Json template objects	Keyword	Possible values	Default	Constraints	Description
entities*	entities	Array of ”entity” objects	-	-	Definition of the entities of the model.
entity	name	Dataset variable name	-	-	Define the name of an entity type.
	hidden_state_dimension	Natural number	-	-	Dimension of their hidden_states.
	features	Array of ”feature” objects	-	-	Features to be initialised in the hidden_state.
feature	name	Dataset variable name	-	-	Name of a feature.
	normalization	main.py function name	None	-	Name of the normalization function defined in the main.py file.
message_passing*	num_iterations	Natural number	-	-	Number iterations to repeat the message_passing phase.
	stages	Array of ”stage” objects	-	The messages are defined in chronological order	Stage of the message passing phase
stage	stage_name	String	-	-	Name to identify this stage.
	stage_mp	Array of ”single_mp” objects	-	-	Pairwise mp forming this stage.
single_mp	destination_entity	Entity name	-	-	Entity of the destination nodes.
	source_entities	”Source_entity” object	-	-	Defines how a source entity sends messages to the destination_entity
	aggregation	”aggregation” object	-	-	Defines how all the source entities aggregate together
	update	Operation	-	No input must be defined in the Operation, as it uses the aggregated input and the current hidden_state	Defines how to update the hidden_state given the aggregated information.
source_entity	name	Dataset name	-	-	Name of the source entity
	adj_vector	Dataset name	-	-	Adjacency list from source to destination nodes
	message	Array of ”operation” objects	-	We don’t allow this operation to use RNN.	Defines how to form the message given the current hidden_state.
Operation	type	”recurrent_neural_network”, ”neural_network”, ”direct_assignation”, ”activation”	-	If ”recurrent_neural_network” or ”neural_network” define ”input” and ”nn_name”.	Defines if a nn is needed to form the message.
	input	Array of strings	-	”hs_source” to refer to the source hidden-state. ”hs_dest” to refer to the destination hidden-state. ”edge_param” to refer to the edge information. Otherwise, match with an ”output_name”.	Number of parameters that all edges from this source to destination must have.
	nn_name	String	-	-	Defines the architecture of the nn.
	output_name	String	-	-	The output of the operation can be later referred to using this name.
aggregation	type	”sum” / ”ordered” / ”attention” / ”GCN” / ”interleave / ”concat”	-	If ”interleave”, then ”interleave_definition” object must be defined. If ”concat”, then ”concat_axis” can be defined.	Defines the type of aggregation
	concat_axis	1 / 2	1	-	If concat_axis = 1, then all the input messages are concatenated to a longer input message list. Otherwise, the messages are concatenated pair-wise to obtain longer messages (e.g message1 is the concat of all the first messages).
	interleave_definition	Dataset name	-	-	Defines the interleave sequence to combine together several source entity messages.
readout*	readout	Array of ”readout_operation” objects	-	-	Definition of the readout model.

Json template objects	Keyword	Possible values	Default	Constraints	Description
readout_operation	type	”predict”/ ”pooling” / ”neural_network” / ”product” / ”pooling” / ”extend_adjacencies”	-	In all operations, the ”input” field must be defined. For ”product”, also define ”type_product” and ”output_name”. For ”pooling”, define ”type_pooling” and ”output_name”. For ”neural_network”, define ”nn_name” and ”output_name”. For ”extend_adjacencies”, define ”adj_list”, ”output_name_src” and ”output_name_dst”. For ”predict”, at least ”nn_name” and ”label” must be defined.	Defines the readout operation to be applied
	input	Entity or previous output$_name$	-	-	Entity of the input hidden_states or previous name of the previous output variables to be used as input.
	label	Dataset name	-	-	Labels aimed to predict.
	label_normalization	main.py function name	None	-	Normalization function to be applied to the labels.
	label_denormalization	main.py function name	None	-	Denormalization function to recover the original labels.
	nn_name	String	-	It must match with the name of a neural network	Reference to the architecture of the neural network to use as readout.
	type_pooling	”sum”/”max”/”mean”	-	-	Aggregates all the indicated ”input” tensors into a single one. To do so, it applies the indicated operation position-wise.
	type_product	”element_wise”/”dot_product”	-	-	Applies the product operation to the two inputs defined.
	adj_list	Dataset name	-	-	Defines the adj_list to use for extending.
	output_name_src	String	-	-	Defines the name where the source extended values are stored.
	output_name_dst	String	-	-	Defines the name where the dst extended values are stored.
	output_name	String	-	-	Defines the name where the output is stored.
neural_networks*	neural_networks	Array of ”neural_network” objects	-	-	Definition of the necessary feed-forward and recurrent models of the GNN model.
neural_network	nn_name	String	-	-	Identifier of the neural network
	nn_type	”feed_forward”/”recurrent”	-	If ”feed_forward”, define the ”nn_architecture”. Otherwise, define at least the ”recurrent_type” as well as any additional Keras parameter.	Indicates the types of NN aimed to define.
	nn_architecture	Array of ”layer” objects	-	-	Architecture of the feed-forward neural network
	recurrent_type	Recurrent cell name (Keras documentation).		-	Recurrent Neural Network model definition
	(Keras parameter name)	Parameter value (Keras docu)	-	-	We can add any parameter accepted by Keras library model of the type defined
layer	type	Layer type (Keras docu)	-	-	Type of layer
	(Keras parameter name)	Parameter value (Keras docu)	-	-	We can add any parameter accepted by Keras library model of the type defined
learning options*	loss	Function name (Keras docu)	-	-	Define the loss function to be used
	optimizer	”Optimizer” object	-	-	Define the optimizer options
optimizer	type	Name of the optimizer (Keras docu)	-	-	Define the name of the optimizer to be used
	(Keras parameter name)	Parameter values (Keras docu)	-	-	We can add any parameter accepted by Keras library to costume the optimizer strategy.
	schedule	”schedule object”	None	-	-
schedule	type	Name of the schedule (Keras docu)	-	-	Define the name of the schedule strategy
	(Keras parameter name)	Parameter values (Keras docu)	-	-	We can add any parameter accepted by Keras library to costume the schedule strategy