Supplementary Information for

Optimized Feed-Forward Neural Networks to Address CO_2-equivalent Emissions Data Gaps – Application to Emissions Prediction for Unit Processes of Fuel Life Cycles Inventories for Canadian Provinces

Sayyed Ahmad Khadem[†‡], Farid Bensebaa[†], and Nathan Pelletier[‡]*

[†]Energy, Mining and Environment, National Research Council Canada, 1200 Montreal Road, Ottawa, ON K1A 0R6, Canada
[‡] Irving K. Barber Faculty of Science, The University of British Columbia, 3247 University Way, Kelowna, BC V1V 1V7, Canada
*Corresponding author. E-mail addresses: sayyed.khadem@ubc.ca and sayyedahmad.khadem@nrc-cnrc.gc.ca

Table of Contents

Supplementary Note S1. Initialization-dependency of the training stage

To show the initialization dependence of network performance, the following experiment was carried out. For the "Fuel production" unit process, the network was trained with 20 random initial points. Supplementary Figure S2 presents the results of this experiment, confirming that the optimizer used for finding optimal model parameters (i.e. weights and biases) can depend on the initial point.

Supplementary Note S2. Optimal Selection of Activation Function, Optimizer, and Loss Function Based on Heuristics

To show the optimality of ReLU (Rectified Linear Unit) as an activation function, we train the topologically optimal FNNs with ten different activation functions available in Keras library (Chollet, 2015) and then compare the resulting performances. To save time, we only evaluate the top three contributors, accounting for 81 % overall contribution in fuel life cycles.

As can be seen in Supplementary Figure S3 and Supplementary Table S3-S5, five out of nine activation functions are good, which are ReLU, SELU (Scaled Exponential Linear Unit), softsign, ELU (Exponential linear unit), and TanH (Hyperbolic tangent). However, ReLU generally yields optimal performance in comparison to the other activation functions. In other words, in our study, other activation functions are incapable of outperforming ReLU, hence confirming the optimality of our choice of ReLU for the activation function.

In a similar vein, we train the topologically optimal FNNs with eight different optimizers available in Keras library (Chollet, 2015) and then compare the resulting performances. Supplementary Figure S4 and Supplementary Table S6-S8 demonstrate that four out of eight optimizers are good; adam, Nadam, Adamax, and RMSprop. In particular, the adam optimizer leads to optimal FNN performances. Similarly, none of the optimizers noticeably outperform adam although, Nadam, Adamax, and RMSprop can also work as well as adam depending on the FNNs. We observe that optimization among adam, Nadam, Adamax, and RMSprop seems unnecessary

because they can end up with nearly the same FNN performances. This result (i.e. that optimization on certain hyperparameters is unnecessary) has also been reported in (Song et al., 2017). Hence, the analysis presented herein confirms that our choice of the adam method for the optimizer is optimal.

Regarding the optimal choice of the loss function, we also perform a systematic experiment to validate our *a posteriori* assumption on the optimality of MSE in our study. The networks are trained with three loss functions; MSE (Mean Squared Error), MAE (Mean Absolute Error), and Huber. Thereafter, the performances of the trained networks are measured by R^2 score. Supplementary Table S9 shows the results of this experiment, confirming the other loss functions are not superior to the chosen loss function, i.e. MSE.

Note that for all experiments mentioned above, in the training stage, the number of random initializations of model parameters is 50, meaning that each FNN is trained with 50 random model initializations and then the best performance is selected. As explained in the manuscript, this increases the chance of finding the global minimum of the non-convex loss function.

In a nutshell, the general optimality of ReLU and adam as activation function and optimizer, respectively, in FNNs has been reported and explained in (Goodfellow et al., 2016). Furthermore, the reason that MSE was chosen is that there are no outliers in the data. In this section, we also performed experiments with these three hyperparameters to assure their optimality in our work. Hence, we conclude that, in our study, ReLU, adam, and mse are optimal activation function, optimizer, and loss function, respectively, leading to optimal FNN performance.

Supplementary Note S3. Attributes impacts on identical networks

To compare the attributes impacts under identical hyperparameters (values are listed in Table 1) and hidden topology, one hidden topology is required to be chosen for each contributor. Consequently, the hidden topology of Scenario I is chosen the same as the optimal hidden topology

found for Scenario II. The reason lies in the fact that the optimal hidden topologies of Scenario II are shallower and, in consequence, less complex than that of Scenario I (see section 3.2). To save time, we only evaluate the top three contributors, accounting for 81 % of overall contributions to GHG emissions in fuel life cycles.

Supplementary Figure S5 shows the results of the comparison of Scenario I and Scenario II under identical hyperparameters and hidden topology. Each panel shows the results for the prediction of CO_2-eq emissions for the contributor named in the panel's title. Moreover, performances of twelve networks are indicated in each panel. From left to right, the first ten networks correspond to Scenario I with ten different initializations and their hidden topologies are the same as the optimal hidden topology obtained for Scenario II. Note that regarding the first ten networks, we train 50 networks with different random initializations, and then the top networks in terms of performance are illustrated in Supplementary Figure S5. The two networks on the right correspond to Scenario I and II with their optimal hidden topologies.

As seen in Supplementary Figure S5 for the prediction of CO_2-eq emissions for each contributor, Scenario II consistently outperforms Scenario I in terms of network performance under the condition that the hidden topology and hyperparameters are kept identical. Supplementary Figure S5 confirms the superiority of Scenario II compared to Scenario I. Additionally, Supplementary Figure S5 shows that the optimization carried out by GA is successful because, for each contributor, the performance of Scenario I becomes poorer if the hidden topology is non-optimal.

All in all, assessment of the attributes scenarios under identical hidden topologies unequivocally shows that Scenario I is incapable of reaching the accuracy obtained by Scenario II. The improvement is particularly noticeable for "Fuel production" and "Feedstock recovery".

Supplementary Note S4. GA validation

Supplementary Figure S6 illustrates the fitness evolution associated with the unit processes, as obtained through GA generations. The best fitness function, i.e. validation error, for each unit process drops sufficiently across generations. There is also no improvement during the last generations. In light of these two observations, at least local minima are successfully captured by the GA optimizer. The optimality of GA results can also be appreciated from the analysis carried out in Supplementary Note S3 where the performances obtained by non-optimal hidden topologies are less than optimal ones. Furthermore, Supplementary Note S5 presents learning curves for optimal ANNs, in which learning curves are acceptable due to the right network capacity obtained by GA, confirming the valid performance of GA. Taken altogether, the optimal hidden topologies proposed by GA are validated.

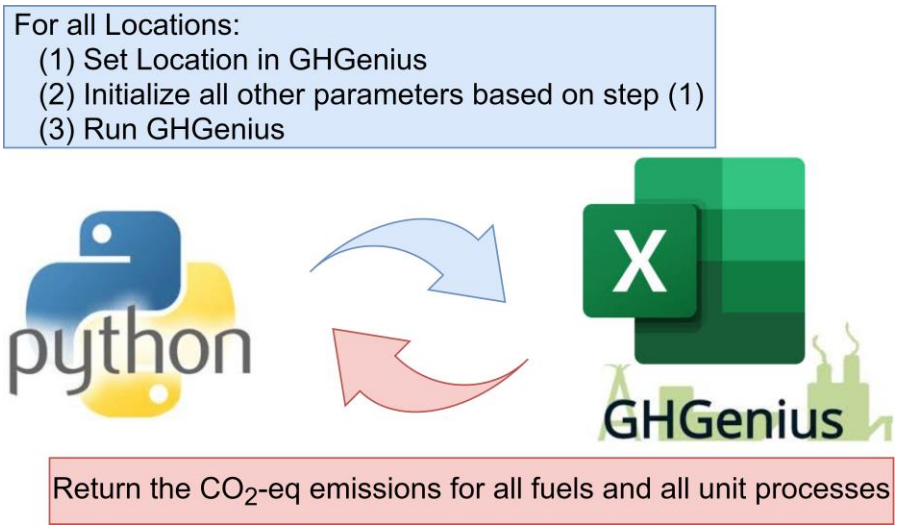
Supplementary Note S5. Learning Curves

The "early stopping" criterion is a regularization by which training stops if validation error increases for a certain number of epochs, known as "patience", as learning progresses. Supplementary Figure S7 illustrates the RMSE errors of the training and validation sets during training by the consideration that the "early stopping" regularization along with "patience" of 25 epochs is applied.

Supplementary Figure S7 shows that the "early stopping" approach with 25 epochs as "patience" is generally successful for regularization in our study to avoid overfitting although negligible overfitting can be seen in Supplementary Figure S7(c, j) as training errors decrease while validation errors remain almost unchanged. Another result obtained from Supplementary Figure S7 is that the networks are not underfitted since the loss functions associated with training sets drop sufficiently; hence, models learn while the epoch marches forward. Consequently,

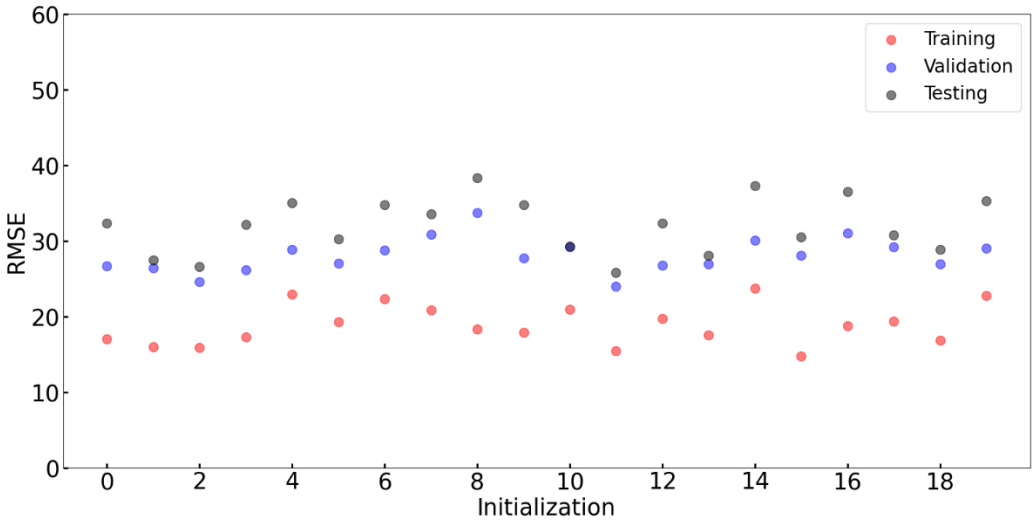
Supplementary Figure S7 reinforces the appropriate capacity of the networks designed in sections 2.4 and 3.2, also supporting the valid performance of GA.

Supplementary Figure S1. Data Extraction From GHGenius



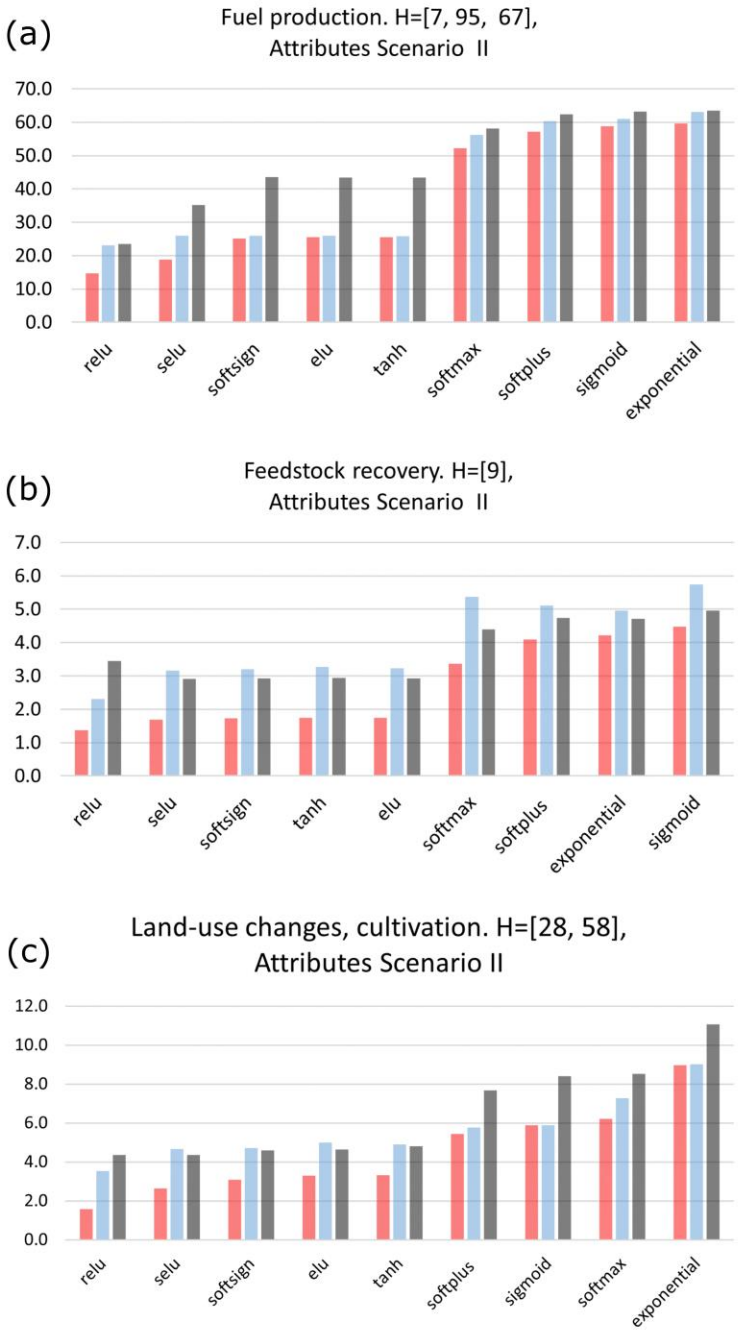
Supplementary Figure S1. The automation COM-based workflow for data extraction from GHGenius through Python. Python initializes GHGenius and runs the Macro Codes behind GHGenius for the specific cases. Once the GHGenius computation is finished, Python collects the resulting data from GHGenius. This bidirectional communication is performed through "win32com", which is a python library.

Supplementary Figure S2. Initialization impact



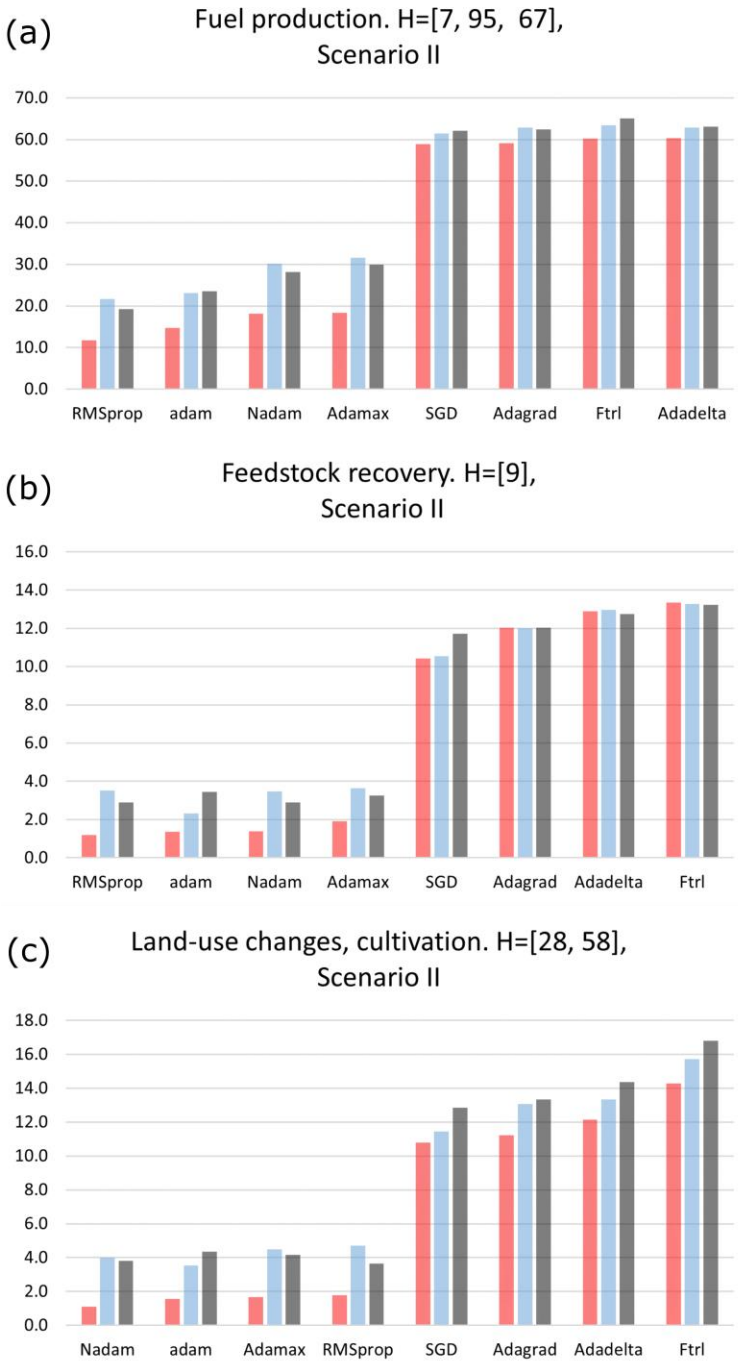
Supplementary Figure S2. Representative performances of the networks whose trainings were initialized by different random values. Early stopping regularization was applied for each network.

Supplementary Figure S3. Optimality of ReLU (Rectified Linear Unit) as an activation function



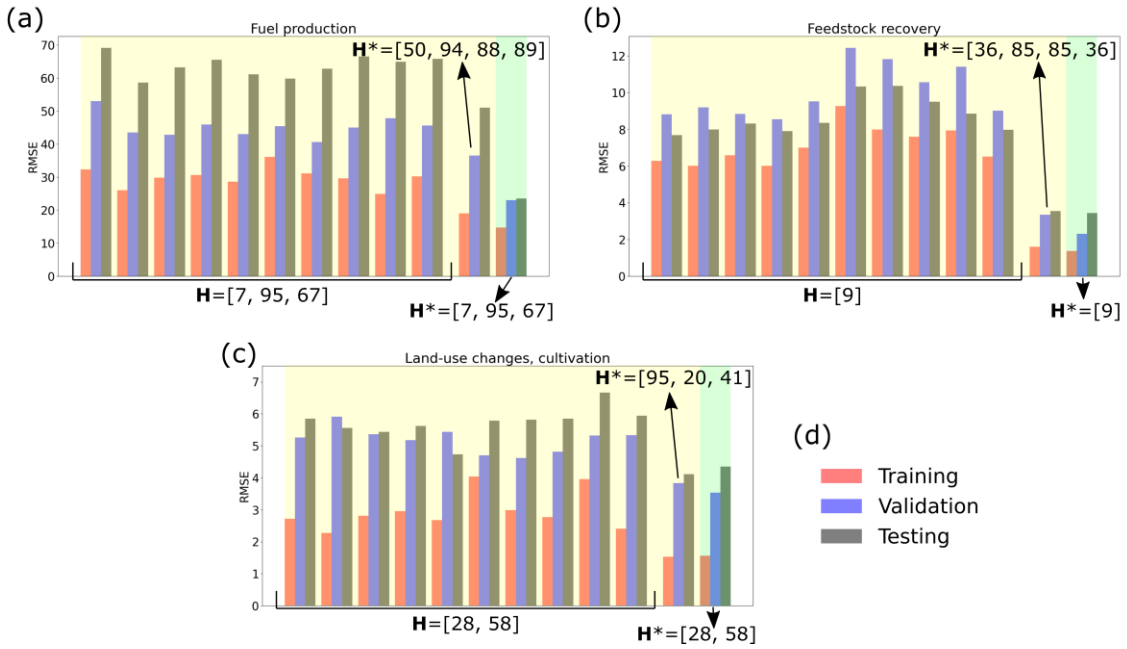
Supplementary Figure S3. The network performances for different activation functions. Red, blue, and grey show the RMSE for training, validation, and test sets, respectively. The title of each panel indicates the name of the unit process and the hidden topology of the FNN. Note that each network was trained with 50 random initial model parameters and the best performance of each activation function is presented here; this multiple initializations approach noticeably increases the chance of finding the global minimum of the loss function.

Supplementary Figure S4. Optimality of adam (adaptive moments) as an optimizer



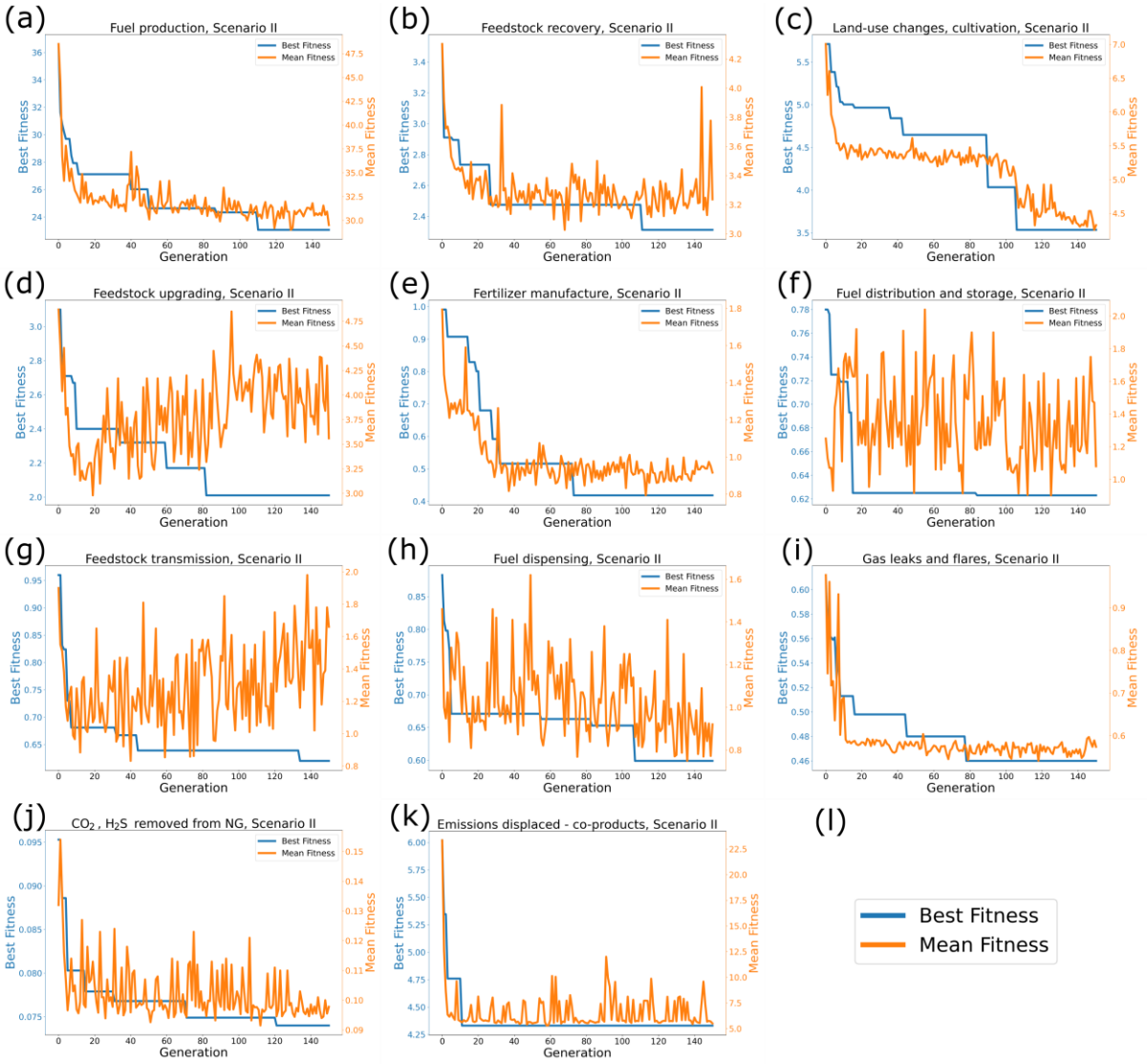
Supplementary Figure S4. The network performances for different optimizers. Red, blue, and grey show the RMSE for training, validation, and test sets, respectively. The title of each panel indicates the name of the unit process and the hidden topology of the FNN. Note that each network was trained with 50 random initial model parameters and the best performance of each optimizer is presented here; this multiple initializations approach noticeably increases the chance of finding the global minimum of the loss function.

Supplementary Figure S5. Attributes impacts on identical networks



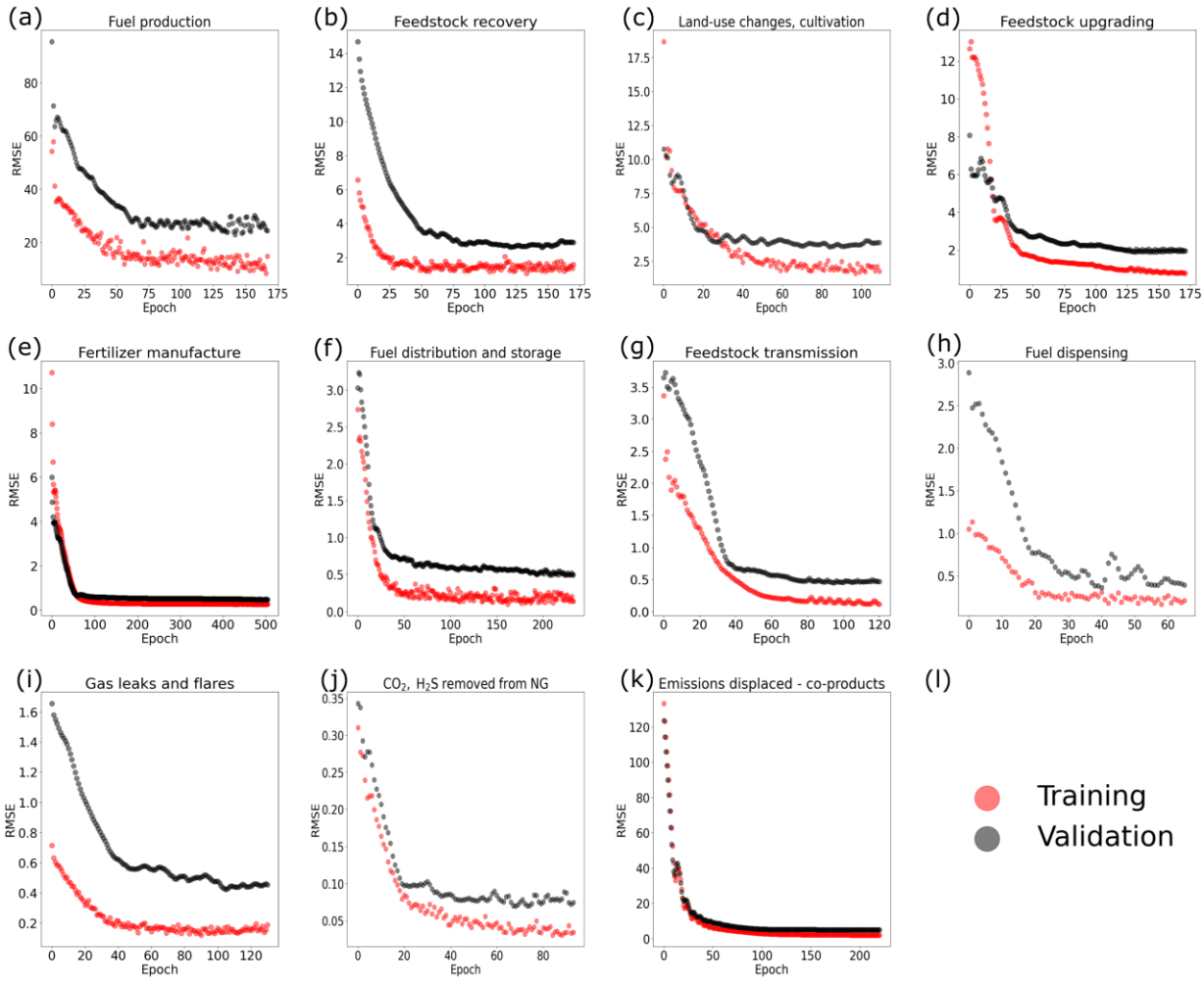
Supplementary Figure S5. Attributes impacts under the condition that both hyperparameters and hidden topology are identical for each contributor. The error bars highlighted in yellow and green correspond to Scenario I and II, respectively. The hidden topology is indicated by the arrow. Note the superscript * shows the optimal hidden topology, and each panel's title shows the name of the unit process.

Supplementary Figure S6. Fitness Evolutions



Supplementary Figure S6. (a-k) Best fitness and mean fitness through evolution for optimal networks. The title of each panel indicates the name of the unit process. (l) shows the legend for all panels.

Supplementary Figure S7. Learning curves



Supplementary Figure S7. Learning curves, variation of the training and validation errors versus epoch. The maximum number of the epoch is 750.

Supplementary Table S1. List of Canada's provinces and unit processes

Supplementary Table S1. List of provinces and unit processes considered in this study. For detailed information, readers are referred to (GHGenius).

Provinces	Abbreviation	Unit Processes	
Alberta	AB	Fuel dispensing	Fertilizer manufacture
Atlantic Canada	AT	Fuel distribution and storage	Gas leaks and flares
British Columbia	BC	Fuel production	CO_2, H_2S removed from NG
Manitoba	MB	Feedstock transmission	Emissions displaced - co-products
Ontario	ON	Feedstock recovery	
Quebec	QC	Feedstock upgrading	
Saskatchewan	SK	Land-use changes, cultivation	

Supplementary Table S2. List of fuel pathways

Supplementary Table S2. The feedstock-to-fuel list considered in this study. For detailed information, readers are referred to (GHGenius).

Feedstock	Fuel	Feedstock	Fuel	Feedstock	Fuel	Feedstock	Fuel
coal	Coal	Jatropha Oil	Biodiesel	NG	FT diesel	(W0/G100) Ethanol	CH2
Crude Oil	Gasoline	Camelina	Camelina Oil	Coal	FT diesel	Wheat Ethanol	CH2
Crude Oil	Gasoline (LowS)	Camelina Oil	Biodiesel	Wood Residue	BTL	Gasoline	CH2
Crude Oil	CG0/RFG100	Algae	Algae Oil	RDF	FT diesel	FT Diesel	CH2
Crude Oil	Hwy diesel	Algae Oil	Biodiesel	LFG	FT diesel	LPG	CH2
Crude Oil	Offroad Diesel	Animals	Tallow	electricity	FT Diesel	NG100/Water0	CH2
Crude Oil	Marine/Rail Diesel	Tallow	Biodiesel	NG	Gasoline	Used Oil	Diesel
Crude Oil	Fuel oil	Waste Grease	Yellow Grease	Wood Residue	Gasoline	Uranium	Nuclear
Crude Oil	Jet Fuel	Yellow Grease	Biodiesel	NG	DME	Wheat Straw	Grass
Crude Oil	Refinery Fuel Gas	Palm Effluent	Palm Sludge Oil	Wood Residue	DME	Wood Residue	Wood
Crude Oil	Coke	Palm Sludge Oil	Biodiesel	LFG	DME	Whole Corn	Corn Stover
Crude Oil	LPG	Spent Bleaching Earth	SBE Oil	Wood Residue	RNG	Wood Residue	Wood Pellets
NG	LPG	SBE Oil	Biodiesel	Wood Pellets	RNG	MSW	RDF
NGL94/RF6	LPG	Corn	Corn Oil	Coal	SCNG	power	NG
NG	LNG	Corn Oil	Biodiesel	process	LFG	industry	NG
NG	CNG	Fish	Fish Oil	NG	MA100	commerce	NG
Corn	Ethanol	Fish Oil	Biodiesel	Wood Residue	MA100	Fuel Production	NG
(W0/G100)	Ethanol	Canola Oil	HRD	RDF	MA100	NG pipeline	NG
Wet Stover	Ethanol	Corn Oil	HRJ	Wood Residue	Bio Oil	NG field	NG
Wheat	Ethanol	Soybean Oil	HRG	Wood Pellets	Bio Oil	power	H2
Barley	Ethanol	Canola Oil	HRP	Wheat Straw	Bio Oil Ag Res	pipeline	H2
Peas	Ethanol	Corn Ethanol	ETJ	Wood Residue	Refined Bio Oil	power	Syn gas
Sugarcane	Ethanol	NG	Methanol	Wood Pellets	Refined Bio Oil	gas pipeline	Syn gas
Sugar Beet	Ethanol	coal	Methanol	Wheat Straw	Refined Bio Oil Ag Res	EV	Electricity
Sorghum	Ethanol	Wood Residue	Methanol	water	CH2	EV	Nat. Gas
Corn	Butanol	LFG	Methanol	off grid elec	CH2	power	Electricity
Canola	Canola Oil	electricity	Methanol	NG	CH2	power	User Grid

Feedstock	Fuel	Feedstock	Fuel	Feedstock	Fuel	Feedstock	Fuel
Canola Oil	Biodiesel	NG100/C0	Methanol	Thermo nuclear	CH2	to power	User Generator
Soybeans	Soybean Oil	LFG	CRNG	Wood Residue	CH2	Wood Residue	RNG (Engine)
Soy Oil	Biodiesel	AD	CRNG	Coal	CH2	Wood Pellets	RNG (Engine)
Palm	Palm Oil	Organic Waste	CRNG	Methanol	CH2	Wood Residue	RNG (Turbine)
Palm Oil	Biodiesel	Electricity	CRNG	Methanol LFG	CH2	Wood Pellets	RNG (Turbine)
Jatropha	Jatropha Oil	H20/NG80	Hythane	Corn Ethanol	CH2		

Supplementary Table S3-S5. Optimality of ReLU as an activation function

Supplementary Table S3. Optimality of ReLU for "Fuel production" unit process

Fuel production. H=[7, 95, 67], Scenario II

	Network Performance		
relu	14.7	23.1	23.5
selu	18.9	26.0	35.2
softsign	25.1	25.9	43.5
elu	25.5	25.9	43.4
tanh	25.5	25.9	43.4
softmax	52.2	56.2	58.1
softplus	57.2	60.4	62.4
sigmoid	58.8	61.1	63.3
exponential	59.6	63.1	63.5

Supplementary Table S4. Optimality of ReLU for "Feedstock recovery" unit process

Feedstock recovery. H=[9], Scenario II

	Network Performance		
relu	1.4	2.3	3.4
selu	1.7	3.2	2.9
softsign	1.7	3.2	2.9
tanh	1.7	3.3	2.9
elu	1.7	3.2	2.9
softmax	3.4	5.4	4.4
softplus	4.1	5.1	4.7
exponential	4.2	5.0	4.7
sigmoid	4.5	5.7	5.0

Supplementary Table S5. Optimality of ReLU for “Land-use changes, cultivation” unit process

Land-use changes, cultivation. H=[28, 58], Scenario II

	Network Performance		
relu	1.6	3.5	4.3
selu	2.6	4.7	4.3
softsign	3.1	4.7	4.6
elu	3.3	5.0	4.6
tanh	3.3	4.9	4.8
softplus	5.4	5.8	7.7
sigmoid	5.9	5.9	8.4
softmax	6.2	7.3	8.5
exponential	9.0	9.0	11.1

Supplementary Table S6-S8. Optimality of adam (adaptive moments) as an optimizer

Supplementary Table S6. Optimality of adam for "Fuel production" unit process

Fuel production. H=[7, 95, 67], Scenario II

RMSprop	11.7	21.7	19.3
adam	14.7	23.1	23.5
Nadam	18.1	30.2	28.1
Adamax	18.4	31.6	30.0
SGD	58.9	61.5	62.1
Adagrad	59.2	62.9	62.4
Ftrl	60.3	63.5	65.1
Adadelta	60.4	62.9	63.1

Supplementary Table S7. Optimality of adam for "Feedstock recovery" unit process

Feedstock recovery. H=[9], Scenario II

RMSprop	1.2	3.5	2.9
adam	1.4	2.3	3.4
Nadam	1.4	3.5	2.9
Adamax	1.9	3.6	3.3
SGD	10.4	10.5	11.7
Adagrad	12.0	12.0	12.0
Adadelta	12.9	13.0	12.8
Ftrl	13.3	13.3	13.2

Supplementary Table S8. Optimality of adam for “Land-use changes, cultivation” unit process

Land-use changes, cultivation. H=[28, 58], Scenario II

Nadam	1.1	4.0	3.8
adam	1.6	3.5	4.3
Adamax	1.7	4.5	4.2
RMSprop	1.8	4.7	3.6
SGD	10.8	11.4	12.8
Adagrad	11.2	13.1	13.3
Adadelta	12.2	13.3	14.4
Ftrl	14.3	15.7	16.8

Supplementary Table S9. Optimality of MSE (Mean Square Error) as a loss function

Supplementary Table S9. The R^2 score for the network performances trained with MSE, MAE, and Huber. Note that $\delta=1$ for the Huber loss function.

		Unit Process		
		Fuel production	Feedstock recovery	Land-use changes, cultivation
MSE	Training	0.94	0.99	0.98
	Validation	0.87	0.97	0.93
	Testing	0.86	0.92	0.90
MAE	Training	0.98	0.99	0.97
	Validation	0.87	0.93	0.89
	Testing	0.81	0.92	0.90
Huber	Training	0.96	0.99	0.98
	Validation	0.81	0.92	0.86
	Testing	0.83	0.91	0.89

References

Chollet, F., 2015. Keras, https://github.com/keras-team/keras, https://keras.io, , last access : September 24, 2021

GHGenius, GHGenius, https://www.ghgenius.ca/, last access : September 24, 2021

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, Cambridge, Massachusetts, United States.

Song, R., Keller, A.A., Suh, S., 2017. Rapid life-cycle impact screening using artificial neural networks. Environ. Sci. Technol., 51(18), 10777-10785.