Table 6: Temporal Operators in Linear Temporal Logic

Symbol	G	F	R	W	M	X	U
Meaning	globally	finally	release	weak until	strong release	next	until

A Temporal Operator Notations

The temporal operators supported by ϕ is presented in Table 6.

B Experimental Settings

B.1 Environment

Experiments were performed on a cluster with four Intel 24-Core Gold 6248R CPUs, 1TB DRAM, and eight NVIDIA QUADRO RTX 6000 (24GB) GPUs.

B.2 Dataset Description

The specifications (ϕ) for `Short` and `Diverse` is constructed by Spot. The corresponding systems (B) for both Spot generated specifications and RERS are generated by LTL3BA.

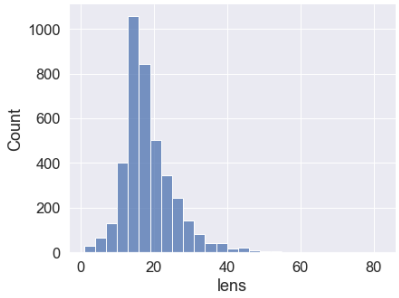
The purpose of the three datasets are to test the generalization capability of OCTAL across different distributions and varying length specifications. All three datasets comprise of specifications and corresponding systems of varying length. The distribution of `Diverse` subsumes `Short` and `RERS`, and comprises of the most complex and lengthy specifications and corresponding systems, for which LTL3BA takes the longest time. The generalization on `Diverse` tests the performance of OCTAL on out of distribution data, and the generalization accross the other scenarios proves OCTAL's capability to perform independently when varying length systems and specifications are present in the same dataset, which is a limitation addressed by the previous related work(s).

B.2.1 Spot Generated Datasets

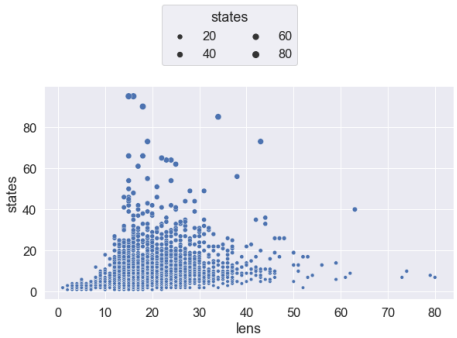
The `randltl` feature of Spot is used to generate LTL formulae of varying lengths, that can be specified as a parameter. The default size of the expression tree is 15. `randltl` generates a variety of LTL formulae, where no two formulae are syntactically the same and less than 10% of the formulae are semantically equivalent. There are two types of datasets generated through Spot: `Short`, where the length of the LTL formulae (noted as #Lens) range from 1 to 80 and `Diverse`, where the length of the LTL formulae range from 1 to 144. The distribution of the length of the formula with respect to frequency is plotted in Figures 3(a), 4(a). The number of states and edges of these two datasets are described in Figures 3(b), 3(c), 4(b), 4(c) respectively. Analyzing the distributions, it can be concluded that the diverse dataset comprises of the largest automata systems and an uniform distribution of the length of LTL formulae. The range of LTL formulae length and states are similar between `Short` and `RERS19`, but the corresponding transition range is less than 160 for `RERS19`, while 1,711 for `Short`. The distribution of `Diverse` set both in terms of ϕ and B is different from other two datasets.

B.2.2 RERS dataset

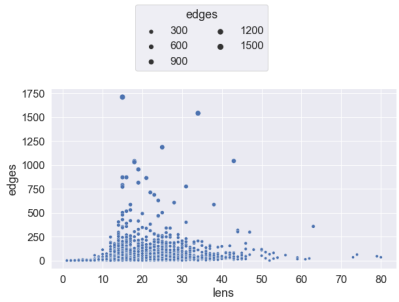
RERS19 Rigorous Examination of Reactive Systems (RERS) is a model checking competition track organized every year, that invites teams from all over the world to participate. We evaluate their dataset on the Sequential LTL track, which comprises of 1800 problems and compare the performance of OCTAL in terms of accuracy with time constraints, with respect to the top 3 teams. The details of the distribution is plotted in Figure 5.



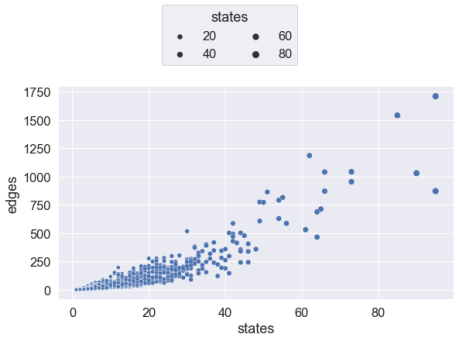
(a) Length distribution



(b) Length v.s. State distribution

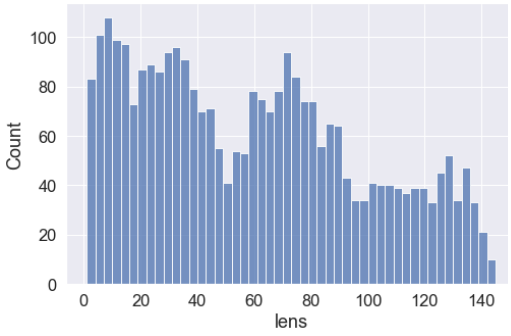


(c) Length v.s. Edge distribution

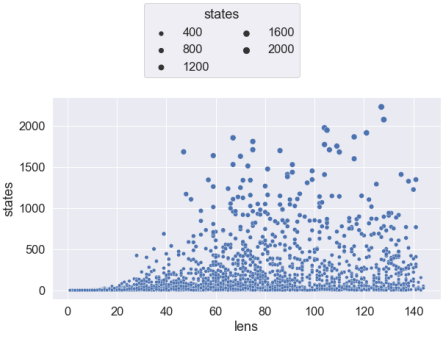


(d) Edge v.s. State distribution

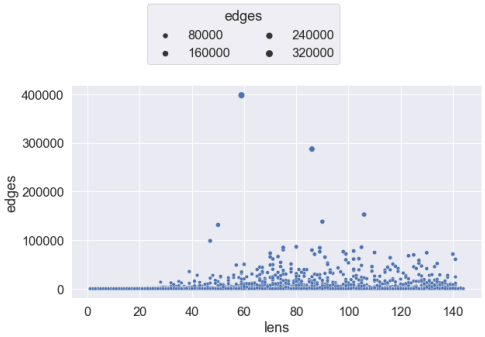
Figure 3: Distribution of Short Dataset



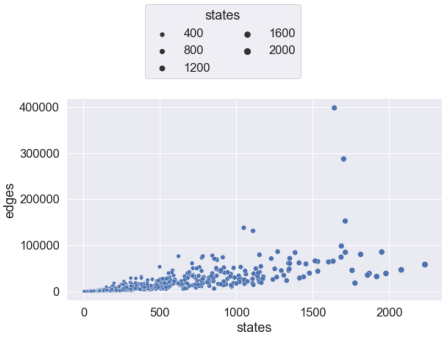
(a) Length distribution



(b) Length vs State distribution

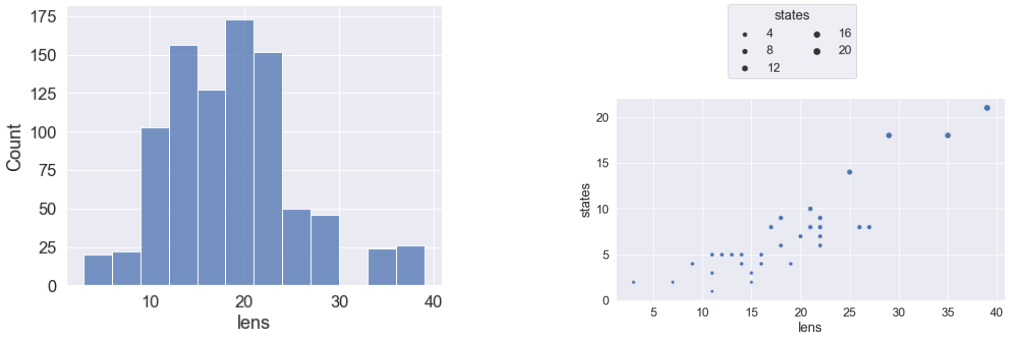


(c) Length vs Edge distribution



(d) Edge vs State distribution

Figure 4: Distribution of Diverse Dataset



(a) Length distribution (b) Length vs State distribution

(c) Length vs Edge distribution (d) Edge vs State distribution

Figure 5: Distribution of RERS Dataset

B.3 Architecture and Hyperparameters

The architecture of OCTAL (described in Figure 6) comprises of a three layer GNN, followed by a two layer MLP to classify whether B satisfies ϕ. Mean pooling is used to aggregate the node embeddings. A dropout rate of 0.1 is used, along with 1d batch normalization in every convolution layer of GNN. RELU [1] is used as the activation layer in GNN and MLP. Every node in $\mathcal{C}$ has an embedding of length 64 (66 in case of Directed G as described below), The GNN component produce an encoding for $\mathcal{C}$ of length 128. The MLP takes this encoding as the input, and produces a 0/1 result as the final output.

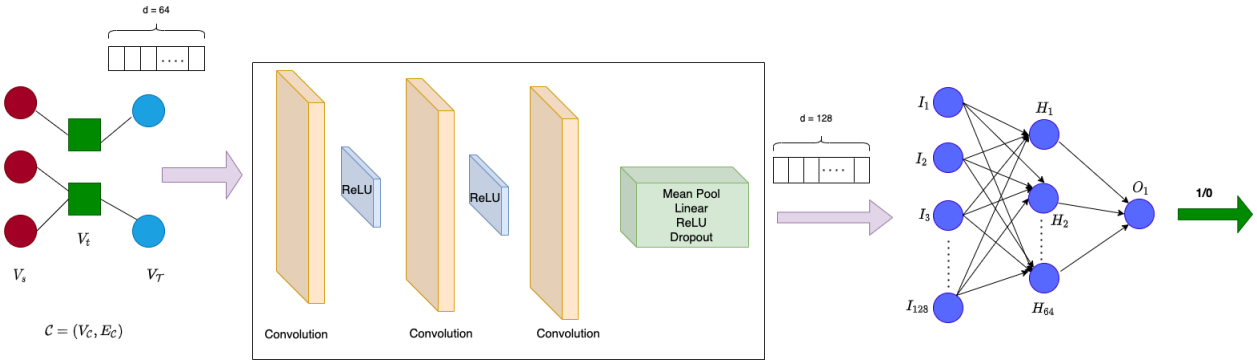
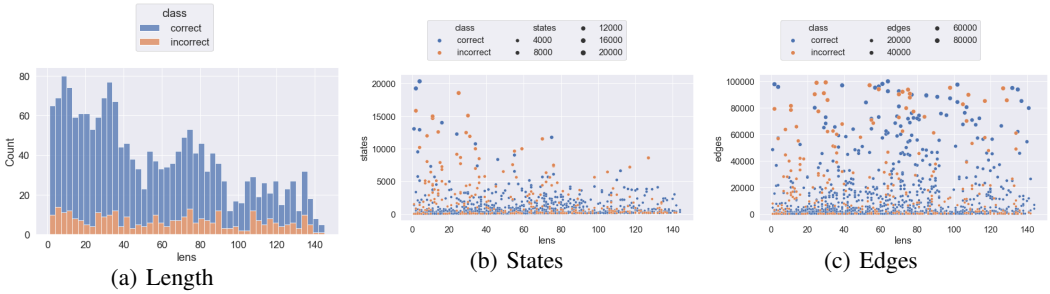


Figure 6: Architecure of OCTAL

B.4 Implementation Details

The code base is implemented on PyTorch 1.8.0 and pytorch-geometric [19].



(a) Length (b) States (c) Edges

Figure 7: Distribution of Predicted Results on `Diverse` dataset.

Table 7: Classification accuracy and precision/recall for 1-hot Encoding. `RERS19(S)` and `RERS19(D)` represent models being trained on `Short` and `Diverse`, and then tested on `RERS19`.

	Models	Short	Diverse	RERS19(S)	RERS19(D)
	OCTAL	95.80±0.00	88.33±0.01	95.20±0.01	94.67±0.01
OCTAL	**Precision**	95.40±0.01	88.33±0.01	97.40±0.01	92.33±0.01
	Recall	96.00±0.00	86.33±0.01	92.60±0.02	98.00±0.01

C Additional Results

C.1 Generalization on `Diverse`

We further tested the generalization of OCTAL for a larger and diverse distribution by training OCTAL on `Short` and testing with `Diverse`. OCTAL (GIN) achieves 84.2% accuracy on average which is slightly decayed from the performance in Table 3 but still generalizes reasonably well across larger B's and ϕ's. Figure 7 shows the distribution of correctly and incorrectly predicted samples. OCTAL generalizes pretty well across varying length ϕ and for B's of sizes much larger than the range of `Short`. The distribution pattern for the correct prediction is similar to `Diverse`'s (Figure 4 in Appendix) so that OCTAL is capable of good generalization for larger unseen samples in testing.

C.2 1-hot Encoding

Each node $v \in \mathcal{C}$ consisting of operands/variables is represented by a vector as follows

$$\{\underbrace{[\_]}_{1/0}\underbrace{[\_,\_,\_,\cdots,\_]}_{\forall a\in\mathcal{A}}\underbrace{[\_,\_,\_,\cdots,\_]}_{!a/\neg a,\forall a\in\mathcal{A}}\underbrace{[\_,\_,\_,\cdots,\_]}_{\forall o\in\{\mathcal{O}/!\}}\underbrace{[\_,\_]}_{q\in Q}\}$$

Part I of 1 bit is reserved for the special variable **true**(1) or **false**(N). Part II encodes $\forall a \in \mathcal{A}$, with size of $|\mathcal{A}|$. Part III of size $|\mathcal{A}|$ encodes variables/operands in either $\neg\mathcal{A}$ or $\tilde{\mathcal{A}}$, as both of them are semantically equivalent. Part IV corresponds to operators in $\mathcal{O}$ except !, with the size of $|\mathcal{O}| - 1$. The last part V represents the type of state q of B in 2 bits. Part I and V use one-hot encoding for indication. Here, instead of sampling from a distribution, each variable/operator is represented with the value 1 in its respective position, if v contains the representation for it. Otherwise, the position has the value 0. The true form a and negated form $!a/\neg a$ of a variable share the same value but are located in different sections ($\mathcal{A}$ in Part II, $\neg\mathcal{A}/\tilde{\mathcal{A}}$ in Part III).

C.3 Directed G

Directed G takes the unified framework $\mathcal{C} = (V_\mathcal{C}, E_\mathcal{C})$, the union of $\mathcal{G}$ and $\mathcal{T}$. Each node $v \in \mathcal{C}$ consisting of operands/variables is represented by a vector as follows

$$\{\underbrace{[\_]}_{1/0}\underbrace{[\_,\_,\_,\cdots,\_]}_{\forall a\in\mathcal{A}}\underbrace{[\_,\_,\_,\cdots,\_]}_{!a/\neg a,\forall a\in\mathcal{A}}\underbrace{[\_,\_,\_,\cdots,\_]}_{\forall o\in\{\mathcal{O}/!\}}\underbrace{[\_,\_]}_{q\in Q}\underbrace{[\_,\_]}_{v\in V_s}\}$$

Table 8: Classification accuracy and precision/recall for Directed G. `RERS19(S)` and `RERS19(D)` represent models being trained on `Short` and `Diverse`, and then tested on `RERS19`.

	Models	Short	Diverse	RERS19(S)	RERS19(D)
	OCTAL	96.20±0.00	87.33±0.01	95.40±0.01	93.33±0.01
OCTAL	**Precision**	95.60±0.01	88.67±0.01	97.40±0.01	88.33±0.02
	Recall	97.00±0.02	84.67±0.02	93.00±0.03	99.67±0.01

Part I of 1 bit is reserved for the special variable **true**(1) or **false**(N). Part II encodes $\forall a \in \mathcal{A}$, with size of $|\mathcal{A}|$. Part III of size $|\mathcal{A}|$ encodes variables/operands in either $\neg\mathcal{A}$ or $\tilde{\mathcal{A}}$, as both of them are semantically equivalent. Part IV corresponds to operators in $\mathcal{O}$ except !, with the size of $|\mathcal{O}| - 1$. Part V represents the type of state q of B in 2 bits. Parts I and V use one-hot encoding for indication. Each variable/operand has a distinct meaning but sharing across $\mathcal{G}$ and $\mathcal{T}$. Without loss of generality, the value of $i \in \mathcal{A} \cup \mathcal{O}$ is sampled from a normal distribution $\mathcal{N}(\mu_i, \sigma)$ with the distinct mean μ_i and controlled variance to avoid overlaps. The true form a and negated form $!a/\neg a$ of a variable share the same value but are located in different sections ($\mathcal{A}$ in Part II, $\neg\mathcal{A}/\tilde{\mathcal{A}}$ in Part III). Lastly, Part VI represents the direction. In this case, the direction needs to be encoded only in $\mathcal{G}$, which is the bipartite representation of B. Every $v \in V_e$, stores the source and destination vertices of the corresponding transition. The first field represents source $s \in V_s$, and the second destination $d \in V_s$. For Part VI, V_s and $V_{\mathcal{T}}$ would have zeros as the direction is sufficiently encoded in V_e.

C.4 Additional Result Analysis

The experimental results on 1-hot encoding and Directed G are presented in Tables 7 and 8 respectively. The experiments were run under similar settings for five independent iterations, with mean and standard deviation reported for accuracy, precision and recall.

From the tables, it can be observed that both 1-hot and directed G perform equally well on all configurations. On comparison with the undirected G presented in Table 3, it can be observed that undirected G performs better on average with `Diverse` but has worse standard deviation. The rest of the configurations report comparable performance for accuracy, precision and recall with respect to Tables 7 and 8. From the results analyzed, we can conclude that the position specific representation for variables, their negated form, and operators is sufficient for OCTAL to report consistent performance, with similar generalization capabilities.