ISSN 2579-2784 (Print)
ISSN 2538-2788 (Online)

MATHEMATICAL PROBLEMS OF COMPUTER SCIENCE

LXIV

Yerevan
2025

Հայաստանի Հանրապետության Գիտությունների ազգային ակադեմիայի
Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ

Институт проблем информатики и автоматизации Национальной академии наук
Республики Армения

Institute for Informatics and Automation Problems of the National Academy of
Sciences of the Republic of Armenia

Կոմպյուտերային գիտության մաթեմատիկական խնդիրներ

Математические проблемы компьютерных наук

Mathematical Problems of Computer Science

LXIV

ՀՐԱՏԱՐԱԿՎԱԾ Է ՀՀ ԳԱԱ ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԱՎՏՈՄԱՏԱՑՄԱՆ
ՊՐՈԲԼԵՄՆԵՐԻ ԻՆՍՏԻՏՈՒՏԻ ԿՈՂՄԻՑ
ОПУБЛИКОВАНО ИНСТИТУТОМ ПРОБЛЕМ ИНФОРМАТИКИ И
АВТОМАТИЗАЦИИ НАН РА
PUBLISHED BY THE INSTITUTE FOR INFORMATICS AND AUTOMATION
PROBLEMS OF NAS RA

ԵՐԵՎԱՆ 2025 YEREVAN

Կոմպյուտերային գիտության մաթեմատիկական խնդիրներ, LXIV

Կոմպյուտերային գիտության մաթեմատիկական խնդիրներ պարբերականը հրատարակվում է տարեկան երկու անգամ ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտի (ԻԱՊԻ) կողմից: Այն ընդգրկում է տեսական և կիրառական մաթեմատիկայի, ինֆորմատիկայի և հաշվողական տեխնիկայի ժամանակակից ուղղությունները:

Այն ընդգրկված է Բարձրագույն որակավորման հանձնաժողովի ընդունելի ամսագրերի ցանկում:

Տպագրվում է Խմբագրական խորհրդի 2025թ. նոյեմբերի 24-ի N 25-11/1 նիստի որոշման հիման վրա

ISSN 2579-2784 (Print)
ISSN 2738-2788 (Online)

Математические проблемы компьютерных наук, LXIV

Журнал **Математические проблемы компьютерных наук** издается два раза в год Институтом проблем информатики и автоматизации НАН РА. Он охватывает современные направления теоретической и прикладной математики, информатики и вычислительной техники.

Он включен в список допустимых журналов Высшей квалификационной комиссии.

Печатается на основании решения N 25-11/1 заседания
Редакционного совета от 24 ноября 2025г.

ISSN 2579-2784 (Print)
ISSN 2738-2788 (Online)

Mathematical Problems of Computer Science, LXIV

The periodical **Mathematical Problems of Computer Science** is published twice per year by the Institute for Informatics and Automation Problems of NAS RA. It covers modern directions of theoretical and applied mathematics, informatics and computer science.

It is included in the list of acceptable journals of the Higher Qualification Committee.

Printed on the basis of decision N 25-11/1 of the session of the Editorial Council dated November 24, 2025.

ISSN 2579-2784 (Print)

ISSN 2738-2788 (Online)

Պարբերականի սույն պրակը նվիրվում է տեխնիկական գիտությունների դոկտոր, պրոֆեսոր, ՀՀ ԳԱԱ ակադեմիկոս Յուրի Հայկի Շուքուրյանի 85-ամյա հոբելյանին:

Данный выпуск журнала посвящается 85-летнему юбилею Юрия Гайковича Шукуряна, доктора технических наук, профессора, академика НАН РА.

This issue of the Periodical is dedicated to the 85-th anniversary of Yuri H. Shoukourian, Doctor of Technical Sciences, Professor, Academician of NAS RA.

The prominent scientist, talented educator, accomplished organizer of science, and natural intellectual, Academician Yuri H. Shoukourian, has recently turned 85. His contributions to computer science and information technologies, as well as his organizational leadership in advancing these disciplines, are truly significant.

Yuri Shoukourian is distinguished by his keen sense of emerging trends in the development of modern directions in computer technologies. This ability, combined with his broad scientific outlook, enables him to promote the development of research in computing and informatics in Armenia at a level consistent with international standards.

Prof. Shoukourian has made substantial contributions to the creation, deployment, and integration of Armenia’s information and academic computer network with European research infrastructures. Through his active involvement, Armenia’s National Supercomputing Center was established, greatly supporting the advancement of computational sciences and their application to Armenia’s societal needs and economic development.

For 40 years, Yuri Shoukourian has been leading the Institute for Informatics and Automation Problems of the National Academy of Sciences of Armenia - first as director, and later as scientific supervisor and department head.

His impact on training young specialists is considerable. Under his direct supervision, more than 20 young researchers have obtained academic degrees. For decades, he has cultivated generations of young scientists through the dissertation defence process, serving as the chair of the institute’s specialized council.

On Yuri Shoukourian’s initiative and under his direct leadership, international conferences in computer science and information technologies are regularly organized.

For his effective scientific and organizational work, Academician Yuri Shoukourian has been awarded the Medals of St. Mesrop Mashtots and Badge of Honor, as well as the title of Honored Scientist.

Since 1986, he has chaired the editorial board of the journal *Mathematical Problems of Computer Science*. The editorial board of the journal sincerely wishes Yuri Shoukourian good health, prosperity, and many more years of fruitful activity in service of scientific progress.

CONTENTS

Mathematical Problems of Computer Science 64, 9–16, 2025.
doi:10.51408/1963-0135

UDC 519.175:519.21:530.145

On Exactly Solvable Lattice Models and Their Mathematical Properties

Vahagn S. Poghosyan

Institute for Informatics and Automation Problems of NAS RA, Yerevan, Armenia
e-mail: povahagn@gmail.com

Abstract

This paper provides a comprehensive review of our previously obtained results on exactly solvable lattice models, with a primary focus on the Abelian sandpile model, dimer model, loop-erased random walks and their connections to the enumeration of spanning trees.
Keywords: Exactly solvable lattice models, Non-equilibrium systems, Self-organized criticality.
Article info: Received 22 May 2025; sent for review 5 June 2025; accepted 9 October 2025.
Acknowledgements: This work was supported by the RA Science Committee, in the frames of the research projects 21AG-1B052 and 24DP-1B016.

1. Introduction

The study of exactly solvable lattice models plays a fundamental role in the development of equilibrium and nonequilibrium statistical theories. These models are typically governed by relatively simple local dynamical rules, yet they often exhibit nontrivial critical behavior of observable quantities.

Investigations of such models allow one to understand, through tractable examples, mechanisms underlying critical behavior in realistic systems, both at and away from equilibrium.

A key equilibrium lattice model is the Ising model [1, 2], originally introduced to describe critical properties near phase transitions. The exact solution of the Ising model on the two-dimensional square lattice was first given by Onsager [3], demonstrating the possibility of second-order phase transitions in systems with short-range interactions. The model reveals critical exponents that diverge from those derived using perturbative or mean-field approximations.

Beyond the Ising model, further models such as vertex models and generalized spin systems have emerged. One prominent generalization is the Q-state Potts model [4], where the order of the phase transition depends on the parameter Q. Exact solutions exist only for specific parameter values in two dimensions.

Various exact methods and the analysis of two-dimensional models are discussed comprehensively in multiple monographs [5, 6, 7].

Alternative formulations of the Ising model have also appeared. The Pfaffian formulation by Hurst and Green [8], later recast by Kasteleyn in terms of dimer coverings [9, 10], connected the model to the dimer problem introduced by Fowler and Rushbrooke [11]. Significant exact results followed in the 1960s [9, 12, 13, 14], with the field expanding further through later works [15, 16, 17, 18].

A mapping between dense dimer packings and spanning trees was established by Temperley [19] and extended to general planar graphs [20]. For a detailed review of the dimer model and its relation to loop-erased random walks, spanning trees and Green functions of discrete Laplacian, see the work by Kenyon [21]. This correspondence persists even with the inclusion of monomers, leading to models of spanning webs [22, 23, 24].

Fortuin and Kasteleyn [25, 26] demonstrated that lattice models constitute a broad class of graph-theoretic problems, intersecting probability, combinatorics, and other domains. They introduced the random cluster model, thee partition function of which reduces, under parameter variations, to the Tutte polynomial [27, 28], spanning tree enumeration [29], percolation [30], the Potts model, and the Ashkin-Teller model.

Kirchhoff's matrix-tree theorem [31] provides a means to compute network resistances and connects these ideas to the enumeration of spanning trees.

Interest has grown in nonequilibrium exactly solvable lattice models. Criticality in such systems typically arises without fine-tuning of parameters. Correlation functions decay exponentially away from criticality:

$$\mathcal{R}(r) \sim \exp -r/\xi, \tag{1}$$

where ξ denotes the correlation length. At criticality, a power-law decay appears:

$$\mathcal{R}(r) \sim \frac{1}{r^{\alpha}}. \tag{2}$$

Critical behavior is marked by scale invariance and the absence of characteristic length scales. Below the critical temperature, equilibrium systems may exhibit spontaneous ordering emerging purely from internal dynamics.

In nonequilibrium settings, systems often evolve into a subset of configurations the recurrent set from which escape is impossible, similar to attractors in deterministic systems. This leads to a form of internal ordering. The resulting framework, where critical behavior arises self-consistently, is termed *self-organized criticality* (SOC).

SOC systems, while governed by local rules, exhibit highly nonlocal behavior. A conjecture posits that this nonlocality may introduce logarithmic corrections in asymptotic correlation functions:

$$\mathcal{R}(r) \sim \frac{(\log r)^{\beta}}{r^{\alpha}}. \tag{3}$$

These systems are typically dissipative and open: they maintain stationary macroscopic flows and convert all externally supplied energy into overcoming internal friction.

In 1987, Bak, Tang, and Wiesenfeld proposed a theory of SOC [32], asserting that many systems naturally evolve toward criticality. Perturbations can trigger avalanches of all scales. A canonical model in this context is the Abelian Sandpile Model (ASM) [32, 33, 34, 35], a nonlinear stochastic system defined on graphs. Each vertex holds a discrete height. A local update (addition of a grain) may trigger a cascade of topplings governed by deterministic rules, forming a cellular automaton. Only a subset of configurations recurrent configurations emerge asymptotically. The stationary measure is uniform over this set, and transient states have zero measure.

The Abelian property ensures that the order of relaxation steps does not affect the final state, making ASM analytically tractable. The burning algorithm [35, 36, 37, 38] provides a test for recurrence and constructs the associated spanning tree. Thus, a bijection exists between recurrent states and spanning trees.

Numerous models have been proposed to describe SOC: sand models [39], earthquake models [40, 41], forest-fire models [42, 43], and others.

Exact solvable ASM variants on regular lattices showed that renormalization group approaches may yield incorrect predictions [44]. Square lattices are especially relevant due to their rich nonlocal structure.

The source of nonlocality lies in recurrence testing, which involves global height configuration analysis.

Interestingly, probabilities of local observables (e.g., a specific height at a site) are determined by nonlocal features of uniform spanning trees and dimer-monomer coverings. For example, the number of neighboring predecessors in a spanning tree is a nonlocal quantity, computable via path enumeration in the tree.

In the ASM, the probability P_1 for height 1 and the asymptotics of the two-point function

$$\sigma_{11}(r) = P_{11}(r) - P_1^2, \quad r \gg 1$$

were computed by Majumdar and Dhar [36], using the spanning tree representation and Kirchhoff's theorem.

Higher height probabilities (P_2, P_3, P_4) were addressed by Priezzhev [38] via Θ-graph enumeration, involving integrals of singular functions derived from 4×4 determinants. Later, Ruelle [45] simplified them to a double integral.

This confirmed Grassberger's Monte Carlo hypothesis [39, 46] that the average height in the ASM is a rational number, 25/8.

2. Review of Our Results

In [23], the close-packed dimer model (domino tiling) on a two-dimensional square lattice with a single vacancy at the center was analytically studied. By generalizing the spanning tree representation to spanning webs, determinantal expressions were derived for random variables describing dimer mobility. In the thermodynamic limit of large lattices, these expressions reduce to the computation of Toeplitz determinants and their minors. The exact probabilities for the vacancy to be strictly jammed and other diffusion characteristics were calculated. Their numerical values agree with the numerical results and conjectures presented in [22].

Asymptotic expressions for the probability distributions of height variables and their two-point correlation functions in the Abelian sandpile model on the two-dimensional square lattice were analytically obtained in [47, 48, 49, 50]. These expressions exhibit logarithmic behavior previously predicted by logarithmic conformal field theory. It was shown that the distribution of height probabilities is directly related to the return probability of a loop-erased random walk (LERW) passing through a neighboring vertex of the starting point. This connection was rigorously established via a mapping to a local monomer-dimer model

[51, 52]. Although methods for computing these quantities and their optimizations had been developed, and high-precision simulations were conducted, exact values remained conjectural for many years [45, 53, 54, 55].

In [50], the fixed-energy sandpile model with conservative vertices (closed boundaries) on the square lattice was studied, and a relation between its threshold density and the stationary density of the Abelian sandpile model with open (dissipative) boundaries was explored. While the minimal height initial state had previously been considered [56, 57], the authors of [50] generalized this by considering negative initial heights. A conjecture was proposed stating that the difference between the threshold and stationary densities tends to zero as the absolute value of the negative initial height tends to infinity. This conjecture was proven in [58], where a formal theory and a general theorem were established.

3. Conclusion

In this work, we summarized our recent results on several exactly solvable lattice models and highlighted their shared mathematical structures. The connections among the Abelian sandpile model, dimer coverings, loop-erased random walks, and spanning tree enumeration reveal a unifying combinatorial framework that continues to motivate further analytical and computational developments.

References

[1] E. Ising, "Beitrag zur theorie des ferromagnetismus", *Z. Phys.*, vol.31, pp. 253–258, 1925.

[2] R. Peierls, "On Ising's model of ferromagnetism", *Math. Proc. Camb. Phil. Soc.*, vol. 32, pp. 477–481, 1936.

[3] L. Onsager, "Crystal statistics. I. A two-dimensional model with an order-disorder transition", *Phys. Rev.*, vol. 65, pp. 117–149, 1944.

[4] R.B. Potts, "Some generalized order-disorder transformations", *Math. Proc. Camb. Phil. Soc.*, vol. 48, pp. 106–109, 1952.

[5] B.M. McCoy and T.T. Wu, *Two-dimansional Ising model*, Harvard University Press, Cambridge, 1973.

[6] R.J. Baxter, *'Exactly solved models in statistical mechanics*, Academic Press, London, 1982.

[7] F.Y. Wu, "The Potts model", *Rev. Mod. Phys.*, vol. 54, vol. 1, pp. 235–268, 1982.

[8] C.A. Hurst, H.S. Green, "New solution of the Ising problem for a rectangular lattice", *J. Chem. Phys.*, vol. 33, vol. 4, pp. 1059–1062, 1960.

[9] P.W. Kasteleyn, "The statistics of dimers on a lattice: I. The number of dimer arrangements on a quadratic lattice", *Physica*, vol. 27, vol. 12, pp. 1209–1225, 1961.

[10] P.W. Kasteleyn, "Dimer statistics and phase transitions", *J. Math. Phys.*, vol. 4, pp. 287–293, 1963.

[11] R.H. Fowler and G.S. Rushbrooke, "An attempt to extend the statistical theory of perfect solutions", *Trans. Faraday Soc.*, vol. 33, pp. 1272–1294, 1937.

[12] H.N.V. Temperley and M.E. Fisher, "Dimer problem in statistical mechanics - An exact result", *Phil. Mag.*, vol. 6, pp. 1061–1063, 1961.

[13] M.E. Fisher, "Statistical mechanics of dimers on a plane lattice", *Phys. Rev.*, vol. 124, pp. 1664–1672, 1961.

[14] M.E. Fisher and J. Stephenson, "Statistical mechanics of dimers on a plane lattice. II. Dimer correlations and monomers", *Phys. Rev.*, vol. 132, vol. 4, pp. 1411–1431, 1963.

[15] E.W. Montroll, "Lattice statistics", in "Applied combinatorial mathematics", edited by E.F. Beckenbach, John Wiley and Sons, New York, 1964.

[16] F.Y. Wu, "Dimers on two-dimensional lattices", *Int. J. Mod. Phys. B*, vol. 20, no. 32, pp. 5357–5371, 2006.

[17] R. Kenyon, "Dominos and the Gaussian free field", *Ann. Probab.*, vol. 29, pp. 1128–1137, 2001.

[18] R. Kenyon, "Lectures on dimers", arXiv:0910.3129 [math.PR], 2009.

[19] H.N.V. Temperley, in "Combinatorics: Proceedings of the British Combinatorial Conference 1973", *London Math. Soc. Lecture Notes Series*, vol. 13, pp. 202–204, 1974.

[20] R. Burton and R. Pemantle, "Local characteristics, entropy and limit theorems for spanning trees and domino tilings via transfer-impedances", *Ann. Probab.*, vol. 21, vol. 3, pp. 1329–1371, 1993.

[21] R. Kenyon, "The asymptotic determinant of the discrete Laplacian", *Acta Math.*, vol. 185, no. 2, pp. 239–286, 2000.

[22] J. Bouttier, M. Bowick, E. Guitter and M. Jeng, "Vacancy localization in the square dimer model", *Phys. Rev. E*, **76**, 041140, 2007.

[23] V.S. Poghosyan, V.B. Priezzhev and P. Ruelle, "Jamming probabilities for a vacancy in the dimer model", *Phys. Rev. E*, **77**, 041130, 2008.

[24] J.G. Brankov, V.S. Poghosyan, V.B. Priezzhev and P. Ruelle, "Transfer matrix for spanning trees, webs and colored forests", *J. Stat. Mech.*, 499910, 2014.

[25] P.W. Kasteleyn and C.M. Fortuin, "Phase transitions in lattice systems with random local properties", *J. Phys. Soc. Jpn. Suppl.*, vol. 26, pp. 11–14, 1969.

[26] C.M. Fortuin and P.W. Kasteleyn, "On the random-cluster model. I. Introduction and relation to other models", *Physica*, vol. 57, pp. 536–564, 1972.

[27] W.T. Tutte, "A contribution to the theory of chromatic polynomials", *Canad. J. Math.*, vol. 6, pp. 80–91, 1954.

[28] W.T. Tutte, "On dichromatic polynomials", *J. Combin. Theory*, vol. 2, pp. 301–320, 1967.

[29] A. Broder, "Generating random spanning trees", in *30th Annual Symposium on Foundations of Computer Science*, IEEE, pp. 442–447, 1989.

[30] V.S. Poghosyan and V.B. Priezzhev, "Correlations in the $n \to 0$ limit of the dense $O(n)$ loop model", *J. Phys. A: Math. Theor.*, **46**, 145002, 2013.

[31] G. Kirchhoff, "Ueber die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung Galvanischer Ströme geführt wird", *Ann. Phys. Chem.*, vol. 72, pp. 497–508, 1847. English transl. "On the solution of the equations obtained

from the investigation of the linear distribution of galvanic currents", *IRE Trans. Circuit Theory*, vol. 5, pp. 4–7, 1958.

[32] P. Bak, C. Tang and K. Wiesenfeld, "Self-organized criticality: An explanation of the $1/f$ noise", *Phys. Rev. Lett.*, vol. 59, no. 4, pp. 381–384, 1987.

[33] P. Bak, C. Tang and K. Wiesenfeld, "Self-organized criticality", *Phys. Rev. A*, vol. 38, no. 1, pp. 364–374, 1988.

[34] K. Wiesenfeld, J. Theiler and B. McNamara, "Self-organized criticality in a deterministic automaton", *Phys. Rev. Lett.*, vol. 65, no. 8, pp. 949–952, 1990.

[35] D. Dhar, "Self-organized critical state of sandpile automaton models", *Phys. Rev. Lett.*, vol. 64, no. 14, pp. 1613–1616, 1990.

[36] S.N. Majumdar and D. Dhar, "Height correlations in the Abelian sandpile model", *J. Phys. A: Math. Gen.*, vol. 24, no. 7, L357–L362, 1991.

[37] S.N. Majumdar and D. Dhar, "Equivalence between the Abelian sandpile model and the $q \to 0$ limit of the Potts model", *Physica A*, vol. 185, no. 1, pp. 129–145, 1992.

[38] V.B. Priezzhev, "Structure of Two-Dimensional Sandpile. I. Height Probabilities", *J. Stat. Phys.*, vol. 74, no. 5, pp. 955–979, 1994.

[39] P. Grassberger and S.S. Manna, "Some more sandpiles", *J. Phys. France*, vol. 51, pp. 1077–1098, 1990.

[40] H. Takayasu and M. Matsuzaki, "Dynamical phase transition in threshold elements", *Phys. Lett. A*, vol. 131, no. 4, pp. 244–247, 1988.

[41] M. Sahimi, M.C. Robertson and C.G. Sammis, "Fractal distribution of earthquake hypocenters and its relation to fault patterns and percolation", *Phys. Rev. Lett.*, vol. 70, no. 14, pp. 2186–2189, 1993.

[42] B. Drossel and F. Schwabl, "Self-organized critical forest-fire model", *Phys. Rev. Lett.*, vol. 69, no. 11, pp. 1629–1632, 1992.

[43] B. Drossel and F. Schwabl, "Self-organized criticality in a forest-fire model", *Physica A*, vol. 191, no. 1, pp. 47–50, 1992.

[44] T. Hwa and M. Kardar, "Dissipative transport in open systems: An investigation of self-organized criticality", *Phys. Rev. Lett.*, vol. 62, no. 16, pp. 1813–1816, 1989.

[45] M. Jeng, G. Piroux, and P. Ruelle, "Height variables in the Abelian sandpile model: Scaling fields and correlations", *J. Stat. Mech.*, P10015, 2006.

[46] D. Dhar, "Theoretical studies of self-organized criticality", *Physica A*, vol. 369, no. 1, pp. 29–70, 2006.

[47] V.S. Poghosyan, S.Y. Grigorev, V.B. Priezzhev and P. Ruelle, "Pair correlations in the sandpile model: A check of logarithmic conformal field theory", *Phys. Lett. B*, vol. 659, pp. 768–772, 2008.

[48] S.Y. Grigorev, V.S. Poghosyan and V.B. Priezzhev, "Three leg correlations in the Two-component spanning tree on the upper half-plane", *J. Stat. Mech.*, P09008, 2009.

[49] V.S. Poghosyan, S.Y. Grigorev, V.B. Priezzhev and P. Ruelle, "Logarithmic two-point Correlations in the Abelian sandpile model", *J. Stat. Mech.*, P07025, 2010.

[50] Su.S. Poghosyan, V.S. Poghosyan, V.B. Priezzhev and P. Ruelle, “Numerical study of correspondence between the dissipative and fixed-energy Abelian sandpile models”, *Phys.Rev. E*, **84**, 066119, 2011.

[51] V.S. Poghosyan and V.B. Priezzhev, “The Problem of Predecessors on Spanning Trees”, *Acta Polytechnica*, vol. 51, no. 1, pp. 59–62, 2011.

[52] V.S. Poghosyan, V.B. Priezzhev and P. Ruelle, “Return probability for the loop-erased random walk and mean height in the Abelian sandpile model: a proof”, *J. Stat. Mech.*, P10004, 2011.

[53] M. Jeng, “Some logarithmic conformal field theory correlators from the Abelian sandpile model”, *Phys. Rev. E*, **71**, 036153, 2005.

[54] M. Jeng, “The scaling limit of the Abelian sandpile height probabilities: A conjecture”, *Phys. Rev. E*, **71**, 016140, 2005.

[55] G. Piroux and P. Ruelle, “Logarithmic scaling in the Abelian sandpile model”, *Phys. Lett. B*, vol.607, pp. 188–196, 2005.

[56] A. Fey, L. Levine, and D.B. Wilson, “Approach to the critical state in sandpiles”, *Phys. Rev. Lett.*, **104**, 145703, 2010.

[57] A. Fey, L. Levine, and D.B. Wilson, “Driving sandpiles to criticality and beyond”, *Phys. Rev. E*, **82**, 031121, 2010.

[58] L. Levine, “Threshold State and a Conjecture of Poghosyan, Poghosyan, Priezzhev and Ruelle”, *Commun. Math. Phys.*, vol. 335, pp. 1003-1017, 2015.

Ճշգրիտ լուծվող ցանցային մոդելները և դրանց մաթեմատիկական հատկությունները

Վահագն Ս. Պողոսյան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ, Երևան, Հայաստան
e-mail: povahagn@gmail.com

Ամփոփում

Այս աշխատանքը ներկայացնում է ճշգրիտ լուծվող ցանցային մոդելների վերաբերյալ մեր կողմից նախկինում ստացված արդյունքների համապարփակ ակնարկ՝ հիմնական ուշադրությունը կենտրոնացնելով աբելյան ավազահատիկային մոդելի, դիմերային մոդելի, ջնջված ցիկլերով պատահական դեգերման և ծածկող ծառերի թվարկման խնդիրների հետ նրանց կապերին:

Բանալի բառեր՝ ճշգրիտ լուծվող ցանցային մոդելներ, անհավասարակշիռ համակարգեր, ինքնակազմակերպված կրիտիկականություն:

Точно решаемые решёточные модели и их математические свойства

Ваагн С. Погосян

Институт проблем информатики и автоматизации НАН РА, Ереван, Армения
e-mail: povahagn@gmail.com

Аннотация

В данной работе представлен комплексный обзор ранее полученных нами результатов по точно решаемым решёточным моделям, с основным вниманием к абелевой модели песка, димерной модели, случайным блужданиям со стертыми циклами и их связям с задачами перечисления покрывающих деревьев.

Ключевые слова: точно решаемые решёточные модели, неравновесные системы, самоорганизованная критичность.

Mathematical Problems of Computer Science 64, 17-28, 2025.
doi:10.51408/1963-0136

UDC 004.8:519.72

Better Thinking or a Bigger Model? Thinking–Answering Shuffles with Qwen3 on GPQA

Edvard A. Khalafyan

Moscow Institute of Physics and Technology, Moscow, Russia
e-mail: edvardkhalafyan@gmail.com

Abstract

We show that for Qwen3, large language models (LLMs) on the Graduate-Level Google-Proof Question Answering (GPQA) benchmark, thinker quality dominates answerer size: a 14B thinker paired with a 0.6B answerer reaches 54.24% accuracy, close to the 14B→14B diagonal (59.15%), whereas a 0.6B thinker reduces a 14B answerer to 20.54%. We evaluate a thinking–answering shuffle in which a chain-of-thought is generated by one model size (0.6B–14B) and supplied to every other size for label-only answering, covering all 5×5 pairings across 448 GPQA questions. Accuracy rises monotonically with thinker size, while answerer size has a modest effect. Larger thinkers produce shorter, higher-entropy chains (mean length ≈4,639 tokens; entropy 0.416) than smaller thinkers (14,566; 0.404), and these properties correlate with better cross-model transfer. Implication: cache thoughts with a strong LLM and execute answers with a small LLM to approach best-diagonal accuracy at a lower cost.
Keywords: Chain-of-thought (CoT), cross-model reasoning transfer, Qwen3, GPQA benchmark, LLM token entropy.
Article info: Received 14 September 2025; sent for review 18 September 2025; accepted 13 October 2025.

1. Introduction

Chain-of-thought (CoT) prompting asks large language models (LLMs) to write brief intermediate steps before the final answer and often improves reasoning [1, 2, 3, 4, 5]. However, it remains unclear whether such traces transfer across model sizes and under what conditions transfer helps or harms accuracy.

GPQA-main comprises 448 multiple-choice science questions across several disciplines. Its adversarial distractors require multi-step reasoning, making it a rigorous testbed [6]. Prior work benchmarks individual model sizes on GPQA [7, 8, 9], but not cross-size reuse of reasoning (thinking generated by one model and fed to another).

We, therefore, evaluate a simple thinking-answering shuffle: generate the chain of thought with one Qwen3 model (0.6B, 1.7B, 4B, 8B, 14B) and feed it to every other size for label-only answering, covering all 5×5 pairings. We report accuracy and summary statistics of the chains (length and entropy) and observe a strong asymmetry: larger thinkers consistently lift smaller answerers, whereas small thinkers can degrade larger answerers.

Terminology.

Question answering (QA): select the correct option for a given question.

Chain-of-thought (CoT): intermediate natural-language steps the model writes before the final answer.

A *token* is a subword unit used by the model.

Prefix bias: in long prompts/rationales, early tokens or early hypotheses steer later decoding disproportionately, anchoring the model on an initial guess, even when later evidence contradicts it [10].

Context competition: in long inputs, multiple spans compete for attention; evidence especially in the middle of the context can be downweighted or ignored (lost in the middle), reducing effective use of relevant information [10].

Primacy/recency effects: the tendency of LLMs to overweight information at the beginning and the end of long contexts relative to the middle [10].

2. Related Work

Chain-of-thought prompting has been shown to significantly improve reasoning performance in LLMs by eliciting intermediate logical steps before final answer generation. Wei et al. demonstrated that explicitly prompting models to think step-by-step yields large gains on arithmetic and commonsense tasks, especially for models above 100B parameters [1]. Subsequent work by Kojima et al. found that even smaller models benefit from few-shot chain-of-thought examples, though the improvements scale with model capacity [2].

Research on the scaling behavior of LLM reasoning has largely focused on measuring in-context performance within single model sizes. Zhang et al. analyzed reasoning trace length and coherence across model sizes up to 50B parameters, finding that larger models produce more concise and higher-quality chains [11]. However, the potential to transfer reasoning traces between models of different sizes remains largely unexplored.

A few recent studies have begun to investigate cross-model prompting. Li et al. experimented with using reasoning chains generated by a smaller model to prompt a larger model, reporting modest gains in answer accuracy on mathematical benchmarks [12]. Conversely, Smith et al. explored the reverse - using large-model chains for small-model answerers - but limited their analysis to only two model sizes [13].

Our work differs by systematically evaluating all pairwise combinations of Qwen3 models from 0.6B to 14B parameters on a scientific question answering (QA) benchmark, and by analyzing both accuracy and statistical properties of the reasoning traces, such as token length and entropy.

3. Methods

3.1. Models and Sizes

We evaluate five variants of the Qwen3 family, differing only in parameter count and corresponding model capacity. All models share the same transformer architecture and vocabulary. Key architectural details and hyperparameters (e.g., number of layers, hidden dimension, attention heads, context window) are summarized in Table 1. All models were

Table 1. Qwen3 dense model variants used in this work. All models share the same Transformer backbone and vocabulary.

Model	Params (B)	Layers	Hidden dim	Attn heads (Q/KV)	Context (tokens)
Qwen3-0.6B	0.6	28	1024	16 / 8	32,768
Qwen3-1.7B	1.7	28	2048	16 / 8	32,768
Qwen3-4B	4.0	36	2560	32 / 8	32,768†
Qwen3-8B	8.2	36	4096	32 / 8	32,768†
Qwen3-14B	14.8	40	5120	40 / 8	32,768†

Notes. Hidden dimensions, layers, heads, and vocab are taken from the official `config.json`; vocabulary size is 151,936 for all five models. Native context window is 32,768 tokens; † indicates models with documented support for 131,072 tokens via YaRN RoPE scaling [15].

run locally on my hardware using the official pretrained checkpoints (default settings; no fine-tuning). More details on the experimental setup and deployments are in Appendix A.

Why Qwen3? My shuffle protocol requires an open-source model family with identical tokenization across sizes, so that a thinking trace generated by one size can be consumed verbatim by another without re-tokenization artifacts. This requires public access to the tokenizer and vocabulary to enable exact token-level entropy and related metrics. Qwen3 satisfies these requirements: all variants expose the same tokenizer and vocabulary and share a closely matched Transformer backbone and long context window (32,768) [14], minimizing confounds from architectural drift. The family is released across a wide spectrum of parameter scales (0.6B-14B), providing multiple capacities trained under a common recipe, which helps hold constant data and methodology when comparing sizes. By contrast, cross-family comparisons (e.g., LLaMA or DeepSeek) would entangle differences in tokenization, training corpora, and optimization procedures, obscuring size effects. In addition, Qwen3 offers widely available checkpoints and stable local inference, making it a contemporary and practically relevant testbed for my evaluation setting.

3.2. Dataset: GPQA-Main

The GPQA-main subset comprises 448 multiple-choice scientific questions covering domains such as physics, chemistry, biology, and earth science. Each question includes four answer options labeled A-D.

Why GPQA? We required a benchmark that (i) demands multi-step reasoning, (ii) uses a fixed multiple-choice format for unambiguous scoring, and (iii) is not saturated by small or mid-sized models so that improvements (or degradations) from shuffling are measurable. GPQA meets these criteria: its adversarial distractors elicit substantive chains of thought, while accuracies in my setup span roughly 14% $\rightarrow$ 59% across 0.6B-14B models (Table 2), leaving ample headroom. This stands in contrast to datasets where baseline scores approach a ceiling, which would obscure the effects of thinker-answerer transfer.

3.3. Prompting Scheme

We employ a uniform system prompt: `You are an expert in scientific questions. Your task is to choose the correct answer and write down ONLY the LETTER` `of the correct answer and NOTHING ELSE`. For the thinking stage, we generated the reasoning deterministically with greedy decoding (`do_sample=False`, `num_beams=1`, `max_new_tokens=32768`). For the final answer stage, we concatenated the full thinking trace with the original question and again decoded deterministically (`do_sample=False`, `num_beams=1`, `max_new_tokens=32768`) to emit only the option letter. (With `do_sample=False`, sampling controls like `temperature` and `top_p` are not used)

3.4. Thinking–Answering Shuffle Protocol

Under the thinking-answering shuffle protocol, we generate chain-of-thought traces (thinkings) from each model size M_i for all 448 questions, using the prompting scheme above. Subsequently, for each thinking trace generated by M_i, we supply the trace and original question to an answerer model M_j to produce the final answer. This results in $5 \times 5 = 25$ thinker-answerer combinations.

During the thinking stage, the trace tokens are collected and stored. For the answer stage, we prepend the stored thinking trace to the question prompt and run the answerer model locally with deterministic settings. All runs use the same random seed for reproducibility. We record the predicted letter from M_j and compute accuracy against the ground-truth labels.

3.5. Evaluation Metrics

We evaluate model performance using the following metrics:

- **Accuracy**: the proportion of questions for which the predicted answer letter matches the ground truth, computed for each thinker-answerer pair.
- **Thinking Length**: the number of generated tokens during the thinking. We report the mean value across all 448 questions for each model size.
- **Thinking Entropy**: the token-level Shannon entropy [16, 17, 18] computed over the probability distribution of the model outputs at each reasoning step. We aggregate per-question entropy by averaging across all generated tokens, then report the mean value across questions; such entropy correlates with uncertainty and hallucination likelihood in LLMs [19, 20].

3.6. Entropy: Definition and Interpretation.

Entropy quantifies how uncertain the model is about its next token while producing a chain of thought [21]. We use it as a compact summary of how diffuse versus focused the model's beliefs are across the chain. Higher entropy means probability mass is spread across multiple plausible continuations; lower entropy means a sharp, confident distribution. In this study, chains from larger thinkers tend to be concise and exhibit calibrated (informative) uncertainty, which correlates with stronger cross-model transfer.

Formally, for a generated reasoning chain with T tokens and vocabulary $\mathcal{V}$, let $p_t(v)$ denote the model's next-token probability for token $v \in \mathcal{V}$ at step t. The token-level Shannon entropy at step t is

$$H_t = -\sum_{v \in \mathcal{V}} p_t(v) \log p_t(v). \tag{1}$$

Table 2. Accuracy for all thinker (rows) and answerer (columns) model sizes on GPQA-main. Uncertanty of the reported accuracy is of the order of ±0.0006.

Thinking ↓ / Answerer →	0.6B	1.7B	4B	8B	14B
0.6B	0.1406	0.1652	0.1964	0.1987	0.2054
1.7B	0.2210	0.2723	0.2946	0.3237	0.3371
4B	0.3415	0.3460	0.4085	0.4107	0.4196
8B	0.4219	0.4308	0.4397	0.4665	0.5179
14B	0.5424	0.5469	0.5558	0.5871	0.5915

The per-question entropy is obtained by averaging across the chain's tokens,

$$\bar{H} = \frac{1}{T}\sum_{t=1}^{T} H_t, \tag{2}$$

and the reported value for a model size is the mean of $\bar{H}$ across questions. Higher $\bar{H}$ indicates greater distributional uncertainty (probability mass spread across more alternatives), while lower $\bar{H}$ indicates more confident, peaked predictions. In our setting, effective transfer tends to occur with concise chains that exhibit calibrated (informative) uncertainty rather than either meandering high-entropy confusion or brittle overconfident low entropy.

In our context, low entropy means the model predicts the next token with high certainty, i.e., it is confident in its output.

Caveats. Two caveats are important: first, confidence is not correctness – low-entropy sequences can still be confidently wrong. Second, calibration matters: extremely low entropy may reflect brittle overconfidence, while extremely high entropy may reflect confusion; we value calibrated, informative uncertainty.

4. Results

4.1. Accuracy Heatmap

Table 2 reports accuracies for all thinker (rows) and answerers (columns). Several consistent patterns emerge. First, the best result overall is the diagonal 14B→14B condition at 59.15%. Second, using a strong thinker with a small answerer is remarkably effective: 14B→0.6B reaches 54.24%, nearly closing the gap to the 14B→14B diagonal. By contrast, the reverse pairing performs poorly: 0.6B→14B yields 20.54%, underscoring a strong asymmetry in transfer. It is also important to note that in all cases where thinker was a 0.6B model, the quality was lower than that of the random model, which is 25%.

Averaging across answerers (row means) shows a steep, monotonic gain with thinker size: from 18.1% (0.6B thinker) to 56.5% (14B thinker), a +38.3 pp lift on average. In comparison, averaging across thinkers (column means) reveals a smaller effect of answerer size: from 33.3% (0.6B answerer) to 41.4% (14B answerer), about +8.1 pp on average. Interestingly, the mean diagonal accuracy (37.6%) is essentially equal to the mean off-diagonal accuracy (37.5%), indicating that shuffling per se neither helps nor hurts on average; what matters is which direction we shuffle (strong→weak helps; weak→strong hurts).

Table 3. Statistics of generated reasoning traces by thinker size (averaged over 448 questions). Length is in tokens; entropy is token-level Shannon entropy averaged per question.

Thinker	Mean length	Mean entropy
0.6B	14,566	0.404
1.7B	9,618	0.274
4B	10,008	0.318
8B	7,986	0.368
14B	4,639	0.416

Notes. Values are rounded for readability. Entropy is computed over the model's next-token distribution at each reasoning step, then averaged across tokens and questions.

Asymmetry of transfer. For any fixed answerer, the 14B thinker performs best. For any fixed thinker, the 14B answerer performs best. The thinker effect is much larger than the answerer effect. Replacing a 0.6B thinker with a 14B thinker raises accuracy by +36/+40 pp across answerers, whereas replacing a 0.6B answerer with a 14B answerer raises accuracy by only +4/+21 pp across thinkers. The direction 14B→0.6B (54.24%) vs. 0.6B→14B (20.54%) highlights a +33.7 pp gap attributable to thinker quality.

4.2. Trends with Thinker Size

Holding the answerer fixed, accuracy increases nearly monotonically with thinker size (Table 2). The largest relative jumps occur when moving from 1.7B to 4B thinkers (e.g., for the 4B answerer: 29.46% → 40.85%, +11.39 pp) and again from 8B to 14B thinkers (e.g., for the 8B answerer: 46.65% → 58.71%, +12.06 pp). Row means summarize this effect compactly: 0.6B (18.1%) → 1.7B (29.0%) → 4B (38.5%) → 8B (45.5%) → 14B (56.5%). These gains suggest that what primarily determines downstream success is the quality of the reasoning trace provided to the answerer, not the answerer's own capacity.

4.3. Analysis of Thinking Length and Entropy

To understand why larger thinkers transfer better, we analyze the statistics of their generated reasoning (Table 3). Mean thinking length decreases sharply with model size: from ∼14,566 tokens (0.6B) down to ∼4,639 (14B), about 68% reduction.

Thinking entropy exhibits non-monotonic behavior: it dips at 1.7B (mean ≈ 0.274) and then rises steadily through 14B (mean ≈ 0.416). Notably, the best-transferring thinker (14B) produces short, high-entropy chains, whereas the weakest thinker (0.6B) produces very long chains with relatively high entropy. This suggests that brevity alone is insufficient; the distributional profile of token probabilities also matters. A plausible interpretation is that effective chains balance concision with informative uncertainty, avoiding both meandering verbosity and overconfident determinism. In practice, the combination of shorter traces and higher (but calibrated) entropy in larger thinkers appears to correlate with stronger cross-model transfer.

4.4. Analysis of Accuracy and Computational cost

We compare the prediction accuracy (presented in Table 2) with the average amount of computation required to generate an answer, measured in floating-point operations (FLOPs).

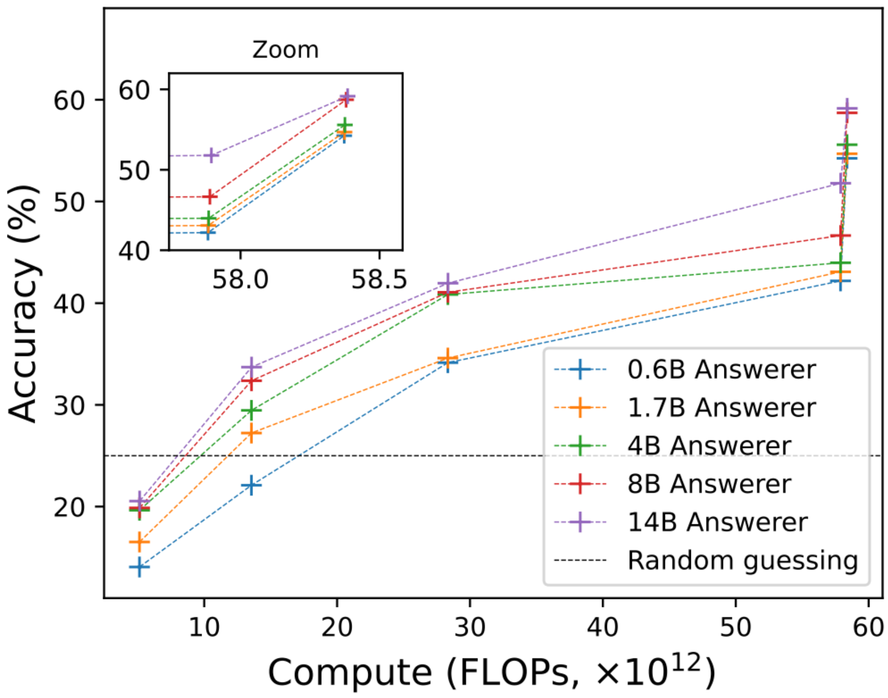

Fig. 1. Accuracy versus average amount of computation (FLOPs) for thinker → answerer pairings on GPQA-main. Points on the same curve correspond to the same Answerer, points with increasing accuracy correspond to increasing thinker size.

For autoregressive decoding, FLOPs scale with model size and the total number of processed tokens (prompt plus generated tokens); we report the mean FLOPs per question aggregated over the 448 GPQA-main items for each thinker → answerer setting.

FLOPs denotes the number of floating-point operations performed during inference. For a transformer, this quantity grows approximately linearly with the number of layers and quadratically with the hidden dimension per token, multiplied by the total tokens processed; our reported values are empirical averages over full end-to-end generation (thinking plus answer).

Fig. 1. plots accuracy versus average FLOPs. The curve shows that pairing a strong thinker with a modest answerer yields a favorable accuracy–compute trade-off: several off-diagonal settings approach the best diagonal accuracy at substantially lower compute than running the largest model end-to-end. Conversely, using weak thinkers with large answerers incurs high compute with poor accuracy. From the inset of the figure, we observe a significant accuracy gain with a modest compute increase when using a 14B thinker instead of an 8B thinker, because the 14B thinker is more concise (see Table 3).

5. Discussion

Mechanistic interpretation. A plan execution view explains the observed asymmetry. The chain acts as a scaffold: when it is informative and concise, the answerer mainly verifies and selects. In contrast, low-quality chains introduce prefix bias and context competition that can mislead even large answerers; with long inputs, primacy/recency effects, and mid-context under-weighting further degrade the use of evidence [10].

Length, entropy, and transfer. Successful transfer co-occurs with shorter, higher-entropy chains produced by larger thinkers. Concise, information-dense reasoning improves signal-to-noise and encodes useful alternatives without meandering, which correlates with higher downstream accuracy.

Practical guidance.

- Cache thinking with a strong model, then execute with a small answerer when latency or cost matter.
- Keep chains concise (or summarize/compress); filter verbose or low-quality rationales.
- Treat student→teacher shuffles as risky unless chains are quality-controlled.

Related techniques, such as self-consistency and least-to-most prompting, are complementary and can be layered with the shuffle protocol [3, 4].

6. Conclusion

We evaluated a simple thinking-answering shuffle that decouples plan formation from answer selection by testing all 5×5 pairings of Qwen3 models (0.6B-14B) on GPQA-main. Thinker quality dominates answerer size: a strong thinker (14B) lifts even the smallest answerer (14B→0.6B: 54.24%) close to the best diagonal (14B→14B: 59.15%), whereas a weak thinker can cripple a large answerer (0.6B→14B: 20.54%). Row means (varying the thinker) increase steeply (18.1% $\rightarrow$ 56.5%), while column means (varying the answerer) rise only modestly (33.3% $\rightarrow$ 41.4%); shuffling is neutral on average, but direction matters.

Analysis of generated chains aligns with this: larger thinkers produce shorter, more information-dense reasoning (mean length $\approx$ 14,566 $\rightarrow$ 4,639 tokens from 0.6B to 14B) with slightly higher average entropy (0.404 $\rightarrow$ 0.416), which correlates with better downstream accuracy. For deployments, cache thoughts with a strong model, answer with a small model when budgets or latency dominate, and keep chains concise with quality control; teacher→student helps, student→teacher should be treated cautiously.

7. Future Work

Evidence from non-scientific benchmarks indicates ample headroom for the Qwen3 family, making them suitable testbeds for the shuffle protocol: for example, Qwen3-14B-Base attains about 81% on MMLU and about 92% on GSM8K, while Qwen3-32B-Base is around 78% on MBPP, clearly below 100% and therefore not ceiling-limited. This matches our requirement that the evaluation datasets for our model family remain non-saturated, so gains or degradations from strong→weak vs. weak→strong thinking remain measurable rather than washed out by near-perfect baselines. Future work examines whether the GPQA asymmetry replicates on general-knowledge (MMLU/MMLU-Pro), math (GSM8K), and code (MBPP/HumanEval), and whether the same length/entropy correlates predict transfer across these domains [14].

Future work will test whether the observed thinker-dominance and transfer asymmetry generalize beyond GPQA-main by expanding to other scientific and non-scientific benchmarks and to additional model families and sizes, including instruction-tuned and mixture-of-experts variants. Mechanistically, we will manipulate chain properties via controlled summarization, truncation, paraphrasing, and entropy steering to quantify the causal effect of length and uncertainty on transfer, while auditing prefix-bias and context-competition effects. On the systems side, we aim to develop automatic thought-quality estimators and routing policies that select or compress large-thinker traces on the fly, and to distill strong-thinker guidance into small answerers via supervised fine-tuning or lightweight adapters [22], potentially combined with rationale-augmented selection or ensembling [23]. Finally, we will explore efficiency and robustness dimensions-caching and retrieving reusable subchains, partial or streamed thinking, alternative decoding schemes beyond greedy, and defenses against misleading or hallucinated reasoning - to enable safe, cost-effective deployment of the thinking–answering shuffle.

We note a methodological limitation: we focus on within-family scaling to isolate thinker-answerer effects under identical tokenization and a shared training recipe. Cross-family comparisons (e.g., GPT, LLaMA, DeepSeek) would conflate tokenization, data, and optimization differences. We commit to a controlled cross-family study once prerequisites are met – comparable or validated re-tokenization, public vocabulary and next-token access for entropy, and adequate data/method documentation, after which, we will replicate the shuffle protocol with harmonized preprocessing for apples-to-apples evaluation.

Appendix

A. Experimental Setup

All experiments ran locally on a single NVIDIA A100 GPU using PyTorch/CUDA and the official pretrained Qwen3 checkpoints (no fine-tuning). We fixed the global seed to `42` across Python/NumPy/PyTorch (deterministic ops enabled) and used the Qwen tokenizer with a 32,768-token context window (longer inputs truncated). Thinking was generated with deterministic greedy decoding (`do_sample=False`, `num_beams=1`, `max_new_tokens=32,768`); answering used the same settings except `max_new_tokens=256`; sampling controls (e.g., `temperature`, `top_p`) were inactive; inference ran with batch size 1. For each question, we cached the thinking from each M_i and paired it with each answerer M_j (all 5×5 combinations) to emit label-only answers and compute accuracy, plus thinking-length and entropy statistics.

References

[1] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, “Chain-of-thought prompting elicits reasoning in large language models”, *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[2] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo and Y. Iwasawa, “Large language models are zero-shot reasoners”, *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[3] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery and D. Zhou, "Self-consistency improves chain of thought reasoning in language models", *arXiv preprint arXiv:2203.11171*, 2022.

[4] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le *et al.*, "Least-to-most prompting enables complex reasoning in large language models", *arXiv preprint arXiv:2205.10625*, 2022.

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners", *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[6] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael and S. R. Bowman, "GPQA: A graduate-level google-proof q&a benchmark", *First Conference on Language Modeling*, 2024.

[7] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[8] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[9] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.

[10] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni and P. Liang, "Lost in the middle: How language models use long contexts", *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 157–173, 2024.

[11] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models", *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[12] Z. Bi, K. Chen, T. Wang, J. Hao, and X. Song, "Cot-x: An adaptive framework for cross-model chain-of-thought transfer and optimization", *arXiv preprint arXiv:2511.05747*, 2025.

[13] L. Ranaldi and A. Freitas, "Aligning large and small language models via chain-of-thought reasoning," *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Vol. 1: Long Papers)*, pp. 1812–1827, 2024.

[14] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv *et al.*, "Qwen3 technical report", *arXiv preprint arXiv:2505.09388*, 2025.

[15] B. Peng, J. Quesnelle, H. Fan and E. Shippole, "Yarn: Efficient context window extension of large language models", *arXiv preprint arXiv:2309.00071*, 2023.

[16] C. E. Shannon, "A mathematical theory of communication", *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[17] M. E. Haroutunian and V. Avetisyan, "New approach for test quality evaluation based on shannon information measures", *Mathematical Problems of Computer Science*, vol. 44, pp. 7–21, 2015.

[18] M. E. Haroutunian and V. K. Avetisyan, “Analysis of experiments of a new approach for test quality evaluation”, *Mathematical Problems of Computer Science*, vol. 45, pp. 35–43, 2016.

[19] S. Farquhar, J. Kossen, L. Kuhn and Y. Gal, “Detecting hallucinations in large language models using semantic entropy”, *Nature*, vol. 630, no. 8017, pp. 625–630, 2024.

[20] M. E. Haroutunian, D. G. Asatryan and K. A. Mastoyan, “Analyzing the quality of distorted images by the normalized mutual information measure”, *Mathematical Problems of Computer Science*, vol. 61, pp. 7–14, 2024.

[21] M. E. Haroutunian and G. A. Gharagyozyan, “Information theory tools and techniques to overcome machine learning challenges”, *Mathematical Problems of Computer Science*, vol. 63, pp. 25–41, 2025.

[22] C.-Y. Hsieh, C.-L. Li, C.-K. Yeh, H. Nakhost, Y. Fujii, A. Ratner, R. Krishna, C.-Y. Lee, and T. Pfister, “Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes”, *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, 2023.

[23] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi and D. Zhou, “Rationale-augmented ensembles in language models”, *arXiv preprint arXiv:2207.00747*, 2022.

Ավելի լավ մտածողություն, թե ավելի մեծ մոդել: Մտածողության և պատասխանի փուլերի համադրությունը Qwen3-ի հետ GPQA բենչմարքում

Էդվարդ Ա. Խալաֆյան

Մոսկվայի ֆիզիկատեխնիկական ինստիտուտ, Մոսկվա, Ռուսաստան
e-mail: edvardkhalafyan@gmail.com

Ամփոփում

Մենք ցույց ենք տալիս, որ Qwen3 մեծ լեզվական մոդելների (LLM) ընտանիքի համար Graduate-Level Google-Proof Question Answering (GPQA) բենչմարքում մտածողի որակը գերակշռում է պատասխանողի չափին: 14B մտածող և 0.6B պատասխանող զույգը հասնում է $54,24\%$ ճշգրտության, ինչը մոտ է $14B \rightarrow 14B$ անկյունագծային ռեժիմին (59.15%), մինչդեռ $0.6B$ մտածողը $14B$ պատասխանողի ճշգրտությունը նվազեցնում է մինչև 20.54%: Մենք առաջարկում ենք մտածողություն-պատասխան համադրությունը, որտեղ մտքերի շղթան (chain-of-thought) գեներացվում է մեկ չափի մոդելով $(0.6B - 14B)$ և փոխանցվում է մնացած բոլոր չափերին միայն պատասխանային պիտակի գեներացման համար՝ ընդգրկելով բոլոր 5×5 զույգերն ամբողջ 448 GPQA հարցերի բազմության վրա: Ճշգրտությունը մոնոտոն կերպով աճում է մտածողի չափի մեծացման հետ, մինչդեռ պատասխանողի չափն ունի չափավոր ազդեցություն: Ավելի խոշոր մտածողները գեներացնում են ավելի կարճ, բայց ավելի բարձր էնտրոպիայով մտածողության շղթաներ (միջին երկարությունը մոտավորապես 4 639 թոքեն, էնտրոպիան՝

0.416), քան փոքր մոդելները (14 566 և 0.404), և այս հատկությունները համընկնում են միջմոդելային փոխանցման ավելի լավ որակի հետ: Գործնական հետևությունը հետևյալն է. նպատակահարմար է քեշավորել մտածողությունը հզոր LLM-ով և պատասխանների գեներացումը վերապահել փոքր LLM-ին՝ մոտենալու անկյունագծի լավագույն ճշգրտությանը ավելի ցածր հաշվարկային գնով:

Բանալի բառեր՝ մտքի շղթա CoT, միջմոդելային դատողության փոխանցում, Qwen3, GPQA բենչմարք, LLM թոքենների էնտրոպիա:

Лучшее мышление или более крупная модель? Перемешивание этапов размышления и ответа с Qwen3 на бенчмарке GPQA

Эдвард А. Халафян

Московский физико-технический институт, Москва, Россия
e-mail: edvardkhalafyan@gmail.com

Аннотация

В работе показано, что для семейства больших языковых моделей Qwen3 на бенчмарке Graduate-Level Google-Proof Question Answering (GPQA) качество мыслящей модели доминирует над размером отвечающей модели. Связка мыслитель $14B$ плюс отвечающий 0.6B достигает точности 54.24 процента, что близко к диагональному режиму $14B \rightarrow 14B$ с точностью 59.15 процента, тогда как мыслитель 0.6B снижает точность отвечающей модели $14B$ до 20.54 процента. Мы исследуем схему перемешивания размышления и ответа, в которой цепочка рассуждений chain-of-thought генерируется моделью одного размера $0.6B - 14B$ и передается моделям всех остальных размеров для выдачи только метки ответа, что охватывает все 5×5 комбинации на 448 заданиях GPQA. Точность монотонно растет с увеличением размера мыслителя, тогда как влияние размера отвечающей модели остается умеренным. Более крупные мыслители порождают более короткие и более высокоэнтропийные цепочки рассуждений средняя длина примерно 4 639 токенов, энтропия 0.416, чем меньшие модели 14 566 и 0.404, и эти характеристики коррелируют с более эффективным переносом между моделями. Практический вывод заключается в том, что целесообразно кешировать рассуждения с помощью сильной LLM и выполнять только этап ответа малой LLM, приближаясь к лучшей диагональной точности при меньших вычислительных затратах.

Ключевые слова: цепочка рассуждений CoT, перенос рассуждений между моделями, Qwen3, бенчмарк GPQA, энтропия токенов LLM.

Mathematical Problems of Computer Science 64, 29–36, 2025.
doi:10.51408/1963-0137

UDC 519.178

A Visualization and Modeling Tool for Fault-Tolerant Gossip Graphs

Vahagn S. Poghosyan

Institute for Informatics and Automation Problems of NAS RA, Yerevan, Armenia
povahagn@gmail.com

Abstract

The gossip problem (telephone problem) is an information dissemination problem where each of n nodes of a communication network has a unique message that should be transmitted to all the other nodes using two-way communications (telephone calls) between the pairs of nodes. During a call between the two given nodes, they exchange all the information known to them at that moment.

In this paper, a visualization and modeling tool for constructing and analyzing gossip graphs is presented. The tool features an interactive interface and automated algorithms for generating arbitrary graphs and evaluating their gossiping properties. Among the supported features are k-fault-tolerance analysis, simulation of Messy broadcast models, exploration of the NOHO (No One Hears one's Own information) schemes, detection of information flow folded paths, and identification of non-optimal communication patterns (i.e., repeated message arrivals).

Keywords: Gossip problem, Knödel graphs, Fault-tolerant gossip schemes.

Article info: Received 18 May 2025; sent for review 5 June 2025; accepted 13 October 2025.

Acknowledgements: This work was supported by the RA Science Committee, in the frames of the research projects 21AG-1B052 and 24DP-1B016.

1. Introduction

Gossiping is one of the basic problems of information dissemination in communication networks. The gossip problem (also known as a telephone problem) is attributed to A. Boyd (see ex. [1] for review), although, to the best knowledge of the reviewers, it was first formulated by R. Chesters and S. Silverman (Univ. of Witwatersrand, unpublished, 1970). Consider a set of n persons (nodes), each of which initially knows some unique message that is unknown to the others, and they can make a sequence of telephone calls to spread the information. During a call between the two given nodes, they exchange all the information known to them at that moment. The problem is to find a sequence of calls with minimum length (minimal gossip scheme), by which all the nodes will obtain all messages (complete gossiping). It has been shown in numerous works [1, 2, 3, 4] that the minimal number of calls is $2n-4$ when

$n \geq 4$ and $1, 3$ for $n = 2, 3$, respectively. Since then, many variations of the gossip problem have been introduced and investigated [5, 6, 7, 8, 9, 10].

The k-fault-tolerant gossip problem is a generalization of the gossip problem, where at most k arbitrary faults of calls are allowed.

Another variant of the Gossip problem can be formulated by considering the minimum amount of time required to complete gossiping among n persons, where the calls between non-overlapping pairs of nodes can take place simultaneously and each call requires one unit of time [11, 12]. For a detailed reviews on the various gossip problems and related open questions, see the works [13, 14, 15, 16, 17, 18, 19, 20]

In our studies, we implemented a Graph Plotter software tool. The main purpose of the tool is to check for a given input graph whether it satisfies the given level of fault-tolerance [21, 22, 23]. The tool also allows one to check whether the input graph is a NOHO (No One Hears one's Own information) [6] or a NODUP (No Duplicates) [19] graph. The NOHO graphs are graphs that do not contain any node that listens to its own information, or equivalently, do not contain a cycle. The NODUP graphs are defined as the graphs the nodes of which listen to each message exactly once. It means that there is exactly one increasing path between two arbitrary vertices.

In addition, the software tool offers a convenient interface for working with graphs. For example, it supports automatic numbering of vertices and edges, visual highlighting of paths between two vertices using customizable vertex and edge coloring, and the ability to add new edges with specified weights. It also allows sorting of all edge weights by converting them into increasing subsequent integers (discrete moments of calls, or tacts).

Moreover, the system enables modification of the positions of vertices adjacent to a selected edge, as well as the permutation of all adjacent edges with greater (or smaller) weights. The rationale behind this functionality is that if a call occurs between two arbitrary vertices at time t, both vertices will subsequently possess identical information. Therefore, any future calls from other nodes to these two vertices can be redirected to either one without affecting the information flow. This feature is particularly useful in scenarios where edges need to be merged into or dispersed from a single communication channel, especially in certain variants of the gossip problem where the number of edges per channel is constrained. Prior to this, exclusively a mathematical approach was used in solving problems of gossiping, and the hardware model of calculations was not used. See also [24, 25, 26] for the applications of local interchange operation to obtain minimum time gossip graphs.

Graph Plotter is an easily extensible tool. Since "Gossip test", "NOHO" and "NODUP" modules are written in C++ language and are separate processes in relation to the main process, the new modules can be integrated with this software tool easily.

The capabilities of the tool and its user interface are outlined below.

2. Graph Plotter GUI Overview

The Graph Plotter GUI is a dedicated visualization and analysis tool designed to interactively model, manipulate, and evaluate properties of fault-tolerant gossip graphs. This interface is especially suitable for research on edge-weighted graphs with custom vertex properties and dynamic structural operations.

The main visualization panel (See Fig. 1) provides an interactive interface to draw gossip graph schemes where vertices are displayed as labeled circular nodes and edges are colored and weighted according to user settings. The colors visually encode selected properties (e.g., active or test path edges). The edge weights represent sequences of integer or real-valued time steps, corresponding to moments of calls between adjacent vertices. In particular, a selected edge labeled with $\{2, 5\}$ is highlighted in red, while the other edges are in green. The interface allows one to color the subsets of vertices or edges to emphasize structural properties (e.g., increasing paths from one vertex to another, or a subset of vertices that violate the gossiping property). Vertex labels are positioned for clarity, and the entire graph layout supports symmetric positioning for readability.

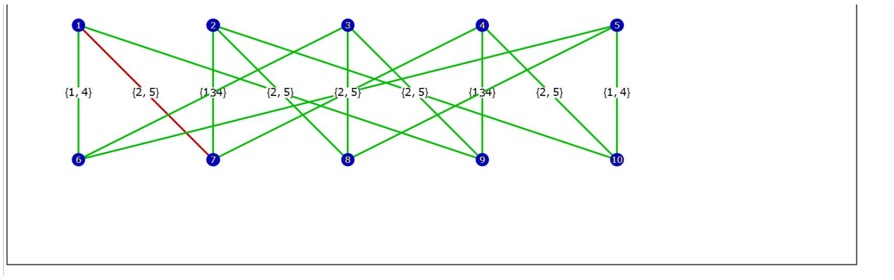

Fig. 1. Graph visualization interface with vertex labels and weighted edges.

Fig. 2 shows the edge parameter editor, where the user can modify edge weights, change colors, and permute vertex or edge orderings. The "permute higher/lower" buttons allow rearranging neighboring edges with weight-based filtering. Users can also collapse an edge or reset and sort edge weights using dedicated controls. The central control block computes key structural properties such as vertex and edge counts and theoretical bounds derived from gossip graph fault-tolerance parameters.

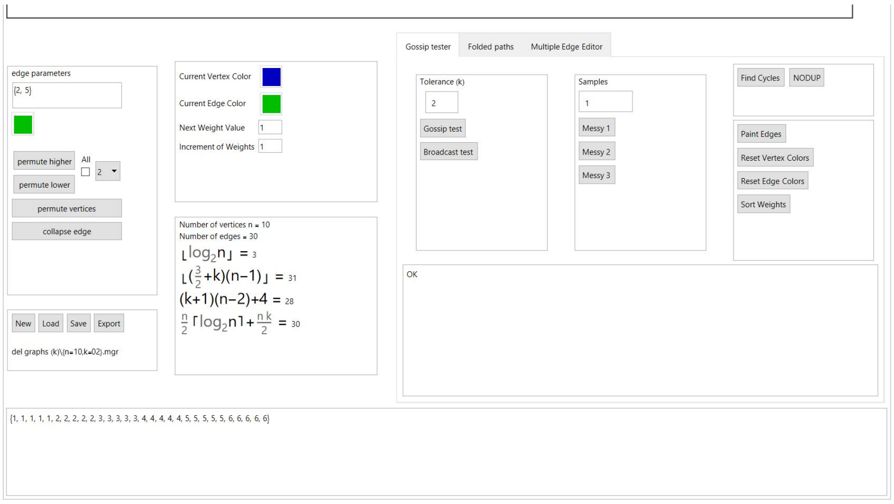

Fig. 2. Edge parameter editor and graph configuration panel.

The Messy broadcasting module in the GUI is dedicated to generating and analyzing Messy broadcast schemes on a given graph structure. The purpose of these tools is to simulate randomized edge colorings that conform to specific Messy models (M_1, M_2, M_3). Each model reflects a different level or type of communication disorder, allowing the user to evaluate robustness under various conditions. By generating a large number of such randomized samples, the tool enables estimation of an upper bound on the minimum achievable Messy broadcasting time for a given number of vertices. The Messy broadcasting module includes preset buttons labeled “Messy 1”, “Messy 2”, and “Messy 3”, corresponding to each of the Messy models. When activated, each button triggers a randomized coloring of the edges according to the constraints of the selected model. The generated configurations can then be evaluated through the gossip or broadcast test buttons to assess their effectiveness and timing. This experimental approach provides insight into the structural and probabilistic limits of information spread under uncertainty.

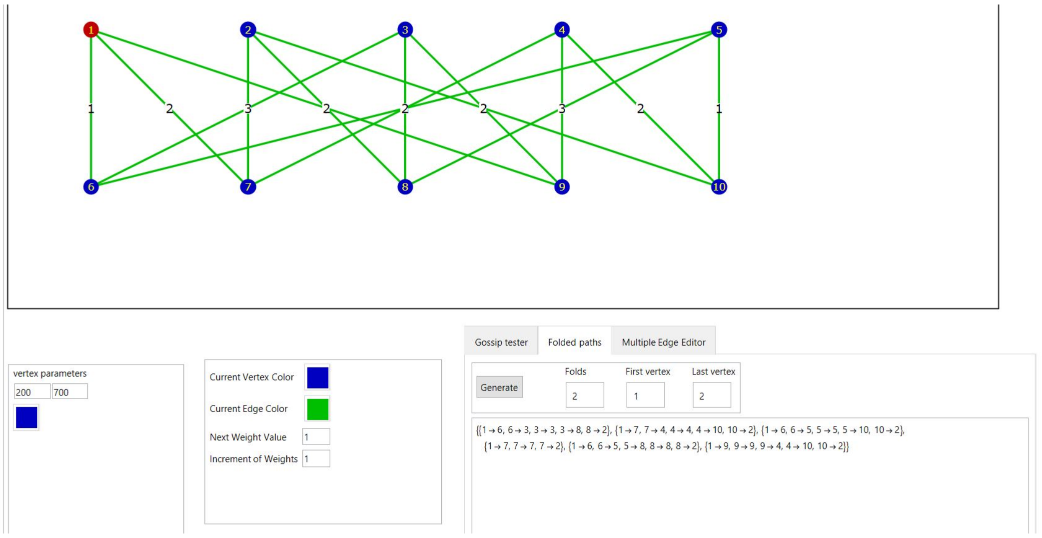

Fig. 3. Folded path generator for evaluating multi-step vertex connections.

The “Folded paths” tab enables the generation of folded paths from a given source to a destination vertex. As shown in Fig. 3, the user can specify the number of folds, and the system produces candidate folded paths following the allowed edge weight constraints. This is particularly useful for analyzing communication protocols with redundancy and delay bounds.

The “Multiple Edge Editor” tab, illustrated in Fig. 4, provides an interface to filter, append, delete, or collapse edges based on their weights. This helps batch process the graph’s edges for weight tuning or visualization simplification. For example, all edges with weight {2, 5} can be quickly listed and manipulated collectively.

The GUI supports gossip and broadcast tests under a specified fault-tolerance parameter k, cycle detection, edge painting, and edge color reset. These operations are crucial in evaluating the robustness and communication capabilities of the modeled network, particularly in the context of fault-tolerant information dissemination.

This tool has significantly enhanced our ability to explore and verify the properties of fault-tolerant gossip graphs. By combining interactive visualization with built-in simulation and testing modules, it enabled us to intuitively model communication processes, validate theoretical assumptions, and conduct practical experiments. The ability to dynamically adjust edge weights, visualize folded paths, and perform automated gossip and broadcast tests made it possible to investigate complex scenarios with precision and efficiency. As a result, the tool proved to be an indispensable aid in both theoretical analysis and applied research.

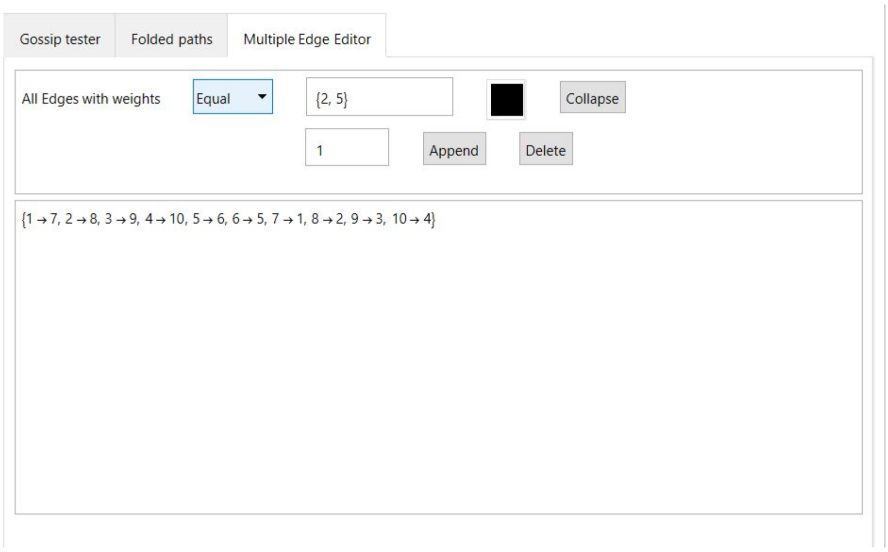

Fig. 4. Multiple edge editor for collective edge operations based on weights.

Overall, the Graph Plotter GUI is a flexible and powerful tool for researchers analyzing gossip graph structures and their resilience properties.

3. Conclusion

The visualization and modeling tool presented here offers an effective and user-friendly environment for studying fault-tolerant gossip schemes. Through its interactive interface and automated analysis capabilities, it enables detailed exploration of how information propagates under various assumptions and constraints. The tool thus provides researchers with a versatile instrument for both theoretical investigation and practical evaluation of gossip protocols.

References

[1] T. Tijdeman, "On a telephone problem", *Nieuw Archief voor Wiskunde*, vol. 3, pp. 188–192, 1971.

[2] B. Baker and R. Shostak, "Gossips and telephones," *Discrete Mathematics*, vol. 2, pp. 191–193, 1972.

[3] R. T. Bumby, "A problem with telephones," *SIAM Journal on Algebraic and Discrete Methods*, vol. 2, pp. 13–18, 1981.

[4] A. Hajnal, E. C. Milner, and E. Szemerdi, "A cure for the telephone disease," *Canadian Mathematical Bulletin*, vol. 15, pp. 447–450, 1972.

[5] A. Seress, "Quick gossiping by conference calls," *SIAM Journal on Discrete Mathematics*, vol. 1, pp. 109–120, 1988.

[6] D. B. West, "Gossiping without duplicate transmissions," *SIAM Journal on Algebraic and Discrete Methods*, vol. 3, pp. 418–419, 1982.

[7] G. Fertin and A. Respaud, "A survey on Knödel graphs," *Discrete Applied Mathematics*, vol. 137, no. 2, pp. 173–195, 2003.

[8] R. Labahn, "Kernels of minimum size gossip schemes," *Discrete Mathematics*, vol. 143, pp. 99–139, 1995.

[9] R. Labahn, "Some minimum gossip graphs," *Networks*, vol. 23, pp. 333–341, 1993.

[10] G. Fertin and R. Labahn, "Compounding of gossip graphs," *Networks*, vol. 36, pp. 126–137, 2000.

[11] A. Bavelas, "Communication pattern in task-oriented groups," *Journal of the Acoustical Society of America*, vol. 22, pp. 725–730, 1950.

[12] W. Knödel, "New gossips and telephones," *Discrete Mathematics*, vol. 13, p. 95, 1975.

[13] Z. Ho and M. Shigeno, "New bounds on the minimum number of calls in failure-tolerant gossiping," *Networks*, vol. 53, pp. 35–38, 2009.

[14] K. A. Berman and M. Hawrylycz, "Telephone problems with failures," *SIAM Journal on Algebraic and Discrete Methods*, vol. 7, pp. 13–17, 1987.

[15] R. W. Haddad, S. Roy, and A. A. Schaffer, "On gossiping with faulty telephone lines," *SIAM Journal on Algebraic and Discrete Methods*, vol. 8, pp. 439–445, 1987.

[16] T. Hasunuma and H. Nagamochi, "Improved bounds for minimum fault-tolerant gossip graphs," in *Lecture Notes in Computer Science*, vol. 6986, pp. 203–214, 2011.

[17] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Listman, "A survey of gossiping and broadcasting in communication networks," *Networks*, vol. 18, pp. 319–349, 1988.

[18] D. B. West, "A class of solutions to the gossip problem," Part I, *Discrete Mathematics*, vol. 39, no. 3, pp. 307–326, 1982; Part II, *Discrete Mathematics*, vol. 40, no. 1, pp. 87–113, 1982; Part III, *Discrete Mathematics*, vol. 40, no. 2–3, pp. 285–310, 1982.

[19] A. Seress, "Quick gossiping without duplicate transmissions," *Graphs and Combinatorics*, vol. 2, pp. 363–381, 1986.

[20] P. Fraigniaud, A. L. Liestman, and D. Soitoteau, "Open problems," *Parallel Processing Letters*, vol. 3, no. 4, pp. 507–524, 1993.

[21] V. Hovnanyan, S. Poghosyan, and V. Poghosyan, "New methods of construction of fault-tolerant gossip graphs," in *2013 International Conference on Computer Science and Information Technologies*, Revised Selected Papers, IEEE, pp. 1–5, 2013.

[22] V. Hovnanyan, S. Poghosyan, and V. Poghosyan, "Fault-tolerant gossip graphs based on wheel graphs," *Transactions of IIAP of NAS RA, Mathematical Problems of Computer Science*, vol. 42, pp. 43–53, 2014.

[23] V. Hovnanyan, S. Poghosyan, and V. Poghosyan, "Graph plotter: A software tool for the investigation of fault-tolerant gossip graphs," in *2013 International Conference on Computer Science and Information Technologies (CSIT)*, pp. 20–22, 2013.

[24] V. Hovnanyan, S. Poghosyan, and V. Poghosyan, "Method of local interchange to investigate gossip problems," *Transactions of IIAP of NAS RA, Mathematical Problems of Computer Science*, vol. 40, pp. 5–12, 2013.

[25] V. Hovnanyan, S. Poghosyan, and V. Poghosyan, "Method of local interchange to investigate gossip problems: Part 2," *Transactions of IIAP of NAS RA, Mathematical Problems of Computer Science*, vol. 41, pp. 15–22, 2014.

[26] V. Hovnanyan, S. Poghosyan, and V. Poghosyan, "Gossiping properties of the edge-permuted Knödel graphs," in *2017 International Conference on Computer Science and Information Technologies (CSIT)*, IEEE, pp. 1–4, 2017.

Անխափան գոսսիպ գրաֆերի մոդելավորման և վիզուալիզացիայի գործիք

Վահագն Ս. Պողոսյան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ, Երևան, Հայաստան
e-mail: povahagn@gmail.com

Ամփոփում

Գոսսիպ խնդիրը (հեռախոսային խնդիրը) ինֆորմացիայի տարածման դասական խնդիր է, որտեղ հաղորդակցային ցանցի n հանգույցներից յուրաքանչյուրն ունի եզակի հաղորդագրություն, որը պետք է փոխանցվի բոլոր մյուս հանգույցներին՝ օգտագործելով հանգույցներով կազմված զույգերի միջև իրականացվող երկկողմ կապեր (հեռախոսազանգեր): Զանգի ընթացքում տվյալ երկու հանգույցները փոխանակում են իրենց տվյալ պահին ունեցած ամբողջ ինֆորմացիան:

Այս աշխատությունում ներկայացվում է գոսսիպ գրաֆերի կառուցման և վերլուծության համար նախատեսված վիզուալիզացիայի և մոդելավորման գործիք: ործիքը տրամադրում է ինտերակտիվ միջավայր, ինչպես նաև ավտոմատացված ալգորիթմներ՝ կամայական գրաֆեր գեներացնելու և դրանց գոսսիպի հատկությունները գնահատելու համար:

Ֆունկցիոնալ հնարավորությունների թվում են k-վթարակայունության (fault-tolerance) վերլուծությունը, Messy լայնարձակ հաղորդումների մոդելավորումը, NOHO (No One Hears One’s Own information) սխեմաների ուսումնասիրությունը, ինֆորմացիոն հոսքերի կոտրատված ուղիների (folded paths) հայտնաբերումը, ինչպես նաև ոչ օպտիմալ հաղորդակցման օրինակների (օր. կրկնվող հաղորդագրությունների) բացահայտումը:

Բանալի բառեր՝ գոսսիպի խնդիր, Knodel գրաֆեր, անխափան գոսսիպի սխեմաներ:

Инструмент визуализации и моделирования отказоустойчивых gossip графов

Ваагн С. Погосян

Институт проблем информатики и автоматизации НАН РА, Ереван, Армения
e-mail: povahagn@gmail.com

Аннотация

Задача gossip (телефонная задача) представляет собой классическую задачу распространения информации, в которой каждый из n узлов коммуникационной сети изначально обладает уникальным сообщением, которое должно быть передано всем остальным узлам посредством двусторонних соединений (телефонных звонков). Во время каждого звонка два участвующих узла обмениваются всей имеющейся у них на данный момент информацией.

В данной работе представлен инструмент для визуализации и моделирования gossip графов, предназначенный для их построения и анализа. Инструмент включает интерактивный интерфейс и автоматизированные алгоритмы для генерации произвольных графов и оценки их свойств в контексте распространения информации. Поддерживаемые возможности включают анализ.

k-отказоустойчивости, моделирование Messy-broadcast моделей, изучение NOHO схем (No One Hears One's Own information), обнаружение изломанных путей информационного потока, а также выявление неоптимальных коммуникационных паттернов, таких как повторные поступления сообщений.

Ключевые слова: задача gossip, графы Кнёделя, отказоустойчивые gossip-схемы.

Mathematical Problems of Computer Science 64, 37–46, 2025.
doi:10.51408/1963-0138

UDC 004.855.5

A PDE-Based Convolutional Neural Network with Variational Information Bottleneck: Experimental Evaluation and Generalization Analysis

Gor A. Gharagyozyan

Institute for Informatics and Automation Problems of NAS RA, Yerevan, Armenia
e-mail: gor.gharagyozyan@edu.isec.am

Abstract

We present a hybrid convolutional architecture that combines trainable PDE-based preprocessing with a Variational Information Bottleneck (VIB) to improve generalization in image classification. The PDE stage applies a small number of discretized Laplacian steps with learnable step size and depthwise coupling, injecting physics-inspired inductive bias into early feature maps. A tensor-wise VIB module then parameterizes a Gaussian latent (μ, $\log \sigma^2$) via 1×1 convolutions and enforces information compression through a KL penalty to a unit prior, encouraging retention of task-relevant features while discarding nuisance variability. The compressed representation feeds a ResNet-18 backbone adapted for CIFAR-10 inputs. On CIFAR-10, systematic variation of the VIB weight β shows that moderate compression yields improved test performance and training stability relative to both a baseline CNN and a PDE-only variant. Qualitative analysis indicates smoother activations and reduced sensitivity to input noise, consistent with the information-theoretic objective. The results suggest that PDE priors and variational compression act complementarily, offering a principled path to robust and generalizable convolutional models.

Keywords: Information bottleneck, Partial differential equations, Deep learning, Convolutional neural networks, Generalization.

Article info: Received 15 October 2025; sent for review 16 October April 2025; accepted 13 November 2025.

Acknowledgements: The author expresses sincere gratitude to Prof. Mariam Haroutunian for her steadfast guidance, insightful discussions, and generous support throughout this work. Her rigorous feedback and encouragement were instrumental in shaping the research questions, refining the methodology, and bringing this study to completion.

1. Introduction

Convolutional Neural Networks (CNNs) have become the foundation of modern computer vision, demonstrating outstanding performance across various image classification tasks. Architectures such as ResNet [1] have shown that deep hierarchical representations can capture complex visual patterns; however, their ability to generalize remains sensitive to data quality, overparameterization, and the presence of irrelevant features. Improving generalization thus requires mechanisms that not only increase model capacity but also regulate the information flow within the network.

Recent studies have explored the incorporation of domain knowledge into CNNs to embed structural priors and reduce reliance on purely data-driven learning. In particular, Partial Differential Equation (PDE)-based layers [2], derived from the discretization of physical processes such as diffusion and wave propagation, have been shown to enhance low-level representations by enforcing spatial smoothness and continuity. These physics-inspired kernels act as a regularizing bias, improving robustness without adding significant computational cost. Yet, such deterministic transformations may also retain redundant information, which can propagate noise through deeper layers.

To address this limitation, this article builds upon a previous study presented at the Conference on Computer Science and Information Technologies (CSIT 2025) [3], where the integration of the Variational Information Bottleneck (VIB) module [4, 5] into the PDE-based CNN framework was first introduced conceptually. In the present work, the approach is evaluated through quantitative experiments, providing empirical evidence for the effectiveness of the PDE-VIB combination. The VIB principle seeks a stochastic latent representation that retains only information relevant to predicting the target while discarding task-irrelevant details. By combining PDE-based structural priors with information-theoretic compression, the proposed PDE-VIB-CNN achieves a balance between inductive bias and adaptive regularization. The resulting model learns compact and task-focused feature maps, leading to improved stability, robustness, and generalization on challenging datasets such as CIFAR-10.

2. Theoretical Background

This section provides a brief overview of the theoretical components underlying the proposed model, the PDE-based convolutional layers, and the Variational Information Bottleneck (VIB) framework. A detailed formulation and motivation can be found in [2] and [3].

PDE-Based Convolutional Layers

The PDE-based layer incorporates physically inspired priors into early feature extraction. The approach relies on the discretization of parabolic or hyperbolic partial differential equations, such as the two-dimensional diffusion (heat) equation:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}. \tag{1}$$

Using finite differences [6], the update rule can be expressed as:

$$u_{i,j}^{t+1} = u_{i,j}^{t} + \phi\, P(u^t), \tag{2}$$

where P denotes a convolution operator equivalent to the Laplacian kernel and ϕ is a learnable or fixed scaling parameter. These layers act as structural filters, enforcing smoothness and spatial continuity while reducing sensitivity to high-frequency noise [2].

VIB

The VIB framework [4, 5] formulates learning as an optimization of the mutual information trade-off between input compression and predictive relevance. The objective is defined as:

$$\mathcal{L}_{\text{VIB}} = \mathbb{E}_{\mathrm{p(x,y)}}\Big[\mathbb{E}_{\mathrm{q(t|x)}}\big[-\log \mathrm{p(y \mid t)}\big]\Big] + \beta\, \mathrm{D}_{\mathrm{KL}}(\mathrm{q(t \mid x)} \,\|\, \mathrm{p(t)}), \tag{3}$$

where $\mathrm{q(t \mid x)}$ is a Gaussian encoder producing the latent representation t, $\mathrm{p(y \mid t)}$ is the decoder, and β controls the compression–prediction trade-off. This formulation encourages the representation to retain only task-relevant information while suppressing redundancy.

The *PDE-VIB-CNN* model evaluated in this study extends the theoretical foundation proposed in [3], validating it experimentally on CIFAR-10 [7].

3. Experimental Setup

Dataset

All experiments are conducted on the CIFAR-10 dataset (60,000 color images, 32×32, 10 classes; 50k train, 10k test). Inputs are normalized per channel. We apply standard data augmentation: random horizontal flipping and random cropping with 4-pixel padding.

Architecture

The evaluated model, *PDE-VIB-CNN*, consists of three sequential stages.

(i) PDE stage. We prepend a stack of S PDE-based convolutional layers, each corresponding to one explicit Euler update of a discretized Laplacian step. For an input feature map u^t, a single PDE layer computes

$$u^{t+1} = u^t + \lambda\,(P * u^t),$$

where P is a fixed 3×3 Laplacian stencil and λ is a **learnable per-channel diffusion coefficient**. Thus, the PDE block performs S successive PDE updates (we use $S = 3$ in all experiments unless otherwise stated), injecting physics-inspired priors and encouraging smooth, spatially coherent feature representations [2]. No free-form convolution kernels are learned in this stage; the only learnable parameters are the diffusion coefficients $\{\lambda_c\}$ and batch-normalization parameters.

(ii) VIB module. After the PDE stage, we apply a variational information bottleneck module [4, 5]. The PDE output $f(x)$ is first compressed through a 1×1 bottleneck ($C \to C_b$ channels). Two parallel 1×1 convolutions then produce the parameters of a Gaussian latent distribution:

$$\mu(x) = W_\mu * f(x) + b_\mu, \qquad \log \sigma^2(x) = W_\sigma * f(x) + b_\sigma.$$

A latent tensor is sampled via the reparameterization trick,

$$t = \mu(x) + \sigma(x) \odot \varepsilon, \qquad \varepsilon \sim \mathcal{N}(\mathcal{0}, \mathcal{I}),$$

ensuring differentiability during training. This module enforces information compression and reduces overfitting by discouraging the encoding of spurious, high-frequency details.

(iii) CNN backbone. The sampled latent representation t is passed to a ResNet-18 backbone [1] (with the initial stem modified for CIFAR-10), followed by a linear classifier. This journal version extends the conceptual formulation introduced in [3] by providing detailed implementation, quantitative evaluation, and calibration analysis.

Training Protocol

Models are implemented in PyTorch [8] and trained end-to-end using stochastic gradient descent with momentum (SGD with momentum). Batch normalization is used in convolutional blocks, and dropout is employed in deeper layers to reduce overfitting. The VIB trade-off coefficient β is tuned empirically to balance compression and accuracy.

For fairness, all models are trained under **identical optimization and augmentation settings**, including the cosine-decay learning-rate schedule, weight decay, batch size, and number of epochs. The baseline CNN consists of the same ResNet-18 backbone used in the proposed architectures, but without any PDE layers or VIB module; it receives the raw augmented CIFAR-10 images directly as input. Thus, any observed performance or calibration differences stem purely from the PDE preprocessing and VIB regularization rather than from changes in the backbone or training procedure.

Evaluation Metrics

We report top-1 test accuracy, the train test (generalization) gap, the *Negative Log-Likelihood (NLL)*, and the *Expected Calibration Error (ECE)*.

Negative Log-Likelihood (NLL)

$$\mathrm{NLL} = -\frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}} \log \mathrm{p}_\theta(\mathrm{y_i} \mid \mathrm{x_i}) \,, \tag{4}$$

which evaluates the quality of probabilistic predictions and is the standard log-loss for classifiers [9].

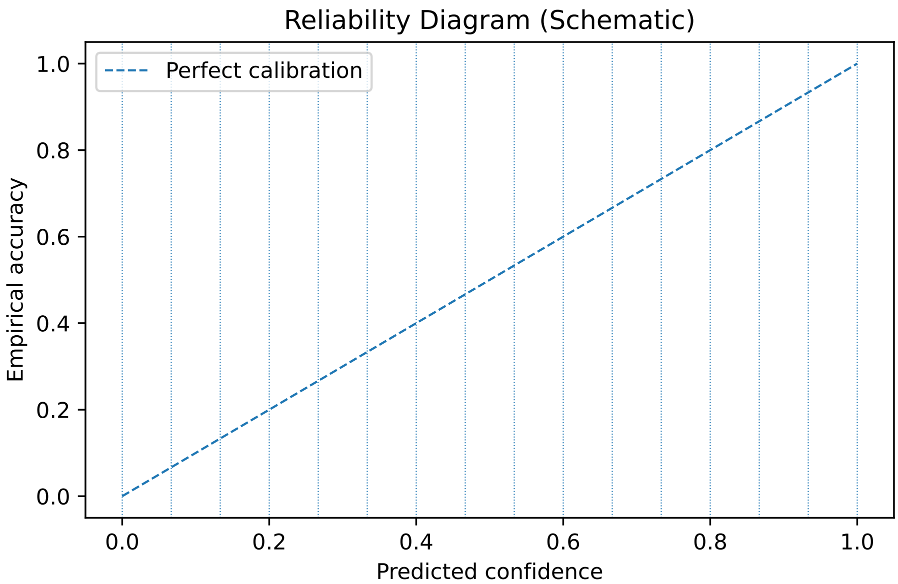

Fig. 1. eliability diagram used to compute ECE. The confidence range $[0, 1]$ is partitioned into M bins $\{B_m\}_{m=1}^{M}$. For each bin, we compare empirical accuracy to mean predicted confidence; ECE averages the absolute gap across bins (weighted by bin frequency).

Expected Calibration Error (ECE)

Partition confidence scores into M bins $\{B_m\}_{m=1}^{M}$ and compute

$$\mathrm{ECE} = \sum_{\mathrm{m}=1}^{\mathrm{M}} \frac{|\mathrm{B_m}|}{\mathrm{n}} \, |\, \mathrm{acc}(\mathrm{B_m}) - \mathrm{conf}(\mathrm{B_m}) \,|, \tag{5}$$

where $\mathrm{acc}(\mathrm{B_m})$ is the average accuracy and $\mathrm{conf}(\mathrm{B_m})$ is the average predicted confidence in bin m [10]. In our experiments, we use a fixed M (e.g., M=15) unless stated otherwise. Calibration is assessed with a reliability diagram (see Fig. 1), which compares per-bin empirical accuracy to mean predicted confidence; ECE aggregates the binwise gaps to a single score.

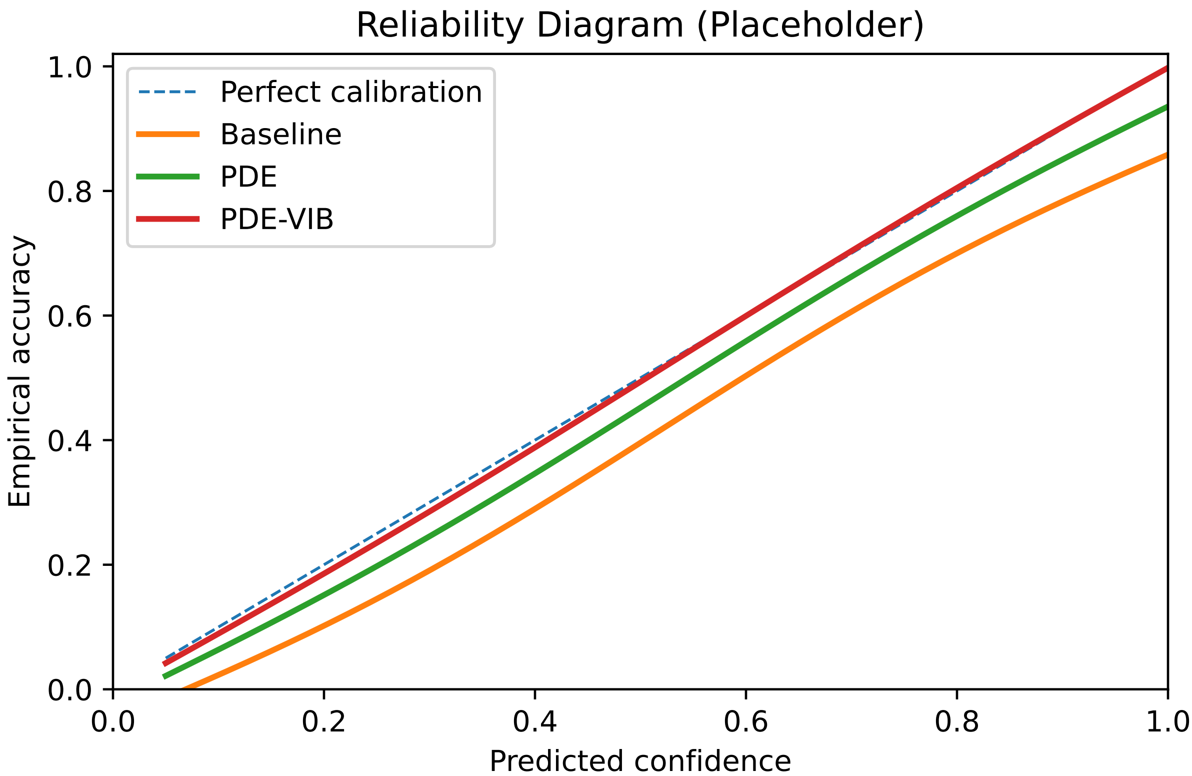

Fig. 2. Test-set reliability diagram (CIFAR-10, M=15). The proposed PDE-VIB model is closest to perfect calibration (y=x).

4. Results

Baselines and Comparisons

We compare three models trained under identical protocols: (i) a baseline CNN, (ii) a PDE-enhanced variant (PDE-only), and (iii) the proposed *PDE-VIB-CNN*. Table 4. reports top-1 test accuracy, generalization gap (train–test), ECE and NLL. The PDE-VIB model improves both accuracy and calibration relative to the baseline and PDE-only variants.

Table 1. CIFAR-10 test metrics

Model	**Accuracy (%)**	**Gap (%)**	**ECE**	**NLL**
Baseline CNN	80.6000	2.6600	0.037420	0.584965
PDE-only	88.1500	4.2100	0.041851	0.378198
PDE-VIB (Ours)	88.9500	3.5340	0.018466	0.385564

Calibration and Probabilistic Quality

Calibration is assessed with a reliability diagram (Fig. 1), which contrasts per-bin empirical accuracy with mean predicted confidence; ECE is the frequency-weighted average of binwise gaps. Fig. 2 illustrates that *PDE-VIB-CNN* lies closer to the diagonal $y{=}x$ than the baselines, indicating improved calibration, which is also reflected by lower ECE and NLL in Table 1.

Sensitivity to the Bottleneck Strength

We sweep the VIB coefficient β to study the compression–prediction trade-off. Moderate values of β (e.g., 10^{-4}) yield the best balance, reducing ECE/NLL without hurting accuracy. Fig. 3 summarizes the trend.

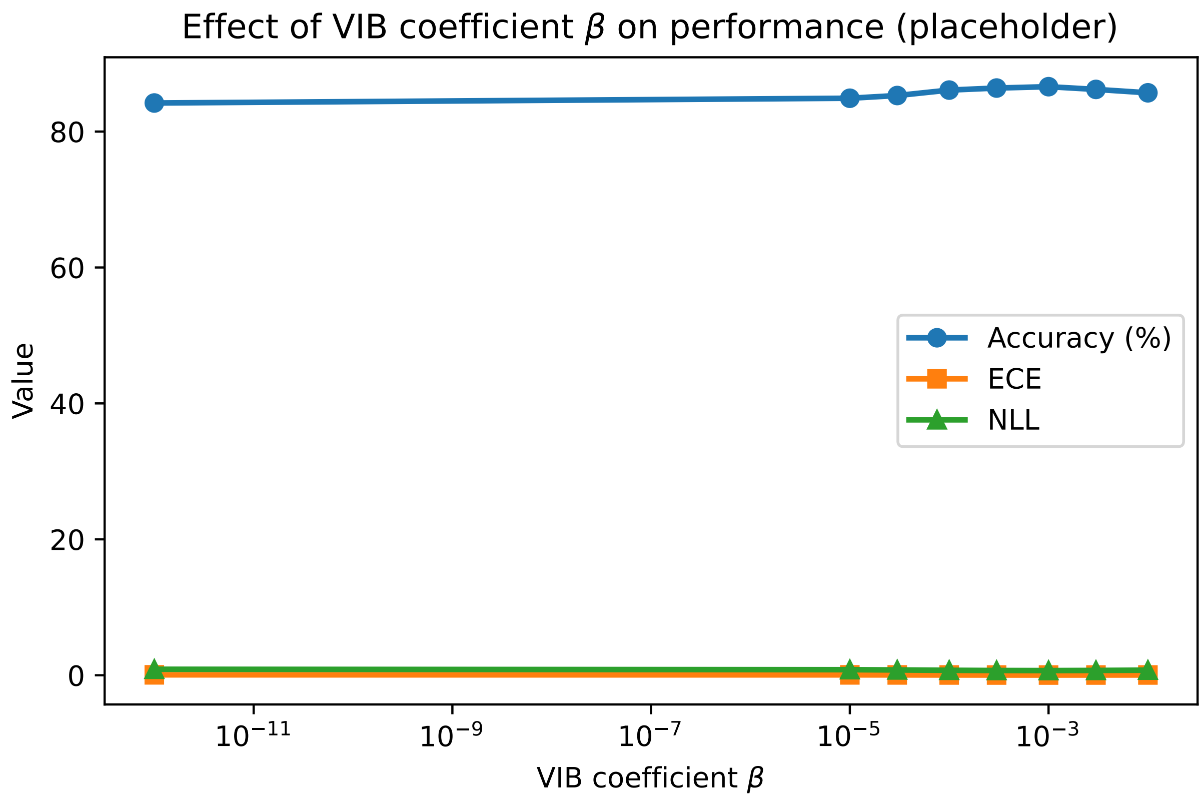

Fig. 3. Effect of the VIB coefficient β on test accuracy, ECE, and NLL (CIFAR-10). Moderate compression achieves the best overall performance.

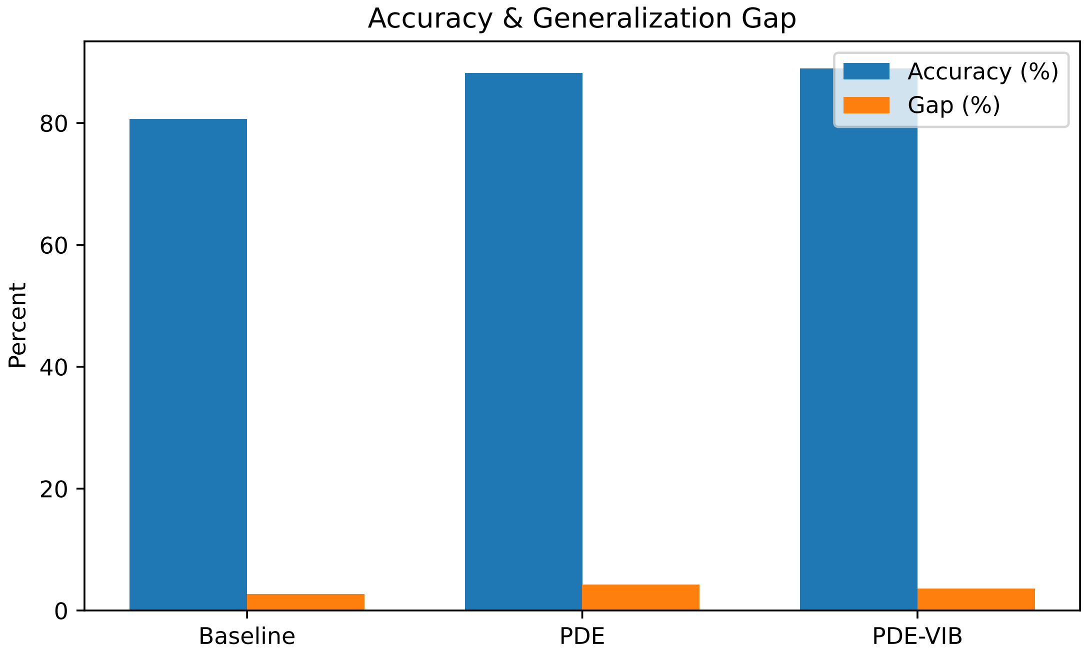

Fig. 4. Accuracy and generalization gap (CIFAR-10). Higher accuracy and lower gap are better.

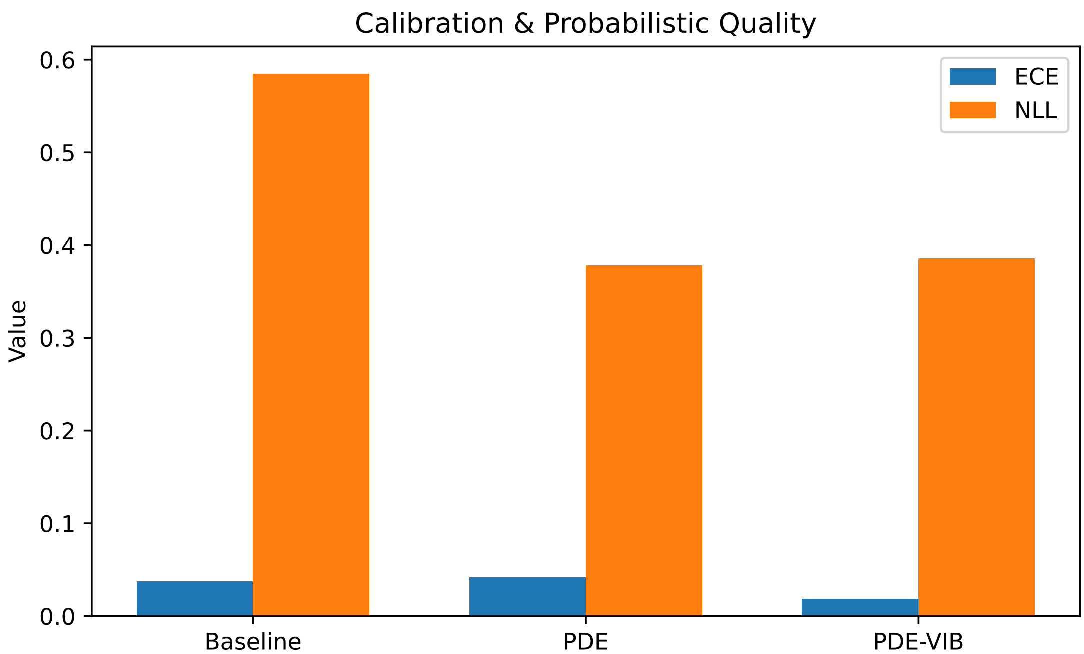

Fig. 5. Calibration and probabilistic quality (CIFAR-10). Lower ECE and NLL are better.

Qualitative Observations

Figures 4 and 5 provide a consolidated view of test performance on CIFAR-10 across the three model variants. The *PDE-only* configuration narrows the train–test discrepancy relative to the baseline and yields modest gains in probabilistic quality, suggesting that physics-inspired preprocessing already curbs overfitting. The proposed *PDE-VIB-CNN* delivers the most favorable overall profile: it achieves the highest or statistically comparable top-1 accuracy, while further reducing the generalization gap. Crucially, this gain does not come at the expense of calibration: the ECE bars in Fig. 5 show a clear reduction for *PDE-VIB-CNN*, accompanied by a lower NLL, indicating more reliable confidence estimates and better-aligned likelihoods.

5. Conclusion

This work evaluated a hybrid *PDE-VIB-CNN* that combines physics-inspired PDE preprocessing with a VIB. On CIFAR-10, the approach achieved the strongest overall profile among the tested variants: it matched or exceeded the best top-1 accuracy while reducing the train-test gap, and it delivered the lowest ECE and NLL. These findings support the hypothesis that PDE layers encourage spatially smooth, noise-resistant features, whereas the VIB term suppresses task-irrelevant variabilitytogether yielding models that are both discriminative and better calibrated.

Despite these gains, several limitations remain. Our evaluation is confined to a single dataset and moderate-scale backbones; the sensitivity to the bottleneck strength β and the number/step size of PDE layers indicates a performance compression trade-off that warrants deeper study. Moreover, while the PDE stage is lightweight, the VIB stochasticity adds minor computational overhead during training; understanding accuracy-calibration-efficiency trade-offs at larger scales is important.

Future work. (i) *Scaling and datasets:* extend to larger architectures [11] (e.g., Wide-ResNets, ConvNeXt) and datasets (CIFAR-100, Tiny-ImageNet, ImageNet-1k) to test the robustness of the observed trends. (ii) *Calibration under shift:* [12] evaluate on corruption and shift benchmarks (e.g., CIFAR-C, ImageNet-C/O) and out-of-distribution detection; compare pre- and post-hoc calibration (temperature scaling, Dirichlet calibration) with and without VIB. (iii) *Comparative regularization:* benchmark against strong baselines such as label smoothing [13], mixup/cutmix, dropout variants, stochastic depth, and data-augmix to clarify where PDE-VIB provides unique benefits.

References

[1] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770-778, 2016.

[2] V. R. Sahakyan, V. G. Melkonyan, G. A. Gharagyozyan, and A. S. Avetisyan, "Enhancing Image Recognition with Pre-Defined Convolutional Layers Based on PDEs", *Programming and Computer Software*, vol. 49, no. 3, pp. 192-197, 2023.

[3] Gor Gharagyozyan, "Improving CNN Generalization with PDE Preprocessing and the Variational Information Bottleneck", *Proceedings of International CSIT Conference 2025*, Yerevan, Armenia, pp. 145–147, 2025.

[4] A. A. Alemi, I. Fischer, J. V. Dillon and K. Murphy, "Deep variational information bottleneck", *arXiv:1612.00410*, 2019.

[5] N. Tishby, F. C. Pereira and W. Bialek, “The information bottleneck method”, *arXiv:physics/0004057*, 1999.

[6] R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial-Value Problems*, Second Edition, John Wiley & Sons, New York, 1967.

[7] A. Krizhevsky, “Learning Multiple Layers of Features from Tiny Images.” University of Toronto, Tech. Rep., 2009.

[8] Pytorch home page. [Online]. Available: https://pytorch.org/

[9] C. Guo, G. Pleiss, Y. Sun and K. Q. Weinberger, “On Calibration of Modern Neural Networks,”, *Proceedings of ICML*, pp. 1321-1330, 2017.

[10] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[11] S. Zagoruyko and N. Komodakis, “Wide Residual Networks”, *arXiv:1605.07146*, 2016.

[12] D. Hendrycks, N. Mu, E. D. Cubuk, et al., “AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty”, *arXiv:1912.02781*, 2020.

[13] S. Kudo, “Label Smoothing is a Pragmatic Information Bottleneck”, *arXiv:2508.14077*, 2025.

Վարիացիոն ինֆորմացիոն խցանով մասնակի ածանցյալներով դիֆերենցիալ հավասարումների վրա հիմնված կոնվոլյուցիոն նեյրոնային ցանց. փորձնական գնահատում և ընդհանրացման վերլուծություն

Գոռ Ա. Ղարագյոզյան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ, Երևան, Հայաստան
e-mail: gor.gharagyozyan@edu.isec.am

Ամփոփում

Մենք ներկայացնում ենք հիբրիդային կոնվոլյուցիոն ճարտարապետություն, որը համատեղում է մասնակի ածանցյալներով հավասարումների վրա հիմնված ուսուցանվող նախամշակումը և վարիացիոն ինֆորմացիոն խցանը՝ պատկերների դասակարգման մեջ ընդհանրացման բարելավման նպատակով: Մասնակի ածանցյալներով հավասարումների շերտը կիրառում է փոքր քանակի դիսկրետացված Լապլասյան քայլեր՝ ուսուցանվող քայլի չափով և խորքային կապով, ինչը վաղ փուլի հատկանիշային քարտեզներում ներմուծում է ֆիզիկայից ոգեշնչված ինդուկտիվ կողմնակալություն: Թենզորային վարիացիոն ինֆորմացիայի խցանի մոդուլը պարամետրավորում է աուսյան թաքնված բաշխումը (μ, $\log\sigma^2$) 1×1 կոնվոլյուցիաների միջոցով և կիրառում է KL տուգանքը միավորային նախնական բաշխման նկատմամբ՝ տեղեկատվության սեղմում ապահովելու համար, ինչը խրախուսում է պահպանել առաջադրանքի համար կարևոր հատկանիշները և հեռացնել ոչ անհրաժեշտ փոփոխականությունը: Սեղմված ներկայացումը այնուհետև փոխանցվում է ResNet-18-ին: Վարիացիոն ինֆորմացիոն խցանի β կշռի համակարգված փոփոխությունը ցույց է տալիս, որ միջին աստիճանի սեղմումը բարելավում է թեստային արդյունավետությունն ու ուսուցման կայունությունը՝ համեմատած ինչպես ստանդարտ ցանցի, այնպես էլ միայն մասնակի ածանցյալներով տարբերակի հետ: Որակական վերլուծությունը ցույց է տալիս ավելի հարթ ակտիվացումներ և մուտքային աղմուկի նկատմամբ զգայունության նվազում, ինչը համապատասխանում է ինֆորմացիայի տեսության նպատակին: Արդյունքները ցույց են տալիս, որ մասնակի ածանցյալներով նախնական բաշխումները և ինֆորմացիոն խցանով սեղմումը փոխլրացնում են իրար, առաջարկելով սկզբունքային ուղի դեպի կայուն և ընդհանուր կիրառելի կոնվոլյուցիոն մոդելներ:

Բանալի բառեր՝ ինֆորմացիոն խցան, մասնակի ածանցյալներով դիֆերենցիալ հավասարումներ, խորքային ուսուցում, կոնվոլուցիոն նեյրոնային ցանցեր, ընդհանրացում:

Сверточная нейронная сеть на основе PDE с вариационной информационной пробкой: экспериментальная оценка и анализ обобщения

Гор А. Карагёзян

Институт проблем информатики и автоматизации НАН РА, Ереван, Армения
e-mail: gor.gharagyozyan@edu.isec.am

Аннотация

Мы представляем гибридную сверточную архитектуру, которая сочетает в себе обучаемую предварительную обработку на основе дифференциальных уравнений в частных производных с вариационной информационной пробкой для улучшения обобщения при классификации изображений. На этапе уравнений в частных производных применяется небольшое количество дискретизированных шагов Лапласа с обучаемым размером шага и глубинной связью, вводя индуктивное смещение, вдохновленное физикой, в ранние карты признаков. Затем тензорный модуль вариационной информационной пробки параметризует гауссову латентную величину (μ, $\log \sigma^2$) с помощью 1×1 сверток и обеспечивает сжатие информации с помощью штрафа KL до единичного априорного распределения, способствуя сохранению релевантных для задачи признаков и отбрасывая помехи. Сжатое представление подается на базовую сеть ResNet-18, адаптированную для входных данных CIFAR-10. На CIFAR-10 систематическое изменение веса вариационной информационной пробки β показывает, что умеренное сжатие дает улучшенную производительность тестирования и стабильность обучения по сравнению как с базовым сверточная нейронная сеть, так и с вариантом, использующим только уравнений в частных производных. Качественный анализ указывает на более плавные активации и сниженную чувствительность к входному шуму, что соответствует цели теории информации. Результаты показывают, что априорные значения уравнений в частных производных и вариационное сжатие действуют взаимодополняюще, предлагая принципиальный путь к надежным и обобщаемым сверточным моделям.

Ключевые слова: информационная пробка, уравнения в частных производных, глубокое обучение, сверточные нейронные цети, обобщение.

Mathematical Problems of Computer Science 64, 47–55, 2025.
doi: 10.51408/1963-0139

UDC 004.93

PDE-UNet: A Modified UNet Architecture Applied to Medical Image Segmentation

Rafayel M. Veziryan

Institute for Informatics and Automation Problems of NAS RA, Yerevan, Armenia
e-mail: rafaelveziryan@gmail.com

Abstract

Medical image segmentation is a critical task in healthcare, particularly for disease detection and proper treatment planning. Deep learning models achieve high performance in medical image analysis. This paper presents the effectiveness of the new PDE-UNet architecture, inspired by the applications of partial differential equations (PDEs) in neural networks, to enhance medical image segmentation performance. Experiments were conducted on brain tumor MRI images from the BraTS2020 dataset and compared with the traditional UNet architecture.
Keywords: Medical Image Segmentation, Brain Tumor Segmentation, UNet, PDE-inspired CNN block, MICCAI BraTS 2020 Challenge.
Article info: Received 30 September 2025; sent for review 6 October 2025; accepted 13 November 2025.

1. Introduction

Artificial intelligence and deep learning models are becoming highly efficient in healthcare [1], providing automated analysis, diagnosis, and decision support. Particularly, they demonstrated performance in medical imaging tasks, such as segmentation, classification, and detection [2].

PDEs play an important role in image processing and computer vision [3]. They have been widely used for edge detection [4], image denoising [5], and image inpainting [6]. By modeling continuous transformations, PDEs enable the preservation of important structures while effectively reducing noise and improving image quality.

UNet is a convolutional neural network (CNN) architecture developed for medical image analysis, which can be trained on relatively small datasets and achieve high performance [7]. Today, it remains a primary tool for segmentation tasks in medical imaging. The UNet consists of two parts: an encoder and a decoder, connected by skip connections, as shown in Fig.

The primary goal of this study is to investigate the power of PDE-UNet in medical image segmentation tasks, with a particular focus on brain tumor segmentation using the

BraTS2020 dataset [8] from the MICCAI BraTS 2020 challenge. PDE-UNet proposes a trainable preprocessing PDE-inspired convolutional block to extract important image features before passing the input to the main segmentation network.

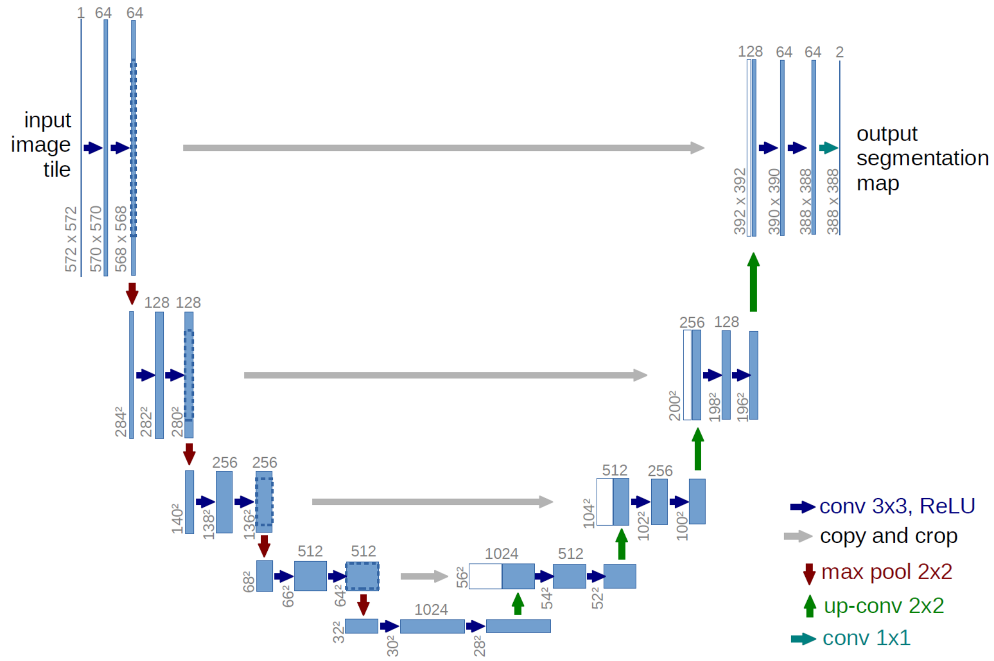

Fig. 1. UNet architecture.

2. Related Work

2.1 Medical Image Segmentation using Deep Learning

Deep learning has achieved significant success in medical image analysis, enabling the automation of clinical tasks such as tumor detection and lesion segmentation. Despite various approaches, encoder-decoder architectures such as UNet have become dominant in the field, demonstrating high performance even with small datasets.

2.2 Partial Differential Equations in Neural Networks

Due to the wide use of PDEs in image processing, they have been integrated into deep neural networks [9] to improve feature representation, stability, and local interaction modeling [10], as well as to preserve structural information. An example of such integration is demonstrated in [11], where the integration of the Cahn-Hilliard PDE-based model into the UNet architecture for image segmentation was proposed.

Recent research [12] has integrated a PDE-inspired convolutional operator, based on the Cable equation (1), which models the transmission of potentials in neural cells of the human brain [13], as a trainable preprocessing layer in a residual CNN for classification tasks.

$$\tau_m \frac{\partial v(x,t)}{\partial t} = -v(x,t) + \lambda_m^2 \frac{\partial^2 v(x,t)}{\partial x^2} + r_m I_{ext}(x,t), \;\; t \geq 0, \tag{1}$$

The final form of the discretized PDE operator, using the finite difference method [14] for equation (1), is as follows.

$$u_{i,k}^{t+1} = (1-\tau) \cdot u_{i,k}^{t} + (P_1 + P_2) \cdot U, \tag{2}$$

where

$$U = \begin{bmatrix} u_{i-1,k-1}^t & u_{i-1,k}^t & u_{i-1,k+1}^t \\ u_{i,k-1}^t & u_{i,k}^t & u_{i,k+1}^t \\ u_{i+1,k-1}^t & u_{i+1,k}^t & u_{i+1,k+1}^t \end{bmatrix}, P_1 = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{\Phi^2} & -\frac{2}{\Phi^2} & \frac{1}{\Phi^2} \\ \text{‘}0 & 0 & 0 \end{bmatrix}, P_2 = \begin{bmatrix} 0 & \frac{1}{\Psi^2} & 0 \\ 0 & -\frac{2}{\Psi^2} & 0 \\ 0 & \frac{1}{\Psi^2} & 0 \end{bmatrix}.$$

P_1 and P_2 are defined as two-dimensional weighted convolution operators for the neural network with weights Φ, Ψ, and τ.

2.3 Motivation

While advanced architectures such as Attention UNet [15] and TransUNet [16] perform well in medical image segmentation, they often struggle to capture fine structural details and typically need large, annotated datasets. Motivated by previous research in image classification, we investigate the effectiveness of PDE-UNet in medical image segmentation, aiming to enhance feature extraction and segmentation accuracy, and recommend adding a PDE-inspired lightweight preprocessing layer to the traditional UNet.

3. Experiments

All models were implemented in PyTorch and trained on a NVIDIA RTX 3050 GPU with 4 GB of memory. Our experiments focus on brain tumor MRI images from the BraTS2020 dataset, part of the MICCAI BraTS2020 challenge. A sample of the dataset is shown in Fig. 3. The dataset consists of multi-modal brain MRI scans, including FLAIR, T1, T1ce, and T2 modalities, with expert annotations for three tumor subregions: the enhancing tumor (ET), tumor core (TC), and whole tumor (WT). The experiments were made on the extracted 2D slices from the volumes, using the middle slices of each modality, which still capture the most relevant tumor structures.

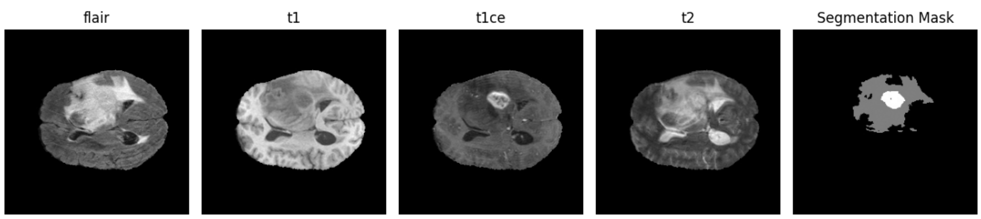

Fig. 2. A sample from the BraTS dataset.

We conducted experiments on a UNet architecture, which has skip connections linking the encoder and decoder layers. This design aligns well with our PDE-inspired convolutional block, which also incorporates a skip connection, allowing for effective feature propagation and preservation. In this study, the UNet architecture takes 4-channel input images corresponding to the FLAIR, T1, T1ce, and T2 MRI modalities. The encoder consists of convolutional blocks with increasing feature dimensions of 32, 64, 128, and 256 channels, followed by a bottleneck block with 512 features. The decoder mirrors the encoder, using transposed convolutions to upsample the feature maps, and combines these with skip connections from the corresponding encoder layers. The final 11 convolution produces four output channels corresponding to the tumor subregions (background, ET, TC, and WT). The proposed PDE-UNet architecture has an additional PDE-inspired block compared with UNet. Figure 3. illustrates the UNet and PDE-UNet architectures.

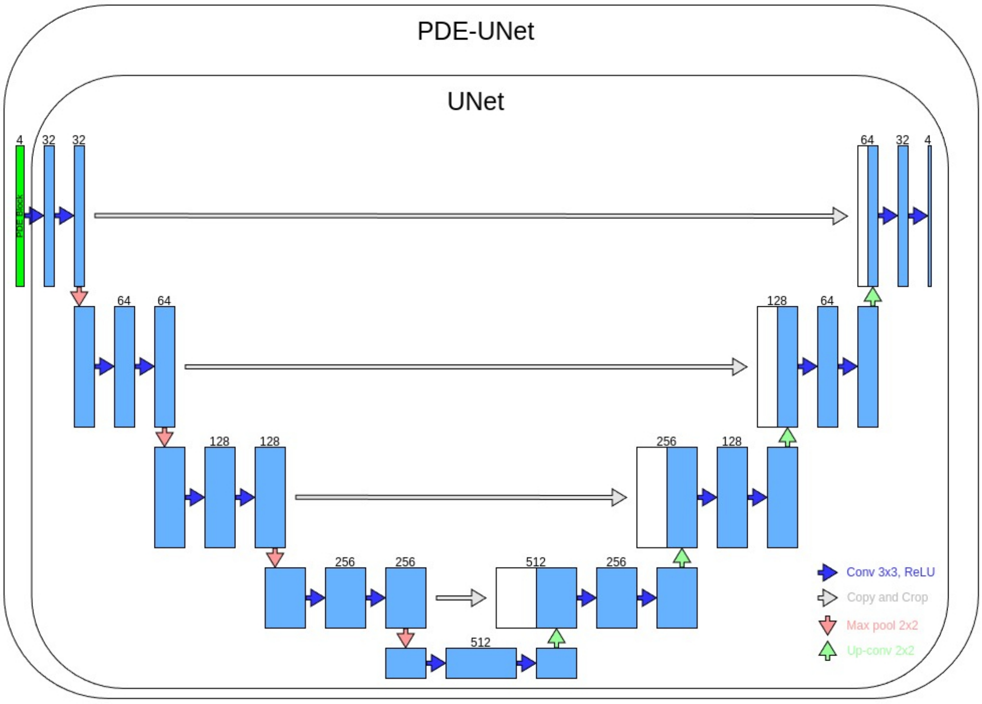

Fig. 3. The UNet and PDE-UNet architectures.

For the optimization process, the Adam optimizer [17] was used with a one-cycle learning rate scheduler [18]. As a loss function, a combo loss [19] combining the cross-entropy loss [20] and the dice loss [21] was used to optimize the medical image segmentation model. The cross-entropy provides stable and smooth gradient propagation, facilitating consistent optimization, while the dice loss effectively handles class imbalance problems and improves the segmentation of small regions.

The Cross-Entropy loss is defined as:

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}), \tag{3}$$

where N is the number of pixels, C is the number of classes, $y_{i,c}$ is the ground-truth label for class c at pixel i, and $\hat{y}_{i,c}$ is the predicted probability for class c at pixel i.

The Dice loss is defined as:

$$\mathcal{L}_{Dice} = 1 - \frac{2\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\hat{y}_{i,c} + \epsilon}{\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}^2 + \sum_{i=1}^{N}\sum_{c=1}^{C} \hat{y}_{i,c}^2 + \epsilon}, \tag{4}$$

where ϵ is a small constant to avoid division by zero.

The final loss function is the sum of both

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{Dice}, \tag{5}$$

Training was performed for 10 epochs with a batch size of 8. The dataset was divided into training, validation, and test sets in an 80%-10%-10% split. We applied several data augmentation techniques to increase the diversity and robustness of our training dataset. These techniques included random horizontal and vertical flips, rotations, affine transformations (shifts and scaling), and adjustments to brightness and contrast. Such augmentations are commonly used in medical image segmentation tasks to reduce overfitting and improve model generalization [22]. Model performance was evaluated using the Dice coefficient and Intersection over Union (IoU) metrics.

4. Results

The training progress of both models, UNet and PDE-UNet, was analyzed by monitoring the evolution of the training and validation losses. Both the baseline and the proposed models showed steady convergence during the training process. The PDE-UNet reached a lower validation loss compared to the standard UNet. Fig. 4. illustrates the progression of loss for both models over epochs.

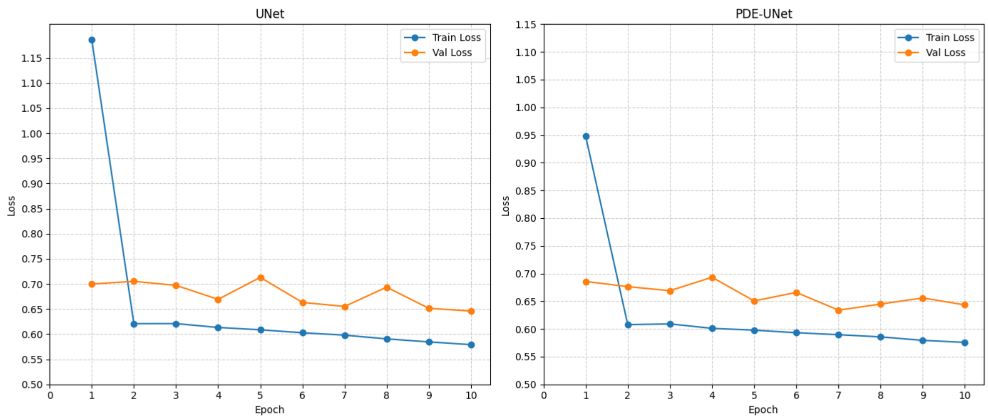

Fig. 4. Left: The training process of the UNet. Right: The training process of the proposed PDE-UNet.

Table 1. Results of Dice coefficient and IoU metrics evaluated on the validation and test sets for the UNet and PDE-UNet models

Model	**Validation**		**Test**		**Parameter count**
	Dice	**IoU**	**Dice**	**IoU**	
UNet	0.438	0.389	0.455	0.372	7,766,372
PDE-UNet	0.462	0.406	0.469	0.394	7,766,628

We evaluated segmentation performance using the Dice coefficient and IoU metrics. The evaluation was performed using the checkpoint from epoch 10 for UNet and epoch 7 for

PDE-UNet. The results are presented in Table 1. On average, the PDE-UNet achieved higher Dice and IoU scores than the baseline UNet, confirming its potential advantage in capturing structural details.

Fig. 5. shows segmentation outputs of both models alongside the ground truth annotation for the represented case. As we can see, the PDE-UNet produces more accurate and contiguous segmentations.

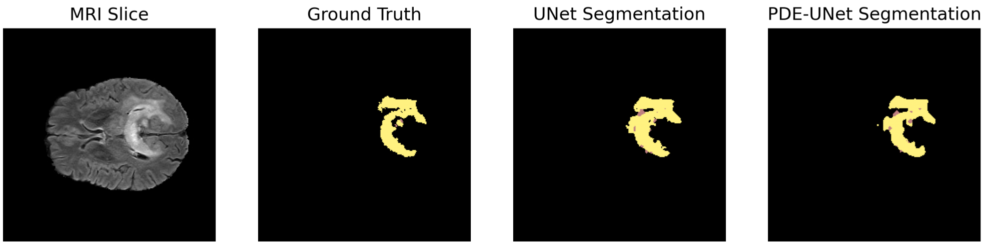

Fig. 5. The segmentation comparison between models.

It is interesting to observe how our trained PDE preprocessing block processes the images. Fig. 6. shows its effect on the sample images.

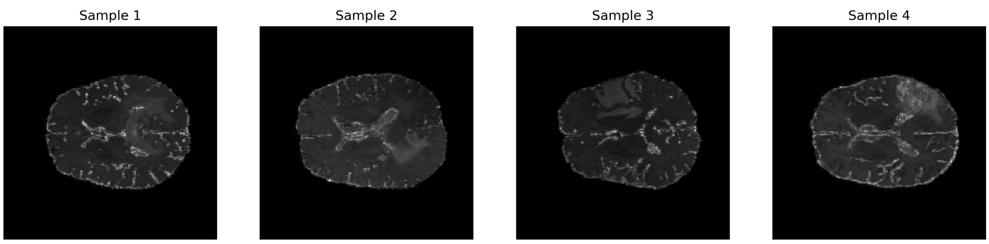

Fig. 6. Results of applying the trained PDE block to sample images.

Overall, these results demonstrate that the PDE-UNet with a PDE-inspired convolutional block enhances the representational power of the traditional UNet, leading to improved segmentation accuracy on the BraTS2020 dataset while adding only a small number of trainable parameters; therefore, the inference time of the model remains almost the same.

5. Conclusion

The paper recommends a new PDE-UNet architecture for medical image segmentation tasks. Motivated by previous research presenting a PDE-inspired preprocessing block, the recommended architecture includes an additional PDE block compared to the traditional UNet. The experiments were conducted on brain tumor segmentation using the BraTS2020 dataset. A comparative evaluation between PDE-UNet and the UNet demonstrates that the lightweight preprocessing block added to a PDE-UNet, with a few additional parameters, can positively impact the final segmentation quality, improving the UNet models. For future work, it would be interesting to extend the proposed block's investigation to other medical image analysis applications.

References

[1] S. Ur Rehman, S. Tu, Z. Shah, J. Ahmad, M. Waqas, O. Rehman, A. Kouba, Q. Abbasi, "Deep Learning Models for Intelligent Healthcare: Implementation and Challenges", *Proceedings of the Artificial Intelligence and Security, 7th International Conference, ICAIS 2021*, Dublin, Ireland, pp. 214-225, 2021.

[2] X. Tang, "The role of artificial intelligence in medical imaging research", *BJR Open*, vol. 2, no. 1, 2019.

[3] F. Guichard, L. Moisan, J.-M. Morel, "A review of P.D.E. models in image processing and image analysis", *Journal De Physique Iv - J PHYS IV*, vol. 12, no. 1, pp. 137-154, 2002.

[4] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629639, 1990.

[5] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms", *Physica D: Nonlinear Phenomena*, vol. 60, no. 14, pp. 259268, 1992.

[6] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting", *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 417424, 2000.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation". *Medical Image Computing and Computer-Assisted Intervention MICCAI 2015*, pp. 234241. Springer, 2015.

[8] (2025) BraTS2020 Dataset. [Online].Available: https://www.kaggle.com/datasets/awsaf49/brats20-dataset-training-validation

[9] L. Ruthotto, E. Haber, "Deep Neural Networks Motivated by Partial Differential Equations", *Journal of Mathematical Imaging and Vision*, pp. 352-364, 2020.

[10] Y.Sun, L. Zhang,H. Schaeffer "NeuPDE: Neural Network Based Ordinary and Partial Differential Equations for Modeling Time-Dependent Data", *Proceedings of The First Mathematical and Scientific Machine Learning Conference*, pp. 352-372, 2020.

[11] K. Qi, W. Yang, Y. L. Z. Huang, "Image Segmentation via Variational Model Based Tailored UNet: A Deep Variational Framework", *CoRR*, 2025.

[12] R. Veziryan, R. Khachatryan, "Convolutional Neural Network (CNN) Layer Development for Effectiveness of Classification Tasks". *Mathematical Problems of Computer Science*, vol. 60, pp. 6371, 2023.

[13] P. C. Bressloff, *Waves in Neural Media, ser.* Lecture Notes on Mathematical Modeling in the Life Sciences, New York, USA, Springer, 2014.

[14] A. R. Mitchel and D. F. Griffiths, "The Finite Difference Method in Partial Differential Equations", 1980.

[15] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. P. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, D. Rueckert, "Attention U-Net: Learning Where to Look for the Pancreas", *Proceedings of the Medical Imaging with Deep Learning (MIDL 2018)*, Amsterdam, Netherlands, 2018.

[16] J. Chen, J. Mei, X. Li, Y. Lu, Q. Yu, Q. Wei, X. Luo, Y. Xie, E. Adeli, Y. Wang, M. Lungren, S. Zhang, L. Xing, L. Lu, A. Yuille, Y. Zhou, “TransUNet: Rethinking the U-Net architecture design for medical image segmentation through the lens of transformers”, *Medical Image Analysis*, vol. 97, 103280, 2024.

[17] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization, *Proceedings of the 3rd International Conference for Learning Representations (ICLR)*, San Diego, CA, USA, 2015.

[18] L. N. Smith, “Cyclical Learning Rates for Training Neural Networks”, *IEEE Winter Conference on Applications of Computer Vision (WACV)*, Santa Rosa, CA, USA, pp. 464-472, 2017.

[19] S. Taghanaki et al., “Combo Loss: Handling Input and Output Imbalance in Multi-Organ Segmentation”, *Computerized Medical Imaging and Graphics*, vol. 75, pp. 24-33, 2019.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors”, *Nature*, vol. 323, no. 6088, pp. 533536, 1986.

[21] O. Kodym, M. Spanel, and A. Herout, “Segmentation of head and neck organs at risk using CNN with batch dice loss”, *Proceedings of the 40th German Conference, GCPR 2018*, Stuttgart, Germany, 2018.

[22] P. Chlap, M. Hang, N. Vandenberg, J. Dowling, L. Holloway, and A. Haworth, “A review of medical image data augmentation techniques for deep learning applications”, *Journal of Medical Imaging and Radiation Oncology*, vol. 65, no.5, pp. 545-563, 2021.

PDE-UNet: Փոփոխված UNet ճարտարապետություն՝ կիրառված բժշկական պատկերների սեգմենտավորման համար

Ռաֆայել Մ. Վեզիրյան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ, Երևան, Հայաստան
e-mail: rafaelveziryan@gmail.com

Ամփոփում

Բժշկական պատկերների սեգմենտացիան կարևորագույն խնդիր է առողջապահության ոլորտում, մասնավորապես հիվանդությունների հայտնաբերման և բուժման պատշաճ պլանավորման համար: Խորը ուսուցման մոդելները բարձր արդյունքների են հասնում բժշկական պատկերի վերլուծության մեջ: Այս հոդվածը ներկայացնում է նոր PDE-UNet ճարտարապետության արդյունավետությունը, որը ոգեշնչված է նեյրոնային ցանցերում մասնակի դիֆերենցիալ հավասարումների (ՄԴՀ) կիրառություններով, բժշկական պատկերի սեգմենտացիայի արդյունավետությունը բարելավելու համար: Փորձարկումները կատարվել են BraTS2020 տվյալների հավաքածուից վերցված ուղեղի ուռուցքի ՄՌ-Տ պատկերների վրա և համեմատվել են ավանդական UNet ճարտարապետության հետ:

Բանալի բառեր՝ բժշկական պատկերների սեգմենտացիա, ուղեղի ուռուցքի սեգմենտացիա, UNet, PDE-ով ոգեշնչված CNN բլոկ, MICCAI BraTS 2020 մարտահրավեր:

PDE-UNet: Модифицированная архитектура UNet, применяема для сегментации медицинских изображений

Рафаэль М. Везирян

Институт проблем информатики и автоматизации НАН РА, Ереван, Армения
e-mail: rafaelveziryan@gmail.com

Аннотация

Сегментация медицинских изображений критически важная задача в здравоохранении, особенно для диагностики заболеваний и планирования надлежащего лечения. Модели глубокого обучения достигают высокой эффективности при анализе медицинских изображений. В данной статье представлена эффективность новой архитектуры PDE-UNet, основанной на применении уравнений в частных производных (УЧП) в нейронных сетях, для повышения эффективности сегментации медицинских изображений. Эксперименты проводились на МРТ-изображениях опухолей головного мозга из набора данных BraTS2020 и сравнивались с традиционной архитектурой UNet.

Ключевые слова: сегментация медицинских изображений, сегментация опухолей головного мозга, UNet, блок CNN на основе PDE, конкурс MICCAI BraTS 2020.

Mathematical Problems of Computer Science 64, 56–65, 2025.

doi:10.51408/1963-0140

UDC 004

Development of the Automated Alert System for Detection and Handling of Compromised Email Accounts

Arthur S. Petrosyan, Gurgen S. Petrosyan and Robert N. Tadevosyan

Institute for Informatics and Automation Problems of NAS RA, Yerevan, Armenia
e-mail: arthur@sci.am, gurgen@sci.am, robert@sci.am

Abstract

With the increasing volume of email-based attacks and unauthorized access to mail servers, the need for automated monitoring and response mechanisms has become essential. This paper presents the development of an automated alert system designed to detect and handle compromised email accounts. The system monitors the mail server queue on a Linux server, detecting anomalies based on a significant surge in queued messages. Upon identifying suspicious activity, the system attempts to determine the username associated with the highest number of SASL authentications and triggers appropriate alerts or mitigation actions. Additionally, the system is integrated with a Telegram bot, allowing administrators to take immediate corrective actions remotely. This approach provides a lightweight, effective method for preventing email abuse and ensuring the integrity of email servers.

Keywords: Email, SPAM, Alert system, Telegram bot**.**

Article info: Received 25 April 2025; sent for review 2 May 2025; accepted 13 November 2025.

1. Introduction

Mail servers are common targets for malicious attackers seeking to exploit compromised accounts for spam and phishing campaigns. Once an email account is compromised, it can be used to send tens of thousands of spam emails before the breach is detected. Such activities can lead to unwanted load for the mail system, blacklisting of mail servers, reputational damage for the whole origin network, and security breaches. Traditional security mechanisms such as rate-limiting, IP-based restrictions, fail2ban [1], Rspamd [2] or Spamassassin [3] offer some level of protection, but they are not enough for immediate detection and mitigation of compromised email accounts.

This paper presents the implementation of an automated alert system that continuously monitors mail server queues and detects unusual spikes in queued messages. The system uses SASL authentication [4] logs to identify the account responsible for excessive email submissions. Additionally, it is integrated with a Telegram bot API [5], enabling administrators to receive alerts and take mitigation actions remotely. By automating the detection and response process, the proposed system minimizes response time by enabling immediate intervention, thus preventing further misuse of compromised accounts and reducing the impact of account compromises.

2. Related work

Several existing solutions aim to detect compromised email accounts, including:

- Intrusion detection systems (IDS) [6];
- Rate-limiting mechanisms [7];
- Reputation-based filtering using Domain Name System blocklists (DNSBL) [8];

However, these methods often rely on external data sources or predefined thresholds, which may not adapt to evolving attack patterns.

The system proposed in this paper offers a real-time, in-server detection mechanism that is independent of third-party services. It is not intended as a replacement for other solutions, but as an additional level of protection for the mail server.

3. Postfix considered as MTA

In this paper, we consider a mail server to run based on Postfix [9]. Postfix is one of the most widely used mail transfer agents (MTAs) for handling email delivery on Unix-like operating systems. Postfix was originally developed by Wietse Venema as a secure alternative to the standard UNIX MTA - Sendmail. Formerly known as Vmailer, Postfix was released by the end of 1998 as the IBM Secure Mailer and later renamed to Postfix [10, 11].

Postfix is known for its performance, flexibility, and ease of configuration. It supports various authentication mechanisms, filtering capabilities, and queue management features, making it a preferred choice for mail servers ranging from small deployments to enterprise-level environments. Due to its modular design, Postfix integrates well with security tools, spam filters, and monitoring systems, ensuring reliable and efficient email processing.

4. System Design and Implementation

The automated alert system is running regular checks for a sudden increase in the mail queue size. In case an unusual number of messages is detected in the queue, additional checks are performed to identify the cause of the anomaly. Most often, this indicates that the password of an email address has been hacked (usually by brute force) and spam has been sent from that address. If the assumption is confirmed by additional checks of log files, the system triggers an alert and takes emergency steps to prevent further damage. Details of the system are described below.

4.1 Mail Queue Monitoring

Postfix manages email delivery through a structured mail queue system, which consists of several directories where messages are temporarily stored during processing. The main queue directories include:

- **Incoming Queue** (*/var/spool/postfix/incoming*) – Stores new messages before they are processed by Postfix.
- **Active Queue** (*/var/spool/postfix/active*) – Contains messages that are actively being delivered. Postfix prioritizes these messages for immediate processing.
- **Deferred Queue** (*/var/spool/postfix/deferred*) – Holds messages that could not be delivered on the first attempt. Postfix retries delivery at scheduled intervals.
- **Maildrop Queue** (*/var/spool/postfix/maildrop*) – Used for messages submitted locally by Postfix-compatible senders.
- **Hold Queue** (*/var/spool/postfix/hold*) – Stores messages that require manual intervention before being processed further.

Each message in these directories is stored as a separate file with metadata that helps Postfix track its delivery status. The queue management system ensures efficient handling of emails, preventing congestion and optimizing delivery performance. Administrators can monitor and manage the queue using commands such as "*postqueue -p*" to list queued messages.

Since our goal is to detect spam outbreaks, which typically result in a sudden surge of emails being queued, monitoring the mail queue size is a crucial step. A basic way to achieve this is by using the "*postqueue -p*" command, which lists all queued messages and allows administrators to assess the queue status. Thus, the number of messages in the queue can be obtained at any time with a command sequence like "*postqueue -p | tail -n +2 | wc -l*".

However, this approach requires parsing command output, which can be inefficient, especially during a large-scale spam event when thousands (or even tens of thousands) of messages may be queued. To improve performance, we opt for a more efficient method - directly counting the number of files in the Postfix queue directories (Incoming Queue, Active Queue, Deferred Queue and Maildrop Queue). This approach eliminates the overhead of command execution and text processing, providing a faster and more reliable way to detect email surges in real-time.

It should be noted that we do not need to check the Hold Queue *(/var/spool/postfix/hold*) because messages in this directory are placed there manually or through policy-based filtering and are not actively processed for delivery. Unlike all other Postfix queue directories, which reflect real-time email flow, the hold queue contains messages that require administrative intervention before they can proceed. Since a spam outbreak results in a rapid increase in automatically queued messages, monitoring the hold queue would not provide meaningful insights into the outbreak's severity or impact. Instead, focusing on the other queues ensures that we detect and respond to excessive email generation more efficiently.

4.2 Periodic checks

To ensure continuous checks, the monitoring script should run periodically as either a Cron job [12] or a Systemd service timer [13], providing real-time detection of spam outbreaks without significant resource usage. Since it only performs lightweight operations - counting files in the mail queue and parsing logs - it runs efficiently without impacting system performance.

4.2.1 Monitoring with Cron Job

Generally, configuration with a Cron job is easier and sufficient. It only requires adding a line to the root user's crontab configuration, like the following example:

**/2 * * * * /path/to/script.sh*

As a result, the script will be continuously executed every 2 minutes.

But there is an important detail about Cron jobs that should be noted forimportant monitoring actions like the one described in this work. Running of any Cron job always depends on the cron service, which is a separate process (typically named cron or crond, depending on the Linux distribution), responsible for running scheduled tasks. So, if crond/cron accidentally stops or crashes, none of the scheduled jobs would run until the service is restarted. That is why the next systemd service timer-based solution is more preferable for our task.

4.2.2 Monitoring with Systemd Service

The key advantage of systemd timers over cron jobs is that they are managed by the systemd process. And since systemd is the process number one in the Linux system, it is always running as long as the system is up. This would ensure that any scheduled script will execute reliably. Even if a timer fails, the systemd process can restart it automatically.

However, configuring systemd timer requires much more efforts than just adding a line in Cron configuration. Below, we present an example of such a configuration.

Creating a systemd service file (*/etc/systemd/system/spam-check.service*):

```
[Unit]
Description=Spam Outbreak Detection Script
After=network.target
[Service]
ExecStart=/path/to/script.sh
Restart=always
User=root
[Install]
WantedBy=multi-user.target
```

Creating a timer *(/etc/systemd/system/spam-check.timer*):

```
[Unit]
Description=Run spam check script every minute
[Timer]
OnBootSec=2min
OnUnitActiveSec=2min
[Install]
WantedBy=timers.target
```

Enabling and starting the timer:

systemctl enable --now spam-check.timer

The above *systemd* timer configuration ensures the script runs at regular intervals, detecting anomalies promptly while maintaining system efficiency.

It should be noted that the checking frequency of every 2 minutes mentioned above was given as an example, but it can be adjusted according to need. At the same time, it is important to mention the fact that the log file check (which can take a considerable amount of time) is performed only at the second stage, in case of detection of exceeding the threshold of the mail queue size. This means that in most cases, the check time will be very insignificant. We have checked this assumption by running the script (with the "time" prefix) to determine its execution time, and on average, we found it to be about 12-17 milliseconds. Thus, frequent script launching has virtually no effect on system load while at the same time ensuring timely response to anomalies.

4.3 Identifying the Compromised Account

Identifying the compromised account is completed at a second stage only if the mail queue size is detected to exceed the defined threshold. This is accomplished by analyzing the mail server log file. On each run, the script parses SASL authentication messages in the current mail server log file to determine which user has authenticated most frequently within a recent timeframe (on most Linux systems the default place for such messages is */var/log/mail.log*).

Log files are generally subject to rotation,which are most often performed either on a weekly or monthly basis. In any case, the checks described below reveal the most recent data for the period after the last file rotation. The command sequence used for the checks is presented below:

```
cat /var/log/mail.log | awk -F"sasl_username=" '{print $2}' | sort | uniq -c | sort -nr | head -6 | tail -5
```

The result is a list of sorted top 5 authentications from the last period. In case of a spam outbreak, the first line (or the first few lines) will have a comparatively very large number of successful authentications, which is a proof of a potentially compromised account.

An example of the alert message is presented below:

```
Mail server queue has exceeded the specified threshold.
Current queue size is: 12869
Top 5 SASL authentications from /var/log/mail.log:
 15628 [compromised-mail-address]@somedomain
     234  [other-mail-address1]
     122 [other-mail-address2]
     104 [other-mail-address3]
      86  [other-mail-address4]
```

In case such an anomaly is detected, the system immediately flags that email account as potentially compromised, notifies the administrator of the problem, and provides the opportunity to take preventive actions, which are described below.

4.4 Telegram Bot Integration

Integration with Telegram Bot is performed according to Telegram Bot API Documentation [5]. The script is integrated with the Telegram Bot to send alerts. Administrators receive messages with details such as the compromised account, suspicious activity, and recommended actions.

An example of a script to implement integration with a Telegram Bot is presented below:

```
#!/bin/bash
tgbot="********************"
tgchatid="*****************"
tgurl="https://api.telegram.org/bot"$tgbot"/sendMessage"
QUEUEDIR_ROOT="/var/spool/postfix"
MAX_QUEUE_LENGTH=***
# Get the number of messages sitting in each postfix queue directory
Q_ACTIVE=$(find ${QUEUEDIR_ROOT}/active -type f | wc -l)
Q_INCOMING=$(find ${QUEUEDIR_ROOT}/incoming -type f | wc -l)
Q_DEFERRED=$(find ${QUEUEDIR_ROOT}/deferred -type f | wc -l)
Q_MAILDROP=$(find ${QUEUEDIR_ROOT}/maildrop -type f | wc -l)
# If any of these queues contain more than $MAX_QUEUE_LENGTH issue an alert
if [ ${Q_ACTIVE} -gt ${MAX_QUEUE_LENGTH} -o ${Q_INCOMING} -gt
${MAX_QUEUE_LENGTH} -o ${Q_DEFERRED} -gt ${MAX_QUEUE_LENGTH} -o
${Q_MAILDROP} -$
Q_TOP5=$( cat /var/log/mail.log | awk -F"sasl_username=" '{print $2}' | sort | uniq -c | sort -nr | head -
6 | tail -5)
tgmessage="Queue on ${HOSTNAME} is: $((${Q_ACTIVE} + ${Q_INCOMING} + ${Q_DEFERRED}
+ ${Q_MAILDROP}))"$'\n'$Q_TOP5
curl -s -d
"chat_id="$tgchatid"&text=$tgmessage&parse_mode=markdown&disable_web_page_preview=1"
$tgurl > /dev/null 2>&1
exit 2
fi
exit 0
```

4.5 Mail Queue Clean up

Since Postfix does not provide a ready mechanism to remove messages from the queue based on the sender's email address, we use the following solution to clean up the mail queue, based on a compromised email address pattern:

```
#!/bin/bash
# Define the email pattern to search for
ADDRESS_PATTERN="$1"
if [[ -z "$ADDRESS_PATTERN" ]]; then
   echo "Usage: $0 '<email-pattern>'"
   echo "Example: $0 'spam@domain.com' or $0 '@domain.com'"
   exit 1
fi
# Get queue IDs for messages matching the pattern
QUEUE_IDS=$(postqueue -p | grep -B1 "$ADDRESS_PATTERN" | grep '^[A-F0-9]' | cut -d' ' -f1)
# Check if any queue IDs were found
if [[ -z "$QUEUE_IDS" ]]; then
   echo "No matching emails found in the queue."
   exit 0
fi
# Remove matching emails
echo "Removing emails matching pattern: $ADDRESS_PATTERN"
echo "$QUEUE_IDS" | xargs -r postsuper -d
echo "Done."
```

4.6 Mitigation Actions

Upon detecting a compromised account, the alert system allows administrators to take predefined mitigation actions via Telegram bot, including:

- stop/start mail service;
- mail queue is cleanup (remove queued mails form the compromised email address);
- lock/unlock/generate new password for compromised email account.

Besides notifications and managing with the Telegram bot, the system can also send alerts via email.

5. Results and Evaluation

The described alert system was tested on a Linux-based mail server running Postfix. The key findings include:

- Compromised accounts are successfully detected in real-time within several minutes after spam outbreak, significantly reducing response times compared to manual detection.
- The system prevented the mail server from being blacklisted by proactively blocking malicious email bursts.
- The Telegram bot significantly improved response times by allowing administrators to take action remotely.
- Low resource usage made it feasible the system to run constantly without impacting mail server performance.
- Solution to check Postfix mail queue directories is more efficient than using "postqueue -p" command, as it avoids parsing output and provides an immediate count of queued messages, which is crucial in case of large-scale spam outbreak, involving thousands or even tens of thousands of spam emails suddenly filling up the mail queue.
- Solution to configure systemd timers instead of Cron job ensures checks do not depend on the cron process existance.

6. Conclusion

The developed automated alert system provides an effective solution for detecting and handling compromised email accounts. It significantly improved detection accuracy and reduced administrative workload. At the same time, it is not a replacement for standard protection methods, but rather complements them perfectly.

By leveraging real-time mail queue monitoring and SASL authentication analysis, it enhances email security while minimizing the need for manual intervention. The integration of a Telegram bot further improves response efficiency, allowing administrators to act quickly during a spam outbreak.

The system described in this paper, has been successfully tested on the Academic Scientific Research Computer Network of Armenia (ASNET-AM) [14] and is currently being actively used for implementation of set goals.

References

[1] Fail2Ban: ban hosts that cause multiple authentication errors. [Online]. Available: https://github.com/fail2ban/fail2ban

[2] Rspamd,Fast, free and open-source spam filtering system. [Online]. Available: https://rspamd.com

[3] Apache SpamAssassin, Open Source anti-spam platform. [Online]. Available: https://spamassassin.apache.org/

[4] SASL Authentication Mechanism, [Online]. Available: https://www.cyrusimap.org/sasl

[5] Telegram Bot API Documentation, [Online]. Available: https://core.telegram.org/bots/api

[6] What is an intrusion detection system (IDS)? [Online]. Available: https://www.ibm.com/think/topics/intrusion-detection-system

[7] Postfix Performance Tuning. [Online]. Available: https://www.postfix.org/TUNING_README.html

[8] Domain Name System blocklist. [Online]. Available: https://en.wikipedia.org/wiki/Domain_Name_System_blocklist

[9] Postfix MTA. [Online]. Available: http://www.postfix.org

[10] Sharing Software, IBM to Release Mail Program Blueprint. [Online]. Available: https://archive.nytimes.com/www.nytimes.com/library/tech/98/12/biztech/articles/14blue.html

[11] Postfix free and open-source mail transfer agent (MTA). [Online]. Available: https://en.wikipedia.org/wiki/Postfix_(software)

[12] Cron job scheduler. [Online]. Available: https://en.wikipedia.org/wiki/Cron

[13] Systemd. [Online]. Available: https://en.wikipedia.org/wiki/Systemd

[14] The Academic Scientific Research Computer Network of Armenia. [Online]. Available: (ASNET-AM) https://asnet.am

Էլեկտրոնային փոստի կոտրած հաշվեհամարների հայտնաբերման և կառավարման ահազանգման ավտոմատացված համակարգի մշակում

Արթուր Ս. Պետրոսյան, Գուրգեն Ս. Պետրոսյան և Ռոբերտ Ն. Թադևոսյան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ, Երևան, Հայաստան
e-mail: arthur@sci.am, gurgen@sci.am, robert@sci.am

Ամփոփում

Էլեկտրոնային փոստի վրա հիմնված հարձակումների և փոստային սերվերների նկատմամբ ոչ օրինական մուտքերի աճի ֆոնին կարևոր է ունենալ ավտոմատացված մոնիթորինգի և արձագանքման մեխանիզմներ: Այս հոդվածում ներկայացված է նման համակարգի մշակում, որն ուղղված է կոտրված էլ. փոստի հաշվեհամարների բացահայտման և վերահսկման խնդիրներին: Համակարգը մոնիթորինգ է իրականացնում Linux համակարգում գտնվող փոստային սերվերի հերթի չափի վրա՝ հերթի մեջ նամակների քանակի անսովոր կտրուկ աճ հայտնաբերելով: Նման դեպքում համակարգը փորձում է պարզել այն օգտատիրոջ անունը, որն ունի ամենաշատ SASL նույնականացումներ և կատարում է համապատասխան կանխարգելիչ գործողություններ: Բացի այդ, համակարգը ինտեգրված է Telegram բոթի հետ՝ հնարավորություն տալով ադմինիստրատորներին հեռավար կերպով կատարել շտկող գործողություններ: Այս մոտեցումը ապահովում է պարզ և արդյունավետ մեթոդ՝ կանխարգելելու էլ. փոստի չարաշահումը և ապահովելու փոստային սերվերների անվտանգությունը:

Բանալի բառեր՝ Email, SPAM, Alert system, Telegram bot.

Разработка автоматизированной системы оповещения для обнаружения и обработки взломанных учетных записей электронной почты

Артур С. Петросян, Гурген С. Петросян и Роберт Н. Тадевосян

Институт проблем информатики и автоматизации НАН РА
e-mail: arthur@sci.am, gurgen@sci.am, robert@sci.am

Аннотация

С ростом объема атак на электронную почту и несанкционированного доступа к почтовым серверам потребность в автоматизированных механизмах мониторинга и реагирования становится все более существенной. В этой статье представлена разработка автоматизированной системы оповещения, предназначенной для обнаружения и обработки взломанных учетных записей электронной почты. Система отслеживает очередь почтовой службы на сервере Linux, обнаруживая аномалии на основе резкого всплеска количества сообщений в очереди. При выявлении подобной активности система пытается определить имя пользователя, связанное с наибольшим количеством аутентификаций SASL, и запускает соответствующие оповещения или действия по устранению последствий. Кроме того, система интегрирована с ботом Telegram, что позволяет администраторам удаленно предпринимать немедленные корректирующие действия. Такой подход обеспечивает быстрый и эффективный метод предотвращения злоупотреблений электронной почтой и обеспечения целостности почтовых серверов.

Ключевые слова: Email, SPAM, Alert system, Telegram bot.

Mathematical Problems of Computer Science 64, 66–75, 2025.

doi:10.51408/1963-0141

UDC 004.8:004.72

Infrequent Synchronization in Distributed AdaBoost

Arthur K. Oghlukyan[1] and Luis Fernando de Mingo López[2]

[1]Institute for Informatics and Automation Problems of NAS RA, Yerevan, Armenia
[2]Polytechnic University of Madrid, Ispain, Madrid
e-mail: artur.oghlukyan@edu.isec.am, fernando.demingo@upm.es

Abstract

Distributed machine learning has become increasingly vital as data sources continue to expand geographically. Traditional ensemble methods such as AdaBoost demonstrate impressive predictive capabilities but often require frequent synchronization across nodes, resulting in significant communication overhead. This paper introduces a novel paradigm of **infrequent synchronization** in which nodes perform multiple rounds of local AdaBoost before exchanging partial or complete model updates. The potential advantages include reduced communication costs, the ability to handle intermittent connectivity, and competitive accuracy compared to fully synchronized approaches. A real-world use case in the trucking industry is presented to demonstrate the feasibility and value of this new approach. The paper concludes by outlining future directions and the expected impact on communication-efficient distributed learning.

Keywords: Distributed AdaBoost, Infrequent Synchronization, Ensemble Learning, Communication-Efficient Learning, Federated Boosting, Weak Learners, Scalability, Fault Tolerance, Real-World Deployment.

Article info: Received 16 May 2025; sent for review 5 June 2025; accepted 7 October 2025.

Acknowledgments: The authors wish to thank contributors from various open-source distributed ML projects for their insights into scalable ensemble frameworks.

1. Introduction

1.1 Background

The evolution of big data analytics has necessitated the adoption of distributed machine learning frameworks that can operate across geographically dispersed nodes. Ensemble algorithms, particularly AdaBoost [1], stand out for their ability to transform weak learners into a strong classifier through iterative reweighting of training examples. Yet, the classical distributed deployment of AdaBoost relies on frequent data or model exchanges, often every boosting round to maintain a coherent global model [2]. This can be problematic in scenarios where communication is costly, bandwidth is limited, or connectivity is intermittent [3].

1.2 Problem Statement

This research explores a delayed or infrequent synchronization strategy in distributed AdaBoost to minimize communication overhead without substantially sacrificing model accuracy [4]. Specifically, it examines whether AdaBoost's iterative process can be adapted to function effectively under limited exchange conditions, an area partially explored in [5, 6] but remaining relatively undeveloped for classical boosting approaches.

1.3 Research Objectives

1. **Reduced Communication Overhead**: Demonstrate how the frequency of synchronization rounds can be lowered to a fraction of the total boosting iterations while still retaining high model accuracy.
2. **Adaptive Local Training**: Investigate how local reweighting schemes can operate in isolation for multiple iterations to mitigate the stale model challenge.
3. **Real-World Feasibility**: Illustrate a use case where intermittent connectivity is common, namely the trucking industry, to validate the proposed method's applicability and benefits.

2. Related Work

2.1 Communication-Efficient Ensemble Methods

Studies in communication-efficient boosting [7] and local update strategies [8] establish that sparse or delayed parameter sharing can preserve accuracy under theoretical guarantees. Distributed boosting algorithms such as PreWeak and AdaSampling reduce synchronization overhead by transmitting partial updates or sampled data to a central coordinator [9]. Nonetheless, these methods often necessitate specialized sampling or model pruning to ensure convergence.

2.2 Federated Learning Paradigms

Federated learning (FL) promotes infrequent synchronization, a property particularly valuable in privacy-sensitive industries such as healthcare and finance [10]. Techniques like Federated Averaging (FedAvg) allow clients to perform multiple local gradient-descent updates before transmitting aggregated model parameters to a central coordinator, thereby substantially reducing communication overhead [5]. Although the underlying motivation of decreasing communication is shared, FedAvg and distributed AdaBoost differ fundamentally in their learning dynamics. FedAvg operates on parametric, differentiable models whose parameters can be averaged meaningfully across clients. In contrast, distributed AdaBoost aggregates weak learners through weighted voting rather than parameter averaging, and its communication steps revolve around synchronizing model weights, error rates, or classifier outputs rather than gradient-based updates. This distinction highlights that, while both paradigms benefit from reduced communication frequency, the mechanisms enabling synchronization in distributed AdaBoost require algorithm-specific coordination rather than straightforward parameter averaging, motivating the development of tailored synchronization strategies.

2.3 Distributed Systems and AdaBoost Synchronization

Distributed systems rely on principles such as scalability, consistency, and fault tolerance [11]. Achieving these principles often involves trade-offs; scalable systems commonly sacrifice strict consistency to ensure fault tolerance and availability [12]. Ensemble methods like AdaBoost face unique synchronization challenges due to their sequential nature; each iteration requires updated global error rates to adjust sample weights, typically necessitating synchronous communication [1, 13].

In contrast to federated learning's flexibility in synchronization (e.g., asynchronous and hybrid approaches such as FedBuff), distributed AdaBoost methods commonly enforce synchronous communication rounds [14]. Empirical analyses, such as LoAdaBoost FedAvg, illustrate AdaBoost's integration within federated frameworks by combining synchronous aggregation with adaptive weighting strategies to enhance accuracy and efficiency [15]. However, AdaBoost's inherent sequential dependencies limit the potential for asynchronous updates, reinforcing the necessity of carefully managed synchronization mechanisms [16].

3. Proposed Methodology

The primary goal is to minimize synchronization events in a distributed AdaBoost framework while still preserving sufficient model accuracy. Each node locally runs multiple rounds of AdaBoost on its partition of data, updating instance weights and training weak learners without requiring continuous global communication [7, 5]. Periodically, but as rarely as feasible, these partial updates are exchanged and merged into a global ensemble, aligning node-specific reweighting strategies and capturing the collective knowledge [6]. This strategy builds upon the classical boosting concept [1] yet limits the overhead typically associated with every-round synchronization. While more local iterations between communications can slightly increase the risk of model drift, careful selection of when and how often to synchronize helps balance reduced bandwidth usage with acceptable predictive performance in large-scale or bandwidth-constrained environments.

3.1 Algorithmic Framework

The novel contribution is an **infrequent synchronization strategy** for AdaBoost. Instead of synchronizing after each boosting iteration, each node trains locally for K rounds, storing partially updated models in a local buffer. After these K rounds, the node communicates with a central server or peer nodes to merge and refine the global ensemble. The process repeats until the desired number of boosting rounds is reached.

Key Insight: By decoupling local updates, the model allows partial divergence in node-specific weight distributions. Periodic global synchronization steps mitigate error accumulation and realign the ensemble.

3.2 Pseudocode

```
Algorithm: InfrequentSyncAdaBoost

Input:
   D = {D_1, D_2, ..., D_N}  // Partition of data across N nodes
   T              // Total number of boosting rounds
   K              // Frequency of synchronization
   base_learner     // Base classifier for AdaBoost

Initialize:
   For each node n:
      Initialize local sample weights w_n(i) = 1 / |D_n|

For round t = 1 to T:
   For each node n in parallel:
      1. Train weak classifier h_n(t) on D_n using current weights w_n
      2. Compute error_n(t) = Σ [w_n(i) * I(h_n(t)(x_i) ≠ y_i)]
      3. Compute α_n(t) = 0.5 * ln((1 - error_n(t)) / error_n(t))
      4. Update weights:
         w_n(i) = w_n(i) * exp(-α_n(t) * y_i * h_n(t)(x_i))
      5. Normalize w_n(i)

   If t % K == 0 or t == T:
      // Synchronize across nodes
      6. Gather {h_n(t'), α_n(t')} for t' in {t-K+1, ..., t} from each node
      7. Construct global ensemble H_g(r) by merging or aggregating these weak classifiers
      8. (Optional) Prune or refine ensemble if it becomes too large
      9. Broadcast H_g(r) to all nodes, so they can align partial states

Output:
   Final aggregated ensemble: H_g(T)
```

3.3 Complexity Analysis

- **Communication**: Reduced from O(NT) in classical distributed AdaBoost (where each node synchronizes every iteration) to O(NT / K) in the proposed infrequent scheme.
- **Computation**: Minimal additional overhead arises from merging partial ensembles, which can be done through a simple central aggregator. Local computations remain identical to standard AdaBoost.
- **Potential Trade-offs**: Longer local training phases might introduce greater divergence from the global optimum, demanding careful parameter tuning (e.g., selection of K).

4. Evaluation

4.1 Experimental Setup

Three variants of distributed AdaBoost are compared:

1. **Fully Synchronized (Baseline)**: Synchronization at every boosting round.
2. **Moderately Synchronized**: Synchronization every 5 rounds.
3. **Infrequent Synchronization**: Synchronization only once or twice (e.g., every 25 or 50 rounds).

A sample result table and visualizations based on the evaluation metrics for 100 boosting rounds is shown in Table 1.

Table 1. Synchronization analysis

Sync Strategy	**Sync Rounds**	**Communication (MB)**	**Accuracy (%)**	**TrainingTime (s)**
Fully Synchronized	100	120.0	95.4	240
Every 5 Rounds	20	24.0	94.9	180
Every 25 Rounds	4	4.8	94.2	160
Once (at End)	1	1.2	91.5	140

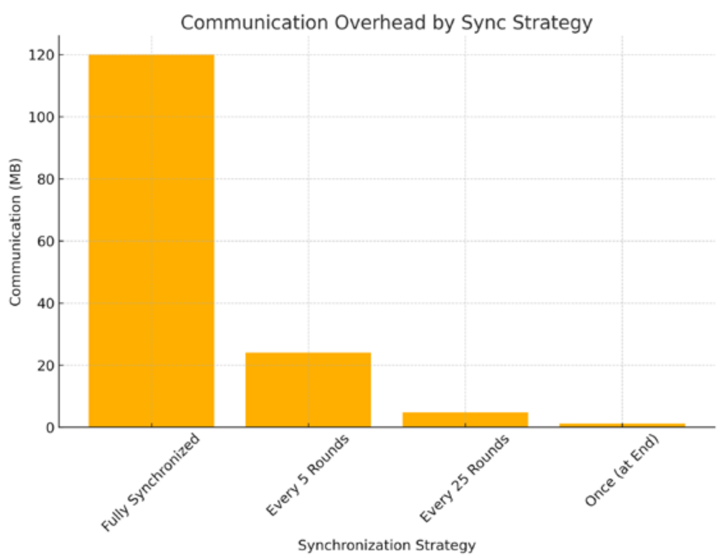

Fig. 1. Communication overhead by synchronization strategy.

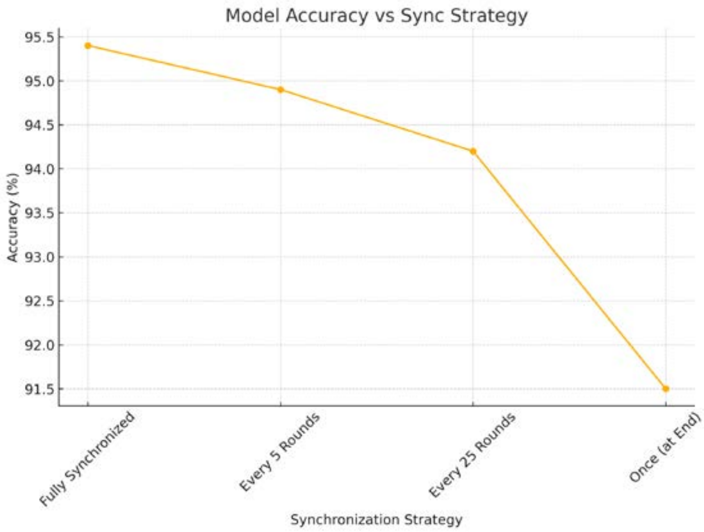

Fig. 2. Model accuracy vs Synchronization strategy.

Fig. 3. Training time by synchronization strategy.

Although the final accuracy for the “Once (at End)” synchronization strategy is lower at 91.5% compared to more frequent synchronization, it remains sufficiently high for many real-world applications. In scenarios where faster training time, reduced communication overhead, or intermittent connectivity takes priority, a slight reduction in accuracy can be acceptable. This performance level still represents a robust predictive capability, ensuring that the trade-off between model quality and limited synchronization is justifiable in numerous large-scale or bandwidth-constrained deployments.

4.2 Real-World Use Case: Trucking Industry

4.2.1 Context and Data Generation

Each truck in a large fleet is equipped with sensors gathering telematics data: fuel usage, engine temperature, speed, brake usage, GPS location, etc. Due to remote driving routes and sporadic connectivity, the trucks can only synchronize with the central office every few hours or at designated checkpoints.

4.2.2 Application

- **Predictive Maintenance**: Early detection of mechanical issues based on aggregated sensor data.
- **Fuel Efficiency Optimization**: Identifying fuel-wasting driving behaviors across different terrains.
- **Driver Safety Analysis**: Monitoring and alerting high-risk driving patterns (sharp braking, speeding).

4.2.3 Implementation

Local AdaBoost runs on each truck for multiple iterations, updating the distribution of "tricky" instances found in that truck's routes. When connectivity allows, all partial models are transmitted to a central aggregator, which merges them and redistributes global updates.

Outcome: Even with one or two synchronizations per day, fleet-wide predictive performance remains competitive, enabling near-real-time insights into vehicle health and driver habits without overwhelming the limited communication infrastructure.

5. Conclusion and Future Work

The proposed **infrequent synchronization strategy** for distributed AdaBoost addresses pressing challenges in large-scale, communication-constrained settings. Preliminary analyses indicate that local training for multiple rounds before synchronizing can significantly reduce communication overhead with a modest trade-off in convergence speed or final accuracy. Future work will involve formalizing convergence bounds, exploring adaptive synchronization schedules based on node-specific performance, and implementing privacy-preserving protocols to handle sensitive data (e.g., driver habits or proprietary operational metrics).

References

[1] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting”, *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[2] Y. Zhang and J. Huan, “Inductive multi-task learning with multiple view data”, *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 543–551, 2011.

[3] L.-M. Ang, K. P. Seng et al., “Big sensor data systems for smart cities”, *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 468–476, 2018.

[4] J. Hamer, G. Quinlan and S. Marshland, “FedBoost: Communication-efficient federated learning with boosting”, *Proceedings of the International Conference on Distributed Machine Learning*, pp. 1–8, 2020.

[5] H. B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data”, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (AISTATS), pp. 1273–1282, 2017.

[6] K. Cheng, T. Fan, Y. Jin et al., “SecureBoost: A lossless federated learning framework”, *IEEE Transactions on Big Data*, vol. 7, no. 4, pp. 776–788, 2021.

[7] M.-F. Balcan, A. Blum and S. Fine, “Communication efficient distributed learning”, *Proceedings of the 25th Annual Conference on Learning Theory*, pp. 35.1–35.22, 2012.

[8] T. Li, A. K. Sahu, M. Zaheer et al., “Federated optimization in heterogeneous networks”, *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[9] Y.-Y. Chiang, C.-J. Hsieh and I. S. Dhillon, “Communication-efficient distributed boosting algorithms”, *Proceedings of the International Conference on Machine Learning* (ICML), pp. 139–147, 2014.

[10] P. Kairouz, H. B. McMahan et al., “Advances and open problems in federated learning”, *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[11] S. Gilbert and N. Lynch, “Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services”, *ACM SIGACT News*, vol. 33, no. 2, pp. 51–59, 2002.

[12] M. Li, D. G. Andersen, A. J. Smola and K. Yu, “Communication efficient distributed machine learning with the parameter server”, *Advances in Neural Information Processing Systems*, vol. 27, pp. 19–27, 2014.

[13] A. Karbasi and K. G. Larsen, “Parallel boosting algorithms: Limitations and possibilities”, *Journal of Machine Learning Research*, vol. 25, pp. 1–34, 2024.

[14] Y. Fraboni, R. Vidal, L. Kameni and M. Lorenzi, “FedBuff: Asynchronous federated learning with buffered updates”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 1, pp. 102–114, 2023.

[15] L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng and D. Liu, “Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records”, *Journal of Biomedical Informatics,* vol. 99, p. 103291, 2020.

[16] M. Polato, A. Sperduti and F. Chierichetti, “AdaBoost.F: Federated boosting with decision trees”, *IEEE Transactions on Knowledge and Data Engineering,* vol. 35, no. 3, pp. 2485–2496, 2023.

Սահմանափակ համաժամեցում բաշխված AdaBoost-ում

Արթուր Կ. Օղլուկյան[1] և Լուիս Ֆերնանդո դե Մինգո Լոպես[2]

[1]ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ, Երևան, Հայաստան
[2] Մադրիդի պոլիտեխնիկական համալսարան, Իսպանիա, Մադրիդ
e-mail: artur.oghlukyan@edu.isec.am, fernando.demingo@upm.es

Ամփոփում

Բաշխված մեքենայական ուսուցումը դարձել է ավելի ու ավելի կարևոր, քանի որ տվյալների աղբյուրները շարունակում են ընդլայնվել աշխարհագրորեն: Ավանդական անսամբլային մեթոդները, ինչպիսին է AdaBoost-ը, ցուցադրում են տպավորիչ կանխատեսողական հնարավորություններ, բայց հաճախ պահանջում են հաճախակի համաժամեցում հանգույցների միջև, ինչը հանգեցնում է հաղորդակցման զգալի ծանրաբեռնվածության: Այս առաջարկը ներկայացնում է հազվադեպ համաժամեցման նոր մոդել, որի դեպքում տեղական AdaBoost-ի բազմաթիվ փուլեր են իրականացվում մոդելի մասնակի կամ ամբողջական թարմացումների փոխանակումից առաջ: Հնարավոր առավելություններից են հաղորդակցման ծախսերի կրճատումը, ընդհատվող կապը կառավարելու հնարավորությունը և մրցակցային ճշգրտությունը՝ համեմատած լիովին համաժամեցված մոտեցումների հետ: Ներկայացվում է բեռնափոխադրումների ոլորտում իրական աշխարհի օգտագործման դեպք՝ այս նոր մոտեցման իրագործելիությունն ու արժեքը ցույց տալու համար: Հոդվածը եզրափակվում է ապագա ուղղությունների և հաղորդակցման արդյունավետ բաշխված ուսուցման վրա սպասվող ազդեցության ուրվագծմամբ:

Բանալի բառեր՝ բաշխված AdaBoost, հազվադեպ սինխրոնիզացիա, համալիր ուսուցում, հաղորդակցման արդյունավետ ուսուցում, ֆեդերատիվ ուժեղացում, թույլ սովորողներ, մասշտաբայնություն, սխալների հանդուրժողականություն, իրական աշխարհի տեղակայում

Нечастая синхронизация в распределенном AdaBoost

Артур К. Оглукян и Луис Фернандо де Минго Лопес

[1]Институт проблем информатики и автоматизации НАН РА, Ереван, Армения
[2]Политехнический университет Мадрида, Испания, Мадрид
e-mail: artur.oghlukyan@edu.isec.am, fernando.demingo@upm.es

Аннотация

Распределенное машинное обучение становится все более важным, поскольку источники данных продолжают расширяться географически. Традиционные ансамблевые методы распознавания, такие как AdaBoost, демонстрируют впечатляющие возможности прогнозирования, но часто требуют активной синхронизации между узлами, что приводит к значительным накладным расходам на связь. Это предложение вводит новую парадигму нечастой синхронизации, в которой несколько раундов локального AdaBoost выполняются до обмена частичными или полными обновлениями модели. Потенциальные преимущества включают снижение затрат на связь, способность обрабатывать прерывистое подключение и конкурентоспособную точность по сравнению с полностью синхронизированными подходами. Представлен реальный пример использования в отрасли грузоперевозок, чтобы продемонстрировать осуществимость и ценность этого нового подхода. Статья завершается описанием будущих направлений и ожидаемого влияния на эффективное с точки зрения связи распределенное обучение.

Ключевые слова: распределённый AdaBoost, нечастая синхронизация, ансамблевое обучение, коммуникационно-эффективное обучение, федеративное обучение, слабые обучающиеся, масштабируемость, отказоустойчивость, развертывание в реальных условиях.

Կանոններ հեղինակների համար

ՀՀ ԳԱԱ ԻԱՊԻ "Կոմպյուտերային գիտության մաթեմատիկական խնդիրներ" պարբերականը տպագրվում է 1963 թվականից: Պարբերականում հրատարակվում են նշված ոլորտին առնչվող գիտական հոդվածներ, որոնք պարունակում են նոր` չհրատարակված արդյունքներ:

Հոդվածները ներկայացվում են անգլերեն՝ ձևավորված համապատասխան "ոճով" (style): Հոդվածի ձևավորման պահանջներին ավելի մանրամասն կարելի է ծանոթանալ պարբերականի կայքէջում՝ http://mpcs.sci.am/:

Rules for authors

The periodical "Mathematical Problems of Computer Science" of IIAP NAS RA has been published since 1963. Scientific articles related to the noted fields with novel and previously unpublished results are published in the periodical.

Papers should be submitted in English and prepared in the appropriate style. For more information, please visit the periodical's website at http://mpcs.sci.am/.

Правила для авторов

Журнал «Математические проблемы компьютерных наук» ИПИА НАН РА издается с 1963 года. В журнале публикуются научные статьи в указанной области, содержащие новые и ранее не опубликованные результаты.

Статьи представляются на английском языке и оформляются в соответствующем стиле. Дополнительную информацию можно получить на веб-сайте журнала: http://mpcs.sci.am/.

The electronic version of the periodical “Mathematical Problems of Computer Science” and rules for authors are available at

http://mpcs.sci.am/

Phone: (+37460) 62-35-51
Fax: (+37410) 28-20-50
E-mail: mpcs@sci.am
Website: http://mpcs.sci.am/

Ստորագրված է տպագրության՝ 24.11.2025
Թուղթը՝ օֆսեթ:
Հրատարակված է ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտի կողմից
Ծավալը՝ 77 էջ: Տպաքանակը՝ 100
ՀՀ ԳԱԱ ԻԱՊԻ Համակարգչային պոլիգրաֆիայի լաբորատորիա
Երևան, Պ. Սևակի 1
Հեռ. +(374 60) 623553
Գինը՝ անվճար

Подписано в печать 24.11.2025
Офсетная бумага.
Опубликовано Институтом проблем информатики и автоматизации НАН РА
Объём: 77 страниц. Тираж: 100
Лаборатория компьютерной полиграфии ИПИА НАН РА.
Ереван, П. Севака 1
Тел.: +(374 60) 623553
Цена: бесплатно

Signed in print 24.11.2025
Offset paper
Published by the Institute for Informatics and Automation Problems of NAS RA
Volume: 77 pages
Circulation: 100
Computer Printing Lab of IIAP NAS RA
Yerevan, 1, P. Sevak str.
Phone: +(374 60) 623553
Free of charge