AutoMATES

Automated Model Assembly from Text, Equations and Software

Clayton Morrison

ml4ai.github.io/automates

THE UNIVERSITY OF ARIZONA

Machine Learning for Artificial Intelligence

Computational Language Understanding

ASKE Kickoff PI Meeting

5-6 December 2018

Motivation

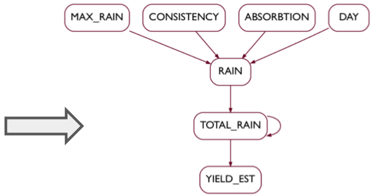
Many detailed expert models encoded in software and described in text

... however...

- Software requires manual curation to integrate
- Textual descriptions and software are not integrated

Goal: Construct and curate semantically-rich representations of scientific models by integrating: ***natural language descriptions*** and ***equations*** from publications and documentation, with the ***software*** that implements those models.

```
************************************************************************
*     UPDATE_EST - Updates the estimated yield of magic beans given
*       some additional amount of rainfall
************************************************************************
*
*     VARIABLES
*
*     INPUT RAIN      = Additional rainfall
*
*     INOUT YIELD_EST = Crop yield to update
*
************************************************************************
    SUBROUTINE UPDATE_EST(RAIN, TOTAL_RAIN, YIELD_EST)
      DOUBLE PRECISION RAIN, YIELD_EST, TOTAL_RAIN
      TOTAL_RAIN = TOTAL_RAIN + RAIN

*       Yield increases up to a point
      IF(TOTAL_RAIN .le. 40) THEN
          YIELD_EST = -(TOTAL_RAIN - 40) ** 2 / 16 + 100

*       Then sharply declines
      ELSE
          YIELD_EST = -TOTAL_RAIN + 140
      ENDIF

    END SUBROUTINE UPDATE_EST

 .
 .
 .
```



Text

Subroutine LAIS is called for both phases to compute the change in leaf area index (dLAI). During vegetative period, LAI increases as a function of the rate of leaf number increase. The potential rate is limited by soil water stress (both deficit and saturation), through SWFAC, and temperature, through PT. Its value is given by: <eqn 1>

Where PD is the plant density (plants/m²), EMP1 is the maximum leaf area expansion per leaf, (0.104 m /leaf) and a is given by: <eqn 2>

Where EMP2 and nb are coefficients in the expolinear equation and N is the development age of the plant (leaf number).

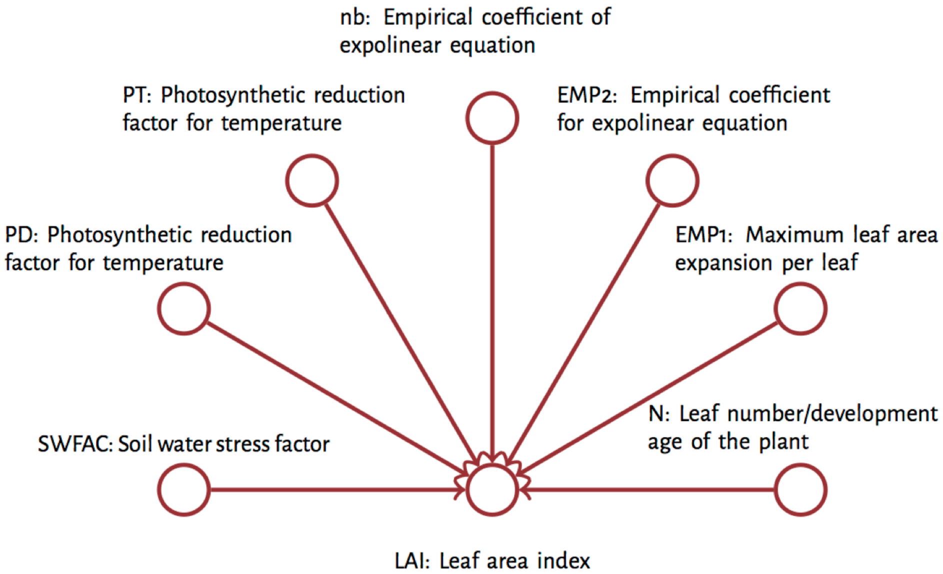
Equations

$$dLAI = SWFAC \cdot PT \cdot PD \cdot EMP1 \cdot \frac{a}{1+a}$$ <eqn 1>

$$a = e^{EMP2 \cdot (N - nb)}$$ <eqn 2>

Software

```
* dLAI = daily increase in leaf area index
(m2/m2/d)
* PD = plant density m-2
* EMP1 = empirical coef. for expoilinear eq.
* EMP2 = empirical coef. for expoilinear eq.
* nb = empirical coef. for expoilinear eq.
* dN = incremental leaf number
* N = leaf number
* PT = photosynthesis reduction factor for
temp.

a = exp(EMP2 * (N-nb))
dLAI = SWFAC * PD * EMP1 * PT * (a/(1+a)) * dN
```



$$\frac{d(\mathrm{LAI})}{dN} = \mathrm{SWFAC} \cdot \mathrm{PT} \cdot \mathrm{PD} \cdot \mathrm{EMP} \cdot \frac{e^{\mathrm{EMP} \cdot 2(N-nb)}}{1+e^{\mathrm{EMP} \cdot 2(N-nb)}}$$

Semantically-Enriched Grounded Function Network (GrFN)

Text

Subroutine LAIS is called for both phases to compute the change in leaf area index (dLAI). During vegetative period, LAI increases as a function of the rate of leaf number increase. The potential rate is limited by soil water stress (both deficit and saturation), through SWFAC, and temperature, through PT. Its value is given by: <eqn 1>

Where PD is the plant density (plants/m^2), EMP1 is the maximum leaf area expansion per leaf, (0.104 m /leaf) and a is given by: <eqn 2>

Where EMP2 and nb are coefficients in the expolinear equation and N is the development age of the plant (leaf number).

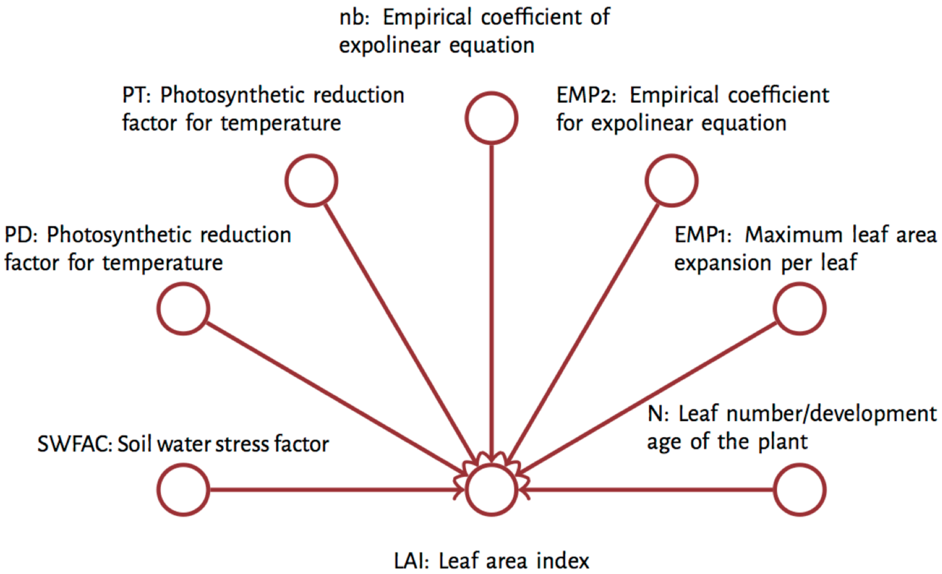
Equations

$$dLAI = SWFAC \cdot PT \cdot PD \cdot EMP1 \cdot \frac{a}{1+a}$$ <eqn 1>

$$a = e^{EMP2 \cdot (N-nb)}$$ <eqn 2>

Software

```
* dLAI = daily increase in leaf area index
(m2/m2/d)
* PD = plant density m-2
* EMP1 = empirical coef. for expoilinear eq.
* EMP2 = empirical coef. for expoilinear eq.
* nb = empirical coef. for expoilinear eq.
* dN = incremental leaf number
* N = leaf number
* PT = photosynthesis reduction factor for
temp.

a = exp(EMP2 * (N-nb))
dLAI = SWFAC * PD * EMP1 * PT * (a/(1+a)) * dN
```



$$\frac{d(\mathrm{LAI})}{dN} = \mathrm{SWFAC} \cdot \mathrm{PT} \cdot \mathrm{PD} \cdot \mathrm{EMP} \cdot \frac{e^{\mathrm{EMP} \cdot 2(N-nb)}}{1+e^{\mathrm{EMP} \cdot 2(N-nb)}}$$

Semantically-Enriched Grounded Function Network (GrFN)

Text

Subroutine LAIS is called for both phases to compute the change in leaf area index (dLAI). During vegetative period, LAI increases as a function of the rate of leaf number increase. The potential rate is limited by soil water stress (both deficit and saturation), through SWFAC, and temperature, through PT. Its value is given by: <eqn 1>

Where PD is the plant density (plants/m²), EMP1 is the maximum leaf area expansion per leaf, (0.104 m /leaf) and a is given by: <eqn 2>

Where EMP2 and nb are coefficients in the expolinear equation and N is the development age of the plant (leaf number).

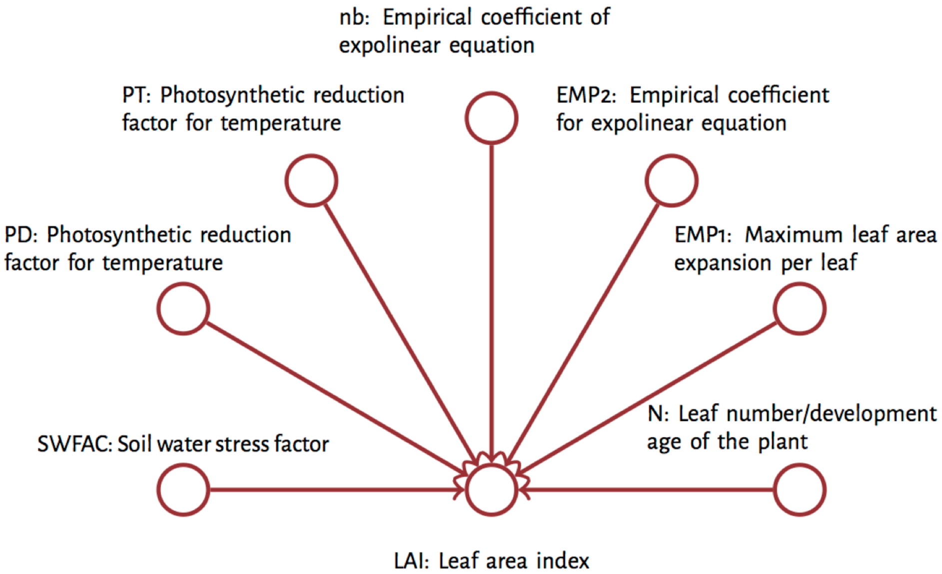
Equations

$$dLAI = SWFAC \cdot PT \cdot PD \cdot EMP1 \cdot \frac{a}{1+a}$$ <eqn 1>

$$a = e^{EMP2 \cdot (N - nb)}$$ <eqn 2>

Software

```
* dLAI = daily increase in leaf area index
(m2/m2/d)
* PD = plant density m-2
* EMP1 = empirical coef. for expoilinear eq.
* EMP2 = empirical coef. for expoilinear eq.
* nb = empirical coef. for expoilinear eq.
* dN = incremental leaf number
* N = leaf number
* PT = photosynthesis reduction factor for
temp.

a = exp(EMP2 * (N-nb))
dLAI = SWFAC * PD * EMP1 * PT * (a/(1+a)) * dN
```



$$\frac{d(\mathrm{LAI})}{dN} = \mathrm{SWFAC} \cdot \mathrm{PT} \cdot \mathrm{PD} \cdot \mathrm{EMP} \cdot \frac{e^{\mathrm{EMP} \cdot 2(N-nb)}}{1+e^{\mathrm{EMP} \cdot 2(N-nb)}}$$

Semantically-Enriched Grounded Function Network (GrFN)

Text

Subroutine LAIS is called for both phases to compute the change in leaf area index (dLAI). During vegetative period, LAI increases as a function of the rate of leaf number increase. The potential rate is limited by soil water stress (both deficit and saturation), through SWFAC, and temperature, through PT. Its value is given by: <eqn 1>

Where PD is the plant density (plants/m²), EMP1 is the maximum leaf area expansion per leaf, (0.104 m /leaf) and a is given by: <eqn 2>

Where EMP2 and nb are coefficients in the expolinear equation and N is the development age of the plant (leaf number).

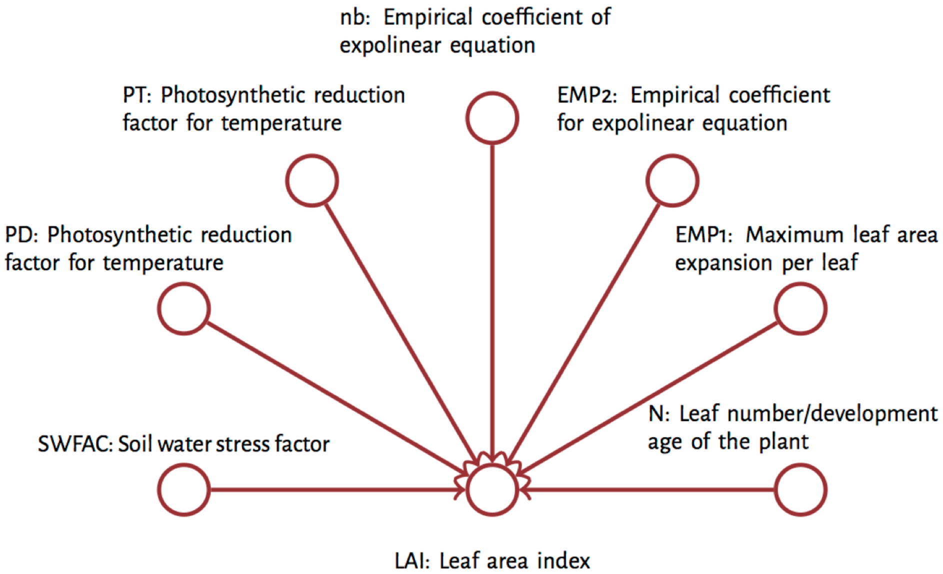
Equations

$$dLAI = SWFAC \cdot PT \cdot PD \cdot EMP1 \cdot \frac{a}{1+a}$$ <eqn 1>

$$a = e^{EMP2 \cdot (N - nb)}$$ <eqn 2>

Software

```
* dLAI = daily increase in leaf area index
(m2/m2/d)
* PD = plant density m-2
* EMP1 = empirical coef. for expoilinear eq.
* EMP2 = empirical coef. for expoilinear eq.
* nb = empirical coef. for expoilinear eq.
* dN = incremental leaf number
* N = leaf number
* PT = photosynthesis reduction factor for
temp.

a = exp(EMP2 * (N-nb))
dLAI = SWFAC * PD * EMP1 * PT * (a/(1+a)) * dN
```



$$\frac{d(\mathrm{LAI})}{dN} = \mathrm{SWFAC} \cdot \mathrm{PT} \cdot \mathrm{PD} \cdot \mathrm{EMP} \cdot \frac{e^{\mathrm{EMP} \cdot 2(N-nb)}}{1+e^{\mathrm{EMP} \cdot 2(N-nb)}}$$

Semantically-Enriched Grounded Function Network (GrFN)

Text

Subroutine LAIS is called for both phases to compute the change in leaf area index (dLAI). During vegetative period, LAI increases as a function of the rate of leaf number increase. The potential rate is limited by soil water stress (both deficit and saturation), through SWFAC, and temperature, through PT. Its value is given by: <eqn 1>

Where PD is the plant density (plants/m^2), EMP1 is the maximum leaf area expansion per leaf, (0.104 m /leaf) and a is given by: <eqn 2>

Where EMP2 and nb are coefficients in the expolinear equation and N is the development age of the plant (leaf number).

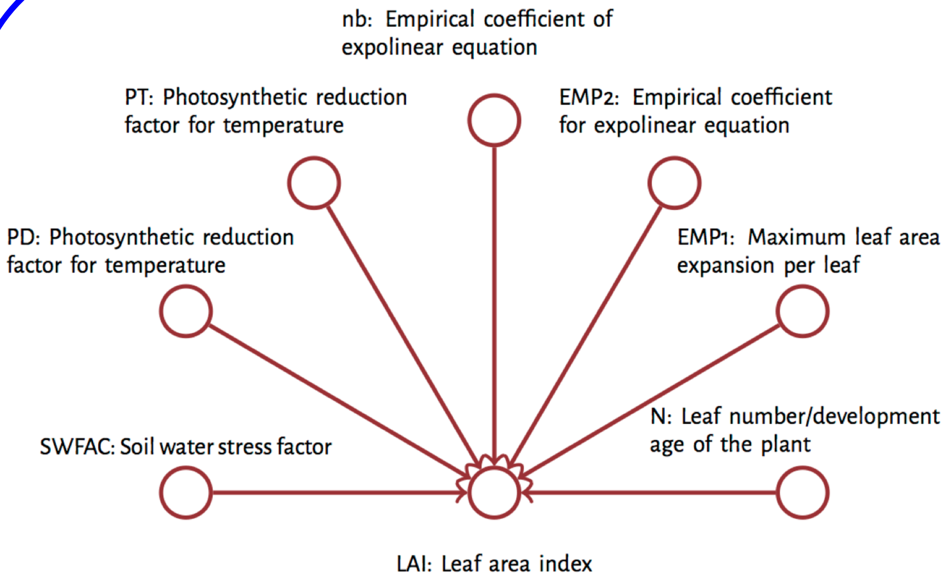
Equations

$$dLAI = SWFAC \cdot PT \cdot PD \cdot EMP1 \cdot \frac{a}{1+a} \quad \text{<eqn 1>}$$

$$a = e^{EMP2 \cdot (N-nb)} \quad \text{<eqn 2>}$$

Software

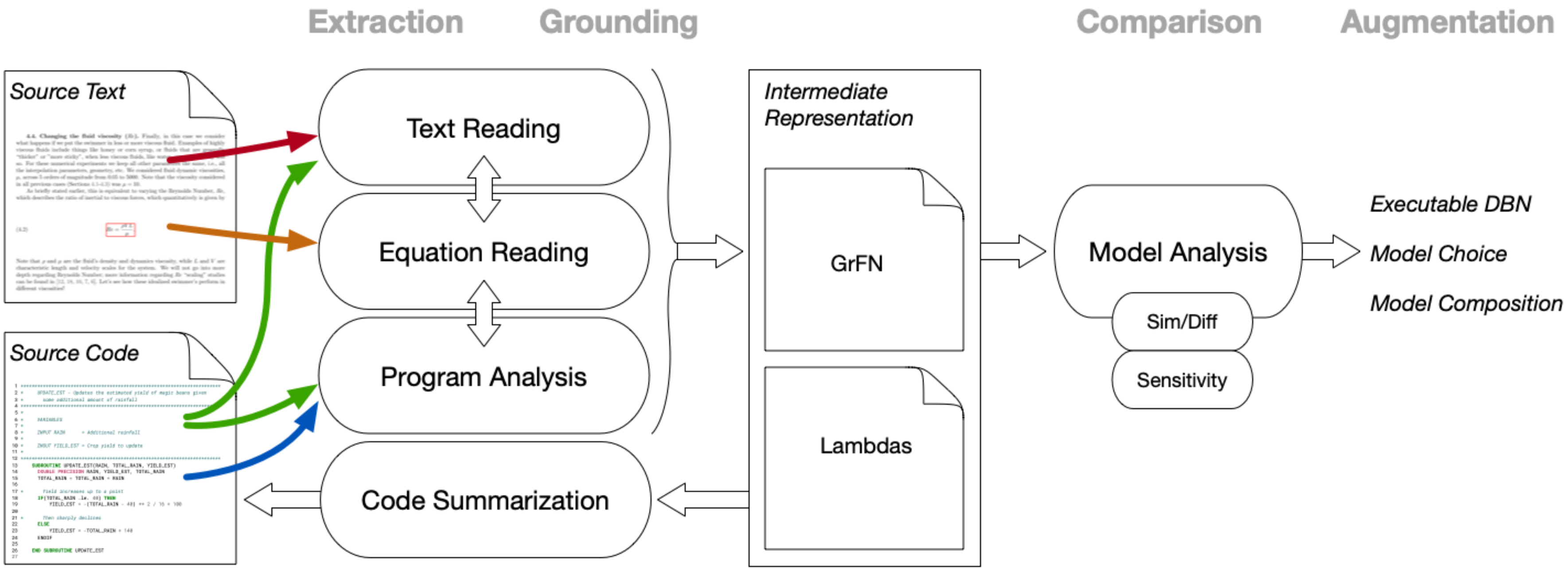
```
* dLAI = daily increase in leaf area index
(m2/m2/d)
* PD = plant density m-2
* EMP1 = empirical coef. for expoilinear eq.
* EMP2 = empirical coef. for expoilinear eq.
* nb = empirical coef. for expoilinear eq.
* dN = incremental leaf number
* N = leaf number
* PT = photosynthesis reduction factor for
temp.

a = exp(EMP2 * (N-nb))
dLAI = SWFAC * PD * EMP1 * PT * (a/(1+a)) * dN
```



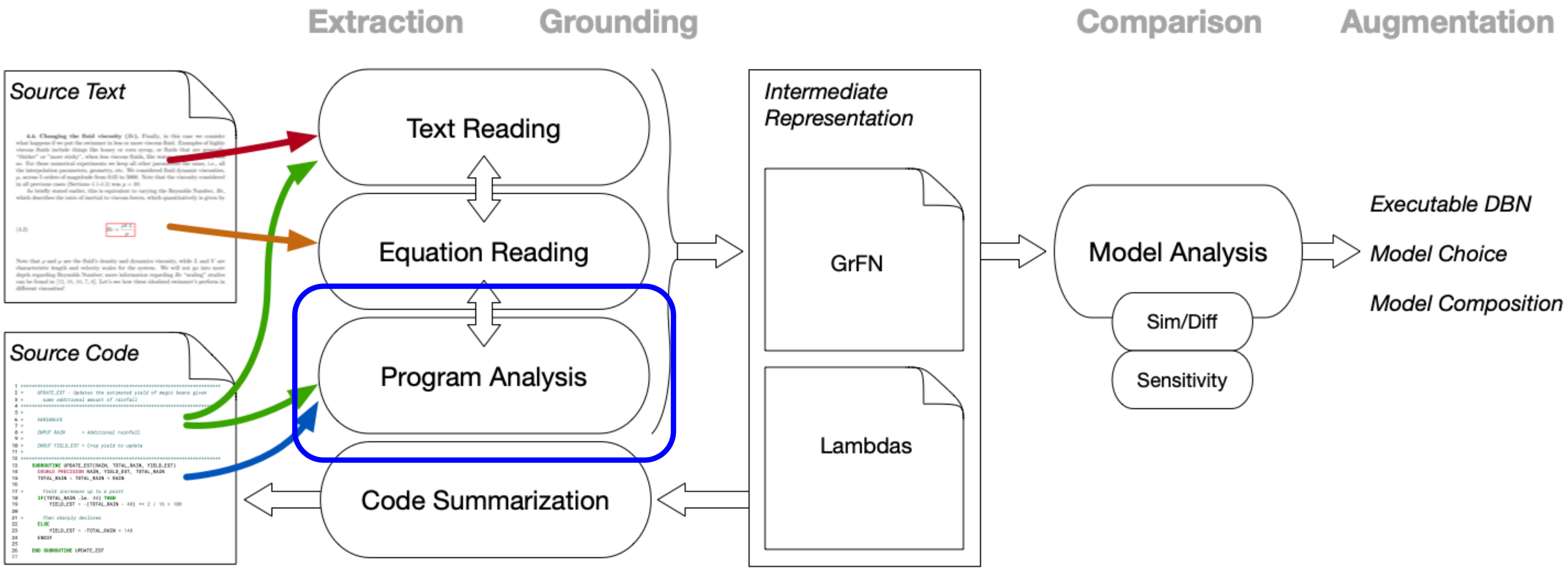
$$\frac{d(\mathrm{LAI})}{dN} = \mathrm{SWFAC} \cdot \mathrm{PT} \cdot \mathrm{PD} \cdot \mathrm{EMP} \cdot \frac{e^{\mathrm{EMP}\cdot 2(N-nb)}}{1+e^{\mathrm{EMP}\cdot 2(N-nb)}}$$

Semantically-Enriched Grounded Function Network (GrFN)

AutoMATES Architecture



AutoMATES Architecture



From source…

crop_yield.f

```
**********************************************************************
*     UPDATE_EST - Updates the estimated yield of magic beans given
*       some additional amount of rainfall
**********************************************************************
*
*     VARIABLES
*
*     INPUT RAIN      = Additional rainfall
*
*     INOUT YIELD_EST = Crop yield to update
*
**********************************************************************
      SUBROUTINE UPDATE_EST(RAIN, TOTAL_RAIN, YIELD_EST)
        DOUBLE PRECISION RAIN, YIELD_EST, TOTAL_RAIN
        TOTAL_RAIN = TOTAL_RAIN + RAIN

*         Yield increases up to a point
        IF(TOTAL_RAIN .le. 40) THEN
          YIELD_EST = -(TOTAL_RAIN - 40) ** 2 / 16 + 100

*         Then sharply declines
        ELSE
          YIELD_EST = -TOTAL_RAIN + 140
        ENDIF

      END SUBROUTINE UPDATE_EST

```

function

(program, subroutine)

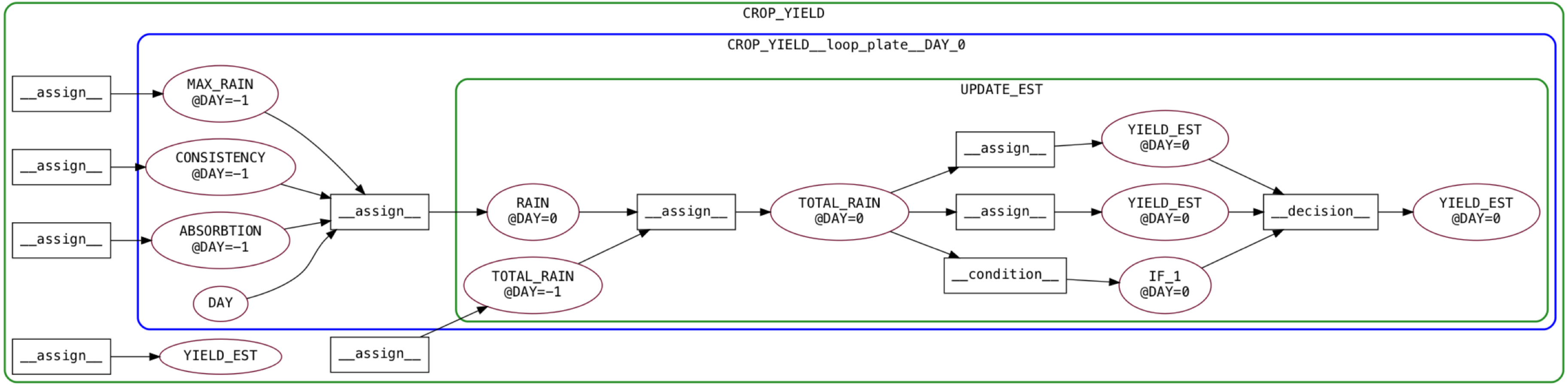
conditional

```
**********************************************************************
*     CROP_YIELD - Estimate the yield of magic beans given a simple
*       model for rainfall
**********************************************************************
*
*     VARIABLES
*
*     INPUT MAX_RAIN    = The maximum rain for the month
*     INPUT CONSISTENCY = The consistency of the rainfall
*       (higher = more consistent)
*     INPUT ABSORBTION  = Estimates the % of rainfall absorbed into the
*       soil (i.e. % lost due to evaporation, runoff)
*
*     OUTPUT YIELD_EST = The estimated yield of magic beans
*
*     DAY              = The current day of the month
*     RAIN             = The rainfall estimate for the current day
*
**********************************************************************
      PROGRAM CROP_YIELD
      IMPLICIT NONE

      INTEGER DAY
      DOUBLE PRECISION RAIN, YIELD_EST, TOTAL_RAIN
      DOUBLE PRECISION MAX_RAIN, CONSISTENCY, ABSORBTION

      MAX_RAIN = 4.0
      CONSISTENCY = 64.0
      ABSORBTION = 0.6

      YIELD_EST = 0
      TOTAL_RAIN = 0

      DO 20 DAY=1,31
*         Compute rainfall for the current day
        RAIN = (-(DAY - 16) ** 2 / CONSISTENCY + MAX_RAIN) * ABSORBTION

*         Update rainfall estimate
        CALL UPDATE_EST(RAIN, TOTAL_RAIN, YIELD_EST)
        PRINT *, "Day ", DAY, " Estimate: ", YIELD_EST

   20 ENDDO

      PRINT *, "Crop Yield(%): ", YIELD_EST

      END PROGRAM CROP_YIELD
```

var declarations

var assignments

loop

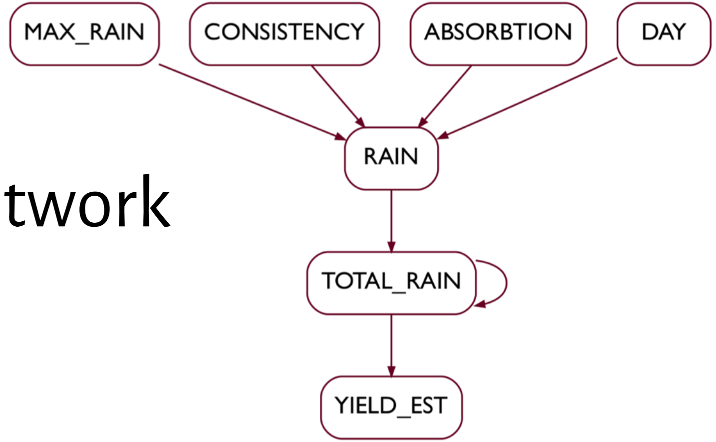


... derive:

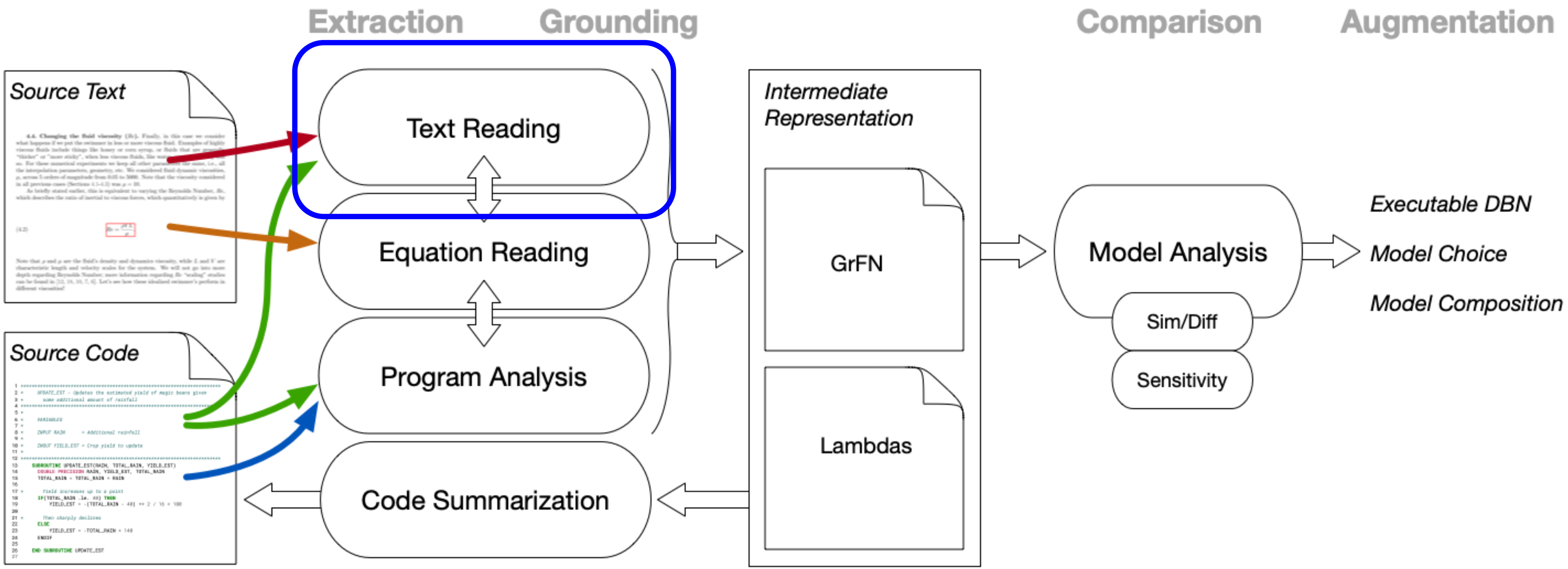
GrFN: Grounded Function Network

Executable DBN

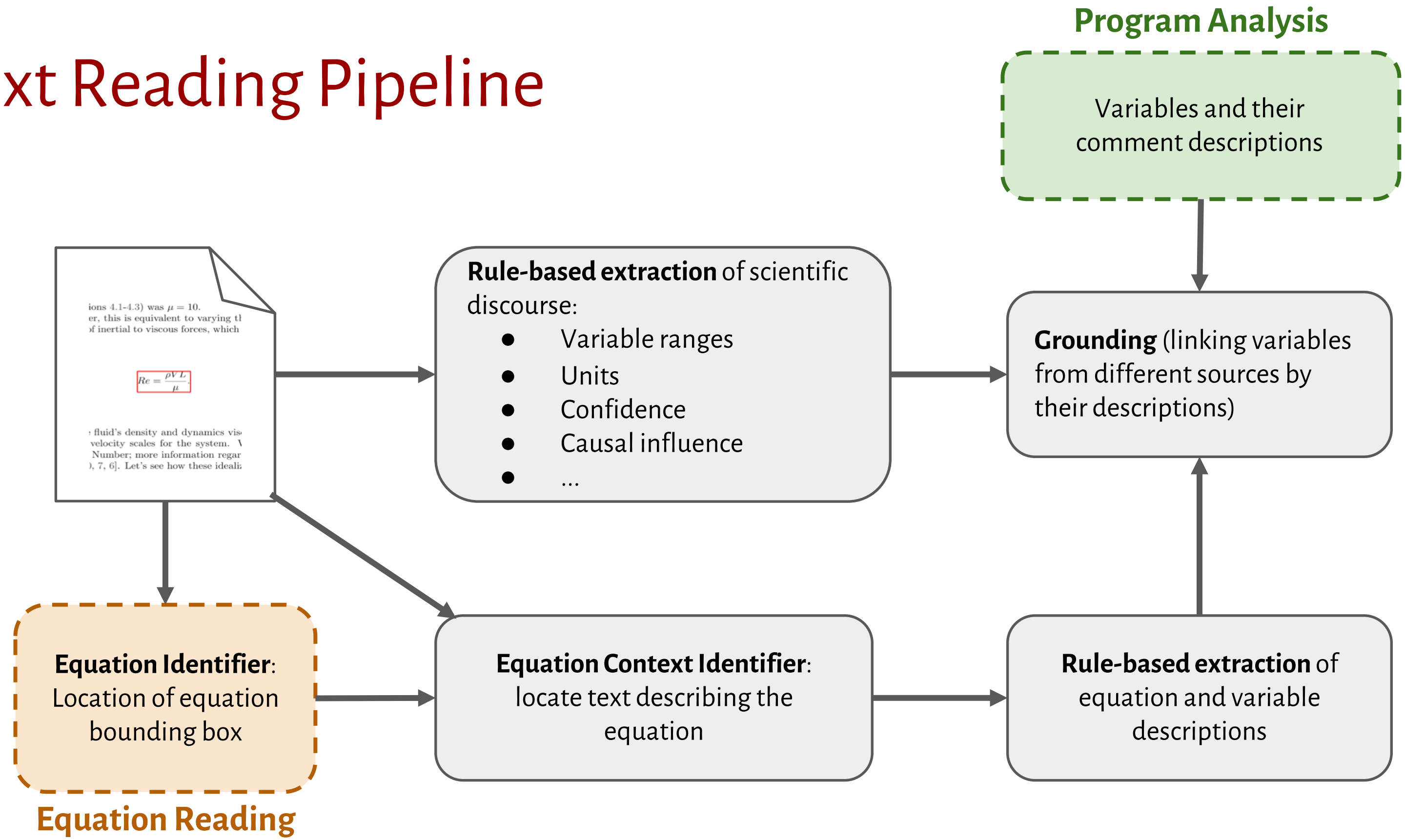
... with associated comments



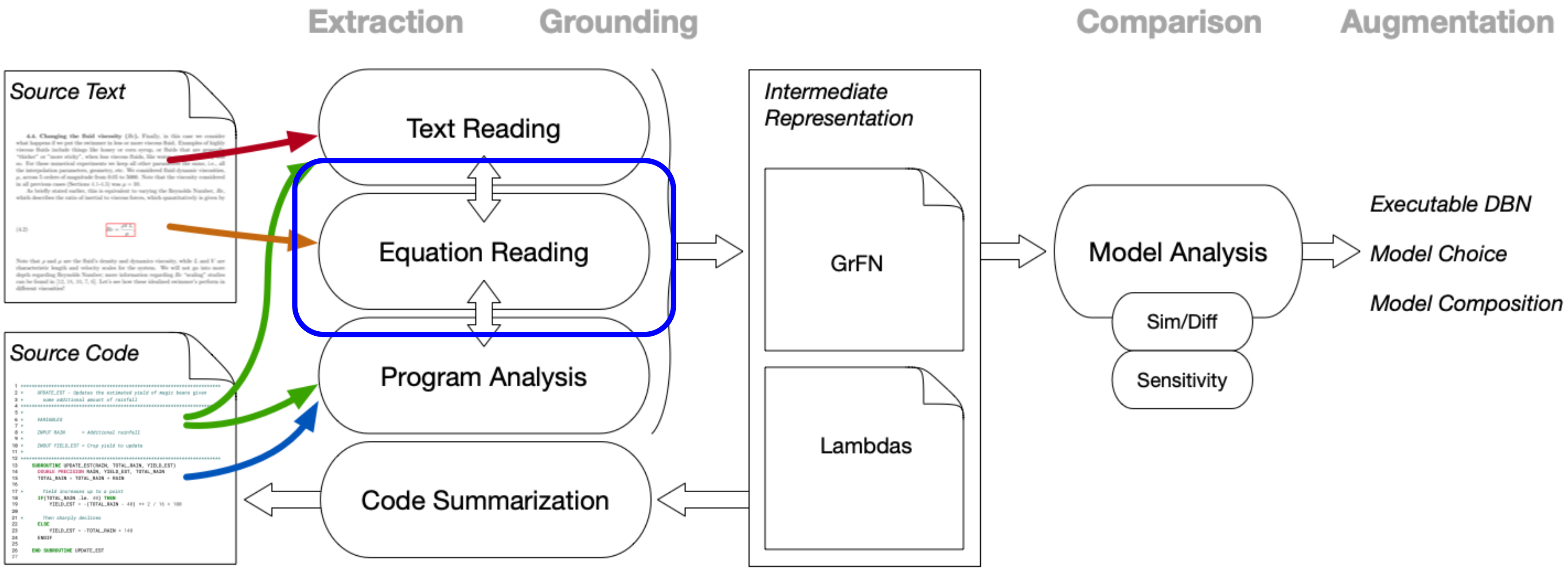
AutoMATES Architecture



Text Reading Pipeline



AutoMATES Architecture



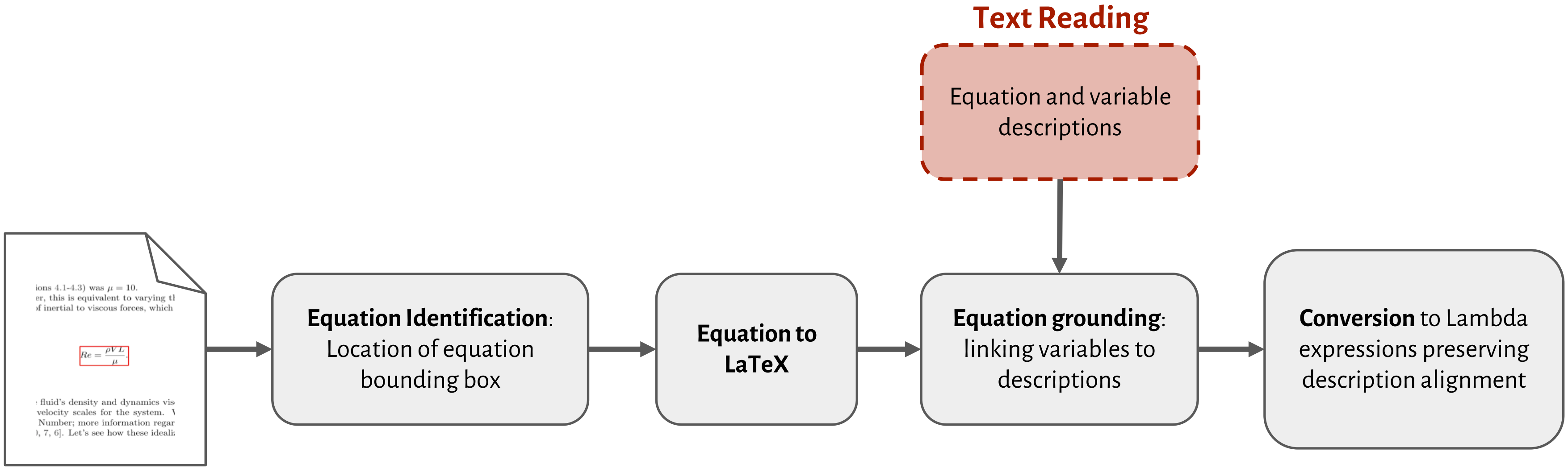
Equation Reading

As briefly stated earlier, this is equivalent to varying the Reynolds Number, Re, which describes the ratio of inertial to viscous forces, which quantitatively is given by

$$\boxed{Re = \frac{\rho V L}{\mu}.} \tag{4.2}$$

Note that ρ and μ are the fluid's density and dynamics viscosity, while L and V are characteristic length and velocity scales for the system. We will not go into more depth regarding Reynolds Number; more information regarding Re "scaling" studies can be found in [12, 18, 10, 7, 6]. Let's see how these idealized swimmer's perform in different viscosities!

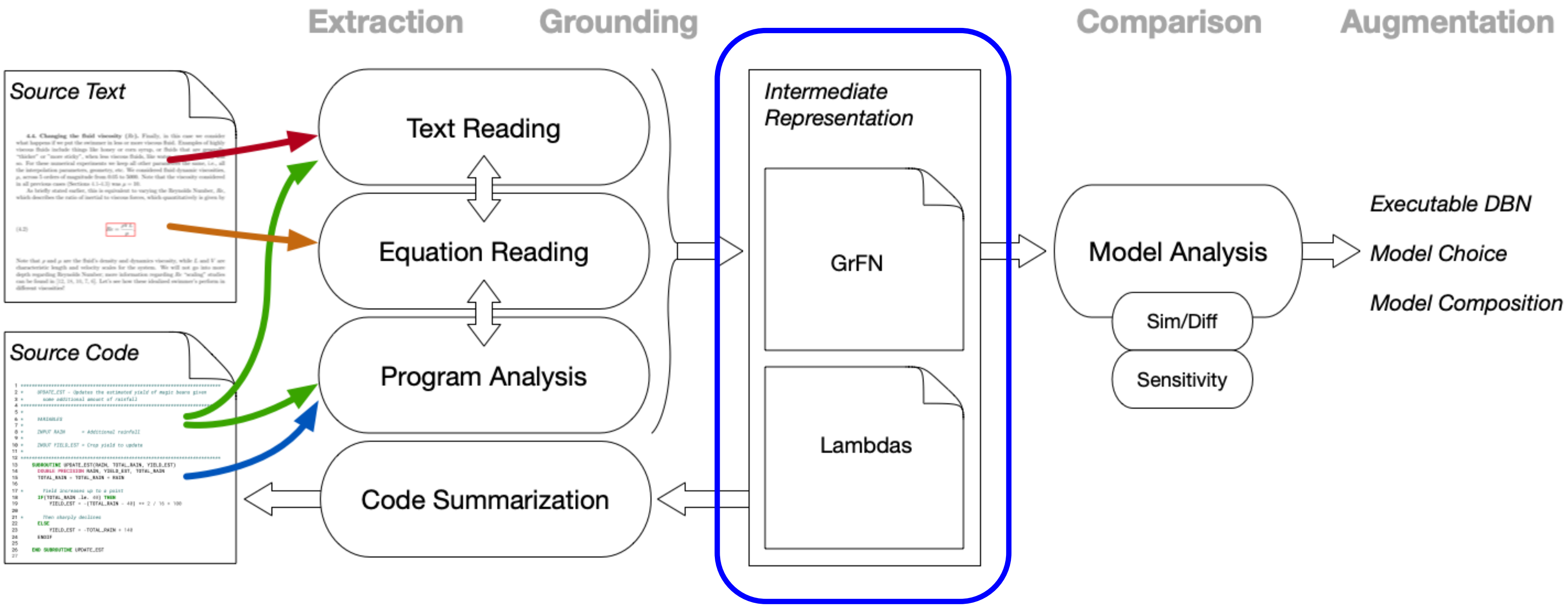
Equation Reading Pipeline



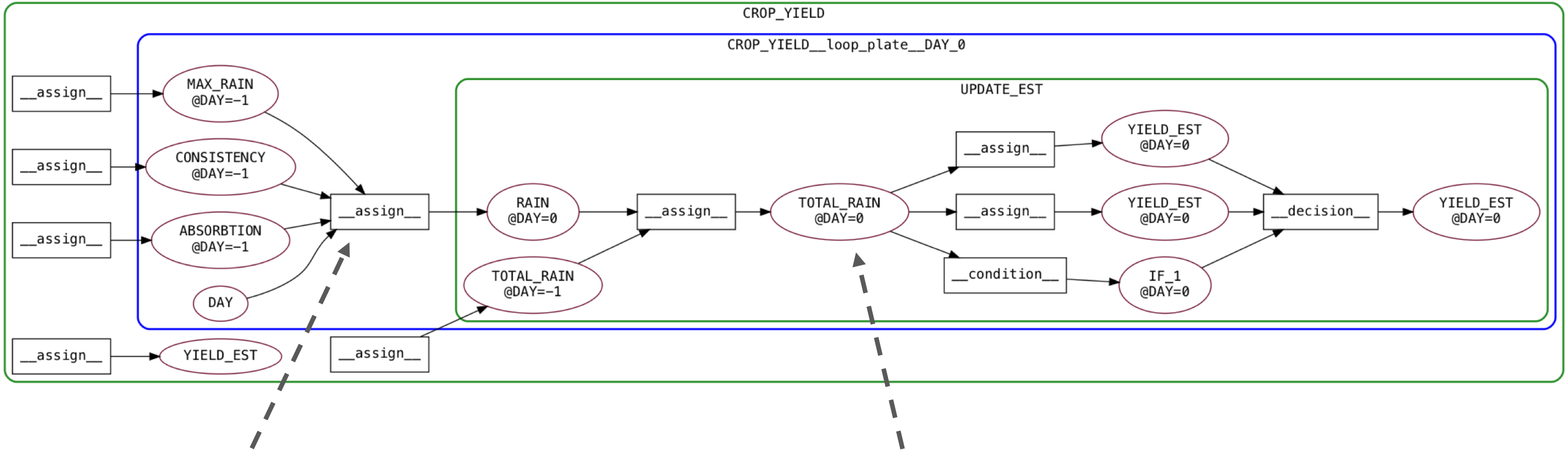

```
Re = \frac{\rho V L}{\mu}
```

```python
def Re(rho, V, L, mu):
    return (rho * V * L) / mu
```

AutoMATES Architecture



GrFN: Grounded Function Network



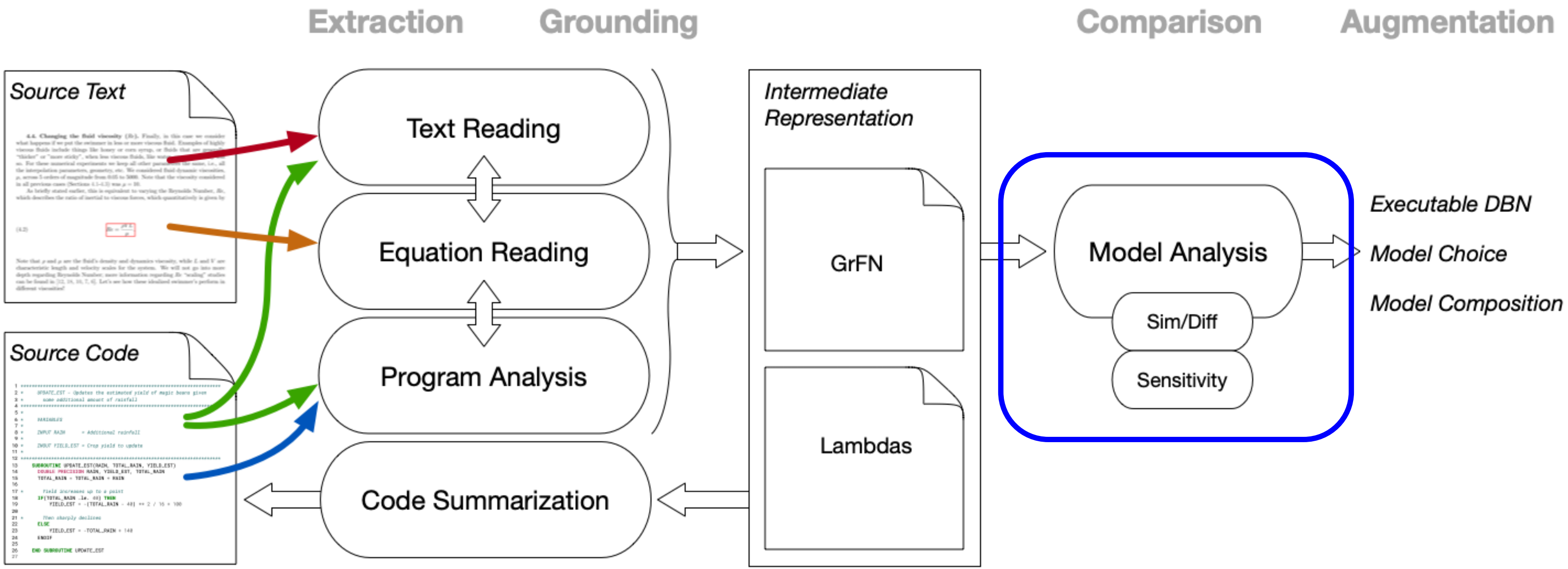
Lambdas

```
def assign_RAIN(MAX_RAIN, CONSISITENCY, ABSORPTON, DAY)
    return (-(DAY - 16) ** 2 / CONSISTENCY + MAX_RAIN)
           * ABSORBTION
```

Variable Metadata

- Comments
- Types, ranges, default values

AutoMATES Architecture



Model Analysis: Structural Comparison

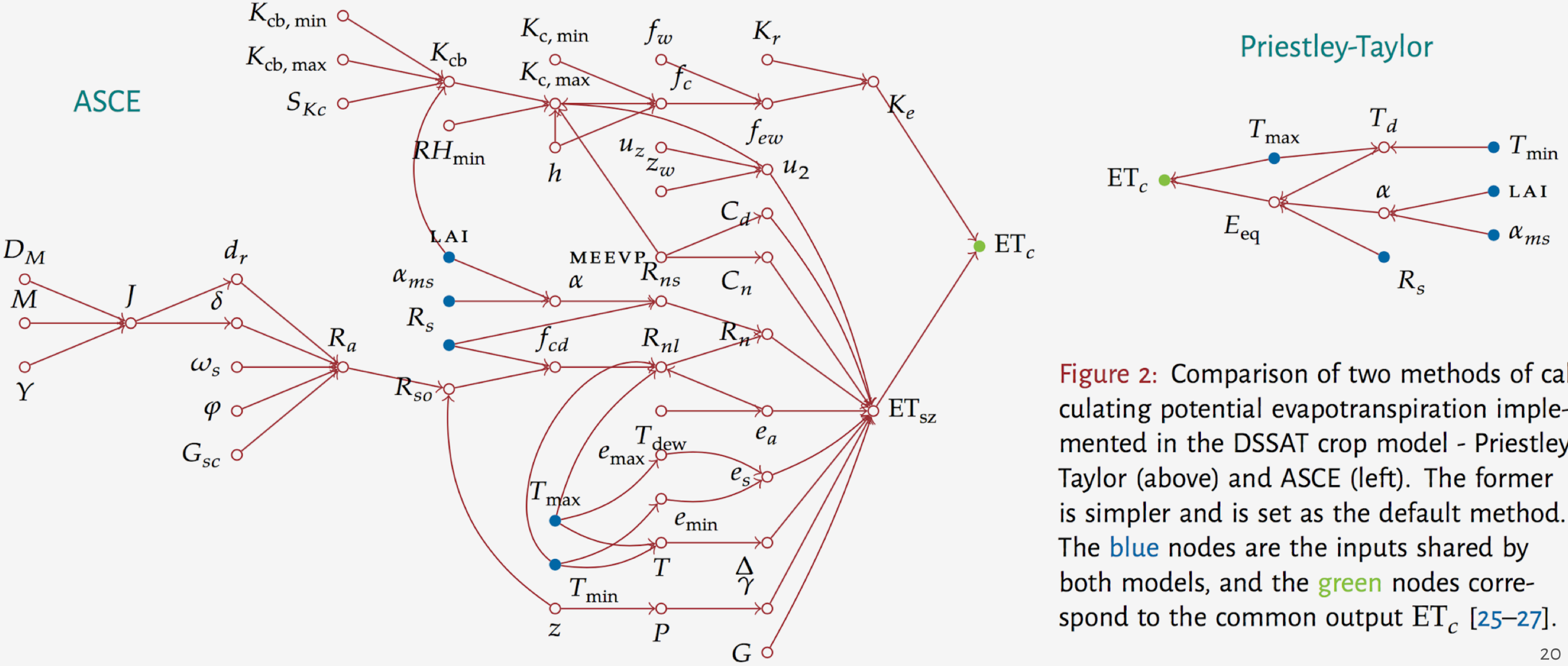
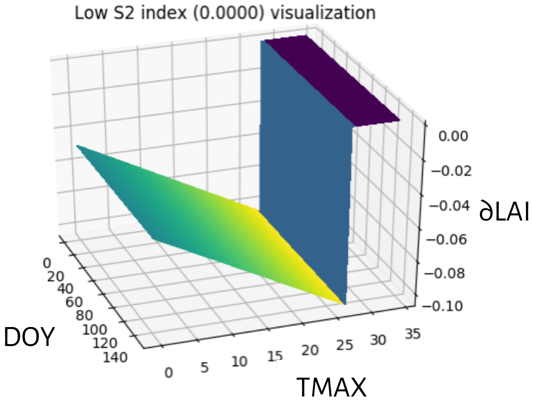
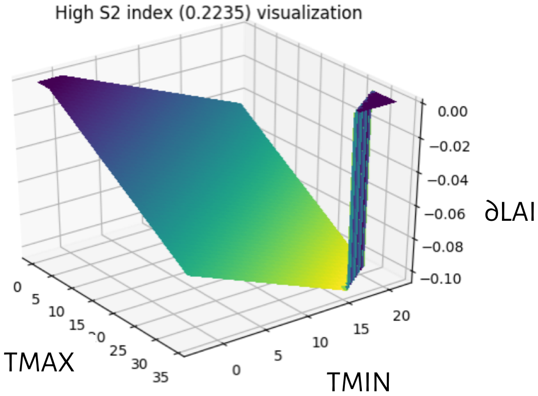


Figure 2: Comparison of two methods of calculating potential evapotranspiration implemented in the DSSAT crop model - Priestley-Taylor (above) and ASCE (left). The former is simpler and is set as the default method. The blue nodes are the inputs shared by both models, and the green nodes correspond to the common output ET_c [25–27].

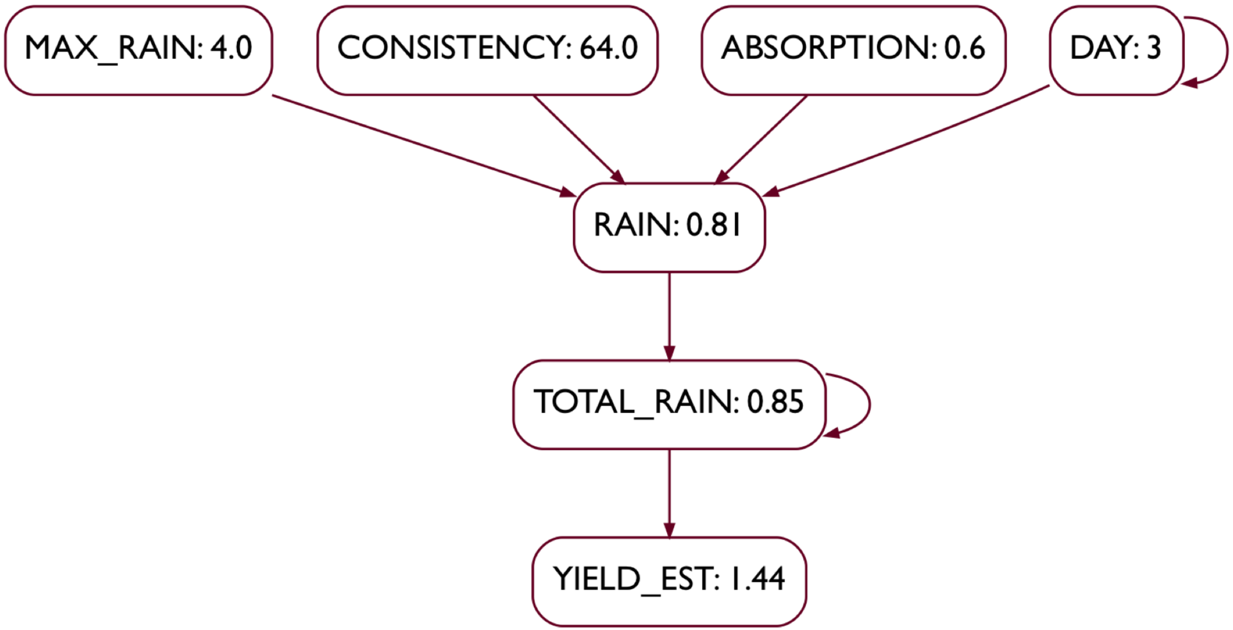
Model Analysis: Sobol Sensitivity Analysis



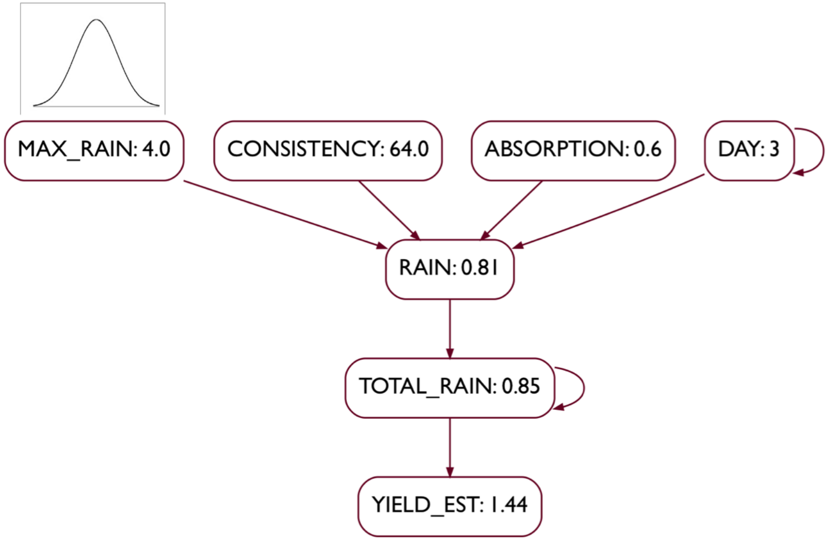


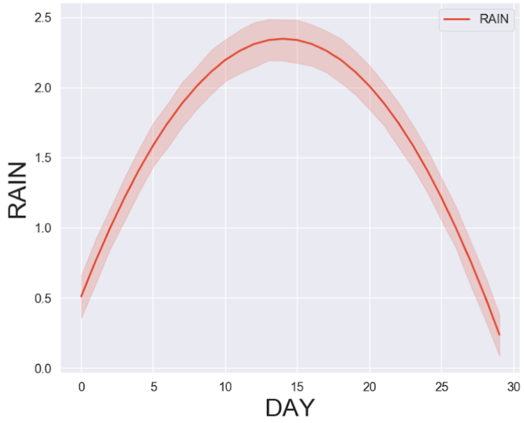
- Sensitivity index computation is done using Sobol sensitivity analysis
- Computing the sensitivity indices allows us to see which variables and which pairs of variables account for the most variance in overall model output

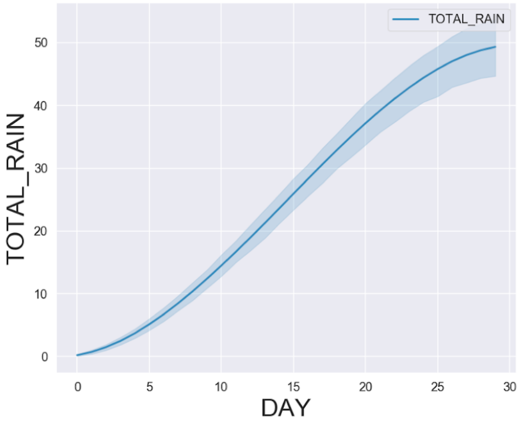
Model Analysis: GrFN as Dynamic Bayesian Network

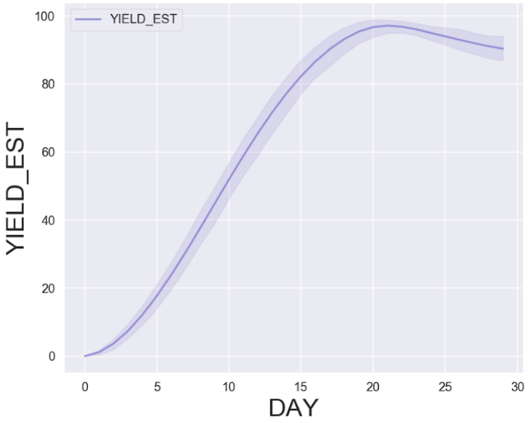


Model Analysis: DBN

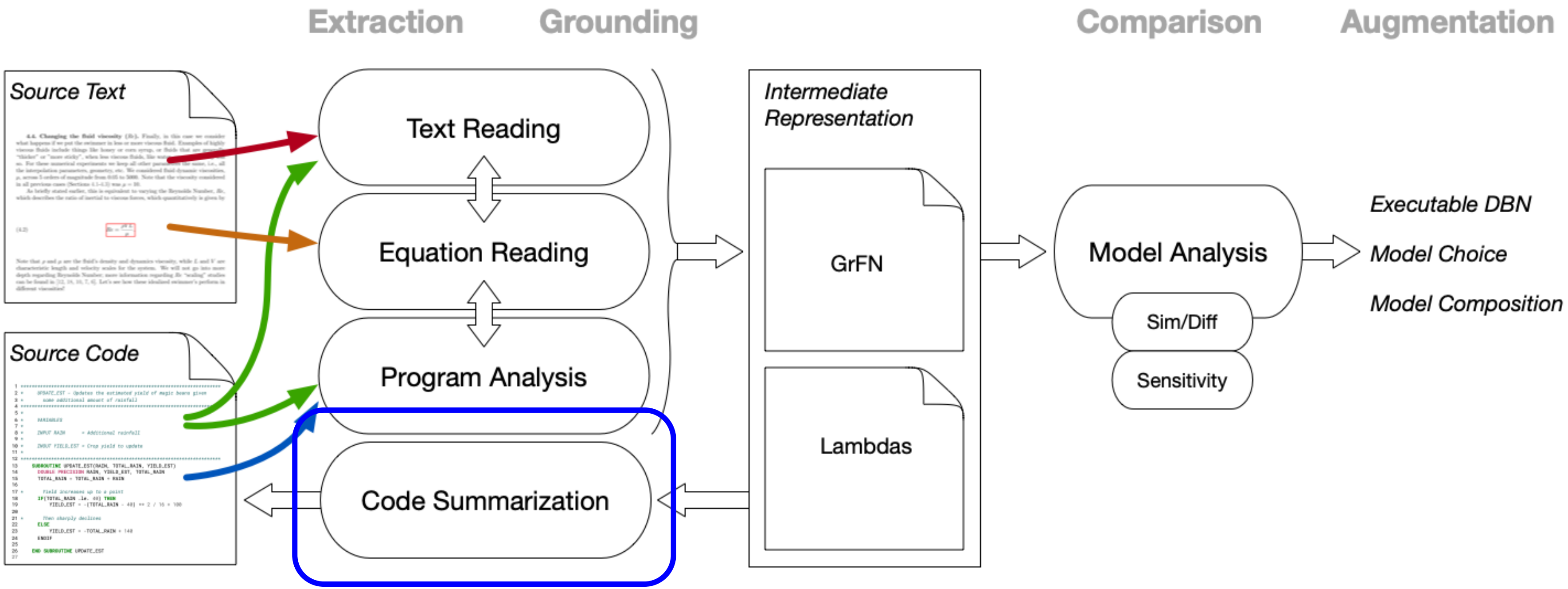








AutoMATES Architecture



Code Summarization: Training Corpus

- Neural network architecture that can encode functions and generate natural language summaries
- Trained using existing, well-documented open source code!

… and more…

Summary

Integration of model semantics
from ***text***, ***equations*** and ***software***
into a ***uniform framework*** for analysis.

- Automating linking of software to text discourse context
- Model comparison and sensitivity analysis in a uniform framework
- Contextual debugging and model communication
- Facilitate import of source code to libraries (e.g., MINT model store)
- Comparison of natural language -derived Causal Analysis Graphs (CAG^{NL}s) to software CAGs (CAG^{S}s)
    - Fill in causal details missing in CAG^{NL}s
    - Expose assumed common-sense knowledge underlying CAG^{NL}s

Thank You!

- Masha Alexeeva
- Pratik Bhandari
- Saumya Debray (co-PI)
- Paul Hein
- Jennifer Kadowaki
- Clay Morrison (PI)
- Adarsh Pyarelal (co-PI)
- Becky Sharp (co-PI)
- Marco Valenzuela-Escárcega (co-PI)