

Development and Artefact-Fidelity Evaluation of STEAM-Oriented Geometry Resources Using a Python-Supported GeoGebra Workflow

Yuchen Xie*

Department of Mathematics and Applied Mathematics, School of Mathematics and Statistics, Shaoguan University, China

**Corresponding author: Yuchen Xie.*

Abstract

Dynamic-geometry resources need tasks that link manipulation to an explainable geometric relation and remain technically inspectable when teachers adapt them. This study develops two STEAM-oriented GeoGebra activities and examines their artefact fidelity through a Python-supported, GeoGebra-based workflow. In Triangular Bridge-Frame Exploration, learners vary sliders, measure a triangle, and justify a bridge-frame design decision. In Transformation-Pattern Design, they construct a motif and justify a visual pattern after rotation, reflection, and translation. Python remains teacher-facing and checks the shared specification. A remote AutoDL SSH audit covered five display cases, a 165-case triangle grid, six out-of-range triangle inputs, 25 audited angles, and two out-of-range angles. All in-range cases met the declared conditions, whereas every out-of-range input was rejected. The transformation check covers rotation, reflection, and translation as one sequence. The results verify artefact fidelity across display, grid, and boundary cases; they do not provide classroom-outcome evidence. A classroom-evaluation protocol is therefore specified as the next step.

Keywords

STEAM education, dynamic geometry, GeoGebra, Python-supported validation, teaching resource development

1. Introduction

In a dynamic-geometry task, learners can alter a parameter, inspect the resulting configuration, and decide which relation remains unchanged. A static diagram can label a triangle, rotation, or pattern, but it cannot support that variable-and-invariant comparison. GeoGebra makes this comparison possible in routine geometry and algebra activities. Its educational value still depends on the task, teacher support, and local conditions, as recent reviews note [1–3].

The practical challenge is therefore not access to software alone. Teachers need resources that connect dragging to an interpretable question and that can be checked before a lesson. GeoGebra activities have been designed around modelling, transformations, three-dimensional geometry, and computational thinking [4–7].

Intervention studies motivate this work, but their outcomes remain conditional on their participants, implementation, and measures [8–11].

This study develops two resources with GeoGebra for learners and a separate Python routine for technical checking. Within GeoGebra, learners construct, drag, measure, and explain. The teacher-facing Python routine uses the shared specification to generate audit cases and verify the listed invariants. Separating these functions specifies validation conditions without calling GeoGebra a Python tool. It also leaves learner activity in GeoGebra and reserves classroom-effect questions for a later study.

Triangular Bridge-Frame Exploration asks learners to vary span, height, and joint position, then record what changes in the triangle. Prediction, controlled variation, and recording provide an inquiry routine; the bridge frame gives the task an engineering-design setting, not a structural simulation. In Transformation-Pattern Design, learners combine rotation, reflection, and translation to make and justify a repeated motif. Visual composition and digital production frame the transformation work. Within this paper, STEAM-oriented identifies the selected practices. Broader expertise and assessment would require a different construct and evidence.

The study asks whether the specified GeoGebra activities and the parallel Python audit agree with the stated triangle and transformation conditions across representative, boundary, and out-of-range inputs. The present evidence answers these artefact questions. Learner achievement, engagement, affect, and comparative learning benefit would require a separately implemented and approved classroom study.

2. Related Work and Design Rationale

Dynamic geometry supports mathematical work in which an object can be constructed, moved, and discussed. A GeoGebra task on matrix products, for example, combines manual and digital work to connect symbolic operations with transformations [7]. Visual-geometry work similarly treats dynamic models as prompts for conjecture, testing, and explanation rather than as decorations around a fixed answer [12]. These studies support an activity design that gives learners an object to manipulate and a reason to account for what they observe.

The present contribution belongs to a design-oriented resource-development strand. Its unit of analysis is a teaching-resource specification and its technical audit, rather than a learner-effect estimate. The paper reports the activity conditions, input bounds, mathematical tolerances, and audit outcomes needed to interpret the evidence. Intervention research needs outcome measures beyond this technical standard.

The closest studies have distinct purposes. Dos Santos et al. link GeoGebra activity design to STEAM modelling, Ziatdinov and Valles connect modelling, visualization, and programming in STEM topics, and Chytas et al. examine computational-thinking work in a classroom setting [4–6]. The present paper adds an auditable pair of GeoGebra resources with a separate Python support layer. The layer makes the resource specification reviewable without displacing learner construction or explanation from GeoGebra.

STEAM-oriented refers here to particular elements of the two activities, not to a claim that learners achieve comprehensive STEAM mastery. In the bridge-frame task, learners measure a triangle, make a prediction, record observations, and discuss a design choice. In the pattern task, they use transformations to compose a visual form for digital production. The bridge frame is a setting for geometry-based design reasoning, not a structural-safety model. This narrow design follows work linking digital geometry with modelling, computational thinking, representation, and task design [4–6].

Published GeoGebra interventions vary in participants, duration, comparison conditions, and outcome measures. Examples include teachers' work with three-dimensional geometry and students' conceptual understanding and critical thinking [8–10]. Reports on device access, teacher preparation, and initial difficulty with the software show that implementation conditions also need to be specified [13]. A mathematics-anxiety study similarly depends on a defined instrument and analysis plan [14]. For the present resources, their direct use is to specify the future protocol rather than estimate an effect size.

The same parameter specification supports both learner work and technical checking. Each task begins with a GeoGebra object that can be manipulated and an explanation prompt; dragging is used to generate conjectures rather than complete the task. The bridge-frame prompt requires a geometric design rationale, whereas the

pattern prompt requires a relation that supports a visual choice. When a teacher adapts either resource, the stated conditions and audit criteria indicate what should be reviewed.

3. Resource Development Method

3.1 Hybrid workflow

Figure 1 presents the workflow. A teacher starts from a learning objective and a design context, then builds the learner-facing construction in GeoGebra. Each activity combines sliders or movable points with a prediction, observation, and explanation task. GeoGebra's construction tools and protocol features support activity preparation and review [15]. Python does not operate GeoGebra. It independently reads the shared parameter specification, generates auditable cases, and checks the geometric relations that learners are invited to notice. The reported audit outcomes provide teacher-facing resource-quality evidence.

Figure 1: Hybrid resource-development and fidelity-check workflow. GeoGebra supports learner-facing construction and exploration; Python checks the shared resource specification independently

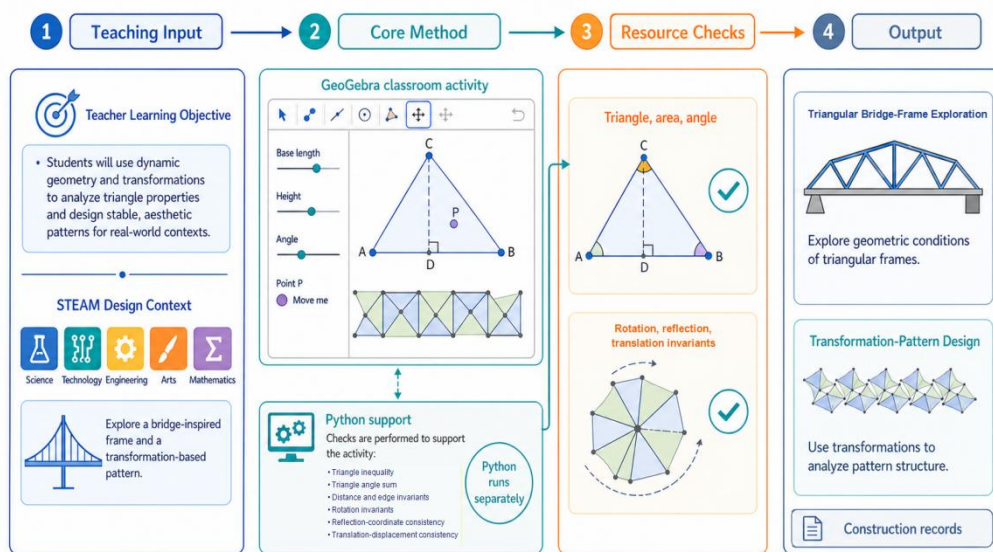


Table 1: Two GeoGebra-centred STEAM-oriented resources and their Python support

Resource	GeoGebra learner activity and STEAM-oriented purpose	Python-supported fidelity check
Triangular Bridge-Frame Exploration	Adjust span, height, and joint-position sliders; drag a vertex; predict one parameter change; record measurements; and defend a bridge-frame geometry decision.	Five display cases, a 165-case grid, and six out-of-range inputs check triangle validity, the base–height area relation, and angle sum.
Transformation-Pattern Design	Create a triangular motif; rotate, reflect, translate, and vary the angle slider; then produce and justify a labelled visual pattern.	Five display cases, a 25-angle sweep, and two out-of-range angles check the complete transformation sequence.

Table 2: Technical audit results and permissible evidence claims

Resource audit	Technical result	Permissible inference
Triangular Bridge-Frame Exploration	5/5 display cases and 165/165 grid cases passed triangle, area, and angle conditions; 6/6 out-of-range inputs were rejected.	The stated triangle resource specification is internally consistent for the audited inputs.
Transformation-Pattern Design	5/5 display cases and 25/25 angle-sweep cases passed the full transformation sequence; 2/2 out-of-range angles were rejected.	The stated transformation specification is internally consistent for the audited inputs.
Classroom outcomes	No participant data, comparison condition, or classroom records were collected.	No learner-outcome or comparative-benefit claim is supported.

Table 1 separates the learner action from the support computation. It also makes the interdisciplinary role of each task explicit. The first resource uses a science-oriented practice of prediction, controlled variation, and recording. Its engineering connection is a bridge-frame design discussion, not a structural simulation. The second resource connects mathematics with visual design and digital production. Table 2 summarises the corresponding technical audit and the permitted scope of its claims.

Teachers introduce the design setting and relevant mathematical relation, but do not supply an explanation. Learners then use GeoGebra controls to produce contrasting configurations and annotate what they observe. They subsequently explain the relation with words, diagrams, or measurements and apply it to a bridge-frame or pattern decision. Python results remain with the teacher, leaving learners responsible for the explanation.

3.2 Audit Trace and Adaptation Record

Traceable design choices organise the workflow. The GeoGebra activity defines the objects, controls, transformations, and prompt for the learner-facing task. Learners predict, observe, record, and explain through the activity. The Python audit uses the same initial coordinates, slider bounds, and transformation order to create a separate technical check. Its structured results contain the input values, calculated conditions, pass or rejection status, and numerical tolerance. Coordinate, distance, and area comparisons use an absolute tolerance of $\varepsilon = 10^{-9}$; the derived triangle angle sum uses 10^{-8} degrees. The reported conditions and outcomes make the scope of each check inspectable without treating a computational record as a learner trace.

Bounded adaptation follows the same logic. A teacher who changes the target angle, motif, range, or prompt should update the activity definition and the corresponding parameter bounds and expected relations. New display examples should then be selected, and the grid or sweep should be rerun as a preparation check. A local implementation requires the teacher to author the activity in GeoGebra and apply the stated validation conditions. The procedure makes a change visible and testable before a lesson enters a classroom; its educational value remains a matter for later classroom study.

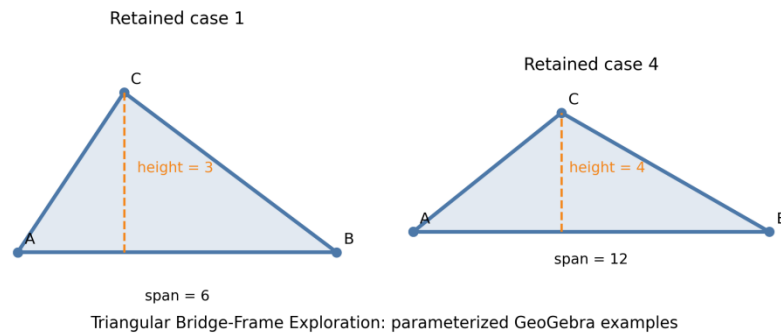
The separation of tools is consequential. GeoGebra provides the direct manipulation through which learners make and test conjectures. Python provides a reproducible check outside the interaction. The audit focuses on whether the resource behaves as specified; a future study of learner understanding would need de-identified work, its own instrument definitions, scoring rules, and data-governance decisions.

3.3 Triangular Bridge-Frame Exploration

The teacher sets three sliders: span s , height h , and horizontal joint position u . Here, s denotes the horizontal span from A to B , h denotes the positive vertical height of the third vertex, and u denotes that vertex's horizontal coordinate measured from A ; A and B are the base endpoints, and C is the third vertex. GeoGebra places them at $A = (0,0)$, $B = (s,0)$, and $C = (u,h)$ and displays polygon ABC , its side lengths, and area. Learners change one control, record a prediction and observation, then use base and height to explain an area change. The bridge-frame question requires a geometric rationale and makes no physical-safety claim.

Five display triples (s, h, u) make the resource inspectable in a lesson plan. The technical audit is broader than those examples. It tests 165 in-range combinations, with integer spans from 2 to 12, heights of 0.25, 1, 2, 4, and 8, and offsets at the left endpoint, midpoint, and right endpoint. Six deliberately out-of-range inputs test input-bound enforcement. Every in-range case must satisfy the triangle inequalities, the base-height relation $A = sh/2$, and an angle sum of 180° within the stated absolute tolerance. Figure 2 shows two display cases rather than a learner dataset.

Figure 2: Representative triangle-bridge parameter cases. The complete audit checks display cases, a parameter grid, and out-of-range inputs

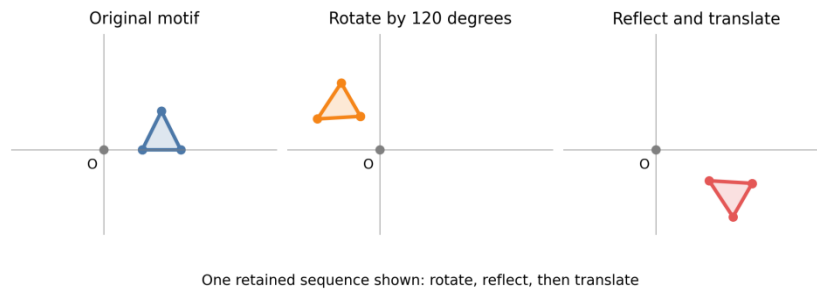


3.4 Transformation-Pattern Design

The second resource uses a triangular motif with an angle slider α . Learners construct the motif, rotate it around an origin, reflect the rotated image across the horizontal axis, and translate the reflected image. For a point (x, y) , where x and y denote its horizontal and vertical coordinates, the audit applies the rotation $R_\alpha(x, y) = (x\cos\alpha - y\sin\alpha, x\sin\alpha + y\cos\alpha)$. It then applies the reflection $S_x(x, y) = (x, -y)$ and the translation $T_{\Delta x, \Delta y}(x, y) = (x + \Delta x, y + \Delta y)$. Learners compare patterns at several angle values, then produce a labelled design with a written or spoken justification. The activity retains the familiar transformation sequence while allowing learners to decide which relation best explains their pattern.

The five display angles are 60° , 90° , 120° , 180° , and 240° . The Python audit extends this set to every 15 degrees from 0° to 360° , and it tests two out-of-range angles. For each in-range angle, it checks radial-distance and edge-length preservation after rotation, edge preservation and coordinate reflection across the horizontal axis, and edge preservation plus the specified displacement after translation. Thus, the audit matches the complete rotate–reflect–translate activity rather than rotation alone. Figure 3 depicts one representative sequence.

Figure 3: Representative motif transformations. Python checks the complete rotate–reflect–translate sequence across the angle sweep



4. Artefact-Fidelity Evidence

The Python audit was executed on the AutoDL SSH backend, which retained the runtime record, parameter inputs, and structured outputs. The triangle resource passed all five display cases and all 165 in-range grid cases. Its six deliberately out-of-range inputs were rejected by the declared slider-domain check. The transformation resource passed all five display cases and all 25 audited angles at 15° increments from 0° through 360° . Its two out-of-range angles were rejected. Each transformation pass includes rotation, reflection, and translation checks, rather than a rotation-only test.

During an audit, the declared task conditions and each parameter value are passed to independent geometric calculations. A pass-or-reject record shows whether the displayed configuration matches the stated condition. When a teacher adapts a prompt for a different grade or setting, this record can reveal a contradiction before the activity is used. The paper supplies the bounds, tolerances, and aggregate results needed to interpret the audit.

The test sets have deliberately different roles. Display cases are compact examples shown in the figures. The triangle grid probes short, tall, centred, and endpoint configurations across the declared domain. The transformation sweep probes a full turn at regular 15-degree intervals and follows the same rotation–reflection–translation order as the GeoGebra task. Out-of-range cases test whether the audit rejects values that the resource has declared unsuitable. Passing a case consequently establishes that the condition is well defined for the documented input; learner problem solving remains a classroom question.

The audit uses independent mathematical calculations from the same specification rather than a direct automated export from GeoGebra. This choice makes the division of labour transparent. Before formal use, the teacher should inspect the activity in GeoGebra and compare its visible controls, labels, and prompts with the stated task conditions. Automated cross-tool screen-level comparison remains a useful future resource-engineering extension, especially after a lesson is localised or the software version changes.

The stronger audit should still be read at its proper level. It samples a declared grid and angle sweep rather than every possible floating-point state or GeoGebra interface interaction. It also does not assess wording clarity, accessibility, teacher orchestration, or how learners reason with the activity. Those questions require formative review and classroom evidence. The present contribution is therefore a technical safeguard for resource design, not an estimate of learner outcomes.

5. Discussion and Conclusion

Two resources place learner activity in GeoGebra: students construct, drag, measure, transform, and explain configurations. A distinct Python procedure inspects the resource specification for teachers. The split assigns interactive work to GeoGebra and validation to Python, so the paper does not conflate the two systems.

Technical evidence comes from the declared input tests. The triangle resource was checked across 165 parameter combinations and rejected six out-of-range inputs. The transformation resource was checked at 25 angles, with rotation, reflection, and translation tested in sequence. Those results establish compliance with the stated geometry; claims about learning, engagement, anxiety, and STEAM competence remain matters for classroom data.

5.1 Future Classroom Evaluation Protocol

A classroom study must address questions that a mathematical audit cannot resolve. It could compare a static worksheet, a GeoGebra-only activity, and the hybrid workflow at a stated grade level and instructional duration. Before recruitment, the protocol needs teacher-preparation arrangements, consent, an assessment rubric, missing-data procedures, and uncertainty analysis. Validated affective measures and de-identified implementation indicators could complement the primary assessment.

Learners' assessments should require them to interpret a changed configuration, identify an invariant, and justify a design decision with geometric evidence. The rubric can score mathematical accuracy, use of a transformation or measurement, and explanation clarity. Reports need to give condition sizes, exclusions, missingness, uncertainty, and teacher variation. Drag and slider records show implementation rather than learning, and should be reported accordingly.

The resulting resources and audit define a reproducible starting point for classroom implementation. Their technical fidelity has been checked, but any claim about educational benefit must come from a subsequent empirical evaluation.

References

- [1] Azis, Y. M., and Rohaeti, E., "A systematic literature review on implementation of GeoGebra: Benefits and challenges in mathematics education," *Infinity Journal*, 14, (3), 655–672, 2025, <https://doi.org/10.22460/infinity.v14i3.p655-672>.
- [2] Fitriana, L., Hendriyanto, A., Sahara, S., Muhaimin, L. H., Wintarti, A., Listiawan, T., Saputro, A. D., and Dewi, D. A., "The role of digital technology in the design of mathematics teaching and learning: A

systematic literature review,” *Journal of Educational Technology and Learning Creativity*, 3, (2), 282–297, 2025, <https://doi.org/10.37251/jetlc.v3i2.2503>.

- [3] Seftiana, G., Lestari, N., Isnawan, M. G., and Rusmayadi, Muh., “Utilization of GeoGebra software in mathematics learning: A literature systematic review,” *Polyhedron International Journal in Mathematics Education*, 2, (1), 156–163, 2024, <https://doi.org/10.59965/pijme.v2i1.95>.
- [4] Dos Santos, J. M. D. S., Silveira, A. P., and Trocado, A. E. S., *GeoGebra e situações que envolvem modelação numa abordagem STEAM*, arXiv preprint arXiv:1907.02099, 2019, <https://arxiv.org/abs/1907.02099>.
- [5] Ziatdinov, R., and Valles, J. R., *Synthesis of modeling, visualization, and programming in GeoGebra as an effective approach for teaching and learning STEM topics*, arXiv preprint arXiv:2202.01415, 2022, <https://arxiv.org/abs/2202.01415>.
- [6] Chytas, C., Borkulo, S. V. van, Drijvers, P., Barendsen, E., and Tolboom, J., “Computational thinking in secondary mathematics education with GeoGebra: Insights from an intervention in calculus lessons,” *Digital Experiences in Mathematics Education*, 10, (2), 228–259, 2024, <https://doi.org/10.1007/s40751-024-00141-0>.
- [7] De La Cruz Serrano, F., *Visualizing the matrix product as a transformation: A task design using GeoGebra in secondary mathematics education*, arXiv preprint arXiv:2602.02747, 2026, <https://arxiv.org/abs/2602.02747>.
- [8] Kirabo, E., Batiibwe, M. S., Mbulankende, J., and Lie, J., “GeoGebra-supported problem-based learning for in-service primary mathematics teachers’ 3D geometry understanding in a core primary teachers’ college in uganda,” *Frontiers in Education*, 10, 1713271, 2025, <https://doi.org/10.3389/educ.2025.1713271>.
- [9] Ramadhan, F., Mahmudi, A., and Nabilla, H. A., “The effectiveness of GeoGebra-assisted realistic mathematics education in enhancing students’ conceptual understanding,” *Edumatica: Jurnal Pendidikan Matematika*, 15, (3), 402–416, 2025, <https://doi.org/10.22437/edumatica.v15i3.45514>.
- [10] Ramidha, N. F., Shodikin, A., and Abdullah, A. H., “The effect of dynamic geometry software-based interactive media with the RME approach in improving critical thinking on the rotation topic,” *Jurnal Pendidikan Progresif*, 15, (3), 1640–1652, 2025, <https://doi.org/10.23960/jpp.v15i3.pp1640-1652>.
- [11] Rexhepi, S., and Mustafa, V., “The inclusion of GeoGebra in learning rotation of figures for the eighth grade and comparison with the classical teaching method,” *Annales Mathematicae et Informaticae*, 2026, <https://doi.org/10.33039/ami.2026.05.004>.
- [12] Hoa, T. A., and Duong, T. M., “Developing learners’ mathematical problem-solving competence in teaching visual geometry to grade 7 students using the dynamic geometry software GeoGebra,” *Vietnam Journal of Education*, 138–149, 2026, <https://doi.org/10.52296/vje.2026.893>.
- [13] Ardina, G., Bautista, G., and Santos, M. S. delos, “GeoGebra integration in mathematics teaching: Bridging research and classroom practice,” *Journal of Interdisciplinary Perspectives*, 3, (8), 845–856, 2025, <https://doi.org/10.69569/jip.2025.505>.
- [14] Aba, M., Cholily, Y. M., Baiduri, Inganah, S., Wula, Z., and Rahardjanto, A., “GeoGebra as a visualization tool: Implications for mathematics anxiety in analytical geometry lectures,” *Jurnal Pendidikan Matematika (JUPITEK)*, 8, (1), 63–77, 2025, <https://doi.org/10.30598/jupitekvol8iss1pp63-77>.
- [15] GeoGebra, *GeoGebra manual*, Online documentation, 2026, <https://geogebra.github.io/docs/manual/en/>.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Open Access

This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

