Computational Creativity by Heuristic Search and Machine Learning

Amit Konar, Sayantani Ghosh

Artificial Intelligence Laboratory, Electronics and Telecommunication Engineering Department, Jadavpur University, India

Abstract

Computational creativity refers to synthesis of human-like creativity on machines. This paper aims at synthesizing computational creativity by taking up an interesting problem to develop chapter-end problems of mathematics and physics using Artificial Intelligence techniques. The methodology employed to address the problem includes traditional random search and heuristic search, and modern techniques including Generative Adversarial Networks and Large Language Models. The true spirit of these models is to autonomously enhance diversity of problems by random exploration, structured search and learning. In addition, the paper demonstrates the scope of inductive learning to learn problem-solving from analogous problems, and to employ it to solve or generate similar problems. A set of metrics is proposed to compare the relative merits of the proposed and existing algorithms with a view to a have a uniform framework of comparison for the present and future research.

Keywords

Computational creativity, diversity, problem generation, scientific domain

1. Introduction

Computational creativity (CC) [1, 2] deals with artificial modality of synthesis of creativity by intelligent computational models. It has wide scope in diverse disciplines of knowledge, covering linguistics, music, poetry, and even hard science like physics and mathematics. With the advent of Large Language Models (LLMs) [3] the scope of CC has enhanced significantly as scientific and literary ideas now can be presented in a human-like fashion. For instance, the poems of Lord Byron, which convey strong emotion of love, have now been captured by modern LLM, such as ChatGPT-3.5 [4]. It is indeed important to note that the re-discovered narration of ChatGPT-3.5, which writes: *"She walks the earth with grace and pride, a beauty that cannot be denied..."*, conveys a more realistic thoughts of a lover that a machine could hardly generate in the history of science. The above example demonstrates the computational power of LLMs in perceiving real-world thoughts and its representation in a natural language. Besides the power of perception and linguistic skill, CC also has a great role to play in synthesis of scientific creativity, including development of new mathematical theory [5, 6], raising interesting problems in a selected scientific domain [7], and many others.

The 2024 Sixth Doctoral Symposium on Intelligence Enabled Research (DoSIER 2024), November 28–29, 2024, Jalpaiguri, India

konaramit@yahoo.co.in (A. Konar); sayantani.sonrisa25@gmail.com (S. Ghosh)
0000-0002-9474-5956 (A. Konar); 0000-0002-3156-9772 (S. Ghosh)

The present work focuses on designing innovative approaches to generate chapter-end problems in science, particularly in mathematics and physics, by leveraging both classical and modern AI techniques. Classical methods for fostering creativity include random search and heuristic search while modern techniques involve Generative Adversarial Networks (GANs) [8] LLMs. The study also explores the potential of inductive learning for developing computational creative systems by drawing insights from analogous problems and applying them to solve or generate similar ones. The methodologies for problem synthesis and problem-solving are demonstrated through illustrative examples. Furthermore, a set of metrics is proposed to evaluate and compare the effectiveness of the proposed generative algorithms against existing state-of-the-art (SOTA) techniques. While this work provides a foundation for achieving scientific creativity through machines, it acknowledges the limitations of the current models and emphasizes the need for future advancements in integrating contextual and environmental understanding to enhance the diversity and/or quality of generated problems, making them comparable to those created by humans.

The rest of the paper is organized as follows. Section 2 covers computational creativity using random search techniques, while Section 3 focuses on heuristic-guided algorithms for generating trigonometric identity problems. Section 4 explores inductive learning in mathematical creativity, and Sections 5 and 6 discuss scientific problem generation using GANs and LLMs. Section 7 presents the performance analysis of the proposed algorithms in comparison to SOTA methods. Section 8 addresses limitations and future directions, and Section 9 concludes the paper.

2. Creativity by Random Search and Experiments

Random experimentation refers to the process of conducting experiments in which certain variables or conditions are assigned randomly. This approach often leads to unexpected discoveries by enabling researchers to explore outcomes without preconceived notions or biases, fostering an environment of genuine exploration and creativity. History demonstrates that random experimentation can yield groundbreaking results, serving as a quintessential example of innovation in science.

Many revolutionary breakthroughs have stemmed from the process of random experimentation, where unexpected outcomes have paved the way for transformative insights [9]. For instance, Faraday's Law of Electromagnetic Induction resulted from simple experimental setups that, through systematic and open-ended exploration, fundamentally reshaped our understanding of electromagnetism. Similarly, the discovery of carbon's atomic structure and its various allotropes including graphite, diamond, and graphene, underscores how randomness and curiosity in experimental approaches can lead to groundbreaking findings. Another striking example is the discovery of DNA's double-helix structure by Watson and Crick. Their work was guided by experimental data and serendipitous results, where seemingly random but insightful observations played a crucial role in unraveling the mysteries of genetic inheritance, revolutionizing biology in the process.

To demonstrate the power of random search, let us instantiate the simple identity, often taught in middle schools:

$$(a - b)^2 = a^2 + b^2 - 2ab \tag{1}$$

where a and b are real numbers. Let us randomly select $a = \sqrt{x}$ and $b = \sqrt{y}$. The above substitution yields,

$(\sqrt{x} - \sqrt{y})^2 = x + y - 2\sqrt{xy} \geq 0$, as a whole-square of a real number is always positive or zero.

$$\Rightarrow \frac{x+y}{2} \geq \sqrt{xy} \tag{2}$$

$\Rightarrow$ arithmetic mean of 2 numbers: x and $y \geq$ geometric mean of the same numbers.

The above result indeed is an innovative outcome that unexpectedly follows as a result of random substitution for a and b. Now let us substitute $a = re^{jv}$, $b = re^{-jv}$ in (1) just as a random selection. The substitution yields:

$$(re^{j\theta} - re^{-j\theta})^2 = r^2e^{2j\theta} + r^2e^{-2j\theta} - 2r^2e^{j\theta}e^{-j\theta} \tag{3}$$

$$\Rightarrow (2jr\sin\theta)^2 = r^2(2\cos 2\theta - 2)$$

$$\Rightarrow -4r^2\sin^2\theta = 2r^2(\cos 2\theta - 1)$$

$$\Rightarrow \cos 2\theta = 1 - 2\sin^2\theta \tag{4}$$

It is noteworthy that the expression (4) is an important identity in trigonometry. However, its sudden appearance makes us rethink that random substitution may give rise to interesting and often unexpected results. Can we enhance the random search by adding structures to the search process? This will be undertaken in our next problem by exploring search and pruning unwanted search space by a heuristic algorithm. To guide the exploration process, we add a new term: *diversity cost* in the estimation of the diversity of a generated trial solution with respect to an initial trial solution, called root node in a search-tree.

3. Heuristic Search Approach to Computational Creativity

Consider the problem of generating trigonometric identities as the chapter-end problems of a first learner of trigonometry. Here, an approach similar to A* algorithm [10] is proposed to handle the afore-said identity generation problem. The rules utilized for generating identities of trigonometric first course are provided in Table 1. We here try to maximize $D(n) - g(n)$, where $D(n)$ is the diversity cost of node n with respect to the root node and $g(n)$ is the cost of generation with respect to the root node. Diversity cost between a parent and child node = no. of mismatched symbols in the left of the parent and child node + no. of mismatched symbols in the right of the parent and child node. Diversity cost of a node n = diversity cost of all the nodes lying between the root node and the node n.

3.1. Computation of Diversity Cost

The diversity cost of a node q with respect to its parent node p is evaluated by (5)

$$d(q) = |LT_p - LT_q| + |RT_p - RT_q| \tag{5}$$

Table 1
Some rules utilized for automatic trigonometric identity generation

No.	Rules
R.1	$sin^2x + cos^2x = 1$
R.2	$sec^2x - tan^2x = 1$
R.3	$cosec^2x - cot^2x = 1$
R.4	$sinx = 1/cosecx$
R.5	$cosx = 1/secx$

where,

LT_p= set of terms (operands) in the left side of node p.

RT_p= set of terms (operands) in the right side of node p. Similarly, LT_q and RT_q represent the left and right hand terms of the node q respectively. $|C|$ represents the number of elements in a given set C. The diversity cost of the root node $d(r)$ is always 0. Now if the path from the root node r to node q covers a sequence of nodes $n_1, n_2, ..., n_l$ then the diversity cost of node q with respect to root is evaluated by (6)

$$D(q) = d(q) + \sum_{i=1}^{l} d(n_i) \tag{6}$$

Let us now illustrate the computation of diversity cost of a node from the root node. Let the node q be the identity $1/cotx = sinx/cosx$, its parent node p be the identity $tanx = sinx/cosx$ and the root node r is 1=1. Here, there is no mismatch between the right sides of the parent and the current node i.e., RT_p={$sinx$, $cosx$}, RT_q={$sinx$, $cosx$} and thus, $|RT_p{}^{\smile}RT_q| = 0$. But there exists 2 mismatches in the left hand sides of the parent and child node, i.e., LT_p= {$tanx$}, LT_q={1,$cotx$} and so, $|LT_p{}^{\smile}LT_q| = 2$. Thus, $d(q) = 0 + 2 = 2$. Since, $d(r) = 0$ and so $D(q) = 0 + 2 = 2$.

3.2. Algorithm for Heuristic Guided Search

The algorithm for the automatic generation of trigonometric identifies utilizing the heuristic guided search is outlined in Algorithm 1. For the present context, the terminating condition is imposed on number of selection of nodes for expansion. An illustration of the heuristic guided search tree is provided in Figure 1 for a branching factor of 2 and pre-defined search depth of 3. The value of the cost function (i.e., $D(n) - g(n)$) is provided in $\{\cdot\}$ beside each generated node.

4. Inductive Learning Approach to Computational Creativity

An alternative modality of problem generation is through inductive learning [11, 12]. Here, the program learns a theme from examples and the learnt knowledge is utilized to develop a new problem. As a simple example, suppose a program learns the rule: *um* $\rightarrow$ *a* to transform the plural form: bacteria from the singular form: bacterium, and employs the derived knowledge to compute the plural form of penicillium. Similar examples hold for

Algorithm 1 : Heuristic guided search algorithm for automatic trigonometric identity generation

1. Initialize a list L with a start-up element e: 1=1.
2. **While** terminating condition is not reached **do**

   Begin

   a) Expand e by using the rules 1 to 5 k times randomly to generate k offspring of e, called n_i, i=1 to k.

   b) Add them in descending order of their $D(n_i) - g(n_i)$, i=1 to k. Delete e from list.

   c) Rename the first element of the list as e.

   End-While

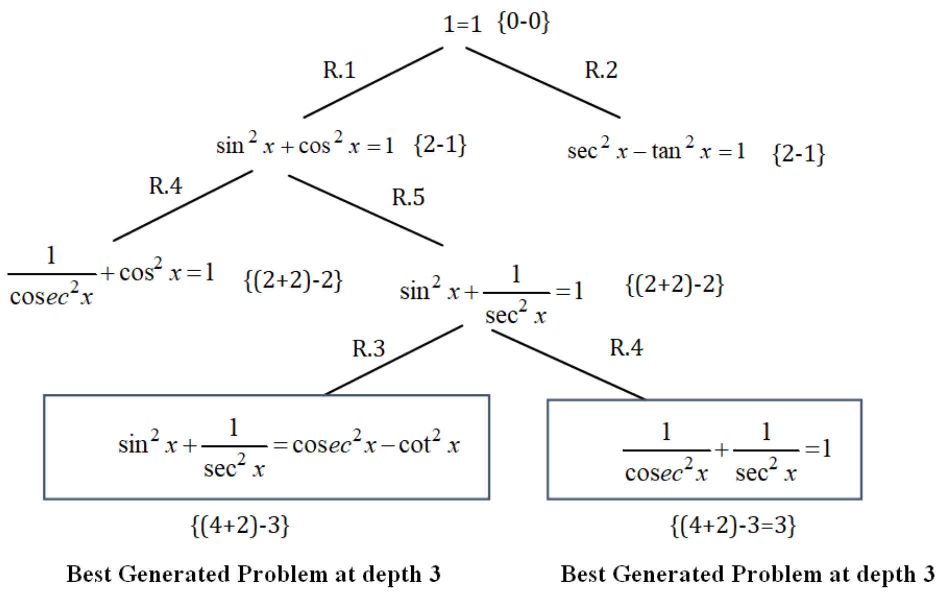


Figure 1: Illustration of the heuristic guided search tree for trigonometric identity generation for a branching factor of 2 and user-defined depth of 3.

learning the knowledge *us* $\rightarrow$ *i* for transformation of the singular form alumnus to alumni, and uses it to derive fungi from its singular form fungus. Naturally, the question arises: can we utilize similar inductive knowledge for the generation of new problems? The answer to this question is illustrated though an example from integration problems in mathematics.

Suppose a machine learns some examples of solving integrals which include:

Example 1: $\int \frac{1}{x}\, dx = \ln(x) + c$
Example 2: $\int \frac{2x}{x^2+1}\, dx = \ln(x^2+1) + c$

If the machine can accidently identify that the generalization of the above integrals is

Table 2
Table for $f(n)$ and $f'(n)$

$\mathbf{f(n)}$	$\mathbf{f'(n)}$
$f_1(n) + f_2(n)$	$f_1'(n) + f_2'(n)$
$x^n, n \geq 1$	nx^n

$$\int \frac{f'(x)}{f(x)}\, dx = \ln f(x) + c \tag{7}$$

which can be learnt by random experiments with known tables of $f(n)$ and $f'(n)$ (Table 2), then using the acquired knowledge (7), the machine can generate several problems on integrals using (7) such as

$$\int \frac{2x+5}{x^2+5x+2}\, dx = \ln(x^2+5x+2) + c \tag{8}$$

5. Computational Creativity by Generative Adversarial Network

A GAN [8] is a deep learning framework consisting of two primary components: a generator and a discriminator. These components work in a competitive setup to enable the creation of realistic outputs. The generator is tasked with producing synthetic data, while the discriminator acts as a classifier that evaluates the authenticity of the generated data. The competition arises as the generator simultaneously tries to fool the discriminator by improving the quality of its outputs. This adversarial process continues until the generator produces outputs so realistic that the discriminator fails to identify them as fake. The LeakGAN (Generative Adversarial Network with leaked Information) [2] is an advanced variant of GAN designed for text generation, where, in addition to the adversarial setup, the discriminator leaks high-level features of partially generated sequences to guide the generator, improving coherence and diversity in the generated text outputs. This feedback mechanism allows LeakGAN to produce more structured and meaningful sequences compared to standard GANs. The present work utilizes the LeakGAN framework for generating trigonometric identity problems.

In LeakGAN, the discriminator, implemented as a Convolutional Neural Network (CNN), and the generator, which employs Long Short-Term Memory (LSTM) units, are initially pre-trained on a dataset of mathematical identities sourced from textbooks. During the pre-training phase, all identity problems are tokenized and converted into embedding vectors, as detailed in Appendix [13].

The generator consists of two LSTM modules: the manager and the worker. The generation process begins with the selection of a random term, represented as an embedding vector, which serves as the starting point for generating a new identity. This vector is first passed to the discriminator, which produces a pooled vector. This pooled vector is then leaked to the manager module. The manager (LSTM) generates a goal vector that predicts the next term being an operand or an operator. This goal vector is concatenated with the initial vector and fed into the

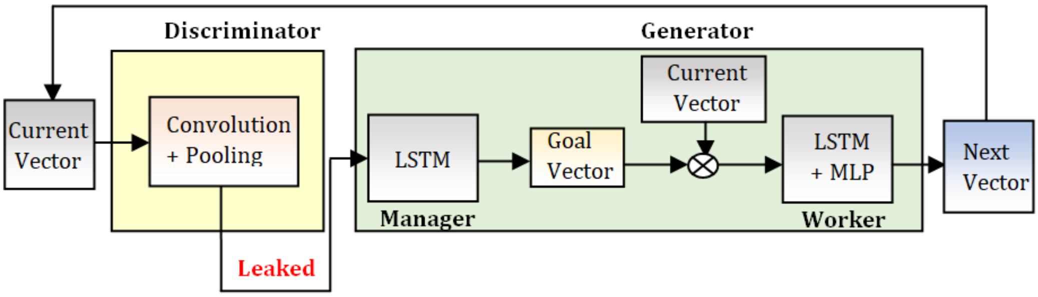


Figure 2: Block diagram of LeakGAN for generating trigonometric identities

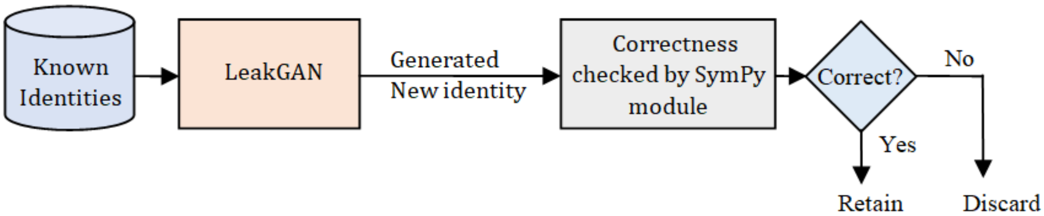


Figure 3: Correctness checking of newly generated identity by SymPy module

worker module, which comprises an LSTM and a Multi-Layer Perceptron (MLP). The worker predicts the next term based on the above concatenated input. This iterative process continues until the complete identity is generated. The detailed procedure for generating new identities is described in Algorithm 2 and its schematic view is shown in Figure 2. Additionally, an exemplar problem is included in Appendix [13] to demonstrate the generation process due to space limitations.

Once an identity is generated, it is passed to the discriminator to classify it as either real or fake. Identities classified as real are further evaluated for mathematical correctness using the SymPy module (a Python library for equation solving) as shown in Figure 3. If an identity is verified as correct, it is considered for inclusion as a chapter-end exercise; otherwise, it is discarded.

6. Large Language Models for Scientific Problem Generation

The Blackboard approach in AI [14] offers an effective framework for generating physics problems by integrating symbolic reasoning with the capabilities of large language models (LLMs). At the heart of this system is the blackboard manager. The blackboard serves as a shared workspace (see Figure 4) that displays the initial variables, such as final velocity v, time t, and initial velocity u, making this information accessible to various agents for computation and problem generation via LLMs. Suppose the blackboard manager displays the values of u, v,

Algorithm 2 : Generation of identity problems by LeakGAN

1. Initialize any random embedded vector as the start-up element.
2. **While** entire identity is not generated **do**

  Begin

    a) Transfer the current embedded vector to the discriminator that performs convolution and pooling upon this vector.

    b) Transfer the pooled vector to the manager module to generate a goal vector.

    c) Concatenate the goal vector and the current embedded vector and feed to the worker module to predict the next vector.

    d) Append the next vector to the current vector. Also, update the current vector with the next vector.

  End-While

3. Return the completely generated trigonometric identity problem by decoding its embedded vector.

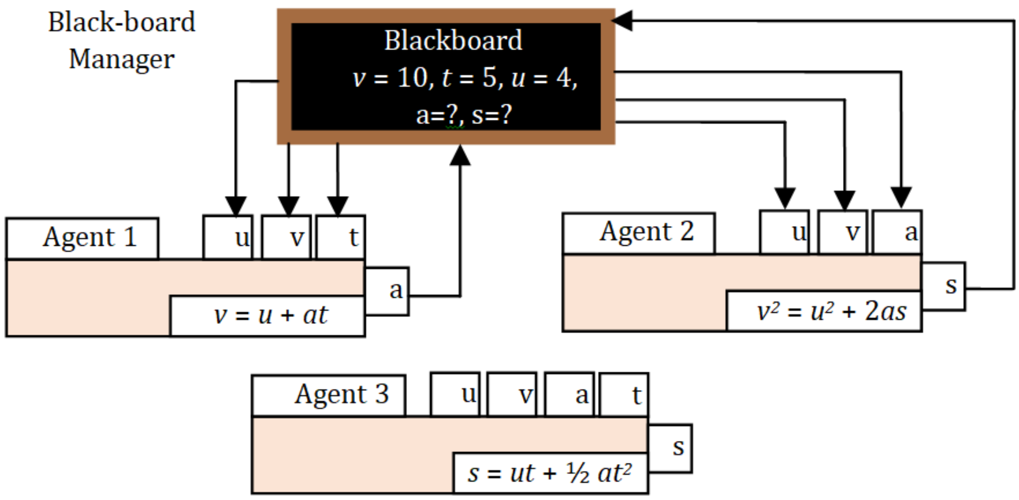


Figure 4: Illustration of Blackboard approach for problem generation

and t. These values are passed to Agent 1, which calculates the value of acceleration a. Next, this agent transfers the value of a back to the blackboard manager. The blackboard manager then displays the value of a along with u, v, and t. The values of u, v, and a are then taken up by Agent 2, which uses them to calculate the value of s.

The detailed procedure of this approach is provided in Algorithm 3. When $u = 5m/s$, $v = 10m/s$, $t = 5s$, find a and s is represented by LLM as: "A ball is thrown with an initial velocity $u = 5m/s$ downwards. When the ball reaches a distance s, the velocity becomes $v = 10m/s$. Find the time of traversal of the ball and the distance traversed."

Algorithm 3 : **Blackboard approach for problem generation in physics**

1. Initialize Agent 1 to compute $v = u + at$, Agent 2 to compute $v^2 = u^2 + 2as$, Agent 3 to compute $s = ut + 1/2at^2$, and iteration $i = 0$.
2. **For** each agent do in parallel
   a) If one but all parameter is unknown, evaluate the parameter using Agents.
   b) Submit it to the blackboard manager for updating.
   c) $i \leftarrow i$+1
   End-For
3. The blackboard manager updates the newly recorded parameters on the blackboard.
4. a) If $i \leq$ maximum number of parameters and $\tau \leq$ user defined runtime, loop through Step 2.
   b) If i > run-time, stop.
5. If condition 1 holds and i= maximum number of parameters, then the blackboard manager passes on the known parameters and unknown parameters to the LLM to prepare a question.

7. Experiments: Relative Performance Analysis

The performance of the proposed techniques in comparison to the existing ones is discussed below.

7.1. Comparative Study of Search Based Methods for Identity Generation

Table 3 presents a comparative analysis of identity problems generated by SOTA methods and the proposed approaches, evaluated using two key metrics: i) Mean Diversity per Unit Depth (MDPUD), defined as the mean diversity of the best node normalized by its depth across 150 problems, and ii) average run-time complexity over 150 instances. For this comparative study, the problems generated by SOTA methods were back-traced to the root, and the proposed algorithms generated identities at the same depth within the search tree to ensure consistency. The findings demonstrate that the proposed approaches outperform traditional methods by producing more diverse and less predictable problems while achieving greater efficiency in run-time performance.

Although the MDPUD for both the random and heuristic-based search methods are similar, the random search method requires more time to generate identities. This is because the random search explores a larger search space to find a solution (creative outcome). In contrast, the heuristic-guided approach limits the depth of the search tree, resulting in lower runtime complexity.

7.2. Comparative Study of Generative Neural Models for Identity Generation

Table 4 presents the performance comparison between the LeakGAN model and traditional generative neural networks in generating 150 trigonometric identities. The metrics used for comparison include the Bilingual Evaluation Understudy (BLEU) score (a commonly employed to assess the quality of both text and equation generation [2]) and runtime complexity. The BLEU score evaluates how closely a machine translation matches reference data by measuring n-gram overlap between the generated and reference sequences. Common n-grams include

Table 3
Relative performance analysis of problems generated by the proposed approaches and the traditional methods

Algorithm	MDPUD	Run-time (sec)
Polozov et al. [15]	27.33	90.23
Papasalouros [16]	25.87	82.14
Pearce et al. [17]	14.96	42.45
Liu et al. [18]	15.20	35.02
Tabuguia et al. [19]	10.88	43.15
Briggs et al. [20]	24.50	96.72
Proposed Random Search	**30.74**	**56.43**
Proposed Heuristic Guided Search	**31.18**	**32.07**

Table 4
Comparative study of the LeakGAN model with traditional algorithms

Algorithms	BLEU-3 (%)	BLEU-4 (%)	BLEU-5 (%)	Run-time Complexity (secs)
LSTM [21]	55.12	28.33	16.43	57.27
VAE [22]	65.04	30.10	19.78	104.20
SeqGAN [23]	70.56	39.74	31.22	247.06
RankGAN [24]	75.09	43.86	33.65	208.54
LeakGAN	**86.52**	**74.81**	**60.32**	**353.23**

tri-grams (3-grams), quadra-grams (4-grams), and penta-grams (5-grams). To avoid favoring shorter outputs, a brevity penalty is applied. The final score is the geometric mean of n-gram precisions, adjusted by this penalty.

The results in Table 4 show that the proposed approach significantly outperforms all traditional networks in terms of BLEU scores. However, the runtime complexity of the LeakGAN model is higher than that of the traditional networks. This trade-off between improved quality and increased complexity reflects a balance between performance and efficiency.

7.3. Comparative Study of Generative Neural Models for Identity Generation

Table 5 showcases the performance comparison between the Blackboard approach and traditional LLMs in generating 100 physics problems. The evaluation is based on the same metrics outlined in Section 7.2. The results clearly indicate that the proposed approach generates a more diverse set of physics problems compared to established LLMs, as evidenced by its superior BLEU scores. Additionally, the computational efficiency of the proposed method is noteworthy, making it a practical solution for creating chapter-end exercises in physics.

8. Further Scope of Computational Creative Models

Existing computationally creative models offer numerous advantages but also face significant limitations. One notable drawback is their inability to effectively contextualize problems within

Table 5
Comparative study of the Blackboard approach for physics problem generation

Algorithms	BLEU-3 (%)	BLEU-4 (%)	BLEU-5 (%)	Run-time Complexity (secs)
Google Gemini [25]	82.22	72.86	55.44	45.56
ChatGPT [3]	85.20	75.32	59.88	41.42
MetaLlama [26]	80.57	70.90	53.81	48.87
Black board Approch	**87.03**	**76.27**	**61.05**	**41.72**

environmental representations. For instance, in Figure 5 (a) and (b), there are 2 scenarios illustrated which are as follows.

Scenario 1: A lotus of height $H + h$ stands upright in a pond, where h is above water and H is submerged. A gust of wind bends the lotus, making its tip touch the water a units away from its original position. Given h and a, find H. The above scenario can be addressed using Pythagoras theorem by equating $(H + h)^2 = H^2 + a^2$.

Scenario 2: An electric pole of height H has an electric wire attached to it at the top, with the lower end initially free. Next, the wire is stretched to $H + 1$ inches and fixed 2 inches from the base. Find H. The above scenario can be addressed using Pythagoras theorem by equating $(H + 1)^2 = H^2 + 2^2$.

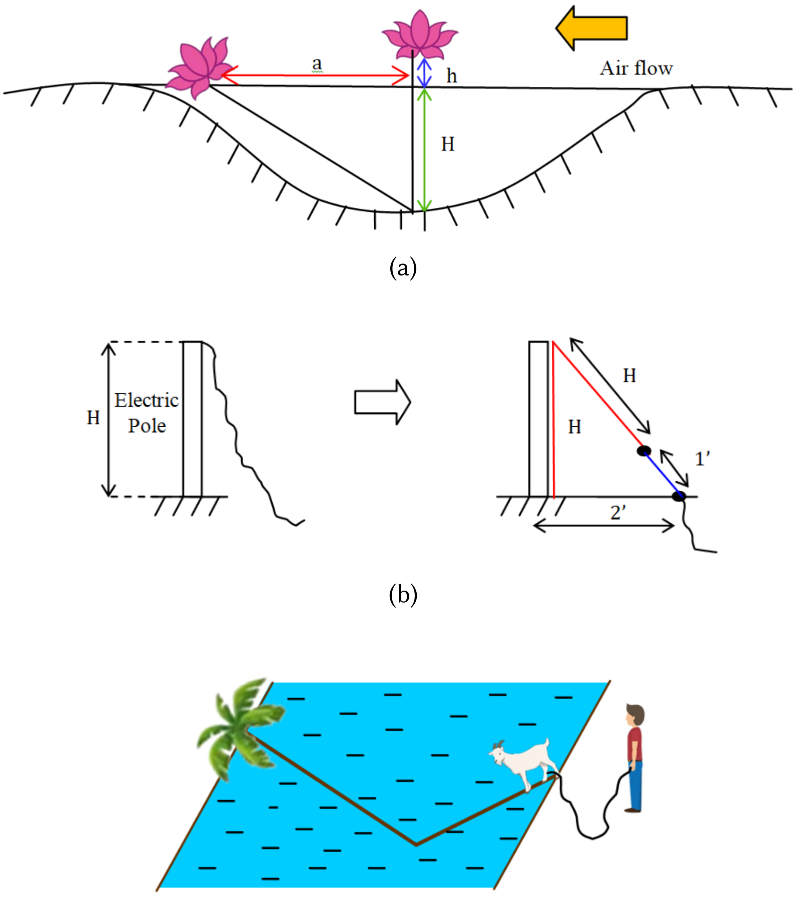
A machine capable of understanding the concepts depicted in scenarios 1 and 2, recognizing the Pythagorean theorem as the underlying principle across these problems, and tagging this rule to generate a novel problem in a different scenario would demonstrate remarkable creative reasoning. This type of learning mechanism is referred to as cross-domain learning [27]. The afore-said novel problem generation in a different scenario is illustrated in Figure 5 (c) and is described as follows.

Developed New Scenario: A coconut tree breaks and falls across a canal after a storm. A goat, tied to a string held by a man, walks along the trunk to the other side. Determine the canal's width.

However, current CC models fall short of achieving this. This limitation stems from the lack of perceptual knowledge about real-world entities, such as ponds or electric poles, and the inability to grasp the physical and spatial relationships [28, 29] inherent in these objects and their environments. In other words, these models are constrained by their limited capacity for environmental context, perceptual reasoning [30], and spatial understanding. Addressing this gap requires the development of more advanced computational frameworks that integrate environmental context and reasoning capabilities. By incorporating a richer understanding of the physical world and its representations, future models could enable spatial reasoning and generate innovative solutions or problems across diverse domains.

9. Conclusions

The present study highlights the potential of CC to replicate human-like creative processes within the scientific domain. By combining classical AI techniques, such as random experimentation, heuristic search, and inductive learning, with modern approaches like GANs and



(c)

Figure 5: Illustration of application of cross-domain knowledge to generate new problems in another domain (a) Scenario 1 depicting the lotus problem, (b) Scenario 2 depicting the electric pole problem (c) Scenario 3 denotes the problem generated from Scenarios 1 and 2

LLMs, this work demonstrates innovative methodologies for generating complex chapter-end problems in mathematics and physics. A comparative performance analysis confirms the efficacy of the proposed models in synthesizing creativity compared SOTA methods. While this research lays a strong foundation for computational creativity in scientific problem generation, it also emphasizes the need for future advancements that integrate contextual reasoning and perceptual knowledge to unlock AI's full potential for cross-domain learning, enabling the application of acquired knowledge across diverse domains.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] D. Mateja, A. Heinzl, Towards machine learning as an enabler of computational creativity, IEEE Transactions on Artificial Intelligence 2 (2021) 460–475.

[2] S. Ghosh, A. Konar, A. K. Nagar, Computational creativity by generative adversial network with leaked information, in: 2024 International Joint Conference on Neural Networks (IJCNN), IEEE, 2024, pp. 1–8.

[3] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han, Y. Tang, A brief overview of chatgpt: The history, status quo and potential future development, IEEE/CAA Journal of Automatica Sinica 10 (2023) 1122–1136.

[4] B. Porter, E. Machery, Ai-generated poetry is indistinguishable from human-written poetry and is rated more favorably, Scientific Reports 14 (2024) 26133.

[5] L. R. Varshney, Mathematical limit theorems for computational creativity, IBM Journal of Research and Development 63 (2019) 2–1.

[6] T. Veale, R. Pérez y Pérez, Leaps and bounds: An introduction to the field of computational creativity, New Generation Computing 38 (2020) 551–563.

[7] S. Colton, G. A. Wiggins, Computational creativity: The final frontier?, in: ECAI 2012, IOS Press, 2012, pp. 21–26.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Communications of the ACM 63 (2020) 139–144.

[9] J. C. Kaufman, R. J. Sternberg, The Cambridge Handbook of Creativity, Cambridge University Press, 2019.

[10] A. Konar, Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain, CRC press, 2018.

[11] L. Ghosh, R. Kar, A. Konar, A. Chakraborty, A. K. Nagar, Identification of brain activation regions in inductive learning based scientific creativity test, in: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2018, pp. 950–957.

[12] A. M. AlMana, M. Aksoy, An overview of inductive learning algorithms, International Journal of Computer Applications 88 (2014) 20–28.

[13] The appendix is uploaded in the google drive, 2024. URL: https://drive.google.com/file/d/1OsEUWFsEJRj87k0SQZNTWa7Okz5kpIp7/view?usp=drive_link.

[14] H. P. Nii, The blackboard model of problem solving and the evolution of blackboard architectures, AI magazine 7 (1986) 38–38.

[15] O. Polozov, E. O'Rourke, A. M. Smith, L. Zettlemoyer, S. Gulwani, Z. Popović, Personalized mathematical word problem generation, in: Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.

[16] A. Papasalouros, Automatic exercise generation in euclidean geometry, in: Artificial Intelligence Applications and Innovations: 9th IFIP WG 12.5 International Conference, AIAI 2013, Paphos, Cyprus, September 30–October 2, 2013, Proceedings 9, Springer, 2013, pp. 141–150.

[17] M. Pearce, J. McKinney, C. Alvin, Query-based generation of trigonometric identity problems and solutions, in: The Thirty-Third International Flairs Conference, 2020.

[18] Z. Liu, Y. Li, Z. Liu, L. Li, Z. Li, Learning to prove trigonometric identities, arXiv preprint

arXiv:2207.06679 (2022).
[19] B. T. Tabuguia, Hypergeometric-type sequences, Journal of Symbolic Computation 125 (2024) 102328.
[20] I. Briggs, P. Panchekha, Synthesizing mathematical identities with e-graphs, in: Proceedings of the 1st ACM SIGPLAN International Symposium on E-Graph Research, Applications, Practices, and Human-factors, 2022, pp. 1–6.
[21] A. Graves, A. Graves, Long short-term memory, Supervised sequence labelling with recurrent neural networks (2012) 37–45.
[22] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, S. Bengio, Generating sentences from a continuous space, arXiv preprint arXiv:1511.06349 (2015).
[23] L. Yu, W. Zhang, J. Wang, Y. Yu, Seqgan: Sequence generative adversarial nets with policy gradient, in: Proceedings of the AAAI conference on artificial intelligence, volume 31, 2017.
[24] K. Lin, D. Li, X. He, Z. Zhang, M.-T. Sun, Adversarial ranking for language generation, Advances in neural information processing systems 30 (2017).
[25] M. Imran, N. Almusharraf, Google gemini as a next generation ai educational tool: a review of emerging educational technology, Smart Learning Environments 11 (2024) 22.
[26] M. Masalkhi, J. Ong, E. Waisberg, N. Zaman, P. Sarker, A. G. Lee, A. Tavakkoli, A side-by-side evaluation of llama 2 by meta with chatgpt and its application in ophthalmology, Eye (2024) 1–4.
[27] H. Xu, Y. Liu, L. Liu, S. Zhi, S. Sun, T. Liu, M. Cheng, Step-wise distribution alignment guided style prompt tuning for source-free cross-domain few-shot learning, arXiv preprint arXiv:2411.10070 (2024).
[28] S. Ghosh, A. Konar, A. K. Nagar, Decoding subjective creativity skill from visuo-spatial reasoning ability using capsule graph neural network, in: 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–8.
[29] S. Ghosh, A. Konar, A. K. Nagar, Cognitive assessment of scientific creative-skill by brain-connectivity analysis using graph convolutional-interval type-2 fuzzy network, IEEE Transactions on Cognitive and Developmental Systems (2024).
[30] M. Laha, S. Ghosh, A. Konar, Exploration of depth perception in human binocular vision using eeg-based neuro-fuzzy classifier, in: 2023 8th International Conference on Computers and Devices for Communication (CODEC), IEEE, 2023, pp. 1–2.