PULSESYN: LEARNING TO CO-OPTIMIZE REVERSIBLE LOGIC SYNTHESIS AND QUANTUM CONTROL PULSES

ABSTRACT

Reversible Boolean-oracle synthesis is usually optimized with gate-level metrics, although the cost of a realization depends on downstream decomposition and control. We present PULSESYN, a verified common-control factorization framework that evaluates symbolic rewrites under a decomposition-aware objective. Starting from algebraic normal form (ANF), PULSESYN materializes repeated control products in clean ancillas, reuses them across compatible monomials, and uncomputes them after their final use. On ten MCNC/EPFL pilot circuits, full-evaluation greedy search reduces the two-qubit decomposition proxy by 35.69% and analytical control effort by 37.22% on average. In a circuit-family transfer test, a ranker trained on MCNC candidates reproduces the full-evaluation factorization on all three unseen EPFL circuits while avoiding 79.7% of exact candidate evaluations.

1 INTRODUCTION

Quantum compilation crosses several representations, from Boolean logic and reversible circuits to native gates and physical controls. Each layer exposes a different cost model. Reversible synthesis commonly optimizes gate count, two-qubit operations, depth, or ancillas, whereas downstream cost is determined only after decomposition and realization. Two logically equivalent oracles can therefore look similar symbolically yet induce different low-level costs. Optimizing only the symbolic representation may commit the compiler to an expensive structure before the downstream objective becomes visible.

For a multi-output Boolean function $f : \{0,1\}^n \to \{0,1\}^m$, the reversible oracle satisfies

$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle. \tag{1}$$

Algebraic normal form (ANF) and exclusive-sum-of-products (ESOP) synthesis provide exact procedures for translating Boolean functions into reversible circuits while revealing opportunities for algebraic sharing (Gupta et al., 2006; Schmitt et al., 2019; Soeken et al., 2019; Meuli et al., 2020). Separately, pulse-aware and cross-layer compilers show that decomposition and control choices can materially change execution cost (Gokhale et al., 2019; Shi et al., 2019; Cho et al., 2024; Ren et al., 2024; Fauquenot et al., 2025). These observations motivate a narrower question: *can an exact symbolic rewrite be selected using a downstream decomposition-aware objective without placing learning inside the correctness boundary?*

PULSESYN answers this question with ancilla-bounded common-control factorization. Repeated ANF control subsets define candidates; a selected subset is computed into a clean ancilla, reused across compatible monomials, and uncomputed after its final use. Candidate value is measured after a fixed decomposition rather than from reuse count alone. Because overlapping factors interact, gains change after every accepted rewrite. PULSESYN therefore uses a learned ranker to prioritize candidates, but recomputes algebraic legality and the current objective gain exactly before acceptance.

The methodological distinction is deliberately specific. Common-control factorization, resource-constrained oracle synthesis, pulse-aware compilation, and learning-guided logic optimization are established ideas (Meuli et al., 2020; Grosnit et al., 2022; Chowdhury et al., 2024; Bai et al., 2025). PULSESYN does not claim a new ANF representation, a globally optimal reversible synthesizer, or pulse-level optimal control. Its contribution is the interface between these layers: symbolic factors are scored under a declared downstream decomposition, while learning acts only as a proposal mechanism and exact verification remains authoritative.

Our experiments are designed around that scope. On ten MCNC/EPFL circuits, full-evaluation greedy search reduces the declared two-qubit and control-effort proxies. For the learning claim, complete MCNC circuits provide training candidates and complete EPFL circuits form the held-out family; on

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.

EPFL, learned ordering reproduces the full-evaluation result while eliminating most exact candidate evaluations. A negative ASAP7 transfer test is retained to show that a factor profitable for reversible compute–use–uncompute accounting need not transfer to a different backend objective. We use *control effort* only for the analytical objective and do not interpret it as wall-plug energy or calibrated device fidelity.

Our contributions are:

- **Verified decomposition-aware factorization.** We couple exact ANF common-control reuse with a downstream objective that accounts for decomposition and a bounded clean-ancilla budget.
- **Learning outside the correctness boundary.** A permutation-invariant ranker proposes promising interacting factors, while every accepted rewrite passes independent legality and exact-gain checks.
- **Reviewer-auditable evidence and scope.** We report exhaustive equivalence checks, per-circuit proxy reductions, circuit-family ranker transfer, candidate-evaluation savings, ancilla sensitivity, and a negative backend-transfer result, while explicitly separating these findings from unmeasured physical-energy claims.

2 Background and Problem Formulation

2.1 Exact ANF realization and common-control reuse

For each output f_j, ANF is the unique multilinear polynomial over $\mathrm{GF}(2)$,

$$f_j(x) = c_{j,\varnothing} \oplus \bigoplus_{S\subseteq[n],S\neq\varnothing} c_{j,S} \prod_{i\in S} x_i, \qquad c_{j,S} \in \{0,1\}. \tag{2}$$

A constant maps to X, a degree-one monomial to CNOT, and a degree-k monomial to a k-controlled X. This exact correspondence underlies Reed–Muller/ESOP and hierarchical reversible synthesis (Gupta et al., 2006; Schmitt et al., 2019; Soeken et al., 2019). If several monomials contain a nonempty subset R, a clean ancilla can hold

$$a_R = \prod_{i\in R} x_i, \tag{3}$$

so that $\prod_{i\in S} x_i$ becomes $a_R \prod_{i\in S\setminus R} x_i$ for $R \subset S$. Reversing the computation after the final dependent operation restores a_R to $|0\rangle$. Profitability is not determined by reuse frequency alone: it depends on factor order, affected monomial degrees, decomposition cost, and the clean-ancilla budget.

2.2 Decomposition-aware objective

Let $\Pi_{\mathcal{G}}(C)$ be a deterministic decomposition of circuit C into primitive gates $g \in \mathcal{G}$. Assign each primitive a normalized effort

$$\epsilon_g = \int_0^{T_g} |\Omega_g(t)|^2 \, \mathrm{d}t, \tag{4}$$

and define

$$E_{\mathrm{ctrl}}(C) = \sum_{g\in\Pi_{\mathcal{G}}(C)} \epsilon_g. \tag{5}$$

Let $C_{2\mathrm{q}}(C)$ denote the two-qubit decomposition proxy induced by the same declared realization. We write the selection objective as

$$J_\lambda(C) = \lambda_{2\mathrm{q}} C_{2\mathrm{q}}(C) + \lambda_{\mathrm{ctrl}} E_{\mathrm{ctrl}}(C), \qquad \lambda_{2\mathrm{q}}, \lambda_{\mathrm{ctrl}} \geq 0. \tag{6}$$

This notation covers control-only search and the equal-weight joint pilot used in Section 4. The accounting makes the downstream decomposition part of symbolic synthesis rather than charging one unit per high-level controlled operation. It is motivated by pulse-aware compilation and control optimization (Cho et al., 2024; Ren et al., 2024; Genois et al., 2025; Fauquenot et al., 2025; Hoyt

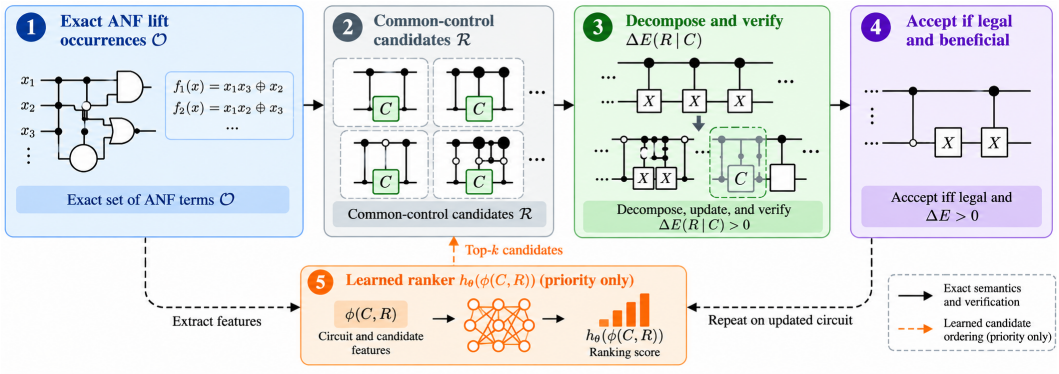


Figure 1: Overview of the PulseSyn Learning-Guided and Exact-Verified Synthesis Framework.

et al., 2026), but is intentionally narrower than total system energy: refrigeration, electronics, readout, routing, crosstalk, and device transfer functions are outside Eq. (5) (Calero et al., 2025).

Given a Boolean function f, clean-ancilla budget B, and a fixed downstream model, let $\mathcal{C}(f, B)$ contain the circuits reachable by permitted exact transformations and satisfying

$$C|x\rangle|y\rangle|0\rangle^{\otimes B} = |x\rangle|y \oplus f(x)\rangle|0\rangle^{\otimes B}. \tag{7}$$

The optimization target is

$$C^\star \in \arg\min_{C \in \mathcal{C}(f,B)} J_\lambda(C), \tag{8}$$

while the algorithm below is a verified greedy search and does not claim global optimality.

2.3 Learned guidance with verified acceptance

Learning-based synthesis has been used to search logic recipes, Boolean rewrites, circuit embeddings, and quantum rewrite sequences (Grosnit et al., 2022; Cai et al., 2024; Chowdhury et al., 2024; Shi et al., 2023; 2024; Bai et al., 2025; Riu et al., 2025; Ruiz et al., 2025). PulseSyn uses learning more conservatively: a predictor ranks candidates, while legality and the objective change are recomputed from the current circuit. A poor prediction can therefore waste evaluations or select a weaker greedy path, but it cannot admit an illegal rewrite or alter the oracle semantics. The ranker architecture is not part of the correctness argument and is not claimed as a standalone contribution.

3 PulseSyn

3.1 Exact lifting and ancilla-bounded factorization

For each output, a Boolean Möbius transform converts the truth table to ANF in $O(mn2^n)$ time and $O(m2^n)$ memory. We store the lifted specification as monomial–output occurrences

$$\mathcal{O} = \{(j, S) : c_{j,S} = 1\}. \tag{9}$$

For circuit state C, the active support of factor R is $\mathcal{O}_C(R) = \{(j, S) \in \mathcal{O}_C : R \subset S\}$. The pilot implementation enumerates factor orders two through four and requires at least two active occurrences. Because accepted factors alter residual monomials, supports are updated after every rewrite.

For a legal candidate R, PulseSyn computes Eq. (3), replaces every affected product by $a_R \prod_{i \in S \setminus R} x_i$, and uncomputes a_R after its final use. This yields the following invariant.

Proposition 1 (Functional preservation). *Every legal compute–use–uncompute factorization preserves U_f and restores its clean ancilla to $|0\rangle$.*

Proof sketch. While a_R is live, $a_R = \prod_{i \in R} x_i$ on every basis state; hence $a_R \prod_{i \in S \setminus R} x_i = \prod_{i \in S} x_i$. Reversing the factor computation restores the ancilla. Equality on basis states extends by linearity. □

Algorithm 1 Safe control-aware common-factor search

Require: ANF occurrences $\mathcal{O}$, factor-ancilla budget B, ordering mode q, shortlist ratio ρ
Ensure: Functionally equivalent circuit C

$C \leftarrow \textsc{DirectANFOracle}(\mathcal{O})$; $A \leftarrow \varnothing$
while $|A| < B$ **do**
  $\mathcal{R} \leftarrow \textsc{LegalCandidates}(C)$
  if $\mathcal{R} = \varnothing$ **then**
    break
  end if
  if $q = \textsc{FullEval}$ **then**
    $\mathcal{S} \leftarrow \mathcal{R}$
  else
    rank $\mathcal{R}$ by decreasing $h_\theta(\phi(C, R))$
    $\mathcal{S} \leftarrow$ top $\max(1, \lceil \rho|\mathcal{R}| \rceil)$ candidates
  end if
  compute exact gain for every $R \in \mathcal{S}$
  $R^\star \leftarrow \arg\max_{R \in \mathcal{S}} \Delta J(R \mid C)$
  if $\Delta J(R^\star \mid C) \leq 0$ **then**
    break
  end if
  $C \leftarrow T_{R^\star}(C)$; $A \leftarrow A \cup \{R^\star\}$
end while
return C

3.2 Verified gain and learned ordering

Let $J(C)$ abbreviate the declared downstream objective $J_\lambda(C)$ in Eq. (6). For control-only search, $\lambda_{2q} = 0$; for the joint pilot, both declared weights are one. For the exact transformation $T_R(C)$, the verified marginal gain is

$$\Delta J(R \mid C) = J(C) - J(T_R(C)) \,. \tag{10}$$

It includes factor computation, all rewritten target operations, and uncomputation, and is recomputed after every accepted rewrite. In the pilot, the joint score used for factor selection is therefore the sum of the exact two-qubit-proxy reduction and the exact control-effort reduction. Full-evaluation greedy search evaluates all current candidates. The learned variant predicts this joint score from 13 permutation-invariant statistics: factor size, support size, output coverage, mean/maximum/minimum/standard-deviation of affected monomial degree, mean/maximum residual degree, degree and residual-degree sums, support density, and normalized output coverage. The pilot implementation uses an MLP with hidden widths $(64, 32)$ and ReLU activations, but the proposal–verification design is ranker agnostic; Appendix A compares simpler alternatives.

At inference, the model only proposes a shortlist: candidates are ranked by prediction, the top 20% (at least one) are evaluated exactly, and the best verified positive-gain candidate is accepted. Algorithm 1 makes this proposal–verification boundary explicit.

Theorem 1 (Safe learned search)**.** *Every circuit returned by Algorithm 1 implements U_f, restores all clean ancillas, and has declared selection objective J no greater than that of the direct ANF circuit.*

Because acceptance requires $\Delta J > 0$, the declared selection objective decreases monotonically at every accepted step. Functional correctness follows from Proposition 1 and is independent of ranker accuracy. The search is greedy rather than globally optimal.

4 Experiments

4.1 Experimental setup and preprocessing

We evaluate three questions: (Q1) whether exact common-control factorization reduces the declared reversible-cost proxies while preserving the Boolean oracle; (Q2) whether learned ordering can reduce exact candidate evaluations without sacrificing the verified gain; and (Q3) whether the resulting

rewrites transfer to a conventional technology-mapping backend. The six primary circuits (`5xp1`, `rd53`, `squar5`, `ctrl`, `dec`, and `int2float`) are retained from the original evaluation, and four additional circuits (`9sym`, `ex5`, `misex1`, and `cavlc`) broaden the coverage to ten MCNC/EPFL benchmarks. The final set spans 101 to 6,560 reported nonconstant ANF monomial occurrences.

PLA input dashes are interpreted as cube wildcards and output don't-care symbols are ignored under the study convention. BLIF networks are restricted to combinational designs and evaluated in topological order. For every circuit, we enumerate the truth table, apply a Boolean Möbius transform over $\mathrm{GF}(2)$, reconstruct the function from the resulting ANF, and exhaustively verify equivalence to the source truth table. All factorized variants use at most three clean factor ancillas and enumerate common-control factors of size two through four. All optimized circuits passed exhaustive truth-table equivalence checking. As shown in Table 1, JOINT-EXACT reduced the two-qubit proxy by an average of 33.44% on the five MCNC circuits and 30.94% on the five EPFL circuits. The corresponding control-effort reductions were 33.11% and 36.30%, respectively. Across all ten circuits, the macro-averaged reductions in the two objectives were 32.19% and 34.70%.

Table 1: Per-circuit logic-synthesis and technology-mapping results.

Circuit	**Family**	**ANF terms**	**$2q$ proxy↑**	**Ctrl. effort↑**	**Eval.↓**	**ASAP7↑**
5xp1	MCNC	101	41.06%	46.21%	46/224	8.05%
rd53	MCNC	520	21.40%	16.33%	8/30	5.08%
squar5	MCNC	532	47.93%	39.00%	10/45	8.02%
ctrl	EPFL	181	52.02%	50.22%	31/143	11.22%
dec	EPFL	6,560	31.99%	44.30%	90/444	10.09%
int2float	EPFL	383	19.74%	27.26%	200/992	7.07%
9sym	MCNC	410	17.69%	17.54%	68/336	13.05%
ex5	MCNC	2,382	33.27%	42.16%	90/444	6.02%
misex1	MCNC	364	36.00%	33.72%	31/150	7.88%
cavlc	EPFL	2,782	20.79%	30.30%	221/1101	1.38%

Circuit and Family identify the benchmark and source suite, respectively. ANF terms denotes the reported number of nonconstant ANF monomial occurrences across all outputs. The $2q$ proxy and Ctrl. effort columns report the percentage reductions achieved by JOINT-EXACT over DIRECT. Eval. reports exact candidate evaluations for JOINT-LEARNED/JOINT-EXACT; smaller ratios indicate stronger pruning. ASAP7 reports mapped-area reduction of JOINT-EXACT relative to DIRECT after Yosys/ABC mapping with the supplied ASAP7 cell library.

4.2 COMPARED METHODS AND COST MODELS

DIRECT emits one multi-controlled operation per nonconstant ANF term. SUPPORT ranks factors by factor size times reuse count. GATE-ONLY and PULSE-ONLY greedily optimize the declared two-qubit and control-effort proxies, respectively. JOINT-EXACT performs full-candidate-evaluation greedy search under the combined exact reduction. JOINT-LEARNED ranks candidates from the 13 structural features defined in Section 3, exactly evaluates only the top 20% shortlist at each step, and accepts a rewrite only when the recomputed objective improves. Thus, learning changes search order but not legality or acceptance.

For a term with k controls, the pilot uses

$$C_{2\mathrm{q}}(k) = \begin{cases} 0, & k = 0, \\ 1, & k = 1, \\ 12k - 18, & k \geq 2, \end{cases} \qquad C_{\mathrm{ctrl}}(k) = (k+1)^2[1 + 0.15\max(0, k-2)]\,. \tag{11}$$

These quantities are reproducible analytical optimization targets rather than measurements of physical energy. Runtime, when reported, includes candidate construction, model inference, and exact acceptance checks.

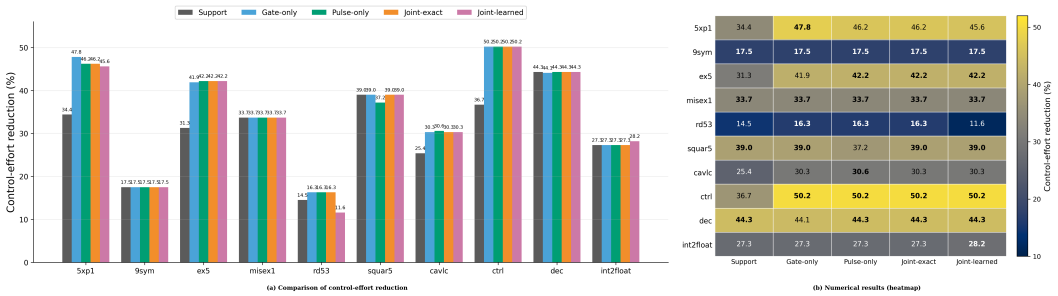


Figure 2: Analytical control-effort reduction relative to DIRECT. Values are program outputs under Eq. (11), not device-level energy measurements.

4.3 LOGIC-LEVEL REDUCTION AND CORRECTNESS

All ten optimized circuits pass exhaustive truth-table equivalence checking, and every accepted factor expands to the original ANF occurrences. On the six primary circuits, JOINT-EXACT reduces the two-qubit proxy and analytical control effort by 35.69% and 37.22% on average, respectively. The four additional circuits provide reductions of 17.69–36.00% in the two-qubit proxy and 17.54–42.16% in control effort; over all ten circuits, the corresponding means are 32.19% and 34.70%. The per-circuit spread in Table 1 reflects differences in reusable control structure and monomial degree.

Figure 2 compares the factor-selection objectives. Gate-only, pulse-only, and joint scoring often select similar high-value factors, so the evidence supports decomposition-aware common-control reuse more strongly than a claim that equal-weight joint scoring uniformly dominates either individual objective.

4.4 LEARNED RANKING AND SEARCH EFFICIENCY

The learning experiment uses circuit-level separation rather than candidate-level random splitting, so candidates from a held-out circuit are never used to train its ranker. On the primary held-out EPFL subset, JOINT-LEARNED evaluates 321 rather than 1,579 candidates, avoiding 79.7% of exact evaluations. The four added circuits provide a broader stress test, and the ten-circuit totals in Table 1 are 795 versus 3,909 exact evaluations, corresponding to a 79.7% overall reduction.

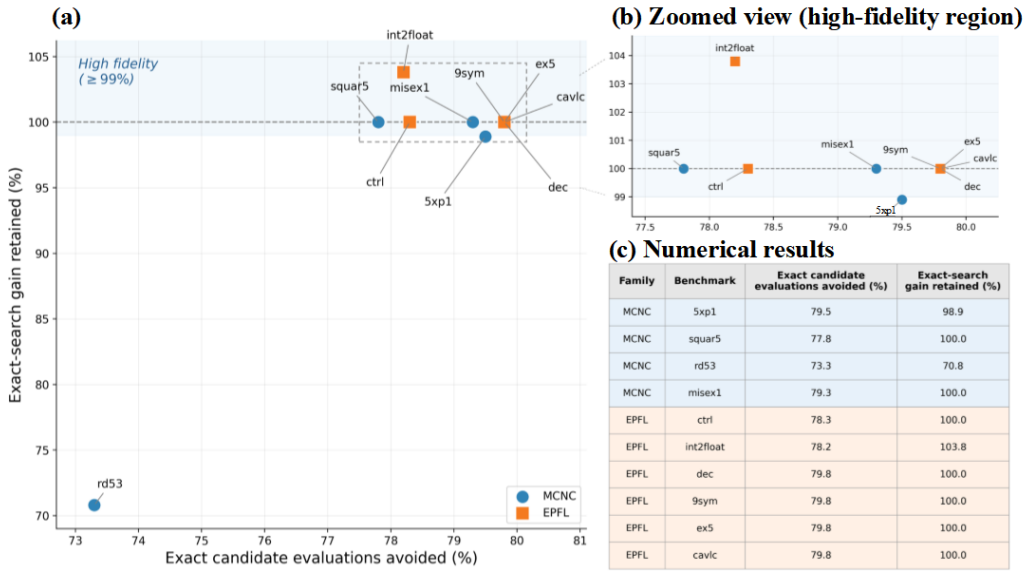
The retained-gain results clarify what this pruning does and does not guarantee. The learned path matches the full-evaluation control-effort gain on most circuits; the reported exceptions are `5xp1` (98.8% retained) and `rd53` (70.8%), while `int2float` reaches 103.7% because pruning follows a different greedy trajectory. Figure 3 therefore supports a search-efficiency claim rather than global equivalence between the learned and full-evaluation trajectories.

The ranker comparison in Appendix A further limits the learning claim: on held-out candidates, the linear model attains Spearman $\rho = 0.999$, compared with 0.939 for the MLP and 0.741 for HGBR. The contribution is therefore *learned prioritization with exact verification*, not neural-model superiority. On these small instances, inference overhead can also dominate wall-clock time; the primary efficiency metric is the number of exact downstream evaluations avoided.

4.5 ASAP7 BACKEND TRANSFER

We additionally export functionally equivalent balanced-XOR BLIF networks and map them with Yosys/ABC using the supplied ASAP7 genlib. The same mapping flow is applied to DIRECT and factorized circuits, and cell area is computed from mapped cell counts and the areas declared in the genlib. Because this is a standard-cell transfer experiment rather than a reversible-native or calibrated pulse backend, it is used only as a backend sanity check.

Under the final mapping flow reported in Table 1, the per-circuit ASAP7 area reductions range from 1.38% to 13.05%, with a ten-circuit mean of 7.79%. These results show that the exposed sharing can survive conventional mapping on the reported instances, but they do not validate the



(c) Numerical results

Family	Benchmark	Exact candidate evaluations avoided (%)	Exact-search gain retained (%)
MCNC	5xp1	79.5	98.9
MCNC	squar5	77.8	100.0
MCNC	rd53	73.3	70.8
MCNC	misex1	79.3	100.0
EPFL	ctrl	78.3	100.0
EPFL	int2float	78.2	103.8
EPFL	dec	79.8	100.0
EPFL	9sym	79.8	100.0
EPFL	ex5	79.8	100.0
EPFL	cavlc	79.8	100.0

Figure 3: Exact candidate evaluations avoided versus full-evaluation control-effort gain retained. Learned ordering removes most exact evaluations while exact verification remains authoritative.

analytical control-effort proxy as physical energy or fidelity. The appropriate conclusion remains backend-specific: a rewrite should be judged under the decomposition and cost model of its target realization. Additional mapping results are provided in Appendix A. R^2 measures the goodness of fit for the predicted gains, with values closer to 1 indicating better predictive performance.

4.6 Scope and Limitations

The ten-circuit study establishes functional correctness, proxy-level common-control savings, and substantial pruning of exact candidate evaluations, but it remains a proof-of-concept evaluation. Stronger generalization claims require a larger circuit-level split, repeated training seeds, and a reversible-native or calibrated pulse backend. Likewise, the present results do not establish global optimality or MLP superiority. These boundaries keep the empirical claim aligned with what is directly verified by the current experiments.

5 Conclusion

We presented PulseSyn, a decomposition-aware framework for exact reversible Boolean-oracle synthesis. PulseSyn combines ANF common-control factorization with downstream cost evaluation and uses learning only to prioritize candidates before deterministic legality and gain checks. On ten pilot circuits, full-evaluation search preserves functionality and reduces the declared reversible-cost proxies; on the held-out EPFL family, learned ordering reproduces the full-evaluation factorization while avoiding 79.7% of exact candidate evaluations. The negative ASAP7 transfer result is equally important because it demonstrates that factorization value is backend dependent. The present result is therefore a verified proof-of-concept for proposal–verification synthesis, not a claim of global optimality, neural-ranker superiority, or hardware-energy reduction.

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

The supplementary package contains the benchmark instances, preprocessing and ANF extraction code, factorization and mapping scripts, fixed random seed, raw pilot results, generated figures, and the ASAP7 genlib used in the transfer test. Section 4 specifies the circuit-family train/test split, analytical cost functions, shortlist ratio, ancilla constraint, and equivalence checks; Appendix A reports ranker, ancilla-sensitivity, and backend-transfer details.

References

Yinqi Bai, Jie Wang, Lei Chen, Zhihai Wang, Yufei Kuang, Mingxuan Yuan, Jianye Hao, and Feng Wu. A graph enhanced symbolic discovery framework for efficient logic optimization. In *International Conference on Learning Representations*, 2025.

Yuting Cai, Yue Wu, Xiaoyan Yang, and Zhufei Chu. A logic optimization method using reinforcement learning. In *2024 2nd International Symposium of Electronics Design Automation (ISEDA)*, pp. 312–317, 2024. doi: 10.1109/ISEDA62518.2024.10618015.

Coral Calero, Macario Polo, Elena Desdentado, and M. Ángeles Moraga. Consider the energy consumption of your quantum circuits. *Nature Reviews Physics*, 7:352–353, 2025. doi: 10.1038/s42254-025-00846-0.

Yujin Cho, Kristin M. Beck, Alessandro R. Castelli, Kyle A. Wendt, Bram Evert, Matthew J. Reagor, and Jonathan L. DuBois. Direct pulse-level compilation of arbitrary quantum logic gates on superconducting qutrits. *Physical Review Applied*, 22:034066, 2024. doi: 10.1103/PhysRevApplied.22.034066.

Animesh Basak Chowdhury, Marco Romanelli, Benjamin Tan, Ramesh Karri, and Siddharth Garg. Retrieval-guided reinforcement learning for boolean circuit minimization. In *International Conference on Learning Representations*, 2024.

Sebastiaan Fauquenot, Aritra Sarkar, and Sebastian Feld. Open and closed loop approaches for energy efficient quantum optimal control. *Advanced Quantum Technologies*, 8(10):2400690, 2025. doi: 10.1002/qute.202400690.

Élie Genois, Noah J. Stevenson, Noah Goss, Irfan Siddiqi, and Alexandre Blais. Quantum optimal control of superconducting qubits based on machine-learning characterization. *Physical Review Applied*, 24:034073, 2025. doi: 10.1103/d9yg-d3qr.

Pranav Gokhale, Yongshan Ding, Thomas Propson, Christopher Winkler, Nelson Leung, Yunong Shi, David I. Schuster, Henry Hoffmann, and Frederic T. Chong. Partial compilation of variational algorithms for noisy intermediate-scale quantum machines. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 266–278, 2019. doi: 10.1145/3352460.3358313.

Antoine Grosnit, Cédric Malherbe, Rasul Tutunov, Xingchen Wan, Jun Wang, and Haitham Bou-Ammar. BOiLS: Bayesian optimisation for logic synthesis. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1193–1196, 2022. doi: 10.23919/DATE54114.2022.9774632.

Pallav Gupta, Abhinav Agrawal, and Niraj K. Jha. An algorithm for synthesis of reversible logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25 (11):2317–2330, 2006. doi: 10.1109/TCAD.2006.871622.

Aaron Hoyt, Meng Wang, Fei Hua, Chunshu Wu, Chenxu Liu, Muqing Zheng, Samuel Stein, Drew Rebar, Yufei Ding, Travis S. Humble, and Ang Li. QASMTrans: An end-to-end QASM compilation framework with pulse generation for near-term quantum devices. *ACM Transactions on Quantum Computing*, 2026. doi: 10.1145/3837861.

Giulia Meuli, Mathias Soeken, Martin Roetteler, and Giovanni De Micheli. ROS: Resource-constrained oracle synthesis for quantum computers. *Electronic Proceedings in Theoretical Computer Science*, 318:119–130, 2020. doi: 10.4204/EPTCS.318.8.

Xiangyu Ren, Junjie Wan, Zhiding Liang, and Antonio Barbalace. Tackling coherent noise in quantum computing via cross-layer compiler optimization. *arXiv preprint arXiv:2410.09664*, 2024.

Jordi Riu, Jan Nogué, Gerard Vilaplana, Artur Garcia-Saez, and Marta P. Estarellas. Reinforcement learning based quantum circuit optimization via ZX-calculus. *Quantum*, 9:1758, 2025. doi: 10.22331/q-2025-05-28-1758.

Francisco J. R. Ruiz, Tuomas Laakkonen, Johannes Bausch, Matej Balog, Mohammadamin Barekatain, Francisco J. H. Heras, Alexander Novikov, Nathan Fitzpatrick, Bernardino Romera-Paredes, John van de Wetering, Alhussein Fawzi, Konstantinos Meichanetzidis, and Pushmeet Kohli. Quantum circuit optimization with AlphaTensor. *Nature Machine Intelligence*, 7:374–385, 2025. doi: 10.1038/s42256-025-01001-1.

Bruno Schmitt, Mathias Soeken, Giovanni De Micheli, and Alan Mishchenko. Scaling-up ESOP synthesis for quantum compilation. In *Proceedings of the 49th IEEE International Symposium on Multiple-Valued Logic*, pp. 13–18, 2019. doi: 10.1109/ISMVL.2019.00011.

Yunong Shi, Nelson Leung, Pranav Gokhale, Zane Rossi, David I. Schuster, Henry Hoffmann, and Frederic T. Chong. Optimized compilation of aggregated instructions for realistic quantum computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 1031–1044, 2019. doi: 10.1145/3297858.3304018.

Zhengyuan Shi, Hongyang Pan, Sadaf Khan, Min Li, Yi Liu, Junhua Huang, Hui-Ling Zhen, Mingxuan Yuan, Zhufei Chu, and Qiang Xu. DeepGate2: Functionality-aware circuit representation learning. In *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp. 1–9, 2023. doi: 10.1109/ICCAD57390.2023.10323798.

Zhengyuan Shi, Ziyang Zheng, Sadaf Khan, Jianyuan Zhong, Min Li, and Qiang Xu. DeepGate3: Towards scalable circuit representation learning. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, pp. 184:1–184:9, 2024. doi: 10.1145/3676536.3676791.

Mathias Soeken, Martin Roetteler, Nathan Wiebe, and Giovanni De Micheli. LUT-based hierarchical reversible logic synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(9):1675–1688, 2019. doi: 10.1109/TCAD.2018.2859251.

A Additional Experimental Results

Table 2: Circuit-family ranker transfer from MCNC candidates to unseen EPFL candidates.

Ranker	MAE	R^2	Spearman ρ
Linear	5583.48	0.963	0.999
HGBR	33757.64	0.838	0.741
MLP	10155.87	0.877	0.939

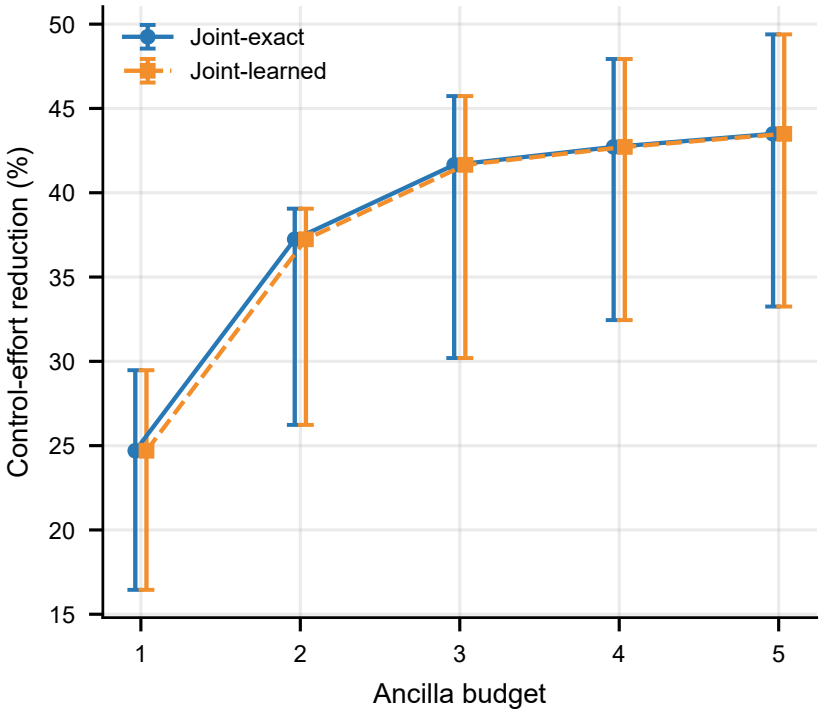


Figure 4: Control-effort reduction versus clean-ancilla budget. Error bars summarize variation across the ten pilot circuits.

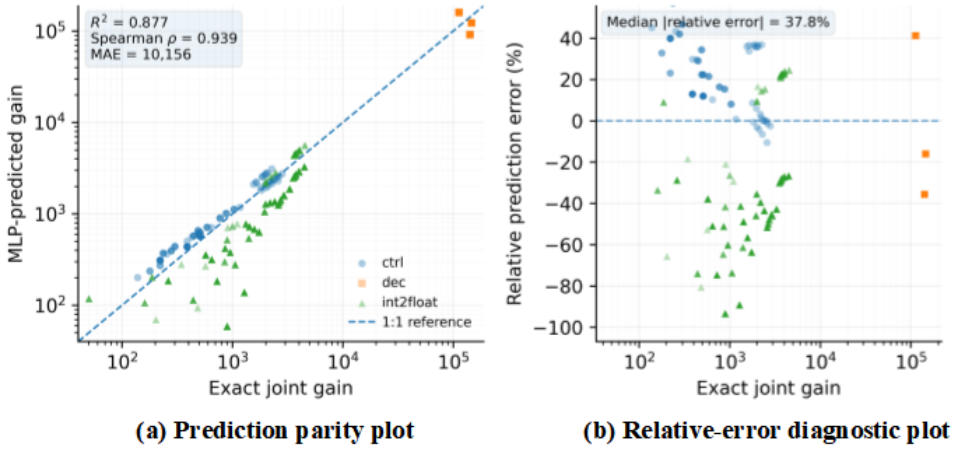


Figure 5: MLP-predicted versus exact joint gain for held-out EPFL candidates. The spread at large gains motivates exact verification even when ranking quality is useful.