Human-AI Co-Development of a Geometric Algorithm for Infinite Shuriken Artwork Design Generation

Saleem A. Al Dajani [*]
Department of Civil and Environmental Engineering (CEE),
Massachusetts Institute of Technology (MIT),
77 Massachusetts Avenue, Cambridge, MA, USA 02139
<sdajani@mit.edu>

Abstract

We present a human–AI co-developed geometric algorithm for generating infinite origami shuriken assemblies using radial symmetry, affine transformations, and programmable crease patterns. By combining analytic theory, interactive design tools, and fabrication pipelines, we show how generative AI augments human creativity in digital and physical domains. Our contributions include (1) a framework for $\mathbb{Z}_n$-symmetric gadget assemblies, (2) a geometric algorithm compatible with fabrication, and (3) an interactive web application for real-time co-design. This work exemplifies how humans and machines can collaboratively explore large design spaces and reinterpret historical artifacts through creative computation.

[*]Drive: /gdrive/sdajani-shurikens [1]. Repo: /saleemaldajani/gadgetassembly [2, 3].
Webapp: https://gadgetassembly-production.up.railway.app [4].

Figure 1: **Left:** Folded 32-gadget origami shuriken, adapted from 30-gadget shuriken found to be largest number of edges on YouTube [5]. **Right:** New double unit cell 64-gadget shuriken design.

1 Introduction & Background

The rise of generative AI and large language models (LLMs) prompts an essential question: how can humans and machines collaborate effectively, leveraging their complementary strengths? Beyond productivity, such collaborations help us examine what it means to be human [6], especially as we interact with systems that replicate human work [7], generate content by tuning creativity [8] (e.g., LLM temperature), hallucinate [9], and most crucially, persuade and direct both people and machines. These collaborations deepen cultural understanding and shed light on complexities of human history.

In this work, we present a human-AI co-development of a geometric algorithm for infinite shuriken design generation. This serves as an illustrative example of how machines can enhance human creativity, particularly when navigating large, open-ended design spaces. Historically, humans have always collaborated with the tools of their time to invent and tinker. Our goal is to show that AI can support this culture, facilitating design and discovery in ways that benefit humanity [10], while remaining mindful of the broader responsibilities and potential risks that accompany any powerful

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Creative AI Track.

technology. With the right motivations and oversight, human-AI collaboration can demystify these dangers and guide technology toward good [11], such as designing novel therapeutics [12].

The first digitally recorded shuriken advertisement (Fig. A.1) appeared in *Black Belt* magazine in 1967 (Source: Vintage Ninja [13]). A decade later, shurikens appeared in Kung Fu promotions via an ad in *Asian World of Martial Arts* (Fig. A.2, Vintage Ninja [13]). However, historical research has found no evidence that ninja (忍者) used star-shaped shurikens (手裏剣) [14–16], instead concluding that ninja were intelligence agents, not mystical assassins [15–17]. Recent excavations uncovered archeological specimens hypothesized to be shuriken prototypes (Fig. A.3) [18], including flat, carved stones of varying shapes. Yet no definitive historical design has been identified—leaving open the question: what geometries can shurikens take?

Today, humans can revisit this question with the help of AI-driven tools for simulation and design [19–21]. Despite the lack of historical precedent, shuriken origami continues to thrive through both informal circulation and formal documentation by groups like OrigamiUSA (Source: Squared Ninja Star, The Fold [22]; Star Box Rose, Winter 2024, The Paper [23]; Yellow star, Oversize Competition, Autumn 2024, The Paper [24]). Inspired by these models, we aim to quantify how many edges can be folded into a single shuriken assembly [25], explore edge doubling, and generalize to infinite configurations—much like carving stars from stone A.4.

In an exploratory process, we folded publicly available origami shuriken designs, including the standard 4-edge model (Fig. A.5), its nested 8-edge counterpart, and 4- and 8-edge transformer designs (Fig. A.27) [26], as well as larger assemblies with 16 and 30 edges. While folding the 30-edge design, we observed it could accommodate up to 32 modular gadgets—i.e., origami subroutines (Fig. 1, Fig. A.6). This led us to focus on the underlying gadget unit.

Building on this, we created a 64-edge shuriken using a double-unit gadget (Fig. 1, Fig. A.7), exceeding any design publicly documented online. To explore its full design space, we conducted a human-AI co-design process to generalize and generate infinite variants. This not only extended the catalog of origami shurikens but also demonstrated how structured human-AI collaboration can enable cultural reinterpretation [27], creative innovation [28] and computation [29], and mathematical exploration of large design spaces [30, 31]. With the folded shuriken artworks in hand, different throwing tests (Fig. A.8) can be done by simulating fluid against a fixed rotating shuriken cell to characterize their mechanical and aerodynamic properties. This can enable us to answer whether it was realistic for ninja to use shurikens, since it is currently unclear whether these assemblies can even fly if thrown.

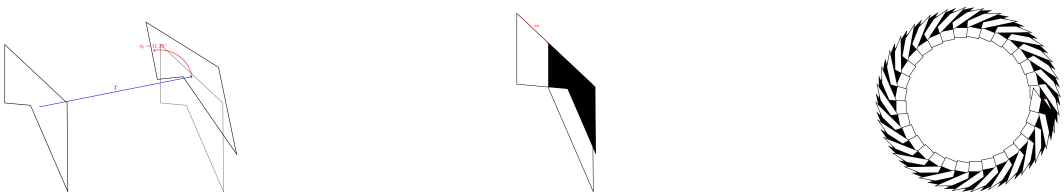

Figure 2: **Left:** First gadget assembly operator. **Center:** Second gadget assembly operator. **Right:** Double-ring assembly with interweaved gadgets.

2 Human-AI Computer-Aided Co-Design

The overall goal of this work is to develop a geometric algorithm to design infinitely many shuriken designs for any given n with a set of continuous parameters. This was done using generative AI foundation models in three parts of human-AI collaboration through co-formulation of theoretical foundations for shuriken gadget assemblies, algorithmic co-development and co-implementation of a shuriken gadget assembly generative tool, and computer-aided co-design of shuriken gadget assembly prototypes. Prototyping was done in a broader human-machine collaborative effort using various 3D printing technologies of the resulting human-AI co-created designs.

2.1 Human-AI Theory Co-Formulation

We describe the geometric construction of the shuriken gadget assembly, starting from a two-unit cell system (Fig. 2, Fig. A.9–A.17) and generalizing to any number n of units. The design is parameterized by the assembly radius R, secondary offset ΔR, offset angle τ, and scale factors s_1 and s_2 for the primary and secondary gadgets. Using n-gon geometry and trigonometric transformations, we generate rotationally symmetric configurations.

This formulation emerged from a human-AI collaboration: initial sketches were developed by hand and then refined using an LLM [32]. The LLM helped formalize variable definitions, apply standard transformations, and accelerate derivations using rotation matrices and symmetry operators.

Each gadget is modeled as a planar polygon $G \subset \mathbb{R}^2$, centered at its centroid c, yielding

$$G_0 = G - c. \tag{1}$$

For n radially arranged gadgets, each instance undergoes rotation R_{ϕ_i} and translation T_i, forming

$$G_i = \{R_{\phi_i}(p) + T_i \mid p \in G_0\}, \tag{2}$$

with the full configuration given by

$$\mathcal{R}_n = \bigcup_{i=0}^{n-1} G_i, \tag{3}$$

defining a $\mathbb{Z}_n$-symmetric tiling in the plane.

The design problem is formalized as: given $n \in \mathbb{N}$ and continuous parameters $\boldsymbol{\alpha} \in \mathbb{R}^k$, find a gadget G, center c, and scale factors s_1, s_2 such that the centered shape G_0 defines a valid unit for modular assembly. The assembly map

$$\mathcal{A} : (n, \boldsymbol{\alpha}) \longmapsto S_{n,\boldsymbol{\alpha}} \subset \mathbb{R}^2 \tag{4}$$

must satisfy:

$$\begin{aligned} &\textbf{Consistency:} \quad && \boldsymbol{\alpha} \text{ corresponds to a realizable origami} \Rightarrow \mathcal{A}(n, \boldsymbol{\alpha}) \text{ reproduces it,} && (5)\\ &\textbf{Diversity:} \quad && \{S_{n,\boldsymbol{\alpha}}\} \text{ contains infinitely many distinct assemblies.} && (6) \end{aligned}$$

2.2 Human-AI Algorithmic Co-Development: Implementing the Algorithm

We developed a webapp to explore and showcase these designs using interactive sliders for key parameters. The tool was first prototyped in a Jupyter notebook (Fig. 4), then converted into a GitHub repository, deployed via Railway. An LLM assisted in code generation [8], building on our co-formulated theory to create the notebook sliders, which then guided web development in Replit. From a Replit interface screenshot, the LLM regenerated the repository for free public deployment via Railway (Fig. 4). The resulting tool illustrates configurations that can be designed and fabricated.

2.3 Human-AI Computer-Aided Co-Design of 3D Model

To demonstrate that the theoretical designs are physically realizable for any n, we prototyped shurikens using paper, PLA, resin, and metal printing. Starting from a single gadget's crease pattern, we simulated folding in Origami Simulator, exported CAD models at various fold stages (Fig. 3), aligned them in MeshLab, generated polar arrays in Rhino, and used MeshMixer to create printable halves and mirrored counterparts. Models were then sliced using printer-specific software (e.g., MakerBot CloudPrint, GrabCAD Print, Desktop Metal Live Studio). PLA artworks were printed at the Martin Trust Center for MIT Entrepreneurship and MIT Morningstar Academy for Design (MAD); resin and metal artworks were printed at the MIT SHED Lab (Fig. 5). An LLM assisted in CAD troubleshooting (e.g., aligning to principal axes in MeshLab) and in generating Python code to voxelize and thicken `.stl` files using the `trimesh` package for post-processing and print preparation.

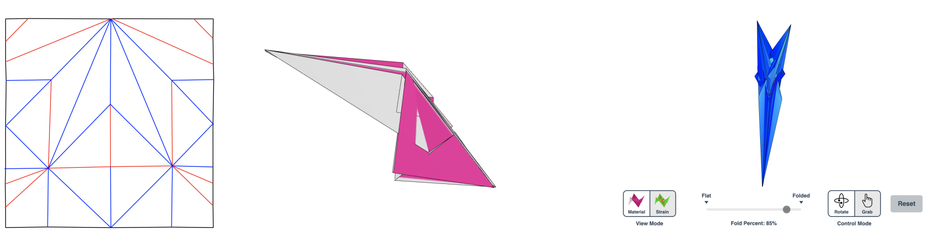

Figure 3: **Left:** Gadget crease pattern. **Center:** Flat-folded state. **Right:** 85% folding insertion.

3 Results

We present a complete pipeline for generating, assembling, and fabricating origami shurikens from geometric and trigonometric principles. Starting with an algorithmically defined crease pattern

(Algorithm 1, Figure A.18), foldable gadget units are arranged via affine transformations—scaling, rotation, and translation—over a discrete n-gon layout, forming single or double concentric rings (Figures A.15–A.17). These constructions are implemented in both a Jupyter notebook and a web-based app for interactive design. To validate physical realizability, we simulate folding, generate watertight meshes, and fabricate prototypes in PLA, UV-cured resin, and stainless steel. These assemblies confirm geometric and structural soundness in digital and physical form.

3.1 Shuriken Assembly Geometric Folding and Assembly Algorithm

We propose a two-part pipeline to generate modular origami shurikens. First, we define a structured crease pattern encoding the local folding logic of a single gadget. Then, we position these gadgets radially using affine transformations controlled by tunable parameters: ring radius, angular spacing, and scale. This enables precise, customizable assembly of single- and double-layered shurikens.

3.1.1 Geometric Gadget Folding Algorithm

Each gadget's crease pattern is derived by folding and unfolding an adapted 30-unit shuriken ([video]), then codified using color-coded folds: black (edges), blue (valley), and red (mountain). Algorithm 1 summarizes the steps: start with a square, fold a central subsquare with valley folds, insert mountain folds at fractional heights for controlled collapse, octisect the top edge for symmetric fan folds, and add vertical/diagonal creases for alignment and stiffness. The resulting crease pattern (Figure A.18) serves as the blueprint for all simulated and fabricated geometries.

3.1.2 Geometric Gadget Assembly Algorithm

We arrange n gadgets radially, with angular positions $\phi_i = \frac{2\pi i}{n}$. For a double-layered design, secondary gadgets are offset by $\psi_i = \phi_i + \frac{\pi}{n}$ and placed at radius $R' = R + \Delta$ (Figures A.15–A.17).

Each gadget undergoes scaling, rotation, and translation:

$$P_i(p) = T_{r_i}\left(R_{\theta_i}(S_{s_i}(p))\right),$$

where s_i is the scale, $\theta_i \in \{\phi_i, \psi_i\}$, and $r_i \in \mathbb{R}^2$ is the radial placement vector. Primary gadgets use (s_1, ϕ_i, R), while secondary ones (if enabled) use (s_2, ψ_i, R').

The full configuration is:

$$\mathcal{S}_n = \bigcup_{i=0}^{n-1} P_i^{(1)} \cup \begin{cases} \bigcup_{i=0}^{n-1} P_i^{(2)} & \text{if double-layered,} \\ \varnothing & \text{otherwise.} \end{cases}$$

Figures A.9–A.14 visualize each transformation step. Assembled gadgets form closed-loop structures (Figure A.13), with optional interleaving for added complexity (Figure A.17).

3.2 Shuriken Design Tool

Our development workflow spans three main components: a Jupyter notebook for prototype logic and visualization, a modular GitHub repository for version control and reproducibility, and a deployed web application for interactive exploration of geometric parameters [33].

3.2.1 Jupyter Notebook: Assembly

The shuriken gadget is flattened to a polygon of its flat-foldable vertices. We then compute the centroid and translate the shape to the origin. These centered coordinates are cached for future use, minimizing redundant computation.

Each gadget is positioned in a ring configuration defined by continuous parameters: angular spacing n, ring radius R, primary and secondary scale factors s_1, s_2, and radial offset Δ. Optionally, gadgets may be arranged in a single ring or as two interleaved concentric rings. For each gadget, we compute its angular position, apply rotation to align tangentially to the ring, scale it, and then translate it to its location. Sliders are created for each parameter using the 'ipywidgets' framework, and are bound to callback functions that trigger updates to the geometric transformations and redraw the visualization in real time. This provides interactive control and feedback for design exploration.

3.2.2 Codespaces and Railway Webapp: Shuriken Design Sliders

The full open-source implementation, including scripts for geometry processing, assembly, and interactivity, is available at `https://github.com/saleemaldajani/gadgetassembly` [2, 3]. A prototype web application was first built on Replit using React and TypeScript for the frontend and Express (Node.js 20) for the backend. The app features a canvas-based renderer and interactive sliders for all key parameters (n, R, s_1, s_2, Δ), enabling live geometric manipulation. With an LLM, the project was then co-developed into a Python codebase via Dash based on screenshots from the Replit frontend. We refactored the logic into Dash components (‘dcc.Slider‘, ‘dcc.Graph‘) and implemented callbacks to recompute transforms and update the figure in response to user input. The web app was tested within GitHub Codespaces from the `main` branch, and verified to replicate the notebook.

The live version of the app[1] was deployed to Railway using a continuous integration and deployment/delivery (CI/CD)-ready setup. The `railway` branch of the repository contains the necessary ‘Dockerfile‘, ‘requirements.txt‘, and ‘Procfile‘ to build a web-accessible container. Railway is connected to the GitHub repo and automatically builds and deploys the `railway` branch application. This setup ensures synchronization between the development codebase and the deployed service.

3.3 Shuriken Artworks

To transition from crease patterns to physical models, we employed a multi-stage workflow combining simulation, mesh processing, and multi-material 3D printing. We begin by simulating the folded geometry from the 2D crease pattern. Figure A.19 shows fully flat-folded and partially folded configurations exported from an origami simulator.[2] The resulting mesh is aligned to its principal axes using MeshLab for standardized orientation (Figure A.20). To generate symmetric structures, we replicate the folded gadget in a polar array using Rhino’s `PolarArray` feature. Figure A.21 shows the result from a single gadget to full ring. The resulting surface mesh is paper-thin, so we apply thickening and voxelization using `trimesh` in Python to produce a watertight 3D volume (Figure A.22). To prepare mirrored geometries, we use MeshMixer to apply symmetry cuts after solidification and automatic repair (Figure A.23).

We fabricated the designs using multiple printing techniques. Figure A.24 shows a prototype printed in PLA using a MakerBot Replicator+. To test finer resolution and edge scaling, we used UV-cured resin via a Stratasys J35 with VeroUltra WhiteS ink, producing 64-, 96-, and 128-edge versions (Figure A.25). We printed the 64-edge design in metal using Desktop Metal’s metal 3D printing system with 17-4 PH stainless steel (Figure A.26).

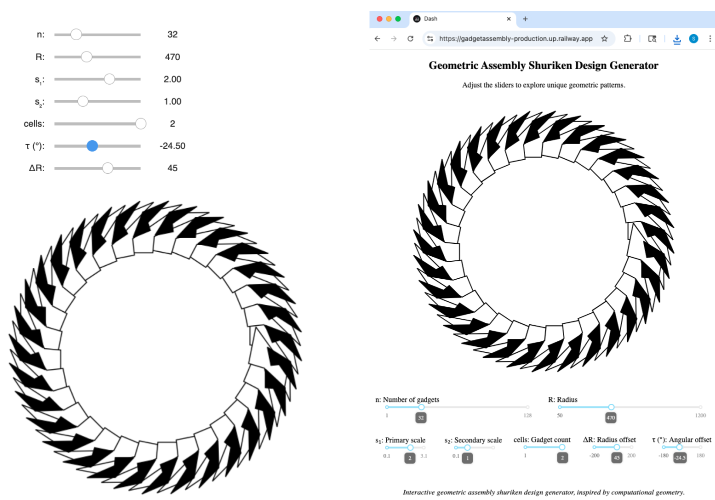

Figure 4: **Left:** Optimal configuration obtained empirically for 64-edge design using Jupyter notebook sliders. **Right:** Same configuration reproduced using Dash webapp sliders.

4 Discussion

Our geometric algorithm shows that for any given n and set of continuous parameters, an infinite set of designs exist for the specific origami shuriken assembly we analyzed. The algorithm also successfully reproduces the geometry of the folded assembly. The analysis of the crease pattern uncovered specific patterns that can likely be extended to design new gadgets and, in turn, new assemblies. This is proposed to be done by using different polygonal starting papers—for example, by hexasecting the first angle on equilateral triangular paper or decisecting it on equilateral pentagonal paper, instead of octisecting it on square paper. By demonstrating a geometric algorithm to design

[1] `https://gadgetassembly-production.up.railway.app/` [4]

[2] `https://youtu.be/z0mzEnLzdus` [34]

infinitely many shurikens for any given n with a set of continuous parameters, this work paves the way for generalizing the approach across families of origami shuriken folds (Fig. A.27) and other gagdet assemblies, such as origami swan assemblies and hypar pavilion structural assemblies.

Based on our interactive webapp, an optimal configuration of continuous parameters (Fig. A.28) can be demonstrated empirically. With constraints on the sliders, the optimal configuration can be solved for by including the edge-to-edge and stopping insertion points of each primary and secondary gadgets. This can be done for dynamically updating sliders based on n, for the continuous parameters to only slide between the constrained parameters of the folded assembly.

4.1 Limitations, Broader Impacts, and Safeguards

Our approach is limited to static, planar shuriken assemblies with interleaved primary and secondary gadgets of identical geometry. We did not account for structural relaxation (e.g., paper loosening) or finite paper thickness, as the analysis is two-dimensional. All designs were manually created and may deviate slightly from simulated configurations. The webapp supports empirical exploration but does not solve for optimal assemblies.

Despite these constraints, the methodology generalizes to transforming and interlocking shurikens, as well as to 3D assemblies when gadgets have a closed-form description (e.g., hypar equations or vertex coordinates) [35]. It can also be extended to heterogeneous structures combining different gadget types, such as hybrid shuriken-hypar assemblies. The interactive tool could be adapted to other fold families and exported to CAD platforms like Rhino or Fusion 360 for precise fabrication.

No safeguards were necessary for origami shurikens. The Webapp is limited to two-dimensional geometries and cannot be used to generate printable CAD files. The shuriken artworks were thickened by voxelization, so the edges are not sharp.

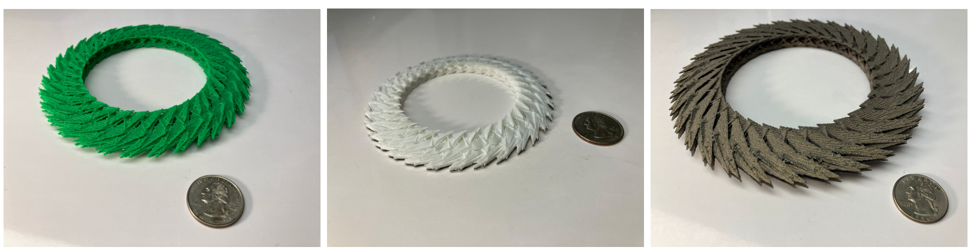

Figure 5: **Left:** PLA print using MakerBot Replicator+. **Center:** Resin prints with 64 edges using Stratasys J35. **Right:** Stainless steel print of 64-edge design using Desktop Metal.

5 Conclusion

This work develops a geometric algorithm to design infinitely many shurikens for any given n with a set of continuous parameters by parameterizing radial symmetry over n-gons and encoding foldable gadget geometries via a structured crease pattern. By combining continuous design parameters with a reproducible folding sequence, we generate interleaved assemblies that exhibit $\mathbb{Z}_n$ symmetry and allow both analytical and empirical design exploration. We validate the design space through an interactive web application and a series of physical prototype artworks, demonstrating consistency between algorithmic output and real-world fabrication [36].

The presented framework lays the foundation for geometric modeling of modular origami assemblies beyond shurikens, including swan clusters, hypar structures, and interlocking gadget systems. Future work may integrate physical constraints—such as insertion tolerances, paper thickness, and relaxation effects—to optimize foldability and mechanical behavior [37]. While empirical calibration using slider-constrained parameters could enable automated searches for valid configurations [38], the ultimate goal is to optimize design parameters in terms of the others. Aerodynamic simulations and physical throw tests may clarify the practical viability of historical shuriken-like designs. Extensions to three-dimensional assemblies, heterogeneous gadget tilings, and CAD-integrated versions of the webapp could significantly broaden the applicability of this method across design, fabrication, and interactive tooling. The method may also generalize across different polygonal bases and angular subdivisions, extending the folding approach beyond square paper to alternative geometries such as equilateral triangles and regular pentagons.

6 NeurIPS Code of Ethics Statement

Research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics [39].

7 Declaration of LLM Usage

This paper adheres to the NeurIPS 2025 Policy on the Use of Large Language Models [40] and transparently documents the ethical use of LLMs throughout the research. LLM involvement is described in detail in Section 2 Human-AI Computer-Aided Co-Design and Section 3.2 Shuriken Design Tool. Transcripts of LLM interactions are included as embedded links that are cited in the paper bibliography, with the specific model used noted in each transcript and in the list of references.

- **Geometry:** Interactive geometry sketch [41]
- **Sliders:** Slider-based design interface [42]
- **Gadget placement:** Gadget placement configuration [43]
- **Explaining code:** Code explanation and logic [44]
- **Replit workspace:** Live Replit workspace [45]
- **Replit development URL:** Deployed dev environment [46]
- **Reproducing Replit website:** Replit-to-website setup [47]
- **Using Railway:** Deployment via Railway [48]
- **Problem statement:** Formal problem formulation [49]
- **Beautify:** Diagram beautification [50]
- **Tau:** Incorporating τ parameter [51]
- **Paper-writing:** Creative AI paper drafting [52]
- **Organizing transcripts:** Transcript organization [53]

Acknowledgments and Disclosure of Funding

S.A.A. thanks Erik Demaine for teaching the MIT class on Geometric Folding Algorithms (6.5310, Spring 2025) in which this work was a final project, and for advising this research. S.A.A. thanks Michael P. Short for his gracious support, and for inviting him over to the lab to fold artworks while shadowing ring-making. S.A.A. was funded by the MIT Department of Civil and Environmental Engineering (CEE) under the Friesecke (1961) Fellowship Fund. S.A.A. acknowledges support from Omar Al Dajani, James Weaver, Stephen Rudolph, Kimberly Benard, Jack Greenfield, Duc-Tam Ho, Josephine Carstensen, John Ochsendorf, Abel Sanchez, John R. Williams, Franz-Josef Ulm, Herbert Einstein, Tal Cohen and Cohen Lab members, Lydia Bourouiba, Haruko Wainright, Heidi Nepf, and Henning Soller. S.A.A. thanks his rotation lab members, Elvira Kinzina, Ju Jung, Owen Mulledy, Oscar Pitcho, Jason Lequyer, Nic Fishman, Tanush, Sanah Langer, Patrick Paine, Thomas Kesheshian, Janet Gallo, Dmitrii Glubokov, Jesse Poganik, and his rotation lab principal investigators, Jonathan Gootenberg, Omar Abudayyeh, and Vadim Gladyshev. S.A.A. is grateful for support from his family and friends, particularly his daughter, Raidah, waking up saying 'Baba made a star!' after the first successful 3D-printed artwork and 'I need this!' after seeing the webapp. Thank you, Sarah.

References

[1] Saleem A. Al Dajani. Al Dajani_Saleem_6.5310 Final Project Notebook. https://drive.google.com/drive/folders/1GKA_gBU6Rcpk-0kNujIO-ewnDmQW1oYG?usp=drive_link, 2025. Final project for 6.5310: Geometric Folding Algorithms.

[2] Saleem A. Al Dajani. Geometric assembly shuriken design generator. https://github.com/saleemaldajani/gadgetassembly, 2025. GitHub repository.

[3] 65310-2025. gadgetassembly. https://github.com/65310-2025/gadgetassembly, 2025. GitHub repository.

[4] Saleem A. Al Dajani. Geometric Assembly Shuriken Design Generator Webapp. https://gadgetassembly-production.up.railway.app/, 2025. Interactive web application.

[5] Paper Folds – Origami & Crafts. 30 point ninja star blade | wreath | modular origami - 1054. https://youtu.be/EpmffTtMEQc?si=9mtDD9ltPlYYzxUi, 2023. YouTube video.

[6] Michael Muller, Plamen Agelov, Shion Guha, Marina Kogan, Gina Neff, Nuria Oliver, Manuel Gomez Rodriguez, and Adrian Weller. NeurIPS 2021 Workshop Proposal: Human Centered AI. *Conference and Workshop on Neural Information Processing Systems*, 2021.

[7] Harry H Jiang, Lauren Brown, Jessica Cheng, Mehtab Khan, Abhishek Gupta, Deja Workman, Alex Hanna, Johnathan Flowers, and Timnit Gebru. AI Art and its Impact on Artists. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 363–374, 2023.

[8] Hai Dang, Lukas Mecke, Florian Lehmann, Sven Goller, and Daniel Buschek. How to prompt? Opportunities and challenges of zero-and few-shot learning for human-AI interaction in creative applications of generative models. *CHI'22 Workshops, May 10, 2022, New Orleans, LA. arXiv: 2209.01390*, 2022.

[9] Yunpu Zhao, Rui Zhang, Wenyi Li, and Ling Li. Assessing and understanding creativity in large language models. *Machine Intelligence Research*, 22(3):417–436, 2025.

[10] Diego Misseroni, Phanisri P Pratapa, Ke Liu, Biruta Kresling, Yan Chen, Chiara Daraio, and Glaucio H Paulino. Origami engineering. *Nature Reviews Methods Primers*, 4(1):40, 2024.

[11] Ramya Srinivasan and Kanji Uchino. The role of arts in shaping AI ethics. In *AAAI Workshop on reframing diversity in AI: Representation, inclusion, and power*, 2021.

[12] Kyle Swanson, Wesley Wu, Nash L. Bulaong, John E. Pak, James Zou, et al. The Virtual Lab of AI agents designs new SARS-CoV-2 nanobodies. *Nature*, 2025.

[13] Keith J. Rainville. The oldest days of the shuriken trade. Online, via Vintage Ninja: `https://vintageninja.net/the-oldest-days-of-the-shrunken-trade/`, 2015. Blog post / web article.

[14] Feodor Kubasov. (The Medieval Period: Didn't a Star-Shaped Shuriken Exist?). In Yamada, Yuji and Mie University International Ninja Research Center, editor, *(Lectures on Ninja Studies)*, pages 195–198. CHUOKORON-SHINSHA, Tokyo, 2 edition, 2020.

[15] Miki Hashimoto. Analysis of the Applicability of Ninja Intelligence and Arts in the Contemporary World. Master's thesis, International Master in Security, Intelligence and Strategic Studies (IMSISS), August 2022.

[16] Stephen Turnbull. *Ninja: Unmasking the Myth*. Frontline Books, London, 2018.

[17] Lydia Querry. Shinobi to Ninja: How Historical Reality Shaped Modern Myth. *The Hanover Historical Review*, 19:63–70, 2024. Senior history essay.

[18] Jack Hunter. Archaeologists in Japan uncover possible early version of ninja throwing stars, 2023. CNN Style, published December 15, 2023.

[19] Songwei Ge, Austin Dill, Eunsu Kang, Chun-Liang Li, Lingyao Zhang, Manzil Zaheer, and Barnabas Poczos. Developing creative AI to generate sculptural objects. *Proceedings of International Symposium on Electronic Art (ISEA 2019). arXiv: 1908.07587*, 2019.

[20] Jianan Jiang, Di Wu, Hanhui Deng, Yidan Long, Wenyi Tang, Xiang Li, Can Liu, Zhanpeng Jin, Wenlei Zhang, and Tangquan Qi. HAIGEN: Towards human-AI collaboration for facilitating creativity and style generation in fashion design. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(3):1–27, 2024.

[21] Richard Lee Davis, Thiemo Wambsganss, Wei Jiang, Kevin Gonyop Kim, Tanja Käser, and Pierre Dillenbourg. Fashioning creative expertise with generative AI: Graphical interfaces for design space exploration better support ideation than text prompts. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–26, 2024.

[22] Sy Chen. Diagrams: Squared ninja star. Online article in TheFold, Issue 81 (March–April 2024): `https://origamiusa.org/thefold/article/diagrams-squared-ninja-star`, 2024. Modular ninja star design with diagrams; edited by Jane Rosemarin.

[23] Shoko Aoyagi. Star box rose. The Paper, Winter 2024 Issue 146 (Winter 2024): `https://origamiusa.org/thepaper/paper-issue-146-winter-2024`, 2024. Diagram by Shoko Aoyagi in The Paper Issue146 (Winter 2024).

[24] Yellow star. The Paper, Issue 145 (Autumn 2024): `https://origamiusa.org/thepaper/paper-issue-145-autumn-2024`, 2024. Oversize Competition diagram, page 5; featured during "2024 Annual Convention: Peace, Love, Origami!" introduction by Eric Ma.

[25] Reymond Akpanya, Tom Goertzen, and Alice C Niemeyer. Topologically Interlocking Blocks inside the Tetroctahedrille. *arXiv:2405.01944*, 2024.

[26] Rishi Balaji. Geometry and Origami. In *2023 IEEE Integrated STEM Education Conference (ISEC)*, pages 210–213. IEEE, 2023.

[27] Jinggang Zhuo, Ling Fan, and Harry Jiannan Wang. A framework and dataset for abstract art generation via CalligraphyGAN. *NeurIPS 2020 Workshop on Machine Learning for Creativity and Design, Vancouver, Canada. arXiv: 2012.00744*, 2020.

[28] Fu Feng, Yucheng Xie, Xu Yang, Jing Wang, and Xin Geng. Redefining <creative> in dictionary: Towards an enhanced semantic understanding of creative generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18444–18454, 2025.

[29] Sebastian Berns and Simon Colton. Bridging Generative Deep Learning and Computational Creativity. In *ICCC*, pages 406–409, 2020.

[30] Giorgio Franceschelli and Mirco Musolesi. On the creativity of large language models. *AI & SOCIETY*, pages 1–11, 2024.

[31] Mete Ismayilzada, Debjit Paul, Antoine Bosselut, and Lonneke van der Plas. Creativity in AI: Progresses and challenges. *arXiv: 2410.17218*, 2024.

[32] Xiaomei Li, Alex Whan, Meredith McNeil, David Starns, Jessica Irons, Samuel C Andrew, and Rad Suchecki. A conceptual framework for human—AI collaborative genome annotation. *Briefings in Bioinformatics*, 26(4):bbaf377, 2025.

[33] Qiyu Chen, Richa Mishra, Lia Sparingga Purnamasari, Dina EL-Zanfaly, and Kris Kitani. Origami Sensei: A Mixed Reality AI-Assistant. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2025.

[34] Saleem Al Dajani. shuriken gadget folding .stl export. YouTube video: `https://www.youtube.com/watch?v=z0mzEnLzdus`, 2025. Uploader: Saleem Al Dajani.

[35] Chuan Qiao, Filippo Agnelli, Deepak Kumar Pokkalla, Nicholas D'Ambrosio, and Damiano Pasini. Anisotropic morphing in bistable kirigami through symmetry breaking and geometric frustration. *Advanced Materials*, 36(23):2313198, 2024.

[36] Mohod Madhu. AI as Artist: The Role of Generative Algorithms in Creative Industries. *International Journal of Advanced Research in Education and Technology*, 12(2):542–548, 2025.

[37] Rui Peng and Gregory S Chirikjian. Thick-panel origami structures forming seamless surfaces. *Nature Communications*, 16(1):3881, 2025.

[38] Chuan Qiao, Shijun Chen, Yu Chen, Zhihong Zhou, Wentao Jiang, Qingyuan Wang, Xiaobao Tian, and Damiano Pasini. Inverse Design of Kirigami through Shape Programming of Rotating Units. *Physical Review Letters*, 134(17):176103, 2025.

[39] Neural Information Processing Systems (NeurIPS) Foundation. Neurips code of ethics. Online at NeurIPS website: `https://neurips.cc/public/EthicsGuidelines`, 2025. Guidelines on research ethics, broader societal impact, and reviewer responsibilities for NeurIPS submissions.

[40] NeurIPS Conference. Neurips2025 policy on the use of large language models. NeurIPS official website: `https://neurips.cc/Conferences/2025/LLM`, 2025. Program Chairs: Nancy Chen, Marzyeh Ghassemi, Piotr Koniusz, Razvan Pascanu, Hsuan-Tien Lin; Assistant Program Chairs: Elena Burceanu, Junhao Dong, Zhengyuan Liu, Po-Yi Lu, Isha Puri.

[41] Saleem A. Al Dajani. Geometric relations in gadgets. ChatGPT 4o conversation: https://chatgpt.com/share/68160331-99c4-8008-b57c-0e7966292971, 2025. Generated with ChatGPT 4o, shared via chatgpt.com.

[42] Saleem A. Al Dajani. Adding r slider. ChatGPT 4o conversation: https://chatgpt.com/share/68160334-efec-8008-bccd-294c0485b0e2, 2025. Generated with ChatGPT 4o, shared via chatgpt.com.

[43] Saleem A. Al Dajani. Optimizing gadget placement. ChatGPT 4o conversation: https://chatgpt.com/share/681610e2-b858-8008-a987-421b18b05039, 2025. Shared via chatgpt.com.

[44] Saleem A. Al Dajani. Gadget ring construction. ChatGPT o4-mini conversation: https://chatgpt.com/share/681610ec-db28-8008-a689-faf68ebea0d1, 2025. Code explanation and logic, shared via chatgpt.com.

[45] Saleem A. Al Dajani. Interactive geometric pattern generator: A react and typescript web application. Live Replit workspace: https://replit.com/@saleemabdulfatt/MarketingMaster-1, 2025.

[46] Saleem A. Al Dajani. Deployed dev environment for interactive geometric pattern generator. Replit development deployment preview: https://be683ac0-bdac-4987-b76d-e7006fdd6558-00-esnsay4j68mg.kirk.replit.dev/, 2025. App may be inactive; deployment requires manual restart.

[47] Saleem A. Al Dajani. Github codespace slider app. ChatGPT 4o conversation: https://chatgpt.com/share/6803a55e-23e4-8008-9401-42ddc37735da, 2025. Replit-to-website setup, shared via chatgpt.com.

[48] Saleem A. Al Dajani. How railway hosts websites. ChatGPT 4o conversation: https://chatgpt.com/share/6816116f-9cf0-8008-8819-a5bfb1eeab68, 2025. Deployment via Railway, shared via chatgpt.com.

[49] Saleem A. Al Dajani. Gadget placement problem formulation. ChatGPT o4-mini-high conversation: https://chatgpt.com/share/681af658-64a4-8008-84a4-d84e86a8db0e, 2025. Formal problem formulation, shared via chatgpt.com.

[50] Saleem A. Al Dajani. Beautify diagrams request. ChatGPT 4o conversation: https://chatgpt.com/share/681cbd00-152c-8008-b269-5242a6ee8efb, 2025. Diagram beautification, shared via chatgpt.com.

[51] Saleem A. Al Dajani. Contour detection improvement. ChatGPT 4o conversation: https://chatgpt.com/share/681cbd10-c060-8008-a976-40bfbb15191d, 2025. Incorporating τ parameter, shared via chatgpt.com.

[52] Saleem A. Al Dajani. Beamer to latex conversion. ChatGPT 4o conversation: https://chatgpt.com/share/68208164-8480-8008-9e71-e9b670f2a051, 2025. Creative AI paper drafting, shared via chatgpt.com.

[53] Saleem A. Al Dajani. Latex link embedding. ChatGPT 4o conversation: https://chatgpt.com/share/6821af12-d1a0-8008-8a28-3d71f220e6ca, 2025. Transcript organization, shared via chatgpt.com.

[54] Joe_Tutorials. How to fold an origami ninja star / shuriken. Instructables, under Craft > Paper: https://www.instructables.com/How-to-Fold-an-Origami-Ninja-Star-Shuriken/, 2015. Origami tutorial with step-by-step photos.

[55] binthewin. The different kinds of shurikens and ways to throw them. Reddit post in r/coolguides: https://www.reddit.com/r/coolguides/comments/ru5dza/the_different_kinds_of_shurikens_and_ways_to/, 2022. Accessed via Reddit, user-generated content.

A Technical Appendices and Supplementary Material

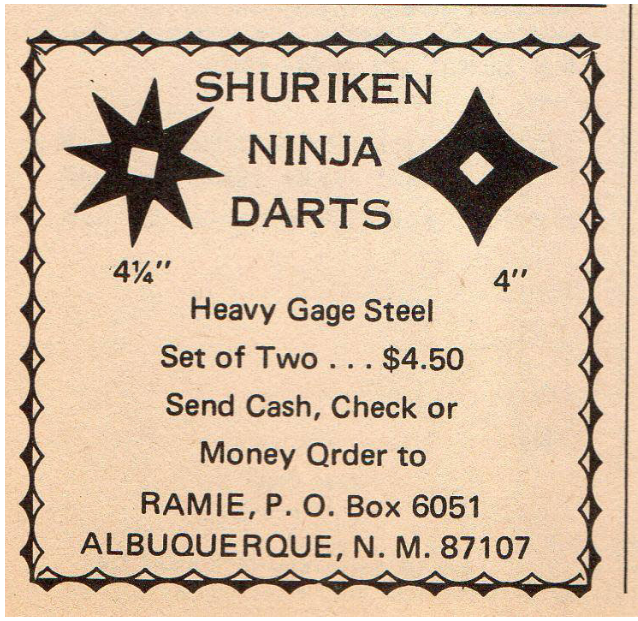

Figure A.1: Oldest ad found for shurikens from the *Black Belt* magazine in 1967 (Source: Vintage Ninja [13]).

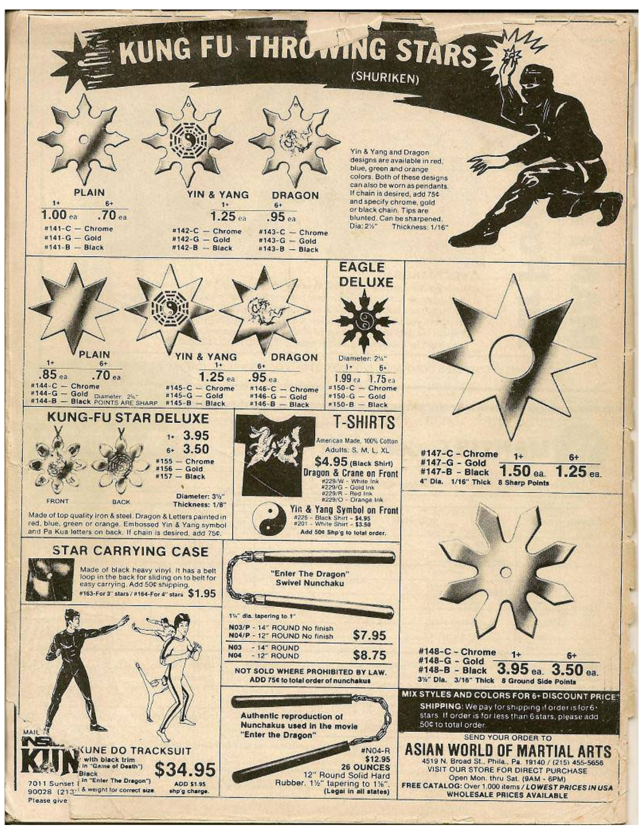

Figure A.2: Later ad found around 1977 for shurikens from the *Asian World of Martial Arts* (Source: Vintage Ninja [13]).

Figure A.3: Recent 2023 excavation uncovered stone specimens hypothesized to be the origin of shurikens (Source: Akihiro Iwata, Saitama Prefecutural Historical Museum, 2023 [18]).

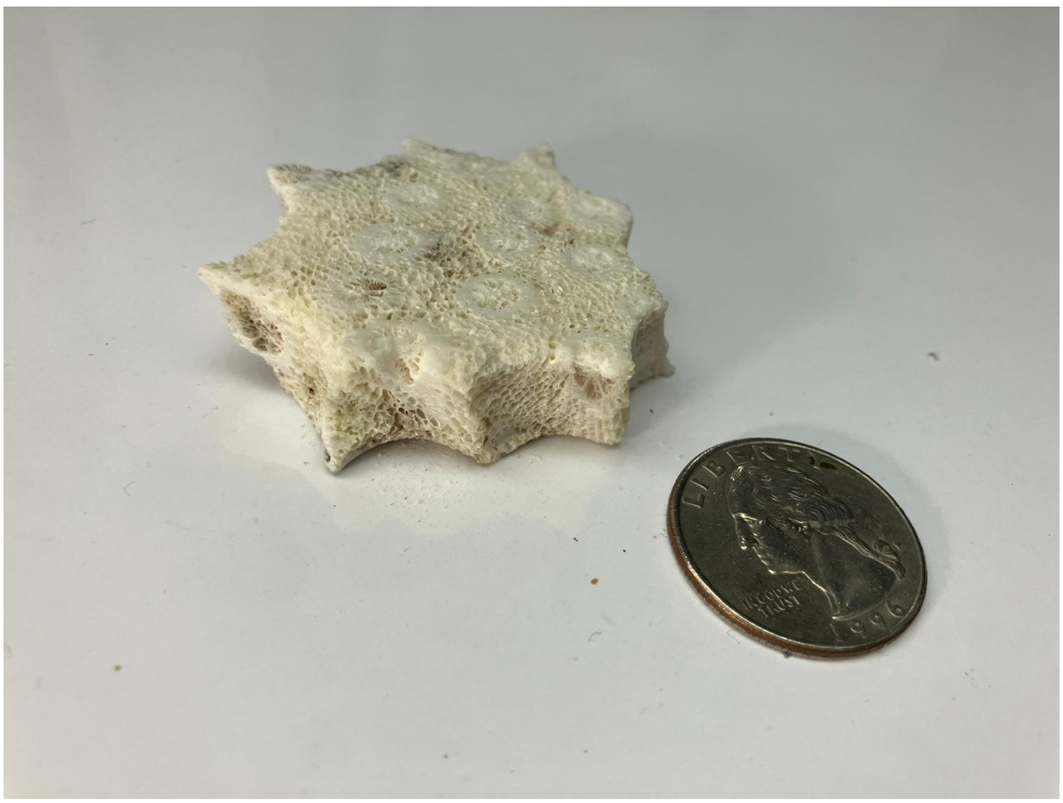

Figure A.4: Carved using a variable speed rotary tool and a diamond rotary point set from a Red Sea coral after two bandsaw $\sim$1 inch slices from a calcified coral skeleton specimen collected on a Red Sea shore in 2022. The specimen is primarily calcium carbonate, $CaCO_3$.

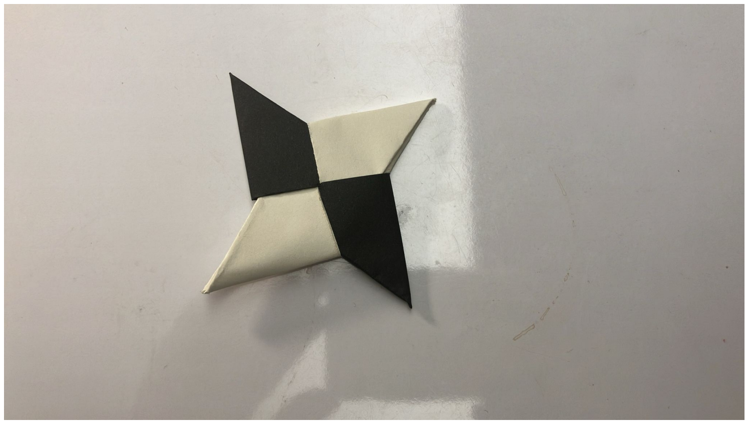

Figure A.5: Most common origami two-gadget shuriken design (Source: Instructables [54]).

Figure A.6: Folded 32-gadget origami shuriken, adapted from 30-gadget shuriken found to be largest number of edges on YouTube [5].

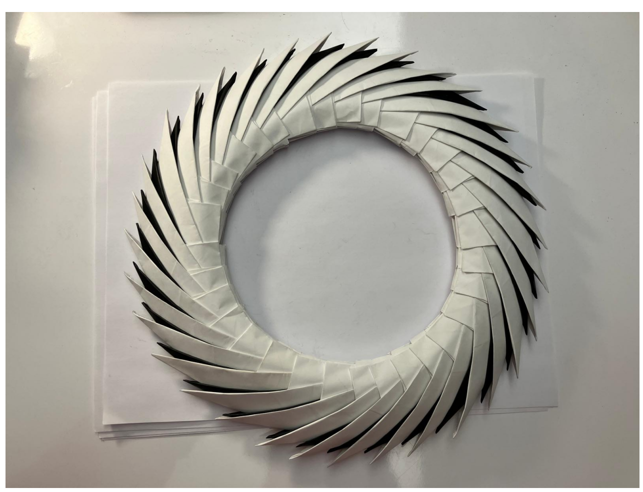

Figure A.7: New double unit cell 64-gadget shuriken design.

Figure A.8: Different throwing tests could be done by simulating fluid against fixed shuriken cell (Source: Reddit/CoolGuides [55]).

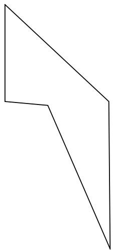

Figure A.9: Base gadget geometry.

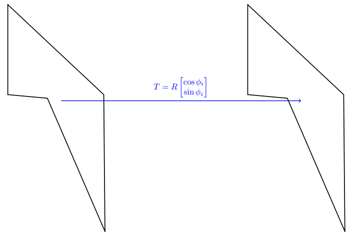

Figure A.10: Translation operator T.

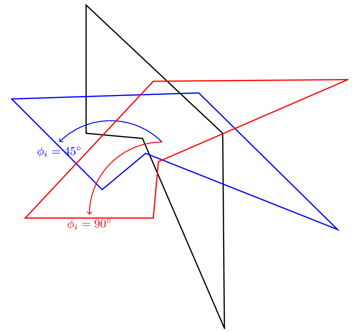

Figure A.11: Rotation operator R.

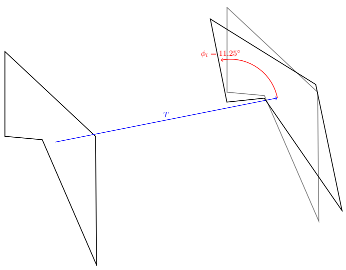

Figure A.12: Combined assembly operator.

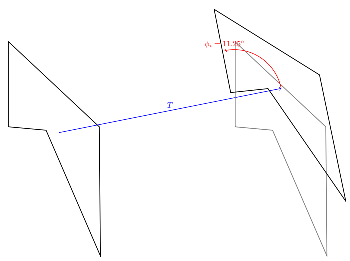

Figure A.13: Two-gadget configuration.

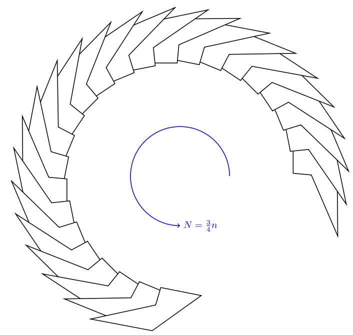

Figure A.14: Replicating N gadgets.

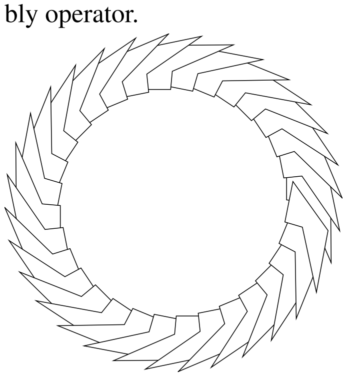

Figure A.15: Ring configuration.

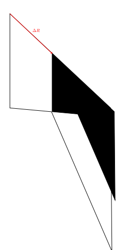

Figure A.16: Double gadget.

Figure A.17: Double-ring assembly with interweaved gadgets.

Algorithm 1 Shuriken gadget crease pattern algorithm

1: Black borders around square perimeter
2: Blue valley fold lines from midpoint of each square edge, forming a subsquare
3: Horizontal blue mountain fold, 4.5/15 down from top border, extended left and right to intersect inner square
4: Horizontal blue mountain fold, 4.5/15 up from bottom border, extended left and right to intersect inner square
5: Octisect upper midpoint; then:
    1. Red mountain folds across first octisection
    2. Blue valley folds across second octisection (already drawn)
    3. Blue valley folds from border to intersection in third octisection
    4. Blue mountain fold at 6/15 of the distance down from upper midpoint
6: Red mountain fold along central octisection, 4.5/15 down from midpoint
7: Blue mountain fold along central octisection, 4.5/15 down to bottom border
8: Connect left/right ends of step 4 with red horizontal line
9: Vertical red mountain folds from inner ends of horizontal lines (steps 3 and 4)
10: From intersection in step 4 and inner square, draw vertical blue valley fold to bottom border
11: Red mountain folds bisecting 90° angle between step 4 horizontals and step 10 verticals
12: Additional red mountain folds bisecting 45° between step 4 and step 11 bisectors
13: Two red mountain folds at top corners:
    - From 1.5/15 along top border at 45° to 1.5/15 down left border
    - From 13.5/15 along top border at 45° to 1.5/15 down right border

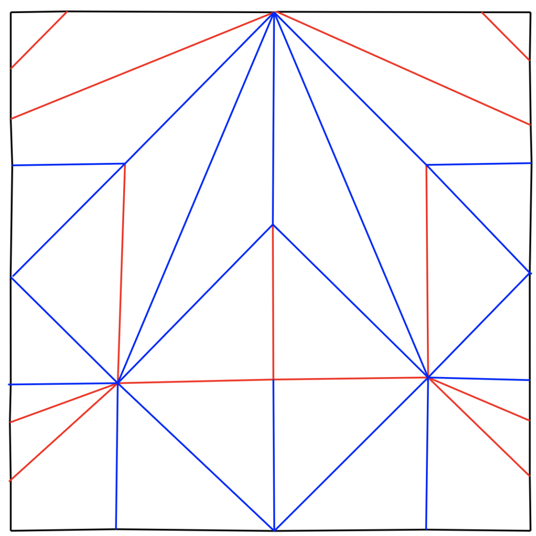

Figure A.18: Crease pattern for a single shuriken gadget.

Figure A.19: Folded states of crease pattern. Left: flat-folded. Right: 85% folding insertion.

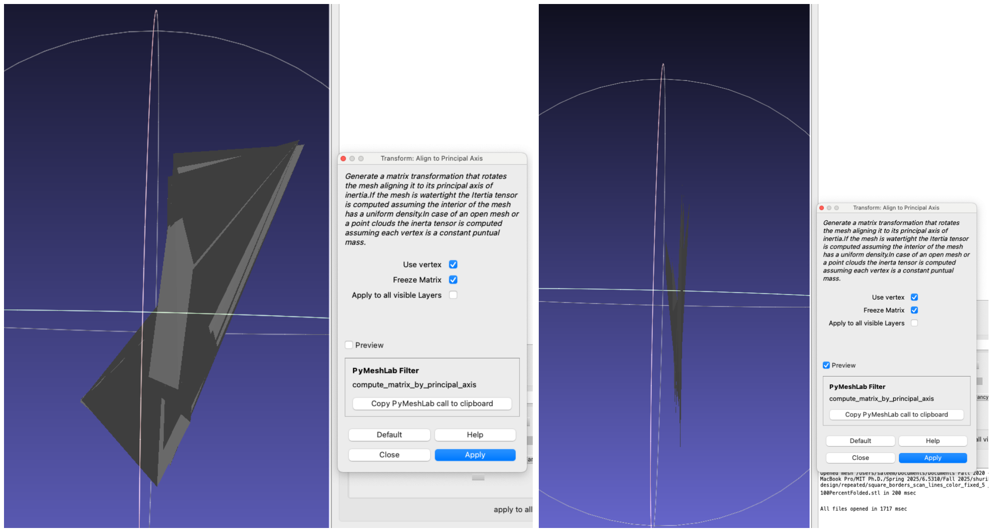

Figure A.20: Left: unaligned geometry. Right: aligned using principal axes in MeshLab.

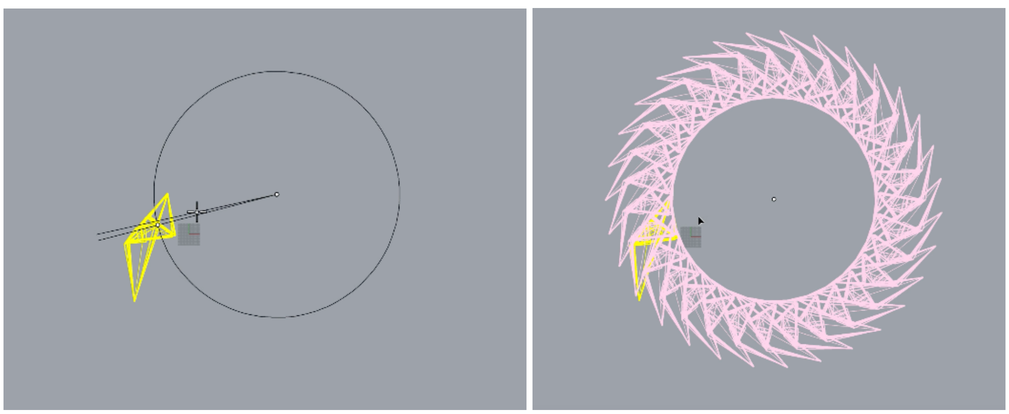

Figure A.21: Left: single gadget. Right: polar array in Rhino.

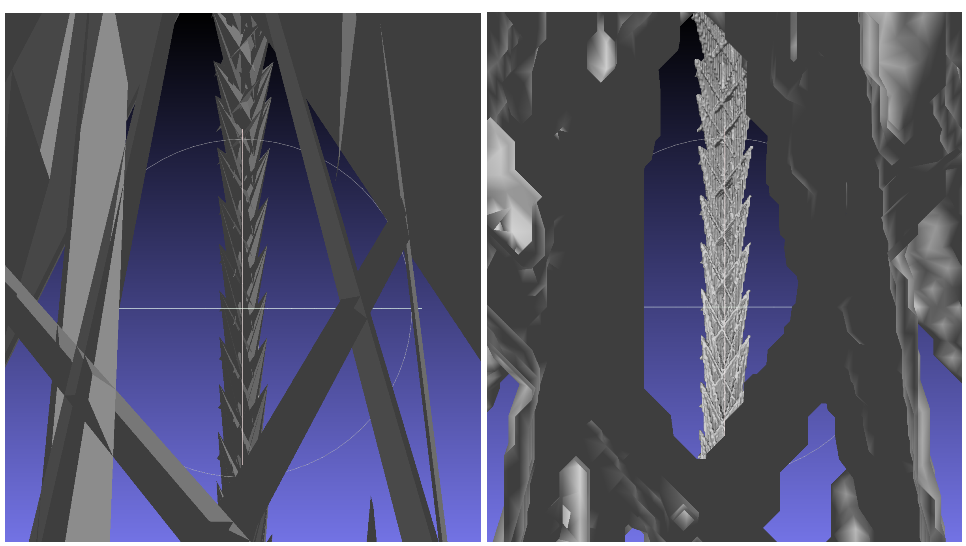

Figure A.22: Left: thin surface mesh. Right: thickened and voxelized using Python.

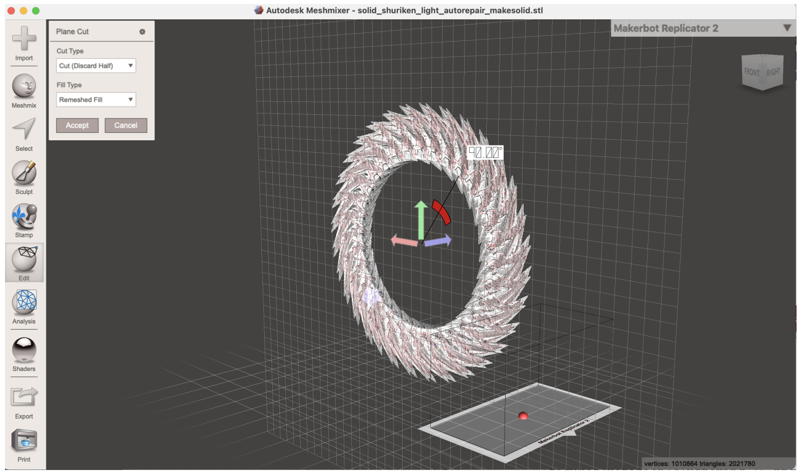

Figure A.23: MeshMixer cut at 90° after applying autorepair and solidification.

Figure A.24: PLA print using MakerBot Replicator+.

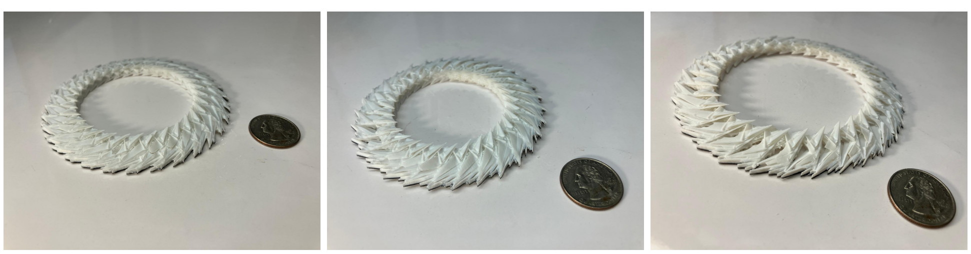

Figure A.25: Resin prints with 64, 96, and 128 edges (left to right) using Stratasys J35.

Figure A.26: Stainless steel prints of 64-edge design using Desktop Metal.

Figure A.27: Origami shurikens with different gadgets and assemblies.

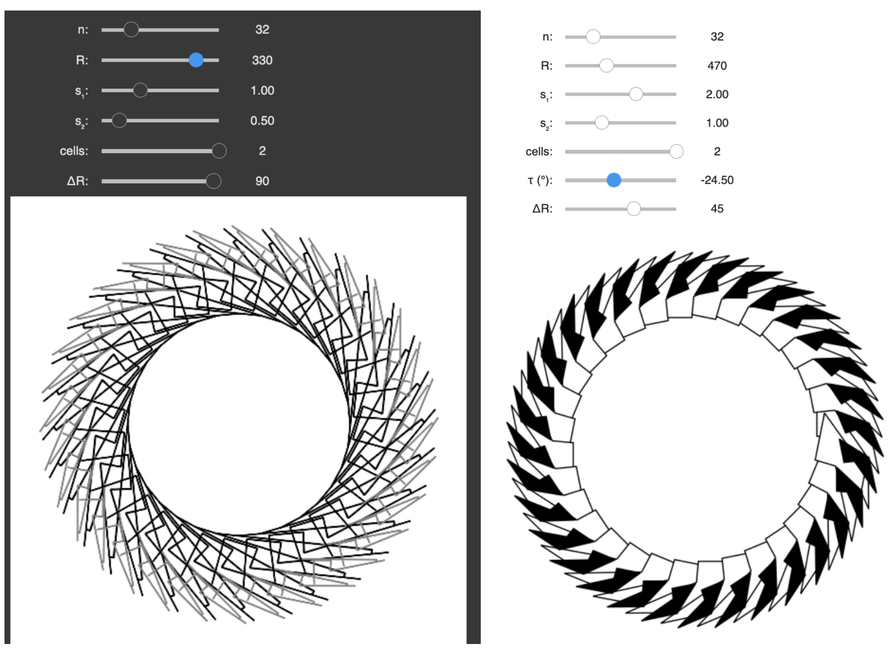

Figure A.28: Optimal configuration obtained empirically for 64 edge design.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope. Our contributions include (1) a formal framework for $\mathbb{Z}_n$-symmetric gadget assemblies, (2) a structured crease-pattern algorithm compatible with fabrication, and (3) an interactive web application for real-time co-design.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations of the work performed by the authors. This can be found in section 4.1 Limitations, Broader Impacts, and Safeguards.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper provides the full set of assumptions and a complete (and correct) proof. The methodology can be found in section 2.1 Human-AI Theory Co-Formulation and the theoretical results in section 3.1 Shuriken Assembly Geometric Folding Algorithm.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper. The code is also provided, and any omitted designs are excluded for safety reasons. This information is cited in the paper bibliography and embedded in URLs in the footnotes or main text, and included here: Drive: /gdrive/sdajani-shurikens [1]. Repo: /saleemaldajani/gadgetassembly [2, 3]. Webapp: https://gadgetassembly-production.up.railway.app [4]..

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper provides open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results. This information is cited in the paper bibliography and embedded in URLs in the footnotes or main text, and included here: Drive: /gdrive/sdajani-shurikens [1]. Repo: /saleemaldajani/gadgetassembly [2, 3]. Webapp: https://gadgetassembly-production.up.railway.app [4]..

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

Justification: The experimental setting is presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them. This information can be found in section 2.2 Human-AI Algorithmic Co-Development and section 2.3 Human-AI Computer-Aided Co-Design, with their respective results in section 3.2 Shuriken Design Tool and section 3.3 Shuriken Artworks.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: The experimental results in the paper are not statistical.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides sufficient information on the computer resources needed to reproduce the results. This information is cited in the paper bibliography and embedded in URLs in the footnotes or main text, and included here: Drive: /gdrive/sdajani-shurikens [1]. Repo: /saleemaldajani/gadgetassembly [2, 3]. Webapp: https://gadgetassembly-production.up.railway.app [4]..

9. **Code of ethics**

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics [39].

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses both potential positive societal impacts and negative societal impacts of the work performed. This can be found in section 4.1 Limitations, Broader Impacts, and Safeguards.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: The paper describes safeguards that have been put in place for responsible release of data or models that have a high risk for misuse, such as thickening and voxelizing the designs to avoid any sharp edges and to increase mass to reduce aerodynamic efficiency. This can be found in section 4.1 Limitations, Broader Impacts, and Safeguards.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators and original owners of the assets used in the paper are properly credited. No licenses or terms of use were needed to perform the work in this research, besides the MIT license on the GitHub repository found here: /saleemaldajani/gadgetassembly.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: New assets introduced in the paper are well-documented, and the documentation is provided alongside the assets, such as a detailed README file on the repository and well-commented code. This information can be found here: Repo: /saleemaldajani/gadgetassembly [2, 3].

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The paper describes the ethical usage of LLMs in this research according to the NeurIPS 2025 Policy on the Use of Large Language Models [40]. This can be found in Section 2 Human-AI Computer-Aided Co-Design, Section 3.2 Shuriken Design Tool, and Section 7 Declaration of LLM Usage. For reference, transcripts of LLM interactions are included as embedded links that are cited in the paper bibliography, with the specific model used noted in each transcript and in the list of references.

    - **Geometry:** Interactive geometry sketch [41]
    - **Sliders:** Slider-based design interface [42]
    - **Gadget placement:** Gadget placement configuration [43]
    - **Explaining code:** Code explanation and logic [44]
    - **Replit workspace:** Live Replit workspace [45]
    - **Replit development URL:** Deployed dev environment [46]
    - **Reproducing Replit website:** Replit-to-website setup [47]
    - **Using Railway:** Deployment via Railway [48]
    - **Problem statement:** Formal problem formulation [49]
    - **Beautify:** Diagram beautification [50]
    - **Tau:** Incorporating τ parameter [51]
    - **Paper-writing:** Creative AI paper drafting [52]
    - **Organizing transcripts:** Transcript organization [53]