Evolutionary Program Synthesis for Educational Puzzles: Towards Rigorous Reasoning and Educational Impact

Proposal for India AI Fellowship Grant

Aryan Arora

under the guidance of Prof. Somak Aditya
(Department of CSE, IIT Kharagpur)

October 2025

Online Resources: A public project page with updates and datasets is available at:
aryan1612.github.io/puzzle-evo-ai
The open-source code repository is hosted on GitHub:
github.com/aryan1612/puzzle-evo-ai

1 Motivation

Puzzle-based learning has long been associated with improved reasoning, spatial ability, and problem-solving skills. UNICEF and related educational studies consistently highlight that interactive, game-like activities foster engagement and measurable improvements in cognitive development and aptitude tests [1, 2, 3].

Why puzzles matter. Spatial and logical puzzles (e.g., jigsaws, Sudoku, visual reasoning tasks) correlate with better performance in STEM-related skills and abstract reasoning. For example, longitudinal studies show early exposure to puzzles leads to stronger spatial transformation skills later in life [4, 5]. UNICEF strongly advocates for 'Learning through Play', identifying activities like puzzle-solving as critical for developing foundational cognitive skills such as problem-solving, critical thinking, and spatial reasoning, which are direct precursors to academic achievement [6].

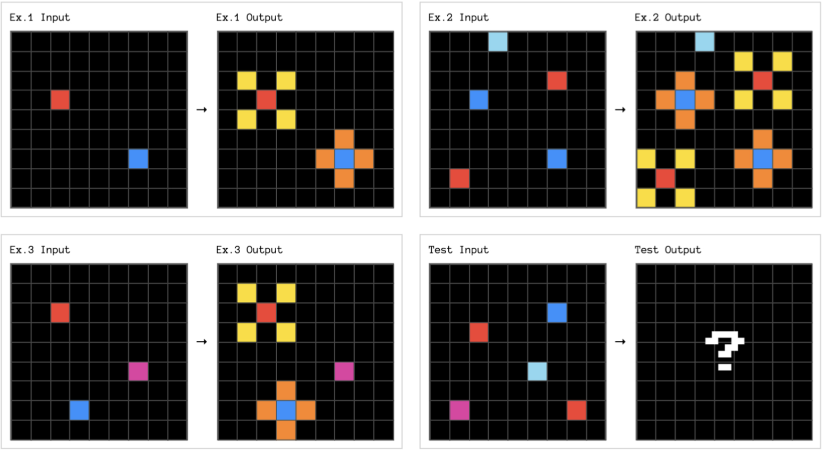


Figure 1: Example puzzle: visual/spatial puzzle task as used in ARCAGI [7, 8].

Why automated puzzle assistants. Despite their value, puzzles are underused in schools due to lack of localized material, teacher training, and adaptive feedback. An *Automated Puzzle Assistant* can fill this gap by:

- Generating and adapting puzzles dynamically (multiple difficulty levels, local languages).
- Providing structured hints and alternate solution strategies.
- Logging student attempts for adaptive learning and research datasets.

Why LLMs, and their weaknesses. Large Language Models (LLMs) are attractive because they can generate puzzles and explanations in natural language, adapt to local contexts, and support multilingual deployment. However:

- LLM outputs lack *correctness guarantees*: generated solutions may be plausible but logically invalid [9].
- Studies on reasoning benchmarks (e.g., PuzzleVQA, PlanBench) reveal systematic failures in multi-step planning and algorithmic reasoning [10, 11].
- Execution, semantics, and compilation errors arise when LLMs attempt to generate executable programs or structured solutions [12].

Why evolutionary program synthesis. Program synthesis offers a pathway to correctness by grounding solutions in formal representations (PDDL/DSL). Execution semantics ensure generated strategies can be verified. Evolutionary synthesis further enables iterative refinement: candidate programs evolve via mutation/crossover guided by correctness and feedback [13, 14, 15].

Local relevance and pilot studies. With Prof. Somak Aditya (Department of Computer Science and Engineering, IIT Kharagpur; affiliated with the Centre for Excellence in Artificial Intelligence), we are well-positioned to deploy and evaluate prototypes in real educational settings. In collaboration with the Centre, we plan to conduct small ablation-style pilot studies in local languages (e.g., Hindi, Bengali), enabling us to assess effectiveness and gather detailed student interaction traces. These studies will not only validate the assistant but also generate valuable data for iterative refinement.

2 Past Work (Foundation)

Over the last year, under the guidance of Prof. Somak Aditya, we developed a line of work that frames visual puzzles as planning problems:

- Constructed a benchmark of $\sim$9.5K PDDL problems across six puzzle types.
- Conducted systematic ablations using natural language, symbolic, and hybrid planning approaches.
- Analyzed syntactic and semantic error modes in generated planning programs.

Our experiments revealed that state-of-the-art LLMs and VLMs exhibit low performance on planning-based puzzles, especially due to compilation errors, incorrect execution semantics, and brittle representations. This directly motivates our proposed evolutionary program synthesis approach, which seeks to reduce such errors while enabling correctness guarantees. This prior work (currently under review) provides both datasets and insights, giving us a strong foundation to scale into educational applications.

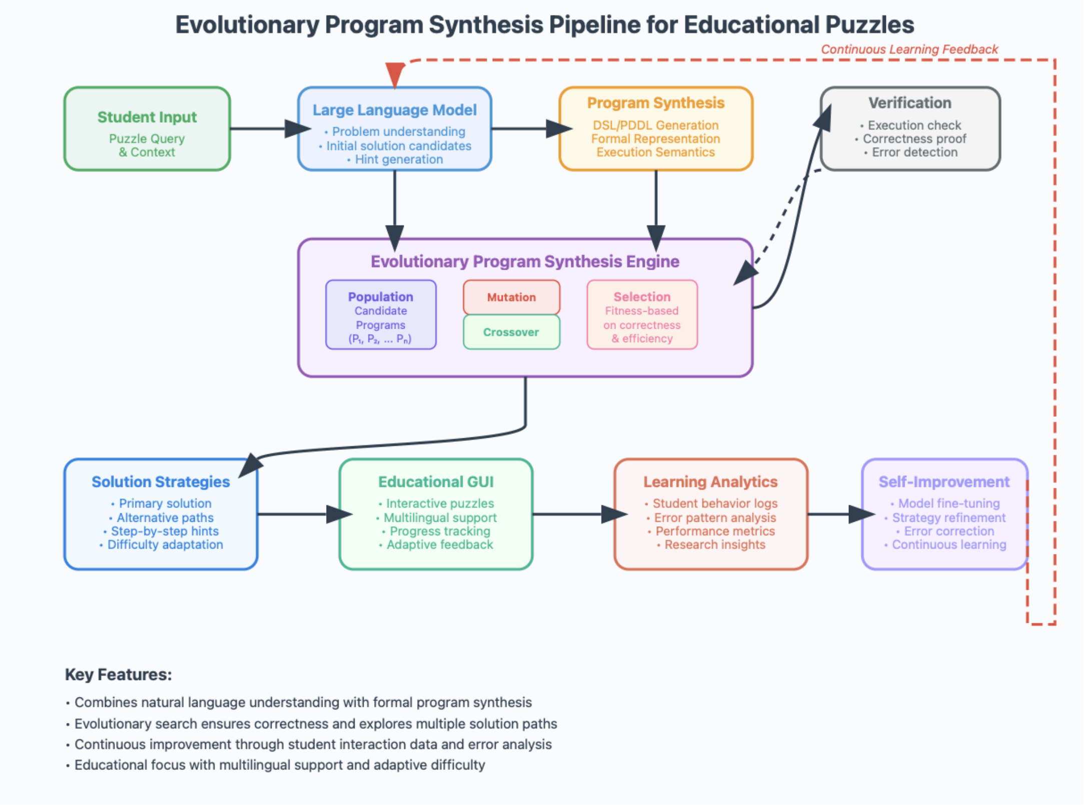
3 Goals

Primary Objective. A rigorous, publishable research paper that:

- Proposes a method combining evolutionary program synthesis with LLM reasoning for solving multi-step educational puzzles.
- Introduces a benchmark of educational puzzles annotated with multiple valid solution paths and stepwise hints.
- Demonstrates gains over baselines (LLM-only, symbolic planners) with systematic error analysis.
- Releases dataset and code for reproducibility.

Secondary Objective. Build a lightweight prototype platform:

- Student-facing GUI for Grades 1–8: puzzle solving, hint requests, multiple strategies.
- Logging of attempts and common error traces for iterative improvement.
- Initial ablation-style pilot studies with local-language deployment in classrooms.



Key Features:

- Combines natural language understanding with formal program synthesis
- Evolutionary search ensures correctness and explores multiple solution paths
- Continuous improvement through student interaction data and error analysis
- Educational focus with multilingual support and adaptive difficulty

Figure 2: Illustrative pipeline: LLM + evolutionary program synthesis + student interaction loop.

4 Proposed Technical Approach

1. **Dataset / benchmark:** Aggregate existing puzzle datasets (PuzzleVQA, AlgoPuzzleVQA, our PDDL puzzle benchmark) [10, 16] and synthesize new instances (including stochastic variants). Annotate with multiple solution paths and stepwise hints.

2. **Representation:** Use compact DSL/PDDL representations for puzzles that map naturally to planning; represent stochastic puzzles via multiple acceptable outcomes or probabilistic descriptions.

3. **Hybrid solver pipeline:** LLMs propose high-level candidate solutions and hint sequences; evolutionary program synthesis (mutation/crossover/selection) refines program candidates; symbolic planners or interpreters verify/execute candidates when applicable [14, 15, 17].

4. **Self-improvement loop:** Use failed attempts and student-history traces to refine the generator/search policy (fine-tune LLMs or update evolutionary operators) following recent self-improving synthesis paradigms [14, 15].

5. **Hint/explanation generation and assessment:** Generate pedagogically-structured hints and multiple solution strategies; assess hint quality via human judgments and proxy metrics (step coverage, alignment to ground-truth plan).

6. **Evaluation:** Measure solution correctness, multiplicity of solutions found, human-rated hint usefulness, generalization to unseen puzzle types, and, if feasible, small pilot studies measuring learning impact.

5 Expected Outcomes

- A peer-reviewed research paper presenting dataset(s), methods (evolutionary program synthesis + LLM reasoning), experimental results and error analyses (primary outcome).

- A working web prototype for demonstration, data collection, and small pilots (secondary outcome).

- Public release of dataset and code for reproducibility.

6 Timeline (12 months, indicative)

Months 1–3	Dataset curation (PuzzleVQA, AlgoPuzzleVQA, PDDL benchmark) and baseline reproduction.
Months 4–6	Implement LLM + symbolic + evolutionary search pipeline; initial experiments on simple puzzles.
Months 7–9	Extend to stochastic and image-based puzzles; develop hint generation module; human evaluation setup.
Months 10–12	Pilot GUI, finalize experiments, write and submit paper; public code/data release.

7 Conclusion

This project combines educational motivation, practical impact, and publishable AI research. By bridging LLM reasoning with evolutionary program synthesis, we address both correctness and explanation gaps. Our prior work on PDDL puzzles, coupled with planned pilots through the Center for AI, provides a unique opportunity for research and social benefit.

References

[1] UNICEF. *Playful Learning at Home: Simple, fun, and inclusive activities to keep children with disabilities engaged and growing*. https://www.unicef.org/india/stories/playful-learning-home. Accessed: September 2025. 2025.

[2] Susan C. Levine et al. “Early Puzzle Play Predicts Preschoolers’ Spatial Transformation Skill”. In: *Developmental Science* 15.5 (2012), pp. 1–12. DOI: 10.1111/j.1467-7687.2012.01147.x. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3289766/.

[3] Joseph Reynolds, Andrea Rozario, et al. “Jigsaw Puzzling Taps Multiple Cognitive Abilities and Is a Potential Protective Factor for Cognitive Aging”. In: *Frontiers in Psychology* 9 (2018), p. 299. DOI: 10.3389/fpsyg.2018.00299. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2018.00299/full.

[4] Nora S. Newcombe and Thomas F. Shipley. “Is Early Spatial Skills Training Effective? A Meta-Analysis”. In: *Frontiers in Psychology* 11 (2020), p. 1938. DOI: 10.3389/fpsyg.2020.01938. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7485443/.

[5] David H. Uttal et al. “The malleability of spatial skills: A meta-analysis of training studies”. In: *Psychological Bulletin* 139.2 (2013), pp. 352–402. DOI: 10.1037/a0028446.

[6] UNICEF and The LEGO Foundation. *Learning through play: Strengthening learning through play in early childhood education programmes*. UNICEF. 2018. URL: https://www.unicef.org/sites/default/files/2018-12/UNICEF-Lego-Foundation-Learning-through-Play.pdf.

[7] Francois Chollet et al. “Arc-agi-2: A new challenge for frontier ai reasoning systems”. In: *arXiv preprint arXiv:2505.11831* (2025).

[8] Julien Pourcel, Cédric Colas, and Pierre-Yves Oudeyer. “Self-Improving Language Models for Evolutionary Program Synthesis: A Case Study on ARC-AGI”. In: *arXiv preprint arXiv:2507.14172* (2025).

[9] Jason Wei et al. “Chain of Thought Prompting Elicits Reasoning in Large Language Models”. In: *NeurIPS*. 2022. URL: https://arxiv.org/abs/2201.11903.

[10] X. Chia et al. “PuzzleVQA: Diagnosing Multimodal Reasoning Challenges of Language Models with Abstract Visual Patterns”. In: *Findings of ACL 2024*. 2024. URL: https://aclanthology.org/2024.findings-acl.962.pdf.

[11] Karthik Valmeekam et al. “PlanBench: An Extensible Benchmark for Evaluating Large Language Models on Planning and Reasoning about Change”. In: *arXiv preprint* (2022). eprint: 2206.10498. URL: https://arxiv.org/abs/2206.10498.

[12] Deepanway Ghosal et al. “Are Language Models Puzzle Prodigies? Algorithmic Puzzles Unveil Serious Challenges in Multimodal Reasoning”. In: *arXiv preprint* (2024). eprint: 2403.03864. URL: https://arxiv.org/abs/2403.03864.

[13] Sumit Gulwani. “Automating String Processing in Spreadsheets Using Input-output Examples”. In: *POPL*. 2011. DOI: 10.1145/1926385.1926423.

[14] Kevin Ellis, Christopher Rabe, et al. “DreamCoder: Bootstrapping Inductive Program Synthesis with Wake-Sleep Library Learning”. In: *Proceedings of PLDI*. 2021. DOI: 10.1145/3453483.3454080. URL: https://dl.acm.org/doi/10.1145/3453483.3454080.

[15] Julien Pourcel, Cédric Colas, and Pierre-Yves Oudeyer. “Self-Improving Language Models for Evolutionary Program Synthesis: A Case Study”. In: *arXiv preprint* (2025). eprint: 2507.14172. URL: https://arxiv.org/abs/2507.14172.

[16] Declare Lab / AlgoPuzzleVQA authors. *AlgoPuzzleVQA: Multimodal Algorithmic Puzzle VQA Benchmark*. https://algopuzzlevqa.github.io/. Dataset and website. 2024.

[17] Y. Zhang et al. “Fast and Accurate Task Planning using Neuro-Symbolic Language Models”. In: *arXiv preprint* (2024). eprint: 2409.19250. URL: https://arxiv.org/abs/2409.19250.