Integrating Neurosymbolic Systems in Advanced Product Design: A Comprehensive Review

Journal Title
XX(X):2–24

Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Siamak Ghodsi[1,2], Gollam Rabby[1,2], Felix Engel[2], Sven Münker[3] and Sören Auer[1,2]

Abstract

Neurosymbolic AI combines neural network adaptability with symbolic reasoning, offering a framework for addressing the complex multi-objective optimization, constraint satisfaction, and interpretability challenges in advanced product design. This survey synthesizes over two decades of research on NS systems applied to advanced product design, spanning generative CAD, manufacturing optimization, and sustainability-driven design. Our analysis reveals a fragmented research landscape, where approaches, including reinforcement learning, show potential for verifiable design automation but still face hurdles in scalability, real-time constraint verification, and integration with CAD-to-manufacturing workflows. To organize the field, we classify current integration paradigms, present representative methods across application domains, and outline future research directions for integrating these approaches into robust, explainable design systems. By linking advanced AI techniques to the operational demands of product design, this work provides a concise roadmap for researchers and practitioners developing next-generation intelligent design automation solutions.

Keywords

Neurosymbolic AI, Product Design, Manufacturing, CAD, Knowledge Graphs, Constraint Satisfaction, Survey

Introduction

Product design has evolved from manual drafting and rule-based Computer-Aided Design (CAD) to data-driven generative methods powered by neural networks. While deep learning has revolutionized design space exploration through automatic feature extraction, generative synthesis, and performance prediction, purely neural approaches lack interpretability and cannot guarantee constraint satisfaction or formal verification of physical and manufacturing requirements such as structural integrity, thermal limits, or manufacturability. This gap has catalyzed interest in neurosymbolic systems (NS) that combine neural generalization with verifiable symbolic reasoning, representing the third wave of intelligent design automation d'Avila Garcez and Lamb (2023).

Modern product design across aerospace, electronics, automotive, and industrial domains requires simultaneously optimizing conflicting objectives, minimizing weight while maximizing strength, reducing cost while improving performance, subject to hard constraints: safety standards, manufacturing tolerances, material availability, and regulatory compliance Roijers et al. (2013). Neural networks excel at learning design-performance mappings but cannot inherently enforce constraints or explain decisions. Symbolic systems rigorously verify requirements but lack the capacity to discover novel patterns in high-dimensional data. Neurosymbolic Artificial Intelligence (NS-AI) integrates neural pattern recognition with symbolic constraint reasoning, enabling design systems that are both data-informed and provably correct.

NS-AI offers the potential to learn from large-scale design databases (e.g., CAD repositories, simulation results, and manufacturing records) while enforcing domain-specific rules through symbolic layers. This approach supports the entire design lifecycle, from conceptual generation and topology optimization to engineering analysis and manufacturing planning. It balances multiple objectives on the Pareto front while satisfying hard constraints. NS reinforcement learning iteratively optimizes design parameters, manages objective tradeoffs, and provides human-readable explanations for decisions. Embedding geometric constraints, material properties, and process capabilities into symbolic layers alongside adaptive neural policies enables continuous refinement with formal guarantees. Yet developing robust architectures for high-dimensional design spaces, real-time constraint verification, multi-stakeholder preferences, and flawless CAD-to-manufacturing workflows remain active research challenges.

[1]L3S Research Center, Leibniz University of Hannover, Germany
[2]TIB – Leibniz Information Centre for Science and Technology, Hannover, Germany
[3]Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Germany

Corresponding author:
Siamak Ghodsi, L3S Research Center, Leibniz University of Hannover, Appelstraße 9a, 30167 Hannover, Germany.
Email: ghodsi@l3s.de

While NS-AI and multi-objective reinforcement learning (MORL) have been extensively studied in isolation Besold et al. (2021); Hayes et al. (2022), their integration for product design remains fragmented. Existing surveys primarily address general NS architectures Yu et al. (2023); d'Avila Garcez and Lamb (2023); Colelough and Regli (2024), abstract MORL formulations Roijers et al. (2013), or domain-agnostic NS-RL frameworks Acharya et al. (2024), without systematically examining how symbolic constraints, multi-objective trade-offs, and verification requirements interact within end-to-end product design workflows. In particular, prior NS-RL studies predominantly focus on single-objective or safety-constrained control, whereas MORL research rarely incorporates explicit symbolic reasoning or design-rule verification, both of which are critical for engineering applications. This review addresses this gap by synthesizing neuro-symbolic integration patterns specifically in the context of product design, from generative CAD and structural optimization to manufacturing and assembly, with an emphasis on verifiability, interpretability, and Pareto-efficient decision-making.

Abbreviations

Table 1 summarizes abbreviations and notation used throughout this survey. All acronyms are defined at first use in the main text.

Survey Methodology

We conducted a structured literature review (SLR) to identify and synthesize neurosymbolic (NeSy) methods for product design, spanning generative CAD, manufacturing and process optimization, assembly/disassembly planning, and sustainability-driven design. The protocol follows a PRISMA-inspired workflow adapted to engineering and AI literature; the study selection process is summarized in Figure 1.

Living resources (ORKG and repository). To improve transparency and facilitate community reuse, we maintain a queryable, structured comparison of the reviewed methods in the Open Research Knowledge Graph *(ORKG)*[*]. An accompanying *GitHub*[†] repository provides a living bibliography (Awesome-style list), metadata artifacts, and supporting material. These resources may extend beyond the static snapshot discussed in the manuscript, while the SLR corpus analyzed in this paper consists of **55** included papers.

Research scope and questions. The review targets methods that integrate *symbolic substrates* (e.g., logic, rules, constraints, ontologies, knowledge graphs, planners, solvers) with *neural components* (e.g., perception, generation, optimization, surrogate modeling) in product design pipelines. Concretely, we ask: (i) which NeSy integration patterns are used across design subdomains, (ii) what symbolic substrates and neural functions are combined, (iii) which objectives and constraints are supported (e.g.,

[*] orkg.org/comparisons/R1569149
[†] github.com/SiamakGhodsi/NS-Product-Design-Survey

Table 1. Abbreviations and notation used in this survey.

Abbrev.	Meaning	Abbrev.	Meaning
General terms			
AI	Artificial Intelligence	CAD	Computer-Aided Design
NS / NeSy	Neuro-Symbolic (interchangeable)	NS-AI	Neuro-Symbolic Artificial Intelligence
KG	Knowledge Graph	CSP	Constraint Satisfaction Problem
PDE	Partial Differential Equation	PLM	Product Lifecycle Management
Machine learning and neural networks			
RL	Reinforcement Learning	MORL	Multi-Objective Reinforcement Learning
NS-RL	Neuro-Symbolic Reinforcement Learning	MDP	Markov Decision Process
CNN	Convolutional Neural Network	GNN	Graph Neural Network
VAE	Variational Autoencoder	GAN	Generative Adversarial Network
LLM	Large Language Model	VLM	Vision-Language Model
PINN	Physics-Informed Neural Network	TINN	Taxonomy-Informed Neural Network
Symbolic AI and formal methods			
SAT	Boolean Satisfiability	SMT	Satisfiability Modulo Theories
QP	Quadratic Programming	DSL	Domain-Specific Language
DL	Description Logic	OWL	Web Ontology Language
OWL 2 EL	OWL 2 EL profile (lightweight DL)	OWL 2 DL	OWL 2 DL profile (expressive DL)
RDFS	RDF Schema		
Evaluation and methodology			
SLR	Systematic Literature Review	PRISMA	Preferred Reporting Items for Systematic Reviews
ORKG	Open Research Knowledge Graph	MAE	Mean Absolute Error
PF	Pareto Front	HV	Hypervolume indicator
Regulatory			
GDPR	General Data Protection Regulation	CCPA	California Consumer Privacy Act
Table 3 notation — Domains (Dom)			
CAD	Computer-Aided Design	Mfg	Manufacturing / production
Asm	Assembly planning	Mat	Materials
Topo	Topology optimization	Dsgn	Design (general)
G	General benchmark (not product-design-specific)		
Table 3 notation — Neural functions (NF)			
P	Perception / feature extraction	Gen	Generation
Opt	Decision / optimization / control	S	Surrogate / prediction
Table 3 notation — Symbolic substrates (SS)			
O	Ontology	K	Knowledge graph
L	Logic / formal specification	C	Constraints / physics
R	Rules		
Table 3 notation — Other			
(pd)	Paper-defined dataset/benchmark	NR	Not reported as directly comparable scalar

manufacturability, safety, sustainability, cost/time), and (iv) how systems are evaluated (datasets, simulators, benchmarks, case studies).

Information sources and search strategy. We queried major scholarly databases and publisher portals commonly used in AI and engineering research, including Google Scholar and Semantic Scholar, as well as ACM Digital Library, IEEE

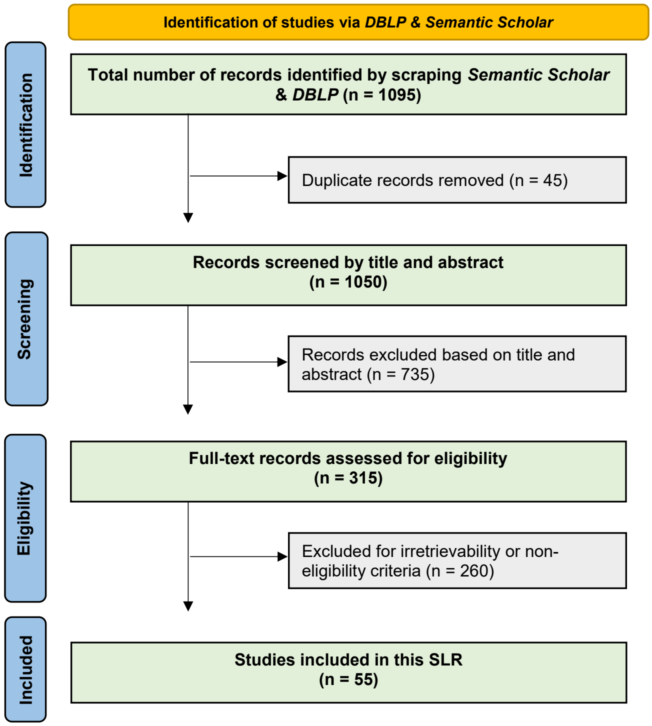

Figure 1. PRISMA-style flow diagram summarizing identification, screening, eligibility assessment, and inclusion for the SLR (final SLR corpus: 55 papers).

Xplore, SpringerLink, Elsevier/ScienceDirect, and arXiv. Searches were executed using combinations of keywords covering (a) the NeSy dimension and (b) the product design dimension. Example queries included: *(neuro-symbolic OR neurosymbolic OR neural-symbolic OR "knowledge graph" OR ontology OR logic OR constraint)* AND *(product design OR CAD OR "computer-aided design" OR manufacturing OR "process planning" OR assembly OR disassembly OR topology optimization)*. Backward and forward snowballing was performed on highly relevant survey and cornerstone papers to reduce the risk of missing influential work.

Eligibility criteria. We included peer-reviewed conference/journal papers and well-established preprints that: (1) address a product-design-related task (e.g., CAD synthesis, manufacturing planning/optimization, assembly planning, topology optimization, materials selection), and (2) contain an explicit integration of symbolic structure with neural learning/inference (or a clearly defined hybrid architecture). We excluded papers that: (i) are purely symbolic expert systems without learning, (ii) are purely neural models without a symbolic component, (iii) focus on generic NeSy reasoning without a meaningful connection to product design, or (iv) provide insufficient technical detail to assess the integration mechanism. Only English-language publications were considered.

Screening and selection process. Records were first de-duplicated, then screened by title and abstract for topical fit. Remaining papers underwent full-text assessment against the inclusion criteria. When multiple versions existed (e.g., arXiv and later published), we retained the most complete peer-reviewed version. The resulting selection flow is reported in Figure 1.

Data extraction and coding. For each included paper we extracted: design domain, symbolic substrate(s), neural function(s), integration pattern (following Kautz's taxonomy), objectives/constraints, required inputs (data modalities, CAD representations, simulators), evaluation setting (benchmarks, case studies), and reported limitations. Extraction was performed using a standardized template and coding schema. The corresponding extraction template, coding fields, and structured records (including ORKG-aligned metadata) are provided in the companion repository.

Synthesis and consistency checks. We synthesized findings along three axes: (1) *integration pattern* (Types I–VI), (2) *symbolic substrates* and *neural functions*, and (3) *product design subdomains*. Ambiguous cases were resolved by re-reading the method and evaluation sections and cross-checking terminology across related work.

Preliminaries

This section establishes foundational paradigms underpinning NS systems in product design. We examine symbolic reasoning approaches that provide constraint verification but lack learning capabilities; neural learning methods that enable data-driven optimization but cannot guarantee formal correctness; and then NS integration paradigms that combine both to address their complementary limitations.

Symbolic Reasoning for Product Design

Symbolic AI, dominant in engineering automation from the 1970s through the 1990s, provides explicit knowledge representation, logical inference, and formal verification, all of which are essential for constraint-driven design. These methods enable transparent reasoning and correctness guarantees but require extensive manual knowledge engineering and cannot adapt from experience Nardi and Brachman (2003).

Ontologies and Knowledge Representation. Description logics provide formal foundations for ontology-based knowledge bases and taxonomies in product design, enabling precise specification of material properties, geometric relations, manufacturing capabilities, and compatibility constraints Nardi and Brachman (2003). Ontology-based systems encode hierarchical classifications in which metallic alloys, for example, inherit properties from parent metal classes while defining specific characteristics, thereby facilitating automated material selection and process compatibility checking Golpayegani et al. (2024); Kim et al. (2006). However, constructing comprehensive ontologies requires substantial domain expertise, and

Table 2. Comparative capabilities of symbolic, neural, and neuro-symbolic (NS) paradigms for product design. Ratings are qualitative: *Strong* indicates the capability is typically native or enforceable by design; *Medium* indicates partial support or strong dependence on task setup; *Weak* indicates the capability is not reliably achieved without substantial additional machinery; *None* indicates the paradigm does not provide the capability in its standard form. OOD stands for out-of-distribution

Capability	Symbolic	Neural	NS
Hard constraint satisfaction	Strong	Weak	Strong
Learning from data	None	Strong	Strong
Perception from unstructured inputs	Weak	Strong	Strong
Interpretability / rationale	Strong	Weak	Strong
Robustness to noise / uncertainty	Weak	Medium	Strong
OOD generalization / novelty	Medium	Weak/Medium	Medium/Strong
Verification / certification readiness	Strong	Weak	Medium/Strong
Data efficiency	Medium	Weak/Medium	Medium/Strong
Computational scalability	Medium	Strong	Medium
Knowledge maintenance cost	High	Low	Medium

maintaining them as materials and standards evolve demands continuous manual curation.

Knowledge Graphs (KG) extend ontological frameworks by capturing historical context and design rationale Hao et al. (2021), linking design parameters to performance outcomes, component relationships, and engineering justifications. This enables traceability for regulatory compliance and design reuse through the identification of similar solutions. However, their rigid structure challenges the incorporation of uncertain or probabilistic relationships.

Constraint Satisfaction and Formal Verification. Constraint satisfaction problems (CSPs) ensure designs meet hard requirements before manufacturing. CSP solvers verify that geometric tolerances, material stress limits, and assembly sequences are satisfied simultaneously Rosell (2004). These formal methods provide mathematical correctness guarantees crucial for safety-critical applications, but suffer from scalability limitations as design spaces grow to thousands of parameters and cannot handle perceptual tasks such as CAD model parsing or defect detection.

Limitations. Purely symbolic approaches (including rule-based expert systems and planning-based configurators) rely on sufficiently complete and up-to-date domain knowledge specified upfront. This assumption is often unrealistic in product design, where requirements evolve, suppliers change, and new materials/processes emerge. Moreover, symbolic systems do not naturally leverage large-scale historical data or simulation logs for generalization; instead, new scenarios typically require manual knowledge engineering and rule maintenance. Finally, under partial observability or noisy sensing, purely symbolic pipelines can be brittle, often failing to produce feasible solutions rather than degrading gracefully. To position these limitations in the broader landscape, Table 2 contrasts symbolic, purely neural, and neuro-symbolic paradigms across recurring design capabilities and trade-offs.

Neural Reasoning for Product Design

Neural networks have transformed product design by learning complex patterns directly from data. They automatically extract features from design repositories, simulation databases, and sensor streams, enabling advances in generative design, performance prediction, and multi-objective optimization Kusiak (2020). Yet, as black-box function approximators, they provide neither formal guarantees nor the traceable rationale that engineering decisions typically require.

Perception and Feature Learning. Convolutional neural networks (CNNs) learn aesthetic quality and functional form relationships from design databases without hand-crafted feature engineering Heidari and Iosifidis (2025). Modern CNNs process 3D CAD models, point clouds, and mesh representations for design similarity retrieval, manufacturability assessment, and defect detection. Transformer-based architectures such as vision-language models extend these capabilities by mapping rendered CAD views to structured design programs. OpenECAD, for example, generates executable sketch and 3D CAD operation sequences from images, producing fully editable CAD models Yuan et al. (2024).

Generative Design. Generative models (LLMs, VAEs, GANs, diffusion models) enable automatic synthesis of novel geometries satisfying learned patterns Alam and Ahmed (2025); Li et al. (2025b). These models learn compressed latent representations, allowing interpolation between known designs and exploration of unconsidered solutions. Graph neural networks extend generation to topological structures such as assemblies or lattices. However, these models typically optimize reconstruction or perceptual similarity losses and do not enforce manufacturing constraints or certification rules; generated CAD is therefore *plausible* rather than guaranteed feasible, and must be post-filtered by symbolic checkers or human experts.

Physics-Informed Neural Networks (PINNs) integrate conservation laws and boundary conditions into their loss functions, providing fast surrogates for stress analysis, thermal simulation, and other PDE-governed use-cases Cuomo et al. (2022). However, neural surrogates are prone to overfitting and provide approximations without formal error bounds. Thus, they may fail on out-of-distribution designs.

Multi-Objective Reinforcement Learning: (MORL) learns policies that trade off cost, performance, weight, and manufacturability by approximating Pareto fronts Hayes et al. (2022). In product design, this supports automated exploration of design variants under competing objectives. However, purely neural MORL policies treat constraints as soft penalties or hidden rewards and cannot guarantee to satisfy all safety, tolerance, or regulatory conditions without additional mechanisms. Moreover, they lack interpretability.

Limitations. Black-box neural models share three critical weaknesses: 1) Lacking explicit design-rule representations and inability to provide structured explanations required for auditability and certification. 2) Generative models may produce geometries that violate manufacturing constraints since these constraints are not embedded in the training objectives. 3) Most critically, neural systems optimize expected performance over training distributions but offer no formal correctness guarantees, especially under distribution shift.

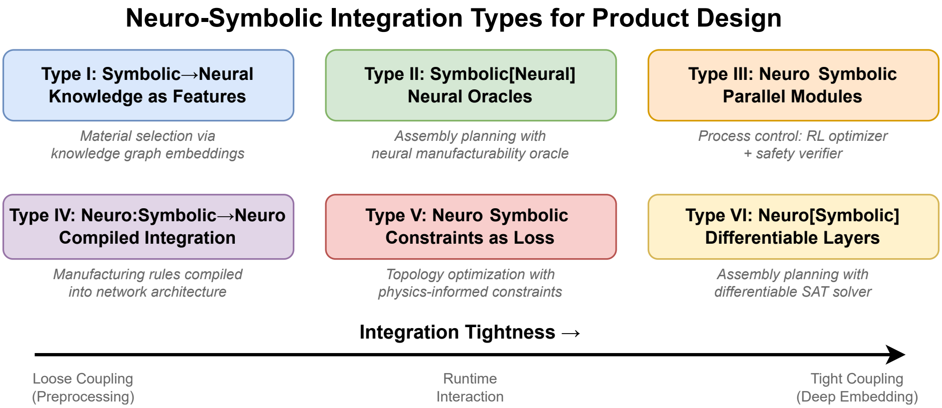

Figure 2. Neurosymbolic integration types (after Kautz), from loose to tight coupling: I neural features, II neural oracles, III parallel neural and symbolic modules, IV compiled rules, V constraints-as-loss, VI differentiable symbolic layers.

Neurosymbolic Integration Paradigms: Product Design Perspective

Neurosymbolic AI integrates neural learning with symbolic reasoning to overcome complementary limitations. Where symbolic systems provide verifiable constraint enforcement but lack learning capabilities, and neural networks enable data-driven optimization but cannot guarantee correctness, NS approaches achieve systems that learn from experience while maintaining formal guarantees.

Integration Taxonomies. Kautz's taxonomy Kautz (2022) categorizes NS systems into six types (Figure 2): *Type I* embeds symbolic knowledge as neural features; *Type II* uses neural oracles within symbolic control; *Type III* maintains parallel modules with structured exchange; *Type IV* compiles symbolic rules into networks; *Type V* incorporates constraints as differentiable losses; *Type VI* interleaves symbolic layers in neural forward passes. Yu et al. propose an orthogonal classification based on information flow Yu et al. (2023): learning for reasoning, reasoning for learning, and fully integrated learning-reasoning systems.

Capabilities and Trade-offs. These integration paradigms enable design systems that learn from data while maintaining verifiable constraint satisfaction and interpretable rationale. Building on the cross-paradigm comparison in Table 2, we next relate each integration type to common design settings. Type II architectures are suitable for scenarios where symbolic planning predominates; Type V methods apply when constraints can be approximated differentiably; Type VI designs enable real-time constraint checking within neural control loops.

Table 3. Neuro-symbolic methods (Types I–VI) relevant to product design. Domain indicates the *evaluation domain*: CAD (computer-aided design), Mfg (manufacturing/production), Asm (assembly planning), Mat (materials), Topo (topology optimization), G (general benchmark, i.e., not evaluated in a product-design workflow), Dsgn (Design), RL (reinforcement learning). NF (neural function): P (perception/extraction), Gen (generation), Opt (decision/optimization/control), S (surrogate/prediction). SS (symbolic substrate): O (ontology), K (knowledge graph), L (logic/specification), C (constraints/physics), R (rules). (Dataset/Benchmark): (pd) short for (paper-defined) NR = not reported as a single directly comparable scalar in the paper (or would require extra space to define fairly). A more comprehensive, extensible, and dynamically updated comparison is available in the Open Research Knowledge Graph (ORKG mentioned in Methodology section).

Method	Type	Dom	NF	SS	Constraint / Symbolic Prior	Dataset / Benchmark	Key metrics
Periodic GT Yan et al. (2022)	I	Mat	S	C	crystal periodicity / structure priors	materials property benchmarks	MAE (property prediction), Infer. Time
Onto-Speech Macioł et al. (2025)	I	Mfg	P	O	ontology-guided IE schema	production disturbance speech logs	Information Extraction quality (NR)
Onto-KR Yang et al. (2023)	I	Mfg	–	O	workflow ontology representation	InPro ontology in industrial knowledge base (5M data)	Consistency, Clarity, Coverage, Adaptability
MatKG+RL Huang et al. (2024)	I	Mat	S	K	element-attribute KG	MatKG / multimodal atom data	Test-MAE, Ablation (#layers)
TINN Terziyan and Vitko (2023)	II	Mfg	P	O	instance-class rel, Industry 4.0 Ontology	multi-source signal fusion (pd)	Lehmer mean loss, robustness
NeSyGeo Wu et al. (2025)	II	RL	Gen	L/O	Geo-DSL + reasoner/verifier	NeSyGeo-Test (2668 Q&A pairs)	generation quality, correctness
LLM4CAD Li et al. (2025a)	II	CAD	Gen	L/C	CAD command (parametric scripts)	Synt: CAD model, sketch, render-img	Parse rate, Intersection/union accuracy
OpenECAD Yuan et al. (2024)	II	CAD	Gen	C/L	CAD command syntax + sketch	SketchGraphs/Fusion360/DeepCAD; test-set: 57 designs	100-pt score; #executable/ #correct; inference time
Onto-RL Golpayegani et al. (2024)	III	Mfg	Opt	O	ontology-based adaptive reward machine	sustainable job-shop scheduling (pd)	resource/throughput/waiting-time
ASAP Tian et al. (2024)	III	Asm	Opt	C	physics feasibility (collision+stability)	Fusion360 (train-test: 1,906-240)	success rate (%); runtime vs #parts
Shield (MDP) Carr et al. (2023)	III	G	Opt	L	formal shield / safety specification	sparse-reward POMDP benchmarks	safety+return/convergence
Shield Odriozola-Olalde et al. (2025)	III	G	Opt	L	adaptive safety const. for robust shields	Control environments: e.g., navigation	safety+performance: efficiency, reward
INSIGHT Luo et al. (2024)	III	G	Opt	L	symbolic policy repr. (equation learner)	Nine Atari games: Arcade Learning Env	game score + textual explanation (NR)
3 Pathway Graf and Emami (2024)	III	G	Opt	L	interpretable/differentiable logic policies	Simul. building energy/ HVAC cntrl env	cost: energy + comfort-penalty, temp
SPRING Jacobson and Xue (2025)	IV	Dsgn	Gen	R	symbolic spatial CSP (sampling/search)	Interior design generation benchmarks	constraint satisfaction + quality
McGAN Wang et al. (2025)	IV	Mfg	Gen	R	Manufacturing rules embedded in cGAN	2D part designs for injection molding	rule-satisfaction; quality; robustness
DL2 Fischer et al. (2019)	V	G	Opt/S	L	first-order logic loss constraint	Image (MNIST, FASHION, CIFAR-10)	Constraint satisfaction rate, accuracy
Neurcadrecon Dong et al. (2024)	V	CAD	S	C	zero Gaussian curvature prior	CAD surface reconstruction (pd)	Chamfer dist, F1-score, Consistency
GINN Berzins et al. (2025)	V	Topo	S	C	Geometric & topological design const.	PDE/geometry tasks, 3D jet-engine (pd)	Constraint-satisfaction errors; diversity
PINN-Topo Jeong et al. (2023)	V	Topo	Opt/S	C	Structural equilibrium & boundary cond.	2D-3D structural topology opt. problems	compliance; volume fraction
OptNet Amos and Kolter (2017)	VI	G	Opt	C	differentiable quadratic program layer	MNIST, 4x4 Sudoku task	loss convergance, runtime, MSE
SATNet Wang et al. (2019)	VI	G	Opt	L	differentiable MAXSAT layer	Sudoku, parity sequences, visual Sudoku	clause satisfaction (solve rate), accuracy
CVXPY Agrawal et al. (2019)	VI	G	Opt	C	differentiable convex program	logistic stochastic control policies (pd)	runtime, control cost, prediction loss
NURBS-Diff Prasad et al. (2022)	VI	CAD	Opt/S	C	Geometric: NURBS curves and surfaces	point-cloud reconst., CAD geometry	accuracy, GPU efficiency evaluation

Formal Characterization. Neurosymbolic product design can be formally characterized as constrained optimization over a hybrid space:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{X}} \; f_\theta(\mathbf{x}) \quad \text{subject to} \quad \Phi(\mathbf{x}) \models \top \tag{1}$$

where $f_\theta : \mathcal{X} \to \mathbb{R}$ denotes a neural objective (e.g., a learned performance surrogate or policy value), $\mathcal{X}$ is the design space (continuous parameters, discrete configurations, or both), and $\Phi(\mathbf{x}) \models \top$ asserts that the symbolic specification Φ—a conjunction of constraints $\phi_1 \wedge \phi_2 \wedge \cdots \wedge \phi_k$ encoding geometric validity, manufacturing rules, safety specifications, or ontological consistency—is satisfied by design $\mathbf{x}$. The six integration types in Kautz's taxonomy differ in *how* and *when* the symbolic specification Φ interacts with the neural objective f_θ:

- **Types I–II:** Φ shapes inputs or outputs (preprocessing/postprocessing)
- **Type III:** Φ verified in parallel, filtering or shielding neural proposals
- **Type IV:** Φ compiled into network structure
- **Type V:** Φ encoded as differentiable loss: $\mathcal{L} = f_\theta(\mathbf{x}) + \lambda \cdot \mathcal{L}_\Phi(\mathbf{x})$ where $\mathcal{L}_\Phi \to 0$ iff $\Phi(\mathbf{x}) \models \top$
- **Type VI:** Φ solved within forward pass via differentiable SAT/SMT/QP layers

For multi-objective settings common in product design, f_θ generalizes to a vector-valued function $\mathbf{f}_\theta : \mathcal{X} \to \mathbb{R}^m$, and the goal becomes approximating the constrained Pareto front:

$$\mathcal{P}^* = \{\mathbf{x} \in \mathcal{X} \mid \Phi(\mathbf{x}) \models \top \;\wedge\; \nexists \mathbf{x}' : \mathbf{x}' \succ_{\text{Pareto}} \mathbf{x}\} \tag{2}$$

The following section applies this taxonomy to classify existing methods across product design domains, revealing systematic patterns in how different integration types align with specific engineering tasks.

Classification of NeuroSymbolic Methods for Product Design

A comprehensive overview of neurosymbolic methods applied to product design is provided in Table 3, categorizing 24 systems by integration type (I–VI), symbolic substrate, neural function, and evaluation domain. This section analyzes the systematic patterns revealed by this classification.

Distribution of integration types. Types I–III constitute the majority of product-design-evaluated systems in our corpus, whereas Types IV–VI appear more often in general benchmarks (denoted "G") than in end-to-end engineering workflows. This asymmetry reflects practical constraints: loose couplings preserve modularity, enable independent validation of symbolic components, and integrate naturally with existing CAD, simulation, and PLM toolchains. By contrast, tighter integrations (Types V–VI) often require domain-specific differentiable formulations—available for geometric primitives (e.g., NURBS-Diff) and physics losses (e.g., PINNs) but less mature for discrete manufacturing rules or assembly precedence constraints.

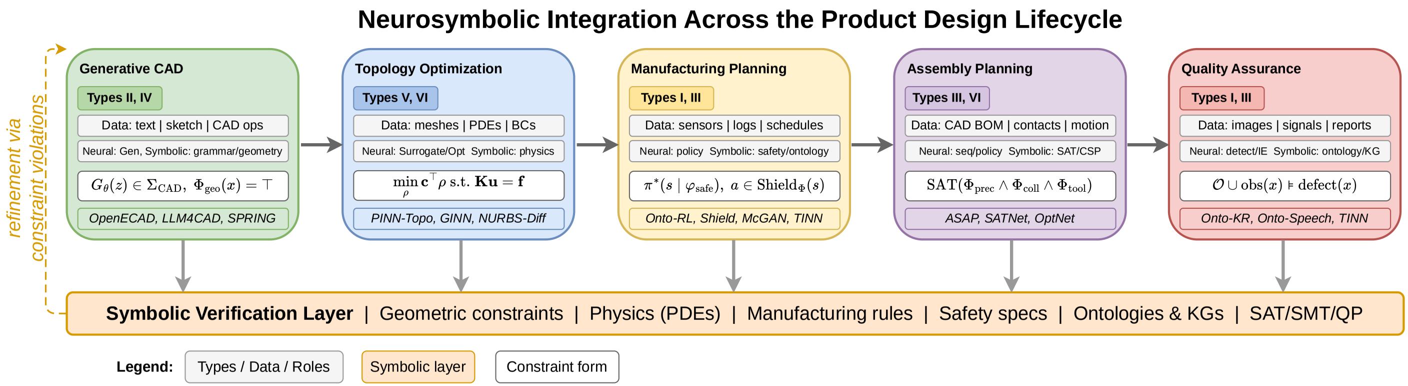

Figure 3. Neurosymbolic integration across the product design lifecycle (survey lens). Stages illustrate typical workflow touchpoints and how a shared *symbolic verification layer*—encoding geometric constraints, physics, manufacturing rules, safety specifications, and ontologies/KGs—interfaces with neural components. Type labels indicate the dominant integration patterns observed in the reviewed corpus (not exclusive, not prescriptive); the dashed feedback arrow denotes iterative refinement driven by constraint violations. Boxed expressions show representative constraint idioms per stage: CAD grammar validity ($G_\theta(\cdot) \in \Sigma_{\mathrm{CAD}}$), PDE-constrained topology optimization ($\mathbf{Ku} = \mathbf{f}$), safety-shielded policies ($\pi^*(\cdot \mid \varphi_{\mathrm{safe}})$), satisfiability of precedence/collision constraints ($\mathrm{SAT}(\Phi_{\mathrm{prec}} \wedge \Phi_{\mathrm{coll}})$), and ontology-based defect inference ($\mathcal{O} \models \mathrm{defect}(x)$).

Lifecycle view. To contextualize the taxonomy within an engineering workflow, Figure 3 maps the dominant integration patterns onto canonical product design stages. The figure is not a normative pipeline but a *survey lens*: it illustrates where neural and symbolic components typically interact—via constraint checking, shielding, or verification—and why feedback loops arise when designs fail symbolic validation. Across stages, these interactions instantiate the constrained optimization formulation of Equation 1; in multi-objective settings, the learned objective becomes a vector of competing criteria, e.g.,

$$\mathbf{J}(\mathbf{x}) = \big[J_{\text{cost}}(\mathbf{x}),\ J_{\text{time}}(\mathbf{x}),\ J_{\text{CO}_2}(\mathbf{x}),\ -J_{\text{quality}}(\mathbf{x})\big], \tag{3}$$

and the goal is to approximate the constraint-feasible Pareto set $\mathcal{P}^*$ (Equation 2).

Symbolic substrates and neural functions. Table 3 also reveals recurring pairings between design tasks and symbolic–neural components. Constraint-based substrates (C) dominate topology optimization and CAD reconstruction, where physical laws admit differentiable formulations. Ontology-based substrates (O) appear primarily in manufacturing contexts requiring interoperability with industrial knowledge bases (e.g., Industry 4.0 taxonomies). Logic and rule substrates (L, R) concentrate in safety-critical shielding and rule-guided generation, where explicit guarantees must be enforced. On the neural side, generative functions (Gen) cluster in CAD and design synthesis, while optimization functions (Opt) dominate RL-based process control and assembly planning.

Domain-specific patterns. Across application domains, generative CAD emphasizes programmatic representations and editability, topology optimization consistently incorporates physical priors, and manufacturing and assembly planning prioritize symbolic feasibility and safety constraints. Together, these patterns highlight both the diversity of neuro-symbolic design strategies and the lack of standardized, cross-domain evaluation protocols.

Evaluation gaps. A key limitation visible in Table 3 is the heterogeneity of evaluation protocols: metrics range from geometric fidelity (e.g., Chamfer distance, F1-score) to task success rates, constraint-satisfaction percentages, and domain-specific throughput measures. This diversity impedes cross-method comparison and motivates standardized, multi-objective benchmarks that jointly assess performance, constraint compliance, and interpretability—a gap revisited in section .

Applications of NeuroSymbolic-AI in Product Design

NS-AI is increasingly adopted in advanced product design because it combines data-driven learning with explicit constraint reasoning, enabling solutions that are both adaptive and verifiable. Across the design-to-manufacturing pipeline, NS-AI supports geometry generation, structural optimization, manufacturing planning, and quality assurance while maintaining consistency with engineering rules, physical laws, and regulatory requirements, capabilities that purely neural systems struggle to guarantee.

Generative CAD and Structural Optimization

In generative CAD, neural models such as VAEs, GANs, diffusion models, and LLM-based generators enable rapid exploration of geometric design spaces but frequently produce outputs that violate geometric, manufacturing, or assembly constraints. NS-AI mitigates this limitation by embedding symbolic structure directly into the generation process, for example, through programmatic CAD representations, constraint-aware decoding, or rule-guided generation pipelines. Recent LLM-symbolic hybrid systems translate images or natural language into executable CAD programs rather than static meshes, enabling editability and downstream verification (e.g., OpenECAD, LLM4CAD) Yuan et al. (2024); Li et al. (2025a).

A similar integration is critical in topology and structural optimization, where neural surrogates accelerate simulation-heavy design cycles but struggle to generalize under novel loads or boundary conditions. NS-AI augments these surrogates with physics-informed formulations that incorporate governing PDEs, and with differentiable logical or SAT/QP layers that enforce safety, connectivity, and manufacturability constraints. Examples include geometry- and physics-informed neural networks and differentiable CAD layers that embed curvature, equilibrium, or geometric validity constraints directly into optimization loops Jeong et al. (2023); Dong et al. (2024); Berzins et al. (2025).

Manufacturing Optimization and Material Compatibility

NS-AI also enhances downstream manufacturing and material-related decisions, where safety, feasibility, and interpretability are essential. In manufacturing process optimization spanning additive manufacturing, machining, curing, and thermal control, pure RL agents may explore unsafe operating regions. NS-RL addresses this by combining neural policies with symbolic monitors, temporal logic specifications, or ontology-guided abstractions that restrict exploration to feasible regions and provide interpretable failure explanations Carr et al. (2023); Odriozola-Olalde et al. (2025); Golpayegani et al. (2024).

Material selection poses similarly complex challenges due to coupled mechanical, thermal, chemical, and cost considerations Kusiak (2020). NS-AI supports this task through KG-based representations and logic–neural frameworks that integrate learned compatibility scores with symbolic reasoning over assembly and environmental constraints Huang et al. (2024). For example, element-attribute material knowledge graphs combined with graph or transformer architectures improve property prediction and generalize to sparse or emerging materials, while periodic graph transformers encode crystal symmetries to enhance prediction accuracy on standard materials benchmarks Yan et al. (2022).

Assembly, Design Exploration, and Quality Assurance

Assembly planning and downstream design exploration require both scalability and formal correctness. NS-AI integrates neural sequence or policy models with symbolic constraint solvers—such as differentiable SAT or physics-based feasibility checks—to ensure collision-free, precedence-aware, and tool-accessible assembly sequences.

Robotic assembly systems exemplify this paradigm by combining learned action proposals with symbolic verification of physical feasibility Tian et al. (2024); Wang et al. (2019).

In design space exploration, NS-AI enables reasoning over hybrid spaces to combine discrete architectural decisions with continuous parameters. Symbolic modules govern architecture selection and constraint enforcement, while neural models optimize parametric configurations or approximate Pareto fronts. Multi-objective NS-RL supports transparent trade-off analysis across cost, performance, manufacturability, and sustainability, while symbolic representations provide interpretable explanations for constraint-driven decisions Hayes et al. (2022); Reymond et al. (2022); Luo et al. (2024).

Quality assurance further benefits from NS integration. Neural vision models detect surface or structural anomalies, while symbolic knowledge, often encoded as ontologies or KGs, supports traceable root-cause analysis and compliance checking Yang et al. (2023). Ontology-based workflow models and NS design frameworks additionally structure inspection criteria and non-conformance handling, enabling more explainable and auditable diagnoses in regulated manufacturing workflows Arachchige et al. (2025).

Sustainable and Human–AI Design

NS-AI also supports sustainability-oriented design and human–AI collaboration by integrating reasoning, learning, and explainability. Ontologies and causal KGs structure lifecycle and sustainability assessment, enabling early-stage what-if analysis and transparent evaluation of environmental trade-offs. Multi-objective NS frameworks balance environmental impact, cost, and performance while generating explanations suitable for certification and stakeholder communication Golpayegani et al. (2024); Bougzime et al. (2025).

Human–AI collaboration is enhanced through LLM–symbolic interfaces that translate natural-language requirements into verifiable design constraints. Parallel NS architectures provide real-time feedback during design exploration, while meta-cognitive components detect high-uncertainty scenarios, solicit human guidance, and maintain audit trails linking decisions, constraints, and rationale. This integration supports traceable and accountable design, bridging the gap between automated exploration and human judgment Luo et al. (2024); Kautz (2022).

Cross-Domain Transfer Potential

While previous section surveyed neurosymbolic methods already evaluated within product design workflows, a complementary body of work develops general-purpose NS architectures in adjacent domains, robotics, game playing, energy systems, and combinatorial optimization, that have not yet been specialized to CAD or manufacturing. These systems appear in Table 3 with domain label “G” (general benchmark) because their symbolic–neural integration patterns are directly transferable: the same differentiable solvers, shielding mechanisms, and structured-prior injections that enforce game rules or safety specifications can, with modest adaptation, enforce geometric validity, assembly precedence, or manufacturing feasibility. This section highlights four such transfer-ready architectural patterns.

Differentiable Symbolic Layers (Type VI) enable real-time constraint verification. OptNet Amos and Kolter (2017) and CVXpyLayers Agrawal et al. (2019) demonstrate differentiable quadratic and convex program layers that could optimize CAD parameters under geometric constraints. SATNet Wang et al. (2019) shows that differentiable satisfiability solving can enforce complex logical constraints, suggesting applications to assembly precedence, collision rules, or configuration compatibility.

Explainable NS-RL (Type III) addresses interpretability requirements in regulated industries. INSIGHT Luo et al. (2024) learns symbolic policy representations with textual explanations, while logic tensor networks Graf and Emami (2024) provide interpretable control policies specified in logical form. Adapting these patterns to design and manufacturing would support audit trails for safety-critical decisions, e.g., explaining why a particular process plan or design variant satisfies given constraints.

Safety-Critical Shielding (Type III) ensures formal correctness during learning and execution. Shield-based methods Carr et al. (2023); Odriozola-Olalde et al. (2025) restrict RL exploration to safe regions via temporal-logic specifications and adaptive constraint sets. The same shielding patterns can be applied to machining, thermal processing, or robotic assembly, where symbolic envelopes define safe operating regimes and filter out unsafe actions before execution.

Structured Priors and Rule-Guided Generation (Types II and IV) show how symbolic structure can be injected into perception and generation without full RL. Taxonomy-informed neural networks (TINN) embed an Industry 4.0 ontology into the network architecture to improve robustness and data efficiency in smart-manufacturing settings Terziyan and Vitko (2023). Similarly, SPRING integrates a symbolic spatial constraint solver into an interior-design generator, demonstrating how rule-guided sampling can enforce layout constraints by construction Jacobson and Xue (2025). These patterns are directly transferable to parametric CAD design, assembly layout, or facility planning tasks where taxonomies and spatial rules are already available.

Comparability Caveat: Direct performance comparison across these cross-domain methods remains challenging. Game scores (e.g., INSIGHT on Atari) Luo et al. (2024), control costs (e.g., logic-based HVAC control in Three Pathways) Graf and Emami (2024), and simulated manufacturing throughput (e.g., Onto-RL for sustainable job-shop scheduling) Golpayegani et al. (2024) rely on incommensurable, domain-specific metrics. Section therefore treats these results as qualitative evidence for particular NS integration types rather than as directly comparable numbers, and argues for product-design benchmarks that expose multi-objective, constraint-aware evaluation protocols.

Evaluation Frameworks

NS systems require dual assessment of symbolic correctness and neural performance. Component-level verification tests individual modules, while system-level validation captures emergent integration failures where components function correctly in isolation yet produce constraint violations when combined Renkhoff et al. (2024). Recent evaluations indicate that bidirectional NS architectures outperform unidirectional designs on iterative refinement tasks, yet even high-performing models remain vulnerable to reasoning shortcuts, where spurious correlations replace genuine logical inference Bougzime et al. (2025); Bortolotti et al. (2024).

Correctness and Reasoning Quality

Correctness in NS systems concerns whether symbolic constraints and logical specifications are satisfied, while reasoning quality captures how faithfully models realize intended concepts rather than exploiting shortcuts. Concept-quality metrics quantify symbolic representations through concept accuracy against ground truth and concept completeness (coverage), and are used to expose reasoning shortcuts in NS benchmarks where models rely on superficial cues instead of structure Bortolotti et al. (2024). In product design contexts, similar shortcuts occur when models learn surface-level CAD correlations rather than engineering principles or constraint hierarchies. Temporal-logic satisfaction rates and constraint-violation frequencies per episode or iteration provide complementary views of correctness, but are still predominantly evaluated in post-hoc verification rather than integrated training loops Arachchige et al. (2025); Carr et al. (2023). At the symbolic layer, benchmarking description-logic reasoners across RDFS, OWL 2 EL, and OWL 2 DL highlights scalability and completeness trade-offs that become critical when NS systems rely on large ontologies in design and manufacturing workflows Singh et al. (2025).

Multi-Objective Optimization Performance

Product design involves conflicting objectives, requiring multi-objective evaluation frameworks rather than single-scalar metrics. Hypervolume indicators quantify Pareto-front quality and are standard in MORL and planning Hayes et al. (2022). Preference-controllable approaches and Pareto Conditioned Networks learn policies conditioned on target trade-offs, substantially improving coverage of the Pareto front over linear scalarization while supporting evolving stakeholder preferences Reymond et al. (2022). Symbolic guidance can further increase constraint-satisfaction rates without degrading Pareto quality, but requires careful evaluation of sample efficiency, since NS methods often operate in combinatorial design spaces where prototyping or simulation is expensive Colelough and Regli (2024).

Trustworthiness and Robustness

Systematic reviews indicate that, while most NS studies focus on learning and inference, only a small fraction devote space to trustworthiness evaluation, including robustness, calibration, and failure modes Colelough and Regli (2024). This quantitative gap is compounded by a qualitative one: conventional similarity metrics often fail to predict functional performance, manufacturability, or lifecycle costs, especially in engineering design Regenwetter et al. (2023). Domain-aware evaluation must therefore consider physical realizability, manufacturing constraints, and lifecycle objectives beyond pure geometric or statistical similarity.

Domain-Specific Validation

Addressing these deficiencies requires domain-tailored protocols. In CAD generation, assessment is multi-layered: geometric measures (e.g., Chamfer and Hausdorff distance) capture shape fidelity, topological checks verify Euler-characteristic preservation, and parametric editability measures downstream operation success Willis et al. (2021). Graph-based representations such as SketchGraphs enable evaluation of whether models learn meaningful relational geometry and patterns rather than surface correlations Seff et al. (2020). Manufacturing feasibility combines manufacturability scores with cost–performance trade-offs, motivating joint AI–domain-expert efforts to define ecologically valid benchmarks that reflect real deployment constraints.

Figure 4. Risks and Challenges of Neurosymbolic AI Deployment.

Risk and Challenges

NS-AI holds transformative potential in product design, manufacturing, and autonomous systems; however, it also introduces multiple technological, ethical, and regulatory risks, as illustrated in Figure 4.

Technological Challenges

Data Integration and Interoperability. For the advanced product design, Neurosymbolic AI relies on heterogeneous sensor, simulation, and operational data. Multi-sensor fusion requires temporal alignment and standardized formats to ensure reliable integration, while deployments in existing industrial systems must preserve safety certifications during incremental upgrades. Interoperability remains critical, motivating the adoption of common protocols, interface standards, and modular architectures so that symbolic components, neural surrogates, and control policies can be updated independently without destabilizing end-to-end workflows.

Security Challenges. Hybrid neural-symbolic architectures introduce novel vulnerabilities for advanced product design and manufacturing. Adversarial attacks may target neural perception, while symbolic logic modules can be exploited through mis-specified rules, missing constraints, or ontology tampering Jalaian and Bastian (2023). Ensuring robustness requires stress-testing under diverse operational conditions and modelling adversarial behaviour at both the data and knowledge levels, while computational overheads of monitoring and verification must remain compatible with real-time control requirements.

Lifecycle Management and Scalability. Maintaining operational integrity and certification of NS-enabled design systems requires robust lifecycle management, including version control for both neural models and symbolic components, regression testing, and periodic re-verification against evolving standards. Heterogeneous fleets of machines and design tools with differing sensors, capabilities, and software versions can encounter compatibility issues if NS components are not encapsulated behind stable interfaces. While modular NS architectures support evolution, practical deployment must carefully balance adaptability with the need to preserve validated safety envelopes and regulatory approvals.

Ethical Challenges

Accountability. Assigning responsibility for autonomous NS AI-driven decisions is non-trivial. Failures may involve multiple stakeholders: product designers, model developers, operators, data engineers, and may span both neural and symbolic components, necessitating transparent regulatory frameworks. Recent work on AI accountability emphasizes the need for traceable decision pipelines, clear documentation of assumptions, and governance mechanisms that allocate responsibility across the lifecycle of AI systems Novelli et al. (2024). Moreover, bias in training data, design priors, or encoded rules can propagate unfair outcomes; embedding fairness constraints and ethical auditing protocols into NS workflows is therefore crucial to ensure transparent and equitable decision-making Álvarez et al. (2024).

Transparency and Explainability. Transparency refers to the visibility of a model's internal structures and data flows, while explainability denotes the ability to provide human-understandable justifications for specific decisions. NS-AI can support both by combining explicit symbolic reasoning traces with learned components, enabling post-incident analysis and the reconstruction of design rationale. However, balancing transparency and explainability with performance and scalability remains challenging, as more interpretable architectures or logging requirements may introduce latency and engineering overheads Novelli et al. (2024).

Public Trust and Social Acceptance. Public trust, understood as the confidence held collectively by affected communities rather than individual users, is critical for the adoption of NS-AI in safety-critical and data-intensive product domains. Hybrid neural–symbolic approaches can improve transparency and controllability, but they do not automatically resolve concerns about bias, accountability, or misuse. Sustained collaboration among regulators, manufacturers, workers, and end users is needed to align NS-AI deployments with societal values and legal expectations, and to maintain trust over time Álvarez et al. (2024).

Regulatory and Governance Challenges

Existing certification pathways, such as aviation and automotive safety standards, provide limited guidance for NS architectures that combine learned components with symbolic reasoning. Verification of such systems requires rigorous simulation, stress testing, and formal analysis to demonstrate reliability under rare or extreme scenarios, and to ensure that safety guarantees are preserved as models and knowledge bases evolve. The dual neural–symbolic nature complicates accountability in incident cases, necessitating legal frameworks that clearly delineate responsibilities among developers, data providers, operators, and manufacturers.

In parallel, adaptive data governance frameworks are needed to ensure compliance with regulations such as the GDPR and CCPA, including transparent data use, robust access controls, encryption, anonymization, and comprehensive audit trails. These mechanisms are essential for building trust, supporting regulatory compliance, and maintaining system resilience in dynamic operational environments, particularly when NS models continually learn from operational data.

Conclusions

Neurosymbolic AI holds significant promise for advancing product design by enabling transparent, adaptive, and constraint-aware solutions that remain compatible with performance-driven engineering workflows. Through a systematic synthesis of existing work, this survey shows that current practice is dominated by loosely coupled neuro-symbolic integrations, particularly Types I–III, where symbolic knowledge serves as structured priors, interfaces, or parallel verification modules, while tighter couplings such as constraint-compiling architectures and differentiable solvers (Types IV–VI) remain confined to narrowly structured subproblems. At the same time, empirical results across CAD, manufacturing, and control highlight persistent challenges in scalability, data integration, real-time constraint verification, and CAD-to-manufacturing interoperability, underscoring the need for standardized benchmarks, lifecycle-aware validation, and closer alignment with engineering workflows. Continued interdisciplinary collaboration among AI researchers, design engineers, and regulators is essential to translate these hybrid architectures into verifiable, efficient, and sustainable design automation systems deployed at scale.

Availability of Resources

Living resource vs. static snapshot: Table 3 provides a curated snapshot to support the narrative structure of this review and to remain stable across versions of the manuscript. To enable continuous updates, structured querying, and community extension (e.g., filtering by integration type, domain, substrate, or evaluation evidence), we maintain a living comparison in the Open Research Knowledge Graph (ORKG). The accompanying GitHub repository serves as the curation and dissemination hub (taxonomy figures, "awesome-list" browsing, and contribution workflow), while ORKG remains the canonical, machine-actionable version of the comparison. Thus, the ORKG comparison table provides a possibility for community contribution and updating/extending the table as the literature evolves.

Acknowledgements

This work was supported by the DFG Priority Programme SPP 2443 *Human Decision* (Grant No. 543081196) and the German Ministry of Education and Research (BmBF) for the project KISSKI AI Service Center (01IS22093C).

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Acharya K, Raza W, Júnior CMJMD, Velasquez A and Song HH (2024) Neurosymbolic reinforcement learning and planning: A survey. *IEEE Trans. Artif. Intell.* 5(5): 1939–1953.

Agrawal A, Amos B, Barratt ST, Boyd SP, Diamond S and Kolter JZ (2019) Differentiable convex optimization layers. In: *NeurIPS*. pp. 9558–9570.

Alam MF and Ahmed F (2025) Gencad: Image-conditioned computer-aided design generation with transformer-based contrastive representation and diffusion priors. *Trans. M.L. Res.* .

Álvarez JM, Colmenarejo AB, Elobaid A, Fabbrizzi S, Fahimi M, Ferrara A, Ghodsi S, Mougan C, Papageorgiou I, Lobo PR, Russo M, Scott KM, State L, Zhao X and Ruggieri S (2024) Policy advice and best practices on bias and fairness in AI. *Ethics Inf. Technol.* 26(2): 31.

Amos B and Kolter JZ (2017) Optnet: Differentiable optimization as a layer in neural networks. In: *ICML*. PMLR, pp. 136–145.

Arachchige PJ, Iancu B and Lilius J (2025) A roadmap toward neurosymbolic approaches in AI design. *IEEE Access* 13: 174368–174392.

Berzins A, Radler A, Volkmann E, Sanokowski S, Hochreiter S and Brandstetter J (2025) Geometry-informed neural networks. In: *ICML*, *Proceedings of Machine Learning Research*, volume 267. PMLR / OpenReview.net.

Besold TR, d'Avila Garcez AS, Bader S, Bowman H, Domingos PM, Hitzler P, Kühnberger K, Lamb LC, Lima PMV, de Penning L, Pinkas G, Poon H and Zaverucha G (2021) Neural-symbolic learning and reasoning: A survey and interpretation. In: *Neuro-Symbolic Artificial Intelligence*, *Frontiers in Artificial Intelligence and Applications*, volume 342. IOS Press, pp. 1–51.

Bortolotti S, Marconato E, Carraro T, Morettin P, van Krieken E, Vergari A, Teso S and Passerini A (2024) A neuro-symbolic benchmark suite for concept quality and reasoning shortcuts. In: *NeurIPS*.

Bougzime O, JABBAR S, Cruz C and Demoly F (2025) Evaluating neuro-symbolic AI architectures: Design principles, qualitative benchmark, comparative analysis and results. In: *19th Intern. Conf. on NeuSy Learn. and Reas.*

Carr S, Jansen N, Junges S and Topcu U (2023) Safe reinforcement learning via shielding under partial observability. In: *AAAI*. AAAI Press, pp. 14748–14756.

Colelough BC and Regli W (2024) Neuro-symbolic AI in 2024: A systematic review. In: *LNSAI@IJCAI*, *CEUR Workshop Proceedings*, volume 3819. CEUR-WS.org.

Cuomo S, Cola VSD, Giampaolo F, Rozza G, Raissi M and Piccialli F (2022) Scientific machine learning through physics-informed neural networks: Where we are and what's next. *J. Sci. Comput.* 92(3): 88.

d'Avila Garcez A and Lamb LC (2023) Neurosymbolic AI: the 3rd wave. *Artif. Intell. Rev.* 56(11): 12387–12406.

Dong Q, Xu R, Wang P, Chen S, Xin S, Jia X, Wang W and Tu C (2024) Neurcadrecon: Neural representation for reconstructing CAD surfaces by enforcing zero gaussian curvature. *ACM Trans. Graph.* 43(4): 51:1–51:17.

Fischer M, Balunovic M, Drachsler-Cohen D, Gehr T, Zhang C and Vechev MT (2019) DL2: training and querying neural networks with logic. In: *ICML*, *Proceedings of Machine Learning Research*, volume 97. PMLR, pp. 1931–1941.

Golpayegani F, Ghanadbashi S and Zarchini A (2024) Advancing sustainable manufacturing: Reinforcement learning with adaptive reward machine using an ontology-based approach. *Sustainability* 16(14): 5873.

Graf P and Emami P (2024) Three pathways to neurosymbolic reinforcement learning with interpretable model and policy networks. *CoRR* abs/2402.05307.

Hao J, Zhao L, Milisavljevic-Syed J and Ming Z (2021) Integrating and navigating engineering design decision-related knowledge using decision knowledge graph. *Adv. Eng. Informatics* 50: 101366.

Hayes CF, Radulescu R, Bargiacchi E, Källström J, Macfarlane M, Reymond M, Verstraeten T, Zintgraf LM, Dazeley R, Heintz F, Howley E, Irissappane AA, Mannion P, Nowé A, de Oliveira Ramos G, Restelli M, Vamplew P and Roijers DM (2022) A practical guide to multi-objective reinforcement learning and planning. *Auton. Agents Multi Agent Syst.* 36(1): 26.

Heidari N and Iosifidis A (2025) Geometric deep learning for computer-aided design: A survey. *IEEE Access* 13: 119305–119334.

Huang C, Chen C, Shi L and Chen C (2024) Material property prediction with element attribute knowledge graphs and multimodal representation learning. *CoRR* abs/2411.08414.

Jacobson MJ and Xue Y (2025) Integrating symbolic reasoning into neural generative models for design generation. In: *AAAI*. AAAI Press, p. 28741.

Jalaian B and Bastian ND (2023) Neurosymbolic AI in cybersecurity: Bridging pattern recognition and symbolic reasoning. In: *MILCOM*. IEEE, pp. 268–273.

Jeong H, Batuwatta-Gamage C, Bai J, Xie YM, Rathnayaka C, Zhou Y and Gu Y (2023) A complete physics-informed neural network-based framework for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering* 417: 116401. DOI: https://doi.org/10.1016/j.cma.2023.116401.

Kautz HA (2022) The third AI summer: AAAI robert s. engelmore memorial lecture. *AI Mag.* 43(1): 93–104.

Kim K, Manley DG and Yang HJ (2006) Ontology-based assembly design and information sharing for collaborative product development. *Comput. Aided Des.* 38(12): 1233–1250.

Kusiak A (2020) Convolutional and generative adversarial neural networks in manufacturing. *Int. J. Prod. Res.* 58(5): 1594–1604.

Li X, Sun Y and Sha Z (2025a) LLM4CAD: multimodal large language models for three-dimensional computer-aided design generation. *J. Comput. Inf. Sci. Eng.* 25(1).

Li Y, Lin C, Liu Y, Long X, Zhang C, Wang N, Li X, Wang W and Guo X (2025b) Caddreamer: CAD object generation from single-view images. In: *CVPR*. Computer Vision Foundation / IEEE, pp. 21448–21457.

Luo L, Zhang G, Xu H, Yang Y, Fang C and Li Q (2024) End-to-end neuro-symbolic reinforcement learning with textual explanations. In: *ICML*, *Proceedings of Machine Learning Research*, volume 235. PMLR / OpenReview.net, pp. 33533–33557.

Macioł A, Macioł P, Gumienny G and Wrzała K (2025) A new ontology-based approach to automatic information extraction from speech for production disturbance management. *The Inter. J. Adv. Manufact. Tech.* 136(7): 3735–3752.

Nardi D and Brachman RJ (2003) An introduction to description logics. In: *Description Logic Handbook*. Cambridge University Press, pp. 1–40.

Novelli C, Taddeo M and Floridi L (2024) Accountability in artificial intelligence: what it is and how it works. *AI Soc.* 39(4): 1871–1882.

Odriozola-Olalde H, Zamalloa M, Arana-Arexolaleiba N and Pérez-Cerrolaza J (2025) Towards robust shielded reinforcement learning through adaptive constraints and exploration: The fear field framework. *Eng. Appl. Artif. Intell.* 144: 110055.

Prasad AD, Balu A, Shah H, Sarkar S, Hegde C and Krishnamurthy A (2022) Nurbs-diff: A differentiable programming module for NURBS. *Comput. Aided Des.* 146: 103199.

Regenwetter L, Srivastava A, Gutfreund D and Ahmed F (2023) Beyond statistical similarity: Rethinking metrics for deep generative models in engineering design. *Comput. Aided Des.* 165: 103609.

Renkhoff J, Feng K, Meier-Doernberg M, Velasquez A and Song HH (2024) A survey on verification and validation, testing and evaluations of neurosymbolic artificial intelligence. *IEEE Trans. Artif. Intell.* 5(8): 3765–3779.

Reymond M, Bargiacchi E and Nowé A (2022) Pareto conditioned networks. In: *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), pp. 1110–1118.

Roijers DM, Vamplew P, Whiteson S and Dazeley R (2013) A survey of multi-objective sequential decision-making. *J. Artif. Intell. Res.* 48: 67–113.

Rosell J (2004) Assembly and task planning using petri nets: a survey. *Proceedings of the institution of mechanical engineers, part B: journal of engineering manufacture* 218(8): 987–994.

Seff A, Ovadia Y, Zhou W and Adams RP (2020) Sketchgraphs: A large-scale dataset for modeling relational geometry in computer-aided design. *CoRR* abs/2007.08506.

Singh G, Tommasini R, Bhatia S and Mutharaju R (2025) Benchmarking neurosymbolic description logic reasoners: Existing challenges and a way forward. *Neurosymbolic Artificial Intelligence* 1: 29498732251339943. DOI:10.1177/29498732251339943.

Terziyan VY and Vitko O (2023) Taxonomy-informed neural networks for smart manufacturing. In: *ISM*, *Procedia Computer Science*, volume 232. Elsevier, pp. 1388–1399.

Tian Y, Willis KDD, Omari BA, Luo J, Ma P, Li Y, Javid F, Gu E, Jacob J, Sueda S, Li H, Chitta S and Matusik W (2024) ASAP: automated sequence planning for complex robotic assembly with physical feasibility. In: *ICRA*. IEEE, pp. 4380–4386.

Wang P, Donti PL, Wilder B and Kolter JZ (2019) Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In: *ICML*, *Proceedings of Machine Learning Research*, volume 97. PMLR, pp. 6545–6554.

Wang Z, Yan X, Melkote SN and Rosen D (2025) Mcgan: Generating manufacturable designs by embedding manufacturing rules into conditional generative adversarial network. *Adv. Eng. Informatics* 64: 103074.

Willis KDD, Pu Y, Luo J, Chu H, Du T, Lambourne JG, Solar-Lezama A and Matusik W (2021) Fusion 360 gallery: a dataset and environment for programmatic CAD construction from human design sequences. *ACM Trans. Graph.* 40(4): 54:1–54:24.

Wu W, Wang Z, Ye J, Zhou Z, Li Y and Guo L (2025) Nesygeo: A neuro-symbolic framework for multimodal geometric reasoning data generation. *CoRR* abs/2505.17121.

Yan K, Liu Y, Lin Y and Ji S (2022) Periodic graph transformers for crystal material property prediction. In: *NeurIPS*.

Yang C, Zheng Y, Tu X, Ala-Laurinaho R, Autiosalo J, Seppänen O and Tammi K (2023) Ontology-based knowledge representation of industrial production workflow. *Adv. Eng. Informatics* 58: 102185.

Yu D, Yang B, Liu D, Wang H and Pan S (2023) A survey on neural-symbolic learning systems. *Neural Networks* 166: 105–126.

Yuan Z, Shi J and Huang Y (2024) Openecad: An efficient visual language model for editable 3d-cad design. *Comput. Graph.* 124: 104048.