Proceedings of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 219–230
DOI: 10.15439/2025F1666
ISSN 2300-5963 ACSIS, Vol. 43

Deep Differentiable Logic Gate Networks Based on Fuzzy Łukasiewicz T-norm

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*—Differentiable Logic Gate Networks (DLNs) offer a compelling framework for symbolic interpretability and reducing inference cost. Building on prior works using Menger and Zadeh T-norms, we investigate the Łukasiewicz T-norm as an alternative relaxation for classical logic gates. While it provides strong gradients in some regions, its flat areas result in vanishing gradients that hinder training. To address this issue, we use an initialization strategy that is analogous to residual connection in Neural Networks to encourage error signal propagation during training. Our empirical results show that Łukasiewicz based DLNs, though slightly less accurate, benefit from faster inference and lower memory requirements compared to Neural Networks, giving the opportunity of practical application in, e.g., resource constrained devices. Due to the structural clarity, DLNs facilitate direct inspection and tracing of information flow, which makes them suitable for application in explainable artificial intelligence (XAI).**

I. Introduction

DIFFERENTIABLE logic gate networks (DLNs) provide a promising framework for combining the interpretability of classical logic circuits with the scalability of gradient-based learning. Prior work has demonstrated that continuous relaxations of binary logic gates, such as those based on the Zadeh and Menger T-norms enable the use of standard backpropagation techniques while allowing conversion back to classical logic post training [19], [28], [15], [29]. Building on this foundation, we extend the study of T-norm-based relaxations by investigating the Łukasiewicz T-norm as a candidate operator for differentiable logic networks.

In production environments, inference efficiency directly affects user experience, operational costs, and energy consumption, making it a critical optimization target, especially for edge devices and real time systems [2], [21] with strict latency and resource constraints. Common strategies such as reduced-precision computation [3], [7], binary networks [22], and sparsity exploitation [8], [16] offer partial solutions but often come with trade-offs in accuracy or generality. DLNs offer an alternative path, where logic gate networks were trained using a continuous relaxation based on the fuzzy logic functions to enable gradient descent. Each computational unit in DLNs learns a probability distribution over 16 of these differentiable fuzzy logic functions using softmax. Post training, these units are assigned the most probable gate, resulting in networks that are naturally sparse and require no weights, leading to exceptional inference speeds. In our previous work [28], we followed this approach using the Zadeh T-norm for continuous relaxation. Although it is not differentiable at $a = b$, we assume the output to be a in those cases to preserve gradient flow. In this paper, we extend this line of research by exploring the Łukasiewicz T-norm as a relaxation operator for logic gate networks. Although similar to the Zadeh T-norm, the Łukasiewicz operator provides strong gradients but only in certain regions; however, it suffers from wide, flat areas with vanishing gradients. To address this, we use initialization strategies [20] that bias the network toward gate configurations that better propagate gradients, especially during the early stages of training.

Recent research on medical applications delving into large-data processing such as MRI [27], [14], EEG [6], [11] data, X-ray images [4] and clinical notes [5] which requires high computational resources. The introduction of DLNs open a new opportunity for processing relatively small data with low latency and small memory footprint e.g. pulsimeter, oxygen blood saturation measuring and non-invasive glucose monitoring.

II. Differentiable Logic Gate Networks

Hardware implementations utilizing logic gates exhibit exceptional performance characteristics nanosecond execution speeds, true parallelism, and deterministic reliability. However, designing complex systems at the gate level presents formidable challenges. Development with Verilog or VHDL lacks convenient debugging tools, requires specialized equipment for verification, and involves lengthy development cycles [23]. These constraints necessitate novel approaches such as symbolic learning that can select optimal logic gate configurations, dramatically reducing design cycles. On the other hand, binary logic gates are inherently discontinuous, making

gradient based training infeasible. Inspired by this, Differentiable Logic Gate Networks was introduced by [19] and further investigated by [28], offering an elegant solution to this challenge by incorporating principles from Differentiable Fuzzy Logics (DFL) [10].

Instead of operating on discrete values $\{0, 1\}$, DLNs allow inputs and outputs to take continuous values in the range $[0, 1]$. This enables smooth transitions between logical states and creates differentiable pathways through which gradients can flow during training.

Architecturally, DLNs differ from standard neural networks in their connection patterns. Each computational unit (analogous to a neuron) processes a pair of randomly assigned inputs, rather than computing a weighted sum across all inputs. This design results in inherently sparse connections, in contrast to the dense layers typical in traditional neural networks.

The core learning mechanism of DLNs involves dynamically selecting the most appropriate fuzzy logic operation at each unit to minimize the overall loss function. Unlike neural networks, which learn weights, DLNs learn which logical operations best model the underlying patterns in the data.

A Boolean operation can be defined as a function $f : \{0, 1\} \times \{0, 1\} \to \{0, 1\}$, leading to a total of 16 possible Boolean operations (Table I). Each of these can be extended to fuzzy logic. For example, the Boolean AND ($\wedge$) operation can be defined as:

$$a \wedge b = \begin{cases} 1 & \text{if } a = b = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The fuzzy logic equivalent to Boolean AND using Menger's T-norm [15] is:

$$T_M(a, b) = a \cdot b.$$

To represent the fuzzy logic operation probabilistically, each computational unit can be represented as:

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

where the probabilities p_i are derived from a softmax over a learnable logit vector $\mathbf{w}$:

$$p_i = \frac{e^{w_i}}{\sum_{j=0}^{15} e^{w_j}}, \quad \text{for } i = 0, 1, \ldots, 15.$$

Each f_i denotes a fuzzy logic function corresponding to one of the 16 Boolean operations, table I.

Each layer in a DLN consists of several such units, and their outputs propagate forward to support complex decision-making processes. The choice of T-norm is a crucial hyperparameter, influencing both accuracy and inference speed, which we explore in the following subsections.

A. Menger Based Differentiable Logic Gate Networks

The original DLN formulation [19] used Menger's T-norm ($T_M(a, b) = a \cdot b$), also known as the probabilistic T-norm. This choice is intuitive due to its smooth and easily computable derivatives:

$$\frac{\partial T_M(a, b)}{\partial a} = b, \quad \frac{\partial T_M(a, b)}{\partial b} = a,$$

which are well-suited for gradient based optimization algorithms [24], [10].

Empirical results show that models based on the Menger T-norm achieve competitive accuracy compared to neural networks on many benchmarks while offering significantly lower memory usage and computational complexity due to their reliance on binary logic operations. Compared to other T-norms [28], Menger based DLNs (DLNs_M) demonstrate robustness to noise, albeit with slightly slower inference times. However, the DLN_M variant suffers from the vanishing gradient problem, as the partial derivatives diminish when $\partial T_M(a, b)/\partial a < 1$, causing the backpropagated error signal to attenuate and hindering effective learning in deeper networks. To address this [19] proposed scaling the gradients by a factor $f > 1$ for DLNs with more than six layers. While this technique mitigates the vanishing gradient problem, it also introduces the risk of exploding gradients, potentially destabilizing training.

B. Zadeh Based Differentiable Logic Gate Networks

To mitigate gradient vanishing, [28] proposed replacing Menger's T-norm with Zadeh's T-norm [29], defined as $T_Z(a, b) = \min(a, b)$. While the forward computation is straightforward, care must be taken during backpropagation to ensure consistent gradient flow, especially when $a = b$, where the gradient could flow to either input.

To address this, the authors [28] chose to default to propagating the gradient through a in the case of equality (although choosing b would yield the same results). The partial derivatives are defined as:

$$\frac{\partial T_Z(a, b)}{\partial a} = \begin{cases} 1 & \text{if } a \leq b, \\ 0 & \text{otherwise,} \end{cases} \quad \frac{\partial T_Z(a, b)}{\partial b} = \begin{cases} 0 & \text{if } a \leq b, \\ 1 & \text{otherwise.} \end{cases}$$

This setup ensures a well-defined and consistent backward pass. One key advantage of Zadeh's T-norm is that the derivatives of all fuzzy operations are limited to the set $\{-1, 0, 1\}$, and it is never simultaneously zero for both a and b. This guarantees that the gradient is neither vanishing nor exploding, enabling more stable and effective training.

Consequently, Zadeh based DLNs (DLNs_Z) are good at recognizing discrete, singular features and outperform DLNs_M in inference speed, albeit with a slight trade off in accuracy on some benchmarks [28].

C. Binary Logic Gate Networks

Compared to traditional neural networks, the inference process in Differentiable Logic Networks can be significantly less efficient due to the need to compute all 16 fuzzy logic operations for every computational unit. In contrast, neural networks rely heavily on matrix multiplication as their computational backbone, a process that is now widely accelerated

by modern hardware, such as Nvidia's Tensor Cores, among others.

One potential optimization is to reduce the number of fuzzy logic operations computed by selecting only the logic gate with the highest probability for each unit. Since fuzzy operations generalize Boolean operations, when inputs are strictly binary (i.e., 0 or 1), the resulting outputs are the same as the binary connectives equivalent; converting from smooth fuzzy operators to crisp binary ones theoretically does not compromise accuracy, especially when the training data is already binary. This transition could also substantially reduce memory consumption and computational complexity due to the efficiency of binary operations.

To implement this optimization, DLNs can be converted into Binary Logic Gate Networks (LGNs) by choosing the fuzzy operation where the softmax function results in the highest probability; thus, this retains the same architectural structure, namely, the number of layers and computational units; however, each unit in LGNs only computes a single binary logic gate instead of a weighted sum of all 16 functions.

III. Łukasiewicz based Differentiable Logic Gate Networks

In this study, we investigate fuzzy logic operations based on the Łukasiewicz T-norm, defined as $T_Ł(a,b) = \max\{0, a+b-1\}$. Like the Zadeh T-norm used in [28], the Łukasiewicz T-norm relies on $\min$ and $\max$ operators. Consequently, it exhibits similar challenges during backpropagation, particularly regarding non smooth gradients and ambiguous gradient flow paths.

To address this, we adopt the same backward pass strategy proposed in [28], which resolves the gradient ambiguity by enforcing consistent choices. The partial derivatives with respect to the inputs are defined as:

$$\frac{\partial T_Ł(a,b)}{\partial a} = \begin{cases} 0 & \text{if } a+b-1 \leq 0, \\ 1 & \text{otherwise,} \end{cases}$$

$$\frac{\partial T_Ł(a,b)}{\partial b} = \begin{cases} 0 & \text{if } a+b-1 \leq 0, \\ 1 & \text{otherwise.} \end{cases}$$

This yields sharp gradients in the set $\{-1, 0, 1\}$ for the network, which can help limit exploding gradients. However, a significant portion of the domain of $T_Ł$ results in zero gradients, leading to the risk of vanishing gradients during training.

To mitigate this, we implement a residual initialization scheme inspired by [20]. Instead of initializing the learnable weights uniformly at random, the authors bias them toward logic gates that act as pass through operators. Owing to the symmetry between inputs, they treat gates corresponding to A and B (i.e., the 3rd and 5th gates in Table I) as functionally equivalent in this context. We experiment with two variants of this initialization: one where gate A is assigned a 90% initial probability (with all other gates initialized to 0.67%) as in [20], and in this work we introduce another variant where both gates A and B are initialized with 45% probability each.

For inference, we transform the computational units of Differentiable Logic Networks into hard logic gates and generate corresponding C code for the network. This C code is compiled using the `-O1` optimization flag for models with fewer than 50,000 computational units and `-O0` for larger models due to limitations in compilation time and memory overhead.

Modern 64-bit CPUs are ubiquitous across consumer devices, which implies that Boolean values must typically be extended via zero extension or sign extension to align with the ALU's expected input width. The implementation proposed by [19] uses a 64-bit encoding for Boolean inputs to match the native word size of contemporary CPUs. For consistency, our initial implementation of Zadeh-based Differentiable Logic Networks (DLNs) [28] followed the same encoding strategy. However, empirical evaluation with our available devices for experiments revealed that using an 8-bit encoding for Boolean values significantly reduces inference time. This change in using different numbers of bits to represent Boolean values does not affect the accuracy of the models. Therefore, in order to maximize performance under our setup, we proceed with using 8-bit encodings for all classical logic operations in the networks instead of 64-bit.

IV. Experiments and Results

To ensure a fair comparison with previous work [19], [28], we evaluate our proposed approach on both structured and unstructured data types. For structured datasets, categorical attributes are one-hot encoded, while continuous features such as age or tumor size are discretized and then followed by one-hot encoding. This preprocessing ensures compatibility with post training discretized models that require binary or discrete input representations. For unstructured data such as CIFAR-10 [12], we adopt a similar discretization strategy as [19], [28] for pixel intensities by applying a series of progressive thresholds, enabling a finer grained binary encoding of continuous values. This technique reduces information loss during binarization and supports more expressive representations in logic based models. We created two variants of the CIFAR-10 dataset for 3 thresholds and 31 thresholds, which we will discuss in more detail in CIFAR-10. In contrast, for the MNIST dataset, we retain the original grayscale pixel values and train directly on the real valued inputs to evaluate DLNs' performance on real values data.

During evaluation, all DLNs, regardless of implementation, are converted into Binary Logic Gate Networks where the computational unit calculates fixed Boolean functions with the highest probability. All the variants of $\text{DLN}_Ł$ are denoted as $\text{DLN}_Ł$, $\text{DLN}_{Ł^*}$, $\text{DLN}_{Ł^{**}}$ for without residual initialization, residual initialization at the feedforward gate A, and residual initialization at gates A and B. All models are implemented using PyTorch [18], Adam optimizer [9] and trained on an NVIDIA RTX 2000 Ada GPU. Inference time is measured using a single threaded setup on an AMD Ryzen 5 3550H CPU and averaged across 100 runs. To indicate statistically significant differences in inference time across models, we

TABLE I
LIST OF ALL ŁUKASIEWICZ T-NORM LOGIC GATES.

ID	Operator	Łukasiewicz T-norm	00	01	10	11
0	$False$	0	0	0	0	0
1	$a \wedge b$	$max\{0, a+b-1\}$	0	0	0	1
2	$\neg(a \Rightarrow b)$	$max\{0, a-b\}$	0	0	1	0
3	a	a	0	0	1	1
4	$\neg(a \Leftarrow b)$	$max\{0, -a+b\}$	0	1	0	0
5	b	b	0	1	0	1
6	$\neg(a \Leftrightarrow b)$	$min\{1, max\{0, a-b\} + max\{0, -a+b\}\}$	0	1	1	0
7	$a \vee b$	$min\{1, a+b\}$	0	1	1	1
8	$\neg(a \vee b)$	$max\{0, 1-a-b\}$	1	0	0	0
9	$a \Leftrightarrow b$	$max\{0, min\{1, 1-a+b\} + min\{1, 1+a-b\} - 1\}$	1	0	0	1
10	$\neg b$	$1-b$	1	0	1	0
11	$a \Leftarrow b$	$min\{1, 1+a-b\}$	1	0	1	1
12	$\neg a$	$1-a$	1	1	0	0
13	$a \Rightarrow b$	$min\{1, 1-a+b\}$	1	1	0	1
14	$\neg(a \wedge b)$	$min\{1, 2-a-b\}$	1	1	1	0
15	$True$	1	1	1	1	1

TABLE II
RESULTS ON THE MONK DATA SETS AVERAGED OVER 100 RUNS.

Method	MONK-1	MONK-2	MONK-3
ANN	100.00%	100.00%	93.5%
DLN_M	100.00%	78.24%	95.37%
DLN_Z	100.00%	89.32%	91.74%
$DLN_Ł$	100.00%	56.94%	93.98%
$DLN_{Ł^*}$	100.00%	61.57%	78.24%
$DLN_{Ł^{**}}$	100.00%	49.77%	69.44%
	# Parameters	Inf. Time	Memory
ANN	162	161.20 ± 26.50ns	648B
DLN_M	144 | 72 | 72	8.06 ± 0.17ns	72B | 36B | 36B
DLN_Z	144 | 72 | 72	7.77 ± 0.13ns	72B | 36B | 36B
$DLN_Ł$	144 | 72 | 72	7.20 ± 0.12ns	72B | 36B | 36B
$DLN_{Ł^*}$	144 | 72 | 72	6.83 ± 0.16ns	72B | 36B | 36B
$DLN_{Ł^{**}}$	144 | 72 | 72	6.78 ± 0.15ns	72B | 36B | 36B

denote: $p \in [0.001, 0.5)$ by *, $p \in [0.0001, 0.001)$ by **, and $p \in [0, 0.0001)$ by ***.

A. MONK

The MONK datasets [26] constitute a benchmark suite of binary classification tasks designed to evaluate the performance of machine learning models on symbolic and logical reasoning problems. Each dataset consists of discrete-valued features and a binary target label, with the classification rules defined using logical expressions.

MONK-1 defines a concept in disjunctive normal form (DNF), making it relatively simple and well suited for symbolic learners such as Differentiable Logic Gate Networks. **MONK-2** resembles a parity problem, where the target function is based on a combination of attributes that cannot be easily expressed in either DNF or conjunctive normal form (CNF). **MONK-3** is structurally similar to MONK-1 and is also defined using DNF. However, it introduces label noise into the training set, making it a useful benchmark for evaluating a model's robustness and generalization under imperfect data.

The ANN achieves the highest accuracy on the MONK-1 and MONK-2 datasets, but it requires significantly more memory than DLNs, regardless of the implementation. DLN_M outperforms all other models on the MONK-3 dataset. Interestingly, $DLN_Ł$ achieves a higher accuracy than the ANN (94.0% vs. 93.5%) in the last dataset (see table II), results of significant difference tests are presented in table III and basic statistical analysis in figure 1. Adding residual initializations to $DLN_Ł$ yields mixed results: accuracy improves on MONK-2 but decreases on the other datasets. In terms of inference time, the $DLN_{Ł^*}$ $DLN_{Ł^{**}}$ methods show similar improvement in performance.

TABLE III
MONK DATA SET SIGNIFICANT INFERENCE TIME DIFFERENCE TESTS. WE DENOTE: $p \in [0.001, 0.5)$ BY *, $p \in [0.0001, 0.001)$ BY **, AND $p \in [0, 0.0001)$ BY ***

	DLN_M	DLN_Z	$DLN_Ł$	$DLN_{Ł^*}$	$DLN_{Ł^{**}}$
DLN_M			***	***	***
DLN_Z			*	***	***
$DLN_Ł$	***	*			*
$DLN_{Ł^*}$	***	***			
$DLN_{Ł^{**}}$	***	***	*		

B. Adult dataset

The Adult dataset [1], also known as the "Census Income" dataset, is a widely used benchmark in the machine learning community. It originates from the UCI Machine Learning Repository and involves a binary classification task aimed at predicting whether an individual's annual income exceeds $50,000. The dataset comprises 48,842 samples and includes both continuous and categorical features, such as age, education, occupation, and work class.

TABLE IV
RESULTS FOR THE ADULT DATA SET AVERAGED OVER 100 RUNS.

Adult	Acc.	#Param.	Infer. Time	Memory
ANN	84.90%	3,810	287.95 ± 6.19ns	$15KB$
DLN_M	84.83%	1,280	59.47 ± 2.34ns	640B
DLN_Z	75.40%	1,280	71.12 ± 0.45ns	640B
$DLN_Ł$	82.62%	1,280	94.98 ± 0.88ns	640B
$DLN_{Ł^*}$	84.57%	1,280	59.99 ± 2.02ns	640B
$DLN_{Ł^{**}}$	84.59%	1,280	81.27 ± 1.91ns	640B

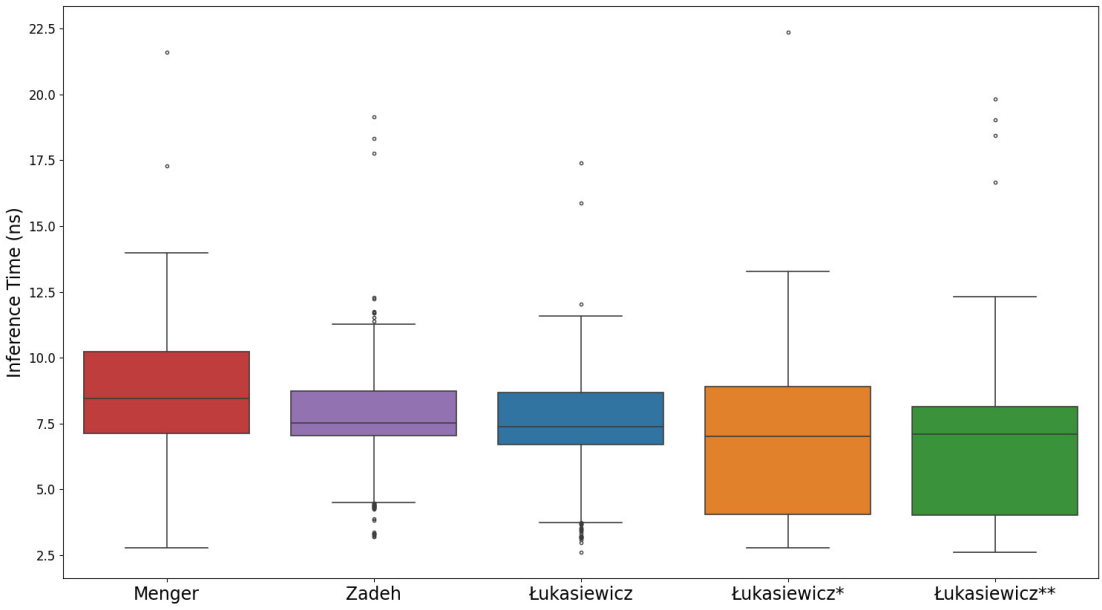

Fig. 1. Basic statistical analysis for MONK dataset inference time (ns) results (lower is better)

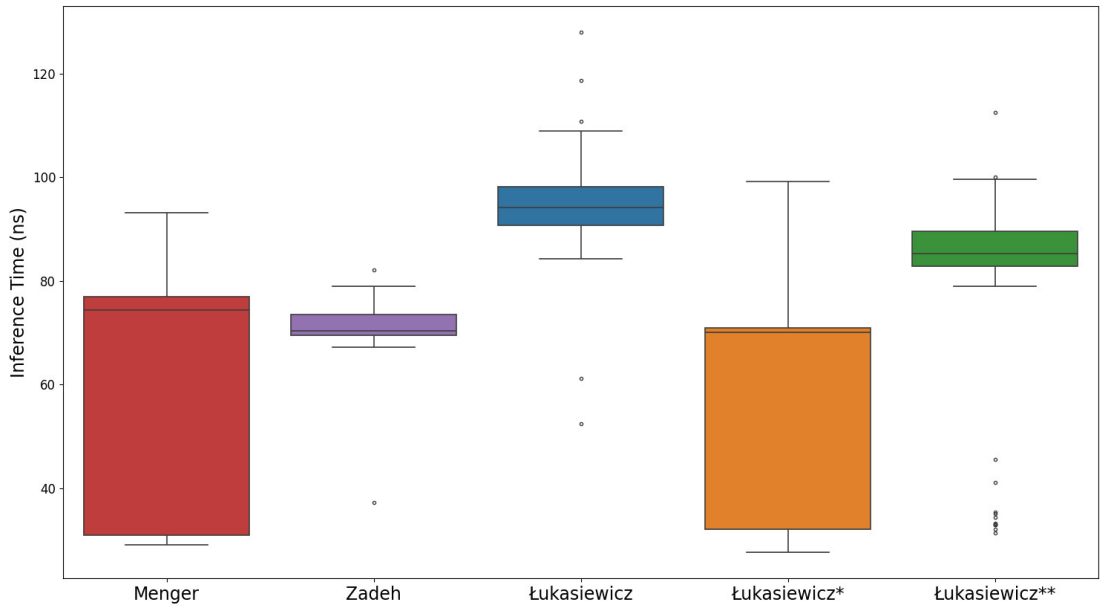

Fig. 2. Basic statistical analysis for Adult dataset inference time (ns) results (lower is better)

TABLE V
ADULT DATA SET SIGNIFICANT INFERENCE TIME DIFFERENCE TESTS. WE DENOTE: $p \in [0.001, 0.5)$ BY *, $p \in [0.0001, 0.001)$ BY **, AND $p \in [0, 0.0001)$

	DLN_M	DLN_Z	$DLN_Ł$	$DLN_{Ł^*}$	$DLN_{Ł^{**}}$
DLN_M		***	***		***
DLN_Z	***		***	***	***
$DLN_Ł$	***	***		***	***
$DLN_{Ł^*}$		***	***		***
$DLN_{Ł^{**}}$	***	***	***	***	

TABLE VII
BREAST CANCER DATA SET SIGNIFICANT INFERENCE TIME DIFFERENCE TESTS. WE DENOTE: $p \in [0.001, 0.5)$ BY *, $p \in [0.0001, 0.001)$ BY **, AND $p \in [0, 0.0001)$

	DLN_M	DLN_Z	$DLN_Ł$	$DLN_{Ł^*}$	$DLN_{Ł^{**}}$
DLN_M		***	***	***	*
DLN_Z	***		***	**	
$DLN_Ł$	***	***		*	***
$DLN_{Ł^*}$	***	**	*		*
$DLN_{Ł^{**}}$	*		***	*	

In this benchmark, DLN_M, $DLN_{Ł^*}$, and $DLN_{Ł^{**}}$ demonstrate comparable accuracy to the artificial neural network (ANN), but with much lower inference time and memory usage (table IV), inference time significant difference tests are in table V and see figure 2 for basic statistical analysis. Among all implementations, DLN_M is the fastest in terms of inference speed and similar accuracy compared to the baseline ANN. Interestingly, when we initialize the Łukasiewicz models with higher probabilities at gates A and B, the accuracy doesn't improve, but inference time goes down. This shows $DLN_{Ł^*}$ a good compromise between speed and performance.

C. Breast Cancer dataset

The Breast Cancer dataset [30] was collected from the University Medical Center, Institute of Oncology, Ljubljana, Yugoslavia. It is one of three domains provided by the Oncology Institute that have frequently appeared in the machine learning literature.

The dataset consists of 286 instances, with 201 belonging to one class and 85 to another. Each instance is described by nine attributes. The dataset is commonly used to assess classification performance on medical diagnostic tasks, particularly in scenarios involving imbalanced class distributions and heterogeneous feature types.

TABLE VI
RESULTS FOR THE BREAST CANCER DATA SET AVERAGED OVER 100 RUNS.

Breast Cancer	Acc.	#Param.	Infer. Time	Memory
ANN	75.31%	434	792.43 ± 100.23ns	1.4KB
DLN_M	71.43%	640	38.34 ± 0.27ns	320B
DLN_Z	68.56%	640	35.14 ± 0.22ns	320B
$DLN_Ł$	70.00%	640	26.72 ± 1.07ns	320B
$DLN_{Ł^*}$	74.29%	640	30.62 ± 1.25ns	320B
$DLN_{Ł^{**}}$	70.00%	640	35.58 ± 1.10ns	320B

In this experiment, table VI shows ANN reaches an accuracy of 75.3%, followed closely by $DLN_{Ł^*}$ at 74.3%, while using only a fraction of the inference time and memory. $DLN_Ł$ stands out as the fastest model, although it has very low accuracy. The results of significant difference tests for inference speed are presented in table VII, and essential statistical analysis is presented in figure 3. Notably, using residual initialization brings a major boost for $DLN_Ł$, raising its accuracy from 70.0% to 74.3%. However, when both gates A and B are initialized together, the model's accuracy actually drops. Once again, $DLN_{Ł^*}$ has a good balance between accuracy and run time complexity.

D. MNIST Dataset

To ensure comparability with prior studies [19], [28], we evaluate our proposed approach on the MNIST dataset [13]. MNIST is a standard benchmark in computer vision and machine learning, consisting of grayscale images of handwritten digits ranging from 0 to 9. The dataset comprises 70,000 samples, partitioned into 60,000 training images and 10,000 test images, each originally sized at 28×28 pixels.

To examine the models' adaptability to reduced input dimensionality, we additionally resize the images to 20×20 pixels by removing the borders, thereby constructing a secondary, lower-resolution version of the dataset. Each image is associated with a label corresponding to the digit it represents, making this a ten class classification problem. The dataset serves as a robust testbed for evaluating both accuracy and generalization across a range of model architectures.

In both the small and normal configurations, the results in table VIII show ANNs outperform DLNs in terms of accuracy across all implementations (ANN *(small)*: 96.3%, ANN: 98.40%), though this comes at the cost of higher inference time and memory usage. Among the smaller DLN models, DLN_M *(small)* achieves the highest accuracy, but it is the slowest ($1.127 \pm 0.03\mu$s) see table IX and figure 4 for inference speed significant difference tests and statistical analysis. On the other end, $DLN_Ł$ *(small)* offers the fastest inference but with the lowest accuracy. $DLN_{Ł^{**}}$ *(small)* yields higher accuracy than $DLN_{Ł^*}$ *(small)* and on average, it also has a faster inference time; however, this is not a statistically significant increase. Both initialization strategies show clear improvements in accuracy compared to the uninitialized base model.

It's also worth pointing out that DLN_M and $DLN_{Ł^*}$ have comparable accuracy (97.6% vs. 97.3%), but $DLN_{Ł^*}$ has the advantage in terms of faster inference.

E. CIFAR-10 Dataset

CIFAR-10 [12] is a well established benchmark dataset composed of 60,000 color images, each with a resolution of 32×32 pixels. The images span 10 distinct object categories, including airplanes, automobiles, birds, cats, and more. The dataset is partitioned into 50,000 training samples and 10,000 test samples. Its compact image size makes it particularly suitable for experimentation under limited computational resources, facilitating the development and evaluation of novel algorithms.

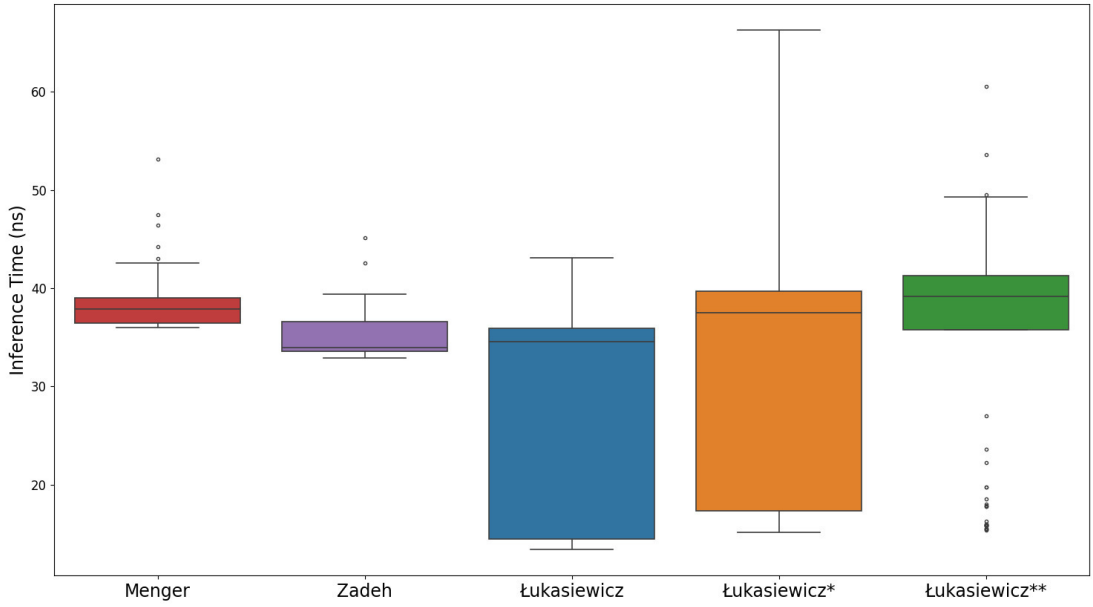

Fig. 3. Basic statistical analysis for Breast Cancer dataset inference time (ns) results (lower is better)

TABLE VIII
RESULTS FOR THE MNIST DATA SET AVERAGED OVER 100 RUNS, WHERE OPS STAND FOR NUMBER OF BINARY OPERATIONS NEEDED TO PERFORM COMPUTATIONS.

MNIST	Acc.	#Param.	Memory	Infer. Time	OPs
ANN (*small*)	97.92%	118282	462KB	$3.66 \pm 0.15\mu$s	236M
DLN_M (*small*)	96.34%	48000	23KB	$1.13 \pm 0.03\mu$s	48K
DLN_Z (*small*)	94.52%	48000	23KB	$1.10 \pm 0.03\mu$s	48K
$\text{DLN}_Ł$ (*small*)	93.93%	48000	23KB	$1.25 \pm 0.30\mu$s	48K
$\text{DLN}_{Ł^*}$ (*small*)	95.55%	48000	23KB	$1.09 \pm 0.03\mu$s	48K
$\text{DLN}_{Ł^{**}}$ (*small*)	96.31%	48000	23KB	$1.05 \pm 0.03\mu$s	48K
ANN	98.40%	22609930	86MB	$1009.17 \pm 9.23\mu$s	45G
DLN_M	97.61%	384000	188KB	$83.16 \pm 0.09\mu$s	384K
DLN_Z	93.33%	384000	188KB	$79.12 \pm 0.06\mu$s	384K
$\text{DLN}_Ł$	95.05%	384000	188KB	$79.63 \pm 0.11\mu$s	384K
$\text{DLN}_{Ł^*}$	97.36%	384000	188KB	$78.58 \pm 0.13\mu$s	384K
$\text{DLN}_{Ł^{**}}$	97.40%	384000	188KB	$78.49 \pm 0.10\mu$s	384K

TABLE IX
MNIST DATA SET SIGNIFICANT INFERENCE TIME DIFFERENCE TESTS. WE DENOTE: $p \in [0.001, 0.5)$ BY *, $p \in [0.0001, 0.001)$ BY **, AND $p \in [0, 0.0001)$

	DLN_M (*small*)	DLN_Z (*small*)	$\text{DLN}_Ł$ (*small*)	$\text{DLN}_{Ł^*}$ (*small*)	$\text{DLN}_{Ł^{**}}$ (*small*)
DLN_M (*small*)			*		
DLN_Z (*small*)			*		
$\text{DLN}_Ł$ (*small*)	*	*		**	***
$\text{DLN}_{Ł^*}$ (*small*)			**		
$\text{DLN}_{Ł^{**}}$ (*small*)			***		
	DLN_M	DLN_Z	$\text{DLN}_Ł$	$\text{DLN}_{Ł^*}$	$\text{DLN}_{Ł^{**}}$
DLN_M		***	***	***	***
DLN_Z	***		***	**	***
$\text{DLN}_Ł$	***	***		***	***
$\text{DLN}_{Ł^*}$	***	**	***		
$\text{DLN}_{Ł^{**}}$	***	***	***		

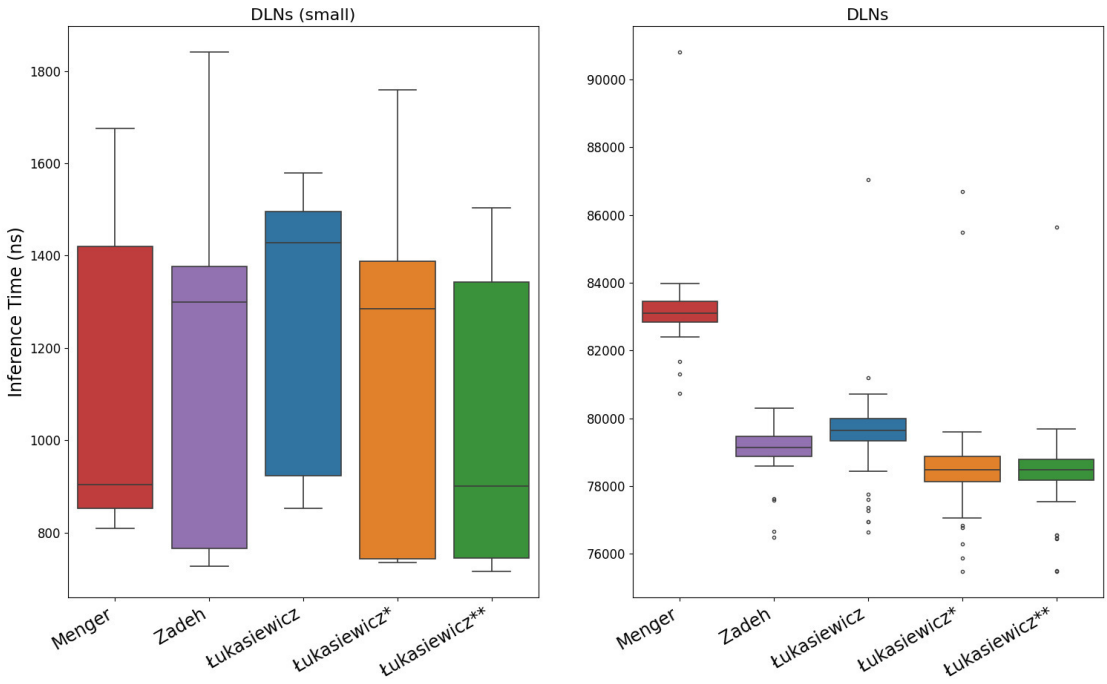

Fig. 4. Basic statistical analysis for MNIST dataset inference time (ns) results (lower is better)

Following the methodology proposed in [19], we reduce the color channel resolution to discrete threshold levels to accommodate binary input constraints of logic based models. Specifically, we apply three value thresholds for the *small* and *medium* models and 31 thresholds for the *large* models. Instead of a simple binarization using a 0.5 cutoff, which often results in significant information loss, we use a set of progressive thresholds (e.g., 0.25, 0.5, 0.75) to discretize pixel intensities more effectively. For instance, a pixel with intensity $p_i = 0.6$ can be represented as $p_i = [1, 1, 1, 0]$, where i denotes the pixel's position in the flattened image. This multi threshold encoding mitigates the discretization loss when converting fuzzy inputs into binary representations, thus improving the suitability of the data for binary logic gate models.

Although dropout and data augmentation are widely used to improve generalization in vision tasks, we intentionally refrain from employing these techniques to maintain consistency with our experimental framework. For all DLN implementations, we use the notation S, M, and L to represent small, medium, and large models, respectively, based on the number of parameters.

In the smaller model configurations, DLN implementations achieve lower accuracy compared to ANNs (see table X). Among the different DLN implementations, DLN_M consistently achieves the highest accuracy across all model sizes (S, M, L). The initialization methods show consistent improvement in accuracy over the base DLN_L model for all model sizes without significant impact on the inference time (see table XI and figure 5).

V. Discussion

The experimental results show consistently low accuracy for Łukasiewicz T-norm based models, which aligns with our initial hypothesis. We attribute this to a substantial portion of the domain in Łukasiewicz based fuzzy logic yielding zero gradients, which prevents error signals from backpropagating effectively to higher layers. Moreover, across all experiments, larger networks benefit significantly from both proposed initialization schemes, showing notable improvements over the baseline DLNs_L. This supports the idea that biasing the model toward feedforward gates while avoiding gates that contribute to gradient vanishing, particularly during the early stages of training when error rates are highest, can substantially enhance error propagation. Interestingly, these initialization methods also increase the models inference speed compared to the baseline models. While increasing the number of feedforward-biased gates does lead to higher accuracy, the improvement is modest and not consistently observed across all experiments. In smaller models, such as those trained on MONK-2, MONK-3, and the Breast Cancer dataset, biasing toward feedforward gates results in a decrease in accuracy performance. We hypothesize that when the number of model parameters is limited, the biasing restricts the model's capacity to flexibly select appropriate logic gates, thereby impairing its learning ability. However, in larger models, the inflexibility introduced by the initialization schemes results in measurable improvements in both accuracy and inference time. One

TABLE X
RESULTS FOR THE CIFAR-10 DATA SET AVERAGED OVER 100 RUNS, WHERE OPS STAND FOR NUMBER OF BINARY OPERATIONS NEEDED TO PERFORM COMPUTATIONS, S,M,L STAND FOR SMALL, MEDIUM AND LARGE NUMBER OF PARAMETERS IN THE MODEL RESPECTIVELY PRESENTED IN THE THIRD COLUMN.

CIFAR-10	Acc.	#Param.	Memory	Infer. Time	OPs
ANN	57.08%	12.6M	48MB	$519.38 \pm 6.70\mu$s	25G
DLN_M (S)	45.75%	48K	24KB	$1.75 \pm 0.00\mu$s	48K
DLN_Z (S)	45.45%	48K	24KB	$2.00 \pm 0.00\mu$s	48K
$\text{DLN}_Ł$ (S)	36.92%	48K	24KB	$1.63 \pm 0.00\mu$s	48K
$\text{DLN}_{Ł^*}$ (S)	41.14%	48K	24KB	$1.66 \pm 0.00\mu$s	48K
$\text{DLN}_{Ł^{**}}$ (S)	42.44%	48K	24KB	$1.88 \pm 0.00\mu$s	48K
DLN_M (M)	57.34%	512K	250KB	$167.19 \pm 0.14\mu$s	512K
DLN_Z (M)	55.55%	512K	250KB	$163.54 \pm 0.11\mu$s	512K
$\text{DLN}_Ł$ (M)	52.23%	512K	250KB	$158.81 \pm 0.25\mu$s	512K
$\text{DLN}_{Ł^*}$ (M)	54.88%	512K	250KB	$159.22 \pm 0.11\mu$s	512K
$\text{DLN}_{Ł^{**}}$ (M)	55.30%	512K	250KB	$159.59 \pm 0.18\mu$s	512K
DLN_M (L)	60.32%	1.28M	625KB	$399.08 \pm 0.34\mu$s	1.28M
DLN_Z (L)	57.28%	1.28M	625KB	$382.81 \pm 0.16\mu$s	1.28M
$\text{DLN}_Ł$ (L)	53.48%	1.28M	625KB	$377.30 \pm 0.21\mu$s	1.28M
$\text{DLN}_{Ł^*}$ (L)	57.78%	1.28M	625KB	$379.28 \pm 0.26\mu$s	1.28M
$\text{DLN}_{Ł^{**}}$ (L)	58.09%	1.28M	625KB	$377.20 \pm 0.23\mu$s	1.28M

TABLE XI
CIFAR-10 DATA SET SIGNIFICANT INFERENCE TIME DIFFERENCE TESTS. WE DENOTE: $p \in [0.001, 0.5)$ BY *, $p \in [0.0001, 0.001)$ BY **, AND $p \in [0, 0.0001)$

	DLN_M (S)	DLN_Z (S)	$\text{DLN}_Ł$ (S)	$\text{DLN}_{Ł^*}$ (S)	$\text{DLN}_{Ł^{**}}$ (S)
DLN_M (S)		***	***	***	***
DLN_Z (S)	***		***	***	***
$\text{DLN}_Ł$ (S)	***	***		***	***
$\text{DLN}_{Ł^*}$ (S)	***	***	***		***
$\text{DLN}_{Ł^{**}}$ (S)	***	***	***	***	
	DLN_M (M)	DLN_Z (M)	$\text{DLN}_Ł$ (M)	$\text{DLN}_{Ł^*}$ (M)	$\text{DLN}_{Ł^{**}}$ (M)
DLN_M (M)		***	***	***	***
DLN_Z (M)	***		***	***	***
$\text{DLN}_Ł$ (M)	***	***			*
$\text{DLN}_{Ł^*}$ (M)	***	***	*		
	DLN_M (L)	DLN_Z (L)	$\text{DLN}_Ł$ (L)	$\text{DLN}_{Ł^*}$ (L)	$\text{DLN}_{Ł^{**}}$ (L)
DLN_M (L)		***	***	***	***
DLN_Z (L)	***		***	***	***
$\text{DLN}_Ł$ (L)	***	***		***	
$\text{DLN}_{Ł^*}$ (L)	***	***	***		***
$\text{DLN}_{Ł^{**}}$ (L)	***	***		***	

possible explanation for the improved runtime efficiency is that a majority of the operations in these models are identity functions. At lower compiler optimization levels (e.g., `-O0`), identity operations are implemented as simple memory transfers between registers. However, under aggressive optimization settings (e.g., `-O3`), compilers often eliminate such operations entirely by reusing registers, resulting in minimal computational overhead. This behavior significantly reduces the runtime burden of inference in models where identity gates dominate. Figure 6 illustrates the gate selection distributions for $\text{DLNs}_Ł$ (Baseline), $\text{DLNs}_{Ł^*}$ (A), and $\text{DLNs}_{Ł^{**}}$ (A - B) on the MNIST dataset. A clear concentration of feedforward gates a for $\text{DLNs}_{Ł^*}$, and feedforward gates a and b for $\text{DLNs}_{Ł^{**}}$, which correspond to identity mappings, are evident in the latter two configurations. This suggests that the initialization schemes effectively steer the models toward simpler computational pathways. Interestingly, these identity heavy models not only benefit from faster inference but also demonstrate improved gradient stability, particularly during the early stages of training. The stable signal propagation enabled by identity operations likely contributes to the observed accuracy gains. However, the dominance of feedforward gates in $\text{DLNs}_{Ł^*}$ and $\text{DLNs}_{Ł^{**}}$ after training also implies underutilization of the network's expressive capacity. While this may be beneficial for efficiency, it raises questions about the models' ability to learn more complex representations when required.

Furthermore, we observe that increasing model size generally leads to an improvement in accuracy. This trend is consistent with the two findings from [19], [28] show that larger DLN models possess greater representational capacity. In current DLNs implementations, input pairs are connected randomly to computational units. This random connectivity hinders memory coalescing during data loading, which in turn leads to data starvation as computational kernels are forced to wait for data to be fetched. It is due to transferring memory from global memory (VRAM) to local memory (Registers) cost hundreds times more compute cycles comparing to the actually computations itself [17]. Replacing a convex combination of gates with a single, discrete gate can lead to significant performance degradation. For instance, if a computational unit

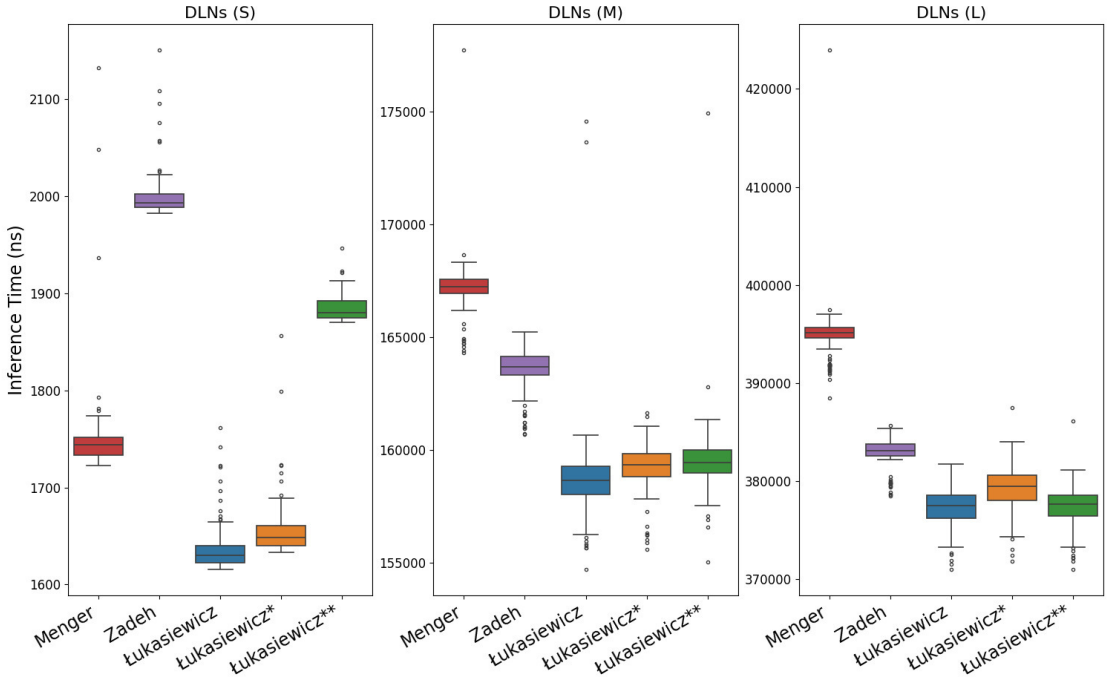

Fig. 5. Basic statistical analysis for CIFAR-10 dataset inference time (ns) results (lower is better)

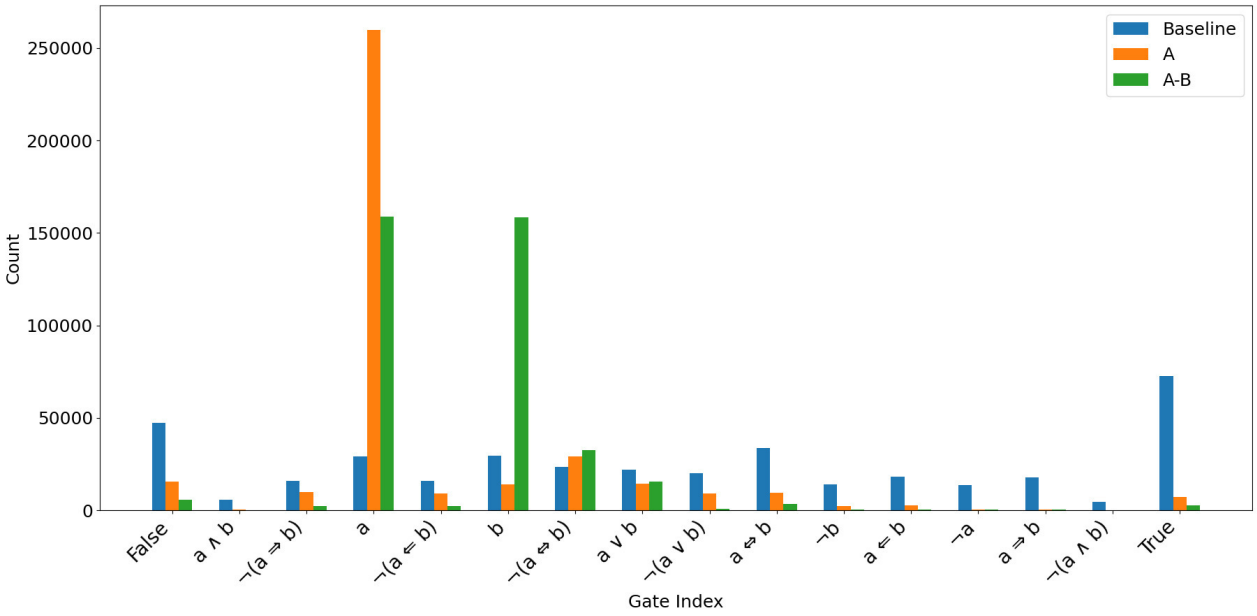

Fig. 6. DLNs$_{Ł}$, DLNs$_{Ł^*}$, DLNs$_{Ł^{**}}$ gate distributions in MNIST dataset

has probabilities $p_1 = 0.60$ and $p_4 = 0.40$ for two logic gates, deterministically selecting a gate at p_1 fails to capture the nuanced behavior expressed in the soft combination. This simplification may overlook critical interactions among the logic operators.

VI. Conclusions

We proposed a differentiable logic gate network based on a relaxation of classical logic gates using the Łukasiewicz T-norm. Our empirical findings support the hypothesis that the Łukasiewicz T-norm is not well-suited for gradient based training due to its gradient vanishing properties. This is consistent with observations reported in [10]. The experiments further demonstrate that encouraging better error propagation by biasing the weights of the computational units toward feedforward logic gates and avoiding the gates that can cause zeros in gradient significantly improves model accuracy and mitigates the gradient vanishing issue.

DLNs offer a distinct advantage over traditional neural networks in terms of interpretability. Due to the structural clarity, this facilitates direct inspection and tracing of information flow, making DLNs particularly suitable for applications in explainable artificial intelligence (XAI). Unlike standard neural networks, where interpretability often relies on post hoc analysis or surrogate explanations, DLNs provide intrinsic transparency through their architecture. This transparency reveals a recurrent pattern in trained DLN models: a disproportionately high number of units are configured as feedforward (identity) gates. While this contributes to computational efficiency and stable training dynamics, it also suggests a significant underutilization of the available computational capacity within the network. Many logic gate units effectively act as passthrough mechanisms, bypassing transformation or decision making roles.

Regardless of implementation details, differentiable logic gate networks generally require less memory and exhibit lower inference time compared to standard neural networks, highlighting their computational efficiency despite variable performance in accuracy. These findings underscore key differences across implementations of DLNs. While logic networks trained using Menger's T-norm may achieve higher accuracy, the Łukasiewicz based models exhibit faster inference times. This emphasizes the importance of careful design and evaluation when developing differentiable logic gate networks. In some applications, reducing inference time with a marginal decrease in accuracy may be preferable to higher accuracy but significantly slower performance.

Further performance gains in DLNs may be realized by optimizing the flow of information through structured and efficient connectivity patterns. One notable limitation in existing architectures is the underutilization of computational units, which can be mitigated by reusing logic elements across input features in a convolution like fashion [20] By enabling each compute unit to process multiple bit pairs in parallel rather than in a one-by-one manner, the network can exploit the capabilities of modern 64-bit arithmetic logic units (ALUs) on CPUs. This approach not only enhances the utilization of computational units in DLNs but also contributes to reducing computational overhead during inference. Introducing regularization strategies to constrain the weight vectors of computational units may help reduce discretization loss when translating from fuzzy to hard logic (Boolean). These offer promising directions for improving both the accuracy and efficiency of DLNs in practical applications.

Finally, while most medical machine learning studies focus on developing high-accuracy and computationally intensive models, DLNs offer a practical potential in low-latency, resource-constrained use cases. Applications such as oxygen blood saturation measuring or non-invasive glucose monitoring suggest alternative research directions for logic gate networks.

Acknowledgment

This paper presents results which will be presented as a part of master thesis of the first author prepare under the supervision of the second author. Authors would like to thank anonymous Reviewers for valuable comments and remarks which enabled to improve the paper.

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. Ślęzak, Eds., IEEE. Available: http://dx.doi.org/10.15439/2023F7745

[3] J. Choi, Z. Wang, S. Venkataramani, P. Chuang, V. Srinivasan, and K. Gopalakrishnan, "PACT: Parameterized Clipping Activation for Quantized Neural Networks," 2018. Available: https://arxiv.org/abs/1805.06085

[4] 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. Ślęzak, Eds., IEEE, 2021, pp. 271–279. Available: http://dx.doi.org/10.15439/2021F86

[5] L. Dey, S. Jana, T. Dasgupta, and T. Gupta, "Deciphering Clinical Narratives – Augmented Intelligence for Decision Making in Healthcare Sector," in *Proc. 18th Conf. on Computer Science and Intelligence Systems*, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., vol. 35, *Annals of Computer Science and Information Systems*, IEEE, 2023, pp. 11–24. Available: http://dx.doi.org/10.15439/2023F3385

[6] D. Długosz, A. Królak, T. Eftestøl, S. Ørn, T. Wiktorski, K. R. J. Oskal, and M. Nygård, "ECG Signal Analysis for Troponin Level Assessment and Coronary Artery Disease Detection: the NEEDED Study 2014," in *Proc. 2018 Federated Conf. Comput. Sci. Inf. Syst.*, vol. 15, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds. IEEE, 2018, pp. 1065–1068. Available: http://dx.doi.org/10.15439/2018F247

[7] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," *in Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*, 2015, pp. 1737–1746.

[8] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste, "Sparsity in deep learning: pruning and growth for efficient inference and training in neural networks," *J. Mach. Learn. Res.*, vol. 22, no. 1, art. no. 241, Jan. 2021, 124 pp.

[9] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014. Available: https://arxiv.org/abs/1412.6980

[10] 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

[11] J. Kosiński, K. Szklanny, A. Wieczorkowska, and M. Wichrowski, “An Analysis of Game-Related Emotions Using EMOTIV EPOC,” *Proc. 2018 Federated Conf. on Computer Science and Information Systems (FedCSIS)*, vol. 15, Annals of Computer Science and Information Systems, pp. 913–917, 2018, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds. IEEE. Available: http://dx.doi.org/10.15439/2018F296

[12] A. Krizhevsky, “Learning Multiple Layers of Features from Tiny Images," Univ. of Toronto, 2012.

[13] Y. LeCun, C. Cortes, and C. J. C. Burges, “The MNIST database of handwritten digits," 1998. Available: http://yann.lecun.com/exdb/mnist/

[14] M. Marcinkiewicz and G. Mrukwa, “Quantitative Impact of Label Noise on the Quality of Segmentation of Brain Tumors on MRI scans,” *Proc. 2019 Federated Conf. on Computer Science and Information Systems (FedCSIS)*, vol. 18, pp. 61–65, 2019. Edited by M. Ganzha, L. Maciaszek, and M. Paprzycki. IEEE. Available: http://dx.doi.org/10.15439/2019F273

[15] 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

[16] D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, and A. Liotta, “Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science," *Nature Communications*, vol. 9, no. 1, Jun. 2018, art. 2383. Available: https://doi.org/10.1038/s41467-018-04316-3

[17] A. Morar, F. Moldoveanu, A. Moldoveanu, O. Balan, and V. Asavei, “GPU Accelerated 2D and 3D Image Processing,” in *Proc. 2017 Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, vol. 11, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds., IEEE, 2017, pp. 653–656. Available: http://dx.doi.org/10.15439/2017F265

[18] A. Paszke et al., “PyTorch: An Imperative Style, High-Performance Deep Learning Library," 2019. Available: https://arxiv.org/abs/1912.01703

[19] 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

[20] 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/db988b089d8d97d0f159c15ed0be6a71-Paper-Conference.pdf

[21] M. Pudo, M. Wosik, and A. Janicki, “Open Vocabulary Keyword Spotting with Small-Footprint ASR-based Architecture and Language Models,” *Proc. 18th Conf. Computer Science and Intelligence Systems (FedCSIS)*, Annals of Computer Science and Information Systems, vol. 35, pp. 657–666, 2023, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., IEEE. Available: http://dx.doi.org/10.15439/2023F8594

[22] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, and N. Sebe, “Binary neural networks: A survey," *Pattern Recognition*, vol. 105, 2020, art. 107281. Available: https://www.sciencedirect.com/science/article/pii/S0031320320300856

[23] M. Rapoport and T. Tamir, “Best Response Dynamics for VLSI Physical Design Placement,” in *Proc. 2019 Federated Conf. on Computer Science and Information Systems (FedCSIS)*, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds., IEEE, vol. 18, 2019, pp. 147–156. Available: http://dx.doi.org/10.15439/2019F91

[24] 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

[25] A. Telikani, A. Tahmassebi, W. Banzhaf, and A. H. Gandomi, “Evolutionary Machine Learning: A Survey," *ACM Comput. Surv.*, vol. 54, no. 8, art. 161, Oct. 2021. Available: https://doi.org/10.1145/3467477

[26] S. Thrun et al.,“The Monk’s Problem’s: A Performance Comparison of Different Learning Methods," 1991. Available: https://api.semanticscholar.org/CorpusID:59810521

[27] L. Uberg and S. Kadry, “Analysis of Brain Tumor Using MRI Images,” in *Proc. 17th Conf. Computer Science and Intelligence Systems (FedCSIS)*, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., vol. 30, *Annals of Computer Science and Information Systems*, IEEE, 2022, pp. 201–204. Available: http://dx.doi.org/10.15439/2022F69

[28] P. Wasilewski, and Ch. D. Nguy, “Deep Differentiable Logic Gate Networks Based on Fuzzy Zadeh’s T-norm,” Proceedings of 6th Polish Conference on Artificial Intelligence (PP-RAI 2025), Katowice, Poland, Apr. 7-9, 2025. to appear in Lecture Notes in Networks and Systems, Springer.

[29] 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

[30] M. Zwitter and M. Soklic, “Breast Cancer," UCI Machine Learning Repository, 1988. Available: https://doi.org/10.24432/C51P4M