Json template objects	**Keyword**	**Possible values**	**Default**	**Constraints**	**Description**
entities	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 featue.
	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.
	architecture	Array of ”mp_step” objects	-	The messages are defined in chronological order	Steps of the message passing phase
mp_step	step_name	String	-	-	Name to identify this step.
	single_mp	Array of ”message” objects	-	-	Pairwise mp forming this step.
single_mp	type	”single_source” / ”multi_source”	-	If ”multi_source”, then ”combined_mp_options” object must be defined.	If ”single_source”, then a destination doesn’t have sending sources from multiple entities.
	source_entity	Entity name	-	-	Entity of the sending nodes.
	destination_entity	Entity name	-	-	Entity of the destination nodes.
	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.
	aggregation	”sum”/”ordered”/”GAT”/ ”GCN”/”combined”	-	-	Defines how to aggregate all the messages received.
	update	Operation	-	-	Defines how to update the hidden_state given the aggregated information.
Operation	type	”apply_nn”, ”apply_rnn”, ”None”, ”activation”	-	If ”apply_nn”, define ”input” and ”nn_name”. If ”apply_rnn”, define ”recurrent_type”	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.
	recurrent_type	Recurrent cell type (Keras docu)	-	-	Defines the type of RNN to be used
	output_name	String	None	-	The output of the operation can be later referred to using this name.
combined_mp_options	combined_message_passing_options	Array of ”combined_mp” objects	-	-	Each of these defines how a destination entity treats multiple-source messages at one step.

Json template objects	Keyword	Possible values	Default	Constraints	Description
combined_mp	step	Step name of all its message_passings	-	-	Time step when the coalition exists.
	destination_entity	Entity name	-	-	Destination entity of the multiple-source messages.
	type_of_combination	”interleave” / ”concat”	-	If ”interleave”, then ”interleave_definition” object must be defined.	Interleave defines a costume sequence from the input messages. Concat simply concatenates them together.
	interleave_definition	Dataset name	-	-	Definition of the costume interleave pattern.
	update_type	Recurrent cell type (Keras docu)	-	-	Defines the type of RNN that should be used.
output	output	Array of readout objects	-	-	Definition of the model’s readouts.
readout	type_of_prediction	”node_prediction”/ ”graph_prediction”	-	-	Whether it makes node or graph predictions.
	entity	Entity name	-	-	Entity of the input nodes.
	output_label	Dataset name	-	-	Labels aimed to predict.
	output_normalization	main.py function name	None	-	Normalization function to be applied to the labels.
	output_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	Architecture of the readout neural network.
feed_forward_models	feed_forward_models	Array of ”neural_network” objects	-	-	Definition of the necessary feed-forward models of this GNN model.
neural_network	model_name	String	-	-	Identifier of the neural network
	model_architecture	Array of ”layer” objects		-	Architecture of the neural network
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
	schedule	”Schedule”object	-	-	Define the schedule options if any
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	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