ASME Journal of Mechanical Design
Online journal at:
https://asmedigitalcollection.asme.org/mechanicaldesign

Christoph Wittig Adão[1]
IPEK—Institute of Product Engineering,
Karlsruhe Institute of Technology (KIT),
Karlsruhe 76131, Germany
e-mail: christoph.wittig@kit.edu

Jiahang Li
IPEK—Institute of Product Engineering,
Karlsruhe Institute of Technology (KIT),
Karlsruhe 76131, Germany
e-mail: jiahang.li@kit.edu

Markus Döllken
IPEK—Institute of Product Engineering,
Karlsruhe Institute of Technology (KIT),
Karlsruhe 76131, Germany
e-mail: markus.doellken@kit.edu

Maxim Möllhoff
IPEK—Institute of Product Engineering,
Karlsruhe Institute of Technology (KIT),
Karlsruhe 76135, Germany
e-mail: maxim.moellhoff@student.kit.edu

Xingang Li
Department of Mechanical Engineering,
The University of Melbourne,
Victoria 3010, Australia
e-mail: xingang.li@unimelb.edu.au

Sven Matthiesen
IPEK—Institute of Product Engineering,
Karlsruhe Institute of Technology (KIT),
Karlsruhe 76131, Germany
e-mail: sven.matthiesen@kit.edu

SheetGen: A Rule-Based Generative Design Algorithm for Manufacturability-Focused Sheet Metal Parts Based on Domain-Specific Design Knowledge

Generative design can accelerate early exploration of design solutions, yet generated designs often fail to meet manufacturing requirements. This shortcoming stems from a disconnect between the generation process of the solutions and the domain-specific design knowledge that experienced engineers apply to ensure manufacturability. To address this gap, this article presents SheetGen, a rule-based generative design algorithm for sheet metal bending parts that systematically translate domain-specific design knowledge into algorithmic rules enforced directly during part generation rather than through post hoc filtering. The algorithm operates on edge connections between planar surfaces and embeds manufacturing requirements at distinct stages of the generative design process, distinguishing between preprocessing, in-process, and postprocessing integration points. Generated solutions are exported directly into a commercial computer-aided design (CAD) environment and validated through industrial computer-aided manufacturing (CAM) simulation. SheetGen is evaluated across three test cases. Compared to a baseline algorithm without rule implementation, it raises manufacturability from 40% to 64%. For an input where a generative approach from the literature produces no manufacturable parts, SheetGen achieves 86%. In a plausibility comparison with human designers who achieved 71% manufacturability, SheetGen reaches 96%. The results demonstrate that systematic knowledge integration can substantially improve the manufacturability of generatively designed sheet metal parts, and the identified strengths and limitations of the rule-based approach point toward hybrid strategies that combine rule-based reliability with artificial intelligence (AI)-driven adaptability to further advance generative design for manufacturing. [DOI: 10.1115/1.4072234]

Keywords: design automation, design for manufacturing, expert systems, generative design, product development

1 Introduction

Generative design is an increasingly adopted tool for exploring design spaces in engineering, particularly valuable in early design phases where broad exploration of alternatives can accelerate convergence toward high-quality solutions [1]. Its primary goal is the automatic generation of many design alternatives, enabling designers to navigate complex trade-off landscapes more effectively than manual iteration allows [2]. Current approaches, however, tend to focus on geometric variety, structural performance, and single-objective optimization [3], producing solutions that are often geometrically diverse but not necessarily manufacturable.

Generated solutions must not only be functional but also manufacturable. Yet manufacturability is frequently either checked only after generation or neglected entirely, leading to high manual rework effort and designs that cannot be manufactured as-is [4]. The core issue is a disconnect between the generation process and the physical, process-specific requirements that govern real manufacturing [5]. As long as these requirements remain external to generation, the designer bears the burden of reconciling algorithmic output with manufacturing reality. In this work, manufacturability is interpreted in a binary sense: a part is considered manufacturable if it satisfies all process-specific requirements and can be processed through standard bending operations without geometric modifications that alter the original design intent.

Designing manufacturable parts has long relied on implicit knowledge, established design methods, and heuristics, informal

[1]Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received February 28, 2026; final manuscript received June 15, 2026; published online July 13, 2026. Assoc. Editor: Liwei Wang.

yet effective strategies that guide decisions under uncertainty [6,7]. This experiential knowledge is rarely formalized in a way that computational tools can leverage [8], and current generative approaches make little systematic use of it. Bridging this gap of translating experiential design knowledge into algorithmic rules is a prerequisite for generating designs that are both creative and manufacturable [9]. To investigate this integration in a concrete manufacturing context, this work focuses on sheet metal bending, a domain where manufacturability is governed by well-documented yet rarely codified design knowledge.

Sheet metal is widely used, cost-effective, and sustainable, yet designing manufacturable sheet metal parts requires extensive experiential knowledge [10]. Sheet metal bending is a forming process in which a flat, pre-cut metal blank is progressively bent along defined lines to produce a three-dimensional part [11]. The process works with planes rather than volumes: each bend transforms a two-dimensional region into an angled flange, and the final part geometry emerges from the cumulative sequence of these bending operations. Because each bend introduces geometric dependencies, like minimum flange lengths required for tooling contact and angular limits dictated by material properties, the design of sheet metal parts is tightly coupled to its manufacturing process [11,12]. Failure to respect process-specific rules directly produces nonmanufacturable parts, making sheet metal bending an ideal domain to study the integration of domain-specific design knowledge into generative design. Throughout this article, the term domain-specific design knowledge refers collectively to design methods, heuristics, and manufacturing requirements.

Despite the availability of domain-specific design knowledge in the form of design methods, heuristics, and manufacturing requirements [13], existing generative design approaches do not sufficiently integrate manufacturability into the generation process [2]. In particular, no algorithm translates design knowledge into algorithmic rules that are enforced during the generation process of sheet metal parts. This gap leads to the following research question:

> How can the integration of domain-specific design knowledge into the generation process of a generative design algorithm improve the manufacturability of generated sheet metal parts?

This article investigates this question within the sheet metal domain, which is characterized by well-defined and extensively documented design requirements and therefore serves as a suitable validation domain. Specifically, the present work makes two primary contributions:

(1) SheetGen, a rule-based generative design algorithm for sheet metal parts that systematically translates domain-specific design knowledge into algorithmic rules enforced at distinct stages of the generative workflow (preprocessing, in-process, postprocessing), producing manufacturable parts by design rather than by post hoc filtering. While implemented here in a rule-based setting, the underlying principle of stage-specific knowledge integration is applicable to *artificial intelligence* (AI)-driven generative approaches as well.
(2) Demonstration of industrial applicability through integration with a commercial computer-aided design (CAD) environment (ONSHAPE) and validation via industrial computer-aided manufacturing (CAM) bending simulation software (TRUMPF TRUTOPS BOOST), enabling direct transfer of generated solutions into product development workflows.

The remainder of this article is organized as follows. Section 2 reviews related works on generative design, manufacturability, and sheet metal design. Section 3 presents the methodology, including the knowledge integration process and the rule-based algorithm. Section 4 reports the experimental setup, the manufacturability assessment protocol, and the results across three test cases. Section 5 discusses the implications for generative design, perspectives for hybrid approaches, and limitations. Section 6 concludes the paper and outlines directions for future work.

2 Related Works

This section lays out the present work within existing generative design approaches, with a focus on sheet metal design, manufacturability, and knowledge integration. The review proceeds from the characteristics of sheet metal as a design domain, through the broader landscape of generative design and its treatment of manufacturability, to the small number of existing approaches that address generative design for sheet metal parts specifically.

2.1 Sheet Metal Bending Design Knowledge.

Sheet metal parts play an important role in virtually all areas of modern mechanical engineering, as they can be manufactured rapidly, precisely, and cost effectively [11]. Compared to conventional manufacturing processes, sheet metal bending leads to significant reductions in energy consumption, emissions, and material waste [14], and the recyclability, efficient material usage, and structural performance of sheet metal parts position them as a key contributor to sustainability goals [10]. Through bending, flat cut blanks are transformed into three-dimensional elements, a fundamental 2D-to-3D transformation that works with planes rather than volumes [15].

Designing for this process requires substantial domain-specific design knowledge, encompassing design methods, heuristics, and manufacturing requirements [6,13,16]. Design methods constitute a significant part of the knowledge that engineers are taught and apply in professional practice and should be understood as a distinct type of design knowledge with explicit procedural character [17]. More specifically, heuristics are informal strategies that enable effective decisions under uncertainty, particularly during ideation and concept development, where formal analysis alone is insufficient. These heuristics guide attention toward promising regions of the design space and help filter out unproductive alternatives early [18]. Manufacturing requirements, in turn, summarize the rules that must be followed to enable manufacturability of a given design. In sheet metal design, these requirements are particularly influential, as both cutting and forming processes impose tight geometric constraints on feasible part designs. Design-for-manufacturing (DfM) principles, as part of the broader DfX framework, provide a structured means of incorporating such constraints into early-stage design decisions to ensure compatibility with available manufacturing processes [13,16].

Manufacturability is broadly defined as the ease or difficulty of manufacturing a design with given processes, often quantified via geometric accessibility, setup complexity, and cycle time [19]. In the context of this work, we adopt a narrower, binary interpretation: a part is considered manufacturable if it satisfies all process-specific requirements and can be successfully processed through standard bending operations without requiring geometric modifications that alter the original design intent. Available methods range from checklists and design guidelines to mixed product/process-based methodologies and decision support tools, applicable even before CAD modeling begins [15].

The domain-specific design knowledge integrated into the algorithm comprises two categories: design methods and heuristics that support the conceptual phase, and manufacturing requirements that enforce process-specific requirements.

2.1.1 Design Methods and Heuristics. *Surface Separation.* Surfaces with multiple mounting positions can be split into separate elements, increasing the number of connectable edges and thereby expanding the topological design space [13].

Systematic Connection of Edges. Alternative ways of connecting edges between adjacent tabs are explored to generate structurally distinct solutions. This method forms the structural backbone of segment generation.

Minimizing the Number of Bends. Each additional bend introduces an extra manufacturing step and increases the risk of cumulative dimensional inaccuracies [20]. Segments are therefore limited to at most two bends.

90-deg Bends Where Possible. Right-angle bends are easiest to manufacture and are prioritized before the algorithm falls back to alternative angular configurations.

Direct Force Transmission. Loads should be transmitted along direct paths to avoid stress concentrations [21]. Segments are aligned along the straight line joining the centroids of the edges to be connected.

Segmentation of the Design Problem. The overall problem is decomposed into smaller subproblems, namely, topology generation, segment generation, and assembly, whose subsolutions are later recombined.

2.1.2 Manufacturing Requirements. *Minimum Flange Length.* Insufficient material support on the die within the bending zone leads to distortion that can negatively affect the part's function. A bending zone satisfying the prescribed minimum is generated around each bend line. [12]

Minimum Hole Distance. Holes placed too close to a bending zone are subject to forming distortion. Edges are displaced outward in preprocessing when spacing falls below the required minimum.

Minimum Bend Angle. Bending beyond the material-specific angular limit increases the risk of cracking. Segments violating this limit are discarded in postprocessing.

2.2 Generative Design and Consideration of Manufacturability. Generative design encompasses computational approaches that automatically generate, explore, and refine design alternatives based on rules, constraints, and objectives [2]. It is characterized by systematic exploration of the design space and the presentation of multiple candidate solutions, following an iterative workflow of definition, generation, evaluation, and refinement [22]. Three core components structure most approaches: a design framework that defines the problem, a generation mechanism that produces candidates, and an evaluation and selection procedure that filters and ranks them [23].

The generation mechanisms employed in the literature span a broad spectrum [24]. Classical approaches include topology optimization and level-set methods [25,26], grammar- and rule-based systems [27–29], parametric CAD-based generation [30,31], and bio-inspired growth strategies [32]. AI-enhanced approaches include deep generative models and generative adversarial networks (GANs) [2,33–35], surrogate models predicting manufacturability alongside mechanical properties [33,36], and combinations of evolutionary, reinforcement learning, and surrogate-based methods [37]. A particularly active subfield concerns the automatic generation of parametric CAD models from diverse input modalities such as point clouds, voxel representations, and images [38,39], increasingly leveraging large language models and vision-language models for multimodal CAD generation pipelines [40–43]. This diversity demonstrates the maturity of generative design as a field. However, the choice of generation mechanism also determines how readily domain-specific design knowledge can be integrated: rule- and grammar-based systems offer natural insertion points for explicit constraints, whereas data-driven methods require design knowledge to be embedded implicitly through training data or reward signals.

Despite this methodological breadth, the dominant objectives in existing work remain structural performance, weight reduction, and stiffness [25,26,44]. Generative design is largely treated as an extension of topology optimization for structural problems [26], with typical case studies drawn from automotive and aerospace parts [25,44]. AI-enhanced work is similarly focused on structural or aerodynamic performance rather than manufacturing [37], and the default manufacturing context is frequently additive manufacturing (AM) or CNC machining [30,45]. This prevailing focus means that domain-specific design knowledge for conventional manufacturing, particularly sheet metal bending, remains largely unaddressed, leaving a significant portion of industrially relevant design problems without adequate generative support.

Where manufacturability is considered, it takes one of three forms. In the *downstream* treatment, the manufacturing process is specified as input, but manufacturability is handled in postprocessing through smoothing, CAM validation, or manual adjustment [25,26,44,45]. In the *filter-based* treatment, parametric variants are generated broadly and then pruned through manufacturability checks, cost evaluation, or geometric viability criteria [29,31,46]. In the *absence of* treatment, manufacturability is neither an explicit objective nor a constraint, and generated designs require manual modification before manufacturing [37,47]. Across all three modes, manufacturability remains external to the generative core. This reactive positioning limits the quality of generated designs and shifts the burden of ensuring manufacturability onto the designer, undermining the efficiency gains that generative design promises.

Domain-specific design knowledge integration has been demonstrated for other manufacturing processes. Patterson and Allison proposed a general framework for converting design knowledge into mathematical constraints for optimization [48]. Injection molding rules have been encoded into conditional GANs [35], and die casting constraints have been integrated during generation via 2D depth-image representations [47]. AM-specific overhang constraints have been embedded in bio-inspired growth rules [32], and ontology-based approaches have linked design features with manufacturing capabilities for real-time feedback [49]. These examples demonstrate that knowledge integration is feasible, but existing implementations are largely limited to additive manufacturing and injection molding. A rule-based generative design algorithm that systematically integrates domain-specific design knowledge into the generation process, particularly for sheet metal bending, has not been proposed.

Several observations consolidate that manufacturability remains predominantly downstream, filter based, or absent from generation [37,50]. Traditional DfM rules are often overly restrictive and require reformulation for generative design contexts [48]. Manufacturability-focused generative design has been explicitly identified as a key open research opportunity [37], and rule-based approaches appear particularly suited for domains with strong manufacturability requirements [28]. The following section examines the small number of existing approaches that address sheet metal specifically.

2.3 Generative Design for Sheet Metal Parts. Generative design for sheet metal parts has very few established computational approaches. Existing CAD tools primarily support later-stage detailing and operation sequencing but offer little assistance during conceptual design, where topological exploration is most valuable. The following two approaches show the current state of generative design for sheet metal parts:

Campbell et al. proposed a shape grammar approach in which parts are modeled as assemblies of rectangular patches connected through manufacturing operations. Starting from a rectangular blank as seed geometry, a predefined grammar of up to 108 operation rules, like slitting, notching, shearing, bending, and punching, is applied iteratively to transform the blank and generate new design states. Candidate designs are represented as graphs where nodes correspond to rectangular patches and edges encode topological and manufacturing relationships, enabling both parametric and topological variation through stochastic search. The approach yields not only geometry but also an explicit manufacturing operation sequence for each design. However, the restriction to rectangular patches and orthogonal bending severely limits geometric expressiveness. Generated results are often rudimentary, with several exhibiting extreme bending angles or self-intersections in the folded state. Critically, manufacturability emerges incidentally from the grammar's geometric restrictions rather than from explicit encoding of design knowledge [29,51,52].

Barda et al. addressed the synthesis of complete sheet metal parts for aerospace fixture applications. The design problem is decomposed into segment discovery and segment composition. In the discovery phase, thin "feeler" segments are stochastically sampled to explore potential load paths between attachment and support regions, followed by continuous geometry optimization using a physics-inspired spring energy formulation. In the composition phase, an optimal subset of segments is assembled into a complete design, balancing structural performance, material efficiency, and basic manufacturability criteria through physical simulation. Results are geometrically free and nonorthogonal, reportedly achieving better strength-to-weight ratios than expert-designed solutions. However, manufacturability is only superficially enforced through high-level constraints on sheet continuity, bend angles, and minimum feature dimensions. Domain-specific design knowledge is largely unaddressed, meaning many results would be difficult or impossible to manufacture in practice [53].

Both approaches demonstrate the feasibility of computational design synthesis for sheet metal but fall short in the systematic integration of design knowledge. Patel and Campbell achieve implicit manufacturability through restrictive geometric primitives at the cost of design diversity [29], while Barda et al. prioritize structural performance and geometric freedom but cannot ensure manufacturability [53]. Neither explicitly translates established design knowledge into algorithmic rules within the generation process nor distinguishes at which stage of the generative workflow different types of design knowledge should be applied.

3 Methodology: SheetGen—A Rule-Based Algorithm for Generative Sheet Metal Design

The preceding analysis identified a key shortcoming in existing generative design approaches for sheet metal: manufacturability is not sufficiently integrated into the generation process, and no algorithm exists that translates design knowledge into algorithmic rules enforced during generation. To address this gap, this section introduces SheetGen, a rule-based, manufacturability-focused generative design algorithm that translates established design knowledge into algorithmic rules and embeds them at appropriate stages of the generation workflow. Given user-defined surfaces and mounting points, the algorithm systematically explores the design space by generating multiple manufacturable design variants for a single part that connect the defined surfaces while adhering to the integrated design knowledge.

A rule-based architecture is chosen over data-driven methods because sheet metal design knowledge exists predominantly as explicit, codifiable methods, heuristics, and manufacturing requirements rather than as large-scale training data. Additionally, a rule-based system offers full transparency over which design principles govern each generated solution. While the manufacturability metrics introduced in Sec. 2.1 could, in principle, be formulated as constraints within an optimization framework, such an approach typically assumes a fixed and well-parameterized design space and aims to identify a limited set of optimal solutions. In contrast, the objective of the present work is to enable a structured yet flexible exploration of the design space and to generate a diverse set of manufacturable design alternatives that can support early-stage design decisions. In this context, the topology and feature structure of sheet metal parts emerge progressively during the generation process, and many manufacturability criteria, such as minimum flange length or bend accessibility, act as local feasibility conditions that guide this construction. Rather than restricting the design space through global constraints, these criteria are embedded directly into the generation logic, enabling efficient exploration while ensuring that generated solutions remain largely manufacturable.

The overall workflow is illustrated in Fig. 1 with a simple example of two noncoplanar rectangular surfaces with three mount points, resembling an angle bracket. The workflow proceeds from user input through preprocessing, topology generation, segment generation, part assembly, visualization, and CAD export. The outcome of each workflow pass is a design knowledge-compliant, editable CAD part ready for further detailing or integration into larger assemblies. The resulting solutions are oriented toward early design phases, already satisfying prescribed design principles while remaining open to further detailing by the designer. Section 3.1 describes how design knowledge is formalized into algorithmic rules, and Sec. 3.2 details the algorithm workflow itself.

3.1 Design Knowledge Formalization.

The generative workflow provides several distinct points at which design knowledge can enter the process. Integration can be carried out (i) in preprocessing, by adapting the user-provided input, (ii) in-process integration during topology and segment generation, via rule-based design of geometry that follows established design knowledge, or (iii) in postprocessing, through constraint-aware filtering of the generated candidates. The appropriate integration point depends on the nature of the knowledge being encoded.

Hard manufacturing requirements, such as minimum hole distance, are well suited for implementation as postprocessing filters but can also be integrated earlier by adjusting the input geometry or generating geometry that automatically satisfies the constraint, thereby preserving a broader design space. Design methods and heuristics, in contrast, are better suited to modifying the user input, guiding the generation workflow, or providing the structural base for the algorithmic architecture itself. While this distinction is developed here for a rule-based system, the same logic can, in principle, be exploited by generative AI techniques. For instance, hard requirements may serve as reward boundaries, while design methods may inform training data structure or generation architectures. The procedure for constructing the algorithm, consisting of translating design knowledge into a processable rule base and then building the generation algorithm around these rules, is illustrated in Fig. 1.

The domain-specific design knowledge mentioned in Sec. 2.1 is deployed at three points, corresponding to the integration points introduced above.

Preprocessing (Adaptation of the User Input Before Generation):

- *Minimum hole distance*: Distances between mounts and surrounding edges are examined; edges are displaced when spacing falls below the minimum.
- *Surface separation*: Surfaces exceeding a user-defined mount threshold are subdivided into independent subsurfaces.

In-Process Integration (Rule-Based Generation of Geometry):

- *Systematic connection of edges:* All valid edge pairings are enumerated rather than a single configuration selected.
- *Ensure minimum flange length:* A bending zone providing sufficient tool support is generated around each bend line and merged with the tab perimeter.
- *Prefer 90-deg bends:* The algorithm first seeks right-angle bends before resorting to alternative strategies.
- *Direct force transmission:* Segments are aligned along the centroid-to-centroid line of the connected edges.
- *Minimize bends/avoid weld seams:* Only segments with zero, one, or two bends are produced; exclusively single-part segments not requiring welding are retained.

Postprocessing (Filtering Stage):

- *Minimum bend angle:* Segments violating the prescribed minimum are discarded.

A layered combination of these mechanisms—preprocessing adaptation, in-process enforcement, and postprocessing filtering—is required to address the widest possible spectrum of DfM considerations within a single generative workflow.

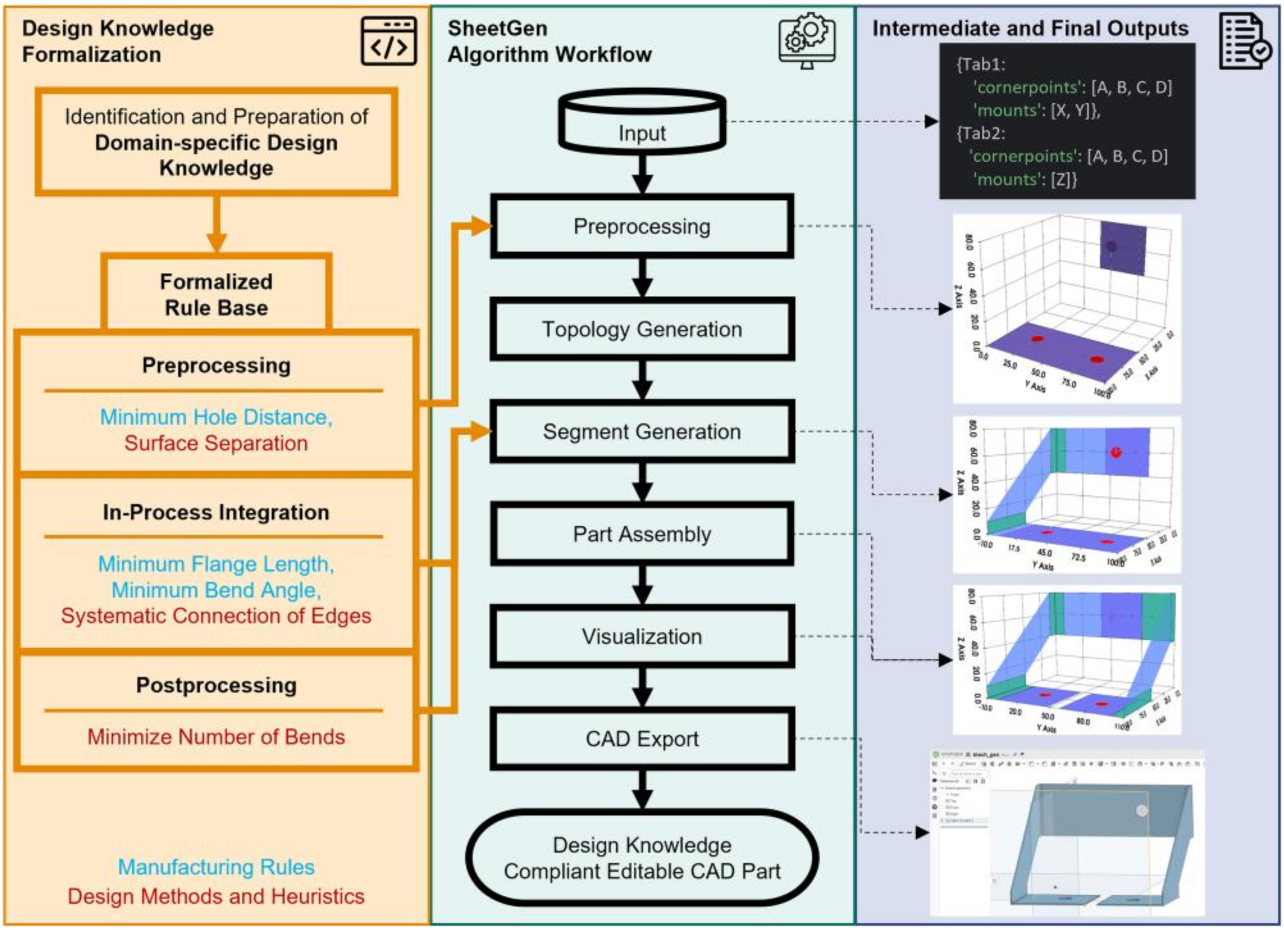

Fig. 1 Overview of SheetGen. Left: Formalization of domain-specific design knowledge into a rule base, distinguishing manufacturing rules, design methods, and heuristics. Center: Generative design algorithm workflow with rules integrated at different workflow stages. Right: Progressive outputs at different workflow stages, from user-defined input surfaces through generated segments to the final editable sheet metal part exported to CAD.

3.2 Algorithm Workflow. Building on the rule base established in Sec. 3.1, the following subsections describe the algorithm workflow and how these rules are realized within each algorithmic stage of SheetGen. Three cross-cutting design principles govern the algorithm throughout. First, the algorithm follows a manufacturability-first philosophy: explicit constraints such as minimum flange lengths are mainly enforced during generation rather than verified after the stage. Second, diversity maximization is pursued by enumerating all topologies and segment sequences, and by expanding the design space using the surface separation design method. Third, modular strategy selection automatically assigns the appropriate connection geometry, zero-bend, single-bend, or double-bend, based on the relative orientation of two tabs, with automatic fallback to the next viable strategy if constraints cannot be satisfied.

3.2.1 Preprocessing. The preprocessing stage transforms raw user input into the data structures required by generation. For each rectangular surface, the user supplies three corner points. The algorithm identifies the right-angled corner via dot-product testing, computes the fourth point analytically, and stores the result as an ordered perimeter.

Mount positions are projected from global 3D coordinates onto each rectangle's local 2D frame. For every mount, the perpendicular distance to each edge is measured; if it falls below the minimum hole-to-edge clearance, the corresponding edge is shifted outward while preserving right-angle geometry. Mount locations remain fixed. The processed information is encapsulated in three hierarchical objects—Part, Tab, and Mount—that store geometry, assembly sequences, and mounting specifications, respectively, and serve as the common data interface for all subsequent stages.

3.2.2 Topology Generation. The topology generation stage determines which pairs of tabs will be linked by segments. It comprises surface separation, topology design, and sibling validation.

When a tab hosts more mounts than a user-defined threshold, it is subdivided into two tabs. A gap between subsurfaces preserves tooling access, and siblings retain their common original identifier. The connectivity graph is constructed by enumerating all spanning trees via recursive backtracking with union-find cycle detection. A sibling-connection validation ensures that split surfaces connect to each other only if each sibling also possesses at least one external connection. Sequences violating this rule are discarded.

3.2.3 Segment Generation. For each tab pair, the algorithm classifies the geometric relationship between the two planes and applies the appropriate connection strategy.

Zero-bend strategy (coplanar tabs): A rectangular intermediate tab directly joins edges from each tab. No bending is required; a gap offset avoids surface interference.

Single-bend strategy (plane intersection): The intersection line of the two tab planes defines the bending line. For each eligible edge pair, bending points are computed on this line, and flange zones of prescribed minimum flange length are generated perpendicular to each plane. A cascade of geometric filters enforces non-parallel intersection, minimum bend angle, minimum flange width, clearance from the opposite plane, corner containment within the

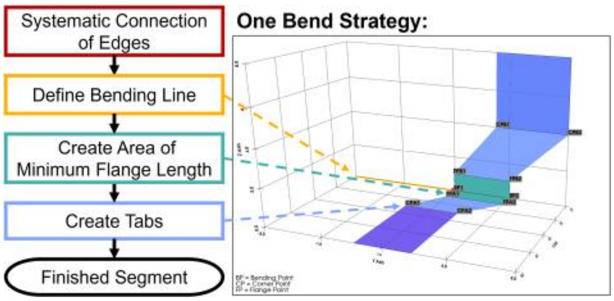

Fig. 2 Visualization of the segment generation process. The edges to be connected are first specified via the corner points *CPAx* and *CPBx*. *A* and *B* denote the respective initial tabs connected by the bend. Based on these, the bending line is determined through the corresponding bending points *BPx*. Subsequently, a flange region with minimal flange length is constructed using the flange points *FPAx* and *FPBx*. Finally, the original edges are connected to the generated flange region, completing the segment.

tab perimeter, and duplicate elimination. Figure 2 illustrates the process.

Double-bend strategy (intermediate plane): An intermediate plane perpendicular to both tab planes is constructed; bending points are shifted outward to accommodate flange requirements, and a triangular intermediate tab bridges the two surfaces. Filters enforce near-perpendicularity, minimum flange widths on both bends, minimum bend angles, and manifold perimeter integrity. If the perpendicular configuration fails, a corner-projection fallback defines the intermediate plane through the nearest corners of the tabs.

The preference ordering—zero bends before one, one before two—directly implements the bend-minimization heuristic, while 90-deg prioritization within the double-bend strategy reflects the right-angle preference.

3.2.4 Part Assembly. The assembly stage combines individually generated segments into complete parts. Tabs referenced by a single segment are used directly. Tabs referenced by multiple segments undergo a multimerge procedure: perimeters are walked simultaneously, shared corners are consumed together, and unique flange sequences are inserted from the contributing tab. Conflicting noncorner modifications cause the merge to be rejected. Assembled parts undergo self-intersection detection and perimeter-ordering verification.

3.2.5 Visualization. Interactive visualization is realized with PYVISTA. An interactive mode allows navigation through the solution set via arrow keys with trackball-style camera control. Hierarchical labels annotate corners, bending points, flange points, and mounts. On-screen elements display the current solution index, a legend, and export controls. After choosing a preferred solution, the user can click a checkbox to export this solution.

3.2.6 Computer-Aided Design Export. Export into ONSHAPE is performed via FEATURESCRIPT. For each tab, a local plane basis is derived, and all 3D points are projected to obtain local planar coordinates. The resulting polygon is sketched, extruded to a default thickness of 1 mm, and the mount holes are subtracted via Boolean operations. After import, the designer transforms the solid geometry into an editable sheet metal model through a sequence of manual operations. These steps are detailed as follows. Initially, the individual extrusions of the tabs are consolidated into a single part using the *Replace Face*, *Move Face*, and *Boolean Union* tools within ONSHAPE. Subsequently, the consolidated geometry is converted into a sheet metal representation using the *Sheet Metal Model* function. The resulting model contains recognized bends and a flat pattern suitable for CAM

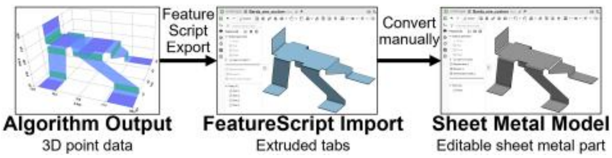

Fig. 3 CAD export pipeline: algorithm output visualized in PYVISTA (left), FEATURESCRIPT import into ONSHAPE (middle), and editable sheet metal model after manual conversion step (left)

simulation. Figure 3 illustrates the complete export pipeline from algorithm output to the final editable sheet metal part.

3.2.7 Computational Cost. The computational cost of the proposed generative method is dominated by the combinatorial growth of candidate assemblies. For n input tabs, the number of feasible configurations increases exponentially, as multiple geometric realizations are evaluated for each topology. Here, s represents the average number of segments generated per adjacent tab pair. While early manufacturability filtering, and a bounded number of retained topologies reduce the effective search space, the overall time complexity remains exponential, with an additional polynomial validation cost, which together can be expressed as $O(s^n \cdot n^2)$.

Memory consumption scales with the number of valid solutions, while intermediate computations are kept tractable through incremental evaluation. For the problem sizes considered in this work ($n \leq 7$), runtimes remain within seconds to minutes on a standard workstation.

4 Evaluation of SheetGen

This section presents three test cases that evaluate SheetGen with respect to the effect of rule integration on manufacturability, comparative performance against an existing generative approach, and plausibility relative to trained human designers. The algorithm is fully deterministic: identical input geometry and parameters produce identical solutions on every run. The reported solution counts and manufacturability rates are therefore exact values, not statistical estimates.

4.1 Experimental Setup. To evaluate SheetGen, three test cases examine the algorithm from complementary perspectives: the effect of rule integration on manufacturability, comparison with an existing generative approach, and plausibility relative to human designers. The input geometries for all three test cases are shown in Fig. 4. This section first describes the manufacturability assessment protocol and then presents the individual test case configurations.

4.1.1 Manufacturability Assessment Protocol. Each generated part was assessed through a two-stage procedure combining CAD reconstruction with CAM bending simulation. In the first stage, parts were reconstructed in ONSHAPE's sheet metal environment. ONSHAPE automatically inserts stress relief cutouts when the minimum flange length is violated; parts receiving such cutouts were classified as nonmanufacturable since these modifications

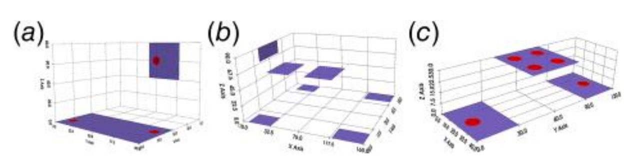

Fig. 4 Input for the described test cases. The rectangles are the defined tabs to be connected by the algorithm, and the circles define the mount positions. (*a*) input case 1, (*b*) input case 2, and (*c*) input case 3

alter the original design intent. Parts passing CAD reconstruction without modification were imported into TRUMPF TRUTOPS BOOST, a commercial CAM software for bending simulation, and classified as follows:

- *No errors or warnings:* The part is classified as manufacturable.
- *Warnings only:* The part is inspected on a case-by-case basis. Minor bending-sequence instability warnings were considered acceptable if visual inspection confirmed a feasible bending sequence.
- *Errors—tool collisions only:* If all errors were attributable solely to tool collisions resolvable through alternative tooling configurations, the part was still classified as manufacturable, as the geometric design itself does not prevent manufacturing. Nonresolvable tool collisions led to a nonmanufacturable classification.
- *Errors—other causes:* All remaining errors led to a nonmanufacturable classification.

This protocol distinguishes between geometric manufacturability (whether the part's form is compatible with bending) and process-level feasibility issues, such as tooling selection that depends on the available equipment.

After assessing each generated part for manufacturability, the CAM manufacturability rate for the entire test case was computed as the ratio of parts without CAM errors to the total number of generated parts. This metric serves as an indicator of the quality of the generation algorithm as it reflects the proportion of designs that can be processed without manufacturing-related issues.

4.1.2 Test Cases

4.1.2.1 Test Case 1: Verification of Manufacturability Improvement Through Rules. SheetGen is compared to a stripped version in which minimum hole distance, minimum flange length, and minimum bend angle are set to zero, and surface separation is disabled. Systematic connection of edges cannot be disabled as it forms the structural foundation of segment generation, but it has no direct influence on manufacturability. The input consists of two noncoplanar rectangular surfaces with three mount points, serving as a basic validation of mount handling and constraint propagation.

4.1.2.2 Test Case 2: Comparison With the State of Research. Results from Barda's approach [53], which does not integrate manufacturability into generation, were reproduced for the same input geometry. To maintain topological comparability, only the connection sequence was controlled via a custom sequence function; segment generation operated entirely according to the rule-based algorithm. The input comprises seven rectangular surfaces connected through six predefined connection pairs.

4.1.2.3 Test Case 3: Plausibility Comparison With Human Designers. Generated results are compared to solutions produced by engineering students who received a 2-day workshop covering the same domain-specific design knowledge encoded in the algorithm. This comparison serves as a plausibility demonstration, not a controlled experiment. The 14 mechanical and mechatronics engineering students were tasked with designing a cylinder mounting bracket, given the dimensions and hole pattern. They were CAD novices, requiring approximately 6–8 h per solution. The algorithm input for the cylinder mounting bracket comprises three surfaces with six mount points: two outer rectangles with single mounts connected to a central surface with a 2×2 mount grid, requiring the algorithm to identify that the central surface should be split.

4.2 Results Overview.

Each test case compares the manufacturability of parts generated by SheetGen against a different reference: an unconstrained baseline without rule implementation (test case 1), an existing generative design approach from the literature (test case 2), and solutions produced by trained human designers (test case 3). Table 1 provides a summary of the key results across all three test cases. The individual test cases are discussed in detail in the following subsections.

4.3 Verification of Manufacturability Improvement Through Rules.

The first test case compares SheetGen against the stripped version in which all design knowledge-based rules are disabled. Without rule implementation, four of ten generated parts (40%) pass the manufacturability assessment. With rules enabled, the manufacturability rate increases to 64% (18 of 28 parts). Two partially independent effects contribute to this improvement.

First, surface separation expands the design space: the total number of generated solutions increases from 10 to 28. This expansion is a direct consequence of the additional connectable edges introduced by subdivision, which enable topological configurations, such as branching rather than purely chain-like arrangements, which are inaccessible without it. Second, the remaining rules (minimum flange length, minimum bend angle, minimum hole distance) increase the fraction of candidates that satisfy the manufacturability assessment. The compounding of these two effects produces a 4.5-fold increase in the number of valid solutions, from 4 to 18. A representative example of the effect of the rule implementation can be observed by comparing solution d) in the baseline with solution b) in SheetGen in Fig. 5. While both designs share an identical topology, the rule-based adaptation modifies critical geometric features. In particular, the bend edge is shifted further away from the hole, and the adjacent edges extend vertically from the bending line before transitioning into an angled segment. This adjustment ensures compliance with the minimum hole distance and minimum flange length constraints, and the resulting part is therefore considered manufacturable.

Analysis of the CAM error reports confirms that all rules encoded in the algorithm are consistently satisfied across every generated solution. The remaining nonmanufacturable parts fail exclusively due to manufacturing requirements not included in the current rule base, most notably tool collisions during the bending sequence. This illustrates a fundamental characteristic of rule-based approaches: the algorithm can only enforce constraints that have been explicitly encoded, and achieving comprehensive manufacturability would require a complete rule base, which is resource intensive to develop and maintain. The comparison is summarized in Fig. 5.

Table 1 Summary of results across three test cases

	Test case 1: rule verification	Test case 2: state of research	Test case 3: human designers
Reference	Baseline (no rules)	Barda et al. [53]	Engineering students
Manufacturable/total (reference)	4/10 (40%)	0/2 (0%)	10/14 (71%)
Manufacturable/total (SheetGen)	**18/28 (64%)**	**48/56 (86%)**	**24/25 (96%)**
Generation time (reference)	<1 s per part	~30 min per part	6–8 h per part
Generation time (SheetGen)	**<1 s per part**	**<1 s per part**	**<1 s per part**

Note: Each test case compares the manufacturability rate and generation time of SheetGen against a different reference.

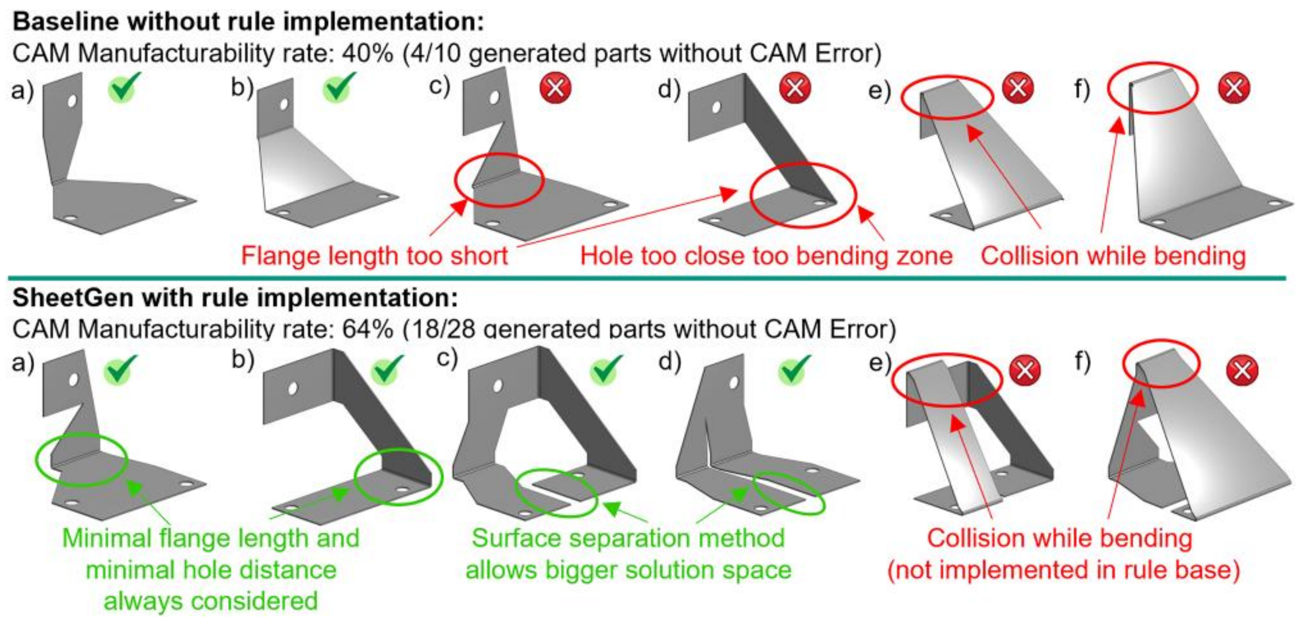

Fig. 5 Comparison of algorithm-generated solutions without (top) and with (bottom) design knowledge-based rules enabled. Design details that are particularly important in terms of manufacturability are marked with circles. Disabling all rules yields a manufacturability rate of 40% (4/10), while enabling them increases it to 64% (18/28), with remaining failures attributable to manufacturing requirements not yet included in the rule base.

4.4 Comparison With the State of Research. Barda et al. reported two parts for the given input configuration [53], neither of which passed the manufacturability assessment when evaluated using the same two-stage protocol (ONSHAPE reconstruction and TruTops bending simulation) applied to SheetGen's results. As similarly observed for the baseline algorithm in Sec. 4.3, the components generated by Barda et al. frequently fail to meet the minimum flange length requirement at the bend edges. This leads to insufficient material support during bending, which can cause warping of the component and may impair its functional performance. These designs were therefore classified as unmanufacturable. In contrast, SheetGen enforces the minimum flange length through rule-based generation, ensuring that this constraint is satisfied by design. As a result, SheetGen achieved a manufacturability rate of 86% across 56 generated designs. The remaining unmanufacturable components generated by SheetGen exhibit the same type of collision issues during the bending process that were already discussed in Sec. 4.3, indicating that they stem from manufacturing constraints not yet included in the current rule base rather than violations of the implemented rules. The higher rate compared to test case 1 (64%) is attributable to the specific geometry of this test case, which produces fewer tool collisions during the bending sequence. Figure 6 illustrates the comparison: Barda's parts fall short of the minimum flange length in the areas highlighted with circles, whereas all the parts generated by SheetGen comply with this constraint.

This comparison is not strictly identical in all conditions. The connection sequence was prescribed via a custom sequence function to obtain comparable topologies, and one input surface was subdivided to introduce additional edges, as SheetGen permits only one segment per edge. These adjustments establish a consistent basis for evaluation, but the results should be interpreted as indicative of manufacturability benefits rather than as a direct replication study.

Regarding computational cost, SheetGen generates individual parts in under 1 s, whereas approximately 30 min per part are reported for the approach by Barda et al. This difference can be traced to the fundamentally different algorithmic structures. As outlined in Sec. 3.2.7, the proposed method follows a bounded enumeration strategy with time complexity $O(s^n \cdot n^2)$, where the exponential term reflects the combinatorial growth of candidate assemblies and the polynomial term arises from geometric validation.

In contrast, the method of Barda et al. does not provide a closed-form complexity, but its structure implies a different scaling behavior. The search space of candidate assemblies is exponential in the number of components and is explored through stochastic Markov chain Monte Carlo sampling combined with beam search. The computational cost can therefore be approximated as proportional to the number of sampling steps and beams, with $O(M \cdot N \cdot C_{\text{sim}})$, where M denotes the number of parallel chains, N the number of iterations, and C_{sim} the cost of evaluating a single candidate. This evaluation includes repeated physics-based simulation and optimization steps, which dominate the overall runtime.

The rule-based formulation avoids such simulation-driven inner loops by discarding infeasible configurations early through deterministic filtering. As a result, the computational effort is concentrated in structured enumeration rather than in evaluation-heavy sampling. While additional constraints, such as finite element analysis or no-design-zone handling, would increase the computational cost of the proposed method, both approaches remain fundamentally limited by the exponential growth of the design space. The observed results, therefore, suggest that a substantial runtime advantage of the proposed method is likely to persist even when extended evaluation steps are incorporated.

4.5 Plausibility Comparison With Human Designers. As a plausibility assessment, SheetGen's results for the cylinder mounting bracket configuration are compared to 14 solutions produced by engineering students who received the same methodological training. SheetGen achieves a manufacturability rate of 96% (24 of 25 solutions), compared to 71% (10 of 14) for the student solutions, as illustrated in Fig. 7. While this difference is notable, the comparison serves primarily as an indication that the algorithm produces results of practical quality, not as a claim of general superiority over human designers.

SheetGen generates all 25 solutions in under seven seconds, whereas students required approximately 6–8 h per solution. This asymmetry is partly attributable to the students' limited CAD

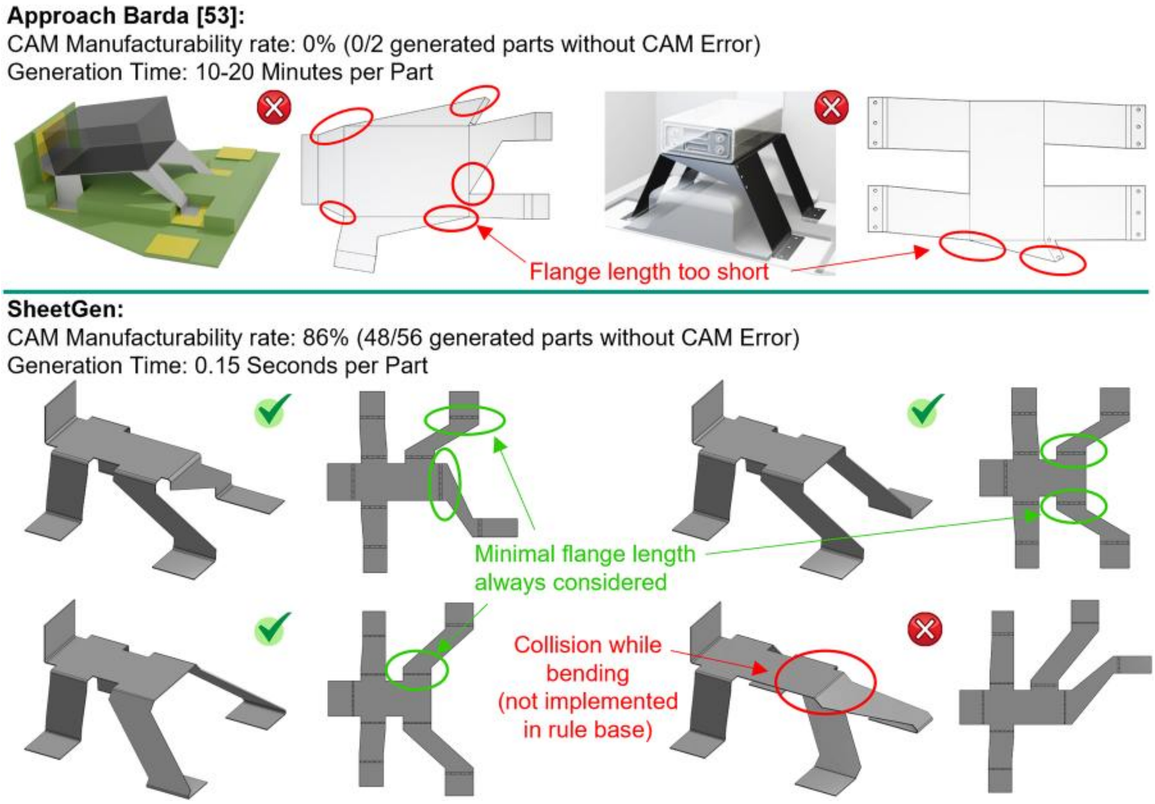

Fig. 6 Comparison of the parts generated by SheetGen with the state of research, design details that are particularly important in terms of manufacturability are marked with circles. The parts without manufacturability considerations (Barda et al. [53]) fall short of the minimum flange length and therefore cannot be manufactured, while the parts generated using our algorithm comply with the manufacturing requirements, and therefore, the manufacturability rate is much higher.

experience, which substantially increased their modeling time, and the comparison therefore reflects the algorithm's speed advantage under these specific conditions rather than a general benchmark.

In terms of design variety, the student solutions exhibit a broader range of conceptual ideas. For instance, using a single tab to connect all three input surfaces, introducing unusual angles, or stacking tabs vertically. SheetGen's variety is bounded by its hard-coded segment generation strategies, which produce systematic but geometrically more constrained alternatives. However, the algorithm also identifies valid configurations that none of the students produced, suggesting that rule-based and human-driven exploration are complementary rather than substitutive. This finding

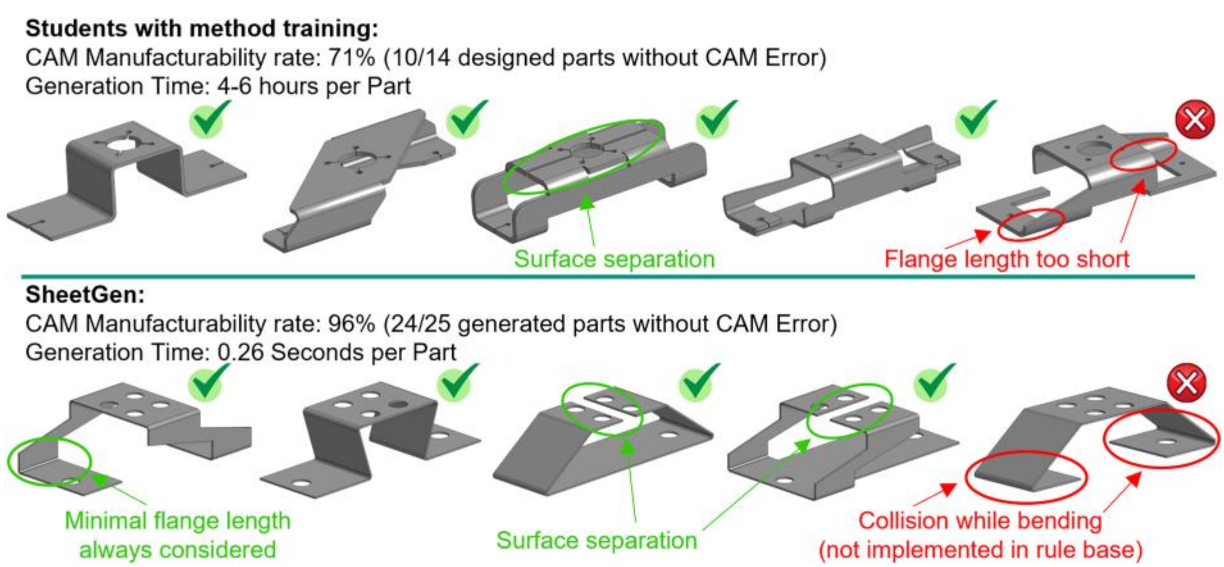

Fig. 7 Comparison of student-designed and algorithm-generated solutions for the cylinder mounting bracket configuration. Design details that are particularly important in terms of manufacturability are marked with circles. SheetGen achieves a manufacturability rate of 96% (24/25) compared to 71% (10/14) for student solutions, while also producing valid configurations not found among the student designs.

highlights an inherent trade-off: systematic enforcement of manufacturing requirements narrows the geometric design space relative to unconstrained creativity, but the resulting solutions are more reliably manufacturable. Hybrid approaches combining both modes of exploration are discussed in Sec. 5.

5 Implications for Generative Design

The evaluation of SheetGen demonstrates that translating domain-specific design knowledge into algorithmic rules and enforcing them at distinct stages of the generative process can substantially improve the manufacturability of generated sheet metal parts, with improvements of 24–86 percentage points across the three test cases. This finding supports a shift away from the prevailing paradigm in generative design, where manufacturability is typically treated as an external verification step, and suggests that knowledge integration during generation is both feasible and effective. The following sections interpret these results within the broader field of generative design for manufacturability, examine their implications for future integration with AI-driven generative approaches, and discuss the limitations of the current work.

5.1 Generative Design for Manufacturability

5.1.1 Manufacturability as an Integral Part of Generation. The results across all three test cases demonstrate that integrating domain-specific design knowledge directly into the generative process rather than applying it as a downstream filter can substantially improve the manufacturability of generated designs. Generated parts satisfy manufacturing restrictions by design, providing engineers with solutions that are immediately usable rather than requiring extensive postprocessing. This stands in contrast to the prevailing treatment of manufacturability in current generative design approaches, where manufacturability is either checked after generation or neglected entirely.

5.1.2 Understanding the Manufacturability Gap. In test case 1, 36% of the generated parts remain nonmanufacturable despite rule integration. Analysis of the CAM error reports reveals that these failures are predominantly caused by tool collisions and self-collisions during the bending sequence, failure modes that are not included in the current rule base. Because these collisions depend on the specific geometric configuration resulting from the user input rather than on violations of encoded rules, their occurrence is input dependent rather than systematic. This pattern confirms two things simultaneously: the implemented rules are consistently effective, as all encoded constraints are satisfied across every generated solution; but the rule base is incomplete with respect to the full spectrum of manufacturing requirements.

Among the implemented rules, the minimum flange length constraint contributed most substantially to the manufacturability improvement, as insufficient flange support was the dominant failure mode in the unconstrained baseline. Surface separation, while not a manufacturability constraint per se, was the primary driver of design-space expansion, enabling topological configurations that would otherwise be inaccessible. No failures attributable to unexpected interactions between rules were observed; each rule contributes along a largely independent axis, which supports the modular architecture of the knowledge integration.

5.1.3 Integration Point Selection. An observation from the implementation process is that several rules currently deployed as postprocessing filters could, in principle, also be enforced during geometry generation. However, in-process enforcement of these constraints would require substantial geometric adaptations that risk generating malformed parts or designs that no longer conform to the user-specified input geometry. Retaining them as filters represents a pragmatic compromise: it preserves the integrity of the generated geometry while ensuring that only manufacturable candidates reach the designer. This observation illustrates a practical dimension of the preintegration/during integration/postintegration: the choice of integration point is not only a matter of knowledge type but also of the geometric consequences that enforcement entails.

5.1.4 Usability for Engineering Practice. The direct export into ONSHAPE enables engineers to receive solutions ready for further refinement without rebuilding them from scratch. The current pipeline is functional and demonstrates the feasibility of bridging algorithmic generation and CAD-based product development, though it still requires manual steps to arrive at a finished sheet metal part. The white-box nature of the algorithm supports transparency and trust: every design decision can be traced back to specific methods, heuristics, and manufacturability requirements, allowing engineers to inspect and justify the generation logic.

5.1.5 Advancing Generative Design Beyond Geometry Generation. The results suggest that generative design can evolve from pure geometry generation toward the generation of constructively meaningful, manufacturable solutions. The integration taxonomy (preprocessing, in-process, postprocessing) provides a structured workflow for reasoning about knowledge integration that may be transferable to other manufacturing domains.

5.2 Perspectives for Hybrid Generative Design Approaches.

The strengths and limitations of rule-based approaches identified in this work suggest that rule-based and data-driven generative approaches are not competitors but complementary. Rule-based systems offer reliability and transparency; AI-driven methods offer adaptability and broader exploration. The question is not whether to combine them, but at what depth of integration. Three levels of hybridization can be distinguished, each building on the infrastructure established by the previous one.

5.2.1 Level 1: Rule-Based Generation as Data Infrastructure. The most immediate and lowest-risk path uses SheetGen as a data generator. The fast, deterministic production of manufacturable parts addresses a critical bottleneck in AI-driven sheet metal design: the near-total absence of labeled training data. Unlike purely synthetic datasets, the generated parts incorporate the same design knowledge that human engineers apply, lending the data practical relevance. This path requires no modification to the rule-based algorithm and can proceed in parallel with other developments.

5.2.2 Level 2: Artificial Intelligence–Driven Assembly With Rule-Based Components. A deeper coupling retains rule-based segment generation to ensure that individual connections are manufacturable, while a learned component, such as a reinforcement learning agent or graph neural network, takes over the combinatorial assembly problem. This division of labor directly addresses the design diversity limitation observed in test case 3: the hard-coded assembly strategies constrain topological variety, but combinatorial exploration is precisely where AI methods excel. The rule base acts as a safety net, ensuring that any assembly the AI proposes is composed of individually valid segments and thereby providing a manufacturability floor that pure AI approaches lack.

5.2.3 Level 3: Knowledge Integration Into the Artificial Intelligence Architecture. The deepest coupling embeds the codified design knowledge directly into the AI's learning or optimization process. Concrete mechanisms include design-space restriction, where explicit rules limit exploration to feasible regions; reward-function definition, where design knowledge is translated into reinforcement learning signals that encourage manufacturable outcomes without hard-coding the geometry; and hard constraint incorporation, where the rule set acts as inviolable boundaries within optimization loops. This level offers the greatest potential for generalization; the AI can discover solutions that the rules alone cannot produce, while the rules prevent violation of fundamental manufacturing limits.

These three levels are not mutually exclusive and can be pursued incrementally. The preintegration/during integration/postintegration workflow from this work provides a conceptual map for deciding where AI augmentation is most beneficial: in-process rules that are geometrically complex to enforce, such as collision avoidance during bending, are strong candidates for replacement by learned models, while rules that are simple and critical, like minimum bend angle, may be best retained as hard constraints.

5.3 Limitations. Three principal limitations of the current work should be acknowledged. First, no explicit functional evaluation, such as finite element analysis (FEA), is performed on the generated parts. The focus on geometric manufacturability is deliberate: in sheet metal design, increasing the sheet thickness can reinforce a part without altering its geometry, meaning that geometric feasibility and structural adequacy are more loosely coupled than in other manufacturing domains. Nevertheless, integrating structural evaluation would strengthen the practical relevance of the generated solutions.

Second, rule-based approaches can only solve cases for which solution strategies have been coded. A notable gap in the current implementation is the absence of no-material zones, which are regions where the generated geometry must not intrude. If all generated solutions happen to overlap with such a zone, the algorithm cannot adaptively adjust its generation, as it lacks a mechanism for responding to spatial exclusions that were not anticipated during rule development.

Third, while the proposed approach substantially increases the proportion of manufacturable designs, the current implementation also still produces a limited number of infeasible cases. To ensure reliability prior to manufacturing, a dedicated CAM-based analysis is therefore required as a validation step to identify residual issues, such as tool or self-collisions during bending.

6 Conclusion and Future Work

6.1 Conclusion. This work introduced SheetGen, a rule-based generative design algorithm for manufacturability-focused sheet metal parts based on domain-specific design knowledge, and demonstrated that the systematic integration of design knowledge into the generative process can substantially improve the manufacturability of generated parts. Across three test cases, SheetGen consistently produced solutions with higher manufacturability rates than both an uninformed baseline (algorithm: 64% versus baseline: 40%) and an existing generative approach from the literature (86% versus 0%), while generating dozens of design alternatives in seconds. A plausibility comparison with human designers indicated that SheetGen achieves higher manufacturability rates (96% versus 71%), though with a narrower range of design concepts than unconstrained human exploration. The complementarity of these two modes of exploration, systematic and constrained versus creative and unconstrained, suggests that neither alone is sufficient and that hybrid approaches represent a promising direction.

By connecting SheetGen to a commercial CAD environment and validating generated parts through industrial CAM bending simulation software, this work demonstrates that generative design with domain-specific design knowledge integration can move beyond academic proof of concept toward practical relevance in industrial product development. In practice, the rapid generation of multiple manufacturable parts allows engineers to explore a broad design space in a fraction of the time required by manual approaches. Additionally, the integrated domain-specific design knowledge reduces the level of specialized expertise needed to arrive at manufacturable solutions.

The knowledge integration points that distinguish preprocessing, in-process, and postprocessing deployment are grounded in the nature of the design knowledge itself, specifically the distinction between hard manufacturing requirements and softer design methods and heuristics, rather than in the specifics of any particular algorithm. As such, they may serve as a reference for other rule-based and potentially also data-driven generative approaches seeking to embed domain-specific design knowledge.

The work also highlights an inherent tension in rule-based generative design: the completeness of the rule base directly determines the achievable manufacturability, and every unconsidered requirement remains a potential source of failure. This observation points to the need for complementary approaches, whether through more comprehensive rule bases, hybrid rule-AI systems, or adaptive methods capable of learning constraints from simulation feedback.

6.2 Future Work. The hybrid integration paths outlined in Sec. 5.2 define the primary research trajectory. The most immediate priority is closing the remaining manufacturability gap by addressing tool and self-collisions during bending, either through rule-based expansion or through integration of bending simulation feedback into the generation loop. The latter would also serve as a first testbed for level 2 hybridization, where a learned component operates within rule-based manufacturability boundaries.

Full automation of the ONSHAPE export pipeline would serve a dual purpose: enabling industrial deployment of SheetGen and enabling batch generation of large, labeled datasets of manufacturable sheet metal parts for AI training, corresponding to level 1 hybridization path.

In addition, integrating optimization strategies into the generative process represents a promising extension. While the current approach focuses on the exploration of diverse, manufacturable design alternatives, optimization methods could be employed to guide this exploration, for example, by ranking, filtering, or refining generated candidates with respect to performance objectives such as stiffness, weight, or cost. Such a hybrid approach would combine the strengths of rule-based manufacturability enforcement with objective-driven search.

Beyond single-part generation, extending SheetGen to multipart assemblies introduces design for assembly considerations, in particular, the question of how to optimally partition a design into individual sheet metal parts. This extension would naturally benefit from AI-driven combinatorial methods operating within rule-based constraints.

Use of AI Tools

AI-assisted tools (Claude Opus 4.6) were used to refine grammar, improve readability, and enhance stylistic consistency throughout the manuscript; however, all substantive content, analysis, and intellectual contributions remain entirely those of the authors.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

References

[1] Oh, S., Jung, Y., Kim, S., Lee, I., and Kang, N., 2019, "Deep Generative Design: Integration of Topology Optimization and Generative Models," ASME J. Mech. Des., **141**(11), p. 111405.

[2] Regenwetter, L., Nobari, A. H., and Ahmed, F., 2022, "Deep Generative Models in Engineering Design: A Review," ASME J. Mech. Des., **144**(7), p. 071704.

[3] Nobari, A. H., Chen, W., and Ahmed, F., 2021, "Range-Constrained Generative Adversarial Network: Design Synthesis Under Constraints Using Conditional Generative Adversarial Networks," ASME J. Mech. Des., **144**(2), p. 021708.

[4] Ulu, E., Gecer Ulu, N., Hsiao, W., and Nelaturi, S., 2020, "Manufacturability Oriented Model Correction and Build Direction Optimization for Additive Manufacturing," ASME J. Mech. Des., **142**(6), p. 062001.

[5] Barroqueiro, B., Andrade-Campos, A., Dias-de-Oliveira, J., and Valente, R. A. F., 2021, "Bridging Between Topology Optimization and Additive Manufacturing via Laplacian Smoothing," ASME J. Mech. Des., **143**(9), p. 091703.

[6] Doellken, M., Zimmerer, C., and Matthiesen, S., 2020, "Challenges Faced by Design Engineers When Considering Manufacturing in Design—An Interview Study," Proceedings of the Design Society: DESIGN Conference, Cavtat, Croatia, June 11, Vol. 1, pp. 837–846.

[7] Yilmaz, S., Seifert, C., Daly, S. R., and Gonzalez, R., 2016, "Design Heuristics in Innovative Products," ASME J. Mech. Des., **138**(7), p. 071102.

[8] Fu, K. K., Yang, M. C., and Wood, K. L., 2016, "Design Principles: Literature Review, Analysis, and Future Directions," ASME J. Mech. Des., **138**(10), p. 101103.

[9] Suresh Kumar, R., Srivatsa, S., Baker, E., Silberstein, M., and Selva, D., 2023, "Identifying and Leveraging Promising Design Heuristics for Multi-Objective Combinatorial Design Optimization," ASME J. Mech. Des., **145**(12), p. 121702.

[10] Krause, A., Dielhenn, J., Jess, S., Wagenmann, S., Doellken, M., Albers, A., Matthiesen, S., and Bursac, N., 2025, "Sustainable Sheet-Metal Design: Employing the Product Carbon Footprint as Support for Engineers in Developing New Product Generations," Procedia CIRP, **136**, pp. 510–515.

[11] Jeswiet, J., Micari, F., Hirt, G., Bramley, A., Duflou, J., and Allwood, J., 2005, "Asymmetric Single Point Incremental Forming of Sheet Metal," CIRP Ann., **54**(2), pp. 88–114.

[12] Pahl, G., Beitz, W., and Wallace, K., 1996, *Engineering Design: A Systematic Approach*, Springer, Berlin, London, Heidelberg.

[13] Doellken, M., Arndt, L., Nelius, T., and Matthiesen, S., 2021, "Identifying an Opportunistic Method in Design for Manufacturing: An Experimental Study on Successful a on the Manufacturability and Manufacturing Effort of Design Concepts," Procedia CIRP, **100**, pp. 720–725.

[14] Cooper, D. R., Rossie, K. E., and Gutowski, T. G., 2017, "The Energy Requirements and Environmental Impacts of Sheet Metal Forming: An Analysis of Five Forming Processes," J. Mater. Process. Technol., **244**, pp. 116–135.

[15] Bursac, N., Krause, A., Batora, M., and Ritzer, K., 2023, "Live-Lab GSD—Generational Sheet Metal Development: A Validation Environment for Methodological Design Support in Sheet Metal Development," Procedia CIRP, **119**, pp. 41–46.

[16] Doellken, M., Peters, J., and Matthiesen, S., 2025, "Experimental Investigation of the Influence of Restrictive and Dual Methods for Design for Manufacturing," IMECE, Portland, OR, Nov. 17–21.

[17] Stacey, M., Eckert, C., Pirtle, Z. G., Poznic, M., Schuelke-Leech, B.-A., and von der Tann, L., 2025, "Methods as a Form of Engineering Knowledge," Des. Sci., **11**, p. e12.

[18] Barner, M. S., Brown, S. A., and Linton, D., 2021, "Structural Engineering Heuristics in an Engineering Workplace and Academic Environments," J. Civ. Eng. Educ., **147**(2), p. 04020014.

[19] Shukor, S. A., and Axinte, D. A., 2009, "Manufacturability Analysis System: Issues and Future Trends," Int. J. Prod. Res., **47**(5), pp. 1369–1390.

[20] Moeeni, H., Javadi, M., and Raissi, S., 2022, "Design for Manufacturing (DFM): A Sustainable Approach to Drive the Design Process From Suitability to Low Cost," Int. J. Interact. Des. Manuf., **16**(3), pp. 1079–1088.

[21] Matthiesen, S., and Grauberger, P., 2024, *Konstruktionswissen für Ingenieure: Innovative Produkte Zielgerichtet Entwickeln*, Springer, Berlin, Heidelberg.

[22] Chew, Z. X., Wong, J. Y., Tang, Y. H., Yip, C. C., and Maul, T., 2024, "Generative Design in the Built Environment," Autom. Constr., **166**, p. 105638.

[23] Jaisawal, R., and Agrawal, V., 2021, "Generative Design Method (GDM)—A State of Art," IOP Conf. Ser.: Mater. Sci. Eng., **1104**(1), p. 012036.

[24] Mountstephens, J., and Teo, J., 2020, "Progress and Challenges in Generative Product Design: A Review of Systems," Computers, **9**(4), p. 80.

[25] Barbieri, L., and Muzzupappa, M., 2022, "Performance-Driven Engineering Design Approaches Based on Generative Design and Topology Optimization Tools: A Comparative Study," Appl. Sci., **12**(4), p. 2106.

[26] Ntintakis, I., and Stavroulakis, G. E., 2020, "Progress and Recent Trends in Generative Design," MATEC Web Conf., **318**, p. 01006.

[27] Ertelt, C., and Shea, K., 2008, "Generative Design and CNC Fabrication Using Shape Grammars," IDETC-CIE, Brooklyn, NY, Aug. 3–6, pp. 25–34.

[28] McCormack, J., Dorin, A., and Innocent, T., 2004, "Generative Design: A Paradigm for Design Research," Futureground—DRS International Conference, Melbourne, Australia, Nov. 17–21.

[29] Patel, J., and Campbell, M. I., 2010, "An Approach to Automate and Optimize Concept Generation of Sheet Metal Parts by Topological and Parametric Decoupling," ASME J. Mech. Des., **132**(5), p. 051001.

[30] Heidari, N., and Iosifidis, A., 2025, "Geometric Deep Learning for Computer-Aided Design: A Survey," IEEE Access, **13**, pp. 119305–119334.

[31] Krish, S., 2011, "A Practical Generative Design Method," Comput.-Aided Des., **43**(1), pp. 88–100.

[32] Price, M., Zhang, W., Friel, I., Robinson, T., McConnell, R., Nolan, D., Kilpatrick, P., Barbhuiya, S., and Kyle, S., 2022, "Generative Design for Additive Manufacturing Using a Biological Development Analogy," J. Comput. Des. Eng., **9**(2), pp. 463–479.

[33] Li, X., Xie, C., and Sha, Z., 2022, "A Predictive and Generative Design Approach for Three-Dimensional Mesh Shapes Using Target-Embedding Variational Autoencoder," ASME J. Mech. Des., **144**(11), p. 114501.

[34] Wang, Z., Melkote, S., and Rosen, D. W., 2023, "Generative Design by Embedding Topology Optimization Into Conditional Generative Adversarial Network," ASME J. Mech. Des., **145**(11), p. 111702.

[35] Wang, Z., Yan, X., Melkote, S., and Rosen, D., 2025, "McGAN: Generating Manufacturable Designs by Embedding Manufacturing Rules Into Conditional Generative Adversarial Network," Adv. Eng. Inform., **64**, pp. 103074.

[36] Wang, H., Zhao, M., Wang, Y., Quan, W., and Yan, D.-M., 2024, "VQ-CAD: Computer-Aided Design Model Generation With Vector Quantized Diffusion," Comput. Aided Geom. Des., **111**, p. 102327.

[37] Peckham, O., Raines, J., Bulsink, E., Goudswaard, M., Gopsill, J., Barton, D., Nassehi, A., and Hicks, B., 2025, "Artificial Intelligence in Generative Design: A Structured Review of Trends and Opportunities in Techniques and Applications," Designs, **9**(4), p. 79.

[38] Li, X., Wang, Y., and Sha, Z., 2023, "Deep Learning Methods of Cross-Modal Tasks for Conceptual Design of Product Shapes: A Review," ASME J. Mech. Des., **145**(4), p. 041401.

[39] Yu, N., Ferdous Alam, M., Hart, A. J., and Ahmed, F., 2026, "GenCAD-Three-Dimensional: Computer-Aided Design Program Generation Using Multimodal Latent Space Alignment and Synthetic Dataset Balancing," ASME J. Mech. Des., **148**(3), p. 031703.

[40] Li, X., Sun, Y., and Sha, Z., 2024, "LLM4CAD: Multimodal Large Language Models for Three-Dimensional Computer-Aided Design Generation," ASME J. Comput. Inf. Sci. Eng, **25**(2), p. 021005.

[41] Makatura, L., Foshey, M., Wang, B., Hähnlein, F., Ma, P., Deng, B., Tjandrasuwita, M., et al., 2024, "How Can Large Language Models Help Humans in Design and Manufacturing?" Harvard Data Science Review(Special Issue 5).

[42] Sun, Y., Li, X., and Sha, Z., 2025, "Large Language Models for Computer-Aided Design Fine Tuned: Dataset and Experiments," ASME J. Mech. Des., **147**(4), p. 041710.

[43] Wu, Z., Wang, Z., Cheng, J. C. P., and Kwok, H. H. L., 2024, "A Knowledge-Informed Optimization Framework for Performance-Based Generative Design of Sustainable Buildings," Appl. Energy, **367**, p. 123318.

[44] Tkachenko, A., and Wang, C. J., 2024, "Analysis of Engineering Structures and Components Created Using the Generative Design," J. Phys.: Conf. Ser., **2762**(1), p. 012089.

[45] Buonamici, F., Carfagni, M., Furferi, R., Volpe, Y., and Governi, L., 2020, "'Generative Design': An Explorative Study," Comput-Aid. Des. Appl., **18**(1), pp. 6–10.

[46] Pilagatti, A. N., Atzeni, E., and Salmi, A., 2023, "Exploiting the Generative Design Potential to Select the Best Conceptual Design of an Aerospace Component to Be Produced by Additive Manufacturing," Int. J. Adv. Manuf. Technol., **126**(11), pp. 5597–5612.

[47] Kim, J., Kwon, Y., and Kang, N., 2024, "Deep Generative Design for Mass Production." 10.48550/arXiv.2403.12098

[48] Patterson, A. E., and Allison, J. T., 2022, "Mapping and Enforcement of Minimally Restrictive Manufacturability Constraints in Mechanical Design," ASME Open J. Engineering, **1**, p. 014502.

[49] Cao, J., Vakaj, E., Soman, R. K., and Hall, D. M., 2022, "Ontology-Based Manufacturability Analysis Automation for Industrialized Construction," Autom. Constr., **139**, p. 104277.

[50] Raines, J., Barton, D., and Hicks, B., 2025, "Scaling Generative Design for Production Through the Use of Standard Parts," Proc. Des. Soc., **5**, pp. 2831–2840.

[51] Campbell, M. I., Nair, S., and Patel, J., 2007, "A Unified Approach to Solving Graph Based Design Problems," IDETC-CIE, Las Vegas, NV, Sept. 4–7, pp. 523–535.

[52] Soman, A., Padhye, S., and Campbell, M. I., 2003, "Toward an Automated Approach to the Design of Sheet Metal Components," AI EDAM, **17**(3), pp. 187–204.

[53] Barda, A., Tevet, G., Schulz, A., and Bermano, A. H., 2023, "Generative Design of Sheet Metal Structures," ACM Trans. Graph., **42**(4), pp. 116:1–116:13.