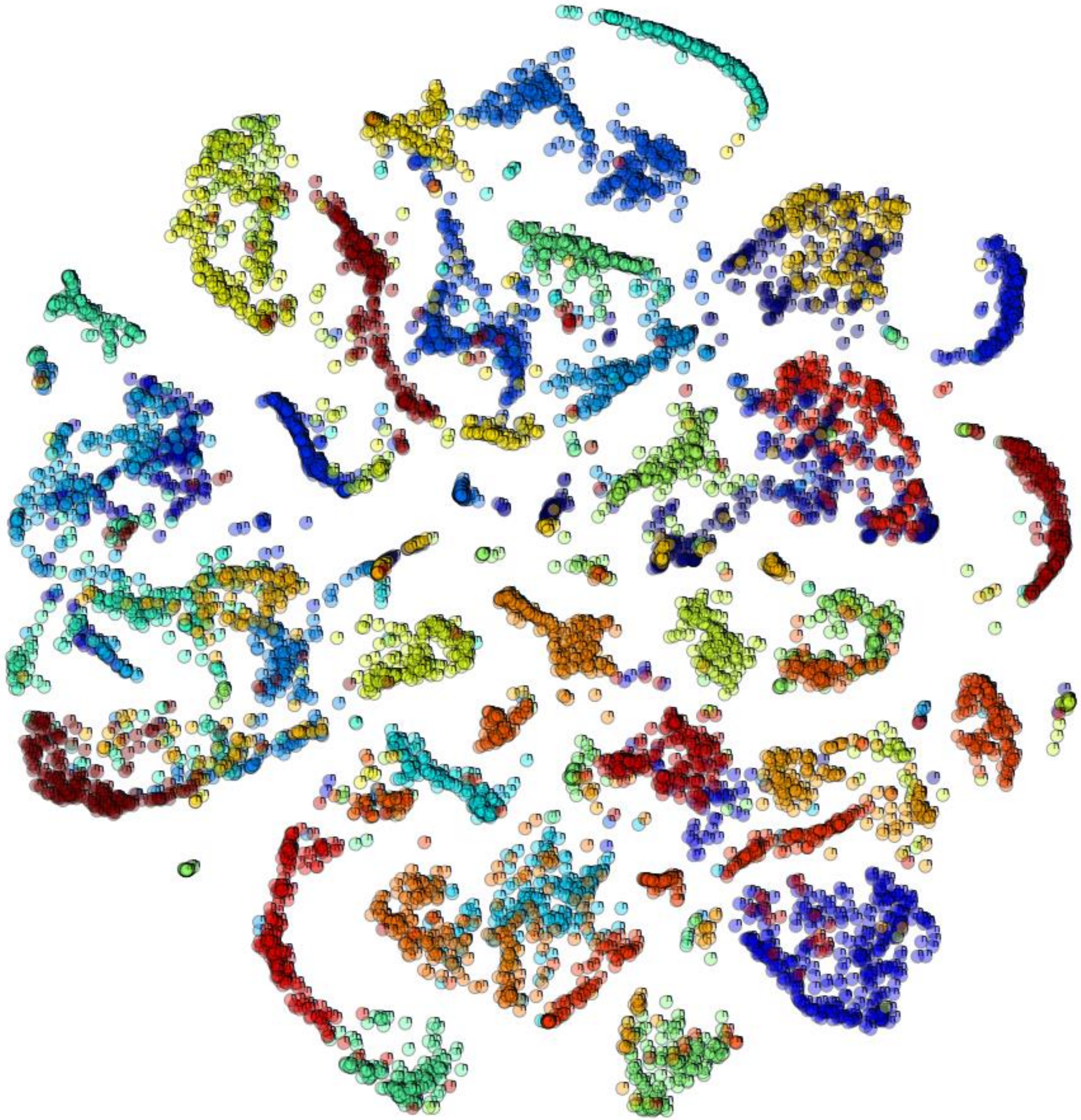
Visualizing Polynomials and their Negatives





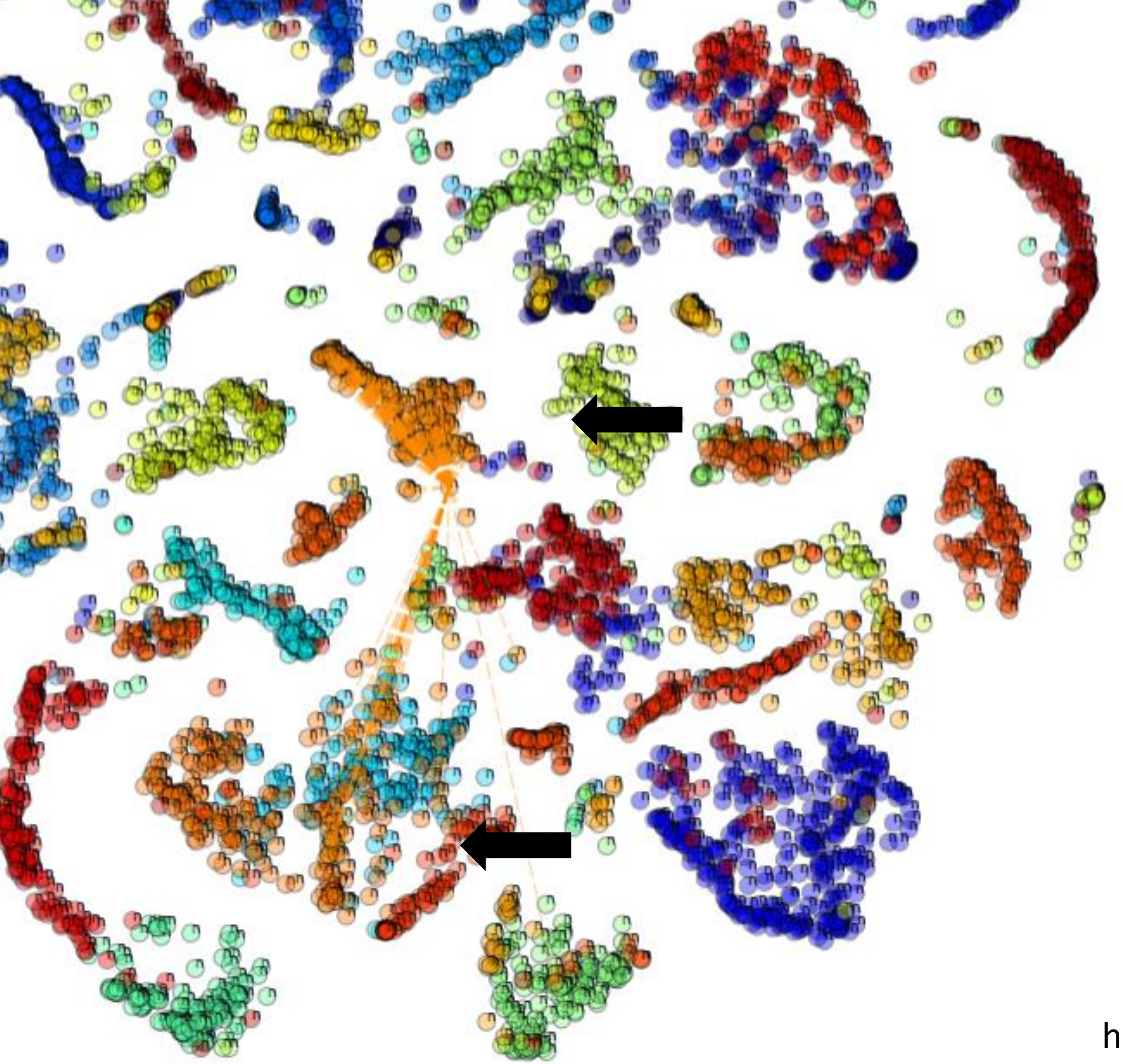
t-SNE Visualization
`SimpPoly8`

http://groups.inf.ed.ac.uk/cup/semvec/



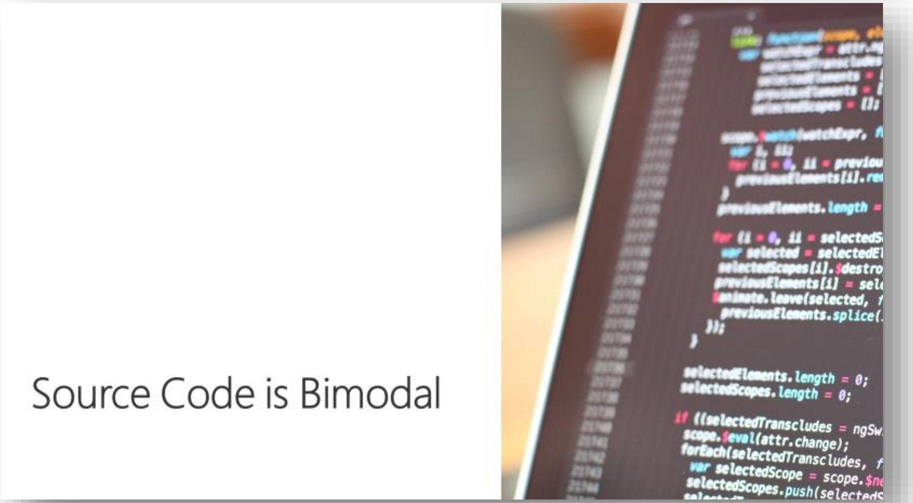
t-SNE Visualization
Bool10-UnseenTest

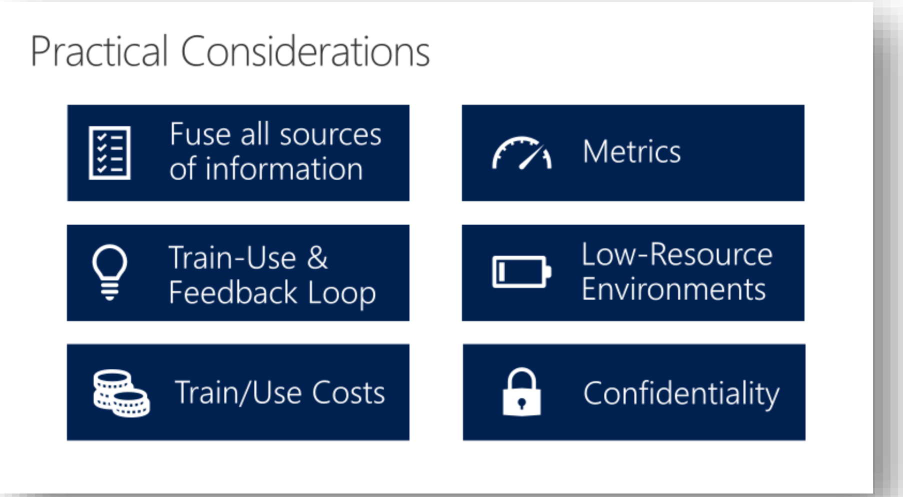
http://groups.inf.ed.ac.uk/cup/semvec/

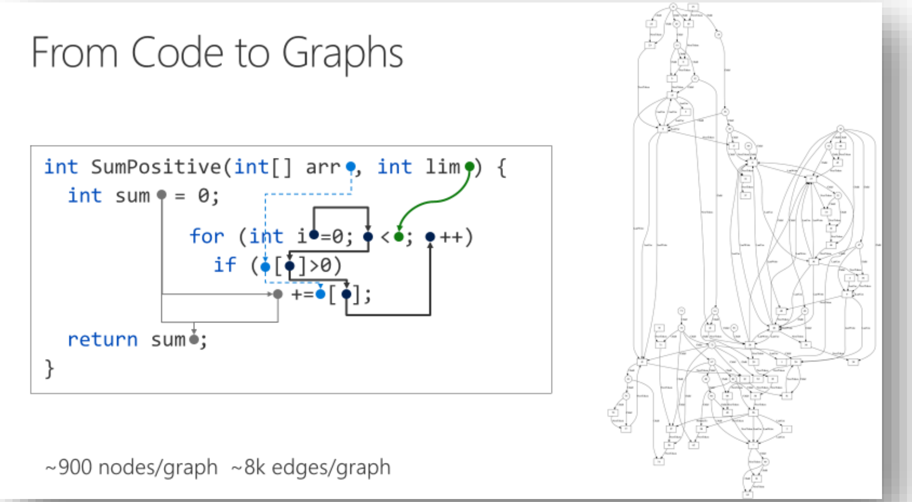


t-SNE Visualization
Bool10-UnseenTest

http://groups.inf.ed.ac.uk/cup/semvec/







Closing Thoughts

Microsoft