

# MDS Capstone Project Final Report

## Empowering Agriculture: Enhanced GHG Emissions Modeling

Yi Han, He Ma, Hancheng Qin

### Table of contents

<b>1</b>	<b>Executive summary</b>	<b>2</b>
<b>2</b>	<b>Introduction</b>	<b>2</b>
<b>3</b>	<b>Data science methods</b>	<b>3</b>
3.1	Model development: PyHolos . . . . .	3
3.1.1	Model Reconstruction . . . . .	3
3.1.2	Enhanced Flexibility . . . . .	3
3.2	Analysis . . . . .	4
3.2.1	Validation of PyHolos Outputs . . . . .	4
3.2.2	Sensitivity Analysis Using PyHolos . . . . .	4
<b>4</b>	<b>Data product and results</b>	<b>4</b>
4.1	Data product - PyHolos . . . . .	4
4.1.1	Model reconstruction: Farmer Mode . . . . .	4
4.1.2	Enhanced flexibility: Scientific Mode . . . . .	6
4.2	Results . . . . .	6
4.2.1	PyHolos model validation . . . . .	6
4.2.2	Sensitivity Analysis Results . . . . .	7
4.3	Advantages and limitations . . . . .	8
<b>5</b>	<b>Conclusions and recommendations</b>	<b>9</b>
<b>6</b>	<b>Acknowledgements</b>	<b>9</b>
<b>7</b>	<b>References</b>	<b>9</b>
<b>8</b>	<b>Appendix</b>	<b>10</b>
8.1	Parameters for Crop Residue Nitrogen Calculation . . . . .	10
8.2	Equations for Crop Residue Nitrogen Calculation . . . . .	12
8.2.1	Total Crop Residues . . . . .	12
8.2.2	Emission Factor . . . . .	15
8.2.3	Emission . . . . .	16

# 1 Executive summary

In collaboration with LiteFarm, we developed PyHolos, a Python-based model designed to enhance greenhouse gas emissions calculations, specifically for nitrous oxide. PyHolos can integrate into the existing LiteFarm framework. It further incorporates external data sources and allows for user-defined parameter ranges, offering improvements over the original Holos model. This tool not only provides farmers with emissions estimates for informed decision-making, but also assists researchers in uncertainty analysis for parameters and regions with significant data variability. Although robust, PyHolos has shown minor discrepancies in emission estimates compared to the original Holos software. To refine these calculations, we recommend ongoing communication with the Holos team for continuous improvement.

## 2 Introduction

Climate change presents a major challenge for our planet, with agriculture significantly contributing to greenhouse gas (GHG) emissions through activities such as livestock farming, fertilizer application, and land management practices (Johnson et al. 2007). Farmers are central to addressing this challenge, as their choices in crop selection and farming practices directly impact carbon emissions (Jantke et al. 2020). Agricultural choices are influenced by immediate profits, long-term soil health, and the financial benefits of conservation and sustainability programs. Carbon credit, for example, offers farmers the opportunity to earn extra revenue by adopting practices that reduce carbon emissions (Barbato and Strong 2023). However, farmers lack scientific support to balance financial losses from less profitable crops with gains from carbon credits.

Addressing this gap, our partner, the LiteFarm team, is developing innovative tools that help farmers achieve both financial and environmental benefits. The LiteFarm team consists of experts in research and development who are committed to providing tools that transform data into actionable insights for sustainability. They have developed an application that helps farmers manage their daily operations, boosting their income while also encouraging environmentally friendly practices. In the next phase, the team plans to enhance the app by integrating greenhouse gas (GHG) emissions calculations to further support sustainable farming decisions.

Selecting a suitable GHG emissions calculation model is crucial for this task. The LiteFarm team chose the Holos model for its open-source nature and its endorsement by the Canadian government for reliability (Agriculture & Agri-Food Canada 2022). Holos stands out as a dependable, cost-free alternative to commercial models. Nonetheless, this model has limitations: it operates as a standalone tool and cannot be directly integrated with the LiteFarm framework. Holos runs exclusively on Windows OS and is developed in C#, restricting its wider usability. Additionally, its fixed parameters limit the model’s flexibility and adaptability in incorporating data from new or alternative sources.

Our project aims to integrate the Holos GHG emission calculation module into the LiteFarm framework, focusing on nitrous oxide ( $N_2O$ ) emissions from crop residue. We enhance the existing Holos model to better align with our partner’s needs. Our objectives are:

- (1) **Model Reconstruction:** We aim to rebuild the Holos model in Python for smoother integration with the LiteFarm application. This shift ensures better compatibility with the app’s existing framework and offers more accessibility than the original Holos software.
- (2) **Enhanced Flexibility:** We aim to enhance the Holos model for parameter sampling from external data sources and user-defined distributions. This adjustment increases flexibility in model parameterization, adapting to diverse global agricultural environments. Including external data fits our partner’s global research scope and mitigates data scarcity in under-researched regions.

- (3) **Improving Reliability through Sensitivity Analysis:** With the enhanced model, we conduct sensitivity analyses to understand how input variations affect GHG emission estimates. This process is vital for identifying and minimizing uncertainties, thereby helping researchers pinpoint research gaps and provide farmers with accurate, reliable data for informed decision-making.

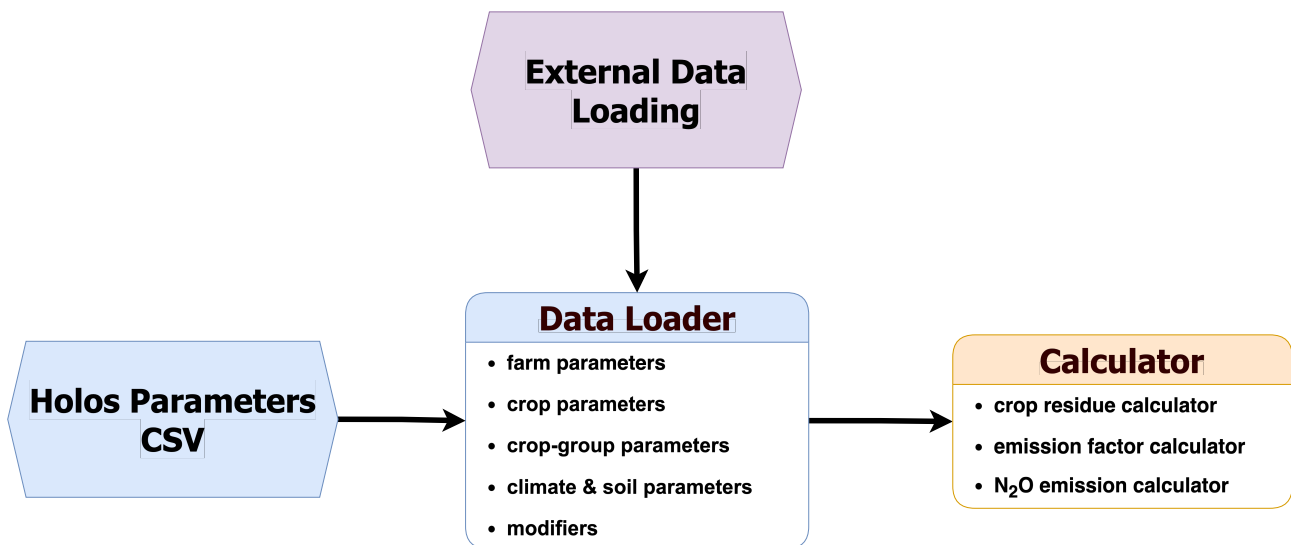
### 3 Data science methods

#### 3.1 Model development: PyHolos

We developed the PyHolos model to calculate nitrous oxide emissions from crop residues. After thoroughly reviewing the original Holos model code and technical reports, we identified equations with 23 parameters required for calculation (see details in Appendix 8.1 and 8.2).

##### 3.1.1 Model Reconstruction

To replicate the original Holos outputs, we developed a Python-based version of the model, named PyHolos. This model consists of two key components: a Data Loader and a Calculator. During development, each component is divided into smaller, manageable steps to improve the model's structure and functionality. The Data Loader imports parameters by organizing them into specific categories, while the Calculator processes these values in sequential steps. This modular approach not only facilitates the storage of intermediate results but also enables validation of the calculation results against the original Holos outputs (Figure 1).



**Figure 1:** PyHolos Model Architecture. This diagram outlines the PyHolos workflow where the Data Loader retrieves predefined Holos parameters from CSV files or pulls in external data as required by users. Parameters are systematically categorized into five groups: farm, crop, crop-group, climate and soil, and modifiers (see Appendix 8.1). The Calculator component subsequently processes these parameters to perform calculations on crop residue, emission factors, and total  $N_2O$  emissions.

##### 3.1.2 Enhanced Flexibility

While PyHolos uses Holos's original fixed parameters for standard operations, we also enhanced its flexibility. We have added modules that retrieve external data, supporting advanced analytical meth-

ods such as bootstrapping and sampling (Figure 1).

For climate parameters, we use API calls to access the National Aeronautics and Space Administration/Prediction of World Wide Energy Resources (NASA/POWER; <http://power.larc.nasa.gov>), and for soil data, we use the Harmonized World Soil Database (Nachtergaele et al. 2023). These enhancements offer global coverage and refined data resolution, boosting the model’s accuracy and relevance.

Additionally, we have introduced customizable modules for setting crop and modifier parameters. This flexibility allows users with domain knowledge to tailor these parameters to their research needs.

## 3.2 Analysis

### 3.2.1 Validation of PyHolos Outputs

To validate PyHolos’ accuracy, we compared its outputs with those of the original Holos software, using identical input parameters across various crops and multiple yield scenarios.

### 3.2.2 Sensitivity Analysis Using PyHolos

We conducted a sensitivity analysis using a hypothetical dataset of 105 farms, each growing three different crop types. For each crop at each farm, we generated 100 samples for each parameter to assess how different parameters affect the model’s  $N_2O$  estimations.

Unlike the original Holos model, which uses average climate and soil parameters from broad regions known as ecodistricts, PyHolos samples point-specific data from each farm’s corresponding ecodistrict. This approach assesses uncertainties associated with the Holos model and provides a detailed view of how environmental factors affect calculation. For other fixed variables in Holos where domain knowledge is lacking, we varied parameters by  $\pm 25\%$  using a uniform distribution.

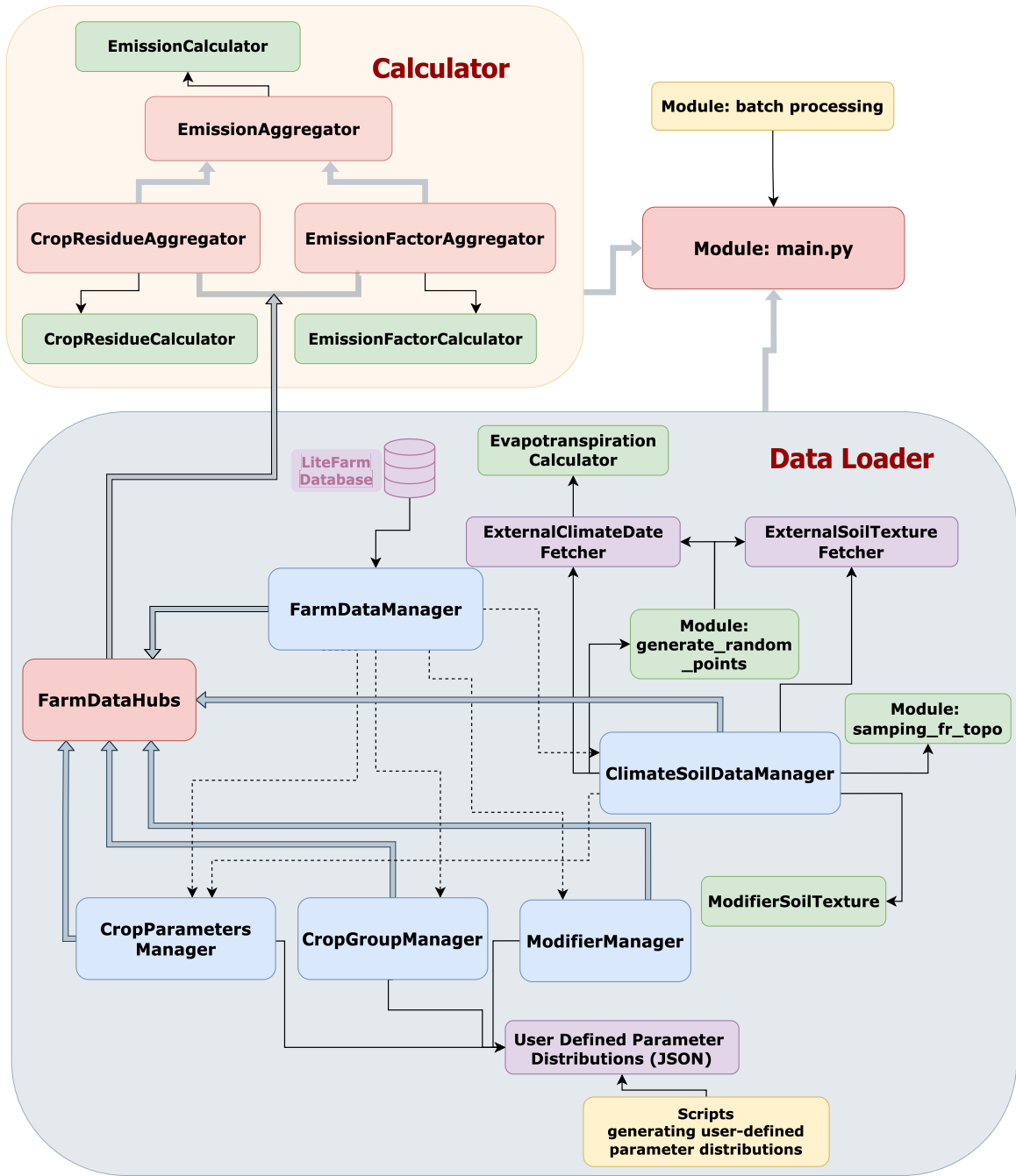
## 4 Data product and results

### 4.1 Data product - PyHolos

The PyHolos model uses predefined parameters to replicate original Holos outputs and incorporates user-defined variables along with external climate and soil data for greater flexibility (Figure 2). It calculates direct  $N_2O$  emissions from crop residue in two modes: Farmer Mode and Scientific Mode. The PyHolos program is operated via command line, allowing users to specify the desired mode—Farmer or Scientific—and configure simulations and other settings.

#### 4.1.1 Model reconstruction: Farmer Mode

Farmer Mode delivers a single, definitive output, ideal for users seeking straightforward estimates, such as actual farmer users. When utilizing default Holos parameters, this mode aims to replicate the original Holos results using identical inputs such as farm location, crop type, yield, and area. Additionally, Farmer Mode supports the incorporation of external data to override standard Holos parameters, offering enhanced flexibility. For example, LiteFarm researchers can input their own research data or apply more precise parameters values to refine the estimation. The results are formatted in JSON as a simple dictionary.



**Figure 2: PyHolos Model Structure Overview.** This diagram details the PyHolos workflow, structured into distinct components and modules for efficient data handling and processing. The Data Loader (gray shaded area) retrieves predefined Holos parameters from CSV files or sources external data as needed. The Calculator (light-yellow shaded area), processes these parameters to calculate crop residue, emission factors, and total  $N_2O$  emissions. The data loader manages farm, crop, climate, soil, and modifier parameters across five modules (blue boxes), which also access external climate and soil data and user-defined parameter distributions (purple boxes). Modules containing supporting classes and functions are indicated by green boxes. The modules/classes invoked by main.py, our main program script, is shown in salmon-colored boxes. Supporting scripts for defining parameter distribution and batch processing reside outside the main program folder `src/`, in a directory `scripts/` on our Github repository (yellow boxes). The main data flows and processes are depicted by thick gray-blue arrows, while black solid lines show module interactions, and black dashed lines indicate the use of farm data across modules. Each box in the diagram contains a hyperlink that directs to the corresponding module, class, or script on our PyHolos Github repo. For a detailed view including key attributes and methods for each class, please refer to the full version of this diagram available [here](#).

### 4.1.2 Enhanced flexibility: Scientific Mode

Scientific Mode runs simulations by sampling across parameter distributions, generating multiple outputs per parameter. This mode is suitable for users conducting in-depth analyses. Results are formatted in JSON as nested dictionaries to accommodate multiple simulation results per parameter.

A key application of this mode is predicting and understanding emissions extremes and trends. For instance, the integrated external climate data source captures real-time changes, enabling temporal analyses that track emissions trends over time.

## 4.2 Results

### 4.2.1 PyHolos model validation

We compared the calculation outputs of PyHolos and Holos for two crop types: potatoes (a root crop) and soybeans (an annual crop). These two crop types use different  $N_2O$  calculation equations and parameter values (see details in Appendix 8.1 and 8.2). While Holos software provides some intermediate results for crop residue calculation, aiding in comparison, it lacks intermediate data on emission factor calculation, hindering identification of sources of discrepancies in this aspect of the calculation (see components in the “calculator” module in Figure 2).

For potatoes, PyHolos accurately replicates Holos’ results for above and below-ground carbon inputs (Table 1). However, there are minor differences in nitrogen from crop residues (Table 1). The discrepancies become larger in the final  $N_2O$  emissions, calculated by multiplying the nitrogen from crop residues by the emission factor (Table 1). The origins of these discrepancies are unclear. According to equation 2.5.6-9 in Holos model (also see Appendix 8.2.1 Total Crop Residues), nitrogen from crop residues should equal the sum of above and below-ground inputs multiplied by farm area, as calculated by PyHolos. Yet, Holos results suggest undocumented adjustments in how farm area data is used in calculation. Furthermore, the enlarged discrepancies in final  $N_2O$  emissions indicate differences in emission factor calculations. However, the absence of intermediate results from Holos in emission factor calculations hinders tracing the sources of these discrepancies.

Table 1: Results comparison between Holos and PyHolos for potatoes (a root crop) across different yields.

Table 1

Source	Crop	Yield	Aboveground Carbon Input	Belowground Carbon Input	N2O-N From Crop Residues	N2O Direct CO2e
Holos	Potatoes	10000	365.32	15.28	0.9	521.08
Software Holos	Potatoes	30000	1095.96	45.86	2.7	1293.98
Software Holos	Potatoes	50000	1826.6	76.43	4.5	2066.88
Software PyHolos	Potatoes	10000	365.32	15.29	0.83	354.47
PyHolos	Potatoes	30000	1095.96	45.86	2.48	1063.4
PyHolos	Potatoes	50000	1826.6	76.43	4.13	1772.34

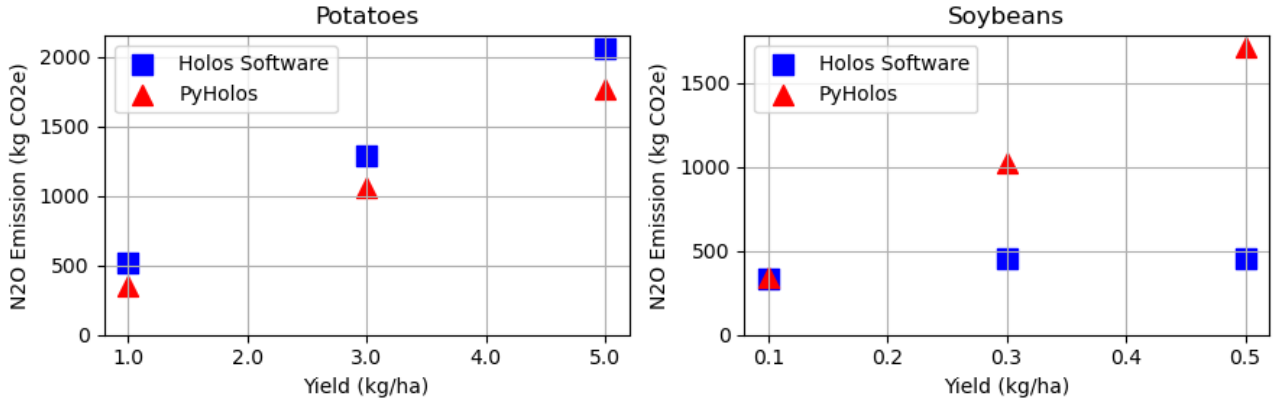
Unlike potatoes, which exhibit consistent above- and below-ground input estimations between PyHolos and Holos, soybeans show discrepancies across all parameters: above- and below-ground inputs, nitrogen from crop residues, and final  $N_2O$  emissions (Table 2). Besides the previously noted discrepancies in farm area usage and emission factor calculations, the sources of other discrepancies remain elusive.

The Holos technical report indicates a linear relationship between crop yield and  $N_2O$  emissions, a pattern observed in root crops like potatoes (Table 1 and Figure 3). However, Holos’ outputs for soybeans do not demonstrate this linearity. This inconsistency may stem from ongoing updates and developments in the Holos software, which may not be fully documented.

Table 2: Results comparison between Holos and PyHolos for soybeans (an annual crop) across different yields.

Table 2

Source	Crop	Yield	Aboveground Carbon Input	Belowground Carbon Input	N2O-N From Crop Residues	N2O Direct CO2e
Holos	Soybeans	1000	912.77	283.75	0.47	336.12
Software Holos	Soybeans	3000	1317.41	532.97	0.75	455.35
Software Holos	Soybeans	5000	1116.49	646.55	0.74	453.66
Software PyHolos	Soybeans	1000	598.71	312.94	0.8	342.3
PyHolos	Soybeans	3000	1796.12	938.81	2.39	1026.9
PyHolos	Soybeans	5000	2993.53	1564.68	3.99	1711.49

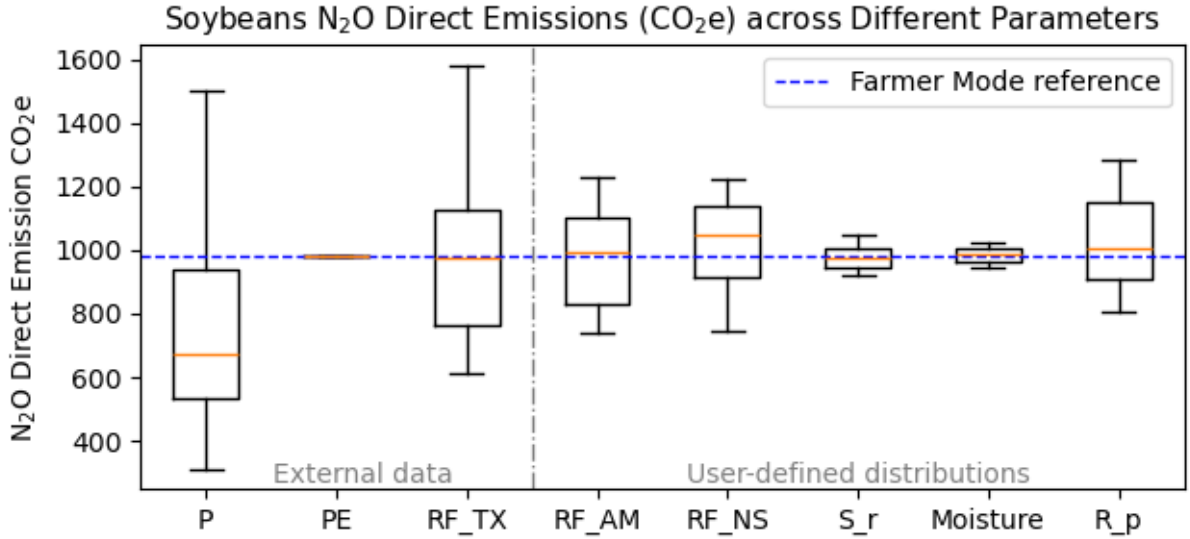


**Figure 3:** Emission Estimates for Soybean across Different Yields. Based on current calculation equations (see Appendix 8.2), emissions should linearly correlate with crop yields, as evidenced by PyHolos results. However, Holos software shows this linearity for potatoes but not for soybeans.

#### 4.2.2 Sensitivity Analysis Results

We have analyzed model outputs across various parameters for different crops and farms using PyHolos’ Scientific Mode. Here we focus on soybean  $N_2O$  emission estimates at a specific farm, illustrating the sensitivity of emission estimates to parameter changes (Figure 4). Each boxplot represents the variability induced by a single parameter while keeping other parameters constant.

For parameters using external data sources, we observed a significant deviation between the Farmer Mode’s definitive estimations (i.e., Farmer Mode reference) and the median value of all samples from the ecodistrict, as observed with precipitation (P) (Figure 4). Thus, using ecodistrict averages in original Holos’s method may fail to accurately represent the specific conditions of farms located at the distribution extremes. Our finding underscores the importance of considering local conditions when refining  $N_2O$  emission estimates.



**Figure 4:**  $N_2O$  Direct Emissions ( $CO_2e$ ) across different input parameters. The blue dashed line indicates the baseline emission value under Farmer Mode. Parameters sampled using external data sources (i.e., P, PE, RF\_TX) are separated by a vertical dashed line from those sampled over user-defined distributions (RF\_AM, RF\_NS, S\_r, Moisture, R\_p). Detailed parameter explanations can be found in Table 3 in Appendix 8.1.

For user-defined parameters, the Farmer Mode reference aligns with the median values, as the sampling distributions are default to uniform distributions with  $\pm 25\%$  uncertainty (Figure 4). Nonetheless, these results remain insightful. Different parameters not only vary in magnitude but are also utilized differently in the calculation process—some are used in exponential equations, others in linear equations. Therefore, even though all parameters are sampled uniformly, their impact on the final  $N_2O$  estimation varies. This variability provides critical insights into how sensitive  $N_2O$  emissions to parameter changes, aiding in identifying areas with large uncertainties and research gaps.

### 4.3 Advantages and limitations

PyHolos enhances the original Holos model by replicating its ability to provide definitive  $N_2O$  emission estimations and integrating external data sources to refine these estimates and evaluate uncertainties. Specifically, Farmer Mode aims to replicate Holos’ results using its original parameters, fulfilling our partner’s primary need to assist farmers with carbon emission estimations. Additionally, integrating external data enables modifying Holos’ parameters, meeting our partner’s needs for refined calculations with their own research data. Furthermore, Scientific Mode supports in-depth analyses by sampling across parameter distributions. This functionality helps pinpoint and reduce uncertainties in estimations, thereby improving calculation accuracy and supporting farmers’ decision-making.

The command-line operation of PyHolos not only facilitates integration with the LifeFarm framework but also offers time efficiency compared to the manual input methods used in the Holos software. For example, in our sensitivity analysis, calculating  $N_2O$  emission estimations for 105 farms, each with three crops across 100 runs, took 13 hours—primarily delayed by NASA POWER API calls. In contrast, using Holos software for the same task would require approximately 10.9 days, assuming continuous operation and with each record taking 30 seconds to complete.

Currently, PyHolos focuses solely on estimating  $N_2O$  emissions from crop residue and has not yet integrated other greenhouse gas emissions components from the Holos model. Given that Holos is still undergoing development and updates, continued efforts are necessary to fully align PyHolos with our partners’ needs. PyHolos is thoroughly documented in both methodology and usage. Its modular



design allows for the addition of new functions and modules as needed, setting the groundwork for continuous development.

## 5 Conclusions and recommendations

Our project with LiteFarm aimed to address the challenge of calculating greenhouse gas emissions to support sustainable farming practices. We developed PyHolos, a Python-based model that not only replicates but also improves upon the original Holos model, offering  $N_2O$  emission estimates from crop residues and integration with the LiteFarm framework.

PyHolos fulfills our partner’s fundamental needs for emission estimations and introduces enhancements over the original Holos model. These improvements include the integration of external or user-specific data for greater calculation accuracy and the capacity to conduct uncertainty analyses. As a result, PyHolos provides a useful tool for both farmers and researchers aiming to mitigate the environmental impact of agriculture.

While PyHolos has proven effective in estimating  $N_2O$  emissions, it currently does not encompass other types of greenhouse gas emissions, limiting its wider utility. Additionally, minor discrepancies exist between PyHolos and the Holos software, likely due to ongoing updates in Holos yet outdated documentation.

Future development efforts of PyHolos should focus on expanding the model’s capabilities to include other greenhouse gases and refining its integration with the LiteFarm application. By continuing to enhance data accuracy and model flexibility, PyHolos will further empower farmers and researchers to make data-driven decisions. We recommend maintaining collaboration with the Holos team to ensure that PyHolos aligns with requirements of the LiteFarm team in greenhouse gas emission calculation and sustainable agricultural practices.

## 6 Acknowledgements

We sincerely thank our mentor, Dr. Simon Goring, for his invaluable guidance and support throughout this project. We also thank our partner, the LiteFarm team at UBC. Special thanks to Dr. Khanh Dao Duc and Professor Hannah Wittman for providing us with the opportunity to work on this project and for their support throughout this journey.

## 7 References

- Agriculture & Agri-Food Canada. 2022. “Holos.” <https://github.com/holos-aafc/Holos>; GitHub.
- Barbato, Clare T, and Aaron L Strong. 2023. “Farmer Perspectives on Carbon Markets Incentivizing Agricultural Soil Carbon Sequestration.” *Npj Climate Action* 2 (1): 26. <https://doi.org/10.1038/s44168-023-00055-4>.
- Jantke, Kerstin, Martina J Hartmann, Livia Rasche, Benjamin Blanz, and Uwe A Schneider. 2020. “Agricultural Greenhouse Gas Emissions: Knowledge and Positions of German Farmers.” *Land* 9 (5): 130. <https://doi.org/10.3390/land9050130>.
- Johnson, Jane M-F, Alan J Franzluebbers, Sharon Lachnicht Weyers, and Donald C Reicosky. 2007. “Agricultural Opportunities to Mitigate Greenhouse Gas Emissions.” *Environmental Pollution* 150 (1): 107–24. <https://doi.org/10.1016/j.envpol.2007.06.030>.
- Nachtergaele, Freddy, Harrij van Velthuisen, Luc Verelst, Dave Wiberg, Matieu Henry, Frederica Chiozza, Yusuf Yigini, et al. 2023. “Harmonized World Soil Database Version 2.0.” Food; Agriculture Organization of the United Nations. <https://www.fao.org/soils-portal/data-hub/soil-maps-and-databases/harmonized-world-soil-database-v20/en/>.

## 8 Appendix

### 8.1 Parameters for Crop Residue Nitrogen Calculation

Table 3: Parameters used in Holos  $N_2O$  emission calculation

Parameter groups	Parameters	Explanation	Data source
Farm	Area	Numerical, total area of the farm (ha)	LiteFarm
Farm	Yield	Numerical, the estimated yield ( $kg/ha$ )	LiteFarm
Crop	Moisture	Moisture content of product (%)	Holos default/External information
Crop	$N_p$	N concentration in the product ( $kg\ kg^{-1}$ )	Holos default/External information
Crop	$N_s$	N concentration in the straw ( $kg\ kg^{-1}$ )	Holos default/External information
Crop	$N_r$	N concentration in the roots ( $kg\ kg^{-1}$ )	Holos default/External information
Crop	$N_e$	N concentration in the extra root material ( $kg\ kg^{-1}$ )	Holos default/External information
Crop	$R_s$	Relative biomass allocation coefficient for straw	Holos default/External information
Crop	$R_p$	Relative biomass allocation coefficient for product	Holos default/External information
Crop	$R_r$	Relative biomass allocation coefficient for roots	Holos default/External information
Crop	$R_e$	Relative biomass allocation coefficient for extra-root material	Holos default/External information
Crop group	$C_{concentration}$	Carbon concentration of all plant parts ( $kg\ kg^{-1}$ )	Holos default/External information
Crop group	$S_p$	Percentage of product yield returned to soil	Holos default/External information

Parameter groups	Parameters	Explanation	Data source
Crop group	$S_s$	Percentage of straw returned to soils	Holos default/External information
Crop group	$S_r$	Percentage of roots returned to soil	Holos default/External information
Climate & soil	$P_i$	Annual growing season precipitation (May – October), in ecodistrict “i” (mm)	Holos default/NASA POWER
Climate & soil	$PE$	Growing season potential evapotranspiration, by ecodistrict (May – October)	Holos default/NASA POWER
Climate & soil	$FR\_Topo_i$	Fraction of land occupied by lower portions of landscape	Holos default
Modifiers / soil	$RF\_TX_i$	Weighted modifier which provides a correction of the $EF_{Topo}$ in ecodistrict “i” based on the soil texture	Holos default/HWSD 2.0
Modifiers	$RF\_CS$	Reduction factor for Cropping System	Holos default/External information
Modifiers	$RF\_NS$	N source modifier RF_NS (SN = Synthetic Nitrogen; ON = Organic Nitrogen; CRN = Crop Residue Nitrogen)	Holos default/External information
Modifiers	$RF\_AM$	Reduction factor based on application method, only applicable to calculations of EF specific for SN	Holos default/External information
Modifiers	$RF\_Till$	Tillage modifier (Conservation or Conventional Tillage)	Holos default/External information

## 8.2 Equations for Crop Residue Nitrogen Calculation

The equations outlined are sourced from the [AAFC Technical Report: Holos V4.0 Algorithm Document REVIEW VERSION 22 Jan 2024](#). The numbers in parentheses following each equation indicate the specific equation numbers in the Holos documentation. Each equation includes a reference link to the corresponding line in the Holos code on GitHub.

### 8.2.1 Total Crop Residues

$$N_{crop\_residues} = (AboveGround_{residue\_N} + BelowGround_{residue\_N}) \times area \quad (2.5.6-9)$$

**Variables:**

- $N_{crop\_residues}$ : N (nitrogen) inputs from crop residue returned to soil ( $kg\ N$ )
- $AboveGround_{residue\_N}$ : Aboveground residue nitrogen ( $kg\ N\ ha^{-1}$ )
- $BelowGround_{residue\_N}$ : Belowground residue nitrogen ( $kg\ N\ ha^{-1}$ )
- $area$ : Area of crop ( $ha$ )

#### 8.2.1.1 AboveGround Residue Nitrogen

$$AboveGround_{residue\_N} = [Grain_N + Straw_N] \quad (2.5.6-6)$$

**Holos Code Reference:** [Holos GitHub - Equation 2.5.6-6](#)

**Variables:**

- $AboveGround_{residue\_N}$ : Nitrogen in above-ground crop residues ( $kg\ N$ )
- $Grain_N$ : Nitrogen content of the grain returned to the soil ( $kg\ N\ ha^{-1}$ )
- $Straw_N$ : Nitrogen content of the straw returned to the soil ( $kg\ N\ ha^{-1}$ )

#### 8.2.1.2 BelowGround Residue Nitrogen

For annual plants:

$$BelowGround_{residue\_N} = [Root_N + Exudate_N] \quad (2.5.6-7)$$

**Holos Code Reference:** [Holos GitHub - Equation 2.5.6-7](#)

For perennial plants:

$$BelowGround_{residue\_N} = [S_r \times Root_N] + Exudate_N \quad (2.5.6-8)$$

**Holos Code Reference:** [Holos GitHub - Equation 2.5.6-8](#)

**Variables:**

- $BelowGround_{residue\_N}$ : Belowground residue nitrogen ( $kg\ N\ ha^{-1}$ )
- $Root_N$ : Nitrogen content of the root returned to the soil ( $kg\ N\ ha^{-1}$ )
- $Exudate_N$ : Nitrogen content of the exudates returned to the soil ( $kg\ N\ ha^{-1}$ )
- $S_r$ : Root turnover fraction

### 8.2.1.3 Grain\_N

$$Grain_N = \frac{C_{p\_to\_soil}}{0.45} \times N_p \quad (2.5.6-2)$$

Holos Code Reference: [Holos GitHub - Equation 2.5.6-2](#)

Variables:

- $Grain_N$ : Nitrogen content of the grain returned to the soil ( $kg\ N\ ha^{-1}$ )
- $C_{p\_to\_soil}$ : Carbon input from product ( $kg\ ha^{-1}$ )
- $N_p$ : N concentration in the product ( $kg\ kg^{-1}$ ) [[Holos Table 7](#)]

### 8.2.1.4 Straw\_N

$$Straw_N = \frac{C_s}{0.45} \times N_s \quad (2.5.6-3)$$

Holos Code Reference: [GitHub - Equation 2.5.6-3](#)

Variables:

- $Straw_N$ : Nitrogen content of the straw returned to the soil ( $kg\ N\ ha^{-1}$ )
- $C_s$ : Carbon input from straw ( $kg\ ha^{-1}$ )
- $N_s$ : N concentration in the straw ( $kg\ kg^{-1}$ ) [[Holos Table 7](#)]

### 8.2.1.5 Root\_N

$$Root_N = \frac{C_r}{0.45} \times N_r \quad (2.5.6-4)$$

Holos Code Reference: [GitHub - Equation 2.5.6-4](#)

Variables:

- $Root_N$ : Nitrogen content of the root returned to the soil ( $kg\ N\ ha^{-1}$ )
- $C_r$ : Carbon input from roots ( $kg\ ha^{-1}$ )
- $N_r$ : N concentration in the roots ( $kg\ kg^{-1}$ ) [[Holos Table 7](#)]

### 8.2.1.6 Exudate\_N

$$Exudate_N = \frac{C_e}{0.45} \times N_e \quad (2.5.6-5)$$

Holos Code Reference: [GitHub - Equation 2.5.6-5](#)

Variables:

- $Exudate_N$ : Nitrogen content of the exudates returned to the soil ( $kg\ N\ ha^{-1}$ )
- $C_e$ : Carbon input from extra-root material ( $kg\ ha^{-1}$ )
- $N_e$ : N concentration in the extra root material ( $kg\ kg^{-1}$ ) [[Holos Table 7](#)]

### 8.2.1.7 C<sub>p</sub>: Above and Belowground Residue Input

$$C_p = [(yield + yield \times \frac{S_p}{100}) \times (1 - \frac{moisture\ content}{100})] \times Carbon\ concentration \quad (2.1.2-6)$$

Holos Code Reference: [GitHub - Equation 2.1.2-6](#)

Variables:

- $C_p$ : Plant C (carbon) in agricultural product ( $kg\ ha^{-1}$ )
- $yield$ : Crop yield ( $kg\ wet\ weight\ ha^{-1}$ , default provided, user override)
- $S_p$ : Percentage of product yield returned to soil (user override)
- $moisture\ content$ : Moisture content (%) of crop product [[Holos Table 7](#)]
- $Carbon\ concentration$ : C concentration of all plant parts ( $kg\ kg^{-1}$ )

### 8.2.1.8 C<sub>p\_to\_soil</sub>: Carbon input from product

$$C_{p\_to\_soil} = C_p \times \frac{S_p}{100} \quad (2.1.2-7)$$

Holos Code Reference: [GitHub - Equation 2.1.2-7](#)

Variables:

- $C_{p\_to\_soil}$ : Carbon input from product ( $kg\ ha^{-1}$ )
- $C_p$ : Plant C (carbon) in agricultural product ( $kg\ ha^{-1}$ )
- $S_p$ : Percentage of product yield returned to soil

### 8.2.1.9 C<sub>s</sub>: Carbon Input from Straw

$$C_s = C_p \times \frac{R_s}{R_p} \times \frac{S_s}{100} \quad (2.1.2-8)$$

Holos Code Reference: [GitHub - Equation 2.1.2-8](#)

Variables:

- $C_s$ : C (carbon) input from straw ( $kg\ ha^{-1}$ )
- $C_p$ : Plant C (carbon) in agricultural product ( $kg\ ha^{-1}$ )
- $R_s$ : Relative biomass allocation coefficient for straw [[Holos Table 7](#)]
- $R_p$ : Relative biomass allocation coefficient for product [[Holos Table 7](#)]
- $S_s$ : Percentage of straw returned to soil (user override)

### 8.2.1.10 C<sub>r</sub>: Carbon input from Root

$$C_r = C_p \times \frac{R_r}{R_p} \times \frac{S_r}{100} \quad (2.1.2-9)$$

Holos Code Reference: [GitHub - Equation 2.1.2-9](#)

Variables:

- $C_r$ : C (carbon) input from roots ( $kg\ ha^{-1}$ )
- $C_p$ : Plant C (carbon) in agricultural product ( $kg\ ha^{-1}$ )
- $R_r$ : Relative biomass allocation coefficient for roots [[Holos Table 7](#)]

- $R_p$ : Relative biomass allocation coefficient for product [Holos Table 7]
- $S_r$ : Percentage of roots returned to soil (user override)

#### 8.2.1.11 C<sub>e</sub>: Carbon Input from Exudate

$$C_e = C_p \times \frac{R_e}{R_p} \quad (2.1.2-9)$$

Holos Code Reference: [GitHub - Equation 2.1.2-9](#)

Variables:

- $C_e$ : C (carbon) input from extra-root material ( $kg\ ha^{-1}$ )
- $C_p$ : Plant C (carbon) in agricultural product ( $kg\ ha^{-1}$ )
- $R_e$ : Relative biomass allocation coefficient for extra-root material [Holos Table 7]
- $R_p$ : Relative biomass allocation coefficient for product [Holos Table 7]

### 8.2.2 Emission Factor

#### 8.2.2.1 Calculate Base Emission Factor - Ecodistrict-level Emission Factor

$$EF\_CT_{i,P>PE} = \exp^{0.00558 \times P_i - 7.7} \quad (2.5.1-1)$$

$$EF\_CT_{i,P \leq PE} = \exp^{0.00558 \times PE_i - 7.7} \quad (2.5.1-2)$$

Holos Code Reference: [GitHub - Equation 2.5.1](#)

Variables:

- $EF\_CT_i$ : Ecodistrict-level emission factor ( $kg\ N_2O-N\ (kg\ N)^{-1}$ )
- $P_i$ : Annual growing season precipitation (May – October), in ecodistrict “i” (mm)
- $PE$ : Growing season potential evapotranspiration, by ecodistrict (May – October) (mm)

#### 8.2.2.2 Calculate Emission Factor Adjustment Due to Position in Landscape/Topography

For humid environments  $P/PE > 1$ :

$$EF\_Topo_i = EF\_CT_{i,P > PE} \quad (2.5.2-1)$$

For non-irrigated sites and dry environments  $P/PE \leq 1$ :

$$EF\_Topo_i = (EF\_CT_{i,P < PE} \times FR\_Topo_i) + [EF\_CT_{i,P > PE} \times (1 - FR\_Topo_i)] \quad (2.5.2-2)$$

For irrigated sites and  $P < PE$ :

$$EF\_Topo_i = EF\_CT_{i,P \leq PE} \quad (2.5.2-3)$$

Holos Code Reference: [GitHub - Equation 2.5.2](#)

Notes:

For non-irrigated sites and  $P/PE \leq 1$ , the fraction of low-lying land and depressions is calculated with the actual PE ( $EF\_CT_{i,P \leq PE}$ ), and the remainder of the land with the standard EF ( $EF\_CT_{i,P > PE}$ ).

For irrigated sites, the assumption is that the irrigation amount is equal to  $PE - P$ , thus making  $P = PE$ .

**Variables:**

- $EF\_Topo_i$ :  $N_2O$  emission factor adjusted due to position in landscape and moisture regime ( $kg\ N_2O-N$ )
- $FR\_Topo_i$ : Fraction of land occupied by lower portions of the landscape

### 8.2.2.3 Calculate Emission Factor Adjustment Due to Soil Texture

$$EF\_Base_i = (EF\_Topo_i \times RF\_TX_i) \times \frac{1}{0.645} \quad (2.5.3-2)$$

**Holos Code Reference:** [GitHub - Equation 2.5.3-2](#)

**Variables:**

- $RF\_TX_i$ : Weighted modifier which provides a correction of the  $EF\_Topo$  in ecodistrict “i” based on the soil texture.
- $EF\_Base_i$ : A function of the three factors that create a base ecodistrict-specific value that accounts for the climatic, topographic, and edaphic characteristics of the spatial unit for lands.
- $\frac{1}{0.645}$ : Fraction of growing season emissions of total annual emissions (Pelster et al. 2022, in prep.).

### 8.2.2.4 Calculate Emission Factor

$$EF_{i,k,l,m,n} = EF\_Base_i \times RF\_NS_k \times RF\_Till_l \times RF\_CS_m \times RF\_AM \quad (2.5.4-1)$$

**Variables:**

- $RF\_NS_k$ : Reduction factor for N source,  $RF\_NS_k$  (SN = Synthetic Nitrogen; ON = Organic Nitrogen; CRN = Crop Residue Nitrogen). For CRN, the value is 0.84. [\[Holos Lookup Function\]](#)
- $RF\_Till_l$ : Reduction factor for tillage practice [\[Holos Lookup Function\]](#)
- $RF\_CS_m$ : Reduction factor for cropping system. For annual cropping system, the value is 1. For Perennial systems, the value is 0.19 [\[Holos Lookup Function\]](#)
- $RF\_AM$ : Reduction factor based on application method, only applicable to calculations of EF specific for SN. For other N source, the value is 1 [\[Holos Lookup Function\]](#)

## 8.2.3 Emission

### 8.2.3.1 Calculate Direct Nitrous Oxide from Inputs

$$N_2O - N_{CRNdirect(t,field,n)} = N_{CropResidues(t,field,n)} \times EF_{i,CRN,l,m,n} \quad (2.6.5-2)$$

**Holos Code Reference:** [GitHub - Equation 2.6.5-2](#)

**Variables:**

- $N_2O - N_{CRNdirect(t,field,n)}$ : Direct  $N_2O$  emissions ( $kg\ N_2O-N\ ha^{-1}$ ) resulting from crop residues and N mineralization on field n in year t.
- $N_{CropResidues(t,field,n)}$ : Amount of crop residues on field n in year t.
- $EF_{i,CRN,l,m,n}$ : Emission factor for crop residue nitrogen specific to the conditions i, l, m, and n.



### 8.2.3.2 Calculate Emission for Each Field, Crop, and Year

$$N_2O - N_{direct} = N_2O - N_{SNdirect} + N_2O - N_{CRNdirect} + N_2O - N_{CRNmindirect} + N_2O - N_{ONdirect}$$

#### Notes:

Currently, only Nitrogen Direct Emission from crop residues is calculated, as this only accounts for crop-related factors and not livestock, fertilizer, and manure.

#### Variables:

- $N_2O - N_{SNdirect}$ :  $N_2O$  emissions ( $kg\ N_2O-N\ kg^{-1}\ N\ ha^{-1}$ ) resulting from fertilizer application
- $N_2O - N_{CRNdirect}$ :  $N_2O$  emissions ( $kg\ N_2O-N\ kg^{-1}\ N\ ha^{-1}$ ) resulting from crop residues and N mineralization
- $N_2O - N_{CRNmindirect}$ :  $N_2O$  emissions ( $kg\ N_2O-N\ kg^{-1}\ N\ ha^{-1}$ ) resulting from N mineralization
- $N_2O - N_{ONdirect}$ :  $N_2O$  emissions ( $kg\ N_2O-N\ kg^{-1}\ N\ ha^{-1}$ ) resulting from organic fertilizers

### 8.2.3.3 Convert $N_2O-N$ to $N_2O$

$$N_2O = N_2O - N \times \frac{44}{28}$$

#### Variables:

- $\frac{44}{28}$ : Conversion factor from  $N_2O-N$  to  $N_2O$  based on molecular mass

### 8.2.3.4 Calculate $CO_2$ Equivalent of $N_2O$ Emissions

$$CO_2e = N_2O \times 273$$

#### Notes:

Holos uses 273 as the Global Warming Potential value for  $N_2O$ . [Holos Reference](#)

#### Variables:

- 273: Global Warming Potential for  $N_2O$  compared to  $CO_2$