

Searching for Accuracy vs. Efficiency Trade-off Solution in Deep Differentiable Logic Gate Networks

Chan Duong Nguy
0009-0006-1339-9841

Vistula University
3 Stokłosy, 02-787, Warsaw, Poland
Email: chanduong1999@gmail.com

Piotr Wasilewski
0000-0003-0027-1102

Systems Research Institute,
Polish Academy of Sciences
6 Newelska, 01-447 Warsaw, Poland
Faculty of Computer Science,
Dalhousie University
6050 University Avenue, Halifax,
Nova Scotia, Canada
Email: pwasilew@ibspan.waw.pl

Abstract—Recent research on Deep Differentiable Logic Gate Networks (DLNs) aim to shift machine learning from traditional weight based models to weightless networks that learn the best logical operators using gradient-based optimization. While DLNs can provide much faster inference and require less memory together with an interpretability of the results, they still face issues such as vanishing gradients in deeper networks and high computational cost during training. We compare several DLN configurations to find the best trade-off between accuracy and efficiency evaluating baseline DLNs against versions that use Residual Initialization (RI) at identity gates (denoted as $DLN_{\alpha^{**}}$ where $\alpha \in \{Z, M, L\}$, representing Zadeh, Menger, and Łukasiewicz T-norm respectively) and a mirror gate architecture (denoted as $DLN_{\alpha^{**}}^m$). We focus on the T-norm choices and on two architecture modifications. Results on benchmark datasets show that $DLN_{\alpha^{**}}^m$ models greatly reduce vanishing gradients and improve accuracy in deeper networks.

I. INTRODUCTION

ONE of the key challenges in the contemporary machine learning (ML) is the interpretability of the results due to the fact that this field is dominated now by artificial neural networks (ANNs) combining gradient-based learning methods, including the backpropagation algorithm [17], with various architectures and used for simulating other ML algorithms. ANNs are very good learners with a very poor explainability of achieved results. Another challenge here is the scalability of induced models which is also a problem to other learning algorithms including symbolical algorithms as well: when decisions trees with 50 nodes still can be easily interpreted, decision trees with 30 000 nodes can be hardly recognized as interpretable. Both challenges (results interpretability and scalability) can be faced by Differentiable Logic Gate Networks (DLNs). One of the possible solutions to these problems is searching for the best trade-off between accuracy and efficiency by analyzing several DLN configurations.

In the perspective of graph theory, Artificial Neural Networks (ANNs) such as Multilayer Perceptrons (MLPs) are

often represented as dense matrices of edge weights, where each connection is a learnable parameter. On the other hand, DLN learn the identity of the vertices in a graph, selecting from 16 possible classical binary logic gates (e.g., AND, XOR) for each computational unit. Because these discrete gates are not differentiable, DLNs relax them using triangular norms (T-norms).

In this study we evaluate the choice of T-norm as a primary hyperparameter with the two architectural modifications. The Menger (probabilistic) T-norm [13] (arithmetical multiplication) provides smooth derivatives but suffers from severe signal attenuation as the network depth increases. The Zadeh T-norm ($\min\{a, b\}$) offers more stable gradients (limited to $\{-1, 0, 1\}$) and is effective at identifying singular patterns, though it may lack the robustness of Menger on certain datasets [19]. Similarly, the Łukasiewicz T-norm ($\max\{0, a + b - 1\}$) provides strong gradients but contains wide “flat” areas in its domain where the gradient is zero, frequently leading to a total failure of error signal propagation [16]. We denote DLN with the T-norm it is based on as DLN_{α} where $\alpha \in \{Z, M, L\}$, Zadeh, Menger, and Łukasiewicz respectively. The first architectural modifications is Residual Initialization (RI) [14], [16]. This strategy biases the learnable probability distributions toward identity/pass-through gates (Gate A or Gate B or both. Similar to residual connection this aid the backpropagation and improve accuracy. Following the previous notation convention we denote the network with RI as $DLN_{\alpha^{**}}$. The mirror gate mechanism [20] is the second modification, denoted as $DLN_{\alpha^{**}}^m$. This approach exploits the structural symmetry of the 16 classical gates, by training on only eight gates and using a learnable mirror coefficient (m) to handle potential inversion, this reduce the training memory and computation by approximately half.

This paper presents a systematic evaluation of DLN_{α} , $DLN_{\alpha^{**}}$, and $DLN_{\alpha^{**}}^m$ across the Menger, Zadeh, and

Łukasiewicz frameworks. Using benchmarks including MONK's problems, Adult Census, Breast Cancer, and MNIST, we analyze the trade-offs between accuracy, parameter count, memory footprint, and inference latency. Our results aim to define the most performant configurations for real world, low latency applications such as medical sensing.

II. DIFFERENTIABLE LOGIC GATE NETWORKS

The development of Differentiable Logic Gate Networks (DLNs) is a specialized field focused on transforming discrete logic circuits into learnable, differentiable structures. DLNs and ANNs can be interpreted as a learning of graph. Each neuron in an ANN is a vertex, and it is densely connected to previous layer neurons. The post training result is dense matrices of edge weights. Whereas in DLNs, the connections between layers are sparsely and randomly assigned and remain fixed thought training. Each vertex of the graph connected to two vertices from the previous layer computes the weighted sum of all 16 logic gates:

$$r = \sum_{i=0}^{15} p_i \cdot f_i(a, b) \quad (1)$$

For $a, b \in \mathbf{B}$, where $\mathbf{B} = \{0, 1\}$ are inputs and f_i is the relax version of the logic gate using one of the 3 T-norms, and p_i is acquired by using a *softmax* function on the trainable logit vector for every neuron. Through the equation 1 the model learning the optimal logic gate at each vertex through backpropagation.

A. Foundational DLN Architectures and Menger T-norms

The seminal work on DLNs was introduced by Petersen et al. [13], who proposed using the Menger (probabilistic) T-norm ($T_M(a, b) = a \cdot b$) [10] to relax logical conjunctions. This approach allowed for smooth derivatives, making standard backpropagation feasible for logic networks. However, the authors also identified that Menger based DLNs (DLN_M) are susceptible to vanishing gradients in deep layers because the product of probabilities consistently attenuates the error signal. To counter this, original implementations required gradient scaling factors to maintain flow in networks exceeding six layers.

B. Alternative T-norms (Zadeh and Łukasiewicz)

To address the stability issues of the Menger norm, we explored [19] the Zadeh T-norm [21]. Zadeh based DLNs (DLN_Z) utilize piecewise differentiable functions where gradients are restricted to the set $\{-1, 0, 1\}$, providing a more stable signal that neither vanishes nor explodes as rapidly as Menger gradients. Further extensions [16] introduced the Łukasiewicz T-norm ($\max\{0, a + b - 1\}$). While the Łukasiewicz operator provides strong gradients in certain regions, it suffers from wide flat areas where the gradient is zero [7], presenting a significant training challenge for deep architectures.

C. Architectural Innovations in DLNs

Recent research has moved beyond the choice of T-norms toward structural optimizations. The concept of Residual Initialization (RI) was introduced to improve the training of deep DLNs [14], [16], particularly those based on the Łukasiewicz norm. RI works by initializing the softmax logits to assign a high prior probability to identity/pass-through gates (Gate A or B or both). This ensures stable signal propagation during training.

Additionally, the development of Convolutional DLNs introduced the idea of reusing logic elements across input features in a sliding-window fashion, mimicking the spatial hierarchies of traditional CNNs while maintaining bit-wise efficiency [14]. Most recently, we proposed the mirror gate architecture proposed to eliminate redundancy in the 16 fuzzy operations [20]. By parameterizing the choice between a gate and its inverse using a learnable mirror coefficient (m) controlled by a sigmoid function, this configuration halves the per-neuron computational cost during the differentiable training phase.

III. METHODOLOGY

To ensure a rigorous evaluation of the various Deep Differentiable Logic Gate Network (DLN) configurations and consisting with previous work, we adopt four benchmarks datasets MONK, Adult Census, Beast Cancer, and MNIST [18], [1], [22], [9]. For tabular datasets such as MONK, Adult, Breast Cancer the continuous features are discretized into binary vectors, while categorical attributes are one-hot encoded to ensure compatibility with the binary logic gate operations required for post-training inference.

We compare three distinct architectural variants across the Menger, Zadeh, and Łukasiewicz T-norms. The standard configuration where each neuron learns a probability distribution over the full set of 16 fuzzy logic gates DLN_α for $\alpha \in \{M, Z, L\}$ for Menger, Zadeh, Łukasiewicz respectively. Configurations employing Residual Initialization (RI), where gates corresponding to identity mappings (a and b) are each assigned a high prior probability (45% each) at the start of training (denoted as DLN_{α}^{**} for $\alpha \in \{M, Z, L\}$). The usage of mirror mechanisms, which uses a learnable mirror coefficient (m) to decide on inversion, theoretically reducing training resources by approximately 50% is denoted as DLN_{α}^{m**} for $\alpha \in \{M, Z, L\}$.

To reduce impact of the random initialization of connections, each model configuration is trained using 5 different random seeds. Results are reported as the average performance across these seeds to ensure statistical significance. For the baseline DLN architectures, connections between layers are randomly assigned and remain fixed throughout training. The network is implemented with Pytorch [12] and optimized using Adam [6]. During inference speed evaluation OS noise can cause the results become unstable. OS noise refers to the performance interference caused by background system processes [11]. To address this issue, we perform repeated inference runs (averaging across 10000 iterations) to eliminate

outliers and minimize the impact of OS noise. By running inference multiple times, we ensure that the recorded speeds accurately reflect the model's architectural efficiency.

For baselines we include several symbolic models: Decision Trees (DT) [15], Random Forest (RF) [3] and XGBoost [4] using default configurations provide by the libraries. Whereas in ANNs case, we opt to use Multilayers Perceptron. The MLPs with 2 layers and 8 neurons are used in MONK dataset regardless of the subset of the dataset i.e., (MONK-1, MONK-2, MONK-3). Moreover, in the adult and breast cancer dataset networks with 5 layers, 256 and 128 neurons are used respectively. As to MNIST dataset, MLPs with 6 layers and 8000 neurons are used for the case of small configuration. Similarly, in the same dataset, the normal configuration has 6 layers but with 64000 neurons.

The evaluation is based on these metrics accuracy, inference time, memory footprint, and binary operations (OPs). Accuracy is evaluated after the discretization of the fuzzy logic network into a "hard" binary logic gate network. Parameters is counted as the total number of learnable units/gates in the network in case of DLNs whereas DT, RF and XGBoost are counted as the total number of nodes in the tree for ANNs is the number of learnable parameters. Inference Time is measured on a single-threaded CPU setup. Memory Footprint the theoretical memory required to run. For the MNIST dataset, we specifically quantify the number of binary operations (OPs) required for a single forward pass.

IV. EXPERIMENTS AND RESULTS

A. MONK

Table I summarized the results on the MONK-1, MONK-2, and MONK-3 datasets. Overall, these datasets were relatively small and rule-based, meaning that many models are able to achieve very high accuracy.

For MONK-1, the ANN and both $DLN_{M^{**}}$ and $DLN_{L^{**}}$ achieve 100.00% accuracy, whereas XGBoost also performs well with 99.31%. Several other DLN variants were also able to reach near perfect performance. This showed that DLNs can successfully learn the underlying logical structure of MONK-1. However, not all variants of DLN perform equally well, DLN_L decreased significantly to 83.28%, with p value less than 0.05 comparing to DLN_M , DLN_Z , and the baselines. This validates a similar observation to the previous study.

For MONK-2, performance decreases for most models, reflecting the higher difficulty of this dataset. Decision Tree and XGBoost achieved around 81%, while Random Forest performed considerably lower at 74.07% ($p \leq 0.001$). DLN models generally struggled on this dataset, with accuracies mainly between 60% and 74%. The best DLN result was obtained by $DLN_{M^{**}}$ at 80.41%, which was comparable to the tree-based baselines but has relatively high variance ($p > 0.005$).

For MONK-3, the strongest baselines were Random Forest with 96.16% followed by XGBoost at about 94.91% this was due to the fact these ensemble models use a collection of decision trees to reduce the noise from the training data, and

the dataset has added noise to test the generalization of the model. $DLN_{M^{**}}$ reached 94.30%, which is close to XGBoost and shows that DLNs can also perform with no substantial different on this dataset ($p > 0.05$).

In terms of efficiency, DLNs consistently have very low inference time (around 2–4 ns) and minimal memory usage (36–72 B). This is order of magnitudes lower than Random Forest, XGBoost, and ANN, which require significantly more memory. Overall, the MONK experiments show that DLNs can achieve strong accuracy on some rule-based problems while remaining very efficient, but performance can drop noticeably on more difficult cases such as MONK-2. Furthermore, these results indicate that DLNs are capable of learning complex logical decision structures, while requiring only minimal computational resources.

B. Adult dataset

The widely used Adult dataset [1], is derived from US Census data. It often contains attributes such as age, education, occupation, working hours. The goal for the dataset is to classify whether an individual earns more than \$50,000 a year. Its real-world making it suitable for evaluating fairness and robustness. In this experiment, the Adult dataset provides a practical scenario for assessing classification performance on structured tabular data.

Table II reported that the baseline models achieved relatively strong predictive performance on the Adult income classification task. The ANN obtained the highest accuracy at 84.90%, closely followed by XGBoost at 84.40% with no statistical different ($p > 0.05$). While Decision Tree and Random Forest achieved 82.12% and 83.33%, respectively ($p \leq 0.001$). In contrast, the standard DLN variants (DLN_M , DLN_Z , and DLN_L) achieved substantially lower accuracies, ranging from 73.55% to 75.68%, indicating a marked reduction in classification performance relative to the baseline models. All observed differences of those versions were statistically measurable with respect to the baselines ($p \leq 0.05$). The residual initialized DLN variants (denoted with **), improved performance, with $DLN_{L^{**}}$ reaching 81.41%, narrowing the gap with traditional machine learning methods. Comparing each $DLN_{\alpha^{**}}$ to their basic configurations DLN_{α} , the method yielded higher accuracy on average; however the differences were insignificant ($p > 0.05$). This is due to the unstable training process of DLN evidence by the high standard deviation. Nevertheless, even the best DLN configuration remained below ANN and XGBoost, suggesting that while the modifications improved backpropagation of error signal, DLNs still struggle to match the accuracy of established models on this dataset. The models with mirror gates (denoted with superscript m) showed reduction in accuracy when comparing with RI variants. Analyzed the models with the based DLNs configurations they exhibited mixed results where Menger ($p \leq 0.05$) and Zadeh ($p > 0.05$) based networks decreased, conversely the performance increased in the case of Łukasiewicz, although the effect is not significant.

TABLE I

RESULTS ON THE MONK-1, MONK-2, AND MONK-3 DATA SETS ON ACCURACY (HIGHER IS BETTER), NUMBER OF PARAMETERS (# PARAM.), INFERENCE TIME (LOWER IS BETTER) (INF.) AVERAGED OVER 10000 RUNS, AND WORKING MEMORY (MEM.).

Model	MONK-1 Acc.	MONK-2 Acc.	MONK-3 Acc.	MONK-1 Param.	MONK-2 Param.	MONK-3 Param.
ANN	100.00 $\pm$ 0.00%	100.00 $\pm$ 0.00%	93.50 $\pm$ 0.01%	162	162	162
Decision Tree	97.04 $\pm$ 1.48%	81.81 $\pm$ 0.19%	92.73 $\pm$ 0.78%	31.00 $\pm$ 0.00	103.00 $\pm$ 00	37.00 $\pm$ 00
Random Forest	95.19 $\pm$ 1.27%	74.07 $\pm$ 1.36%	96.16 $\pm$ 0.19%	6.20 $\pm$ 0.19K	12.33 $\pm$ 0.05K	4.82 $\pm$ 0.48K
XGboost	99.31 $\pm$ 0.00%	80.79 $\pm$ 1.36%	94.91 $\pm$ 0.00%	882.0 $\pm$ 0.00	2144.00 $\pm$ 0.00	778.00 $\pm$ 0.00
DLN _M	99.72 $\pm$ 4.15%	74.44 $\pm$ 5.67%	93.05 $\pm$ 2.36%	144	72	72
DLN _Z	97.77 $\pm$ 4.96%	67.96 $\pm$ 2.12%	91.43 $\pm$ 5.42%	144	72	72
DLN _L	83.28 $\pm$ 5.62%	60.50 $\pm$ 2.88%	74.72 $\pm$ 14.15%	144	72	72
DLN _{M**}	100.00 $\pm$ 0.00%	80.41 $\pm$ 9.13%	94.30 $\pm$ 2.15%	144	72	72
DLN _{Z**}	98.70 $\pm$ 2.89%	70.37 $\pm$ 5.79%	93.10 $\pm$ 3.04%	144	72	72
DLN _{L**}	100.00 $\pm$ 0.00%	61.90 $\pm$ 3.59%	87.41 $\pm$ 5.99%	144	72	72
DLN _{M**} ^m	97.22 $\pm$ 4.28%	67.36 $\pm$ 3.50%	87.87 $\pm$ 5.17%	144	72	72
DLN _{Z**} ^m	53.24 $\pm$ 2.61%	63.52 $\pm$ 4.56%	51.20 $\pm$ 8.10%	144	72	72
DLN _{L**} ^m	92.96 $\pm$ 1.08%	61.43 $\pm$ 4.26%	86.34 $\pm$ 11.56%	144	72	72
Model	MONK-1 Inf.	MONK-2 Inf.	MONK-3 Inf.	MONK-1 Mem.	MONK-2 Mem.	MONK-3 Mem.
ANN	161.20 $\pm$ 26.50 ns	161.20 $\pm$ 26.50 ns	161.20 $\pm$ 26.50 ns	648B	648B	648B
Decision Tree	273.05 $\pm$ 21.80 ns	242.00 $\pm$ 5.97 ns	238.31 $\pm$ 9.05 ns	124.00 $\pm$ 0.00B	412.00 $\pm$ 0.00B	148.00 $\pm$ 0.00B
Random Forest	13.72 $\pm$ 1.33 μ s	14.57 $\pm$ 0.41 μ s	13.11 $\pm$ 0.39 μ s	24.81 $\pm$ 0.77KB	49.34 $\pm$ 0.23KB	19.30 $\pm$ 0.48KB
XGboost	2.57 $\pm$ 0.35 μ s	2.41 $\pm$ 0.10 μ s	2.41 $\pm$ 0.15 μ s	3.52 $\pm$ 0.00KB	8.57 $\pm$ 0.00KB	3.11 $\pm$ 0.00KB
DLN _M	3.69 $\pm$ 0.52 ns	2.58 $\pm$ 0.08 ns	2.61 $\pm$ 0.30 ns	72B	36B	36B
DLN _Z	3.45 $\pm$ 0.32 ns	2.54 $\pm$ 0.32 ns	4.67 $\pm$ 0.35 ns	72B	36B	36B
DLN _L	3.24 $\pm$ 0.14 ns	2.67 $\pm$ 0.35 ns	4.30 $\pm$ 0.06 ns	72B	36B	36B
DLN _{M**}	3.33 $\pm$ 0.24 ns	2.62 $\pm$ 0.32 ns	2.42 $\pm$ 0.09 ns	72B	36B	36B
DLN _{Z**}	3.54 $\pm$ 0.16 ns	2.82 $\pm$ 0.37 ns	2.67 $\pm$ 0.34 ns	72B	36B	36B
DLN _{L**}	3.25 $\pm$ 0.20 ns	2.56 $\pm$ 0.23 ns	2.32 $\pm$ 0.11 ns	72B	36B	36B
DLN _{M**} ^m	3.40 $\pm$ 0.23 ns	2.31 $\pm$ 0.06 ns	2.43 $\pm$ 0.09 ns	72B	36B	36B
DLN _{Z**} ^m	3.23 $\pm$ 0.13 ns	3.67 $\pm$ 0.35 ns	2.59 $\pm$ 0.19 ns	72B	36B	36B
DLN _{L**} ^m	3.51 $\pm$ 0.27 ns	3.31 $\pm$ 0.06 ns	2.68 $\pm$ 0.16 ns	72B	36B	36B

TABLE II

RESULTS FOR THE ADULT DATA SET ACCURACY (HIGHER IS BETTER) (ACC.), NUMBER OF PARAMETERS (# PARAM.), INFERENCE TIME (LOWER IS BETTER) (INF. TIME) AVERAGED OVER 10000 RUNS, AND WORKING MEMORY (MEMORY).

Adult	Acc.	#Param.	Infer. Time	Memory
ANN	84.90 $\pm$ 0.04%	3,810	287.95 $\pm$ 6.19 ns	15KB
Decision Tree	82.12 $\pm$ 0.05%	7.368 $\pm$ 0.01K	193.21 $\pm$ 7.93 ns	29.46 $\pm$ 0.04KB
Random Forest	83.33 $\pm$ 0.07%	72.85 $\pm$ 0.97K	14.36 $\pm$ 0.06 μ s	2.91 $\pm$ 0.00MB
XGBoost	84.40 $\pm$ 0.00%	5.11 $\pm$ 0.00K	536.22 $\pm$ 25.47 ns	20.44 $\pm$ 0.00KB
DLN _M	73.92 $\pm$ 4.93%	1,280	33.52 $\pm$ 1.80 ns	640B
DLN _Z	75.68 $\pm$ 6.95%	1,280	28.70 $\pm$ 1.72 ns	640B
DLN _L	73.55 $\pm$ 6.62%	1,280	34.52 $\pm$ 1.80 ns	640B
DLN _{M**}	80.21 $\pm$ 1.83%	1,280	29.70 $\pm$ 1.72 ns	640B
DLN _{Z**}	80.07 $\pm$ 3.67%	1,280	35.52 $\pm$ 1.80 ns	640B
DLN _{L**}	81.41 $\pm$ 1.52%	1,280	30.70 $\pm$ 1.72 ns	640B
DLN _{M**} ^m	70.67 $\pm$ 6.16%	1,280	29.16 $\pm$ 1.64 ns	640B
DLN _{Z**} ^m	68.27 $\pm$ 8.43%	1,280	30.26 $\pm$ 2.28 ns	640B
DLN _{L**} ^m	77.77 $\pm$ 3.32%	1,280	30.16 $\pm$ 1.64 ns	640B

From an efficiency perspective, DLNs consistently demonstrate substantial advantages. All DLN variants require only 1,280 parameters and approximately 640 B of working memory, compared to the ANN with 3,810 parameters and 15 KB memory. Furthermore, DLNs achieve extremely low inference latency (28–35 ns), outperforming ANN (287.95 ns) and XGBoost (536.22 ns) by an order of magnitude, and vastly outperforming Random Forest (14.36 μ s). In different settings of DLNs show improvement in inference time for residual initialized versions comparing to the baseline (other than DLN_{**} and DLN all differences are significant with $p \leq 0.05$). These results indicate that DLNs provide an attractive trade-off when deployment constraints prioritize inference speed and memory footprint over maximum predictive accuracy.

C. Breast Cancer dataset

Compared to symbolic baselines (Table III), DLN _{α} has a measurable lower precision than Random Forest and XGboost ($p \leq 0.05$); however, Decision Tree showed an insignificant difference ($p > 0.05$). DLNs with residual initialization variants improve performance relative to base configurations of 60s% to 70s%; all differences were significant ($p \leq 0.05$). As to DLNs the best result obtained by DLN_{M**}^m at 74.00% which is comparable to the ANN and only DLN_{Z**}^m showed significant decrease in accuracy (65.71%). Furthermore, the DLN _{α **} variances stabilized the training process, especially for Zadeh based model, evidence in the smaller standard deviation. The results showed high level view in terms of computational efficiency, DLNs again exhibit a clear advantage. Their inference

TABLE III

RESULTS FOR THE BREAST CANCER DATA SET ON ACCURACY (HIGHER IS BETTER) (ACC.), NUMBER OF PARAMETERS (# PARAM.), INFERENCE TIME (LOWER IS BETTER) (INF. TIME) AVERAGED OVER 10000 RUNS, AND WORKING MEMORY (MEMORY).

Breast Cancer	Acc.	#Param.	Infer. Time	Memory
ANN	75.31 $\pm$ 0.01%	434	792.43 $\pm$ 100.23ns	1.4KB
Decision Tree	64.85 $\pm$ 2.96%	109.80 $\pm$ 2.96	1.43 $\pm$ 0.10 μ s	439.20 $\pm$ 2.96B
Random Forest	71.71 $\pm$ 1.56%	12.36 $\pm$ 0.10K	69.98 $\pm$ 2.91 μ s	49.46 $\pm$ 0.47KB
XGBoost	64.28 $\pm$ 0.00%	2.19 $\pm$ 0.00K	11.27 $\pm$ 0.01 μ s	8.79 $\pm$ 0.00KB
DLN _M	65.14 $\pm$ 3.28%	640	14.95 $\pm$ 0.44 ns	320B
DLN _Z	63.14 $\pm$ 4.21%	640	15.25 $\pm$ 0.82 ns	320B
DLN _L	63.14 $\pm$ 5.66%	640	15.95 $\pm$ 0.44 ns	320B
DLN _{M**}	73.14 $\pm$ 2.74%	640	16.25 $\pm$ 0.82 ns	320B
DLN _{Z**}	72.57 $\pm$ 1.19%	640	16.95 $\pm$ 0.44 ns	320B
DLN _{L**}	72.86 $\pm$ 2.02%	640	17.25 $\pm$ 0.82 ns	320B
DLN _{M**} ^m	74.00 $\pm$ 2.55%	640	17.95 $\pm$ 0.44 ns	320B
DLN _{Z**} ^m	65.71 $\pm$ 0.00%	640	18.25 $\pm$ 0.82 ns	320B
DLN _{L**} ^m	72.57 $\pm$ 1.86%	640	18.95 $\pm$ 0.44 ns	320B

time remains extremely low (14 – 19 ns) and their memory footprint is minimal (320 B), whereas XGBoost requires 11.27 μ s, ANN requires 792.43 ns, and Random Forest requires 63.98 μ s. Thus, DLNs provide strong efficiency benefits, but these gains come at the cost of a substantial reduction in predictive performance.

D. MNIST Dataset

Compared to the previous datasets, MNIST is much more challenging since it involves image classification and higher-dimensional inputs. Table IV presents results on the MNIST dataset using two different image sizes: a reduced 20×20 (small) version and the standard 28×28 (normal) version.

For the MNIST small dataset, the ANN and others symbolic baselines achieved the relatively high accuracy. In comparison, between DLNs variants the model with the highest accuracy was DLN_M at 92.53%; all differences were significant ($p \leq 0.05$). The models with residual initialization showed significant decrease in performance for Menger and Zadeh based but increase for Łukasiewicz based, whereas the mirror versions perform noticeably worse in comparison with other models especially in the case of DLN_{Z**m} at 25.68% indicating that some configurations are not well-suited for image data.

Similarly, for the MNIST large dataset, the ANN and other baselines achieved very high accuracy, while the highest accuracy for DLNs was DLN_M at 97.52%; all differences were measurable with $p \leq 0.05$, showing that DLNs can reach competitive accuracy on MNIST when using the right configuration. However, other DLN variants show unstable performance. For example, DLN_{Z**} drops to 77.61%, showing that DLN performance depends heavily on the choices of hyperparameters. When increase the size of the model from small to normal size, Łukasiewicz and Menger based method had lower standard deviation. This suggested that not only increase the size of the model can improve accuracy but it also stabilized the training process.

For the reduced MNIST case, DLNs required approximately 0.79–1.00 μ s per inference, which was substantially faster than ANN inference time (3.66 μ s). For the standard MNIST case, DLNs remained around 81–85 μ s, while the ANN required

approximately 1009 μ s. Additionally, DLNs had significantly smaller memory footprints compared to the ANN, particularly for the 28×28 dataset where the ANN required 86 MB. These results highlight the efficiency advantage of DLNs, especially when scaling to higher dimensional inputs.

V. CONCLUSIONS

This work evaluated the performance of Differential Logic Gate Networks across several modalities of datasets. The experimental results show that DLNs consistently provide an extremely lightweight and efficient inference framework while maintaining competitive predictive performance under appropriate configurations. Across all datasets, DLNs demonstrate an advantage in terms of efficiency. Inference time and memory usage is consistently reduced by orders of magnitude compared to traditional machine learning as well as ANNs. These properties make DLNs particularly suitable for real-time and resource-constrained environments, such as embedded devices, edge computing systems, and low-power medical applications where conventional ensemble models or large neural networks are not feasible [2]. The results imply symbolic and ANNs model tend to have a more stable results regarding to accuracy as evidence of smaller standard deviation to DLNs. Adding residual initialization or increase the size of the model benefits the training process. However, RI is still insufficient to mitigate gradient vanishing. These suggest a solution to stabilize the models during training, similar to residual connection is needed. In contrast, due to the nature of tree-based algorithm the size of the models are not always consistent on the same hyperparameters and dataset, consequently, this effects the inference time. Whereas in DLNs only the OS-noise contributes to the variance in inference time. Overall, the results indicate that depending on the configuration the accuracy of DLNs varies based on the characteristic of the provided inputs and it show a highly favorable trade-off between accuracy and computational cost. While standard baseline methods may still outperform DLNs in peak accuracy on certain datasets, DLNs offer a practical and scalable alternative when inference speed and memory footprint are the primary constraints. Future work may focus

TABLE IV

RESULTS FOR THE MNIST DATA SET ACCURACY (HIGHER IS BETTER) (ACC.), NUMBER OF PARAMETERS (# PARAM.), INFERENCE TIME (LOWER IS BETTER) (INF. TIME) AVERAGED OVER 100 RUNS AVERAGED OVER 10000 RUNS, WHERE OPS STAND FOR NUMBER OF BINARY OPERATIONS NEEDED TO PERFORM COMPUTATIONS, ALL VARIATION OF DLN IN A CATEGORY (SMALL OR NORMAL) HAVE THE SAME #PARAM., MEMORY, OPS.

MNIST	MNIST 20x20 (small)			MNIST 28x28 (normal)		
	Acc.	Infer. Time	#Param. Memory Ops	Acc.	Infer. Time	#Param. Memory Ops
DLN _M	92.53 ± 0.24%	0.99 ± 0.02 μ s		97.52 ± 0.38%	84.60 ± 3.59 μ s	
DLN _Z	83.98 ± 3.74%	0.79 ± 0.05 μ s		96.41 ± 0.41%	81.14 ± 0.21 μ s	
DLN _L	58.24 ± 3.00%	0.99 ± 0.02 μ s		95.48 ± 0.58%	84.60 ± 3.59 μ s	
DLN _{M**}	90.71 ± 0.41%	0.79 ± 0.05 μ s	48K	97.37 ± 0.08%	81.14 ± 0.21 μ s	512K
DLN _{Z**}	84.51 ± 1.38%	1.00 ± 0.02 μ s	24KB	77.61 ± 1.72%	84.60 ± 3.59 μ s	250KB
DLN _{L**}	81.74 ± 2.32%	0.79 ± 0.05 μ s	48K	97.23 ± 0.13%	81.14 ± 0.21 μ s	512K
DLN _{M***m}	84.50 ± 1.47%	1.00 ± 0.02 μ s		96.78 ± 0.29%	84.60 ± 3.59 μ s	
DLN _{Z***m}	25.68 ± 3.09%	0.79 ± 0.05 μ s		86.12 ± 0.62%	81.14 ± 0.21 μ s	
DLN _{L***m}	71.44 ± 4.81%	1.00 ± 0.02 μ s		95.84 ± 0.62%	84.60 ± 3.59 μ s	
Baselines	Acc.		#Param.	Memory	Infer. Time	Ops
ANN 20x20 (small)	97.92 ± 0.08%		118.28K	462KB	3.66 ± 0.15 μ s	236M
Decision Tree 20x20 (small)	87.69 ± 0.11%		8.53 ± 0.00K	34.13 ± 0.02KB	570.23 ± 2.48 ns	2.18M
Random Forest 20x20 (small)	96.97 ± 0.05%		98.00 ± 0.69K	3.92 ± 0.00MB	32.66 ± 0.15 μ s	250.88M
XGboost 20x20 (small)	97.84 ± 0.00%		79.45 ± 0.00K	317.83 ± 0.00KB	5.35 ± 0.06 μ s	5.06M
ANN 28x28 (normal)	98.40 ± 0.06%		22.60M	86MB	1009.17 ± 9.23 μ s	45G
Decision Tree 28x28 (normal)	86.92 ± 0.11%		78.75 ± 1.79K	31.50 ± 0.00KB	397.14 ± 3.56 ns	5.04M
Random Forest 28x28 (normal)	96.71 ± 0.03%		999.60 ± 1.46K	3.99 ± 0.00MB	26.01 ± 0.13 μ s	63.97M
XGboost 28x28 (normal)	97.81 ± 0.00%		71.50 ± 0.00K	286.00 ± 0.00KB	5.15 ± 0.06 μ s	4.57M

on improving DLN training stability, robustness on noisy datasets, extending the architecture to further close the remaining accuracy gap, reduce training time and remove the need for discretizations of continuous features while preserving the efficiency advantages demonstrated in this study.

REFERENCES

- [1] B. Becker and R. Kohavi, "Adult," UCI Machine Learning Repository, 1996. Available: <https://doi.org/10.24432/C5XW20>
- [2] S. Bosse, "IoT and Edge Computing using virtualized low-resource integer Machine Learning with support for CNN, ANN, and Decision Trees," *Proc. 18th Conf. Comput. Sci. Intell. Syst. (FedCSIS)*, vol. 35, pp. 367–376, 2023, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Slezak, Eds., IEEE. Available: <http://dx.doi.org/10.15439/2023F7745>
- [3] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: <https://doi.org/10.1023/A:1010933404324>.
- [4] T. Chen and C. Guestrin, "XGBoost: a Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, vol. 1, no. 1, pp. 785–794, Aug. 2016, doi: <https://doi.org/10.1145/2939672.2939785>.
- [5] L. da Cruz, C. Sierra-Franco, G. Silva-Calpa, and A. Raposo, "Enabling Autonomous Medical Image Data Annotation: A human-in-the-loop Reinforcement Learning Approach," in *Proc. 16th Conf. on Computer Science and Intelligence Systems (FedCSIS)**, vol. 25, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Slezak, Eds., IEEE, 2021, pp. 271–279. Available: <http://dx.doi.org/10.15439/2021F86>
- [6] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014. Available: <https://arxiv.org/abs/1412.6980>
- [7] E. van Krieken, E. Acar, and F. van Harmelen, "Analyzing Differentiable Fuzzy Logic Operators," *Artificial Intelligence*, vol. 302, 2022, pp. 103602. Available: <https://www.sciencedirect.com/science/article/pii/S0004370221001533>
- [8] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," Univ. of Toronto, 2012.
- [9] Y. LeCun, C. Cortes, and C. J. C. Burges, "The MNIST database of handwritten digits," 1998. Available: <http://yann.lecun.com/exdb/mnist/>
- [10] K. Menger, "Statistical Metrics," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 28, no. 12, Dec. 1942, pp. 535–537. Available: <https://doi.org/10.1073/pnas.28.12.535>
- [11] D. B. de Oliveira, D. Casini and T. Cucinotta, "Operating System Noise in the Linux Kernel," in *IEEE Transactions on Computers*, vol. 72, no. 1, pp. 196–207, 1 Jan. 2023, doi: <https://doi.org/10.1109/TC.2022.3187351>.
- [12] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," 2019. Available: <https://arxiv.org/abs/1912.01703>
- [13] F. Petersen, C. Borgelt, H. Kuehne, and O. Deussen, "Deep Differentiable Logic Gate Networks," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 2006–2018. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/0d3496dd0cec77a999c98d35003203ca-Paper-Conference.pdf
- [14] F. Petersen, H. Kuehne, C. Borgelt, J. Welzel, and S. Ermon, "Convolutional Differentiable Logic Gate Networks," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 121185–121203. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/db988b089d8d97d0f159c15ed0b6a71-Paper-Conference.pdf
- [15] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, Mar. 1986, doi: <https://doi.org/10.1007/bf00116251>.
- [16] C. D. Nguy and P. Wasilewski, "Deep Differentiable Logic Gate Networks Based on Fuzzy Łukasiewicz T-norm," 2025 20th Conference on Computer Science and Intelligence Systems (FedCSIS), Kraków, Poland, 2025, pp. 219–230, doi: <https://doi.org/10.15439/2025F1666>
- [17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, Oct. 1986, pp. 533–536. Available: <https://doi.org/10.1038/323533a0>
- [18] S. Thrun et al., "The Monk's Problem's: A Performance Comparison of Different Learning Methods," 1991. Available: <https://api.semanticscholar.org/CorpusID:59810521>
- [19] P. Wasilewski and C. D. Nguy, "Deep Differentiable Logic Gate Networks Based on Fuzzy Zadeh's T-Norm," *Lecture Notes in Networks and Systems*, pp. 57–70, 2026, doi: https://doi.org/10.1007/978-3-032-04197-5_5.
- [20] P. Wasilewski, and Ch. D. Nguy, "Deep Differentiable Logic Gate Networks: Learning with Mirror Gates," *Proceedings of 6th Polish Conference on Artificial Intelligence (PP-RAI 2026)*, Krakow, Poland, Apr. 20–22, 2026, to appear in *Lecture Notes in Networks and Systems*, Springer.
- [21] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, 1965, pp. 338–353. Available: <https://www.sciencedirect.com/science/article/pii/S001999586590241X>
- [22] M. Zwitter and M. Soklic, "Breast Cancer," UCI Machine Learning Repository, 1988. Available: <https://doi.org/10.24432/C51P4M>