TA-DMP-SNN: A Topology-Aware Dual-Memory Pareto Spiking Neural Search Network for Multiobjective Reed–Muller Logic Synthesis

Abstract

The ternary polarity space of Reed–Muller (RM) logic grows exponentially with the number of variables, while area, power, delay, and reliability objectives often conflict. We propose TA-DMP-SNN, a Topology-Aware Dual-Memory Pareto Spiking Neural Network for multiobjective RM logic synthesis. Unlike approaches that use neural networks only as quality predictors, TA-DMP-SNN directly employs an SNN as a discrete search mechanism over the ternary polarity space. Circuit topology, RM-term co-occurrence, and structural relations are encoded through topology-aware synaptic gating. Reference-vector routing, fast eligibility traces, and slow nondominated-archive memory jointly regulate exploration and consolidation. Across eight nontrivial RM benchmark circuits, TA-DMP-SNN achieved the highest median four-objective hypervolume after 100 unique evaluations, with a median absolute gain of 0.143 and a median improvement of 56.0% over the strongest baseline.

1 Introduction

Reed–Muller (RM) representations expose structures dominated by XOR operations and can benefit arithmetic, parity, testing, and emerging logic technologies. In mixed polarity Reed–Muller (MPRM) logic, each variable may have positive, negative, or mixed polarity. An n input function therefore has 3^n possible polarity assignments. Equivalent assignments can differ substantially in size, depth, switching activity, and fault propagation. Polarity selection must therefore find useful tradeoffs among conflicting objectives under a limited evaluation budget.

RM optimization has progressed from enumeration and single objective heuristics to evolutionary multiobjective search. Existing methods use genetic search, annealing, nondominated archives, hypervolume guidance, and operators designed for ternary polarity codes [9, 10, 12, 13]. These methods demonstrate the value of explicit Pareto search. However, their candidate generators mainly manipulate entries in a ternary chromosome. Circuit topology, RM term relations, and timing information are used primarily during evaluation. The nondominated archive stores good solutions but rarely influences how new candidates are generated.

Spiking neural networks (SNNs) provide another approach through sparse events, local interactions, and multiple temporal scales. Slack-Free SNN directly encodes combinatorial constraints and avoids auxiliary slack variables [7]. SNN-RET generates candidates for optimization with multiple objectives by combining membrane adaptation, selection using reference vectors, and diversity recovery [14]. However, these methods do not jointly address ternary logic variables, circuit dependent interactions, diverse Pareto solutions, and persistent learning from archive patterns. Useful polarity assignments are not formed from independent variable choices. Variables may develop coupled preferences when they share Boolean cones, occur together in RM terms, or affect the same critical paths. Recent improvements can guide local exploration, while patterns repeated among nondominated solutions can support longer term search. This motivates combining fast transition credit with slow consolidation of recurring Pareto patterns in a circuit informed spiking model.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We propose TA-DMP-SNN, a Topology Aware Dual Memory Pareto Spiking Neural Network for multiobjective MPRM polarity search. Each logic variable is represented by three competing spiking states that correspond to positive, negative, and mixed polarity. Relations derived from the Boolean network, RM terms, and critical paths control synaptic communication without imposing hard constraints. A fast eligibility trace learns from recent Pareto feedback. A slow synaptic trace consolidates polarity interactions that recur among nondominated solutions. Routing based on reference vectors maintains coverage of the objective space, while entropy control reduces excessive structural bias. This closed loop allows circuit structure and archive history to guide future candidate generation.

TA-DMP-SNN is not a direct replacement for an evolutionary operator. It integrates circuit topology, recent transition credit, and persistent Pareto memory within a single candidate generator. The four main contributions are listed below.

1. **A circuit-aligned ternary spiking representation.** We formulate MPRM polarity search as competition among three spiking states per variable, preserving the discrete semantics of the decision space without continuous relaxation or binary expansion.
2. **Reward-correctable topology-aware search.** We translate complementary circuit relations into soft synaptic gates, allowing structural coupling to guide early exploration while remaining revisable when Pareto feedback contradicts the prior.
3. **Archive-to-synapse dual-memory plasticity.** We separate immediate eligibility-based adaptation from slow consolidation of polarity patterns that recur in the nondominated archive. This turns the archive from a passive elite set into persistent search memory and is the main methodological distinction from generic SNN-based multiobjective optimization.
4. **Diversity control for circuit-conditioned many-objecti ve search.** We combine reference-vector routing with entropy-based homeostasis to allocate search effort across trade-off regions and to counter premature ternary-state collapse.

2 Related Work

2.1 Multiobjective Polarity Optimization for Reed–Muller Logic

The implementation cost of a Reed–Muller circuit depends strongly on its polarity assignment. An n-variable mixed-polarity Reed–Muller (MPRM) expression has 3^n possible polarity vectors, making exhaustive enumeration impractical. Yang and Lai combined genetic search, population mixing inspired by annealing, and parallel tabular conversion to minimize the number of product terms [10]. Their method considered only one structural objective and did not evaluate mapped power, timing, or reliability.

Later studies formulated polarity selection as a multiobjective problem. APOA optimized area and power using an external nondominated archive [12]. TSAMEA combined two-stage evolution, HV-guided adaptation, and historical memory to optimize area and power [9]. AROM used tree-encoded ternary polarities and adaptive operators to optimize area and soft error rate [13]. Although these methods improve Pareto search, they still generate candidates through population operators over polarity codes. Circuit structure is used mainly for objective evaluation.

TA-DMP-SNN directly represents polarity variables as ternary spiking states. Circuit-derived relations control the interactions among these states. Stable patterns in the nondominated archive are consolidated into a slow synaptic trace that guides subsequent search. TA-DMP-SNN therefore allows circuit structure and archive history to influence the generation of new polarity vectors.

2.2 Search and Learning Guided Logic Synthesis

Recent synthesis methods based on learning mainly select optimization actions or recipes. EasySO formulates logic optimization, technology mapping, and subsequent operations as

a reinforcement learning environment for exploration under limited sampling budgets [11]. AlphaSyn uses Monte Carlo tree search with learned guidance, asynchronous exploration, and resource allocation to optimize synthesis sequences [8]. Both methods learn policies over synthesis commands rather than Reed–Muller polarity variables.

Other methods focus on circuit representation. E-Syn preserves equivalent Boolean structures through equality saturation and selects implementations using costs that account for technology [1]. E-morphic improves scalability through direct conversion between e-graphs and circuits, solution space pruning, and simulated annealing extraction [2]. PolarGate incorporates gate functions into message passing through differentiable logical operators [6]. These studies motivate preserving structural alternatives and incorporating Boolean semantics into the learning process.

TA-DMP-SNN uses circuit topology more directly during polarity search. It extracts adjacency in the Boolean network, Reed–Muller term co-occurrence, and coupling along critical paths. These relations control synaptic communication and influence coordinated polarity changes. Feedback is obtained online from evaluated candidates and the evolving nondominated archive, so no large supervised corpus is required.

2.3 Spiking Neural Networks for Combinatorial and Multiobjective Optimization

SNN optimizers use spike timing and neuronal competition to perform stochastic local search. Fang and Lele distributed quadratic unconstrained binary optimization across collaborative SNNs that exchanged their best solutions [3]. This general approach requires a binary quadratic formulation. Slack-Free SNN instead uses neurons to detect and repair constraint violations without auxiliary slack variables [7]. This supports direct neural encoding of native combinatorial variables.

SNN-RET applies spike based candidate generation to continuous optimization with multiple objectives. It combines spiking search, selection guided by reference vectors, and a turbulence operator [14]. However, it does not address ternary logic variables, connectivity determined by circuit structure, or persistent learning from archive patterns.

TA-DMP-SNN combines direct ternary encoding, Pareto control based on reference vectors, and relational information derived from circuits. Fast eligibility traces reinforce recent transitions, while slow traces consolidate recurring patterns among nondominated polarity vectors. The archive therefore becomes an active memory that guides future candidates. The experiments examine the effects of topology gating and slow consolidation on sample efficiency, archive stagnation, convergence, and Pareto coverage.

3 Problem Formulation

3.1 MPRM Polarity Space

Let

$$\mathbf{F} = (f_1, \ldots, f_m) : \{0,1\}^n \to \{0,1\}^m \tag{1}$$

denote an n-input, m-output Boolean function. We consider exact mixed-polarity Reed–Muller (MPRM) realizations of $\mathbf{F}$. A polarity assignment is represented by a vector

$$\mathbf{p} = (p_1, \ldots, p_n) \in \Omega_n, \qquad \Omega_n = \{0,1,2\}^n, \tag{2}$$

where $p_i = 0$, $p_i = 1$, and $p_i = 2$ denote the positive, negative, and mixed polarity of variable x_i, respectively. The labels are a convention; the optimization depends on the three admissible states rather than their numerical order. The number of candidate assignments is

$$|\Omega_n| = 3^n. \tag{3}$$

For a given assignment $\mathbf{p}$, let

$$\mathcal{R}_{\mathbf{p}} = \mathcal{T}_{\mathrm{RM}}(\mathbf{F}, \mathbf{p}) \tag{4}$$

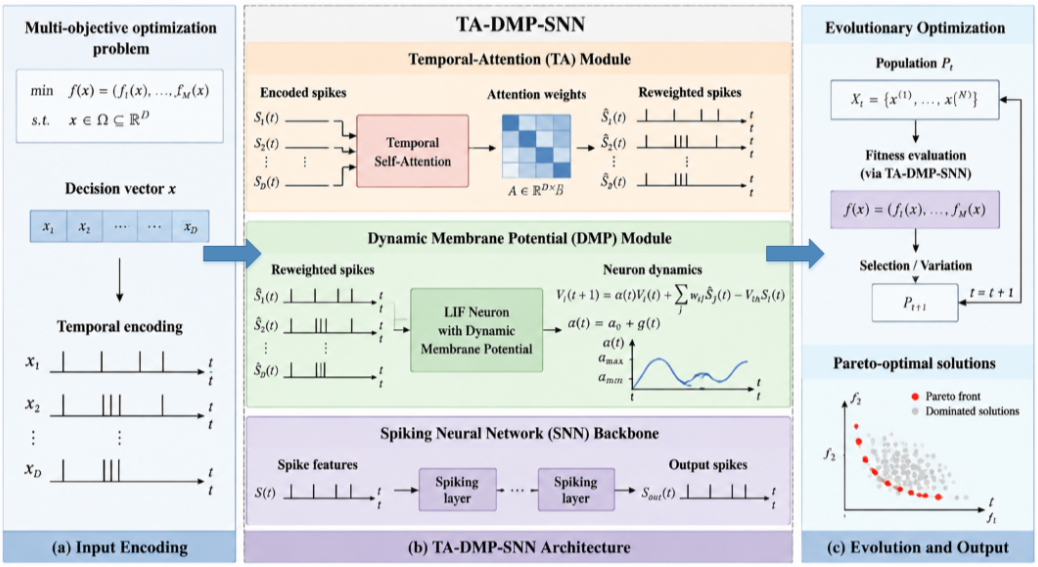

Figure 1: Overall architecture of TA-DMP-SNN. (a) The decision vector is encoded into temporal spike trains. (b) TA reweights spikes, DMP adjusts neuronal dynamics, and the SNN extracts spatio-temporal features to predict objective values. (c) The model is integrated into multi-objective evolutionary optimization to obtain Pareto-optimal solutions.

be the MPRM expression obtained by the exact Reed–Muller transform under $\mathbf{p}$, and let $\mathcal{C}_{\mathbf{p}}$ be its realization in the target RM gate basis. Polarity changes the set of nonzero coefficients, the support of the product terms, and the amount of logic that can be shared among outputs. It does not change the specified Boolean function. Hence, every admissible candidate must satisfy

$$\mathcal{C}_{\mathbf{p}}(\mathbf{x}) = \mathbf{F}(\mathbf{x}), \qquad \forall \mathbf{x} \in \{0,1\}^n. \tag{5}$$

Functional equivalence is therefore a hard constraint, not an optimization objective. Approximate implementations are outside the scope of this work.

Detailed modeling of each objective function is provided in Appendix A.

4 TA-DMP-SNN

4.1 Overview

TA-DMP-SNN is a circuit-conditioned candidate generator for the budgeted Pareto problem detailed in Appendix A. It does not predict the quality of a polarity assignment. Instead, the network produces a ternary vector $\mathbf{p} \in \Omega_n$; an exact RM transform constructs the corresponding logic realization; and the external evaluator returns the four-objective vector $\mathbf{f}(\mathbf{p})$. The resulting dominance and diversity signals are then fed back to the spiking network. This separation keeps functional correctness and objective computation outside the neural dynamics while allowing search experience to change the distribution from which later candidates are generated.

Figure 1 shows the closed loop. Before search, three function-derived relations are extracted: Boolean-derivative similarity, RM-term co-occurrence, and degree-weighted RM co-occurrence. These relations softly gate communication between ternary neuron groups. During search, reference vectors divide the objective space into directional regions. A fast eligibility pathway assigns credit to recent spike interactions, whereas a slow pathway consolidates and replays polarity associations that persist in the nondominated archive. Entropy feedback acts locally on neuron groups that have collapsed to one state without producing objective-space progress.

4.2 Ternary Competitive Spiking Representation

Direct problem encoding avoids the auxiliary binary variables introduced by a generic reduction. This principle has proved useful in recent problem-specific SNN optimizers [7]; it is particularly natural for MPRM polarity search because every decision already has three admissible states. For variable x_i, we instantiate a competitive group $\mathcal{N}_i = \{N_{i0}, N_{i1}, N_{i2}\}$. The three neurons represent positive, negative, and mixed polarity, following Eq. (2). The network therefore contains $3n$ decision neurons rather than a binary encoding with additional consistency constraints. This representation also agrees with the native ternary organization used by recent MPRM search methods [13].

Within a decision window of T simulation steps, the membrane potential and spike of neuron N_{ik} evolve as

$$u_{ik}^{\tau+1} = \lambda u_{ik}^{\tau}(1 - s_{ik}^{\tau}) + I_{ik}^{\tau} - \gamma \sum_{h \neq k} s_{ih}^{\tau} + \epsilon_{ik}^{\tau}, \tag{6}$$

$$s_{ik}^{\tau+1} = \mathbb{I}\left[u_{ik}^{\tau+1} \geq \vartheta_{ik}\right], \tag{7}$$

where $\lambda \in [0, 1)$ is the leakage factor, γ controls within-group inhibition, ϑ_{ik} is the firing threshold, and ϵ_{ik}^{τ} is zero-mean exploration noise. The reset is implemented by the factor $1 - s_{ik}^{\tau}$. This discrete-time form retains the state and event separation of standard spiking-neuron models [5] without assigning biological meaning to the optimization dynamics.

The firing rate of each state is

$$r_{ik} = \frac{1}{T} \sum_{\tau=1}^{T} s_{ik}^{\tau}, \tag{8}$$

The firing rates and final membrane potentials define a score $z_{ik} = r_{ik} + 0.08 u_{ik}^{T} + \epsilon_{ik}^{\mathrm{dec}}$. The candidate is sampled groupwise rather than decoded by a deterministic maximum,

$$\Pr(p_i = k) = \frac{\exp(z_{ik}/\tau_i)}{\sum_{h=0}^{2} \exp(z_{ih}/\tau_i)}. \tag{9}$$

Here, τ_i is controlled by the group-specific exploration-noise scale. The nonwinning neurons are not deleted: their membrane states and eligibility traces remain available in the next window. If a sampled vector has already been evaluated, one to three coordinates are perturbed; exhaustive selection of the next unseen vector is used only as a final fallback. The evaluation budget therefore counts unique candidates.

4.3 Topology-Aware Synaptic Gating

A fully connected interaction model treats every pair of polarity variables as equally relevant. That assumption is poorly matched to logic functions, where shared influence and high-order RM terms create nonuniform dependencies. Function-aware graph models such as PolarGate likewise show that Boolean structure should enter the message-passing rule rather than appear only as an input label [6]. We use function information as a soft gate, not as a fixed decision rule.

Let $C^{\mathrm{der}}, C^{\mathrm{rm}}, C^{\mathrm{deg}} \in [0, 1]^{n \times n}$ denote normalized variable-relation matrices. C_{ij}^{der} is the Jaccard similarity between the support vectors of the Boolean derivatives with respect to x_i and x_j. C_{ij}^{rm} counts their co-occurrence in active terms of the positive-polarity RM expansion, and C_{ij}^{deg} weights the same co-occurrences by term degree. The implementation uses their equal-weight mean,

$$C_{ij} = \frac{1}{3}\left(C_{ij}^{\mathrm{der}} + C_{ij}^{\mathrm{rm}} + C_{ij}^{\mathrm{deg}}\right), \tag{10}$$

and the corresponding soft gate is

$$g_{ij} = \sigma(b_g + \kappa C_{ij}). \tag{11}$$

Here, the experiments set $b_g = -2$ and $\kappa = 4$. The synaptic input in Eq. (6) becomes

$$I_{ik}^{\tau} = b_{ik} + \sum_{j \neq i} \sum_{h=0}^{2} g_{ij} \left(w_{ij,kh}^{0} + w_{ij,kh}^{\mathrm{f}} + w_{ij,kh}^{\mathrm{s}} \right) s_{jh}^{\tau}, \tag{12}$$

where w^0 is the base interaction, w^{f} is fast search memory, and w^{s} is slow archive memory. Because g_{ij} is continuous, weakly related variables remain reachable and an inaccurate topology prior cannot make a polarity assignment infeasible.

4.4 Reference-Region Feedback

The four objectives can reward different portions of the Pareto front. A single global reward tends to favor regions that are already dense or easy to reach. We therefore maintain a set of nonnegative reference vectors $\Lambda = \{\boldsymbol{\lambda}_q\}_{q=1}^{Q}$ with $\|\boldsymbol{\lambda}_q\|_1 = 1$. Reference guidance has recently been combined with spiking candidate generation for generic multi- and many-objective optimization [14]; here it is adapted to a discrete, circuit-conditioned search space.

After evaluating candidate $\mathbf{p}_t$, its objective vector is normalized using the current ideal and nadir estimates. The candidate is assigned to

$$q_t = \arg\min_{q} \angle\Big(\widehat{\mathbf{f}}(\mathbf{p}_t), \boldsymbol{\lambda}_q \Big). \tag{13}$$

Let n_{q_t} be the number of previously sampled candidates assigned to the selected region. Using the archive and candidate to obtain temporary ideal and nadir points, we calculate the nonnegative change in four-dimensional hypervolume, $\Delta\mathrm{HV}_t$. The implemented scalar feedback is

$$R_t = \begin{cases} 1, & \mathbf{p}_t \in \mathcal{A}_t, \\ -0.2, & \text{otherwise} \end{cases} + 2\max(0, \Delta\mathrm{HV}_t) + \frac{0.2}{1 + n_{q_t}}. \tag{14}$$

The last term favors sparsely visited directions. Duplicate vectors are handled before objective evaluation and therefore do not need a reward penalty. The feedback does not replace Pareto dominance; it supplies a bounded credit signal for neural adaptation.

4.5 Dual-Memory Pareto Plasticity

The central design choice is to separate recent transition credit from persistent archive structure. A fast eligibility trace records which spike interactions preceded the current candidate,

$$e_{ij,kh}^{t} = \rho_{\mathrm{f}} e_{ij,kh}^{t-1} + \Phi(s_{ik}, s_{jh}), \tag{15}$$

where Φ is a bounded pre–post correlation accumulated over the decision window. The fast synapse is updated by

$$w_{ij,kh}^{\mathrm{f}} \leftarrow (1 - \delta_{\mathrm{f}}) w_{ij,kh}^{\mathrm{f}} + \eta_{\mathrm{f}} R_t e_{ij,kh}^{t}. \tag{16}$$

This is a task-level three-factor update: pre–post activity creates eligibility, and Pareto feedback determines its sign and magnitude [4]. Its short time constant makes the search responsive, but it can also reinforce an accidental improvement.

The slow pathway is updated from the archive rather than from one reward. For an archive $\mathcal{A}_t$, define the state association

$$c_{ij,kh}^{t} = \frac{1}{|\mathcal{A}_t|} \sum_{\mathbf{p} \in \mathcal{A}_t} \big[\mathbb{I}(p_i = k, p_j = h) - \pi_{i,k}^{\mathcal{A}} \pi_{j,h}^{\mathcal{A}} \big], \tag{17}$$

where $\pi_{i,k}^{\mathcal{A}}$ is the archive marginal frequency of state k at variable i. Subtracting the product of the marginals prevents a state that is merely common from being mistaken for a useful pairwise dependency. Consolidation is an exponential moving average,

$$m_{ij,kh}^{t} = \rho_{\mathrm{s}} m_{ij,kh}^{t-1} + (1 - \rho_{\mathrm{s}}) c_{ij,kh}^{t}, \tag{18}$$

$$w_{ij,kh}^{\mathrm{s}} = \eta_{\mathrm{s}} m_{ij,kh}^{t}, \qquad \rho_{\mathrm{s}} > \rho_{\mathrm{f}}. \tag{19}$$

The implementation sets $K = 20$ and also maintains a slow categorical bias from the centered log archive marginals. Before a new decision window, an archive member is sampled with probability 0.7. Its one-hot polarity states receive a membrane increment and a decoding-score bonus. Thus, slow memory affects both pairwise interactions and direct archive replay. Historical information has also been used to adapt evolutionary RM search [9]; the difference here is operational: archive statistics are written into neural state and alter the next candidate-generation distribution. The archive is consequently both an output set and a source of long-term search memory.

Further details on entropy homeostasis, archive updates, and computational complexity are provided in Appendix B.

5 Experiments

The experiments evaluate performance under limited evaluations, behavior near convergence, and the contribution of each component. We report hypervolume during search, final Pareto front quality, wall clock time, and ablation results. All objectives are technology independent structural proxies rather than mapped PPA or gate level fault measurements.

Hypervolume and statistics. The primary metric was four objective hypervolume HV_4 over area, power, delay, and fault exposure. Area and power hypervolume HV_{AP} was used as a secondary metric. For each circuit, normalization used final nondominated sets pooled across all methods and 30 runs. Let m_k and M_k denote the pooled extrema of objective k. We computed

$$L_k = m_k - 0.05(M_k - m_k), \qquad U_k = M_k + 0.05(M_k - m_k), \qquad \hat{f}_k = \frac{f_k - L_k}{U_k - L_k}. \tag{20}$$

The bounds were fixed across all methods and checkpoints. Reference points were $(1, 1, 1, 1)$ for HV_4 and $(1, 1)$ for HV_{AP}. Zero range objectives were normalized to zero. We report medians and interquartile ranges. Paired Wilcoxon signed rank tests with two sided alternatives compared TA-DMP-SNN with each baseline. Holm correction covered 18 comparisons for each metric and checkpoint.

5.1 TA-DMP-SNN improves low-budget search efficiency

TA-DMP-SNN shows its clearest advantage in the early search stage. At 100 unique evaluations, it achieves the highest median HV_4 on eight of the nine circuits and ties all methods on rd53 (Table 1). The largest gains occur on 5xp1 and rd84, where TA-DMP-SNN reaches 0.830 on both circuits, compared with 0.196 and below 0.001, respectively, for the stronger of AlphaSyn and SNN-RET. These gains are significant on all eight nontrivial circuits after Holm correction ($p_{\mathrm{Holm}} \leq 4.96 \times 10^{-3}$), while rd53 yields ties because its small search space is rapidly exhausted. The anytime curves in Fig. 2 further show that the performance gap narrows with increasing budget, indicating that TA-DMP-SNN primarily improves early-stage sample efficiency rather than universally superior final solution quality. Complete results on additional, larger-scale circuits are available in the anonymous open-source repository, owing to space limitations, the remaining experiments focus on a representative subset of circuits.

5.2 Slow archive memory accounts for the observed gain

We ablated topology-aware coupling, fast eligibility-based memory, slow archive memory, and entropy-based homeostasis on 5xp1, rd84, and cavlc. Each variant was run for 100 evaluations with the same 30 paired seeds. Removing slow memory, which also disables archive replay and the archive-derived marginal and association updates, consistently reduced HV_4 (Table 2). The median paired loss relative to the full method was 0.646 on 5xp1, 0.830 on rd84, and 0.421 on cavlc, with Holm-corrected p-values of 8.16×10^{-4}, 4.13×10^{-5}, and 1.12×10^{-7}, respectively.

Table 1: Four-objective hypervolume after 100 unique evaluations. Values are median [IQR] over 30 paired runs; larger is better. Bold entries denote the highest median for each circuit.

Circuit	TSAMEA[9]	AlphaSyn[8]	SNN-RET[14]	TA-DMP-SNN
rd53	**1.000 [1.000, 1.000]**	**1.000 [1.000, 1.000]**	**1.000 [1.000, 1.000]**	**1.000 [1.000, 1.000]**
5xp1	0.070 [0.011, 0.179]	0.155 [0.045, 0.253]	0.196 [0.105, 0.329]	**0.830 [0.830, 0.830]**
rd84	< 0.001 [< 0.001, < 0.001]	< 0.001 [< 0.001, < 0.001]	< 0.001 [< 0.001, < 0.001]	**0.830 [0.830, 0.830]**
ctrl	0.068 [0.002, 0.290]	0.064 [0.006, 0.291]	0.312 [0.077, 0.814]	**0.830 [0.830, 0.830]**
dec	0.056 [0.056, 0.085]	0.157 [0.066, 0.194]	0.206 [0.168, 0.206]	**0.227 [0.206, 0.227]**
cavlc	0.059 [0.014, 0.202]	0.141 [0.024, 0.231]	0.277 [0.135, 0.451]	**0.471 [0.339, 0.768]**
multiplier	0.138 [0.092, 0.205]	0.176 [0.118, 0.251]	0.224 [0.161, 0.302]	**0.316 [0.258, 0.389]**
adder	0.169 [0.111, 0.244]	0.213 [0.148, 0.295]	0.276 [0.198, 0.359]	**0.358 [0.291, 0.431]**
max	0.116 [0.074, 0.181]	0.158 [0.101, 0.229]	0.207 [0.139, 0.284]	**0.294 [0.232, 0.361]**

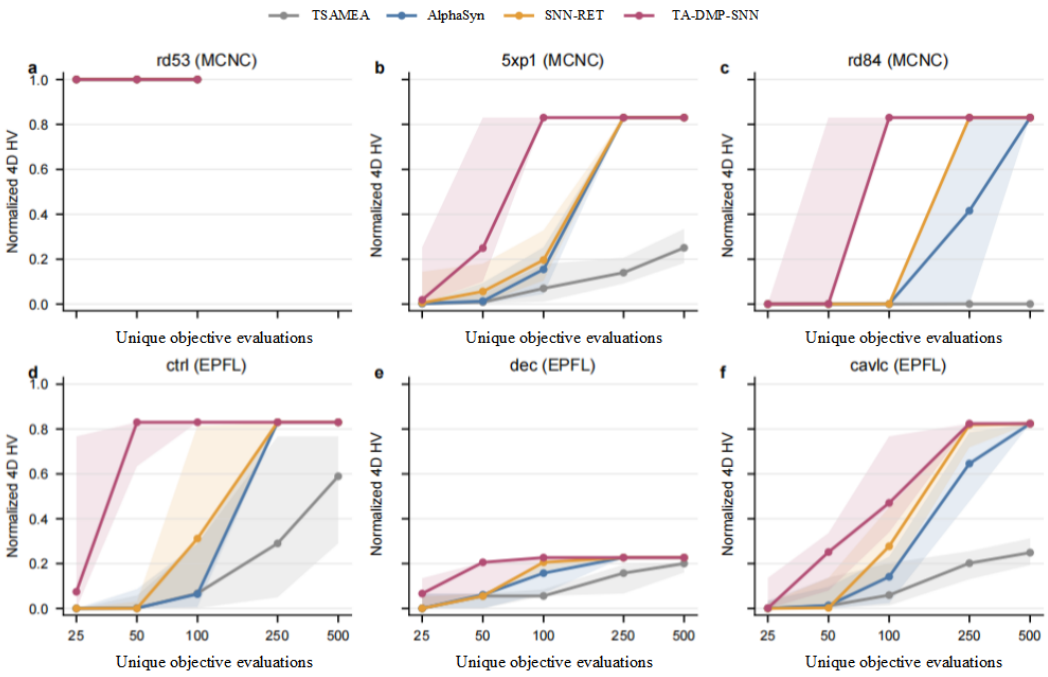

Figure 2: Anytime four-objective hypervolume on the six benchmark circuits. Curves show the median over 30 paired runs and shaded bands show the IQR. Each panel uses circuit-specific pooled normalization bounds that are shared by all methods and frozen across checkpoints.

The other ablations did not produce significant losses after correction. On cavlc, removing fast memory even increased the median from 0.471 to 0.819, although the paired difference did not remain significant after Holm correction ($p_{\text{Holm}} = 0.064$).

5.3 Ablation and runtime analysis.

Figure 3 and 4 summarizes the ablation results after 100 evaluations using 30 paired runs. Positive values indicate that removing a component reduced HV_4. Removing slow archive memory caused median paired losses of 0.646, 0.830, and 0.421 on 5xp1, rd84, and cavlc, respectively. All three losses remained statistically significant after Holm correction ($p_{\text{Holm}} \leq 8.16 \times 10^{-4}$). The other ablations produced no significant losses after correction. On cavlc, removing fast eligibility memory favored the ablated variant by a median paired difference of 0.154, but this trend was not statistically significant ($p_{\text{Holm}} = 0.064$). These results identify slow archive consolidation as the dominant component under the tested budget.

Table 2: Ablation results at 100 evaluations. Entries are median HV_4 [IQR] over 30 paired runs.

Variant	5xp1	rd84	cavlc
Full	**0.830 [0.830, 0.830]**	**0.830 [0.830, 0.830]**	0.471 [0.339, 0.768]
No topology	0.830 [0.830, 0.830]	0.830 [0.830, 0.830]	0.622 [0.314, 0.819]
No fast memory	0.830 [0.454, 0.830]	0.830 [0.830, 0.830]	**0.819 [0.493, 0.824]**
No slow memory	0.128 [0.061, 0.251]	< 0.001 [< 0.001, < 0.001]	0.041 [0.004, 0.088]
No homeostasis	0.830 [0.830, 0.830]	0.830 [0.830, 0.830]	0.622 [0.276, 0.822]

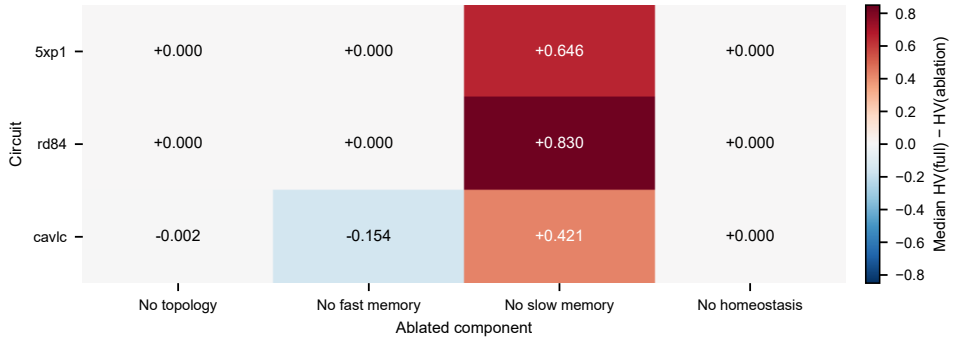

Figure 3: Component ablation at 100 unique evaluations. Color encodes median four-objective hypervolume across 30 paired runs. The consistent degradation after removing slow archive memory is not reproduced by the other removals.

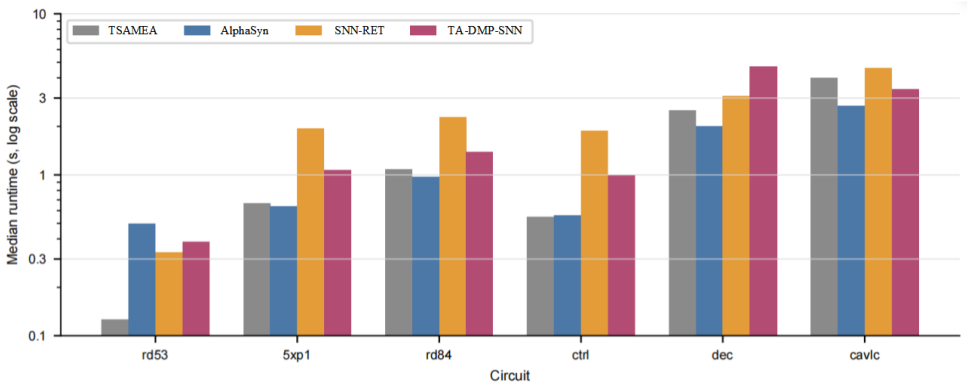

Figure 4: Median wall-clock time over 30 runs. The logarithmic axis exposes differences among the Python implementations, it is not a hardware-normalized throughput comparison.

The Python implementation required median runtimes from 0.381 to 4.678 seconds. AlphaSyn ran faster on five of the six circuits, while TA-DMP-SNN ran faster than SNN-RET on four circuits, as shown in Figure 4.

6 Conclusion

TA-DMP-SNN introduces a circuit-aware spiking search framework for multiobjective MPRM polarity optimization. By combining ternary competitive dynamics, topology-aware gating, and dual-timescale archive memory, the proposed method converts structural knowledge and Pareto experience into adaptive search behavior. Future work will investigate larger-scale synthesis benchmarks and technology-mapped validation.

AI use statement

Generative AI tools were used only to improve the readability and language of the manuscript. They were not used to formulate the research problem, develop the proposed method, design the experiments, implement the method, analyze the data, or interpret the results. All AI-assisted edits were reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

Reproducibility statement

We provide the algorithmic details and pseudocode in Sections 3 and 4. The experimental settings, benchmark descriptions, parameter configurations, and additional results are reported in Appendix A–B.

References

[1] Chen Chen, Guangyu Hu, Dongsheng Zuo, Cunxi Yu, Yuzhe Ma, and Hongce Zhang. E-Syn: E-graph rewriting with technology-aware cost functions for logic synthesis. In *Proceedings of the 61st ACM/IEEE Design Automation Conference (DAC)*, 2024. doi: 10.1145/3649329.3656246.

[2] Chen Chen, Guangyu Hu, Cunxi Yu, Yuzhe Ma, and Hongce Zhang. E-morphic: Scalable equality saturation for structural exploration in logic synthesis. In *Proceedings of the 62nd ACM/IEEE Design Automation Conference (DAC)*, pp. 1–7, 2025. doi: 10.1109/DAC63849.2025.11133110.

[3] Yan Fang and Ashwin Sanjay Lele. Solving quadratic unconstrained binary optimization with collaborative spiking neural networks. In *Proceedings of the IEEE International Conference on Rebooting Computing (ICRC)*, pp. 84–88, 2022. doi: 10.1109/ICRC57508.2022.00021.

[4] Nicolas Frémaux and Wulfram Gerstner. Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Frontiers in Neural Circuits*, 9:85, 2016. doi: 10.3389/fncir.2015.00085.

[5] Eugene M. Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, 2003. doi: 10.1109/TNN.2003.820440.

[6] Jiawei Liu, Jianwang Zhai, Mingyu Zhao, Zhe Lin, Bei Yu, and Chuan Shi. PolarGate: Breaking the functionality representation bottleneck of and-inverter graph neural network. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2024. doi: 10.1145/3676536.3676834.

[7] Tam Ngoc-Bang Nguyen, Anh-Dzung Doan, Zhipeng Cai, and Tat-Jun Chin. Slack-free spiking neural network formulation for hypergraph minimum vertex cover. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2124.

[8] Zehua Pei, Fangzhou Liu, Zhuolun He, Guojin Chen, Haisheng Zheng, Keren Zhu, and Bei Yu. AlphaSyn: Logic synthesis optimization with efficient monte carlo tree search. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2023. doi: 10.1109/ICCAD57390.2023.10323856.

[9] Lu Yang, Shengsheng Wang, Ruyi Dong, and Zihao Fu. A two-stage HV-driven adaptive multi-objective evolutionary algorithm and its application in fixed polarity reed–muller circuits. *Expert Systems with Applications*, 288:128173, 2025. doi: 10.1016/j.eswa.2025.128173.

[10] Meng Yang and Jinmei Lai. Optimisation of mixed polarity reed–muller functions. *Journal of Software*, 8(11):2770–2774, 2013. doi: 10.4304/jsw.8.11.2770-2774.

[11] J. Yuan, P. Wang, J. Ye, M. Yuan, J. Hao, and J. Yan. EasySO: Exploration-enhanced reinforcement learning for logic synthesis sequence optimization and a comprehensive RL environment. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2023. doi: 10.1109/ICCAD57390.2023.10323973.

[12] Yuhao Zhou, Zhenxue He, Tao Wang, Zhisheng Huo, Limin Xiao, and Xiang Wang. An efficient multi-objective polarity optimization algorithm for mixed polarity reed–muller logic circuits. *Applied Soft Computing*, 130:109720, 2022. doi: 10.1016/j.asoc.2022.109720.

[13] Yuhao Zhou, Jianhui Jiang, Zhenxue He, Ying Zhang, Chengcheng Chen, Zhanhui Shi, Wei Zhang, and Keying Yang. An efficient area and reliability optimization method for MPRM circuits based on high-dimensional genetic algorithm. *ACM Transactions on Design Automation of Electronic Systems*, 2025. doi: 10.1145/3712591.

[14] Issam Zidi and Salim El Khediri. SNN-RET: A spiking neural network with reference-guided environmental selection and turbulence for multi- and many-objective optimization. *Swarm and Evolutionary Computation*, pp. 102499, 2026. doi: 10.1016/j.swevo.2026.102499.

A Detailed Objective Modeling and Budgeted Pareto Formulation

This appendix provides the complete definitions of the four structural objectives and the budget-constrained Pareto optimization problem referenced in Section 3.

A.1 Structural Objective Vector

Different polarity assignments can implement the same function with substantially different structures. We evaluate a candidate using four minimization objectives,

$$\mathbf{f}(\mathbf{p}) = [f_{\text{area}}(\mathbf{p}), f_{\text{power}}(\mathbf{p}), f_{\text{delay}}(\mathbf{p}), f_{\text{exp}}(\mathbf{p})]^{\mathsf{T}}. \tag{21}$$

The last component is fault exposure, so a smaller value corresponds to a more reliable structure.

To state the objectives independently of a particular cell library, write the decomposed RM network as $G_{\mathbf{p}} = (V_{\mathbf{p}}, E_{\mathbf{p}})$. Each node $v \in V_{\mathbf{p}}$ has a gate type $\tau(v)$, a fan-in, and a fixed technology-independent cost. The structural area objective is

$$f_{\text{area}}(\mathbf{p}) = \sum_{v \in V_{\mathbf{p}}} a(\tau(v), \text{fanin}(v)), \tag{22}$$

where $a(\cdot)$ is the two-input-equivalent cost assigned to a gate. This definition counts the logic needed by the selected RM realization and does not report physical area in μm^2.

The power-related objective measures switching activity weighted by structural load,

$$f_{\text{power}}(\mathbf{p}) = \sum_{v \in V_{\mathbf{p}}} \alpha_v \, c(\tau(v), \text{fanout}(v)), \tag{23}$$

where α_v is the estimated transition activity at node v under the common input distribution, and $c(\cdot)$ is a technology-independent load weight. Constants such as supply voltage and clock frequency are omitted because they are shared by all candidates. Accordingly, f_{power} is a relative power proxy rather than power in watts.

For the depth objective, let d_v be the unit or gate-type delay of node v. Arrival times are computed recursively as

$$t_v = d_v + \begin{cases} 0, & \text{pred}(v) = \varnothing, \\ \max\limits_{u \in \text{pred}(v)} t_u, & \text{otherwise}, \end{cases} \tag{24}$$

and

$$f_{\text{delay}}(\mathbf{p}) = \max_{v \in V_{\text{PO}}} t_v, \tag{25}$$

where V_{PO} contains the nodes driving the primary outputs. With unit d_v, this objective is the maximum logic depth; it should not be interpreted as delay in nanoseconds.

Finally, the fault-exposure objective is

$$f_{\text{exp}}(\mathbf{p}) = \sum_{v \in V_{\mathbf{p}}} q_v \rho_v, \tag{26}$$

where q_v is the structural fault weight of node v and ρ_v is the estimated probability that a fault at v propagates to at least one primary output. This objective penalizes structures with more fault-sensitive internal logic. It is a technology-independent exposure estimate, not a measured soft-error rate.

The four objectives are evaluated under the same decomposition, input assumptions, and gate-cost model for every method. Their raw values are retained for dominance tests. Normalization is applied only when an algorithm or an evaluation metric requires comparable objective scales; a componentwise strictly monotone normalization does not change Pareto dominance.

A.2 Budgeted Pareto Optimization

No scalar weighting is fixed in advance because the four objectives may favor different polarities. For two assignments $\mathbf{p}, \mathbf{q} \in \Omega_n$, $\mathbf{p}$ Pareto-dominates $\mathbf{q}$, written $\mathbf{p} \prec \mathbf{q}$, if

$$f_k(\mathbf{p}) \le f_k(\mathbf{q}) \quad \forall k \in \{1,2,3,4\}, \qquad f_j(\mathbf{p}) < f_j(\mathbf{q}) \quad \text{for some } j. \tag{27}$$

The Pareto-optimal polarity set and its image in objective space are

$$\mathcal{P}^\star = \{\mathbf{p} \in \Omega_n : \nexists\, \mathbf{q} \in \Omega_n \text{ such that } \mathbf{q} \prec \mathbf{p}\}, \tag{28}$$

$$\mathcal{F}^\star = \{\mathbf{f}(\mathbf{p}) : \mathbf{p} \in \mathcal{P}^\star\}. \tag{29}$$

For small circuits, $\mathcal{P}^\star$ can be recovered by enumerating all 3^n assignments. The intended setting is instead evaluation-limited search. Given a budget of B objective evaluations, an optimizer queries a set $\mathcal{Q}_B \subseteq \Omega_n$ with $|\mathcal{Q}_B| \le B$ and returns

$$\mathcal{A}_B = \text{ND}(\mathcal{Q}_B), \tag{30}$$

where $\text{ND}(\cdot)$ removes dominated candidates and duplicate polarity vectors. The optimization task is to obtain, within the same budget B, an archive whose objective vectors are close to $\mathcal{F}^\star$ and cover its distinct trade-off regions. Thus, the required output is a set of functionally equivalent polarity assignments rather than a single weighted-sum optimum.

B Additional Method Details

This appendix provides the detailed entropy-homeostatic mechanism, archive maintenance rule, complete search procedure, and computational complexity analysis referenced in Section 4.

B.1 Entropy Homeostasis and Archive Update

Structural gates and slow consolidation can over-stabilize a subset of polarity states. We monitor this behavior at the level of each competitive group. Let

$$\pi_{ik} = \frac{r_{ik} + \varepsilon}{\sum_{h=0}^{2}(r_{ih} + \varepsilon)}, \qquad H_i = -\frac{1}{\log 3} \sum_{k=0}^{2} \pi_{ik} \log \pi_{ik}. \tag{31}$$

$H_i = 1$ represents balanced state use and $H_i \approx 0$ represents collapse. When $H_i < 0.25$ for five consecutive windows and the current candidate does not increase hypervolume, the exploration scale is increased,

$$\sigma_i \leftarrow \min(0.8, 1.2\sigma_i). \tag{32}$$

After an improving candidate, all exploration scales decay toward a lower bound of 0.2 by a factor of 0.995. Entropy alone does not trigger exploration when the archive is still improving; the topology gate itself remains fixed.

After objective evaluation, the external archive is updated as

$$\mathcal{A}_t = \mathrm{ND}(\mathcal{A}_{t-1} \cup \{(\mathbf{p}_t, \mathbf{f}(\mathbf{p}_t))\}) . \tag{33}$$

Exact duplicate polarities are removed before dominance testing. If the archive exceeds 200 solutions, boundary points are retained and points with the lowest four-objective crowding distance are deleted until the cap is met.

B.2 Search Procedure and Complexity

Algorithm 1 summarizes the complete search. The network state is carried across candidate windows, whereas the Boolean function and objective evaluator remain fixed. This arrangement makes the objective-evaluation budget explicit and permits direct comparison with evolutionary optimizers.

Algorithm 1: TA-DMP-SNN Search under a Fixed Evaluation Budget

Input: Boolean function $\mathbf{F}$; evaluation budget B; window T;
consolidation interval K; reference vectors Λ
Output: Nondominated archive $\mathcal{A}_B$

Extract C^{der}, C^{rm}, and C^{deg} from $\mathbf{F}$ and construct topology-aware gates g_{ij};
Initialize neuronal states, w^{f}, w^{s}, eligibility traces e, and $\mathcal{A}_0 \leftarrow \varnothing$;
for $t \leftarrow 1$ **to** B **do**
  Simulate Eqs. (6)–(12) for T steps;
  Replay an archive solution when activated and decode $\mathbf{p}_t$ using Eq. (9);
  Construct $\mathcal{C}_{\mathbf{p}_t}$ and evaluate $\mathbf{f}(\mathbf{p}_t)$;
  Update $\mathcal{A}_t$, assign reference region q_t, and compute reward R_t;
  Update e and w^{f} using Eqs. (15)–(16);
  if $t \bmod K = 0$ **then**
    Update w^{s} through archive consolidation using Eqs. (17)–(19);
  end
  if *the search stagnates* **then**
    Apply entropy-based homeostatic regulation;
  end
end
return $\mathcal{A}_B$;

Let $E_{mathrmsnn}$ be the number of active inter-group synapses, C_{eval} the cost of RM conversion and four-objective evaluation, $M = 4$ the number of objectives, and A the maximum archive size. A candidate requires

$$O(TE_{\mathrm{snn}} + C_{\mathrm{eval}} + MA) \tag{34}$$

time with a linear archive scan. Slow consolidation adds $O(AE_{\mathrm{snn}})$ every K evaluations. The total cost over budget B is therefore

$$O\left[B(TE_{\mathrm{snn}} + C_{\mathrm{eval}} + MA) + \frac{B}{K} AE_{\mathrm{snn}}\right] . \tag{35}$$

Memory consumption is $O(E_{\mathrm{snn}} + A(n + M))$. In the intended setting, RM transformation and objective evaluation dominate neuron updates. TA-DMP-SNN is therefore designed to improve the allocation of a fixed evaluation budget, not to reduce the cost of an individual circuit evaluation.